

Année : 2009

THESE

présentée à

**L'UFR des Sciences et Techniques
de l'Université de Franche-Comté**

pour obtenir le

**GRADE DE DOCTEUR DE L'UNIVERSITE
DE FRANCHE-COMTE**

en Automatique

(Ecole Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques)

**Contribution à la constitution d'équipes de conception
couplant la structuration du projet et le pilotage des compétences**

par

Onanong HLAOITTINUN

Soutenue le **2 septembre 2009**

devant le jury suivant :

Président :

M. Jean-Pierre Campagne, Professeur, INSA, Lyon

Rapporteurs :

M. Laurent Geneste, Professeur, ENIT, Tarbes

M. Jean-Luc Paris, Professeur, IFMA, Clermont-Ferrand

Examineurs :

M. Xavier Boucher, Maître de conférences, HDR, ENSMSE, Saint-Etienne

M. Emmanuel Caillaud, Professeur, Université Louis Pasteur, Strasbourg

M. Stéphane Chrétien, Maître de Conférences, UFC, Besançon

Directeurs de thèse :

Mme. Maryvonne Dulmet, Maître de conférences, HDR, UFC, Besançon

M. Eric Bonjour, Maître de conférences, HDR, UFC, Besançon

REMERCIEMENTS

En premier lieu, je souhaite remercier Laurent Geneste, professeur à l'École Nationale d'Ingénieurs de Tarbes ainsi que Jean-Luc Paris, professeur à l'Institut Français de Mécanique Avancée (IFMA) de Clermont-Ferrand pour avoir accepté d'être les rapporteurs de ces travaux de thèse et pour toutes les remarques constructives qu'ils ont pu émettre et qui m'ont permis d'améliorer la qualité de rédaction de ce mémoire.

Mes remerciements vont aussi à Jean-Pierre Campagne, professeur à l'Institut National des Sciences Appliquées de Lyon, qui m'a fait l'honneur de faire partie du jury et ensuite d'en être le Président.

J'adresse également mes sincères remerciements aux examinateurs, qui ont accepté d'évaluer mes travaux, Xavier Boucher, maître de conférences - HDR à l'École Nationale Supérieure des Mines de St Etienne, Emmanuel Caillaud, professeur à l'Université Louis Pasteur de Strasbourg et Stéphane Chrétien, maître de conférences au département de Mathématiques à l'Université de Franche Comté.

Mes remerciements les plus sincères vont à mes directeurs de thèse, Maryvonne Dulmet, HDR à l'Université de Franche-Comté et Eric Bonjour, HDR à l'Université de Franche-Comté pour m'avoir guidé et accompagné au long de cette étude, pour leur patience, pour la confiance qu'ils m'ont accordée et pour les conseils judicieux qu'ils m'ont généreusement donnés.

Les travaux présentés dans ce mémoire ont été réalisés au sein de l'équipe COSMI (COncption et Sûreté de fonctionnement des systèmes mécatroniques) du département AS2M de l'Institut FEMTO-ST. Je tiens donc à remercier le directeur de AS2M, Nicolas Chaillet, pour les conditions de travail favorables dont j'ai bénéficié pour effectuer ces travaux de recherche.

J'adresse également mes remerciements à tous les membres de AS2M, Professeurs, Maîtres de Conférences, doctorants et secrétaires, pour leur patience et leur bonne humeur. Ils ont su faire régner dans le laboratoire une atmosphère studieuse et chaleureuse. Mille mercis à tous mes amis qui m'ont supportée et soutenue. Il y a aussi ceux qui ont déjà quitté le département. En particulier, je pense à Béatrice et Laïtitia.

Enfin et surtout merci à mes parents, à mon frère, à Jean Yves et à sa famille pour m'avoir soutenue durant ces années et avoir cru en moi. Je leur dédie ce mémoire.

TABLE DES MATIERES

Introduction générale.....	1
1 Introduction à la conception de l'organisation d'un projet	5
1.1 Définition de la constitution d'équipes.....	6
1.1.1 Domaine du management des organisations	6
1.1.2 Domaine de la production	7
1.1.3 Domaine du projet de conception.....	8
1.1.4 Synthèse.....	9
1.2 Les thèmes de recherche relatifs à la constitution d'équipes.....	9
1.3 Structuration du projet basée sur des représentations matricielles	10
1.3.1 Domaines du projet : concepts et définitions	10
1.3.2 Modélisation du projet par des représentations matricielles	14
1.3.3 Synthèse.....	16
1.4 Pilotage des compétences	17
1.4.1 Définition de la "compétence" dans la littérature.....	17
1.4.2 Notre définition de la "compétence"	20
1.4.3 Mesures de similarité.....	21
1.4.4 Introduction à la dynamique des compétences.....	21
1.4.5 Différents axes de pilotage des compétences	22
1.5 Premier positionnement	24
2 Etat de l'art sur la constitution d'équipes	27
2.1 Constitution d'équipes contribuant à la structuration du projet.....	28
2.1.1 Travaux traitant la minimisation des flux d'informations.....	28
2.1.2 Travaux traitant le couplage entre l'architecture du produit et l'architecture des acteurs	28
2.1.3 Synthèse.....	29
2.2 Constitution d'équipes contribuant au pilotage des compétences	29
2.2.1 Maximisation des connaissances techniques.....	30
2.2.2 Maximisation des capacités opératoires	31
2.2.3 Maximisation des connaissances techniques et capacités opératoires	32
2.2.4 Maximisation de la satisfaction des clients	36
2.3 Synthèse	38
2.4 Problématique	39
2.4.1 Structuration du projet.....	39
2.4.2 Pilotage des compétences	39
2.4.3 Second positionnement.....	41
3 Etat de l'art sur le problème d'affectation.....	43
3.1 Classification des travaux	44
3.2 Affectation mono-période avec une modélisation statique de compétences	45
3.2.1 Maximisation de l'utilisation des ressources en tenant compte de la qualification.....	46

3.2.2	Maximisation de l'utilisation des compétences des ressources dans l'affectation inter-département	46
3.2.3	Minimisation de l'insatisfaction au travail	48
3.2.4	Conclusion	49
3.3	Affectation multi-périodes avec une modélisation statique des compétences.....	50
3.3.1	Minimisation des coûts d'affectation sur de multiples périodes.....	50
3.3.2	Minimisation de la durée totale du projet.....	51
3.3.3	Minimisation de l'écart des références.....	53
3.3.4	Conclusion.....	53
3.4	Affectation mono-période avec une modélisation dynamique des compétences .	54
3.4.1	Maximisation de l'augmentation des compétences.....	54
3.4.2	Conclusion.....	56
3.5	Affectation multi-périodes avec une modélisation dynamique des compétences	56
3.5.1	Maximisation de la compétence avec effets d'apprentissage.....	57
3.5.2	Minimisation des coûts liés à la formation des ressources.....	59
3.5.3	Conclusion.....	61
3.6	Synthèse	61
3.6.1	Modélisation de l'évolution de compétences dans les problèmes d'affectation.....	61
3.6.2	Méthode de résolution et critère d'optimisation.....	63
4	Affectation des tâches contribuant au pilotage des compétences	65
4.1	Introduction.....	66
4.2	Affectation mono-période.....	66
4.2.1	Qualités attendues de la méthode	67
4.2.2	Hypothèses de caractérisation des compétences	67
4.2.3	Structure générale de la méthode	67
4.2.4	Formulation mathématique et recherche d'une solution optimale	76
4.2.5	Présentation d'un cas d'étude	80
4.2.6	Test du principe	86
4.2.7	Synthèse.....	89
4.3	Affectation multi-périodes avec évolution des compétences	90
4.3.1	Qualités attendues de la méthode	90
4.3.2	Hypothèses de modélisation et construction des données.....	90
4.3.3	Structure générale.....	94
4.3.4	Contribution à l'évolution des compétences	100
4.3.5	Modèle mathématique de l'affectation multi-périodes.....	107
4.3.6	Principe et mise en œuvre de la méthode du recuit simulé	113
4.3.7	Test du principe	119
4.3.8	Éléments de validation et synthèse.....	127
4.4	Conclusion	128
5	Contribution à la constitution d'équipes de conception couplant la structuration du projet et le pilotage des compétences.....	131
5.1	Problématique	132
5.2	Méthode d'identification d'une architecture par optimisation du couplage entre deux domaines	134
5.2.1	Qualités attendues.....	136
5.2.2	Formule matricielle représentant le couplage de deux domaines.....	136

5.2.3	Méthode d'optimisation du couplage pour l'identification de l'architecture des acteurs	138
5.2.4	Conclusion.....	144
5.3	Constitution d'équipes contribuant à la structuration du projet et au pilotage des compétences.....	144
5.3.1	Qualités attendues.....	144
5.3.2	Structure générale.....	145
5.3.3	Mise en œuvre et test de la méthode	146
5.3.4	Éléments de validation	159
5.3.5	Comparaison des solutions proposées	161
5.4	Conclusion et synthèse	163
6	Conclusion et perspectives	165
7	Bibliographie.....	169
8	Annexe	179
8.1	Annexe 1 : Représentations matricielles.....	181
8.2	Annexe 2 : Affectation mono-période	186
8.3	Annexe 3 : Affectation multi-périodes	188
8.4	Annexe 4 : Constitution d'équipes	199
8.5	Annexe 5 : Fonction de régression des compétences	205
8.5.1	Hypothèses	205
8.5.2	Introduction de la courbe d'apprentissage.....	205
8.5.3	Expérimentation de la courbe de régression.....	208
8.5.4	Proposition et paramétrage de la courbe de régression proposée.....	211
8.6	Annexe 6 : Convexité du problème quadratique	214

GLOSSAIRE DES NOTATIONS

	Mono période	Multi-périodes
Tâche générique de type i ; ($i=1,\dots,M$)	T_i	T_i
Tâche particulière de type i ; ($i=1,\dots,M$)	T'_i	$T_i^{k,l}$
Acteur j ; ($j=1,\dots,N$)	A_j	A_j
Connaissance de type c ; ($c=1,\dots,O$)	C_c	C_c
Niveau de compétence requis par la tâche générique i	rq_i	rq_i
Niveau de compétence requis par la tâche particulière	rq'_i	$rq_i^{k,l}$
Niveau de maîtrise requis dans la connaissance c par la tâche générique de type i	$r1_{i,c}$	$r1_{i,c}$
Niveau de maîtrise requis dans la connaissance c par la tâche particulière de type i	$r1'_{i,c}$	$r1_{i,c}^{k,l}$
Niveau de maîtrise acquis dans la connaissance c par l'acteur j .	$r2_{c,j}$	$r2_{c,j}^k$
Indicateur de similarité (a priori) entre la tâche i et l'acteur j ,	$v3_{i,j}$	$v3_{i,j}^{k,l}$
Indicateur de similarité (a posteriori) entre la tâche i et l'acteur j ,	$r3_{i,j}$	$r3_{i,j}^k$
Coefficient correcteur	$\gamma_{i,j}$	$\gamma_{i,j}^{k,l}$
Augmentation du niveau de maîtrise de la connaissance c acquis par l'acteur j au cours de la période k	-	$(\Delta C_{i,l})_{c,j}^k$
Augmentation maximale des $(\Delta C_{i,l})_{c,j}^k$,	-	$DIF_{c,j}^k$
Charge maximale que peut réaliser l'acteur j .	LM_j	$LM_{j,k}$
Temps théorique attribué pour réaliser la tâche i	L_i	$L_i^{k,l}$
Taux de salaire de l'acteur j	S_j	S_j
Taux de salaire des tuteurs	ST	ST
Variable de décision en nombres réels	$x_{i,j}$	$x_{i,j}^{k,l}$
Taux de pénalité	φ_i	φ_i
Nombre objectif d'acteurs à qualifier	O_i	O_i
Nombre d'acteurs qualifiés	R_i	R_i
Taux de pénalité	φ_i	φ_i
Paramètre de normalisation	μ, β	
Paramètre d'homogénéisation	λ	

INTRODUCTION GENERALE

Les entreprises sont confrontées à une situation fortement compétitive. Elles doivent développer rapidement de nouveaux produits innovants afin de répondre aux exigences de plus en plus élevés de leurs clients. Lorsqu'il s'agit de situations avec une forte évolution des technologies et des besoins du marché, la définition des tâches de conception, ainsi que des nouvelles compétences associées peut évoluer assez rapidement. Ces situations peuvent engendrer une carence de compétences pour concevoir les nouveaux produits, à un moment voulu. Le développement des compétences des ressources humaines devient une des priorités de l'entreprise.

La constitution d'équipes peut être considérée comme un outil de développement de compétences permettant d'améliorer la performance durable des projets de conception. La conception de nouveaux produits nécessite un grand nombre de tâches mobilisant des acteurs ayant des compétences variées. L'affectation des tâches aux acteurs est une activité critique de l'organisation d'un projet puisqu'elle a un impact fort sur les performances du projet, en termes de qualité, coût et délai. Elle peut aussi préparer les acteurs pour de futurs projets en leur permettant de développer leurs compétences au cours du projet. Cette activité est souvent considérée, dans la littérature, comme un sous-problème de la constitution d'équipes, qui est l'une des préoccupations principales du chef de projet.

En considérant le caractère dynamique des compétences, le problème d'affectation des tâches aux ressources ne consiste alors plus seulement à rechercher l'acteur le plus approprié pour la période courante mais aussi à anticiper des apprentissages qui seront utiles à des périodes futures. En effet, la compétence n'est pas une donnée statique, assimilable à la capacité d'une machine. D'une part, les tâches à réaliser peuvent évoluer, par exemple, en fonction des nouveaux besoins des clients ou des changements de technologies, créant ainsi un besoin d'évolution des compétences. D'autre part, un acteur apprend lors de la réalisation d'une tâche en situation professionnelle et peut ainsi se préparer à la réalisation de tâches plus difficiles. L'affectation des tâches représente donc un moyen de maintenir ou de développer le niveau de compétence des ressources humaines.

L'affectation des tâches aux ressources que nous traiterons dans le cadre de la constitution d'équipes dans ce document est considérée comme un moyen d'aide à la sélection de la bonne personne pour la réalisation d'une tâche à un moment donné. Elle tentera également d'intégrer la notion d'évolution de compétence dans un problème d'affectation. Actuellement il est rare que les entreprises travaillent sur un projet de conception unique. Le partage des ressources humaines est une préoccupation majeure des chefs de projets. L'affectation multi-projets est inévitable dans ce contexte. Dans ce document, nous étendrons l'affectation de tâches sur des périodes temporelles multiples et sur plusieurs projets.

Par ailleurs, le problème de la constitution d'équipes de conception n'est pas seulement un problème d'affectation. Il s'agit aussi de fournir un regroupement approprié des acteurs du

projet en équipes. Il s'intègre dans un problème plus général de conception d'une organisation de projet. Nous entendons par "organisation de projet" le résultat obtenu par les activités de structuration des processus, d'affectation des tâches aux acteurs, de découpage des acteurs en équipes, de définition des responsabilités hiérarchiques, de définition des règles de communication ...

Dans la littérature sur l'ingénierie de la conception, de nombreux chercheurs renommés ont mis en évidence que l'organisation d'un projet est fortement liée à l'architecture du produit à concevoir. Lorsqu'une entreprise lance un nouveau produit, l'architecte système doit concevoir l'architecture produit et le chef de projet doit concevoir en même temps l'organisation du projet. Cette organisation du projet doit permettre de faire émerger des compétences collectives ou individuelles jugées stratégiques pour les projets en cours et les projets futurs. Ce problème se rencontre dans toutes les organisations dont la complexité des produits nécessite des architectures complexes du produit et de l'organisation du projet (aéronautique, automobile, construction, voire micro-produit, ...).

Les contributions présentées dans ce mémoire s'adressent aux chefs de projets lors des phases préliminaires d'un projet pour lier la structuration d'un projet à l'architecture du produit et aux directeurs de départements "Recherche et Développement" pour décider des affectations multi-périodes multi-projets répondant à des objectifs d'évolution des compétences.

Ce travail de thèse s'inscrit dans la prolongation de travaux réalisés précédemment au sein de l'équipe "COncception et Sûreté de fonctionnement des systèmes MécatronIques (COSMI)" de AS2M.

- La thèse de Farouk Belkadi qui a proposé une méthode de caractérisation des compétences en conception [Belkadi, 2006], [Belkadi *et al.*, 2007].
- La thèse de Ghassen Harmel qui a développé une méthode de propagation des choix d'architecture d'un domaine sur un autre [Harmel, 2007], [Bonjour *et al.*, 2009a].

La problématique générale traitée dans ce mémoire est de concevoir une organisation de projets de conception durablement performante, en formulant et résolvant des problèmes d'affectation qui tiennent compte des compétences des acteurs, de leur évolution au cours des projets et des choix préliminaires d'architecture des produits.

Pour répondre à cette problématique, nous avons traité au préalable trois sous-problèmes :

- Optimisation d'affectation mono-période (chapitre 4.2)
- Optimisation d'affectation multi-périodes avec modélisation de l'évolution des compétences (chapitre 4.3)
- Optimisation de l'architecture des acteurs (chapitre 5.3)

En résumé, nous proposons dans ce travail une méthode de constitution d'équipes contribuant au pilotage des compétences à travers une affectation multi-périodes des tâches et contribuant à la structuration du projet (découpage du projet en équipes) en tenant compte de l'architecture du produit.

Ce mémoire décrit la méthodologie proposée et sa validation par quelques exemples. Il se compose de cinq chapitres :

Le chapitre 1 porte sur la présentation globale des thématiques traitées dans le contexte de la constitution d'équipes. Deux approches différentes (ou thématiques) sont présentées selon l'objectif poursuivi : le pilotage des compétences et la structuration du projet. Nous introduisons les principaux concepts et différentes typologies relatives à ces thématiques.

Le chapitre 2 détaille et analyse des travaux relatifs à la constitution d'équipes pour chacune des thématiques précédentes. Cela nous permet de définir les limites des modèles existants et de mieux positionner nos contributions à la fin de ce chapitre.

Le chapitre 3 présente un état de l'art sur les travaux d'affectation des tâches à ressources considérées comme un sous-problème de la constitution d'équipes. Ces travaux d'affectation seront classés selon deux critères qui sont liés à l'horizon temporel considérés (mono- ou multi-périodes) et au type de modélisation des compétences (statique ou avec évolution). L'analyse de cet état de l'art nous permet de prendre connaissance des différentes formulations des problèmes d'affectation et en synthèse, d'identifier le manque de prise en compte du caractère évolutif des compétences.

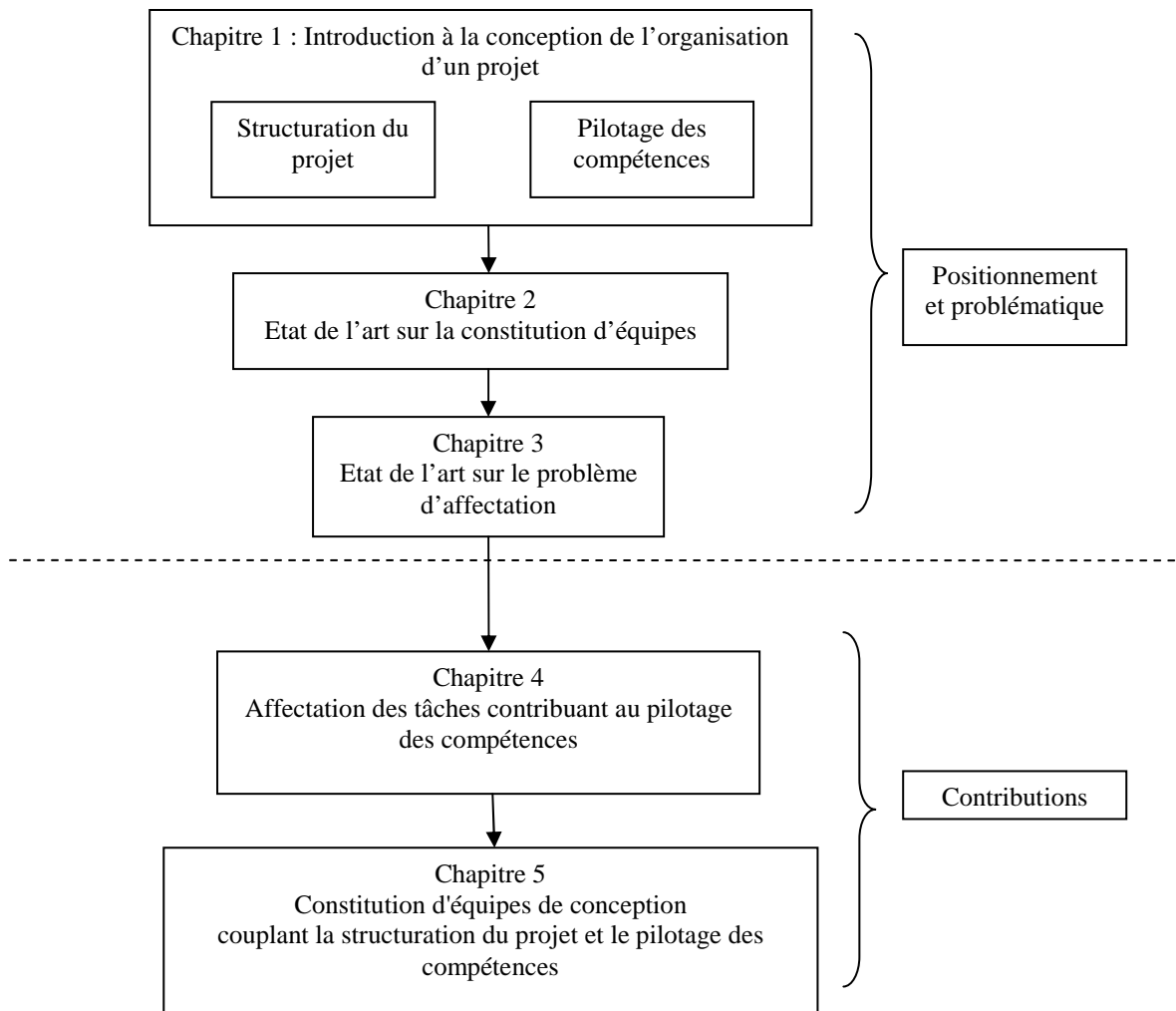
Dans le chapitre 4, nous présentons une démarche d'affectation mono-période des tâches aux ressources. Le problème est formulé sous forme de programmation linéaire en nombre réel, avec une vision statique de la modélisation des compétences. La modélisation obtenue sert de base pour la démarche suivante, où nous traitons un problème d'affectation sur des périodes multiples et des projets multiples, optimisé à l'aide d'un algorithme de recuit simulé. Ce deuxième problème d'affectation introduit une vision dynamique de la modélisation des compétences. Un exemple d'application permet d'illustrer la démarche proposée. Il repose sur un recueil de données réalisé à partir de deux projets de conception de micro-produits (capsule endoscopique pour la thérapie de tumeurs intestinales)¹.

Dans le chapitre 5, nous proposons une méthode qui permet d'optimiser l'architecture organisationnelle des acteurs en tenant compte de contraintes provenant de l'architecture du produit. La formulation du problème aboutit à un programme quadratique convexe, qui peut être résolu en utilisant un algorithme d'optimisation existant. Enfin, nous proposons une démarche globale de constitution d'équipes permettant à la fois de tenir compte d'objectifs de développement de compétences (affectation des tâches aux ressources sur plusieurs périodes et sur plusieurs projets) et de structurer l'architecture des acteurs.

La conclusion générale de ce mémoire résume nos contributions, tout en soulignant leurs limites et les perspectives qu'elles ouvrent.

¹ Le titre du projet est VECTOR (Versatile Endoscopic Capsule for gastrointestinal TumOr Recognition and therapy), le site internet du projet est : <http://www.vector-project.com/>

Le schéma suivant montre la structure globale de ce mémoire.



1 Introduction à la conception de l'organisation d'un projet

Le terme "organisation" a de multiples significations. Il peut s'agir de l'action d'organiser un système, du résultat de l'organisation, ou d'une entité "entière" comme une entreprise, un bureau d'études ou une association.

Nous choisissons la définition suivante de la conception de l'organisation d'un projet : c'est l'activité de structuration d'un projet qui consiste à définir le processus adapté aux objectifs et au contexte de ce projet, à affecter des acteurs aux tâches et à définir leurs responsabilités (missions, liens hiérarchiques, relations) ainsi que les moyens et procédures de communication et de contrôle. Nous n'incluons pas, dans le concept d'organisation, la planification temporelle des tâches que nous considérons comme de la conduite de projet.

Ce premier chapitre vise à positionner globalement l'objectif de notre travail. Nous présentons d'abord la définition de la constitution d'équipes afin de positionner la constitution d'équipes multifonctionnelles en relation avec le développement de leurs compétences. Nous faisons ensuite une présentation globale des thèmes de recherche relatifs à la constitution d'équipes. Ces thèmes porteront en particulier sur la structuration du projet basée sur des représentations matricielles et sur le pilotage des compétences. Nous présentons différentes visions sur ce thème, leurs avantages, leurs inconvénients. Nous terminons ce chapitre par un bilan général et un premier positionnement de notre problématique. Ce positionnement sera affiné à la fin du chapitre II.

1.1 Définition de la constitution d'équipes

La constitution d'équipes (ou team building) a plusieurs définitions selon les domaines d'applications : management des organisations, production, management des projets de conception. Dans cette partie, nous présentons ces différentes définitions.

1.1.1 Domaine du management des organisations

Nous avons étudié la littérature relative au domaine du management des organisations essentiellement sous l'angle du management de projet.

[Cleland, 1994] définit la constitution d'équipes comme "le processus qui consiste à prendre une collection d'individus avec des besoins, des expériences, et des expertises différents et les transformant en une unité de travail intégrée et efficace. Dans ce processus de transformation, les buts et les énergies des collaborateurs fusionnent et soutiennent les objectifs de l'équipe"².

En parlant de la gestion des organisations, [PMI, 2004] définit trois types d'activités :

- la planification des ressources humaines dont le but est d'identifier et de documenter les rôles du projet, les responsabilités, et de planifier des ressources humaines,
- le développement d'équipes de projet dont le but est d'améliorer des compétences et les interactions entre membres d'équipes, et d'augmenter la performance du projet,
- le contrôle des équipes de projet dont le but est de suivre leur performance, de récupérer les remarques, les critiques et les adaptations pour améliorer le projet

En management des organisations, ou en management des ressources humaines, le terme de "team building" se situe souvent dans la phase de développement des équipes de projet dans le but d'améliorer les performances au niveau de l'organisation. Le team building peut être considéré comme un processus permettant de développer des valeurs collectives au sein d'une entreprise, comme la cohésion, l'interdépendance, ou l'implication. Grâce à des activités à caractère événementiel, souvent de nature collective et interactive, les participants cultivent leurs facultés à travailler en équipe. Les activités de "team building" ont pour but d'encourager les membres d'équipes à communiquer informellement afin de créer un climat de confiance et d'établir de bonnes relations de travail.

[Tuckman, 1965] définit quatre étapes dans l'évolution d'une équipe: (1) Forming (étape de rassemblement des acteurs, les objectifs du projet sont incertains), (2) Storming (étape où il existe des conflits, les objectifs ne sont pas encore clarifiés), (3) Norming (étape où tout le monde commence à jouer son rôle et à prendre sa part de responsabilité, les conflits sont résolus) et (4) Performing (étape où les acteurs travaillent harmonieusement). De plus, l'évolution d'une équipe n'est pas complètement séquentielle. Les membres de l'équipe peuvent collaborer difficilement au début, puis améliorer leur compréhension mutuelle et donc mieux collaborer. De plus, les équipes sont souvent confrontées à des conflits ou problèmes techniques qui peuvent dégrader la collaboration au sein de l'équipe, avant d'en

² En Anglo-Saxon: "the process of taking a collection of individuals with different needs, backgrounds, and expertise and transforming them into an integrated, effective work unit. In this transformation process, the goals and energies of individual contributors merge and support the objectives of the team".

améliorer le fonctionnement lors de la résolution collective des problèmes. Ces changements sont importants pour renforcer la performance des équipes.

1.1.2 Domaine de la production

Dans le domaine de la production, on parle le plus souvent de problématiques associées à la planification des ressources humaines. La planification des ressources humaines vise à allouer les ressources humaines pour chaque intervalle de temps sur un horizon donné, de telle sorte que les besoins par intervalle soient couverts et que les différentes contraintes soient satisfaites. Selon [Chan, 2002], de nombreuses contraintes devraient être satisfaites.

- Juridique : la législation en matière de droit du travail (durées de travail, et de repos) sur différents horizons de temps (journalier, hebdomadaire, mensuel et annuel).
- Social : la répartition équitable des tâches entre salariés, préférences individuelles (pour les horaires et les durées du travail).
- Technique : la prise en compte des compétences et des niveaux requis.
- Economique : la meilleure affectation des ressources disponibles aux charges à chaque moment de l'horizon. On cherche à ne pas engager inutilement des ressources.

Les problématiques associées à l'allocation des ressources humaines dépendent de l'horizon temporel [Bennour, 2004], [Bennour *et al.*, 2008]. Sur le long terme, au niveau stratégique, différentes politiques d'affectation sont évaluées pour contrôler l'évolution des compétences d'une entreprise. A moyen terme, on cherche à planifier l'affectation du personnel, dont l'objectif est d'ajuster la capacité de production à la charge prévisionnelle sans une affectation nominative des employés. Enfin, le court terme permet d'ordonnancer et d'allouer effectivement les ressources en prenant en compte des contraintes plus opérationnelles comme leurs disponibilités, leurs temps de repos, leurs jours de congés. En changeant d'horizon de travail au niveau opérationnel, la problématique d'allocation des ressources induit un grand nombre de nouveaux paramètres à prendre en compte. Certains auteurs [Zülch *et al.*, 2004], [Zülch et Rottinger., 2007] proposent un outil informatique permettant de planifier des ressources humaines en fonction de leurs compétences, leurs disponibilités des ressources et de l'évolution des technologies de produit et de fabrication. Ils se sont intéressés à la détermination du nombre de travailleurs et à leurs qualifications. L'outil proposé permet de traiter les planifications à moyen terme et à court terme des ressources humaines, tout en prenant en compte les facteurs stratégiques liés à l'évolution du produit dans la planification.

Nous trouvons la notion d'équipe dans les problématiques concernant l'"équipe autonome de production". L'équipe autonome de production est un mode d'organisation ancien dont la diffusion s'est considérablement accrue dans l'entreprise industrielle depuis plusieurs années. Les problèmes d'affectation des ressources humaines dans les cellules manufacturières s'intéressent en particulier à constituer des équipes de production. La conversion de la production traditionnelle de type "job-shop" à la production de type "cellules manufacturières" apporte une nouvelle perspective dans la gestion des ressources humaines, en particulier, la constitution des équipes de production. En ce qui concerne les cellules manufacturières, les acteurs dont les compétences sont reliées aux processus de production sont alloués aux cellules dont la mission est de fabriquer ou d'assembler un produit. Comme les cellules réalisent des processus de fabrication hétérogènes, les équipes de production dans chaque cellule sont multidisciplinaires. [Fitzpatrick et Askin, 2005] [Jia, 1998] [Vairaktarakis *et al.*, 2002],[Li, 2003] s'intéressent à la gestion des ressources humaines en groupes autonomes de fabrication pour la conduite d'un atelier. Ces groupes sont constitués autour

d'un métier ou d'une activité. Ils permettent d'inclure une dimension sociale dans l'organisation du travail, tout en favorisant l'autonomie. Ces travaux cherchent à améliorer la productivité, le coût de fabrication et le taux d'engagement, la compétence, etc.

1.1.3 Domaine du projet de conception

En général, l'activité d'une équipe dans les projets de conception a pour but de résoudre un problème [Sharifi et Pawar, 1996]. Les activités de conception sont souvent non routinières et sont caractérisées par la complexité et l'incertitude. C'est un travail qui ne peut pas être réalisé par une seule personne ou une unique fonction. Les individus seront alloués au projet (de façon partielle ou dédiée) afin de résoudre le problème et d'apporter leur expertise au projet.

Dans le passé, les entreprises ont organisé leurs étapes de développement d'un produit de manière séquentielle, partant de la conceptualisation du produit jusqu'à l'introduction du produit sur le marché. Cependant, ce type de décomposition empêche les chefs du projet d'avoir la vue globale du projet. Ces processus ont été réalisés par une organisation par départements fonctionnels (ou métiers). Pour réduire les délais de conception et de limiter les boucles de retour arrière souvent liées à un manque de communication entre les différents acteurs, les entreprises ont modifié leur processus de développement selon les principes de l'ingénierie concourante (ou simultanée). L'un des principes de l'ingénierie concourante est de rassembler des experts des processus en aval pour les faire travailler ensemble dès le début du projet afin de résoudre et d'éviter les problèmes techniques qui peuvent arriver ultérieurement. Les tâches du processus sont alors réalisées simultanément par des équipes multi-fonctionnelles.

L'équipe est définie par [Baker et Salas, 1992] de la façon suivante : "un ensemble distinct de deux ou plusieurs personnes qui interagissent dynamiquement, indépendamment pour atteindre un objectif commun, qui sont affectés à des tâches différentes et des rôles différents, et qui se rassemblent pour une durée déterminée".

La constitution d'équipes est une activité importante dans le pilotage d'un projet de conception. Elle met l'accent non seulement sur la prise en compte de la performance technique mais également sur la prise en compte des capacités opératoires facilitant la collaboration entre les membres de l'équipe. Plusieurs dénominations comme "équipe multifonctionnelle", "équipe de projet" sont utilisées pour définir un groupe d'individus de différentes disciplines, rassemblés pour travailler ensemble dans le projet de développement d'un produit [Chen, 2005], [Braha, 2002], [Zakarian et Kusiak, 1999], [Tseng *et al.*, 2004]. La constitution d'équipes dans le contexte de conception du produit est souvent considérée d'une part, comme une approche d'organisation capable de promouvoir l'innovation du produit, d'améliorer la capacité de travailler en équipe, la communication entre fonctions et d'autre part, comme une approche d'organisation capable de former des équipes multifonctionnelles dans le but de développer des compétences collectives [Bourgeon, 2007].

La conception d'un système complexe peut être facilitée par une structuration appropriée du projet. La structuration du projet nous permet d'élargir l'objectif de pilotage du projet de conception, du système-produit vers le système-projet dans sa globalité [Bonjour, 2008a]. De la même manière qu'une identification appropriée de l'architecture du produit est un préalable à une conception détaillée satisfaisante, une structuration appropriée du projet est un préalable à une conduite efficace du projet.

Les choix relatifs à l'architecture du produit ont un impact fort sur les choix organisationnels et par la suite, sur les performances globales du projet et du produit. La constitution d'équipes est donc considérée dans le domaine de la conception des systèmes complexes comme une approche d'organisation permettant de prendre en compte des contraintes de l'architecture de différents domaines (produit, processus, acteurs) [Eppinger et Salminen, 2001] et d'identifier une architecture du projet appropriée.

1.1.4 Synthèse

En comparant l'utilisation des termes "team building" et "constitution d'équipes" dans les trois domaines précédents, nous avons constaté que :

- Le terme "team building" du point de vue du management des organisations met plus l'accent sur le développement des ressources humaines que sur la planification. Il a pour objectif de construire une équipe harmonieuse et d'étudier l'évolution du développement de l'équipe. Il tend à améliorer les capacités opératoires des membres de l'équipe.
- Le terme "team building" dans le domaine de la production est considéré comme un mode d'organisation utilisé souvent au niveau organisationnel, en particulier dans l'allocation des ressources humaines dans les cellules manufacturières. Ces équipes de production sont constituées en fonction d'un grand nombre d'objectifs de performance en production (quantité de produit, coût lié aux opérateurs, niveau de compétence, taux d'engagement, etc.).
- Le terme "team building" dans les projets de conception correspond au processus de construction d'une équipe multifonctionnelle ayant pour mission de réaliser une tâche de conception de produit en tenant compte des compétences des membres de l'équipe et des différents aspects du projet (produit, processus, acteur).

Par la suite nous utiliserons le terme "constitution d'équipes" plutôt que "team building" et nous nous intéresserons essentiellement aux travaux relatifs aux projets de conception.

1.2 Les thèmes de recherche relatifs à la constitution d'équipes

De nombreux travaux de recherche s'intéressent à la constitution d'équipes en conception. Ceux-ci s'intéressent à développer des modèles et des outils pour affecter des tâches à des acteurs "de façon appropriée" ou à structurer les nombreux acteurs d'un projet en équipes pour faciliter les efforts de coordination entre équipes.

Par la suite, nous avons retenu deux thèmes de recherche relatifs à la constitution d'équipes (FIG. 1.1) qui sont critiques dans le pilotage de projets de conception : la structuration du projet basée sur des représentations matricielles et le pilotage des compétences.

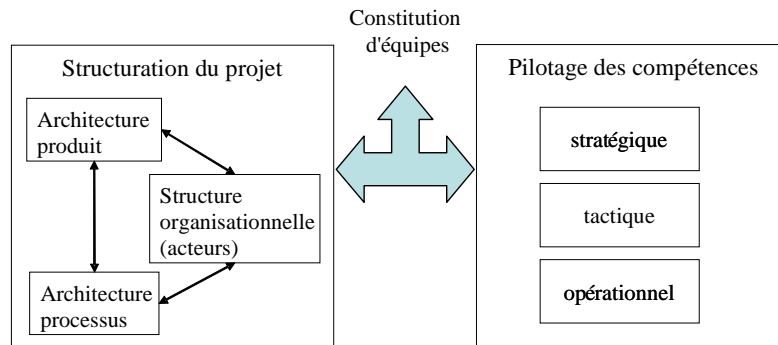


FIG. 1.1 : Thèmes de recherche liés à la constitution d'équipes

1.3 Structuration du projet basée sur des représentations matricielles

Dans cette partie, nous présentons la première thématique de recherche relative à la constitution d'équipes, à savoir la structuration du projet.

La structuration du projet a pour but de définir et concevoir conjointement l'architecture du produit, l'architecture des processus et l'architecture des structures organisationnelles (acteurs). Meinadier [2002] soutient que : "la planification d'un projet consiste à rendre cohérentes entre elles trois logiques, dans le respect des objectifs et contraintes du projet : la logique du produit à réaliser (arborescence des produits), la logique temporelle des processus (enchaînement des activités) et la logique de la structure organisationnelle du projet". Ces trois logiques correspondent à ce que Eppinger et Salminen [2001] appellent les domaines du projet : le produit, le processus et la structure organisationnelle.

1.3.1 Domaines du projet : concepts et définitions

Dans cette partie, nous présentons des définitions concernant les concepts principaux relatifs à la structuration du projet comme projet, produit, processus et structure organisationnelle.

Projet : Les normes qualité ISO9000 version 2000 définissent un projet comme un "processus unique qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques telles que les contraintes de délai, de coûts et de ressources". Les projets concernent tous les secteurs de production de biens et de services, et tous les départements d'une entreprise. Par exemple, le développement d'un nouveau produit, le développement d'un logiciel, la conception innovante d'un nouveau procédé de micro-fabrication, la construction d'un bâtiment, la réorganisation d'un atelier, la maintenance d'un site de production, la mise en place d'une démarche qualité.

Un **projet de conception** consiste en un ensemble d'activités qui génèrent de la valeur pour la parties prenantes et consomment les ressources du projet. La conception est une activité complexe, nécessitant l'intégration de multiples points de vue (cognitif, technique, social, économique, organisationnel,...). Chaque point de vue fournit un ensemble d'exigences et de contraintes qui sont souvent mal connues au début du projet et qui sont évolutives. On distingue deux grandes classes de projet :

- Les projets dits récurrents (ou de conception routinière). Ils concernent des projets où les technologies sont connues et qualifiées, les connaissances sont maîtrisées et la démarche de conception connue.
- Les projets dits non récurrents (ou de conception inventive). Dans ces projets, les technologies sont en cours de développement. Les domaines d'expertises techniques sont critiques. Les connaissances ainsi que la démarche de conception sont à créer.

1.3.1.1 Architecture d'un produit

Les définitions du concept d'architecture du produit sont multiples mais elles se rejoignent toutes pour affirmer que l'architecture du produit se caractérise par : "la structure des composants et des interactions qui les lient" [Pimmler et Eppinger, 1994]. Par ailleurs, il est à noter que l'architecture est une vision structurelle d'un système et peut évoluer au cours du projet.

Au niveau international, les travaux de deux auteurs sont fréquemment cités par les chercheurs en architecture de produit : Ulrich et Eppinger. Ulrich [Ulrich et Tung, 1991] définit l'architecture du produit comme "une représentation générique des produits où un système conceptuel des composants physiques est associé à un système conceptuel des éléments fonctionnels afin de concevoir différents produits lors de l'étape de conception".

Cette définition a été approfondie par [Ulrich, 1995], l'architecture du produit est alors :

- l'arrangement des fonctions ;
- l'allocation des fonctions aux composants ;
- la spécification des interfaces entre les composants.

Ulrich [1995] distingue deux types d'architecture : architecture modulaire (à une fonction ne correspond qu'un seul constituant physique) et architecture intégrale (les fonctions n'impactent pas directement un unique composant). Il est facile d'imaginer qu'il existe une architecture hybride [Sosa *et al.*, 2000] qui est composée à la fois de modules (groupements d'éléments) et d'éléments intégrateurs, en lien avec de nombreux autres éléments, un élément pouvant être une fonction, un constituant, un acteur...

1.3.1.2 Architecture des processus

Un processus peut être défini comme un agencement logico-temporel de tâches permettant de répondre à une mission du système concerné. [Bonjour, 2008b] définit un processus d'entreprise comme étant un agencement (partiellement) ordonné de sous-processus, portant sur des objets intermédiaires de même type, et dont la fin est conditionnée par des critères de transition. Son objectif porte sur la transformation progressive des entrées en résultats attendus, représentant ainsi une chaîne de valeur ajoutée. Le point central de la formalisation d'un processus est la notion d'objet, qui permet un ancrage explicite sur ce qui est concret, tangible. Le processus décrit la succession de changement d'état d'un flux d'objets de même nature (informationnels, matériels, monétaires, ...).

Un processus d'entreprise doit permettre de répondre à une mission générique, définie par le niveau stratégique. La formalisation des processus peut se faire à différents niveaux de détail. La décomposition récursive peut s'arrêter lorsque l'entreprise considère que l'acteur, à qui est attribuée la responsabilité de réalisation d'un sous-processus (sa mission), peut organiser son activité de façon autonome.

Selon [Jin et Levitt, 1996], les tâches (sous-processus) peuvent être classées en deux catégories : la tâche productive apportant de la valeur ajoutée au produit final (par exemple : les tâches de conception), et la tâche de coordination qui facilite la tâche productive (par exemple : communication, résolution de problèmes entre tâches productives, arbitrage). La durée de la tâche productive dépend de la complexité du produit. Cependant, la durée de la tâche de coordination dépend de l'organisation des équipes du projet.

1.3.1.3 Structure organisationnelle (architecture des acteurs)

La recherche de performance impose à l'entreprise plus de flexibilité dans son organisation de travail, une meilleure maîtrise des technologies émergentes mais surtout une meilleure considération de ses ressources humaines. Notre définition de l'architecture des acteurs (ou l'architecture organisationnelle³) est basée sur le travail de [McCord et Eppinger, 1993]. Les auteurs ont défini l'**architecture des acteurs** comme une forme d'organisation efficace permettant de minimiser des efforts de coordination entre les acteurs du projet et d'améliorer l'efficacité et l'innovation dans le projet de conception.

Nous allons présenter ci-dessous différentes typologies d'organisation utilisées dans les projets de conception et approfondir le type "organisation en projet autonome". C'est dans cette forme d'organisation que le concept d' "**architecture des acteurs**" prend toute son importance.

De l'organisation fonctionnelle à l'organisation en projet autonome

Trois typologies d'organisation du projet ont été définies : l'organisation par départements (basée sur les disciplines), l'organisation matricielle et l'organisation par projet autonome.

- L'organisation par départements a été longtemps une organisation traditionnelle dans la plupart des entreprises. Les acteurs sont groupés selon leurs domaines d'expertise. Ce type d'organisation est approprié pour la conception routinière où les processus de conception sont séquentiels. L'avantage de ce type d'organisation est de favoriser le développement de l'expertise technologique. Les inconvénients sont que la gestion de projet est complexe, les processus ne sont pas clairement définis pour tous les acteurs. De plus, ce type d'organisation n'encourage pas les acteurs à travailler en équipe (entre départements).
- L'organisation matricielle se situe entre l'organisation par départements et l'organisation par projet. L'organisation matricielle maintient un bon niveau de développement des métiers, apporte une bonne intégration dans la conception du produit et améliore la capacité de travailler entre départements. Cependant, cette organisation est moins claire que l'organisation par projet en termes de responsabilité.
- Dans une organisation par projet autonome, les chefs du projet ont une responsabilité affirmée sur les membres du projet. Ils sont les garants du résultat du projet et de son organisation. Les acteurs du projet sont détachés de leur département fonctionnel. Ce type d'organisation est réputé pour produire de meilleures performances en termes de délai, coût et qualité, mais souvent au détriment du développement de l'expertise des acteurs.

³ Le terme utilisé par McCord et Eppinger est "organizational structure".

[Wheelwright et Clark, 1992] ont raffiné cette typologie selon le niveau décisionnel du chef de projet. Cette typologie est présentée avec un continuum entre une organisation classique par départements et une organisation par projet autonome : "functional", "lightweight", "heavyweight", "autonomous". L'organisation de type "lightweight" et "heavyweight" correspond à une organisation de type matricielle, les chefs de projet "heavyweight" ont plus de pouvoir décisionnel que le projet "lightweight".

Selon [Huth, 2008], quand on compare l'efficacité entre l'organisation par départements, l'organisation matricielle et l'organisation par projet, l'organisation par départements est moins performante sur les trois aspects (délai du projet, coût, et performance globale).

Acteur

Avant de présenter le concept d'**architecture des acteurs**, nous définissons le concept d'acteur utilisé dans ce mémoire.

Dans des travaux antérieurs, notre équipe [Bonjour et Dulmet, 2006] a défini le concept d'"acteur" comme une ressource humaine de l'entreprise, qui peut être individuelle ou collective.

- Un **acteur individuel** : est une personne ayant reçu la responsabilité de réaliser une mission et agissant de manière permanente, temporaire ou occasionnelle, au service de l'entreprise considérée.
- Un **acteur collectif** : est un ensemble d'acteurs individuels, ayant reçu la responsabilité de réaliser une mission et agissant de manière permanente, temporaire ou occasionnelle, au service de l'entreprise considérée.

Il existe des acteurs collectifs de différents niveaux, depuis le "binôme" jusqu'à l'entreprise considérée dans sa globalité, en passant par toutes sortes d'intermédiaires : groupes de travail, équipes projet, métiers, filières-métiers, Direction de projet, usines, etc. Un acteur collectif peut être décomposé en acteurs de plus petits cardinaux, collectifs ou individuels.

Un acteur peut lui-même participer à des missions de différents acteurs collectifs de plus grands cardinaux. Il est alors important de constater que la différence porte sur l'horizon de ces missions : court, moyen, ou long terme. L'organisation matricielle n'est qu'une façon de séparer ces différents types d'horizon : les acteurs ont en charge des missions à court et moyen terme (par exemple, élaborer l'architecture de l'essieu, rédiger les justifications des choix...), mais aussi des missions à long terme concernant le développement de leurs métiers (par exemple, appropriation de nouveaux outils et méthodologies, d'innovations technologiques).

Architecture des acteurs

Le terme "architecture des acteurs du projet" utilisée dans ce document représente le résultat de la structuration des acteurs en équipes. Selon [McCord et Eppinger, 1993], l'architecture des acteurs est une structure organisationnelle représentant le flux d'informations entre concepteurs dans la conception collaborative. Cette architecture définit les acteurs principaux et leurs interactions. Chaque équipe est composée d'un ensemble d'acteurs fortement liés entre eux et faiblement liés à d'autres, externes à l'équipe. L'architecture des acteurs permet :

- de recenser tous les acteurs qui travaillent dans le projet,

- de documenter les niveaux d'interactions entre acteurs,
- d'identifier les interfaces organisationnelles,
- de diminuer les efforts de coordination entre concepteurs.

1.3.2 Modélisation du projet par des représentations matricielles

Nous nous intéressons ici à la modélisation de trois domaines (produit, processus et acteur) par des représentations matricielles et à l'identification de leur architecture par un algorithme de clustering. Le "clustering" fait référence au classement d'objets dans différents groupes, plus précisément, au partitionnement des données en paquets homogènes qu'on appelle "cluster". Selon [Hartigan, 1975], le but originel du clustering est de trouver des similitudes entre éléments afin de les regrouper. Plusieurs méthodes de clustering se basent sur des représentations matricielles, par exemple [Idicula, 1995] [Fernandez, 1998] [Thebeau, 2001].

Pour comprendre les outils utilisés dans les travaux de recherche, nous présentons succinctement la classification des représentations matricielles et leurs descriptions. Un résumé plus complet est fourni dans l'annexe 1. Ensuite, nous présenterons les travaux sur chaque type de modélisation.

1.3.2.1 Représentations matricielles

[Malmqvist, 2002] distingue deux classes d'outils matriciels : les matrices intra-domaine (ou Design Structure Matrix) et les matrices inter-domaines (ou matrice d'incidence).

- Les **matrices intra-domaines** représentent des couplages entre éléments de même nature, entre composants par exemple. Elles sont aussi appelées Design⁴ Structure Matrix (DSM) ou matrice structurelle de la conception. Il permet de modéliser et d'analyser l'architecture d'un système et de faciliter la gestion du projet de conception (produit, processus, acteur). Des algorithmes de clustering ont été développés pour identifier l'architecture de chaque domaine séparément [Browning, 2001]. Dans une DSM, les éléments en ligne et en colonne sont identiques. Une valeur dans la matrice représente l'intensité du couplage estimée entre les deux éléments concernés par la ligne et la colonne. Les éléments diagonaux de ces matrices sont sans signification. Les différents types de DSM et leurs usages sont présentés dans le tableau (TAB. 1.1).

Type de DSM	Usage
DSM produit	identifier des modules et des éléments intégrateurs
DSM acteur	réduire les efforts de coordination et augmenter la cohésion d'une équipe en plaçant ensemble des acteurs qui échangent d'importants flux d'information
DSM processus	réduire les boucles de retour arrière, estimer la durée d'un projet
DSM paramètre	orienter la tactique de conception en tenant compte des contraintes de précedence ou de causalité entre paramètres

TAB. 1.1 : Usages des DSM

⁴ Certains auteurs utilisent aussi le terme Dependency Structure Matrix qui a le même acronyme : DSM.

- Les **matrices inter-domaines** représentent des couplages entre éléments appartenant à deux domaines différents. Elles sont aussi appelées matrices incidences (MI), matrice d'allocation ou Domain Mapping Matrix (DMM). Ce dernier terme a été proposé par [Danilovic et Browning, 2007] pour les distinguer des matrices DSM.

1.3.2.2 Modélisation de l'architecture d'un produit

La modélisation du produit par DSM produit permet de représenter et visualiser facilement les interactions (de différentes natures) entre les constituants. Elles sont utiles d'abord lors de la définition de l'architecture organique (regroupement de modules, optimisation des couplages) et ensuite lors de l'intégration pour vérifier que les zones de couplages et les interfaces entre constituants sont bien réalisées et vérifiées.

Les DSM produit permettent d'identifier les architectures d'un produit et de distinguer les types d'architecture (architecture modulaire, architecture intégrale, ou architecture hybride) après l'utilisation d'un algorithme de *clustering*. Le but est de minimiser les interfaces entre sous-systèmes (ou modules) tout en maximisant les couplages internes au sous-système.

[Pimmler et Eppinger, 1994] proposent une taxonomie permettant de différencier les différents types d'interactions : spatial (lié à la propagation de contraintes physiques), énergétique (lié aux flux d'énergie), informationnel (lié aux flux d'information), matériel (lié aux flux de matière).

1.3.2.3 Modélisation de l'architecture des processus

Les méthodes de représentation des tâches sont nombreuses et apportent différents avantages selon le cadre d'utilisation. Les méthodes de représentation basées sur des graphes ont pour but de favoriser l'ordonnancement (comme PERT et CPM), tandis que les méthodes de modélisation des tâches basées sur des représentations matricielles ont pour but d'aider à la maîtrise des interactions lors de la conception de systèmes complexes [Malmqvist, 2002], [Harmel, 2007], [Krackhardt et Carley, 1998].

Cependant, les méthodes de représentation graphique prennent difficilement en compte les aspects itératifs des tâches de conception (échanges d'information progressifs et itératifs entre deux tâches), considérés comme une caractéristique fondamentale des tâches dans les projets de conception complexes, [Cho and Eppinger, 2001], [Tang *et al.*, 2000]. Pour pallier ce manque, la matrice DSM processus est utilisée dans la modélisation des tâches de conception. L'avantage de l'utilisation d'une DSM est qu'on surmonte la difficulté de la taille et de la complexité visuelle par rapport aux techniques de représentation graphique. De plus, une structure de données sous forme matricielle est facile à manipuler informatiquement.

De nombreux auteurs ont déjà travaillé sur l'optimisation des processus de conception [Eppinger *et al.*, 1994], [Altus *et al.* 1996], [Browning, 1998], [Tang *et al.* 2000], [Cho et Eppinger, 2001], [Braha, 2002], [David *et al.*, 2002], [Chen et Lin, 2003], [Yassine et Braha, 2003], [Whitfield *et al.* 2005], [Chen et Lin, 2004]. Ces travaux utilisent les DSM-processus pour modéliser des contraintes de précédence et pour proposer une optimisation des processus en limitant soit le nombre total d'itérations, soit les itérations entre groupes de tâches. Ils préconisent ainsi de créer des équipes cohérentes en affectant chaque groupe de tâche à une équipe.

1.3.2.4 Modélisation de l'architecture des acteurs

La modélisation de l'architecture des acteurs permet de modéliser les interactions et les efforts de coordination entre les acteurs dans un projet de conception dans le but de définir une structuration appropriée en équipes par une matrice DSM organisation (ou DSM acteurs). Parmi les travaux les plus récents utilisant les DSM organisation, on citera ceux de [Sosa *et al.*, 2004]. Dans leurs travaux, Sosa et al. proposent la taxonomie des interactions suivantes : qualité (lié aux types de moyens de communication), fréquence (lié à la fréquence des interactions), direction (lié aux sens des interactions), timing (lié à la maturité des informations (partielle ou complète) communiquées dans un projet). L'intensité de couplage dans une DSM organisation peut être estimée à partir de la fréquence ou de l'importance des échanges d'information entre deux acteurs dans l'organisation (s'ils n'ont pas d'interaction, l'intensité est nulle).

La figure (FIG. 1.2) présente un exemple de DSM acteurs, avant l'utilisation d'un algorithme de clustering (partie gauche) et après son utilisation (partie droite). Le rôle d'un algorithme de clustering est de regrouper les acteurs en petites équipes facilement gérables. Son principe est de réaliser un réarrangement des lignes et des colonnes, afin de maximiser les interactions au sein d'une équipe (les acteurs d'une même équipe sont placés proches les uns des autres pour collaborer au quotidien) et de minimiser les efforts de coordination entre les équipes qui coûtent cher lors du management de projets (réunions fréquentes, déplacements entre sites ...) et qui sont source de problèmes techniques (risques d'interfaces entre composants mal définies).

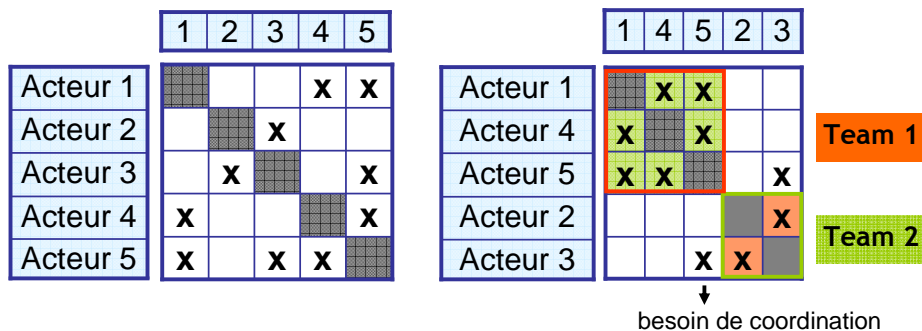


FIG. 1.2 : DSM acteurs (1) avant et (2) après clustering

1.3.3 Synthèse

Lors de la phase de conception préliminaire, le chef de projet doit considérer simultanément les architectures préliminaires de trois domaines clés de la conception : le domaine du produit, le domaine des processus et le domaine des acteurs (ou structures organisationnelles) [Eppinger et Salminen, 2001]. Ces domaines sont fortement couplés et un bon "alignement" est considéré comme un facteur favorable à la performance d'un projet [Galbraith, 1977], [Sosa *et al.*, 2004].

Les deux premiers types de thématiques (identification de l'architecture du produit et des processus) sortent du cadre de notre travail. Nous positionnons nos travaux sur le troisième type de modélisation, à savoir l'identification d'une **architecture des acteurs**. Nous considérons l'identification de l'architecture des acteurs comme une manière de constituer des équipes (structuration des acteurs en équipes).

1.4 Pilotage des compétences

Dans cette partie, nous présentons une deuxième thématique relative à la constitution d'équipes, à savoir le pilotage des compétences.

Les managers doivent satisfaire des objectifs à court terme, concernant les performances opérationnelles d'un projet et à long terme, concernant le développement de compétences stratégiques dans le système de conception. Pour atteindre leurs objectifs, le pilotage des compétences des ressources humaines joue un rôle clé. Il s'intéresse à tout ce qui peut contribuer à la production, à la mise en œuvre et au développement de compétences spécifiques. Le but est l'amélioration des performances de l'entreprise et des compétences des salariés. Ceci est possible parce que l'émergence d'une compétence est à l'intersection entre les capacités d'un acteur et l'opportunité de développer une compétence par l'affectation d'une tâche correspondante à cet acteur. Le pilotage de compétence a donc un lien fort avec le problème d'affectation, qui est souvent considérée comme la première étape dans la constitution d'équipes. L'affectation des tâches à des ressources (*task assignment*) et l'allocation des ressources à des tâches (*resource allocation*) sont deux visions duales.

L'objectif de cette partie est de présenter les problématiques liées au pilotage des compétences. Nous présentons d'abord quelques concepts et définitions de la compétence.

1.4.1 Définition de la "compétence" dans la littérature

Selon la norme ISO FD X50-183 "Outil de management - Ressources humaines dans un système de management de la qualité", la compétence peut être définie comme "la capacité à mettre en œuvre des connaissances, savoir-faire et comportements en situation d'exécution". Elle est généralement synonyme de la capacité à résoudre des problèmes dans un contexte donné. Pour [De Montmollin, 1986], la compétence est "un ensemble hiérarchisé de savoirs, de savoir-faire, de conduites-types, de procédures standard, de types de raisonnements que l'on peut mettre en œuvre dans un apprentissage nouveau". [Levy-Leboyer, 1996] définit la compétence comme "la mise en œuvre intégrée d'aptitudes, de traits de personnalité et aussi de connaissances acquises pour mener à bien une mission complexe dans le cadre de l'entreprise". Dans [Lucia et Lepsinger, 1999], la compétence est considérée comme une combinaison d'un ensemble de "*skills*" avec d'autres ressources (connaissances, attitudes, variables de personnalités...). De ces définitions peuvent être déduites trois caractéristiques essentielles de la compétence :

- La compétence n'existe pas en soi, indépendamment de l'activité et du problème à résoudre. Elle est contextuelle, c'est-à-dire liée à une situation professionnelle donnée.
- La compétence aboutit à la production de résultats (réalisation d'une tâche) et à une performance reconnue.
- La compétence n'est pas livresque, elle n'existe que dans le cadre d'une expérience professionnelle et mobilise tout à la fois des savoirs théoriques et des capacités opératoires (ou savoir-faire) et collaboratives.

Deux types de compétences (connaissances techniques et compétences psychosociologiques) sont également distingués dans la constitution d'équipes.

Connaissance technique : ce sont des compétences liées à la connaissance et à l'expérience techniques concernant la réalisation des tâches, correspondant au domaine d'expertise ou au département fonctionnel auquel le membre de l'équipe appartient. La

modélisation des connaissances techniques est considérée comme primordiale dans le problème d'allocation des ressources humaines. Ainsi, nous avons noté l'utilisation des connaissances techniques dans la majorité des travaux sur la constitution d'équipes, par exemple [Hadj-Hamou et Caillaud, 2004], [Tseng *et al.*, 2004]; [Chen et Lin, 2004], [Zakarian et Kusiak, 1999], [Rose *et al.*, 2006].

Pour définir la connaissance technique, trois termes sont souvent utilisés dans la littérature : "domaine d'expertise" [Hadj-Hamou et Caillaud, 2004], [Tseng *et al.*, 2004]; [Chen et Lin, 2004], [Zakarian et Kusiak, 1999], "activité" [De Korvin *et al.*, 2002], "connaissance technique" [Gronau *et al.*, 2006], [Tsai *et al.*, 2003]. Ces trois termes représentent différents niveaux de granularité. Le "domaine d'expertise" représente le niveau le moins fin. L'"activité" représente le niveau intermédiaire et la "connaissance technique" représente le niveau le plus fin. Par exemple, dans [Zakarian et Kusiak, 1999], les auteurs caractérisent des tâches par des domaines d'expertise. Leur but est de définir/représenter les différents domaines d'expertise (exemple : ingénieur mécanique, ingénieur électronique, finance, etc.) nécessaires à chacune des tâches (exemple : concevoir le système de freinage).

Compétence⁵ psycho-sociologique : ce sont des capacités qui prennent en considération les aspects cognitifs dans le travail. Elles peuvent être utilisées pour caractériser les ressources humaines ou les tâches de conception afin d'assurer une meilleure performance des équipes du projet. Pour construire des équipes ayant une bonne synergie, plusieurs modèles de capacité opératoire ont été présentés dans la littérature.

Certains auteurs [Acuna et Juristo, 2004] distinguent plusieurs types de capacités :

- Capacités intra-personnelles, qui décrivent les comportements élémentaires d'un individu, développés par son éducation et la société, tels que le sens d'analyse, l'esprit de décision, l'indépendance, la créativité, la ténacité.
- Capacités interpersonnelles, qui décrivent les qualités de sociabilité, de coopération et de travail en groupe ainsi que la capacité de négociation.
- Capacités organisationnelles, qui traduisent les capacités de l'acteur à s'auto-organiser dans son activité, d'organiser les ressources de son environnement, par exemple, l'autonomie, la rigueur.
- Capacités managériales, qui concernent les comportements spécifiques essentiels pour manager tels que le leadership, la capacité de gestion d'équipes, la capacité de mener à bien un projet, et les capacités d'évaluation.

D'autres auteurs ont proposé des concepts permettant de caractériser les relations entre des individus dans une équipe (ou groupe) :

- Les cinq facteurs de personnalité (The Big Five) ont été présentés dans [Kichuk et Wiesner, 1997], [Barrick *et al.*, 1998], [Acuna *et al.*, 2009]. Les différentes dimensions du modèle sont les suivantes : névrosisme⁶ (Neuroticism), extraversion⁷, ouverture à l'expérience (Openness), caractère conciliant (Agreeableness), et caractère consciencieux (Conscientiousness). [Acuna *et al.*, 2009] étudient l'impact de cinq

⁵ Nous considérons que le terme de compétence, utilisé ici et repris par de nombreux auteurs, est un abus de langage. Il s'agit plus d'une capacité puisqu'en elle-même, elle ne peut rien "produire".

⁶ Contraire de la stabilité émotionnelle : tendance à éprouver facilement des émotions désagréables comme la colère, l'inquiétude ou la dépression, vulnérabilité.

⁷ Energie, émotions positives, tendance à chercher la stimulation et la compagnie des autres, fonceur

facteurs de personnalité sur la qualité du produit et sur la satisfaction des membres de l'équipe.

- L'indicateur MBTI (Myers-Briggs Type Indicator) a été introduit dans les années 1930 par Carl Gustav Jung (1875-1961). Il identifie 16 grands types de personnalité à partir des deux préférences possibles sur chacune des quatre dimensions.
 - Orientation de l'énergie : Extraversion/ Introversion
 - Recueil d'information : Sensation / Intuition
 - Prise de décision : Pensée / Sentiment
 - Mode d'action : Jugement / Perception
- L'indicateur KCI (Kolbe Conative Index) [Fitzpatrick et Askin, 2005] identifie et explique comment un individu résout un problème, organise ses idées et ses objets, et utilise son temps et son énergie. Le concept KCI classe les comportements d'un individu en quatre catégories: probing (chercheur), patterning (organisateur), innovating (innovateur), demonstrating (metteur en scène).
- Les rôles dans une équipe ont souvent été modélisés. la constitution d'équipes ne concerne pas uniquement les connaissances techniques. Pour avoir une équipe réussie dans la conception de produits nouveaux, il est nécessaire de réunir des équipes combinant plusieurs rôles [Boucher *et al.*, 2006]. L'affectation des rôles dans une équipe a pour but de déterminer des mécanismes de coopération et de compréhension mutuelle. [Belbin, 1981] définit huit types de rôles dans une équipe⁸ : chairman, shaper, plant, monitor-evaluator, resource investigator, team worker, company worker and completer-finisher. Le bon équilibre dans la distribution de ces rôles doit garantir la bonne efficacité de l'équipe. Certains travaux relatifs à la constitution d'équipes sont basés sur le modèle de Belbin [Stickley et Grabot, 1994], [Hadj-Hamou et Caillaud, 2004], [Sommerville et Dalziel, 1998].

Les compétences peuvent être identifiées à la fois sur la tâche (compétences requises) et sur l'acteur (compétences acquises). La première identification est l'identification des tâches (quelle compétence est requise par quelle tâche ?) et la deuxième est l'identification de compétences acquises par les acteurs de l'organisation (quelle compétence est acquise par quel acteur ?).

- Les **compétences requises** sont les compétences nécessaires pour la réalisation de tâches, de missions ou d'actions stratégiques. La compétence requise par une tâche a un caractère évolutif. Elle évolue par rapport à la référence déterminée par des benchmarks avec l'environnement extérieur (par exemple, les pratiques de ses concurrents, les besoins de ses clients, les réglementations ...) ce qui peut augmenter le niveau requis de maîtrise de la compétence concernée. De nouvelles tâches peuvent apparaître, créant ainsi de nouveaux besoins en compétences.
- Les **compétences acquises** sont celles que possède l'individu. Elles sont basées sur le référentiel de compétences de l'entreprise. Le but du référentiel est de déterminer quelles sont les compétences que maîtrisent les salariés, et ce, à quel niveau. Des échelles de compétence ont été créées pour permettre l'évaluation du niveau de compétence d'un individu.

⁸ La traduction de ces termes étant délicate en français, nous préférons ici rapporter directement les termes en anglais.

1.4.2 Notre définition de la "compétence"

Dans cette partie, nous présentons les concepts de mission et de compétence, en nous appuyant sur des travaux antérieurs de notre équipe [Bonjour, 2008a], [Belkadi, 2006].

1.4.2.1 Définition d'une mission

La définition d'une mission comporte une tâche à réaliser⁹, spécifiée sous forme d'objectifs (ou attentes) portant sur le résultat à atteindre, des exigences portant sur les actions à réaliser, un contexte d'action et des critères d'appréciation de la réussite de la mission. Pour remplir sa mission, un acteur décompose la mission en une coordination de sous-buts qui seront affectés à des acteurs collectifs de plus petit cardinal ou des acteurs individuels. Un acteur est alors investi d'une mission, c'est-à-dire de la responsabilité de réalisation d'une activité dans une situation donnée.

1.4.2.2 Définition de la compétence

Les concepts de compétence que nous proposons ici s'appuient sur deux visions : vue fonctionnelle et vue structurelle.

Vue fonctionnelle

La compétence peut être considérée comme un système [Bonjour, 2008b]. Elle n'existe que parce qu'une personne (un modélisateur) a une intention de compréhension, d'action ou d'évaluation, sur ce qu'il appelle une compétence. Dans le contexte professionnel, la finalité d'une compétence est de permettre à l'acteur qui la possède de réussir une mission qui lui a été confiée.

Nous retenons la définition suivante de la compétence [Bonjour *et al.*, 2007] : "La compétence est la mobilisation d'un ensemble de savoirs hétérogènes aboutissant à la production d'une performance reconnue, par rapport à un environnement donné et dans le cadre d'une activité finalisée." L'activité finalisée permet de relier la compétence à la tâche à réaliser.

Vue structurelle

La vue structurelle permet de mettre en évidence l'existence de trois types de composants importants dans la caractérisation d'une compétence : les connaissances techniques, les capacités opératoires et le plan d'actions.

- Les connaissances techniques concernent les savoirs théoriques, ou pratiques relatifs aux entités impliquées dans l'interaction
- Les capacités opératoires concernent quatre caractéristiques dans le cadre de l'activité d'un concepteur : la capacité d'analyse, la capacité d'organisation, la capacité relationnelle et la capacité décisionnelle [Belkadi, 2006].

⁹ Dans la suite de ce mémoire, nous assimilerons la mission à la tâche à laquelle elle se rapporte.

- Un plan d'actions est une manière possible d'organiser la réalisation d'une tâche. Un plan d'actions est défini comme l'organisation retenue par un acteur, pour mener à bien une mission. Au cours de l'activité, le plan d'actions sera affiné ou modifié, pour tenir compte des conditions réelles qui ne peuvent pas forcément être prévues initialement.

La compétence est indissociable de la mission ou de la tâche qu'elle permet de réaliser. Bien piloter les compétences revient d'une part, à avoir une bonne représentation de la tâche et d'autre part, à bien prendre en compte ses composants internes.

Les différents concepts présentés ci-dessus permettent de comprendre les différentes facettes des compétences selon une vue statique. Nous présentons maintenant la vue dynamique des compétences (ou encore, l'évolution des compétences).

1.4.3 Mesures de similarité

Plusieurs mesures de distance sont basées sur la norme p (aussi appelée distance de Minkowski) qui s'écrit de façon générique : $\left(\sum_{c=1}^C |R_{ic}^1 - R_{cj}^2|^p \right)^{1/p}$.

La distance de Hamming (aussi considérée comme la distance de Manhattan, dite norme 1) correspond au cas particulier où $p = 1$: $\sum_{c=1}^C |R_{ic}^1 - R_{cj}^2|$.

Certains travaux peuvent utiliser une norme-2 (aussi appelée distance d'Euclidienne) : $\left(\sum_{c=1}^C |R_{ic}^1 - R_{cj}^2|^2 \right)^{1/2}$.

1.4.4 Introduction à la dynamique des compétences

Dans cette partie, nous introduisons une réflexion sur la dynamique des compétences. Nous avons placé en annexe 5 des travaux concernant des modèles de l'évolution des compétences (courbes d'apprentissage et de régression des compétences).

Il est important de rappeler que la compétence d'un acteur a un cycle de vie. Elle s'acquiert, se mesure, s'observe dans l'exercice d'une activité et se dégrade si elle n'est pas entretenue. Il est donc important de maintenir les compétences acquises ou de développer de nouvelles compétences.

Il est essentiel aussi de noter que la dynamique des compétences est liée, à la fois, à la tâche, à l'acteur et à l'environnement compétitif.

- Le besoin de réaliser de nouvelles tâches crée de nouveaux besoins en compétences. Pour améliorer ses performances, une entreprise doit innover en acquérant de nouvelles technologies, en développant de nouveaux produits ou en renforçant la maîtrise de ses connaissances techniques. Elle doit également améliorer continuellement son organisation pour mieux répondre aux besoins des clients. Ce faisant, l'entreprise fait constamment évoluer ses processus ou ses tâches.
- Il y a un lien fort entre l'affectation des tâches aux ressources et la compétence. Si un acteur est affecté aux tâches, son niveau de compétence augmente et si l'acteur n'est plus affecté aux tâches, son niveau de compétence va diminuer.

- Une compétence produit un résultat qui est jugé dans un environnement compétitif. En ce sens, évaluer un niveau de compétence fournit une valeur relative par rapport à ce que l'on estime être une référence dans l'environnement. Cette référence doit être estimée par des benchmarks permettant de situer les pratiques de l'entreprise par rapport à ses concurrents et à ce qu'attendent dans le futur les autres parties prenantes (clients, organismes de réglementation, actionnaires ...). Cette cible est donc évolutive. Si un acteur ne remet pas en question sa compétence alors elle se dégrade "simplement" parce les autres vont faire mieux.

Le concept de compétence ayant été introduit, nous résumons dans la partie suivante un travail classifiant différents axes de travaux du "pilotage des compétences" afin de positionner notre contribution de la constitution d'équipes par rapport à cette thématique.

1.4.5 Différents axes de pilotage des compétences

Dans une récente publication, [Boucher *et al.*, 2006] ont proposé une classification des différents axes de pilotage des compétences (TAB. 1.2). Cette typologie a été structurée selon le niveau de décision du pilotage des compétences, ainsi que sur trois vues systémiques comportant trois vues (structurelle, fonctionnelle et dynamique).

- Le niveau **stratégique** concerne les décisions sur le management stratégique (formulation de stratégies, identification des compétences stratégiques, etc.). Le but du management stratégique est d'augmenter la compétitivité durable de l'entreprise. Ce thème de recherche est largement abordé par les sciences du management. Il s'agit de conceptualiser les prises de décisions stratégiques et de développer des outils d'aide au diagnostic stratégique externe et interne.
 - La vue structurelle présente un aspect statique concernant l'identification des compétences clés et la définition des objectifs stratégiques.
 - La vue fonctionnelle concerne la mobilisation des compétences, par exemple, le déploiement des stratégies au travers des compétences collectives, le management des relations inter-entreprise.
 - La vue dynamique s'intéresse à la dynamique des compétences (définition de la trajectoire des compétences adaptée aux nouvelles technologies, outils, méthodes, etc.).
- Le niveau **tactique** concerne les décisions sur la conception de l'organisation. Le but de concevoir des structures organisationnelles (métiers, équipes projets, ...) pour répondre aux objectifs de conception ayant une relative autonomie d'action et devant développer des compétences pour faire face à des besoins futurs.
 - La vue structurelle s'intéresse à l'identification des compétences disponibles, à l'évaluation des compétences et à l'intégration de ce concept dans la mesure de performance [Harzallah et Vernadat, 2002], [Berio et Harzallah, 2007], [Ley et Albert 2003].
 - La vue fonctionnelle concerne la constitution d'équipes, l'affectation des tâches ou la sélection des partenaires dans le cadre des organisations distribuées. Le problème de constitution d'équipes dans le thème du pilotage des compétences est souvent considéré comme un problème d'affectation. Les travaux de [Zakarian et Kusiak, 1999], [Tseng *et al.*, 2004] [Hadj-Hamou et Caillaud,

2004], [Gronau *et al*, 2006], [Chen et Lin, 2004], [Chen, 2005] se retrouvent dans ce thème, concernant la constitution d'équipes de conception.

- La vue dynamique s'intéresse à l'anticipation de nouveaux rôles ou de nouvelles tâches et à la préparation de plan d'actions pour développer les compétences.

	Vue structurelle	Vue fonctionnelle	Vue dynamique
Stratégique	Identification des compétences clés Définition des objectifs stratégiques	Déploiement de la stratégie selon les compétences collectives Pilotage des relations inter-firmes	Trajectoires d'évolution des compétences en fonction des évolutions technologiques Le cycle de vie du système de pilotage
Tactique	Identification des compétences	Constitution d'équipes, Allocation des compétences	Anticipation des évolutions du système des compétences
	Evaluation des compétences Analyse des forces et des faiblesses dans les processus de l'organisation	Simulation des processus à base de compétences Sélection de partenaires dans les organisations distribuées	Définition des plans d'actions pour le développement des compétences
Opérationnel	Vision locale sur les performances Analyse des aspects psychologiques et des mécanismes sociaux : autonomie, réactivité, ...		Apprentissage dans l'action Formation

TAB. 1.2 : Typologie des thèmes du pilotage des compétences [Boucher *et al.*, 2006]

- Le niveau **opérationnel** concerne les actions de pilotage en lien avec la réalisation des activités.
 - Les vues structurelle et fonctionnelle s'intéressent à un aspect statique de la compétence. Elles concernent la mise en œuvre de mécanismes managériaux permettant de faciliter, suivre ou d'orienter l'activité des acteurs et de remonter des informations au niveau tactique (par exemple, besoin de réaffectation en cas d'absence d'un acteur)
 - La vue dynamique s'intéresse au suivi des plans d'actions exécutés et au recueillement des besoins des employés en termes de formation ou de plan de carrière.

Le tableau 1.2 ci-dessus résume la typologie des thèmes du pilotage des compétences proposées par [Boucher *et al.*, 2006].

1.5 Premier positionnement

La constitution d'équipes s'intègre dans la chaîne décisionnelle de pilotage d'un projet de conception. Les chefs de projets doivent manipuler une grande quantité d'information sur les tâches et les acteurs ainsi que des couplages difficiles à prendre en compte dans leur globalité avec l'architecture du produit à concevoir. La constitution d'équipes est donc une activité complexe qui nécessite des modèles et des outils appropriés visant à aider la prise de décision.

Dans la partie 1.3, nous avons présenté une première approche de la constitution d'équipes concernant la thématique de la structuration du projet, à savoir l'identification d'une architecture des acteurs.

Dans la partie 1.4, nous avons fait apparaître la deuxième approche de la constitution d'équipes concernant la thématique du pilotage des compétences. Notre travail se positionne sur le pilotage des compétences au niveau tactique, et plus particulièrement la constitution d'équipes qui nécessite au préalable une identification des compétences, ainsi qu'une définition des évolutions souhaitées du système de compétences.

La plupart des travaux de la constitution d'équipes partagent un caractère en commun. Ils considèrent le problème d'affectation comme un sous-problème permettant d'allouer les ressources en fonction des objectifs souhaités et des contraintes imposées.

Jusqu'à présent, la problématique de la constitution d'équipes n'a pas été abordée dans sa globalité. D'une part, les problèmes orientés sur la structuration du projet n'intègrent que la modélisation des tâches ou des interactions entre les acteurs (le problème d'affectation n'est pas traité). D'autre part, les problèmes centrés sur le pilotage des compétences (problème d'allocation) ne cherchent qu'à satisfaire des contraintes portant sur l'adéquation des compétences ou capacités aux tâches à réaliser.

Le schéma ci-dessous présente le positionnement de notre étude au carrefour de deux thématiques de recherche : la structuration du projet et le pilotage des compétences.

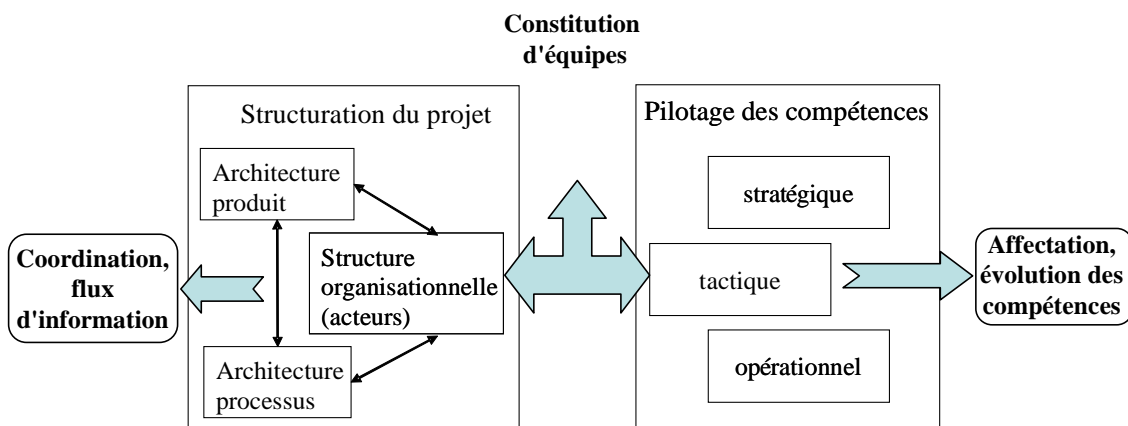


FIG. 1.3 : Positionnement global de notre recherche

Nous souhaitons articuler nos travaux sur la constitution d'équipes selon deux approches complémentaires.

- La première concerne la structuration du projet (identification de l'architecture des acteurs) en se basant sur des représentations matricielles. Ces travaux cherchent à améliorer les flux d'informations et minimiser les efforts de coordination entre les concepteurs.

- La deuxième est relative au pilotage des compétences, plus particulier à l'affectation des tâches à des acteurs. Ces travaux cherchent à constituer des équipes en prenant en compte les compétences des ressources humaines et éventuellement, leur évolution.

Le chapitre 2 présente plus en détail ces deux approches relatives à la constitution d'équipes.

2 Etat de l'art sur la constitution d'équipes

Ce chapitre est organisé en trois parties :

- La première partie du chapitre présente un état de l'art sur la constitution d'équipes contribuant à la structuration du projet. Différentes méthodes de l'identification de l'architectures des acteurs seront présentées, selon les types de couplage considérés reliant ces acteurs (flux des informations entre acteurs, contraintes liées à l'architecture du produit). Une synthèse succincte sera également présentée afin de préciser les aspects auxquels nous nous intéresserons.
 - La deuxième partie du chapitre présente une étude bibliographique de la constitution d'équipes contribuant au pilotage des compétences. Nous présentons ces travaux selon le type de compétence modélisée (connaissances techniques, capacités opératoires, etc.). Une synthèse sera également présentée.
 - Dans la dernière partie, nous présentons un positionnement plus précis de notre problématique et de nos contributions.
-

2.1 Constitution d'équipes contribuant à la structuration du projet

L'identification d'une architecture des acteurs est considérée comme une façon de regrouper des concepteurs en équipes (constitution d'équipes). Les travaux de recherche qui se sont intéressés à l'architecture des acteurs peuvent être classés en deux grandes catégories :

- Travaux traitant la minimisation des flux d'informations et des efforts de coordination entre équipes [McCord et Eppinger, 1993], [Morelli *et al.* 1995], [Sosa *et al.* 2003].
- Travaux traitant le couplage entre les architectures du produit et des acteurs [Gulati et Eppinger, 1996], [Eppinger et Salminen, 2001], [Oosterman, 2001], [Sosa *et al.*, 2004], [Browning, 1999], [Yassine et Wissman, 2007], [Harmel, 2007].

2.1.1 Travaux traitant la minimisation des flux d'informations

Les travaux dans cette catégorie s'intéressent à la conception des organisations basée sur la modélisation du domaine des acteurs par des DSM acteurs. [McCord et Eppinger, 1993], [Morelli *et al.* 1995], [Sosa *et al.* 2003] s'intéressent à la modélisation DSM acteur dans le but de la minimisation des efforts de coordination sur le projet. Ces travaux cherchent à améliorer les flux d'informations et de communication entre concepteurs et utilisent des DSM-acteurs pour regrouper, au sein d'une même équipe, les acteurs fortement interdépendants et minimiser les efforts de coordination entre équipes. Les applications pouvant être envisagées à partir de ces travaux sont relatives à la conception d'organisation, et plus particulièrement à la gestion des interfaces organisationnelles existant entre acteurs.

Le travail de [McCord et Eppinger, 1993] porte sur la construction d'une DSM acteurs en partant des tâches que les acteurs réalisent. Ils proposent alors de quantifier les couplages entre les acteurs par l'intensité des flux d'informations sur différents niveaux (élevé, moyen et bas). Par exemple, le niveau de flux élevé est défini par plusieurs communications par semaine.

La formalisation de la numérisation des DSM acteurs a été réalisée quand les premières taxonomies des types d'interactions entre acteurs ont été proposées, ces taxonomies servaient de base pour la proposition de métriques. [Morelli *et al.*, 1995] ont utilisé la "qualité" comme métrique pour qualifier les moyens de communications entre les acteurs. Les travaux les plus récents quant à eux utilisent beaucoup plus la fréquence des interactions comme métrique [Sosa *et al.* 2003].

2.1.2 Travaux traitant le couplage entre l'architecture du produit et l'architecture des acteurs

Les travaux portant sur cette thématique visent à prendre en compte des contraintes de l'architecture du produit dans la conception de l'organisation d'un projet.

Dans une situation de conception innovante, l'architecte va devoir construire une architecture préliminaire du produit et faire des choix d'affectation de tâches de conception à certains acteurs. Le problème est que bien souvent, il ne peut pas anticiper l'impact de ses choix concernant l'architecture du produit sur les efforts de coordination qui seront nécessaires entre les équipes.

Dès 1977, Galbraith [1977] a souligné la forte interdépendance entre l'architecture du produit et l'organisation du projet de conception. Récemment, [Sosa *et al.*, 2003], [Sosa, 2007] ont analysé cette interdépendance en se focalisant sur les effets de l'architecture du produit sur les communications techniques entre les concepteurs. Des recommandations ont été formulées pour adapter conjointement les architectures des différents domaines du projet. Cependant, peu de travaux ont étudié ou formalisé cette relation. Il n'existe pas d'outil formel permettant d'optimiser le couplage entre les domaines ou de propager des évolutions d'un domaine vers un autre.

Certains travaux [Gulati et Eppinger, 1996], [Eppinger et Salminen, 2001], [Oosterman, 2001], [Sosa *et al.*, 2004]; [Browning, 1999], [Browning, 2001], [Yassine et Wissman, 2007] formulent des recommandations sur la façon d'organiser un projet en tenant compte des interactions entre les modules du produit et des barrières organisationnelles (cloisonnement) qui limitent parfois la communication entre les concepteurs. Certains se placent sur un aspect stratégique de la modularisation quand il s'agit de décider de sous-traiter la conception et la réalisation d'un module [Sanchez et Mahomey, 1996]. [Sosa *et al.* 2004] proposent un outil de diagnostic de l'organisation basé sur la comparaison des interactions réelles entre acteurs avec les interactions attendues en anticipant les choix d'architecture sur le produit. [Harmel, 2007] propose une méthode qui permet de propager des modifications réalisées sur deux domaines (architecture du produit et architecture des acteurs) afin d'assurer une co-évolution cohérente de ces deux domaines.

2.1.3 Synthèse

L'analyse bibliographique précédente résume les différentes façons d'identifier l'architecture des acteurs. Elle distingue deux types de problématiques dans la conception de l'architecture des acteurs : les problématiques orientées "flux d'information entre les acteurs" et les problématiques orientées "couplage entre l'architecture du produit et l'architecture des acteurs". On peut observer que l'architecture des acteurs peut faire émerger une dimension collective des compétences des équipes du projet.

Cette étude montre l'importance de la prise en compte de l'architecture du produit dans la conception de l'architecture des acteurs. Malgré cette importance, nous constatons que très peu de travaux s'intéressent à développer une méthode formelle permettant de propager les contraintes d'une architecture du produit sur une architecture des acteurs ou de les faire co-évoluer. La complexité du couplage entre les architectures du produit et des acteurs explique certainement que peu de travaux de recherche ont été développés sur cette thématique. Il est donc pertinent de développer des méthodes de constitution d'équipes tenant compte de ce couplage, afin d'aider les chefs de projets à prendre des décisions plus globales, permettant d'améliorer alors les performances du projet.

2.2 Constitution d'équipes contribuant au pilotage des compétences

Nous avons classé les travaux en plusieurs catégories :

- Maximisation des **connaissances techniques** [De Korvin *et al.*, 2002], [Gronau *et al.*, 2006], [Coates *et al.*, 2007], [Otero *et al.*, 2009], [Alliger *et al.*, 1993], [Wu, 2007], [Huang *et al.*, 2009]. Le but est de constituer des équipes techniquement performantes.
- Maximisation des **capacités opératoires (compétences psycho-sociologiques)** dans une équipe [Kichuk et Wiesner, 1997], [Barrick *et al.*, 1998], [Shipley et Johnson,

2009]. Ces travaux visent à former une équipe garantissant une bonne synergie entre les profils de personnalité des membres de l'équipe et un recouvrement de tous les rôles nécessaires.

- Maximisation des **connaissances techniques et capacités opératoires** [Acuna et Juristo, 2004], [Chen et Lin, 2004], [Chen, 2005], [Hadj-Hamou et Caillaud, 2004], [Wi *et al.*, 2009], [Fan *et al.*, 2009]. Les modèles proposés dans ces travaux considèrent les deux compétences en même temps.
- Maximisation de la satisfaction des clients en constituant une équipe en fonction des **domaines d'expertise** (ou métiers) [Zakarian et Kusiak, 1999], [Tseng *et al.*, 2004].

2.2.1 Maximisation des connaissances techniques

Plusieurs travaux s'orientent vers ce type de maximisation, dont le but est de constituer des équipes performantes techniquement.

[De Korvin *et al.*, 2002] ont développé un modèle de sélection du personnel pour les projets multi-phases. Ce modèle prend en compte à la fois, les connaissances techniques, les contraintes de précédence et les contraintes budgétaires (salaire du personnel). Ce modèle maximise les connaissances techniques globales des ressources allouées au projet tout en tenant compte des contraintes budgétaires. C'est une affectation de type "plusieurs à plusieurs". Les connaissances techniques ont été modélisées par des ensembles flous pour chaque phase du projet. [Alliger *et al.*, 1993] ont également utilisé la théorie des ensembles flous pour la sélection du personnel.

[Coates *et al.*, 2007] proposent un modèle de sélection de ressources dans l'équipe de projet de conception. L'approche commence par identifier les compétences des ingénieurs et quantifie leurs niveaux de compétence. Ensuite, un algorithme génétique est utilisé pour former l'équipe de concepteurs. Une fois l'algorithme terminé, les auteurs identifient le manque d'acteurs qualifiés dans certaines compétences et proposent les scénarios d'amélioration (recrutement, développement des acteurs) pour les compétences manquantes. Les scénarios sont évalués par rapport au coût du salarié et à la durée du projet.

Dans le domaine de la conception de logiciels, [Otero *et al.*, 2009] proposent un modèle d'affectation des programmeurs à l'aide de l'identification des compétences nécessaires pour la réalisation de tâches, de l'identification des compétences acquises par l'acteur et d'un indicateur de compatibilité entre tâches et acteurs. Le modèle vise à maximiser la performance d'équipes (ce qui revient à minimiser le temps d'apprentissage de l'équipe) en choisissant les acteurs les plus qualifiés. [Gronau *et al.* 2006] ont proposé une méthode de formation d'équipe basée sur une taxonomie (réseau sémantique et base de connaissances), ainsi que sur la modélisation des processus. La figure FIG. 2.1 présente les différentes catégories de liens utilisées dans la taxonomie proposée par les auteurs.

Ces liens et leur signification sont les suivants :

- Synonyme : La compétence 6 est de même niveau que la compétence 8.
- Agrégation : La compétence 7 comprend les compétences 8 et 9. Si une personne possède la compétence 7, elle possède soit la compétence 8, soit la compétence 9, soit les deux compétences.
- Composition : La compétence 1 se compose des compétences 2 et 3. Si une personne possède la compétence 1, alors elle possède également les compétences 2 et 3.

- Généralisation/spécialisation : La compétence 4 est plus générique que les compétences 5 et 6.
- Non défini : Il n'y pas de lien entre la compétence 0 et les compétences 1, 4 ou 7. Si quelqu'un possède la compétence 0, l'algorithme de la constitution d'équipes ne construira pas les associations avec les compétences 1, 4 or 7 et vice versa.

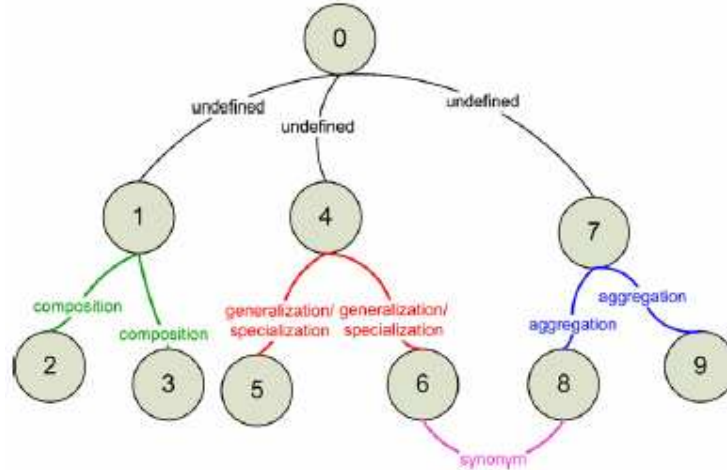


FIG. 2.1 : Catégories de liens de compétences dans le réseau sémantique [Gronau *et al.*, 2006]

[Barreto *et al.*, 2008] examinent le problème d'affectation par la méthode de satisfaction de contraintes, basée sur les fonctions d'utilité. Le but est de trouver la meilleure solution pour l'affectation selon les différents critères demandés. Une fonction d'utilité représente un critère à optimiser. Les fonctions doivent être maximisées ou minimisées selon les membres d'équipe sélectionnés. Les fonctions d'utilité proposées sont :

- o la meilleure équipe (où le niveau de sur-qualification de l'ensemble des membres sélectionnés est minimal),
- o l'équipe la moins chère (où le budget est le plus faible),
- o l'équipe la moins qualifiée (où le niveau de sous-qualification de l'ensemble des membres sélectionnés est minimal),
- o l'équipe la plus petite (où le nombre de membres est le plus petit),
- o l'équipe la plus rapide (où la durée de travail du projet estimée est minimale),
- o la meilleure équipe qualifiée partielle (où l'on autorise à affecter des acteurs ayant des niveaux de maîtrise dans certaines compétences inférieurs à ceux demandés par des tâches).

2.2.2 Maximisation des capacités opératoires

Certains travaux de la constitution d'équipes s'intéressent aux comportements des acteurs (profils de personnalité) dans le but d'améliorer la performance d'une équipe. Plusieurs concepts déjà présentés dans la partie (§1.4.1) ont été utilisés dans des travaux de constitution d'équipes, en particulier : les cinq facteurs de personnalité [Kichuk et Wiesner, 1997], [Barrick *et al.*, 1998], l'indicateur MTBI [Shipley et Johnson, 2009] et les rôles de Belbin [Henry et Stevens, 1999].

Nous avons constaté que peu de travaux de recherche en conception traitent uniquement les capacités opératoires. De nombreux auteurs se sont intéressés à considérer deux types de compétences en même temps. Nous aborderons ces travaux dans la partie suivante.

2.2.3 Maximisation des connaissances techniques et capacités opératoires

Plusieurs travaux concernant la constitution d'équipes modélisent à la fois les connaissances techniques et les capacités opératoires. En maximisant les connaissances techniques pour constituer une équipe, ces modèles ont également maximisé les capacités opératoires. Nous présentons certains travaux significatifs ci-après.

2.2.3.1 Modèle contribuant à l'évaluation des connaissances techniques et des capacités opératoires.

[Acuna et Juristo, 2004], [Acuna *et al.*, 2006] proposent leur propre évaluation sur des compétences non techniques pour l'aide à la constitution des équipes de projets de développement informatique. Le modèle évalue successivement les acteurs à travers deux tests de compatibilité de compétences non techniques : le test du profil de personnalités et le test du rôle dans l'équipe. L'affectation des tâches aux ressources est réalisée sous certaines conditions de compatibilité : si la compatibilité est supérieure à 50, on peut affecter la tâche à cette ressource, sinon l'affectation peut avoir lieu mais nécessite la formation supplémentaire. La validation du modèle est effectuée à travers deux cas d'étude (l'un affecte des tâches par le jugement des experts et l'autre utilise le modèle d'évaluation défini ci-dessus). Les deux cas sont évalués par rapport à certains critères tels que la performance des ressources, le temps du développement et les défauts trouvés pendant la phase de contrôle. Ce modèle ne prend pas en compte la collaboration entre acteurs, ni la modélisation des tâches.

2.2.3.2 Modèle contribuant à la maximisation des deux types de compétences en prenant en compte la collaboration dans l'équipe

Dans le contexte de la conception collaborative, [Hadj-Hamou et Caillaud, 2004] ont proposé un modèle de la constitution d'équipes dans le but de maximiser le niveau de compétences en assurant le recouvrement des métiers et des rôles dans l'équipe. Le modèle cherche à construire une équipe de conception ayant une personne qualifiée dans chaque discipline (mécanique, informatique, électronique, etc.) et couvrant chaque rôle. Les rôles dans une équipe sont définis selon la typologie de [Belbin, 1981]. Un problème de programmation linéaire a été formulé pour assurer la diversité des rôles et des disciplines nécessaires dans une équipe.

Un projet comprend plusieurs acteurs i ($i=1,\dots,m$), disciplines j ($j=1,\dots,n$) et rôles k ($k=1,\dots,p$). Chaque acteur i peut maîtriser une ou plusieurs disciplines j et jouer un ou plusieurs rôles k , les variables x_{ij} et y_{ik} sont des variables de décision binaires pour la discipline et le rôle, respectivement.

$$x_{ij} = \begin{cases} 1 & \text{si l'acteur } i \text{ est sélectionné pour travailler dans la discipline } j \\ 0 & \text{sinon} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{si l'acteur } i \text{ est sélectionné pour jouer le rôle } k \\ 0 & \text{sinon} \end{cases}$$

La fonction objectif (1) est de maximiser le niveau de compétence dans les disciplines et le niveau de coopération des membres d'équipe. La valeur a_{ij} est un niveau de maîtrise acquis dans la discipline j par l'acteur i ($0 \leq a_{ij} \leq 1$), la valeur b_i est un niveau de coopération de l'acteur i ($0 \leq b_i \leq 1$). Pour trouver un compromis entre le niveau de compétence et celui de coopération, le niveau de compétence dans les disciplines est pondéré par α et le niveau de coopération est pondéré par $(1 - \alpha)$.

$$\text{Maximiser } \alpha \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} + (1 - \alpha) \sum_{i=1}^m \sum_{j=1}^n b_i x_{ij} \quad (1)$$

La contrainte (2) assure qu'il y a seulement un acteur qui est affecté à chaque discipline. La contrainte (3) assure qu'il y a seulement un acteur pour chaque rôle d'équipe.

$$\sum_{i=1}^m x_{ij} = 1, \forall j = 1, \dots, n, \quad (2)$$

$$\sum_{i=1}^m y_{ik} < 2, \forall k = 1, \dots, p, \quad (3)$$

Les contraintes (4), (5) et (6) imposent que l'acteur sélectionné ne maîtrise qu'une seule discipline et ne joue qu'un seul rôle dans l'équipe du projet.

$$\sum_{j=1}^n x_{ij} \leq 1, \forall i = 1, \dots, m, \quad (4)$$

$$\sum_{k=1}^p y_{ik} \leq 1, \forall i = 1, \dots, n, \quad (5)$$

$$\sum_{j=1}^n x_{ij} = \sum_{k=1}^p y_{ik}, \forall i = 1, \dots, m \quad (6)$$

[Wi *et al.*, 2009] proposent un modèle d'évaluation des compétences (savoir-faire, savoir-être et savoir-qui) des acteurs permettant de sélectionner les meilleurs acteurs en fonction de mots-clés définissant une mission. Cela permet de distinguer les profils des rôles dans l'équipe (profils manager/intégrateur, profils spécialiste, etc.). [Fan *et al.*, 2009] proposent également un modèle de sélection des membres d'équipe en tenant compte de leurs connaissances techniques et leurs capacités de collaborer avec d'autres membres.

2.2.3.3 Modèle contribuant à la maximisation des deux types de compétences prenant en compte la modélisation des tâches

Dans le contexte de l'ingénierie concurrente, [Chen, 2005] a proposé un modèle relativement complet pour la constitution d'équipes de conception. Ce modèle cherche à

maximiser le niveau de capacités opératoires tout en assurant le niveau de connaissance technique. Trois sous-modèles ont été proposés : modèle du projet, modèle des membres d'équipe, modèle d'affectation.

Le **modèle du projet** analyse la séquence et la dépendance des tâches du projet. Le modèle identifie les séquences temporelles des tâches, montre l'indépendance et la dépendance entre tâches et décompose de grands groupes de tâches en des groupes plus petits, dans le but de faciliter la constitution d'équipes ultérieurement. Trois types d'enchaînements des tâches ont été identifiés : indépendant, dépendant, interdépendant.

Le problème de la constitution d'équipes peut être formulé avec ou sans caractérisation des tâches. Néanmoins, la modélisation des tâches (et du processus) est l'une des premières étapes dans la constitution d'équipes, visant à s'assurer que les tâches seront réalisées pour répondre à une mission clairement identifiée (par exemple, satisfaire les besoins des clients). Les tâches du processus peuvent être regroupées en modules de tâches interdépendantes, ce qui permet de les affecter à une équipe en limitant les efforts de coordination ou les itérations souvent longues entre équipes.

Le **modèle des membres d'équipe** est un modèle qui caractérise les compétences d'un membre d'équipe. Les compétences d'un membre d'équipe sont classées en trois catégories: la connaissance technique, la capacité de travailler en équipe (l'expérience dans le travail en équipe, la capacité de communication, la flexibilité) et la capacité relationnelle (profil de personnalité). La capacité de travailler en équipe et la capacité relationnelle sont des capacités opératoires. L'indicateur MBTI (Myers-Briggs Type Indicator) a été utilisé pour évaluer ce dernier type de capacité, la capacité relationnelle.

Le **modèle d'affectation** est un modèle qui affecte des tâches aux ressources. Les hypothèses utilisées dans ce modèle sont les suivantes : chaque tâche (définie par un type de département) nécessite un seul acteur (on ne fractionne pas la tâche), chaque acteur peut être affecté à plusieurs tâches, tant qu'il ne dépasse pas sa capacité de travail. L'auteur distingue deux types d'affectation selon la dépendance de la tâche : tâches indépendantes et tâches dépendantes (ou interdépendantes).

- Pour l'affectation des tâches indépendantes, l'affectation d'un membre à une tâche considère uniquement les connaissances techniques. La réalisation des tâches indépendantes ne requiert pas la capacité à travailler en équipe puisque le membre choisi est supposé travailler individuellement sur la tâche. On sélectionne donc le membre qui a le niveau de connaissance technique le plus élevé et le niveau de capacité à travailler en équipe le plus bas.
- Pour l'affectation des tâches dépendantes et interdépendantes, l'affectation d'un ensemble de tâches à un ensemble de membres prend en compte les deux types de compétences (connaissances techniques et capacités de travailler en équipe). L'algorithme d'affectation travaille en deux étapes. On sélectionne les membres qui ont le niveau de connaissance technique suffisant, puis on optimise leur capacité de travail en équipe et leur compatibilité selon leur personnalité.

Formulation mathématique

Un projet possède n tâches dans une entreprise ayant p départements. On notera un ensemble de tâches $x_i (i = 1, \dots, n)$, un ensemble de départements $D_k (k = 1, \dots, p)$ et la charge induite par les tâches représentée par les $W_i (i = 1, \dots, n)$.

Pour augmenter la performance d'équipe dans l'affectation des tâches dépendantes, la connaissance technique sera d'abord considérée, tous les acteurs utilisés dans le modèle sont qualifiés pour la tâche (par exemple, niveau de compétence >0.8). Après le choix des acteurs qualifiés, on élabore le modèle mathématique.

La procédure d'affectation détaillée, dans un groupe de m tâches dépendantes, est définie ci-dessous :

- (1) on détermine les départements correspondant aux m tâches (une tâche est associée à un unique département en considérant les connaissances techniques). Pour chaque groupe de tâches, m départements fonctionnels seront déterminés : $\{D_1, D_2, \dots, D_m\}$.
- (2) $j=1$ (on va allouer des ressources pour chaque groupe de tâches)
- (3) pour chaque tâche, on sélectionne tous les membres qui peuvent être qualifiés (les membres qui possèdent le niveau de compétence dans le département D_j supérieur à un seuil de compétence R_m et qui ont une capacité supérieure ou égale à la charge de la tâche W_i), ensuite on place ces membres candidats à l'affectation dans un ensemble noté C_j
- (4) si $j < m$, $j = j+1$, revenir à l'étape (2), sinon aller à l'étape (5). (lorsque $j = m$, nous obtenons la liste des membres qualifiés pour chaque tâche $\{C_1, C_2, \dots, C_m\}$).
- (5) par la suite, l'algorithme recherche les affectations permettant de maximiser la capacité à travailler en équipe et la capacité relationnelle (fonction objectif dans l'équation 1).

$$\max \sum_{a=1}^m \sum_{b=a+1}^m \sum_{\forall i} \sum_{\forall j} w_{ai,bj} x_{ai} x_{bj} \quad (1)$$

$$\sum_{\forall i} x_{ai} = 1 \quad (a = 1, 2, \dots, n)$$

$$\forall x_{ai}, x_{bj} = 0 \text{ ou } 1$$

La fonction objectif est calculée comme la somme de toutes les combinaisons par paires des valeurs $w_{ai,bj}$. La valeur $w_{ai,bj}$ est utilisée pour calculer la "capacité de travailler en équipe" et la "capacité relationnelle" entre deux acteurs.

Pour faire la combinaison par paire, on compare un à un les acteurs candidats de différents ensembles $\{C_1, C_2, \dots, C_m\}$ (la combinaison par paire est représentée par deux variables a et b), sachant que m est le nombre de tâches dans le module considéré. On prend l'acteur i et l'acteur j appartenant à deux départements différents et on suppose qu'ils doivent travailler ensemble. On peut ainsi calculer la valeur $w_{ai,bj}$ par l'équation (2).

$$w_{ai,bj} = \alpha(T_{ai} + T_{bj}) + (1-\alpha)R_{ai,bj} ; (0 \leq \alpha \leq 1) \quad (2)$$

$w_{ai,bj}$ est une mesure de la capacité de travailler en équipe et de la capacité relationnelle du candidat i dans le groupe a et du candidat j dans le groupe b .

T_{ai} est une mesure de la capacité de travailler en équipe du candidat i dans le groupe a .

T_{bj} est une mesure de la capacité de travailler en équipe du candidat j dans le groupe b .

$R_{ai,bj}$ est une mesure de la capacité relationnelle (profil de personnalité) entre le candidat i dans le groupe a et le candidat j dans le groupe b .

x_{ai} et x_{bj} sont les variables de décision binaire, x_{ai} (x_{bj}) vaut 1 si l'acteur i (j) dans le groupe a (b) est choisi et 0 sinon.

α est une valeur relative d'importance entre deux facteurs (capacité de travailler en équipe et capacité relationnelle), elle est déterminée par des chefs du projet.

En résumé, la méthodologie proposée commence par regrouper les tâches dépendantes qui sont affectés à une équipe, de présélectionner les acteurs candidats à l'affectation dans l'équipe, avant de chercher à maximiser la cohérence interne de l'équipe. Même si ce modèle semble relativement complet, il ne permet pas à des acteurs sous-compétents de se former pour réaliser une tâche et ne prend pas en compte l'évolution des compétences des acteurs. D'autre part, les données manipulées peuvent être difficiles à obtenir ou à estimer.

2.2.4 Maximisation de la satisfaction des clients

[Zakarian et Kusiak, 1999] proposent un modèle pour affecter les membres d'équipe nécessaires à la conception de chaque sous-système (du produit) dans le but de maximiser le niveau de satisfaction du client. Un membre d'équipe possède un métier (domaine d'expertise).

Un projet de conception est constitué de plusieurs sous-systèmes i ($i=1, \dots, m$) et de membres d'équipe j ($j=1, \dots, n$). Le modèle est constitué d'une première variable de décision, x_{ij} , $x_{ij} \in \{0,1\}$. Si la variable x_{ij} vaut 1, le membre d'équipe j appartient à la conception du sous-système i et x_{ij} vaut 0 sinon.

La méthode d'aide à la décision multicritère AHP¹⁰ a été utilisée pour obtenir des valeurs d'importance normalisées (w_{ij}). Ces valeurs d'importance sont évaluées par rapport à des critères correspondant aux exigences du client. La valeur w_{ij} sera utilisée comme un degré de satisfaction du client pour affecter un membre d'équipe j à un sous-système i .

La fonction objectif (1) a pour but de maximiser la satisfaction globale du client en tenant compte de l'affectation des membres d'équipe à des sous-ensembles.

$$\text{Maximiser } \sum_{i=1}^m \sum_{j=1}^n w_{ij} x_{ij} \quad (1)$$

$$x_{ij} = \begin{cases} 1 & \text{si le membre d'équipe } j \text{ appartient à un sous-système } i \\ 0 & \text{sinon} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{si l'équipe pour la conception du sous-système } i \text{ est formée} \\ 0 & \text{sinon} \end{cases}$$

¹⁰ Méthode AHP (Analytic Hierarchy Process) est une méthode d'aide à la décision développée par Thomas L.Saaty

Le nombre de sous-systèmes maximal auquel un membre d'équipe j peut appartenir est représenté par la variable m_j . La contrainte dans l'équation (2) limite ce nombre de sous-systèmes maximal.

$$\sum_{i=1}^m x_{ij} \leq m_j, j = 1, \dots, n \quad (2)$$

La contrainte dans l'équation (3) limite le nombre d'équipes maximum. Le nombre d'équipes maximum à constituer est représenté par la variable de décision binaire y_i , $y_i \in \{0,1\}$. La variable y_i vaut 1 si l'équipe du sous-système i est formée et 0 sinon. La valeur p est le nombre d'équipes maximum du projet. La contrainte (4) a pour but d'assurer qu'il y a au moins un membre de l'équipe j affecté au sous-système i si le sous-système i est formé, et sinon qu'aucun membre n'est affecté à l'équipe. La valeur M est un grand nombre positif (arbitraire).

$$\sum_{i=1}^m y_i \leq p, \forall i = 1, \dots, n, \quad (3)$$

$$\sum_{j=1}^n x_{ij} \leq M y_i, i = 1, \dots, m \quad (4)$$

[Tseng *et al.*, 2004] proposent également une approche permettant de regrouper des tâches de conception avec les liens "exigences du client" (ou satisfaction du client) et d'allouer les concepteurs aux tâches. Une tâche de conception est représentée par une "caractéristique d'ingénierie" qui est considérée également comme des domaines d'expertise que possèdent des ressources humaines. Le regroupement de tâches est réalisé par un algorithme basé sur les matrices d'incidence (algorithme de clustering appelé ROC¹¹). Les auteurs utilisent une approche d'aide à la décision¹² pour déterminer une allocation des ressources humaines maximisant la satisfaction des clients. Ce modèle prend en compte uniquement les connaissances techniques dans l'allocation.

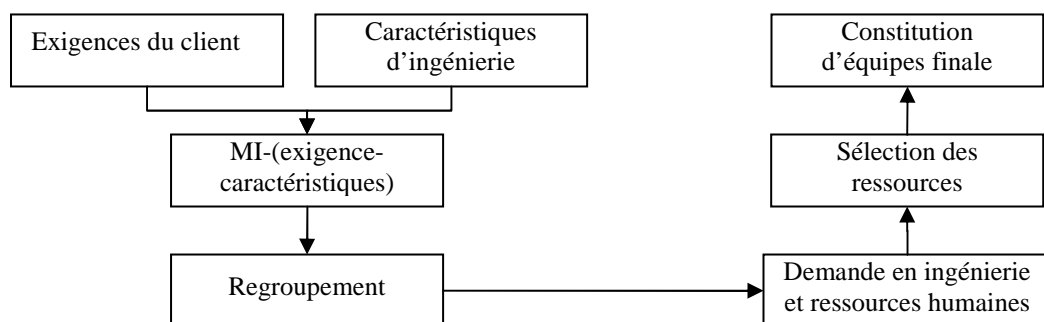


FIG. 2.2 : Démarche de constitution d'équipes [Tseng *et al.*, 2004]

¹¹ ROC (Rank Order Clustering), cet algorithme est proposé par [King, 1980], [King et Nakornchai, 1986].

¹² La méthode utilisée en anglais est "Grey decision-making". La méthode est définie comme un processus traitant des éléments incertains, proposée par J. Deng, 1989.

2.3 Synthèse

Le tableau (TAB. 2.1) présente une synthèse de l'état de l'art sur la constitution d'équipes.

Référence	Domaine d'application	Problème d'optimisation traité		Evaluation des compétences	Compétence	
		Fonction objectif	Méthode de résolution	Indicateur de similarité tâche-acteur	Connaissance technique	Capacité opératoire
[De Korvin, <i>et al.</i> , 2002]	Projet (multi-phases)	--	--	Similarité calculée par utilisation de la logique floue	Modélisation au niveau tâche	--
[Coates <i>et al.</i> , 2007]	Projet	--	Algorithme de génétique	Similarité par le rapport du temps ¹³	Domaine d'expertise	--
[Gronau <i>et al.</i> , 2006]	Conception de logiciel	--	--	Distance basée sur un réseau sémantique	Concept de taxonomie	--
[Barreto <i>et al.</i> , 2008]	Management des ressources humaines	--	Satisfaction des contraintes	--	--	--
[Hadj-Hamou et Caillaud, 2004]	Développement de produit	Maximiser le niveau de compétence et le niveau de coopération	Programmation linéaire en variables binaires	--	Domaine d'expertise	Rôle dans une équipe (Belbin)
[Chen et Lin, 2004], [Chen, 2005]	Développement de produit	Maximiser la capacité de travailler en équipe	Programmation linéaire	--	Fonction/ Département	Capacité de travailler en équipe Profil de personnalité (MBTI)
[Cano et Liern, 2004]	Management des ressources humaines	--	--	Distance de Hamming	Modélisation au niveau tâche	--
[Zakarian et Kusiak, 1999]	Développement de produit	Maximiser la satisfaction des clients	Programmation linéaire en variables binaires	--	Domaine d'expertise (Métier)	--
[Tseng <i>et al.</i> , 2004]	Développement de produit (projet informatique)	--	Approche d'aide à la décision	--	Domaine d'expertise (Métier)	--
[Huang <i>et al.</i> , 2009]	Management du projet	Maximiser le niveau de compétence et minimiser les différences totales de compétences	Goal programming	AHP Logique Floue	Modélisation au niveau tâche	--
[Wi <i>et al.</i> , 2009]	Management du projet R&D	Maximiser le niveau de connaissance technique et le niveau de coopération	Algorithme génétique	Logique floue	Connaissance technique	Savoir-être
[Fan <i>et al.</i> , 2009]	Management du projet R&D	Maximiser le niveau de connaissance technique et le niveau de coopération	Algorithme génétique	AHP	Connaissance technique	Niveau de collaboration entre acteurs

TAB. 2.1 : Synthèse de l'état de l'art sur la constitution d'équipes

Dans ce tableau, nous avons mis en évidence des critères permettant de différencier les

¹³La similarité est représentée par le rapport entre le temps théorique de la tâche et le temps du travail réel de l'individu

travaux existants : domaine d'application, fonction objectif, méthode de résolution, indicateur de similarité de compétence et le type de compétence modélisé (connaissances techniques ou capacités opératoires) [Hlaioittinun *et al.*, 2007a-c]. Quelques points importants peuvent être soulignés :

- Certains travaux traitent la constitution d'équipes sans utiliser de techniques d'optimisation dans l'affectation des tâches aux ressources [De Korvin, *et al.*, 2002], [Gronau *et al.*, 2006], [Canos et Liern, 2004], [Tseng *et al.*, 2004].
- De nombreux travaux formulent des problèmes d'optimisation et les résolvent par diverses techniques d'optimisation [Coates *et al.*, 2007], [Barreto *et al.*, 2008], [Hadj-Hamou et Caillaud, 2004], [Chen et Lin, 2004], [Chen, 2005], [Zakarian et Kusiak, 1999], [Huang *et al.*, 2009], [Wi *et al.*, 2009], [Fan *et al.*, 2009], permettant de trouver des configurations d'équipe qui maximisent les niveaux de compétences ou maximisent le niveau de collaboration entre les membres de l'équipe.
- Différentes méthodes de similarité sont utilisées pour aider à sélectionner les acteurs : distance de Hamming [Canos et Liern, 2004], [Boucher et Burlat, 2003], similarité calculée par la logique floue [De Korvin *et al.*, 2002], [Wi *et al.*, 2009], un réseau sémantique [Gronau *et al.*, 2006], la méthode AHP [Huang *et al.*, 2009] , [Fan *et al.*, 2009].

2.4 Problématique

Nous présentons ci-dessous la problématique qui découle de l'étude bibliographique précédente, plus précisément des limites des travaux existants ou d'extensions envisageables.

2.4.1 Structuration du projet

Concernant la thématique 1 (structuration du projet), de nombreux travaux s'intéressent à la conception de l'architecture des acteurs. Bien que l'importance de l'impact de l'architecture du produit sur l'*architecture des acteurs* ait été soulignée dans la littérature, très peu de travaux utilisent une méthode formelle permettant de propager les contraintes d'une architecture du produit sur une architecture des acteurs. Cela nécessite le développement d'une démarche de constitution d'équipes permettant, aux managers du projet, d'identifier les interactions nécessaires entre les acteurs pour la conception du produit.

2.4.2 Pilotage des compétences

Les points suivants concernent la thématique 2 (pilotage des compétences).

Modélisation des compétences

La synthèse de l'étude bibliographique, présentée dans le tableau (TAB. 2.1), révèle que dans le domaine du management des ressources humaines, la modélisation est globale, au niveau domaine d'expertise ou au niveau tâche. Dans le domaine du management de projet de R&D, la modélisation est souvent plus détaillée, décomposant une compétence en un ensemble de connaissances techniques et éventuellement, en un ensemble de capacités

relationnelles ou opératoires (capacité d'analyse, la capacité d'organisation, la capacité relationnelle et la capacité décisionnelle).

Face aux besoins de réactivité des entreprises pour répondre aux évolutions de l'environnement, la notion de l'évolution des compétences apparaît comme une problématique fondamentale pour la réussite de toute organisation. Malgré cet intérêt, nous constatons un manque d'intégration du concept de l'évolution de compétences dans les modèles et outils existants, fournissant une aide au pilotage des compétences.

Méthode de calcul de similarité

L'affectation des tâches aux ressources requiert souvent une évaluation de similarité entre acteurs et tâches afin de choisir les affectations appropriées. A partir de l'étude bibliographique, on constate qu'il s'agit de différentes méthodes de similarité utilisées pour aider à sélectionner les acteurs pour les tâches (distance de Hamming, similarité calculée par utilisation de la logique floue, etc.).

Nous intéressons aux mesures basées sur le calcul de distance. En étudiant ces mesures de similarité, elles traitent de la même façon les deux cas de sous-compétence et sur-compétence alors que dans la réalité, il existe une dissymétrie entre sur-compétence (performance assurée mais surcoût éventuel ; situation rassurante voire démotivante si c'est trop répétitif) et sous-compétence (performance non assurée mais apprentissage possible au prix d'un surcoût de formation ; situation motivante si challenge acceptable voire stressante si apprentissage trop difficile). En conséquence, un indicateur de compatibilité entre la tâche et l'acteur devrait tenir compte de cette dissymétrie.

Mono projet et multi-projets

Par ailleurs, les travaux identifiés ne prennent pas en compte le partage des ressources entre projets, ils s'intéressent à la constitution d'équipes dans le cas mono-projet. Or, bien souvent, dans une entreprise les projets sont nombreux, s'interpénètrent et mettent en jeu des acteurs de métiers différents. Les ressources étant limitées, il convient de savoir gérer l'utilisation des ressources humaines dans un ensemble de projets [Hendriks *et al.*, 1999]. L'optimisation de l'utilisation des ressources humaines doit être réalisée sur plusieurs projets pour atteindre les objectifs de performance. Il devient impératif de mettre en place une gestion spécifique et claire de tous ces projets, en particulier, le développement d'une démarche d'affectation sur de multiples projets.

A partir de cette étude bibliographique, on constate très aisément qu'il n'y a aucun travail traitant les deux problématiques en même temps, à savoir la problématique de conception de l'architecture des acteurs et la problématique du pilotage des compétences. Les deux thèmes de recherche ont apporté des solutions intéressantes pour la constitution d'équipes mais ils sont traités de manière isolée. Il nous paraît intéressant de proposer une démarche globale tenant compte de toutes ces limites afin de réduire la performance du projet de conception et de promouvoir l'innovation du produit.

2.4.3 Second positionnement

Nous positionnons notre contribution de la constitution d'équipes par rapport aux synthèses présentées précédemment.

Notre contribution à la constitution d'équipes est une articulation entre deux thèmes de recherche : la structuration du projet et le pilotage des compétences. Nous avons donc pour ambition de traiter ces deux aspects de façon intégrée et cohérente.

- En ce qui concerne le pilotage des compétences, nous nous sommes intéressés aux travaux d'affectation des tâches à des ressources, dans le cas multi-projets et multi-périodes, en fonction de l'évolution des connaissances (techniques). Un indicateur de similarité sera proposé pour aider au choix d'affectation. Cette contribution est présentée au chapitre 4.
- En ce qui concerne la structuration du projet, nous nous sommes intéressés spécialement aux travaux traitant le couplage entre l'architecture du produit et l'architecture des acteurs dans le but de concevoir une architecture des acteurs du projet. Cette contribution est présentée au chapitre 5. Une démarche intégrant la structuration du projet et l'affectation des tâches est aussi proposée dans ce chapitre.

L'affectation des tâches aux ressources reste un sujet d'actualité. Le chapitre 3 se consacrera à un état de l'art complémentaire sur les travaux d'affectation des tâches aux ressources afin d'approfondir les types de problèmes traités.

3 Etat de l'art sur le problème d'affectation

Les travaux de constitution d'équipes présentés dans le chapitre précédent ne considèrent pas l'évolution des compétences. Cette évolution est néanmoins prise en compte dans d'autres types de travaux, à savoir les travaux d'affectation des tâches (ou d'allocation des ressources).

Ce chapitre a pour but de présenter un état de l'art sur les problèmes d'affectation permettant de clarifier la façon dont la notion de compétence, et plus particulièrement leur caractère évolutif, est modélisée dans l'affectation des tâches à des ressources.

Nous avons défini deux critères pour classer ces travaux : le type d'affectation (mono ou multi-périodes) et le type de modélisation des compétences (statique ou dynamique). En combinant ces deux critères, nous obtenons quatre classes de problèmes. Un succinct état de l'art est présenté pour chacune de ces classes. Nous proposons ensuite une synthèse permettant de tirer des caractéristiques et des points importants sur ces travaux.

3.1 Classification des travaux

Nous avons constaté que les travaux dans le domaine de la conception présentés dans le chapitre précédent proposent des modélisations de compétence plus élaborées mais ils ne considèrent pas l'évolution des compétences. Cette dernière est beaucoup étudiée dans le domaine de la production car il s'agit de problèmes modélisés sur plusieurs périodes permettant d'affecter des tâches ou des ressources et éventuellement de prendre en compte l'évolution des compétences des ressources. Ainsi, la partie la plus importante de l'étude bibliographique présentée ici tourne autour des travaux d'allocation des ressources dans le domaine de production, en particulier l'affectation tournante ou l'affectation inter-département et autour de travaux dans le management de projet, en particulier la gestion de projets avec prise en compte de compétences multiples (MSPSP). Même si nous nous n'intéressons pas au problème d'ordonnancement, certains travaux seront présentés pour permettre de mieux comprendre le problème d'affectation.

L'étude bibliographique proposé dans ce chapitre a pour but d'observer la formulation des variables de compétence permettant d'affecter une tâche à une personne de façon appropriée, d'identifier le caractère évolutif de ces variables (dépendantes / indépendantes du temps), et d'étudier comment elles sont utilisées pour piloter les compétences des acteurs.

Pour ce faire, nous avons recensé et classé les travaux selon deux critères [Hlaoittinun *et al.*, 2009] :

- Le premier critère permet de distinguer le type de l'affectation des tâches (sur une seule période ou sur plusieurs périodes). L'affectation mono-période est une situation où on affecte des tâches à des ressources sur une seule période. On s'intéresse alors à deux instants de l'affectation (le début et la fin). L'affectation multi-périodes est une situation où on affecte des tâches à des ressources sur plusieurs périodes.
- Le deuxième critère permet de distinguer le type de modélisation des compétences (statique ou dynamique). La modélisation statique des compétences est le cas où l'on considère que le niveau de compétence des ressources n'évolue pas. La modélisation dynamique des compétences représente le fait que le niveau des compétences évolue dans le temps en fonction des décisions d'affectation. Dans le cas de mono-période, nous suivons l'évolution des compétences sur deux instants (le début et la fin). Dans le cas multi-périodes, nous pouvons constater l'évolution des compétences sur plusieurs périodes.

En combinant ces deux critères, nous obtenons quatre classes de problème. Nous présentons ci-dessous un rapide panorama de ces quatre classes.

- Problème d'affectation **mono-période** avec une **modélisation statique des compétences** (sans évolution) [Caron *et al.*, 1999], [Campbell et Diaby, 2002], [Eiselt et Marianov, 2008], [Peters et Zelewski, 2007], [Tsai *et al.*, 2003]. Ces travaux s'intéressent à l'allocation des ressources sur une seule période selon leurs compétences. Les variables de compétence considérées sont binaires (qualification) ou réelles (entre 0 et 1). Les travaux présentés peuvent répondre à différents objectifs mais les modèles présentent au moins une variable modélisant les compétences. Cependant, ces travaux ne s'intéressent pas à l'évolution potentielle de ces compétences. Comme il y a une seule période, il n'existe pas d'ordonnancement des tâches. Nous détaillerons en 3.2 les travaux suivants :

- Maximisation de l'utilisation des ressources en tenant compte de la qualification [Caron *et al.*, 1999]
 - Maximisation de l'utilisation des compétences des ressources dans l'affectation inter-département [Campbell et Diaby, 2002]
 - Minimisation de l'insatisfaction au travail due à l'absence de compétence [Eiselt et Marianov, 2008], [Peters et Zelewski, 2007]
- Problème d'affectation **multi-périodes** avec une **modélisation statique des compétences** (sans évolution) [Miller et Franz, 1996] [Bellenguez-Morineau, 2006], [Corominas *et al.*, 2006], [Cheng *et al.*, 2009]. Les travaux dans ce domaine cherchent à allouer des ressources humaines sur plusieurs périodes selon leurs compétences. Toutefois, les compétences dans ces travaux sont considérées de manière statique. Certains travaux cherchent simultanément à affecter des tâches sur plusieurs périodes avec des contraintes de précédence (ordonnancement des tâches). Nous présenterons en 3.3 les travaux suivants :
- Minimisation des coûts d'affectation sur de multiples périodes [Miller et Franz, 1996]
 - Minimisation de la durée totale du projet (C_{max}) [Bellenguez-Morineau, 2006]
 - Minimisation de l'écart des références [Corominas *et al.*, 2006]
- Problème d'affectation **mono-période** avec une **modélisation dynamique des compétences** (avec évolution) [Sayin et Karabati, 2007]. Ce travail de recherche s'est intéressé à allouer des ressources sur une seule période, mais les compétences considérées ici sont évolutives. Nous pouvons observer le changement des niveaux de compétence entre deux instants (le début et la fin de cette seule période). Nous résumerons ce travail en 3.4 :
- Maximisation de l'augmentation des compétences [Sayin et Karabati, 2007]
- Problème d'affectation **multi-périodes** avec une **modélisation dynamique des compétences** (avec évolution) [Gutjahr *et al.* 2008], [Fowler *et al.*, 2008]. Ces travaux cherchent à allouer des ressources humaines sur plusieurs périodes selon leurs compétences. Les compétences modélisées sont évolutives (les niveaux de compétence changent d'une période à l'autre). Certains travaux considèrent également le problème d'ordonnancement des tâches. Nous présenterons en 3.5 les travaux suivants :
- Maximisation de la compétence avec effets d'apprentissage [Gutjahr *et al.* 2008]
 - Minimisation des coûts liés à la formation des ressources [Fowler *et al.*, 2008]

Les classes de problème définies ci-dessus sont détaillées dans les parties qui suivent.

3.2 Affectation mono-période avec une modélisation statique de compétences

Le but de la présentation de ces travaux est d'introduire des modèles d'affectation de base afin d'approfondir d'autres modèles d'affectation plus complexes. Les travaux présentés dans cette catégorie restent des modèles d'affectation simples (mono-période) présentant une formulation des variables de compétence de manière statique.

3.2.1 Maximisation de l'utilisation des ressources en tenant compte de la qualification

[Caron *et al.*, 1999] s'intéressent à une affectation des tâches à des ressources particulièrement en prenant en compte la qualification des ressources. Leur modèle est présenté comme un modèle d'extension du modèle classique de l'affectation. La formulation du problème peut être présentée comme suit.

$$\begin{aligned} & \text{Maximiser} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ & \text{sous les contraintes} \\ & \quad \sum_{i=1}^m q_{ij} x_{ij} \leq 1, j = 1, \dots, n \\ & \quad \sum_{j=1}^n q_{ij} x_{ij} \leq 1, i = 1, \dots, m \\ & \quad x_{ij} = 0 \text{ ou } 1 \end{aligned}$$

Cette formulation correspond à un problème de programmation linéaire en nombres binaires. Il s'agit d'affecter m tâches à n acteurs. Une personne peut être affectée seulement si elle est qualifiée pour cette tâche. Le modèle cherche à maximiser l'utilisation des ressources, ou à minimiser le temps non productif des ressources. Cela dépend de la définition du coefficient c_{ij} qui peut être un coefficient de coût ou un coefficient de bénéfice. La variable de décision x_{ij} est une variable binaire qui vaut 0 ou 1. La valeur binaire q_{ij} représente le niveau de qualification de la personne i pour réaliser la tâche j , $q_{ij}=1$ si la personne est qualifiée, et 0 sinon.

Pour approfondir ce problème d'affectation qui est un problème classique de recherche opérationnelle, une classification et les travaux associés à chaque classe ont été présentés dans le travail de [Pentico, 2007].

3.2.2 Maximisation de l'utilisation des compétences des ressources dans l'affectation inter-département

Certains travaux [Brusco et Johns, 1998] [Campbell, 1999] [Campbell et Diaby, 2002] s'intéressent au problème d'affectation inter-département (cross utilization ou cross training). Ce problème est considéré comme une stratégie pour l'affectation des ressources dans les services hospitaliers où le phénomène d'absence du personnel est crucial. La haute qualité du service et la contrainte budgétaire s'ajoutent comme contraintes à la prise de décision. Chaque ressource se caractérise par sa culture, ses compétences, sa spécialité et son niveau d'expérience. Selon [Hopp et Van Oyen, 2004], l'affectation inter-département des ressources humaines a plusieurs bénéfices importants en termes de coût, délai, qualité et variété du produit.

- Coût : augmenter la productivité par une meilleure utilisation des ressources,
- Délai : augmenter la flexibilité, la vitesse de l'exécution des tâches,
- Qualité : augmenter l'adéquation entre tâches et ressources,

- Variété du produit : élargir la gamme des tâches à réaliser.

Le modèle de [Campbell, 1999] a été développé avec une série d'expérimentations pour étudier l'utilisation inter-département des ressources. Une fonction pour estimer l'augmentation du niveau de connaissance a été définie. Le but est de maximiser l'utilisation des compétences des ressources. Chaque département d ($d=1, \dots, D$) demande un certain nombre de personnes (non déterminé à l'avance). La personne i ($i=1, \dots, I$) possède différents types de compétence correspondant à des départements disciplinaires d . La variable c_{id} correspond au niveau de connaissance de la personne i dans le département d et varie entre 0 et 1. Si c_{id} est égale à 0.8, cela signifie que la personne i possède une performance de 80% par rapport à la personne qualifiée.

La variable de décision binaire x_{id} vaut 1 si l'acteur i est alloué au département d , elle vaut 0 sinon. La somme des niveaux de compétence des acteurs alloués au département d est représentée par w_d .

La formulation mathématique du problème s'écrit de la manière suivante.

$$\text{Maximiser : } U = \sum_{d=1}^D u_d(w_d) \quad (1)$$

sous les contraintes de:

$$w_d = \sum_{i=1}^I c_{id} x_{id}, \quad \forall d = 1, \dots, D, \quad (2)$$

$$\sum_{d \in \{S_i\}} x_{id} = 1, \quad \forall i = 1, \dots, I, \quad (3)$$

$$x_{id} = 0 \text{ ou } 1, \quad \forall i = 1, \dots, I; \forall d = 1, \dots, D \quad (4)$$

La fonction objectif (1) maximise la somme de l'utilisation des compétences des ressources. Cette fonction est une fonction concave (quadratique). La contrainte (2) impose que la somme des niveaux de compétence des acteurs alloués au département d vaut w_d . La contrainte (3) indique que chaque acteur doit être affecté à un département. Notons que l'ensemble $\{S_i\}$ est l'ensemble des départements pour lesquels la valeur c_{id} est supérieure à 0. La contrainte (4) assure que les variables de décision sont binaires et de valeur non négative.

Les auteurs ont choisi de présenter le problème d'une manière simple. En général, le problème donné par (1)-(4) est difficile à résoudre d'une façon optimale. Cependant, ils précisent que, si les valeurs c_{id} sont binaires, un algorithme de type glouton peut être utilisé pour obtenir une solution optimale.

Pour le cas des c_{id} fractionnels, le problème est une variante du problème d'affectation généralisé¹⁴. La méthode traditionnelle pour résoudre ce problème est d'utiliser la relaxation lagrangienne. Dans ce travail, l'auteur propose une heuristique pour résoudre le problème donnée par (1)-(4). Cette heuristique est basée sur une reformulation de ce problème en problème d'affectation linéaire. Le problème a été résolu par la programmation linéaire en nombre mixte (MIP¹⁵). Dans [Campbell et Diaby, 2002], les auteurs ont aussi comparé cette

¹⁴Le terme en anglais est Generalized assignment problem

¹⁵ MIP=Mix-integer linear programming

heuristique avec deux heuristiques classiques (algorithme glouton et heuristique lagrangienne). Les résultats montrent que l'heuristique proposée est supérieure à l'algorithme glouton et à l'heuristique lagrangienne.

L'intérêt de ce modèle est qu'il présente une formulation du problème d'affectation visant à maximiser l'utilisation des compétences. Cependant, il apparaît difficile de résoudre ce problème car il devient non linéaire. Cela nécessite des méthodes d'optimisation combinatoires ou parfois des reformulations du problème pour qu'il puisse être résolu par une méthode exacte.

3.2.3 Minimisation de l'insatisfaction au travail

Certains travaux [Eiselt et Marianov, 2008], [Peters et Zelewski, 2007] visent à minimiser l'insatisfaction du personnel. Une mauvaise affectation des tâches à des ressources peut conduire à une mauvaise performance au niveau du travail pour plusieurs raisons. Par exemple, la personne affectée n'est pas qualifiée pour la tâche demandée et elle doit donc faire face à un stress supplémentaire. La personne, qui a été affectée malgré ses réticences, peut éventuellement se démotiver.

[Eiselt et Marianov, 2008] cherchent à résoudre le problème de satisfaction au travail liée à l'absence de capacité à réaliser une tâche dans le domaine de production. Une fonction objectifs liée à la notion de compétence est présentée. Elle a pour but de minimiser la distance entre employés et tâches afin de permettre, à l'entreprise, d'améliorer la satisfaction des employés. Le problème est résolu par programmation linéaire en nombres entiers à objectifs multiples¹⁶. Une fonction permettant de mesurer l'écart moyen entre tâche et acteur a été proposée. Néanmoins, cette fonction ne fait pas la différence entre des écarts négatifs et positifs du niveau de compétence.

[Peters et Zelewski, 2007] cherchent à développer un modèle pour l'affectation des employés aux lieux de travail. Les auteurs supposent qu'une mauvaise affectation à un lieu de travail peut engendrer une inefficacité de l'équipe ou une baisse des performances. Dans un premier temps, les évaluations des choix du lieu de travail selon certains critères ont été réalisées par la méthode AHP (Analytic Hierarchy Process). Les critères considérés décrivent les conditions générales associées à un lieu de travail (les heures travaillées par jour, le nombre de jours travaillés dans une semaine, la sûreté opérationnelle etc.). Les ressources sont allouées aux lieux de travail selon leurs compétences et leurs préférences pour s'assurer de la motivation des employés et de la performance des tâches affectées. La méthode d'optimisation par goal programming¹⁷ est utilisée pour résoudre le problème d'affectation.

Le modèle est constitué des variables suivantes :

a_{in} : niveau de compétence de type i acquis par l'acteur n

g_{ij} : niveau de compétence de type i requis par le lieu (ou poste) de travail¹⁸ j

I : nombre de compétences

J : nombre de lieux (ou postes) de travail

¹⁶ multiple-objective integer linear optimization problem

¹⁷ L'essence du goal programming réside dans le concept de satisfaction d'objectifs : on se fixe un but à atteindre pour chacun des critères. Il s'agit de minimiser la somme des valeurs absolues des écarts par rapport aux buts.

¹⁸ Le terme anglo-saxon est difficile à traduire ici : "workplace".

N : nombre d'acteurs

w_{ij} : poids d'importance de la compétence i dans un lieu de travail j

Les auteurs distinguent deux variables de compétence :

d_{ij}^- : variable d'écart négatif indiquant un manque du niveau de compétence de type i au lieu de travail j . C'est la variable de décision du modèle.

d_{ij}^+ : variable d'écart positif indiquant un surplus du niveau de compétence de type i au lieu de travail j .

L'objectif de ce modèle est de minimiser le manque du niveau de compétence (écarts négatifs) des acteurs alloués dans tous les lieux de travail.

$$\text{Minimiser } Z = \sum_{i=1}^I \sum_{j=1}^J w_{ij} d_{ij}^-$$

$$\sum_{n=1}^N a_{in} x_{jn} + d_{ij}^- - d_{ij}^+ = g_{ij}, \forall i = 1, \dots, I, \forall j = 1, \dots, J$$

$$d_{ij}^+, d_{ij}^- \geq 0 \quad \forall i = 1, \dots, I, \forall j = 1, \dots, J$$

$$x_{jn} \in \{0, 1\}, \quad \forall j = 1, \dots, J, \forall n = 1, \dots, N$$

$$\sum_{i=1}^I w_{ij} = 1, \quad \forall j = 1, \dots, J,$$

Même si le modèle fait la distinction entre l'écart négatif et l'écart positif, ces deux variables (d_{ij}^-, d_{ij}^+) sont des variables mesurant les écarts du point de vue "lieu de travail", et non du point de vue "tâche". La compatibilité entre des acteurs alloués et des tâches (indicateur de similarité) n'a pas été évaluée. Le modèle ne permet donc pas de distinguer les acteurs compétents des acteurs incompetents.

3.2.4 Conclusion

Parmi les travaux présentés dans cette partie, nous avons retenu les points intéressants suivants. Les choix d'affectation peuvent être proposés selon différents critères : qualité (maximiser la similarité entre tâches et ressources), efficacité (minimiser le coût d'affectation), satisfaction des salariés, etc. La plupart des travaux caractérisent des acteurs avec de multiples compétences et avec des niveaux de compétence modélisés par des variables réelles (alors que les qualifications sont souvent modélisées par des variables binaires). On constate aussi que certains travaux font la distinction entre l'écart positif et l'écart négatif du niveau de compétence dans l'affectation pour mieux raffiner la configuration des solutions d'affectation. Cependant, les modèles proposés ne prennent pas en compte la dimension évolutive de la modélisation des compétences.

3.3 Affectation multi-périodes avec une modélisation statique des compétences

Des travaux de recherche se sont intéressés à l'affectation multi-périodes sans prendre en compte la dynamique de la compétence. La modélisation des tâches et l'ordonnancement des tâches peuvent être considérées.

3.3.1 Minimisation des coûts d'affectation sur de multiples périodes

[Miller et Franz, 1996] proposent une formulation du problème multi-périodes considérant l'affectation m tâches à n acteurs pendant K périodes. Ce problème nous permet de constater l'apparition de l'indice k représentant la période et permettant d'affecter des tâches aux ressources sur plusieurs périodes.

La fonction objectif proposée est la suivante :

$$\text{Minimiser (ou Maximiser)} = \sum_{k=1}^K \sum_{j=1}^n \sum_{i=1}^m c_{ijk} x_{ijk}$$

$$\sum_{i=1}^m x_{ijk} = 1, \quad \forall j = 1, \dots, n, \quad \forall k = 1, \dots, K \quad (1)$$

$$\sum_{j=1}^n x_{ijk} \begin{matrix} \leq \\ \geq \end{matrix} r_{ik}, \quad \forall i = 1, \dots, m, \quad \forall k = 1, \dots, K \quad (2)$$

$$\sum_{k=1}^K x_{ijk} = s_{ij}, \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, n \quad (3)$$

La fonction objectif de ce problème de programmation linéaire a pour but de maximiser les bénéfices ou de minimiser les coûts de l'affectation, où le coefficient d'affectation est défini par la valeur c_{ijk} . Les variables de décision binaires x_{ijk} sont dépendantes de la période. Si l'acteur i est affecté à la tâche j durant la période k , alors x_{ijk} vaut 1, 0 sinon. La première contrainte (1), liée à la charge du personnel, assure que chaque acteur est affecté à une tâche pendant la période considérée. La deuxième contrainte (2), liée à la tâche, assure que chaque tâche i est réalisée par un certain nombre d'acteurs demandés pendant la période k considérée. La variable r_{ik} est le nombre d'acteurs nécessaire pour assurer la réalisation de la tâche i durant la période k . La troisième contrainte (3), liée également à la tâche, assure que le nombre de périodes que l'acteur j travaille sur la tâche i ne dépasse pas le nombre de périodes demandé. La variable s_{ij} représente le nombre de périodes demandé pour réaliser la tâche i par l'acteur j .

Même si ce travail permet d'affecter des tâches aux ressources sur plusieurs périodes (K), il ne modélise pas les compétences. La modélisation dynamique des compétences n'a donc pas été étudiée.

3.3.2 Minimisation de la durée totale du projet

Les périodes d'affectation des tâches étant multiples, la deuxième catégorie de travaux traite le problème d'allocation des ressources et le problème d'ordonnancement des tâches.

L'ordonnancement des activités a été reconnu comme un élément clé dans la gestion de projet. On parle encore de planification opérationnelle [Giard, 1991]. La définition de l'ordonnancement est une programmation des tâches et des ressources nécessaires à leur exécution. La formulation du problème d'ordonnancement est différente selon les contraintes utilisées. Les contraintes classiques de l'ordonnancement de projet peuvent être liées au temps (la date d'achèvement du projet, les différents retards) ou / et liées aux ressources (la quantité totale de ressources nécessaires pour réaliser un ensemble de tâches, la capacité de chaque ressource). Simultanément à l'ordonnancement des tâches, il faut considérer la disponibilité des ressources humaines, afin d'éviter une situation où l'ordonnancement des tâches est optimal, mais avec des ressources inadéquates [Coates *et al.*, 2004].

[Grabot, 2008] recense les contraintes principales découlant d'une gestion des compétences pour l'ordonnancement. L'auteur constate qu'une meilleure formalisation des compétences des individus est une condition à une modélisation plus précise des contraintes permettant la prise en compte des personnes dans l'atelier.

Tout d'abord, nous présentons le problème de gestion de projet à contraintes de ressources (Resource Constrained Project Scheduling Problem-RCPSP). Ce problème est largement étudié dans la littérature [Lopez et Roubellat, 2001], [Baptiste *et al.*, 2005], [Chen et Askin, 2009]. Ce problème consiste en l'ordonnancement de tâches, également appelées activités ou opérations, sur des ressources renouvelables en quantité limitée. Ce problème ne prend pas en compte la compétence. Les tâches sont liées entre elles par des relations de précédence, ce qui signifie que la tâche i ne peut pas commencer tant que la tâche j n'est pas finie. Il s'agit de trouver une solution qui minimise la date de fin du projet, en respectant à la fois les contraintes de précédence et les contraintes de ressources, ce qui signifie qu'à tout instant, la somme des besoins des tâches en cours d'exécution ne doit pas dépasser la capacité de la ressource. Une extension possible du problème est de le formuler en *multi-projets*. Il s'agit ici de considérer un environnement constitué de plusieurs projets devant s'effectuer au moins partiellement en parallèle. Le problème revient à affecter tout ou partie des ressources nécessaires à chaque projet, puis à minimiser le coût d'exécution des projets.

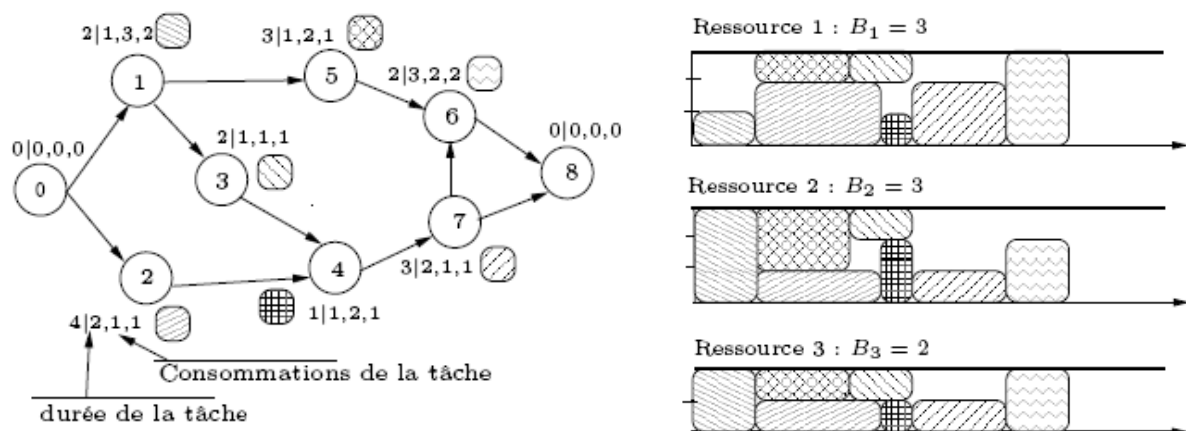


FIG. 3.1 : Exemple d'un graphe de précédence et une solution optimale de RCPSP (GoTHA)

Certains travaux se basant sur le problème RCPSp tiennent compte de la compétence. [Bellenguez-Morineau, 2006], [Drezet et Baillaut, 2008] se sont intéressés aux problèmes d'ordonnement de projets avec des ressources multi-compétences (En Anglais, MSPSP, Multi-skill Project Scheduling Problem). Le modèle de Bellenguez est inspiré par le problème de gestion de projet à contrainte de ressources classique (RCPSp), ce travail se situe dans la phase d'ordonnement. Chacune des personnes affectées maîtrise une ou plusieurs compétences parmi celles nécessaires aux tâches du projet. Le but de ce modèle est de *minimiser la durée totale ou la date d'achèvement global du projet (C_{max})*. L'évaluation obtenue à l'aide de ce critère sert en particulier dans les phases d'avant-vente, c'est-à-dire au cours des négociations avec le client.

Modélisation des compétences dans le problème MSPSP :

- Le projet se compose d'un ensemble d'activités $A_i, i \in \{0, \dots, n\}$, liées par des contraintes de précédence de type fin-début.
- Les ressources considérées pour l'exécution de ce projet sont des personnes $P_m, m \in \{0, \dots, M\}$.
- Chaque personne P_m peut posséder une ou plusieurs compétences $S_k, k \in \{0, \dots, K\}$, la valeur $MS_{m,k}$ est une variable binaire de compétence (ou qualification) de la personne m pour la tâche i . Si $MS_{m,k} = 1$, la personne P_m maîtrise la compétence S_k , et 0 sinon.
- Pour chaque activité A_i et chaque compétence S_k , il existe une donnée $b_{i,k}$ qui est égale au nombre de personnes qui devront exercer la compétence S_k lors de l'exécution de l'activité A_i . La valeur $b_{i,k}$ est nulle si la compétence S_k n'est pas nécessaire pour l'activité A_i .

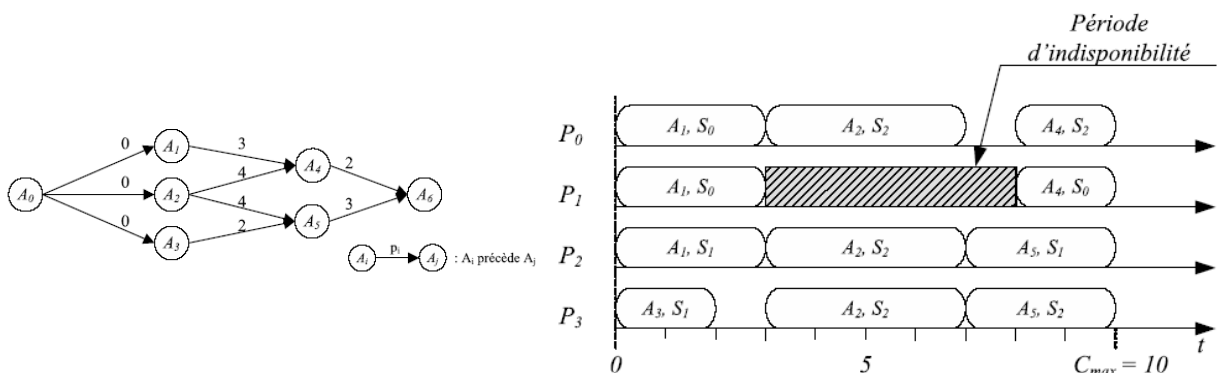


FIG. 3.2 : Exemple d'un graphe de précédence et une solution optimale de l'ordonnement [Bellenguez-Morineau, 2006]

Ce travail traite deux types de problèmes en même temps : le problème d'ordonnement des tâches et le problème d'affectation des tâches à des ressources. Pour ce qui nous intéresse les affectations réalisées prennent en compte les compétences acquises par les acteurs. Cependant le problème est plutôt orienté sur l'ordonnement (but : minimiser la durée totale

du projet). Même si le problème d'affectation est formulé de manière multi-périodes, l'utilisation des compétences des ressources et l'évolution de leurs compétences n'ont pas été prises en considération. Selon [Grabot, 2008], dans la plupart des travaux de recherche comme dans les logiciels de simulation et d'ordonnancement, on fait l'hypothèse que les compétences/qualifications sont statiques. En effet, les variables de compétence sont binaires (qualification ou non) par analogie avec une ressource matérielle qui peut, ou non, réaliser une opération. Cette modélisation n'est pas suffisante. Une "ressource humaine" a la capacité d'apprendre. Une compétence peut être évaluée sur une échelle de valeur, autre que 0 et 1. Un acteur peut apprendre pour devenir compétent.

3.3.3 Minimisation de l'écart des références

[Corominas *et al.*, 2006] proposent un modèle d'affectation multi-périodes dans un centre de fabrication, qui a pour but d'alterner des tâches pour les employés. Le modèle cherche à affecter des tâches aux ressources en respectant la valeur de référence du temps de travail pour chaque tâche (minimiser l'écart par rapport à la valeur de référence du temps de travail).

On suppose que le nombre de périodes consécutives d'affectation de chaque personne pour chaque type de tâche appartient à un intervalle entre un nombre de périodes consécutives minimum et un nombre maximum. Les nombres de périodes minimum et maximum sont définis pour que le personnel ne travaille pas toujours sur la même tâche. Un des avantages de l'alternance des tâches est qu'elle permet aux employés de se former sur différentes tâches.

Deux valeurs sont ajoutées au calcul du coût selon les conditions de l'affectation : valeur positive et valeur négative.

- une valeur positive (bonus) sera ajoutée dans la fonction de coût pour promouvoir une affectation afin que l'acteur i puisse atteindre le nombre de périodes consécutives minimum.
- une valeur négative (pénalité) sera ajoutée dans la fonction de coût pour pénaliser l'affectation où le nombre de périodes effectuées a déjà dépassé le nombre de périodes consécutives maximum.

Ce travail traite le problème d'alternance des tâches dans le but d'inciter les acteurs à réaliser différentes tâches et à pénaliser le coût du projet si ces affectations sont faites de manière trop répétitive. Le modèle d'affectation prend bien en compte les compétences mais les variables de compétence restent binaires. Le modèle vise à minimiser plutôt la déviation par rapport à la valeur de référence définissant un nombre de répétitions dans la réalisation des tâches. On peut constater que ce travail ne traite pas l'évolution des compétences même si le problème est formulé en multi-périodes.

3.3.4 Conclusion

Les travaux présentés dans cette partie concernent essentiellement l'affectation multi-périodes des tâches aux ressources sans évolution des compétences. Certains travaux présentés traitent aussi l'ordonnancement des tâches.

Dans tous les cas, les indices de la période modélisés dans cette partie servent uniquement pour l'affectation des tâches mais pas pour la modélisation dynamique des compétences. Autrement dit, à la fin de chaque affectation, les ressources humaines possèdent le même niveau de compétence qu'au début. Le niveau de compétence est souvent représenté

pauvrement par une variable binaire, par une analogie réductrice avec la modélisation des ressources matérielles.

3.4 Affectation mono-période avec une modélisation dynamique des compétences

Cette catégorie s'intéresse aux travaux qui ne nécessitent pas de planification des tâches (mono-période) et considère l'évolution des compétences sur une période (le niveau de compétence évolue entre deux instants, le début et la fin). Nous avons trouvé un article [Sayin et Karabati, 2007] faisant partie de cette catégorie. Nous présentons ce travail ci-dessous.

3.4.1 Maximisation de l'augmentation des compétences

En prenant en compte l'affectation inter-départements des tâches à des ressources et l'évolution des compétences par la courbe individuelle, [Sayin et Karabati, 2007] proposent un modèle d'affectation en deux étapes où deux fonctions objectifs sont considérées consécutivement. La première optimisation, formulée de façon similaire au modèle d'allocation des ressources aux départements de [Campbell, 1999], vise à maximiser l'utilisation des compétences des ressources (U^*). La seconde vise à maximiser l'augmentation totale des compétences en prenant en compte l'évolution de compétences par une courbe individuelle d'apprentissage.

Le modèle est constitué de plusieurs départements d ($d=1, \dots, D$), chaque département d demande des ressources en quantité R_d et de plusieurs acteurs w ($w=1, \dots, W$), chaque acteur w possède un niveau de compétence (relativement au département d), c_{wd} .

Notations utilisées :

- W est le nombre d'acteurs, ($w=1, \dots, W$).
- D est le nombre de départements, ($d=1, \dots, D$).
- R_d est la quantité des ressources pour le département d , ($d=1, \dots, D$).
- c_{wd} est le niveau de maîtrise de la compétence (relativement au département d) acquis par l'acteur w .
- c'_{wd} représente l'augmentation du niveau de maîtrise de la compétence d concernant l'acteur w (s'il est alloué au département d), cette variable est calculée grâce à la courbe d'apprentissage.
- u_d représente le coefficient d'utilité de chaque département d , ($d=1, \dots, D$).
- β représente la déviation acceptable de la valeur d'utilité optimale.
- X_{wd} est la variable de décision binaire. Elle vaut 1 si l'acteur w est affecté au département d , elle vaut 0 sinon.
- S_d représente le manque de ressources (rupture) dans le département d une fois l'affectation réalisée.

- $U_d(S_d)$ représente la fonction d'utilisation des ressources concernant le département d une fois l'affectation réalisée, $d=1, \dots, D$, $U_d(S_d) = -u_d S_d^2$ si $S_d < 0$ et $U_d(S_d) = 0$ sinon.
- U est le niveau total d'utilisation des compétences des ressources.
- Z est le niveau total d'augmentation du niveau de maîtrise dans la compétence.

La première optimisation (modèle M1) s'écrit comme suit :

$$\text{Maximiser } U^* = \sum_{d=1}^D U_d(S_d) \quad (1)$$

sous les contraintes de:

$$\sum_{d=1}^D X_{wd} = 1, \quad w = 1, \dots, W,$$

$$S_d = \sum_{w=1}^W c_{wd} X_{wd} - R_d, \quad d = 1, \dots, D,$$

$$X_{wd} \in \{0, 1\}, \quad w = 1, \dots, W; d = 1, \dots, D.$$

La deuxième optimisation (modèle M2) s'écrit comme suit :

$$\text{Maximiser } Z = \sum_{d=1}^D \sum_{w=1}^W \frac{c'_{wd}}{c_{wd}} X_{wd} \quad (2)$$

sous les contraintes de :

$$\sum_{d=1}^D X_{wd} = 1, \quad w = 1, \dots, W,$$

$$S_d = \sum_{w=1}^W c_{wd} X_{wd} - R_d, \quad d = 1, \dots, D,$$

$$U = \sum_{d=1}^D U_d(S_d),$$

$$U \geq (1 + \beta)U^*,$$

$$X_{wd} \in \{0, 1\}, \quad w = 1, \dots, W; d = 1, \dots, D.$$

La première étape vise à optimiser l'utilisation des compétences des ressources dans les départements. La valeur optimale d'utilisation des ressources (U^*) est ensuite intégrée dans la deuxième étape de l'optimisation. Cette deuxième étape vise à maximiser les augmentations du niveau de compétence en ne descendant pas en dessous d'un niveau acceptable d'utilisation des compétences des ressources obtenu par la première optimisation. Ce niveau acceptable est exprimé par $(1 + \beta)U^*$ où la constante β représente la déviation acceptable de ce niveau d'utilisation.

La fonction d'utilisation des ressources U est une fonction quadratique en raison de l'expression choisie pour $U_d(S_d)$. Les modèles M1 et M2 sont non linéaires, respectivement car la fonction objectif est quadratique dans M1 et car les contraintes sont quadratiques dans M2. Les auteurs ont utilisé une technique de linéarisation¹⁹ par morceaux pour approximer la

¹⁹ Piecewise linear function

fonction quadratique. Le problème est alors résolu par un solveur de programmation linéaire mixte.

Concernant l'augmentation du niveau de compétence, les auteurs supposent que la compétence va augmenter et prendre la valeur c'_{wd} si l'acteur w a été affecté au département d (implicitement, à une tâche associée à l'expertise de ce département). Les augmentations du niveau de compétence (de c_{wd} à c'_{wd}) sont estimées par une courbe d'apprentissage hyperbolique, proposée par [Nembhard et Uzumeri, 2000]. Ce modèle s'écrit comme suit.

$$c_{wd} = k_{wd} \frac{x_{wd} + p_{wd}}{x_{wd} + p_{wd} + r_{wd}} \quad (3)$$

La valeur c_{wd} est le niveau de compétence de l'acteur w dans le département d . La valeur x_{wd} est une quantité cumulative correspondant au temps que passera l'acteur w dans le département d si son affectation a eu lieu. La valeur p_{wd} est la mesure du temps déjà passé dans ce département par l'acteur w . Le paramètre k_{wd} est la valeur maximale de la productivité de l'acteur w pour le département d . Le paramètre r_{wd} représente la quantité de temps cumulé (même unité de mesure que x_{wd}) pour que la productivité de l'acteur w puisse atteindre la moitié de la valeur maximale de la productivité ($k_{wd} / 2$).

3.4.2 Conclusion

A notre connaissance, le travail de [Sayin et Karabati, 2007] est le seul que l'on peut classer dans la catégorie "affectation mono-période" et "évolution de compétence". Les auteurs s'intéressent à l'affectation de type inter-département "cross-training" dans les services hospitaliers. En maximisant l'augmentation des compétences au cours de la réalisation des tâches, on obtient une proposition d'équipes performantes. La modélisation des compétences des ressources humaines a été pris en compte, sous forme de domaines d'expertise (ou départements). Les acteurs sont caractérisés par de multiples compétences. Ce travail introduit la notion de l'"évolution de compétences" dans le problème d'affectation. Les auteurs supposent dans cette étude que l'affectation des tâches est un mécanisme permettant d'augmenter les niveaux de compétences des individus. Une courbe d'apprentissage individuelle a été envisagée pour estimer l'augmentation du niveau de compétence. Cependant, cette courbe d'apprentissage semble délicate à paramétrer et nécessite de nombreux paramètres qui paraissent difficiles à estimer. Globalement, la formulation du problème en deux sous-problèmes ou la résolution d'un problème complexe en deux étapes est une méthode d'optimisation intéressante pour traiter des problèmes complexes. Le couplage est réalisé ici par le passage de la valeur optimale d'une étape de résolution à l'autre, afin de contraindre l'espace des solutions satisfaisantes. Enfin, ce modèle étant mono-période, son utilité semble limitée dans les applications réelles.

3.5 Affectation multi-périodes avec une modélisation dynamique des compétences

Certains travaux de recherche se sont intéressés à l'affectation des tâches aux ressources sur plusieurs périodes et à la prise en compte de l'évolution des compétences. Les travaux trouvés dans la littérature sont résumés ci-dessous.

3.5.1 Maximisation de la compétence avec effets d'apprentissage

[Gutjahr *et al.*, 2008] proposent un modèle qui optimise la gestion de portefeuille de projets (projet porfolio) et l'affectation des tâches à des ressources à long terme en prenant en compte les effets d'apprentissage et de dépréciation de connaissance. Le projet est nommé, PSSSL²⁰.

Le modèle est constitué d'un ensemble de projets i ($i=1, \dots, n$), la variable y_i est une variable de décision binaire pour la sélection du projet candidat, $y_i \in \{0,1\}, \forall i$. La planification est réalisée selon l'horizon de temps t ($t=1, \dots, T$). Pour chaque projet i , la valeur w_i est le poids d'importance représentant le gain économique que l'entreprise peut tirer, si le projet i est sélectionné. Chaque projet est constitué de plusieurs tâches k ($k=1, \dots, K$). L'entreprise possède du personnel j ($j=1, \dots, m$). Chaque personne j peut posséder une ou plusieurs compétences r ($r=1, \dots, R$). Toutes les compétences n'ont pas les mêmes niveaux d'importance. Donc un niveau d'importance relative v_r est assigné pour chaque compétence. Des variables de décisions réelles sont introduites : $x_{kjrt} \in [0,1]$. La variable x_{kjrt} représente une fraction de réalisation de la tâche pendant lequel la personne j travaille avec la compétence r sur la tâche k pendant la période t .

Contraintes liées à l'évolution des compétences

La valeur z_{jrt} représente le niveau de maîtrise de la compétence r requis par l'acteur j à la période t . La valeur z_{jrt} est une variable de compétence réelle, par exemple $z_{jrt} \in [0,100]$. Le troisième indice indique que le niveau de maîtrise de la compétence r de l'acteur j évolue dans le temps (par l'effet d'apprentissage). La valeur initiale de z_{jrt} ($t=1$) est supposée connue. La valeur γ_{jrt} est une valeur d'efficacité de la personne j pour la compétence r pendant la période t . Une première contrainte liée à la relation entre la valeur d'efficacité γ_{jrt} et le niveau de maîtrise de la compétence z_{jrt} est définie par une fonction.

$$\gamma_{jrt} = \varphi(z_{jrt}).$$

Une deuxième contrainte décrit l'évolution des compétences en prenant en compte l'apprentissage et la dépréciation des compétences. Les paramètres η_r et β_r sont respectivement appelés le taux d'apprentissage et le taux de dépréciation de la compétence r . On suppose que le niveau de maîtrise d'une personne j de la compétence r augmente à chaque période (si la personne j a travaillé pendant une durée x du temps) par un incrément de $\eta_r \times x$. De la même façon, on suppose que le niveau de compétence d'une personne j dans la compétence r diminue d'une quantité de β_r à chaque période en raison d'une dépréciation des compétences. On suppose que $\eta_r > \beta_r$.

²⁰ Project Selection, Scheduling and Staffing with Learning problem

$$z_{jrt} = z_{jr1} - \beta_r(t-1) + \eta_r \sum_{k=1}^K \sum_{s=1}^{t-1} x_{kjrs} \quad \forall j, r, t \quad ^{21}$$

Contraintes liées à la charge du personnel

Cette contrainte assure que, à une période donnée, la charge de travail de chaque acteur ne dépasse pas sa limite, quel que soit le type de compétence r et la tâche k . La valeur $a_{jt} \in [0,1]$ est la capacité de travail de l'acteur j lors d'une période donnée t .

$$\sum_{k=1}^K \sum_{r=1}^R x_{kjrt} \leq a_{jt} \quad \forall j, t$$

Contraintes liées à la charge de la compétence r

Le paramètre d_{kr} est la durée estimée nécessaire pour effectuer la partie de la tâche k liée à la compétence r . Cette durée correspond à une durée de référence correspondant au travail d'une personne ayant une efficacité $\gamma_{jrt} \equiv 1$. La valeur binaire c_{ik} représente l'appartenance de la tâche k au projet i : $c_{ik}=1$ si le projet i est constitué de la tâche k , elle vaut 0 sinon.

La première contrainte impose que chaque tâche appartient à un seul projet.

$$\sum_{i=1}^n c_{ik} = 1, \forall k = 1, \dots, K.$$

La deuxième contrainte assure que le temps réel de mise en œuvre d'une compétence r pour réaliser la tâche k doit être égal à la charge de travail pour la compétence r de la tâche k :

- Le temps réel de mise en œuvre d'une compétence r pour réaliser la tâche k est obtenu à partir du cumul sur toutes les périodes t des durées réelles de travail de toutes les personnes j pour la tâche k liée à la compétence r (x_{kjrt}), multipliée par leur efficacité (γ_{jrt}). La durée de réalisation de la tâche k est la sommation en t partant de la date de début ρ_k ($\rho_k \in \{1, \dots, T\}$) à la date de la fin δ_k ($\delta_k \in \{1, \dots, T\}$).
- Pour tous les projets candidats, on regarde si la tâche k fait partie du projet sélectionné i ($y_i=1$ et $c_{ik}=1$). La somme du terme de droite permet d'obtenir le nombre d'occurrence de cette tâche dans les projets, qui, multiplié par la durée estimée nécessaire d_{kr} , permet d'obtenir la charge de travail pour la compétence r de la tâche k .

$$\sum_{t=\rho_k}^{\delta_k} \sum_{j=1}^m \gamma_{jrt} x_{kjrt} = d_{kr} \sum_{i=1}^n c_{ik} y_i \quad \forall k, r$$

La troisième contrainte assure que l'ensemble des acteurs j ne doit pas dépasser un certain nombre d'heures de travail sur la compétence r de la tâche k lors de la période t . La valeur b_{kr}

²¹ A noter l'utilisation de l'indice s ici pour représenter la période s et pouvoir faire la sommation sur toutes les périodes jusqu'à $t-1$.

représente l'effort maximum consenti pour réaliser la compétence r dans la tâche k pour chaque période t .

$$\sum_{j=1}^m \gamma_{jrt} x_{kjrt} \leq b_{kr} \quad \forall k, r, t$$

Fonction objectif

La fonction objectif est définie comme suit.

$$\text{Maximiser} \quad \sum_{i=1}^n w_i y_i + \sum_{r=1}^R v_r \sum_{j=1}^m (\gamma_{j,r,T+1} - \gamma_{j,r,t})$$

La première partie de la fonction objectif vise à maximiser le gain économique (w_i) de l'entreprise par la sélection d'un ensemble de projet candidat (y_i). La deuxième partie de la fonction objectif maximise les améliorations stratégiques des compétences en multipliant le poids d'importance stratégique d'une compétence r (v_r) par l'incrément de la valeur d'efficacité des ressources (γ_{jrt}) sur un horizon de temps. Les valeurs de variable w_i et v_r doivent être définies de manière comparable pour rendre la fonction objectif homogène et interprétable. Le problème considéré est non linéaire et il est résolu par des heuristiques.

Ce travail est essentiellement basé sur un modèle de gestion de portefeuille de projets. La prise en compte du niveau de compétence par rapport à l'efficacité du travail nous paraît intéressante comme hypothèse. Les compétences des ressources sont supposées évoluer dans le temps en fonction des affectations effectuées et du taux d'apprentissage défini pour chaque compétence.

Cependant, le modèle présente certaines limites. D'une part, le taux de dépréciation est défini par une constante. Par conséquent, le niveau de compétence est supposé diminuer constamment pour toutes les périodes et tous les niveaux de compétence initiaux. En réalité, la dépréciation peut prendre la forme d'une courbe en S où le niveau de compétence diminue lentement pour les experts (haut niveau de compétence) et se dégrade plus rapidement pour les débutants (bas niveau de compétence). D'autre part, la dépréciation du niveau de compétence est prise en compte même si les acteurs mettent en œuvre ces compétences. En effet, si les acteurs travaillent sur une compétence, son niveau de maîtrise devrait être maintenu, voire amélioré.

3.5.2 Minimisation des coûts liés à la formation des ressources

[Fowler *et al.*, 2008] s'intéressent à l'affectation multi-périodes des tâches aux ressources et à l'apprentissage des membres d'équipes. La flexibilité de l'affectation dans les processus de fabrication est reconnue comme une méthode favorisant une meilleure utilisation des acteurs. Les acteurs bénéficient d'une alternance de tâches (cross-training) à chaque affectation. Cela permet également d'améliorer la capacité d'apprentissage des acteurs (utilisation de plusieurs machines), d'augmenter la communication, d'améliorer la motivation des opérateurs, etc. Un modèle de programmation linéaire mixte²² basée sur des heuristiques est proposé pour

²² en anglo-saxon, Mixed integer programming.

déterminer différentes décisions (par exemple : embauche, apprentissage tournant et licenciement). Les techniques d'optimisation de type algorithme génétique ont été également utilisées pour comparer les résultats obtenus avec ceux de la programmation linéaire mixte basée sur les heuristiques proposées. Les résultats montrent que la technique d'optimisation utilisée par les auteurs permet de résoudre des problèmes de taille importante, efficacement et avec un temps de calcul raisonnable, alors que l'algorithme génétique fournit les meilleurs résultats mais nécessite un temps de calcul important.

Dans ce modèle, une compétence représente la capacité d'une personne à travailler sur une machine. Les compétences d'un acteur vont évoluer au cours de la réalisation des tâches (ou opérations). Elles sont définies par un indicateur GCA (General Cognitive Ability) représentant la capacité d'apprendre et de traiter des informations. Il est utilisé pour estimer la capacité d'apprentissage de l'individu sur plusieurs niveaux. Chaque personne possède un niveau g , où $g \in \{1, \dots, G\}$. L'indicateur GCA est calculé à partir de deux variables α et β . La variable α représente l'ensemble des compétences avant l'affectation et la variable β représente l'ensemble des compétences après l'affectation, où $\alpha \subset \beta$.

Les variables de décision sont définies comme suit.

$U_{\alpha g}^t$ = nombre de personnes du niveau g possédant l'ensemble des compétences α embauchées à la période t

$X_{\alpha\beta g}^t$ = nombre de personnes du niveau g à former pour le faire passer de l'ensemble des compétences α à l'ensemble β à la période t

$\xi_{\alpha g}^t$ = nombre de personnes du niveau g possédant l'ensemble des compétences α à la période t

$W_{\alpha g}^t$ = nombre de personnes du niveau g possédant l'ensemble des compétences α , licenciées à la période t

Autres variables :

S = ensemble des sous-ensembles non vide, $|S| = 2^M - 1$

G = nombre de niveaux GCA

$h_{\alpha g}^t$ = coût d'emploi de la personne du niveau g possédant l'ensemble des compétences α , à la période t

$c_{\alpha\beta g}^t$ = coût de formation de la personne du niveau g pour passer de l'ensemble des compétences α à β , à la période t

$s_{\alpha g}^t$ = salaire de la personne du niveau g possédant l'ensemble des compétences α , à la période t

$f_{\alpha g}^t$ = coût pénalisant le licenciement de la personne du niveau g possédant de l'ensemble des compétences α , à la période t

$\xi_{\alpha g}^t$ = nombre de personnes pour chaque ensemble des compétences selon la décision de l'apprentissage de chaque période

La fonction objectif retenue est la suivante :

$$\text{Minimiser } \sum_{g=1}^G \left[\sum_{\alpha \in S} h_{\alpha g}^t U_{\alpha g}^t + \sum_{\beta \in S} \sum_{\alpha < \beta} c_{\alpha \beta g}^t X_{\alpha \beta g}^t + \sum_{\alpha \in S} s_{\alpha g}^t \xi_{\alpha g}^{t-1} + \sum_{\alpha \in S} f_{\alpha g}^t W_{\alpha g}^t \right]$$

La contrainte liée au pilotage des compétences est présentée ci-dessous.

$$\xi_{\alpha g}^t = \xi_{\alpha g}^{t-1} + U_{\alpha g}^t + \sum_{v: v < \alpha} X_{v \alpha g}^t - \sum_{\beta: \alpha < \beta} X_{\alpha \beta g}^t - W_{\alpha g}^t,$$

$$\forall \alpha, \beta \in S, \forall g \in \{1, \dots, G\}, \forall t \in \{1, \dots, T\}$$

La variable $\xi_{\alpha g}^t$ est une variable définissant le nombre de personnes qualifié pour chaque ensemble des compétences. Cette variable est définie selon les décisions de "cross-training" de chaque période. Elle représente le bilan du nombre d'acteurs possédant l'ensemble des compétences (α) à la période t . Elle dépend de plusieurs variables de décision (embauche, apprentissage tournant et licenciement).

Ce travail représente une autre formulation possible du problème d'affectation des tâches aux ressources dans un atelier de fabrication. Il tient compte de plusieurs décisions liées à l'apprentissage tournant pour piloter les compétences des ressources. La prise en compte de la dynamique des compétences des acteurs est évidente à travers les variables modélisées. Cependant, le modèle de l'évolution de compétences nous paraît intéressant mais difficile à mettre en place dans le domaine de conception car les données manipulées peuvent être difficiles à obtenir. Par ailleurs, la compétence est gérée de façon globale sans modéliser ses constituants. Ce type de modélisation semble fréquent dans une approche de type gestion des ressources humaines.

3.5.3 Conclusion

Nous avons trouvé deux travaux [Gutjahr *et al.* 2008], [Fowler *et al.*, 2008] qui peuvent être classés dans la catégorie, "affectation multi-périodes" avec modélisation de "l'évolution de compétence". Le travail de [Gutjahr *et al.*, 2008] est un modèle appliqué au domaine de management de projet. Le travail de [Fowler *et al.*, 2008] est un modèle appliqué au domaine de la gestion des ressources humaines en production.

3.6 Synthèse

A partir de l'étude bibliographique, nous retenons les points principaux suivants.

3.6.1 Modélisation de l'évolution de compétences dans les problèmes d'affectation

Cette étude permet de mettre en évidence les variables concernant l'évolution des compétences.

Le tableau (TAB. 3.1) présente une classification des problèmes d'affectation selon le type d'affectation (mono-période et multi-périodes) et le type de modélisation des compétences (statique et dynamique) [Hlaoittinun *et al.*, 2007b].

Affectation	Mono-période	Multi-périodes
Sans évolution de compétence	<ul style="list-style-type: none"> ▪ [Caron <i>et al.</i>, 1999] ▪ [Campbell et Diaby, 2002] ▪ [Eiselt et Marianov, 2008] ▪ [Peters et Zelewski, 2007] 	<ul style="list-style-type: none"> ▪ [Miller et Franz, 1996] ▪ [Bellenguez-Morineau, 2006] ▪ [Corominas <i>et al.</i>, 2006]
Avec évolution de compétence	<ul style="list-style-type: none"> ▪ [Sayin et Karabati, 2007] 	<ul style="list-style-type: none"> ▪ [Gutjahr <i>et al.</i> 2008] ▪ [Fowler <i>et al.</i>, 2008]

TAB. 3.1 : Classification des problèmes d'affectation

Le tableau de classification ci-dessus suggère les remarques suivantes :

- Les travaux d'affectation "mono-période" et "multi-périodes" de type "sans évolution de compétence" nous permettent d'analyser les intérêts et les limites des formulations mathématiques modélisant l'utilisation des compétences dans l'affectation des tâches à des ressources. Ils ouvrent la voie à la formulation du problème plus complexe en s'appuyant sur la notion de compétence.
- Les travaux d'affectation "mono-période" et "multi-période" avec "évolution de compétence" modélisent l'évolution des compétences. Néanmoins, nous constatons qu'il y a peu de travaux effectués dans ces catégories : un travail dans le cas mono-période [Sayin et Karabati, 2007] et deux travaux dans le cas multi-périodes [Gutjahr *et al.* 2008] et [Fowler *et al.*, 2008].
- Le travail de [Sayin et Karabati, 2007] fournit une formulation intéressante du problème avec l'utilisation d'une courbe d'apprentissage dans l'estimation de l'augmentation du niveau de compétence des acteurs. Les auteurs s'intéressent à l'affectation de type "cross-training" dans les services hospitaliers. Les travaux se focalisent sur l'affectation de type inter-département (l'affectation des ressources aux départements). L'augmentation des compétences est reliée essentiellement au déplacement des ressources dans différentes situations de travail (départements). Une courbe d'apprentissage a été utilisée, mais son application semble compliquée. Néanmoins, ce travail reste un cas mono-période dont l'utilité paraît limitée dans les applications réelles.
- Le modèle de [Fowler *et al.*, 2008] nous paraît une bonne formulation pour résoudre le problème d'affectation dans les processus de fabrication avec un meilleur pilotage des compétences des acteurs. Les objectifs de compétence ont été définis pour mobiliser des compétences des acteurs. Les auteurs s'intéressent à déterminer les décisions de gestion des ressources humaines permettant de piloter les compétences des ressources humaines (par exemple : embauche, apprentissage tournant, licenciement). Cependant, le modèle d'évolution des compétences proposé semble difficile à utiliser dans le domaine de la conception.
- Le travail de [Gutjahr *et al.* 2008] est une autre approche de la formulation du problème d'affectation auquel nous nous intéressons. Il traite à la fois l'*identification de compétences* et l'*affectation des ressources* en fonction de la *dynamique de compétences*. Cependant, le modèle de l'évolution des compétences présente certaines limites concernant la courbe de dépréciation des compétences. Les conséquences (coût

ou durée supplémentaire) de la sélection des acteurs sous-compétents n'ont pas été prises en compte.

Malgré l'importance de la modélisation dynamique des compétences dans le pilotage des compétences, très peu de travaux s'y sont intéressés [Hlaoittinun *et al.*, 2009]. Cela est dû à la complexité de la définition des lois d'évolution. Nous pensons qu'une proposition de lois d'évolution des compétences doit être intégrée dans la démarche afin de permettre aux chefs de projets d'améliorer le pilotage des compétences des ressources pour répondre à de nouveaux objectifs de compétences. Nous proposerons une loi d'évolution des compétences dans le chapitre 4.

3.6.2 Méthode de résolution et critère d'optimisation

Le tableau (TAB. 3.2) présente une synthèse de l'état de l'art sur l'affectation des tâches à des ressources en mettant en évidence les critères différenciant ces travaux : domaine d'application, fonction objectif, méthode de résolution et type de compétence modélisé (connaissance techniques et capacités opératoires).

De nombreuses méthodes de résolution ont été explorées pour traiter les problèmes d'affectation des tâches à des ressources [Hlaoittinun *et al.*, 2008]. Il nous a été impossible de les comparer les unes par rapport aux autres dans la mesure où la complexité des situations étudiées et les contraintes en découlant sont variables d'un problème à l'autre.

Enfin, on constate à travers cet état de l'art qu'en raison de la complexité de la formulation du problème avec évolution de compétences, les résolutions reposent sur l'usage d'heuristiques [Gutjahr *et al.* 2008], [Fowler *et al.*, 2008] ou de méta-heuristiques.

	Référence	Domaine d'application	Problème d'optimisation traité		Compétence	
			Fonction objectif	Méthode de résolution	Connaissance technique	Capacité opératoire
Mono période Sans évolution de compétence	[Campbell et Diaby, 2002]	Service hospitalier	Maximiser de l'incrément de la compétence	Heuristique	Connaissance technique	--
	[Caron <i>et al.</i> , 1999]	Affectation Qualification	Maximiser la compétence	Programmation linéaire	Connaissance technique	--
	[Eiselt et Marianov, 2008]	Affectation Satisfaction des employés	Minimiser la différence entre tâche/acteur	Programmation linéaire	Attribut de compétences	Satisfaction du travail
	[Peter et Zelewski, 2007]	Management des ressources humaines	Minimiser la différence entre tâche/acteur (undercompétence)	Goal programming	Connaissance technique	Lieu de travail, attitude des employés
Mono période Avec l'évolution de compétence	[Sayin et Karabati, 2007]	Service hospitalier	Maximiser l'utilisation totale des ressources Maximiser l'augmentation des compétences	Programmation linéaire mixte	Connaissance technique	--
Multi période Sans l'évolution de compétence	[Bellenguez-Morineau, 2006]	Management du projet	Maximiser la date d'achèvement (Cmax)	Programmation linéaire en nombres entiers	Connaissance technique	--
	[Corominas <i>et al.</i> , 2006]	Altération des tâches	Minimiser l'écart par rapport à la valeur de référence du temps de travail	Algorithme de Jonker et Volgenant ²³	Connaissance technique	--
Multi période Avec l'évolution de compétence	[Gutjahr <i>et al.</i> 2008]	Management du R&D projet, portfolio	Maximiser l'efficacité à travers la compétence	Heuristique	Connaissance technique	--
	[Fowler <i>et al.</i> , 2008]	Cross training	Maximiser les coûts liés aux décisions du personnel	Programmation linéaire, résolution basée sur des heuristiques	Connaissance technique	--

TAB. 3.2 : Répartition des travaux de constitution d'équipes à contraintes de compétences.

²³ Un algorithme proposé pour résoudre les problèmes de type "linear assignment problems", ou LAP.

4 Affectation des tâches contribuant au pilotage des compétences

Dans ce chapitre, nous présentons deux contributions, l'une relative à l'affectation mono-période et l'autre relative à l'affectation multi-périodes.

Dans la première contribution, nous nous plaçons dans un contexte mono-période, mono-projet, sans évolution des compétences. Le modèle proposé servira de base pour développer ensuite le cas multi-périodes. Cette démarche permet d'affecter les tâches aux acteurs lorsque le projet est en phase de préparation. Nous proposons un indicateur de similarité, qui estime la compatibilité entre tâches et acteurs et qui prend en considération la sous-compétence des acteurs. L'exigence des tâches et la polyvalence des acteurs sont modélisées sous la forme de profils multi-connaissances. Le principe est validé à travers un cas d'école.

Dans la deuxième contribution, nous considérons un contexte multi-projets et multi-périodes avec évolution des compétences. Une démarche d'affectation répondant à ce contexte sera présentée. Nous proposons un modèle d'évolution de compétences se basant sur une hypothèse d'apprentissage lors de la réalisation des tâches. Enfin, nous testons notre méthode d'affectation multi-périodes sur un jeu de données extrait d'un projet de conception d'une capsule endoscopique.

4.1 Introduction

Les projets de conception sont organisés de telle manière qu'il est nécessaire à des métiers différents de co-concevoir un produit en maîtrisant la qualité, les délais et les coûts. La constitution d'équipes performantes est d'autant plus cruciale que le produit est complexe, et le nombre de concepteurs important. Les responsables ont besoin d'outils pour les aider à affecter les concepteurs aux projets. L'affectation des tâches à des acteurs est considérée comme un mécanisme important permettant de sélectionner les membres d'équipe compétents pour la réalisation de l'ensemble de tâches du projet.

Dans l'affectation des tâches aux ressources, de nombreuses contraintes sont à satisfaire : les contraintes de compétence, les contraintes de disponibilité, les contraintes organisationnelles, les contraintes de performance, etc. [Bennour *et al.*, 2008]. Les affectations prenant en compte des contraintes de performance tentent souvent d'assurer la performance visée globalement des tâches par les ressources allouées. Traditionnellement, les algorithmes proposés cherchent à affecter les meilleurs acteurs (les experts) en premier. Cette tactique d'affectation ne permet pas aux acteurs sous-compétents d'améliorer leurs compétences.

Dans un environnement compétitif, les compétences des ressources humaines doivent être maintenues voire améliorées. Les entreprises doivent être capables de mobiliser les compétences de leurs salariés, face à des situations de plus en plus complexes. Les directeurs de programmes et les chefs de projets doivent déterminer des solutions d'affectation qui satisfassent les objectifs de compétence fixés par le management stratégique. Ces objectifs peuvent être formulés en fonction du nombre d'experts (par exemple, trois experts en mécanique et quatre experts en vision dans un an,...). Il est donc nécessaire de trouver un système permettant d'affecter des tâches à des acteurs sous-compétents et ainsi de les former. Néanmoins, la non-performance de l'équipe formée peut rendre la tâche réalisée de mauvaise qualité. [Bennour *et al.*, 2008] affirme également que l'on ne peut comparer l'efficacité d'un employé débutant à celle d'un employé expérimenté. Le niveau de compétence influence fortement la durée des activités ainsi que le coût et la qualité associés. Il est possible de s'affranchir de cette situation en considérant que des experts vont accompagner les acteurs sous-compétents et que les compétences des ressources évoluent au cours de la réalisation des tâches.

Nous divisons ce chapitre en deux grandes parties. La première partie formule un modèle mono-période permettant d'allouer des acteurs compétents mais aussi sous-compétents aux tâches, avec l'aide de tuteurs, afin d'atteindre la performance demandée, sans toutefois considérer que les compétences de ces acteurs peuvent évoluer durant l'affectation. Le but de ce premier modèle est de formuler les surcoûts liés au cas de sous-compétence sur un cas simple (mono-période et mono-projet). La deuxième partie s'intéresse à un modèle d'affectation mono-période qui intègre à la fois le surcoût des acteurs sous-compétents et le coût lié à l'évolution de leurs compétences au cours de la réalisation des tâches. Ce deuxième cas sera traité en multi-périodes et multi-projets.

4.2 Affectation mono-période

Dans cette partie, nous présentons d'abord les qualités attendues de la méthode proposée. Ensuite, nous proposons une caractérisation des compétences qui sera intégrée dans la démarche d'affectation. Nous présentons la structure globale du modèle en utilisant un diagramme fonctionnel S.A.D.T. Puis, nous proposons un indicateur de similarité permettant

d'estimer le niveau de compétence d'un acteur pour la réalisation de la tâche associée. Enfin, nous proposons une formulation mathématique et un test du principe utilisant la programmation linéaire comme méthode d'optimisation.

4.2.1 Qualités attendues de la méthode

Les qualités attendues d'une telle méthode sont les suivantes :

- E1 : Affecter des tâches aux acteurs en minimisant le coût salarial lié aux niveaux de compétences. Définir les acteurs pouvant potentiellement réaliser des tâches de façon à ce que le surcoût lié à un manque de compétences soit minimal.
- E2 : Proposer un indicateur de similarité entre tâches et acteurs
- E3 : Prendre en compte l'évolution des compétences dans l'affectation des tâches.

Ces trois exigences ne sont pas de même nature. La première exigence E1 est l'objectif de base souvent considéré dans les problèmes d'affectation. L'exigence E2 est un sous-problème de E1. Il s'agit de définir et d'utiliser un indicateur de similarité entre tâches et acteurs, qui soit approprié. L'exigence E3 est une contrainte ou un objectif du problème d'affectation, dont le but est de prendre en compte l'évolution des compétences lors de l'affectation.

4.2.2 Hypothèses de caractérisation des compétences

Nous utilisons le concept proposé par F. Belkadi (§1.4.2) pour lequel la finalité d'une compétence est de permettre à l'acteur qui la possède de réussir une tâche qui lui a été confiée. Pour activer une compétence, l'acteur doit avoir : des connaissances techniques (savoirs théoriques, savoir-faire, plan d'actions), des capacités opératoires (par exemple : capacités d'analyse, capacités d'organisation) et des règles de conduite. Nous considérons ces composants dans une vue organique des compétences. Nous nommons dans ce document ces différents éléments sous le terme générique de "connaissance".

Ainsi, nous choisissons de modéliser et d'évaluer une compétence à partir des connaissances qu'elle mobilise, plutôt que d'évaluer globalement la compétence.

Nous faisons l'hypothèse que la tâche et l'acteur peuvent être caractérisés par un ensemble de connaissances. Le choix des types de connaissances modélisés dépend de l'objectif attendu, qui est différent d'un travail à l'autre (un objectif orienté connaissance technique, un objectif orienté capacité opératoire, etc.). Le niveau de compétence d'un acteur peut être estimé à l'aide d'un indicateur de similarité entre la tâche et l'acteur.

4.2.3 Structure générale de la méthode

Afin de présenter la structure générale de la méthode proposée, nous utilisons une représentation graphique (S.A.D.T.). La représentation S.A.D.T est une décomposition fonctionnelle partant de la fonction globale du système jusqu'à l'obtention de sous-fonctions interdépendantes.

Nous présentons le système d'affectation des tâches aux acteurs sous 2 angles : la vue externe et la vue interne. La vue externe du système permet de modéliser le système comme une boîte noire avec des interactions avec l'environnement. La vue interne consiste à décomposer le système en sous-systèmes interagissant.

Dans la vue externe (FIG. 4.1), le système reçoit une liste de tâches, une liste de compétences et un ensemble d'acteurs comme des données d'entrée. Ce système contient des mécanismes d'affectation. A travers ces mécanismes, nous obtenons les acteurs alloués au projet comme sorties du système.

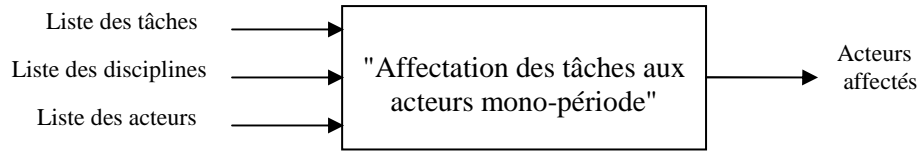


FIG. 4.1 : Vue externe de la démarche d'affectation mono-période des tâches aux acteurs

Nous définissons un ensemble de tâches, indexées par i ($i=1, \dots, M$), un ensemble d'acteurs, indexées par j ($j=1, \dots, N$), un ensemble de connaissances, indexées par c , ($c=1, \dots, C$).

Nous regroupons dans le tableau ci-dessous les notations utilisées dans la formulation du problème.

T_i	Tâche de type i ; ($i=1, \dots, M$)
A_j	Acteur j ; ($j=1, \dots, N$)
C_c	Connaissance de type c ; ($c=1, \dots, O$)
rq_i	Niveau de compétence requis par la tâche générique i
rq'_i	Niveau de compétence requis par une tâche particulière de type i
$R1$	Matrice de niveaux de maîtrise requis dans les connaissances (MI tâche-connaissance), $R1 = (r_{1,i,c})_{1 \leq i \leq M, 1 \leq c \leq O}$
$r_{1,i,c}$	Niveau de maîtrise requis dans la connaissance c par la tâche i
$R2$	Matrice de niveaux de maîtrise acquis dans les connaissances (MI connaissance-acteur), $R2 = (r_{2,c,j})_{1 \leq c \leq O, 1 \leq j \leq N}$
$r_{2,c,j}$	Niveau de maîtrise acquis dans la connaissance c par l'acteur j .
$R3$	Matrice de compétence (matrice de similarité) $R3 = (r_{3,i,j})_{1 \leq i \leq M, 1 \leq j \leq N}$
$r_{3,i,j}$	Indicateur de similarité entre la tâche générique i et l'acteur j (niveau de compétence)
$v_{3,i,j}$	Indicateur de similarité entre la tâche particulière i et l'acteur j

TAB. 4.1 : Variables dans l'affectation mono-période

Dans la vue interne du système (FIG. 4.2), l'activité principale "affecter des tâches aux acteurs mono-période" se décompose en trois phases.

- Première phase : elle vise à caractériser la tâche et l'acteur par les connaissances à maîtriser ou maîtrisées. Nous dressons une liste des tâches à réaliser dans le projet et une liste des connaissances génériques qui sont nécessaires à la réalisation des tâches. Ensuite, nous déterminons et évaluons le niveau de maîtrise dans la connaissance requis

par la tâche et acquis par l'acteur. Nous obtenons donc la **MI (matrice d'incidence) tâche-connaissance** et la **MI acteur-connaissance**.

- Deuxième phase : elle vise à calculer l'indicateur de similarité entre tâches et acteurs. L'indicateur de similarité est obtenu à partir des deux matrices d'incidence de la première phase (MI tâche-connaissance et MI acteur-connaissance). La valeur de l'indicateur de similarité varie entre 0 et 1 et est calculée pour chaque couple tâche-acteur afin d'obtenir une **matrice de compétence** (matrice de similarité entre tâches et acteurs).
- Troisième phase : elle vise à affecter les tâches aux acteurs. Nous obtenons une **matrice d'affectation**.

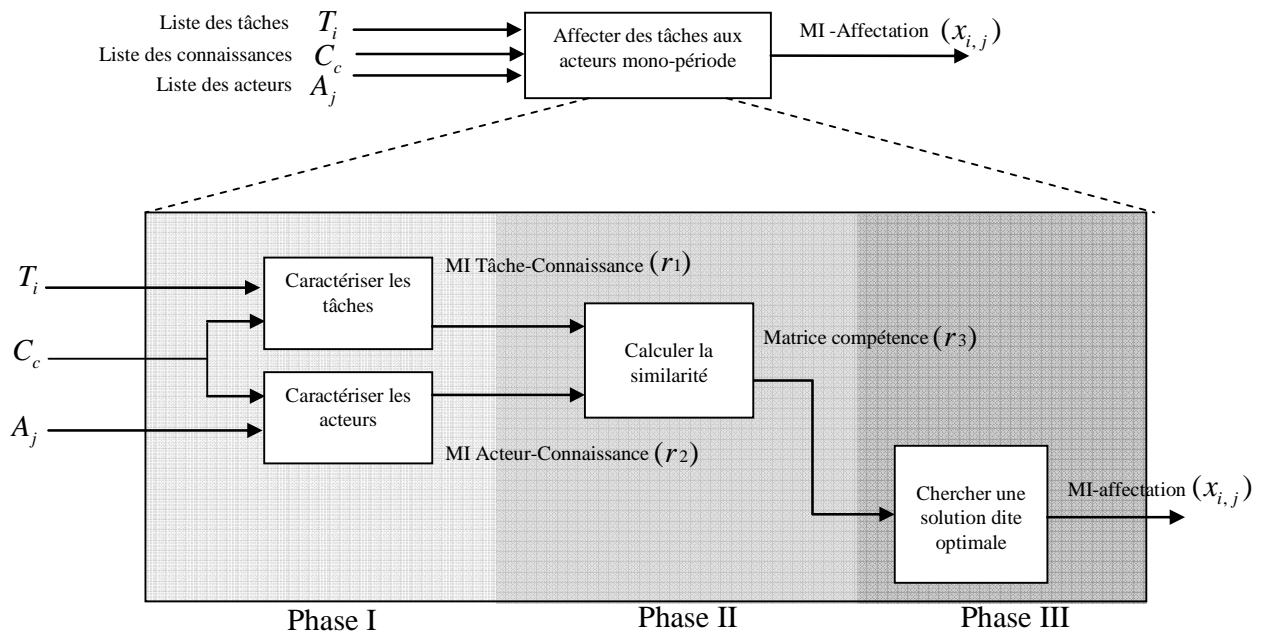


FIG. 4.2: Démarche d'affectation mono-période des tâches aux acteurs

Avant de présenter ces phases plus en détail, nous définissons deux types de tâches utilisés dans ce travail : la tâche générique et la tâche particulière (une instanciation de la tâche générique).

- Tâche générique (T_i)

La réalisation d'une tâche sur un projet est dépendante des événements qui sont propres à ce projet et qui peuvent varier d'un projet à un autre (par exemple : les collaborateurs différents, les clients différents). Cependant, le contenu de la tâche étant semblable, il est pratique d'abstraire la définition de ces tâches en une **tâche générique** qui représente alors une classe de tâches considérées comme semblables. Par exemple, "définir l'architecture du train avant" peut être une tâche générique, qui nécessite la même compétence que ce soit sur une plateforme "bas segment de gamme" ou "moyen de gamme", que le chef de projet soit Monsieur X ou Monsieur Y. Seul le niveau de compétence requis peut alors varier.

- Tâche particulière (T'_i)²⁴

²⁴ Pour des raisons de simplification des notations, nous considérons une seule tâche particulière par projet.

La tâche particulière (T'_i) est propre à un contexte d'application. Elle appelle les mêmes types de connaissances que la tâche générique T_i mais elle peut posséder un niveau requis de compétence différent.

Les trois phases de la démarche d'affectation mono-période que nous avons présentée dans la figure (FIG. 4.2) sont développées dans les parties suivantes.

4.2.3.1 Phase I : Caractérisation de la tâche et de l'acteur

Les compétences peuvent être identifiées à la fois par rapport aux tâches à réaliser et par rapport aux acteurs qui les détiennent. Nous décomposons les variables de compétence en deux niveaux de modélisation (granularité) : niveau "*compétence*" et niveau "*connaissance*". Le niveau "*connaissance*" représente un niveau de granularité plus fin que le niveau "*compétence*".

Nous distinguons 3 entités à modéliser :

- la tâche générique (T_i)
- la tâche particulière (T'_i)
- les acteurs (A_j)

Caractérisation de la tâche générique (T_i)

- au niveau de la "*compétence*" : nous modélisons le niveau requis de compétence concernant la réalisation de la tâche générique de type i (rq_i). La tâche générique étant considérée comme une tâche de référence, le niveau de compétence requis par cette tâche sera pris égal à 1 ($rq_i = 1$).
- au niveau des "*connaissances*" : il s'agit du niveau de maîtrise de la connaissance c requis par la tâche i ($r_{i,c}$). Pour être capable de réaliser la tâche générique T_i , un ensemble de connaissances est requis et chaque type de connaissance est associé à un niveau de maîtrise ($r_{i,c}$). Le tableau (TAB. 4.2) donne un exemple de description d'une tâche générique par un ensemble de connaissances et des niveaux de maîtrise associés.

$(rq_i = 1)$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
$r_{i,c}$	0,8	0	0,2	0	0,1	0,9	0	1	0	0

TAB. 4.2 : Exemple du niveau de maîtrise de la connaissance c requis par la tâche générique i

Ce tableau peut s'interpréter de la façon suivante : réaliser la tâche T_i (avec $rq_i = 1$) est équivalent à réaliser la tâche qui requiert les niveaux de maîtrise dans les "connaissances" ($r_{i,c}$). Une fois les connaissances nécessaires et leurs niveaux de maîtrise identifiés, nous obtenons la **MI tâche – connaissance (RI)**.

$$R1 = (r1_{i,c})_{1 \leq i \leq M, 1 \leq c \leq O}$$

Caractérisation de la tâche particulière (T'_i)

- Concernant la modélisation au niveau de la "compétence", il s'agit du niveau requis de compétence concernant la tâche particulière de type i (rq'_i). Nous faisons l'hypothèse qu'une tâche particulière requiert un niveau de compétence inférieur à celui de la tâche générique ($rq'_i \leq rq_i$).
- Concernant la modélisation au niveau des "connaissances", il s'agit du niveau de maîtrise de la connaissance c requis par la tâche particulière de type i ($r1'_{i,c}$)²⁵.

Nous faisons l'hypothèse que le niveau de maîtrise dans la discipline requis par la tâche particulière de type i ($r1'_{i,c}$) peut être calculé en faisant le produit du niveau de compétence concernant la tâche particulière de type i (rq'_i) et du niveau de maîtrise dans la connaissance requis par la tâche générique de type i ($r1_{i,c}$).

$$r1'_{i,c} = rq'_i \times r1_{i,c}$$

Eq- 4-1

Le niveau de compétence (rq'_i) est considéré comme un paramètre permettant de modéliser le niveau de maîtrise dans les connaissances requis par la tâche particulière de type i ($r1'_{i,c}$).

Nous présentons ci-dessous un exemple de calcul du niveau de maîtrise de la connaissance ($r1'_{i,c}$) requis par la tâche particulière T'_i , ($rq'_i = 0,8$).

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
$r1_{i,c}, (rq_i = 1)$	0,8	0	0,2	0	0,1	0,9	0	1	0	0
$r1'_{i,c}, (rq'_i = 0,8)$	0,64	0	0,16	0	0,08	0,72	0	0,08	0	0

TAB. 4.3 : Exemple du niveau de maîtrise de la connaissance requis par la tâche particulière

Nous représentons ci-dessous sous forme de graphique les niveaux de maîtrise dans les connaissances requis par la tâche particulière de type i ($r1'_{i,c}$).

Dans la figure ci-dessous, il est possible de distinguer trois cas selon la valeur de rq'_i .

- Si $rq'_i = 1$, (voir FIG. 4.3-coté gauche), cela signifie que la tâche particulière T'_i requiert le même niveau de maîtrise que celui de la tâche générique T_i (c'est-à-dire $r1'_{i,c} = r1_{i,c}$).

²⁵ Pour simplifier les notations, nous avons supposé ici qu'une tâche générique ne peut avoir qu'une seule occurrence de tâche particulière. Dans le cas contraire, nous devrions ajouter un autre indice représentant les différentes occurrences de tâches particulières.

- Si $rq'_i \in]0 ; 1[$, (voir FIG. 4.3-coté droit), la valeur $r1'_{i,c}$ sera inférieure à la valeur $r1_{i,c}$ en fonction de la valeur rq'_i , (ou $r1'_{i,c} = rq'_i \times r1_{i,c}$).
- Le cas $rq'_i = 0$ signifie que la tâche générique T_i n'a pas d'occurrence dans le projet.

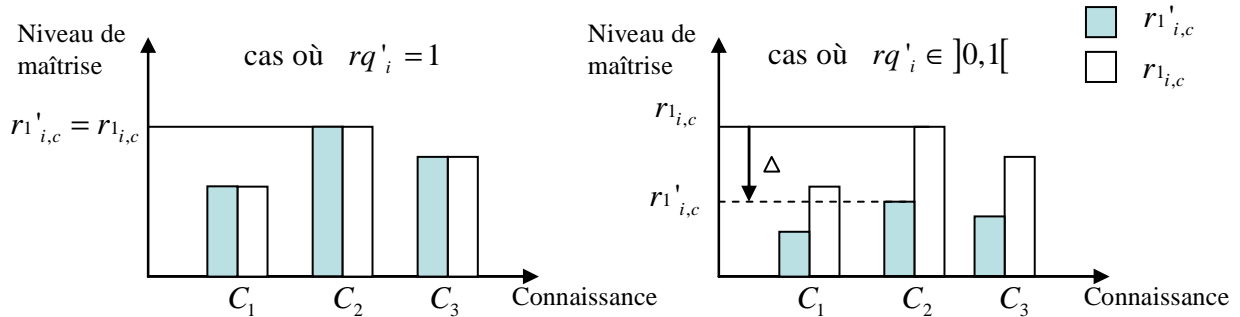


FIG. 4.3 : Niveau de maîtrise dans la connaissance de la tâche particulière

Caractérisation de l'acteur (A_j)

Les membres d'une équipe peuvent posséder plusieurs domaines de connaissances et des niveaux de maîtrise différents. De la même manière que pour la caractérisation des tâches, nous déterminons et évaluons le niveau de maîtrise de la connaissance acquis par chaque acteur ($r2_{c,j}$). Une fois les connaissances et leurs niveaux de maîtrise identifiés, nous obtenons la **MI acteur – connaissance (R2)**.

$$R2 = (r2_{c,j})_{1 \leq c \leq O, 1 \leq j \leq N}$$

Le tableau (TAB. 4.4) résume les variables de compétence présentées dans cette étape (caractérisation des tâches et des acteurs). Ces variables sont utilisées pour caractériser les tâches génériques (T_i), les tâches particulières (T'_i) et les acteurs (A_j).

Type d'entités caractérisé	Niveau de modélisation (granularité)	
	Compétence	Connaissance
Tâche générique (T_i)	Niveau requis de compétence concernant la tâche générique i (rq_i)	Niveau de maîtrise de la connaissance c requis par la tâche générique i ($r_{1_{i,c}}$)
Tâche particulière (T'_i)	Niveau requis de compétence concernant la tâche particulière de type i (rq'_i)	Niveau de maîtrise de la connaissance c requis par la tâche particulière de type i ($r'_{1_{i,c}}$)
Acteur (A_j)	Niveau de compétence relative à la tâche générique de type i acquis par l'acteur j ($r_{3_{i,j}}$)	Niveau de maîtrise de la connaissance c acquis par l'acteur j ($r_{2_{c,j}}$)
	Niveau de compétence relative à la tâche particulière de type i acquis par l'acteur j ($v_{3_{i,j}}$)	-

TAB. 4.4 : Caractérisation des tâches et des acteurs

4.2.3.2 Phase II : Calcul de l'indicateur de similarité entre tâches et acteurs

L'indicateur de similarité entre tâches et acteurs est un indicateur important dans la démarche d'affectation, permettant de sélectionner les acteurs compétents pour la réalisation des tâches. Le terme similarité utilisé dans ce document a la même signification que "matching" en anglo-saxon ou encore "compatibilité" dans certains articles.

D'une manière générale, la similarité est mesurée entre deux objets de même nature. Dans notre problème, nous cherchons à estimer la proximité entre une compétence requise par une tâche et une compétence acquise par un acteur. De ce fait, nous considérons que le terme "indicateur de similarité" peut être utilisé.

Il existe dans la littérature plusieurs mesures de similarité (voir §1.4.3). Ces mesures sont souvent utilisées pour calculer la distance entre deux ensembles. Toutefois, elles ne font pas la différence entre les écarts négatifs et les écarts positifs. On constate rapidement que les écarts négatifs compensent les écarts positifs. Ce mélange entre les écarts positifs et négatifs rend donc moins claire l'interprétation du calcul de similarité. Nous avons donc recherché une méthode de calcul de similarité palliant cet inconvénient.

Nous proposons de définir un indicateur de similarité à travers l'ensemble de connaissances caractérisant à la fois la tâche (compétence requise) et l'acteur (compétence acquise). Deux cas peuvent être distingués pour caractériser l'acteur au niveau des "connaissances" : le cas de sur-maîtrise et le cas de sous-maîtrise du type de connaissance concerné.

- La sur-maîtrise est le cas où le niveau de maîtrise dans la connaissance acquis par l'acteur est supérieur à celui requis par la tâche ($r_{2_{c,j}} > r_{1_{i,c}}$). Ceci donne les écarts positifs pour la connaissance considérée.
- La sous-maîtrise est le cas où le niveau de maîtrise dans la connaissance acquis par l'acteur est inférieur à celui requis par la tâche ($r_{2_{c,j}} < r_{1_{i,c}}$). Ceci donne les écarts négatifs pour la connaissance considérée.

Nous considérons que les écarts négatifs d'une connaissance ne doivent pas être compensés par un excès de maîtrise d'autres connaissances. C'est pourquoi nous proposons un indicateur

de similarité ne prenant pas en compte les excès. La somme des écarts en cas de sous-maîtrise des connaissances est présentée dans l'équation (Eq- 4-2).

$$\sum_{c=1}^o \max(0, r_{1,c} - r_{2,c,j})$$

Eq- 4-2

L'indicateur de similarité ($r_{3,i,j}$) entre la tâche de type i et l'acteur j est présentée dans l'équation (Eq- 4-3). Ce choix revient à supposer que les personnes qui ont un niveau de maîtrise dans la connaissance supérieur au niveau requis obtiendront le même résultat. La valeur de cet indicateur varie entre 0 et 1.

$$r_{3,i,j} = 1 - \frac{\sum_{c=1}^o \max(0, r_{1,c} - r_{2,c,j})}{\sum_{c=1}^o r_{1,c}}$$

$$0 \leq r_{3,i,j} \leq 1$$

Eq- 4-3

Cet indicateur peut être calculé pour la tâche générique ou la tâche particulière. Avec la tâche générique, nous définissons une variable $r_{3,i,j}$ représentant le niveau estimé de compétence de l'acteur j pour la réalisation de la tâche générique i . Nous appelons la matrice contenant ces niveaux de compétence **matrice de compétence (R3)**.

Avec la tâche particulière, nous définissons une variable $v_{3,i,j}$ permettant de déterminer si l'acteur j est compétent pour la "tâche particulière" T'_i . Deux cas peuvent être distingués pour caractériser l'acteur au niveau de "compétence" ($v_{3,i,j}$) : le cas où l'acteur est compétent et le cas où l'acteur est sous-compétent.

- La compétence (1) est le cas où le niveau de maîtrise acquis par l'acteur A_j ($r_{2,c,j}$) est supérieur ou égal au niveau de maîtrise requis par la tâche particulière T'_i ($r_{1',i,c}$) dans toutes les connaissances. Alors l'acteur A_j est compétent pour la tâche T'_i . Son niveau de compétence concernant cette tâche est égal à 1.

$$v_{3,i,j} = 1 \quad \text{si} \quad \sum_{c=1}^o \max(0, r_{1',i,c} - r_{2,c,j}) = 0$$

- La sous-compétence (2) est le cas où il y a au moins une connaissance C_c dans laquelle le niveau de maîtrise acquis par l'acteur A_j ($r_{2,c,j}$) est inférieur au niveau de maîtrise requis par la tâche T'_i ($r_{1',i,c}$). L'acteur est considéré sous-compétent pour cette tâche. Son niveau de compétence concernant la tâche i est donc compris entre 0 et 1.

$$v_{3,i,j} \in [0,1[\quad \text{si} \quad \sum_{c=1}^o \max(0, r_{1',i,c} - r_{2,c,j}) > 0$$

Exemple de calcul de la similarité entre tâches et acteurs

Nous présentons deux exemples concernant la tâche générique et la tâche particulière.

Exemple 1 : tâche générique

Dans cet exemple, nous considérons la tâche générique de type "1", son niveau de compétence requis est (obligatoirement) à 1 ($rq_1 = 1$). Le tableau ci-dessous présente le calcul de similarité entre cette tâche générique et deux acteurs "1" et "2".

	Niveau de maîtrise	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	Indicateur de similarité ($r_{3,i,j}$)
$T_1 (rq_1 = 1)$	$r_{1,c}$	0,8	0	0,2	0	0,1	0,9	0	1	0	0	
A_1	$r_{2,c,1}$	0,7	0	0	0,3	0	0,7	0	0,6	0	0	$r_{3,1,1} = 0,67$
A_2	$r_{2,c,2}$	0,7	0,3	0,2	0	0,1	0,8	0	0,9	0	0,5	$r_{3,1,2} = 0,90$

TAB. 4.5 : Indicateurs de similarité entre la tâche générique et les acteurs

Nous constatons que les deux acteurs ont des niveaux de compétence différents pour réaliser la tâche générique de type "1" puisque leur indicateur de similarité $r_{3,i,j}$ est à 0,67 pour l'acteur 1 et à 0,90 pour l'acteur 2.

Exemple 2 : tâche particulière

Dans cet exemple, nous considérons la tâche particulière de type "1", le niveau de compétence requis est à 0,8. Le tableau ci-dessous présente le calcul de similarité entre cette tâche particulière et deux acteurs "1" et "2".

	Niveau de maîtrise	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	Indicateur de similarité ($v_{3,i,j}$)
$T'_1 (rq'_i = 0,8)$	$r'_{1,c}$	0,64	0	0,16	0	0,08	0,72	0	0,8	0	0	
A_1	$r'_{2,c,1}$	0,7	0	0	0,3	0	0,7	0	0,6	0	0	$v_{3,1,1} = 0,81$
A_2	$r'_{2,c,2}$	0,7	0,3	0,2	0	0,1	0,8	0	0,9	0	0,5	$v_{3,1,2} = 1$

TAB. 4.6 : Indicateur de similarité entre la tâche particulière et les acteurs

Le fait de diminuer le niveau de compétence requis par la tâche de type i ($rq'_i = 0,8$) diminue directement le niveau de maîtrise requis dans la connaissance ($r'_{1,c}$), ce qui permet à l'un des acteurs d'être compétents pour la tâche particulière. Dans ce cas, l'acteur numéro "2" s'avère être compétent pour cette tâche particulière ($v_{3,1,2} = 1$).

Notons que le fait de diminuer la valeur de rq_i augmente la valeur de $v_{3_{i,j}}$ de façon non linéaire. Par exemple, si rq'_1 passe de 1 à 0,8, l'indicateur de similarité $v_{3_{i,j}}$ de l'acteur "1" augmente de 0,67 à 0,81. Cette augmentation n'est pas linéaire puisque $0,81 \neq 0,67 / 0,8$.

En résumé, l'indicateur de similarité proposé nécessite au préalable l'estimation des niveaux de maîtrise des connaissances pour chaque couple tâche-acteur. Ces indicateurs ($r_{3_{i,j}}$, $v_{3_{i,j}}$) sont importants car ils représentent la compatibilité entre tâche et acteur et ils seront utilisés par la suite dans l'affectation des tâches à des ressources.

4.2.3.3 Phase III : Recherche d'une solution optimale

L'objectif de cette dernière étape est d'affecter des tâches aux acteurs en tenant compte de surcoûts lié au cas de sous-compétence. Le résultat attendu est une solution d'affectation. Nous allons proposer un modèle mathématique qui permet de formuler un problème de type programmation linéaire en nombres réels, visant à minimiser ces surcoûts. Ce modèle est présenté en détail dans la partie qui suit (§4.2.4).

4.2.4 Formulation mathématique et recherche d'une solution optimale

Le modèle suppose que:

- les acteurs novices ayant un niveau de compétence inférieur au niveau requis par la tâche pourront être alloués à la tâche considérée. Cependant, cela engendre un surcoût qui sera supposé dépendant du niveau de sous-compétence (durée de réalisation supplémentaire, sollicitation d'un expert pour des demandes d'informations, conseils ou de tutorat).
- chaque tâche est supposée réussie (pas d'échec possible).
- il n'y a pas de contraintes de capacité pour les tuteurs (toutefois, cette contrainte ne serait pas difficile à formaliser).
- les capacités des acteurs sont estimées en unité de temps selon le contexte d'application (jour, semaine, mois, etc.). La capacité est supposée exclure les durées d'absence en raison d'une indisponibilité prévue (congés), imprévue (maladie) et éventuellement l'indisponibilité des employés en raison de contraintes légales ou sociales liées aux durées de travail ou de repos imposées.

Dans la suite de cette partie, nous présentons les éléments permettant de construire le modèle mathématique global :

- Coefficient correcteur,
- Temps de travail supplémentaire,
- Critères étudiés,
- Modèle mathématique.

4.2.4.1 Coefficient correcteur

Par analogie avec les coefficients correcteurs appliqués en gestion de production pour des opérations de montage assurées par des débutants, nous pensons que la durée de réalisation

d'une tâche doit être corrigée en fonction du niveau de compétence ($v_{3_{i,j}}$) de l'acteur. Les acteurs sous-compétents passent généralement plus de temps que les acteurs compétents. C'est pourquoi que nous avons défini un coefficient correcteur qui est fonction de $v_{3_{i,j}}$. Ce coefficient permet d'estimer la durée corrigée de réalisation d'une tâche. Il est présenté dans l'équation (Eq- 4-4).

$$\gamma_{i,j} = 1 + const \times (1 - v_{3_{i,j}})$$

Eq- 4-4

La variable *const* est une constante, permettant de paramétrer l'échelle des valeurs du coefficient correcteur. On remarque que le choix de la valeur *const* peut être délicat et peut dépendre de la difficulté d'apprentissage de nouvelles compétences. Une valeur de *const* trop élevé peut empêcher l'affectation d'acteurs sous-compétents. Nous faisons l'hypothèse qu'un acteur sous-compétent va solliciter un expert (tuteur) pour lui demande des conseils. Cela nécessite un temps du travail supplémentaire et un temps d'accompagnement par un tuteur que nous considérons comme des surcoûts liés à la sous-compétence d'un acteur.

Pour faciliter l'interprétation du coefficient correcteur qui sera expliquée dans la partie suivante, nous avons fixé la valeur de *const* égale à 1, alors

$$\gamma_{i,j} = 2 - v_{3_{i,j}}, \quad const = 1$$

$$\gamma_{i,j} = \begin{cases} 1 & \text{si } v_{3_{i,j}} = 1 \\]1,2] & \text{si } 0 \leq v_{3_{i,j}} < 1 \end{cases}$$

Eq- 4-5

Selon les équations ci-dessus, la valeur du coefficient correcteur appartient à l'intervalle $1 \leq \gamma_{i,j} \leq 2$ et deux cas peuvent être envisagés :

- **Le cas des acteurs compétents** : Si $v_{3_{i,j}} = 1$, l'acteur est compétent pour la tâche particulière, le coefficient correcteur est égal à 1 (pas pénalité). Cela signifie que les acteurs compétents n'ont pas besoin de temps de travail supplémentaire et ils n'ont pas besoin de l'aide de tuteurs pour réaliser la tâche *i*. Leur temps de travail correspond à la durée nominale.
- **Le cas des acteurs sous-compétents** : Si le niveau de compétence $v_{3_{i,j}}$ est compris entre 0 et 1, la valeur du coefficient correcteur est comprise entre $]1,2]$ (nous avons choisi une pénalité allant du simple au double). C'est le cas où l'acteur n'est pas qualifié parfaitement pour toutes les connaissances requises par la tâche. Les acteurs sous-compétents utilisent plus de temps de travail. Par ailleurs, nous supposons que pour réussir leur tâche, ils requièrent un accompagnement qui dépendra également de $\gamma_{i,j}$.

4.2.4.2 Temps de travail supplémentaire

La figure (FIG. 4.4) présente les temps liés à la réalisation d'une tâche. Nous distinguons trois types de temps : temps de travail théorique (ou nominal), temps de travail de l'acteur et temps de formation effectué par le tuteur. Ces temps sont calculés en fonction du coefficient correcteur.

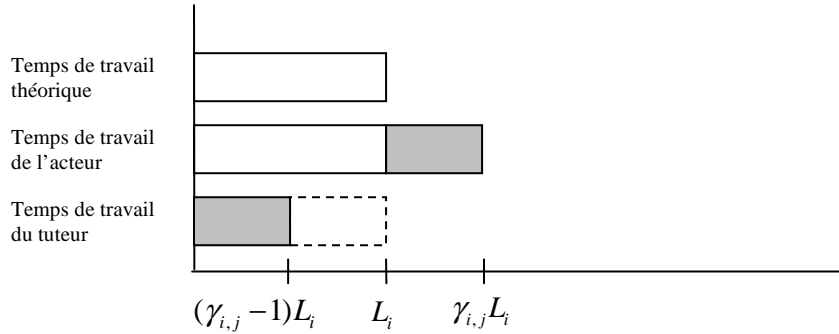


FIG. 4.4 : Temps liés à la réalisation d'une tâche

- Temps de travail théorique L_i .
- Temps de formation effectué par le tuteur : nous faisons l'hypothèse que le temps d'accompagnement de l'acteur j par le tuteur dans la réalisation de la tâche i est $(\gamma_{i,j} - 1)L_i$.
- Temps de travail effectué par l'acteur : nous faisons l'hypothèse que le temps de réalisation d'une tâche par un acteur est la somme de deux temps ci-dessus : le temps de travail de travail théorique et le temps de formation effectué avec/par le tuteur. Ce temps est égal à $L_i + (\gamma_{i,j} - 1)L_i = (\gamma_{i,j}L_i)$.

Le tableau ci-dessous récapitule les temps de travail pris en compte selon le type d'acteur.

	Temps de travail de l'acteur	Temps de formation effectué par le tuteur
Acteur compétent ($v_{3i,j} = 1 ; \gamma_{i,j} = 1$)	L_i	0
Acteur sous-compétent ($0 \leq v_{3i,j} < 1 ; 1 < \gamma_{i,j} \leq 2$)	$\gamma_{i,j}L_i$	$(\gamma_{i,j} - 1)L_i$

TAB. 4.7 : Temps de travail considérés

Notons qu'un acteur "incompétent" ($\gamma_{i,j} = 2$) est un acteur qui n'est pas du tout compétent pour la tâche ($v_{3i,j} = 0$), cet acteur est obligé de travailler pendant la durée $\gamma L_i = 2L_i$ (deux fois plus de temps que la durée théorique) pour accomplir la tâche. On fait l'hypothèse qu'il devra observer un tuteur réalisant la tâche (durée L_i) et ensuite réaliser la tâche seul (durée L_i). En plus, il y a le temps de formation effectué par le tuteur, $(\gamma_{i,j} - 1)L_i = L_i$.

4.2.4.3 Modèle mathématique

Données du problème

$\gamma_{i,j}$	Coefficient correcteur
LM_j	Charge maximale que peut réaliser l'acteur j .
L_i	Temps théorique attribué pour réaliser la tâche i
S_j	Taux de salaire de l'acteur j , (euros par unité de temps).
ST	Taux de salaire des tuteurs (euros par unité de temps)

Variables de décision

$x_{i,j}$	Variable de décision en nombres réels, $x_{i,j} \in R^+$ $x_{i,j} \in]0,1]$ si la tâche i est affectée à l'acteur j $x_{i,j} = 0$, sinon.
-----------	---

Critères étudiés

Nous nous intéressons d'abord à un critère lié aux niveaux de compétences des acteurs affectés aux tâches. Ce critère est basé sur les temps de travail supplémentaires présentés précédemment. Il vise à :

- Minimiser le coût provenant du temps de travail effectué par l'acteur : nous supposons que ce coût est le produit de la durée estimée du travail en cas de sous-compétence de l'acteur ($\gamma_{i,j} \times L_i$), le taux de salaire de l'acteur (S_j) et de la contribution de l'acteur j ($x_{i,j}$). La fonction objectif correspondant à ce critère s'écrit comme suit.

$$\text{Minimiser } \sum_{i=1}^M \sum_{j=1}^N (\gamma_{i,j} \times L_i) \times S_j \times x_{i,j}$$

Eq- 4-6

- Minimiser le coût provenant de la participation du tuteur : nous supposons que ce coût est représentée par le produit de la durée estimée du travail du tuteur $(\gamma_{i,j} - 1)L_i$, le taux de salaire du tuteur (ST) et de la contribution de l'acteur j à la tâche i ($x_{i,j}$). La fonction objectif correspondant à ce critère s'écrit comme suit.

$$\text{Minimiser } \left[\sum_{i=1}^M \sum_{j=1}^N (\gamma_{i,j} - 1) \times L_i \times ST \times x_{i,j} \right]$$

Eq- 4-7

Modèle

La fonction objectif et les contraintes du problème d'affectation mono-période sont présentées ci-dessous.

$$\text{Minimiser } \sum_{i=1}^M \sum_{j=1}^N \left[\gamma_{i,j} L_i S_j + (\gamma_{i,j} - 1) L_i ST \right] x_{i,j}$$

avec les contraintes :

$$\sum_{j=1}^N L_i x_{i,j} = L_i, \forall i = 1, \dots, M, \quad (1)$$

$$\sum_{i=1}^M \gamma_{i,j} L_i x_{i,j} \leq LM_j, \forall j = 1, \dots, N, \quad (2)$$

$$0 \leq x_{i,j} \leq 1, \forall i = 1, \dots, M, \forall j = 1, \dots, N, x_{i,j} \in R^+ \quad (3)$$

$$1 \leq \gamma_{i,j} \leq 2 \quad (4)$$

La fonction objectif décrit la minimisation des coûts d'affectation provenant des temps de travail passés par les acteurs et des temps de formation passé par les tuteurs.

La contrainte (1) assure que toutes les tâches seront affectées. La contrainte (2) assure que tous les acteurs n'ont pas été affectés au-delà de leur capacité. La contrainte (3) garantit que les variables de décision sont des variables réelles. La contrainte (4) définit la valeur d'intervalle du coefficient correcteur.

Ce modèle est un problème de programmation linéaire à variables réelles. Pour le résoudre, nous avons utilisé le solveur de programmation linéaire que propose le toolbox d'optimisation de Matlab (fonction linprog). Ce programme est présenté dans l'annexe 2.

Une limite de cette modélisation provient au cas où plusieurs acteurs sont affectés pour travailler sur une tâche, nous n'ajoutons pas de temps de coordination (ou coût de communication). On pourrait limiter le nombre d'acteurs contribuant à une tâche en fixant une borne à $x_{i,j}$, si $x_{i,j} \neq 0$.

4.2.5 Présentation d'un cas d'étude

Dans cette partie, nous présentons un exemple sur lequel l'ensemble des démarches que nous proposons dans ce mémoire a été appliqué. Le cas est inspiré d'un cas réel d'un projet européen consacré à la conception de micro-produit dans le domaine médical. Les partenaires sont spécialisés en micro et nano technologies. Ils se sont regroupés en 2006 dans le but de concevoir une capsule endoscopique pour la thérapie de la tumeur intestinale.

Tâches génériques

Nous avons identifié les tâches génériques à partir des Work Packages (module capteur, module vision, module traitement d'image, etc.). Le tableau ci-dessous présente les 15 tâches génériques retenues, dont les tâches (T1-T4) sont des tâches liées au management et à l'ingénierie système (étude de faisabilité, intégration du système) et les tâches (T5-T15) sont des tâches de conception de modules. La durée des tâches génériques est également estimée.

Les tâches de conception peuvent être classées en deux catégories différentes :

- tâche innovante (par exemple, concevoir un module de capteur micro-optique) : le niveau de compétence requis par ce type de tâche est proche de 1 (par exemple, 0,9).
- tâche routinière (par exemple, concevoir un module de circuit intégré). le niveau de compétence requis est peu élevé (par exemple, 0,5).

	Durée nominale (mois)	Tâche générique
T1	2	Définir les spécifications du système (étude de marché, étude de faisabilité, définition des exigences)
T2	1	Concevoir l'architecture du système (architecture modulaire, familles du produit)
T3	1	Intégrer les sous-systèmes et valider l'opérabilité du système (interfaces mécaniques, électroniques, thermiques ou biologiques)
T4	1,5	Management & coordination
T5	4	Concevoir le "capteur pour le diagnostic" (exemple : capteur biochimique)
T6	3	Concevoir le module de "vision" (certaines fonctions attendues : caméra et illumination, analyse des images, compression des données, nettoyage des lentilles, analyse des signaux)
T7	3	Concevoir le module de "traitement et manipulation des tissus" (exemple : dispositif mécanique pour le traitement du tissu et le système de distribution des médicaments)
T8	2	Concevoir le module de "locomotion" (exemple : actionneurs, accéléromètre électromagnétique)
T9	1	Concevoir le circuit intégré
T10	2	Concevoir le module de "contrôle commande"
T11	2	Concevoir le module de "stockage des énergies" (exemple : système sans fils, électromagnétique)
T12	1	Concevoir le module de "communication sans fils et télémétrie" (exemple : transmission des données bidirectionnelle)
T13	2	Concevoir le module de "navigation à distance"
T14	4	Concevoir le module de "capteur nano-biotechnologique"
T15	4	Concevoir le module de "capteur micro-optique"

TAB. 4.8: Liste des tâches génériques

Connaissances

Les types de connaissances nécessaires à la conception du produit sont rassemblés dans le tableau suivant. Nous pouvons classer ces 23 types de connaissances par leur nature en trois catégories :

- connaissances techniques liées à des connaissances disciplinaires (électronique, mécanique, traitement du signal (vision), etc.) (C1-C11),
- connaissances pratiques concernant l'utilisation des outils (modélisation et simulation, programmation, etc.) (C12-C18),
- connaissances méthodologiques liées au management et à l'ingénierie système (C19-C23).

	Connaissance (Maîtrise des méthodes et des outils)
C1	Acoustique, Vibration, Propagation d'ondes
C2	Mécanique, Cinématique (mécanique du mouvement)
C3	Thermodynamique, Transfert thermique, Conversion énergie
C4	Mécanique des fluides, Dynamique des fluides
C5	Matériaux
C6	Electromagnétisme
C 7	Electronique, Electrochimique
C 8	Théorie contrôle commande/ Automatique
C 9	Traitement image, Traitement signal, Télécommunication
C 10	Optique et Vision
C 11	Biomédecine (Biochimie)
C 12	Informatique
C 13	CAD, CAM
C 14	LabVIEW, SPICE
C 15	Matlab/Simulink/Dspace
C 16	Modélisation des contraintes mécaniques, Méthodes des éléments finis
C 17	Modélisation et Simulation (UML, IDEF0, SysML)
C 18	Comsol (simulation multi physique)
C 19	Optimisation, Méthodes d'aide à la décision
C 20	Sûreté fonctionnement (fiabilité, maintenabilité, sécurité)
C 21	Ingénierie système
C 22	Management de projet (qualité, délais, coût, risque)
C 23	Marketing

TAB. 4.9: Liste des connaissances (requis)

Profils des acteurs et taux de salaire

Dans le tableau suivant (TAB. 4.10), nous définissons une liste de 20 acteurs, leurs profils de compétence, leurs niveaux d'expertise et les taux de salaire.

Les profils de compétence sont les métiers nécessaires dans la réalisation du projet. Nous distinguons deux catégories de profils : profils lié à l'ingénierie système (chef de projet, architecte, intégrateur) et profils lié aux compétences disciplinaires (mécanique, électronique, automatique, etc.). Les acteurs sont aussi classés selon le niveau d'expertise (expert, confirmé, débutant), qui est lié aux niveaux de maîtrise des connaissances (MI acteur-connaissance présentée dans (TAB. 4.12)).

Les taux de salaire par mois de chaque acteur varient en fonction du niveau d'expertise : 2000 euros pour les débutants, 3000 euros pour les confirmés et 4000 euros pour les experts. Le taux de salaire du tuteur est fixé à 4000 euros par mois.

Acteur	Abréviation	Profils	Taux de salaire
1	SYS	Analyse produit/marketing ou chef de projet débutant	2000
2	SYS	Architecte système ou intégration/validation	4000
3	SYS	Architecte système ou intégration/validation	4000
4	EE	Electronicien (1) expert	4000
5	EE	Electronicien (2) expert	4000
6	EE	Electronicien (3) confirmé	3000
7	EE	Electronicien (4) confirmé	3000
8	EE	Electronicien (5) débutant	2000
9	ME	Mécanicien (1) expert	4000
10	ME	Mécanicien (2) confirmé	3000
11	ME	Mécanicien (3) débutant	2000
12	MT	Mécatronicien (1) expert	4000
13	MT	Mécatronicien (2) confirmé	3000
14	MT	Mécatronicien (3) confirmé	3000
15	MT	Mécatronicien (4) débutant	2000
16	AU	Automaticien vision (1) confirmé	3000
17	AU	Automaticien-vision (2) confirmé	3000
18	AU	Automaticien-vision (3) débutant	2000
19	SYS	Chef de projet ou architecte (expert)	4000
20	SYS	Chef de projet ou intégrateur (confirmé)	3000

TAB. 4.10: Acteurs et taux de salaire

Caractérisation des tâches et des acteurs

Ensuite, nous évaluons les connaissances nécessaires pour créer la MI tâche-connaissance (TAB. 4.11) et identifions les connaissances possédées par chaque acteur pour créer la MI acteur-connaissance (TAB. 4.12).

La construction de la matrice tâche-discipline est dictée par les constatations suivantes :

- Nous adoptons une caractérisation numérique du niveau de maîtrise des connaissances nécessaires pour chaque tâche. Nous estimons la valeur du niveau de maîtrise pour chaque connaissance qui est un nombre réel compris entre 0 et 1. Si le niveau est fort alors la valeur est proche de 1, dans le cas contraire elle est proche de 0.

- Les tâches de type d'ingénierie système requièrent un ensemble de connaissances transversales (mécanique, matériaux, électronique, biomédecine, etc.). Les niveaux de maîtrise dans ces connaissances peuvent être différents selon la complexité du projet.
- Les connaissances concernant l'architecture système (D21, D22) sont des connaissances transversales. Chaque tâche générique requiert ce type de connaissances avec un niveau requis relativement différent. Les modules de conception qui ont beaucoup d'interactions avec les autres modules requièrent un niveau de maîtrise dans ce type de connaissances plus élevé.

Nous estimons les valeurs du niveau de maîtrise requis par les tâches qui sont comprises entre 0 et 1. Cependant, nous sommes conscients que la détermination de ces niveaux est délicate car ils sont subjectifs et relatifs aux exigences de conception qui peuvent être différentes d'un projet à l'autre. En entreprise, leur détermination pourrait être discutée entre le chef de projets, les directeurs des métiers de la conception et le directeur des programmes.

La construction de la matrice acteur-discipline est dictée par les constatations suivantes :

- Nous adoptons une caractérisation numérique pour évaluer les niveaux de maîtrise des connaissances, acquis par chaque acteur. La valeur est définie en nombres réels, comprise entre 0 et 1. Si le niveau est fort alors la valeur est proche de 1, dans le cas contraire elle est proche de 0.
- Les acteurs sont caractérisés par leur profil (électronicien, mécatronicien, automaticien, intégrateur etc.). Les acteurs partageant le même profil possèdent plus ou moins les mêmes types de connaissances avec des valeurs différentes dépendant de leur niveau global. Nous avons défini les connaissances appartenant à chaque profil présentés ci-dessus.
 - Profil d'architecture système (ou chef de projet) : nécessite des connaissances en ingénierie système (C17-C23).
 - Profil d'électronicien : nécessite des connaissances en électronique (C6-C9, C14-C15).
 - Profil de mécanicien : nécessite des connaissances en mécanique (C1-C5, C13, C15-C16).
 - Profil de mécatronicien : nécessite des connaissances en mécanique, électronique et automatique (C2-C3, C5-C8, C13-C15).
 - Profil d'automaticien : nécessite des connaissances en automatique et vision (C7-C10, C12-C13, C15).
- Les connaissances en ingénierie système sont des connaissances de base des concepteurs. Elles sont présentes dans tous les profils, mais le niveau de maîtrise est inférieur à celui acquis par les acteurs "architecte système".
- Nous n'avons pas fixé de règles pour déterminer le niveau de maîtrise dans la connaissance pour chaque niveau d'expertise des acteurs (expert, confirmé, débutant). Nous avons essayé de maintenir les valeurs pour chaque niveau d'expertise sur le même intervalle.
- L'évaluation des niveaux de connaissances des acteurs est aussi délicate et peut se faire de trois façons complémentaires : un test de connaissances de type quiz, une auto-évaluation par l'acteur, une évaluation par l'expert qui a accompagné l'acteur au cours de la réalisation de la tâche.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
T1											0,4									0,2	0,6	0,4	1
T2		0,4			0,4		0,4	0,8			0,4		0,4		0,4		0,4	0,6		0,8	1	0,4	0,4
T3		0,2			0,2		0,8	0,8			0,6									0,8	1	0,4	
T4											0,4								0,6	0,6	0,8	1	0,2
T5							0,8	0,2			0,6			0,6								0,2	0,2
T6								0,8	0,6	1					0,8							0,4	0,4
T7		0,6			0,8			0,6			0,4		0,4		0,8							0,4	0,4
T8		1			0,6	0,6		0,2					0,8									0,4	0,4
T9						0,6	0,8	0,4						0,8								0,2	0,2
T10							0,6	1							0,8			0,6		0,8	0,4	0,4	
T11			0,6		0,4	1	0,6				0,6							0,6				0,2	0,2
T12								0,4	1			0,4										0,2	0,2
T13							0,2	0,6	1			0,6			0,6							0,4	0,4
T14					0,2		0,8	0,6			0,6				0,8							0,2	0,2
T15							0,6	0,6		1				0,8	0,6							0,2	0,2

TAB. 4.11 : MI tâche-connaissance

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
A1											0,2									0,4	0,6	0,6	0,8
A2		0,4			0,4	0	0,4	0,6			0,2		0,4		0,4		0,4	0,4		0,4	0,8	0,6	0,4
A3		0,2					0,2	0,4			0,2		0,2		0,2				0,2	0,4	0,4	0,4	0,2
A4					0,4	0,8	1	0,2	0,6		0,4			0,4	0,4							0,4	0,6
A5					0,4	0,6	0,8	0,6			0,4			0,8	0,6							0,4	0,6
A6						0,6	0,6	0,4			0,4		0,4	0,6	0,4							0,4	0,4
A7						0,4	0,6	0,4	0,4			0,4		0,6	0,2							0,4	0,4
A8						0,4	0,4	0,4	0,4			0,4		0,4	0,4							0,2	0,2
A9	0,8	0,8	0,6	0,6	0,6						0,4		0,8		0,8	0,8						0,4	0,6
A10	0,4	0,6	0,6	0,6	0,4	0,4					0,2		0,4		0,4	0,6						0,4	0,4
A11	0,4	0,4	0,4	0,4	0,4								0,4		0,2	0,4							0,4
A12		0,8	0,2		0,4	0,6	0,8	0,8			0,4		0,6		0,8			0,4		0,4	0,6	0,4	
A13		0,6	0,2		0,4	0,4	0,6	0,8		0,4	0,2		0,4	0,6	0,4			0,4		0,2	0,4	0,6	
A14		0,6			0,2	0,4	0,2	0,6		0,4		0,2	0,6	0,4	0,6			0,6		0,2	0,4	0,2	
A15		0,4			0,2	0,2	0,4	0,4				0,4	0,4		0,6			0,2	0,4	0,2		0,4	
A16							0,4	0,8	0,8	0,6	0,4	0,2			0,8							0,4	0,6
A17							0,4	0,8	0,6	0,6		0,4	0,4		0,8			0,4		0,4	0,4	0,6	
A18		0,4					0,4	0,4	0,4	0,4		0,4	0,4	0,2	0,6			0,2		0,4	0,2	0,2	
A19		0,4			0,4		0,4	0,6			0,4		0,6		0,4		0,4	0,6	0,4	0,4	0,8	1	0,4
A20		0,4			0,4		0,4	0,6			0,4		0,4		0,4		0,4	0,4	0,6	0,6	0,8	0,8	0,2

TAB. 4.12: MI acteur-connaissance

4.2.6 Test du principe

En utilisant le jeu de données extrait du projet de conception d'une capsule endoscopique présenté précédemment, nous cherchons ici à tester la démarche d'affectation mono-période proposée.

Nous ajoutons ici quelques informations propres à l'affectation mono-période.

- la charge induite par la tâche particulière (L_i) est donnée dans le tableau suivant.

(mois)	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
Charge (L_i)	2	2	1	1	2	1	2	1	1	2	2	1	2	1	2

TAB. 4.13 : Charge induite par la tâche

- la capacité maximale pour chaque acteur pendant la réalisation de l'ensemble de tâches du projet est égale à 3,5 mois ($LM_j = 3,5, \forall j = 1, \dots, N$). Cette valeur tient compte des indisponibilités d'un acteur sur le projet.
- nous rappelons les taux de salaire que nous avons fixés dans le tableau suivant (TAB. 4.14). Le taux de salaire du tuteur est fixé à 4000 euros pour tous les tuteurs. Le taux salaire des acteurs est donné dans le tableau suivant (2000€ pour un débutant, 3000€ pour un confirmé et 4000€ pour un expert).

Acteur	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Niveau ²⁶	D	C	D	E	E	C	C	D	E	C	D	E	C	C	D	C	C	D	E	C
Taux salaire (€)	2000	4000	4000	4000	4000	3000	3000	2000	4000	3000	2000	4000	3000	3000	2000	3000	3000	2000	4000	3000

TAB. 4.14 : Taux de salaire des acteurs

En fixant ($rq'_i = 1$) pour toutes les tâches particulières ($\forall i = 1, \dots, M$), nous avons obtenu l'indicateur de similarité entre tâches et acteurs (TAB. 4.15-à gauche) et les coefficients correcteurs (TAB. 4.15-à droite).

²⁶ Débutant (D), Confirmé (C), Expert (E)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0,8	0,7	0,7	0,5	0,5	0,2	0,0	0,3	0,5	0,1	0,2	0,6	0,5	0,3	0,2	0,5	0,2	0,4	0,8	0,7	T1	1,2	1,3	1,5	1,5	1,5	1,5	1,7	1,8	1,5	1,6	1,8	1,4	1,5	1,7	1,8	1,5	1,6	1,8	1,2	1,3
T2	0,3	0,8	0,6	0,4	0,4	0,3	0,1	0,3	0,3	0,2	0,2	0,8	0,6	0,5	0,4	0,4	0,5	0,4	0,8	0,8	T2	1,7	1,2	1,6	1,6	1,6	1,6	1,7	1,8	1,6	1,6	1,7	1,3	1,4	1,5	1,6	1,6	1,5	1,6	1,1	1,1
T3	0,3	0,7	0,7	0,5	0,6	0,3	0,2	0,3	0,3	0,1	0,2	0,8	0,6	0,4	0,4	0,5	0,5	0,4	0,7	0,8	T3	1,7	1,3	1,5	1,5	1,4	1,5	1,6	1,8	1,7	1,7	1,8	1,2	1,4	1,6	1,6	1,5	1,5	1,6	1,3	1,3
T4	0,6	0,6	0,8	0,4	0,4	0,1	0,0	0,2	0,4	0,1	0,1	0,5	0,4	0,2	0,3	0,4	0,3	0,3	0,9	0,9	T4	1,4	1,4	1,5	1,6	1,6	1,7	1,8	1,9	1,6	1,7	1,9	1,5	1,6	1,8	1,7	1,6	1,6	1,8	1,1	1,1
T5	0,2	0,5	0,5	0,9	0,9	0,7	0,5	0,5	0,3	0,1	0,1	0,7	0,8	0,5	0,3	0,5	0,3	0,2	0,5	0,5	T5	1,8	1,5	1,6	1,2	1,1	1,2	1,3	1,5	1,7	1,8	1,9	1,3	1,2	1,5	1,7	1,5	1,6	1,5	1,5	1,5
T6	0,2	0,5	0,5	0,5	0,5	0,2	0,2	0,5	0,2	0,0	0,1	0,6	0,6	0,6	0,4	0,9	0,8	0,7	0,5	0,5	T6	1,8	1,6	1,7	1,5	1,5	1,6	1,6	1,6	1,6	1,7	1,9	1,4	1,4	1,5	1,7	1,1	1,1	1,5	1,6	1,6
T7	0,2	0,7	0,7	0,5	0,6	0,4	0,1	0,4	0,6	0,4	0,4	0,9	0,8	0,7	0,5	0,6	0,5	0,5	0,8	0,8	T7	1,8	1,3	1,5	1,5	1,4	1,5	1,7	1,7	1,2	1,4	1,6	1,1	1,2	1,3	1,5	1,4	1,4	1,5	1,2	1,2
T8	0,2	0,6	0,5	0,5	0,5	0,3	0,1	0,4	0,8	0,5	0,4	0,9	0,7	0,7	0,5	0,3	0,3	0,4	0,6	0,6	T8	1,8	1,5	1,7	1,5	1,5	1,5	1,7	1,8	1,3	1,4	1,6	1,2	1,3	1,4	1,6	1,8	1,7	1,7	1,4	1,5
T9	0,1	0,4	0,4	0,9	1,0	0,7	0,6	0,7	0,1	0,1	0,1	0,7	0,8	0,6	0,4	0,4	0,3	0,3	0,4	0,4	T9	1,9	1,6	1,7	1,2	1,0	1,1	1,2	1,3	1,9	1,7	1,9	1,3	1,2	1,4	1,6	1,6	1,6	1,5	1,6	1,6
T10	0,3	0,7	0,6	0,4	0,6	0,3	0,3	0,4	0,2	0,0	0,1	0,8	0,7	0,6	0,5	0,6	0,7	0,5	0,7	0,7	T10	1,7	1,3	1,6	1,6	1,4	1,5	1,6	1,7	1,7	1,7	1,9	1,2	1,3	1,4	1,5	1,4	1,2	1,5	1,3	1,3
T11	0,1	0,4	0,3	0,6	0,6	0,4	0,2	0,3	0,4	0,4	0,2	0,7	0,6	0,4	0,3	0,3	0,2	0,1	0,5	0,5	T11	1,9	1,6	1,8	1,4	1,4	1,5	1,7	1,7	1,6	1,5	1,8	1,3	1,4	1,6	1,7	1,7	1,7	1,8	1,5	1,5
T12	0,2	0,4	0,4	0,5	0,4	0,2	0,5	0,7	0,2	0,0	0,1	0,4	0,4	0,5	0,5	0,8	0,7	0,7	0,4	0,4	T12	1,8	1,6	1,6	1,5	1,6	1,6	1,3	1,3	1,8	1,8	1,9	1,6	1,6	1,5	1,5	1,2	1,2	1,3	1,6	1,6
T13	0,2	0,5	0,5	0,6	0,6	0,3	0,4	0,7	0,2	0,0	0,1	0,6	0,5	0,6	0,5	0,8	0,7	0,7	0,5	0,5	T13	1,8	1,5	1,6	1,4	1,4	1,5	1,4	1,4	1,6	1,7	1,8	1,4	1,5	1,4	1,5	1,2	1,2	1,4	1,5	1,5
T14	0,2	0,6	0,6	0,7	0,9	0,5	0,4	0,5	0,3	0,1	0,1	0,9	0,7	0,6	0,5	0,8	0,6	0,4	0,7	0,7	T14	1,8	1,4	1,6	1,3	1,1	1,4	1,5	1,5	1,5	1,6	1,8	1,1	1,3	1,4	1,5	1,2	1,4	1,5	1,3	1,3
T15	0,1	0,4	0,4	0,5	0,6	0,5	0,4	0,4	0,3	0,1	0,1	0,5	0,7	0,6	0,3	0,5	0,5	0,3	0,5	0,4	T15	1,9	1,6	1,7	1,5	1,3	1,4	1,5	1,5	1,8	1,8	1,9	1,5	1,3	1,4	1,6	1,4	1,4	1,4	1,6	1,6

Matrice de compétence ($v_{3,i,j}$)

Matrice de coefficient correcteur ($\gamma_{i,j}$)

TAB. 4.15 : Matrice de compétence et matrice de coefficient correcteur

Le problème d'affectation est résolu par la programmation linéaire. A partir des données d'entrée, nous avons obtenu la matrice d'affectation (TAB. 4.16) avec un coût d'affectation total de 99K€.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T3	0	0	0	0	0	0	0	0	0	0	0	0,9	0	0	0	0	0	0	0	0,1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,3	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0,1	0,3	0	0,6	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0,7	0	0	0	0	0	0	0	0	0	0,3	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0,2	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Ch ²⁷	2,3	0	0	0	0	3,4	0	0,9	0	0	0	3,5	3,5	0	1,8	2,5	3,2	3,2	0	3,5

TAB. 4.16 : Matrice d'affectation $x_{i,j}, (r_{q_i} = 1)$

La dernière ligne de ce tableau présente la charge affectée à chaque acteur. Elle est calculée par la formule ($\sum_i \gamma_{i,j} * L_i * x_{i,j}$). Prenons l'exemple de l'acteur A17, la charge totale de cet acteur est de 3,2 mois, répartie pour la tâche T6, 0,33 mois ($1,1 * 1 * 0,3$), pour la tâche T10, 2,4

²⁷ Ch représente la charge de travail des acteurs après correction des temps nominaux

mois ($1,2*2*1$) et pour la tâche T13, 0,48 mois ($1,2*2*0,2$). Nous constatons que la capacité de travail des acteurs (3,5 mois) est globalement supérieure à la charge induite. Cependant, les charges de travail de certains acteurs atteignent la capacité maximale de chaque acteur.

Nous avons choisi une charge maximale égale à 3,5 mois (au lieu d'un chiffre entier de 4 mois) en considérant qu'une partie du temps de travail des acteurs (0,5 mois) doit rester libre pour participer à des activités hors projet. Ces valeurs sont bien sûr ajustables. Cependant, dans notre exemple, cela permet d'autoriser l'affectation de plusieurs tâches à un acteur. Notons qu'ici la capacité de travail des acteurs est globalement supérieure à la charge induite mais cela n'est pas préjudiciable pour illustrer la pertinence de la démarche.

Dans le tableau suivant (TAB. 4.17), nous avons présenté le résultat d'affectation en faisant varier le niveau de compétence des tâches particulières de ce projet (rq'_i) (TAB. 4.17-a) et le taux de salaire des acteurs (TAB. 4.17-b).

Tâche	Solutions optimales			Taux salaire (1) ²⁸	Taux salaire (2) ²⁹	Taux salaire (3) ³⁰
	$rq'_i=1$	$rq'_i=0.8$	$rq'_i=0.6$	$rq'_i=0.8$	$rq'_i=0.8$	$rq'_i=0.8$
Acteur	Acteur	Acteur	Acteur	Acteur	Acteur	Acteur
1	1	1	1	1	1	1
2	20	20	20	19, 20	20	20
3	12, 20	20	20	20	20	20
4	20	1, 20	1	20	1, 20	1
5	6	6	6	6	6	6
6	16, 17	16, 17	18	16, 17	16, 17	16,17
7	12, 13, 15	13; 15	13, 20	13	13; 15	11,15
8	12	12	13	12	12	11
9	6	6	8	6	6	6, 8
10	17	17	17	17	17	17
11	13	13	12, 13	12	13	11,13
12	8, 18	8	8, 17	16	8	8
13	16, 17	16, 17, 18	8	16, 17	16, 17, 18	8, 18
14	11	15	15	5	15	15
15	18	18	18	13, 18	18	18
Coût total (euros)	99K	85K	72K	73K	85K	99K
Charge des débutants³¹	29%	42%	46%	13%	42%	61%

(a)

(b)

TAB. 4.17: Résultats de l'affectation

Dans le coté gauche du tableau (TAB. 4.17-a), si nous choisissons ($rq'_i = 0,8$) pour toutes les tâches ($\forall i = 1, \dots, M$), nous avons obtenu une nouvelle solution d'affectation. Nous avons constaté que si nous diminuons le niveau de compétence concernant la tâche de 1 à 0,8, les débutants pouvant être choisis (de 29% à 42%), cela engendre moins de coût pour le projet (de 99K€ à 85K€).

²⁸ 2000 € pour un débutant, 2500 € pour un confirmé et 3000 € pour un expert.

²⁹ 2000 € pour un débutant, 3000 € pour un confirmé et 4000 € pour un expert.

³⁰ 2000 € pour un débutant, 4000 € pour un confirmé et 6000 € pour un expert.

³¹ La charge des débutants représente le rapport de la somme des charges affectées à des débutants sur la charge induite par toutes les tâches du projet.

Cela est expliqué par le fait que si rq'_i diminue (le projet exige un niveau de compétence plus bas), les niveaux de maîtrise dans les connaissances $r_{i,c}$ sont moins exigeants, ce qui augmente l'indicateur de similarité (le coefficient correcteur diminue). La diminution du coefficient correcteur explique la diminution du coût global. Autrement dit, si l'entreprise doit réaliser une tâche facile (bas niveau de la compétence), il y a plus de possibilités pour affecter les tâches à des ressources juste compétentes (débutants). Par exemple, la tâche T6 (module vision) est affectée à des automaticiens experts (A16 et A17) si le niveau requis de compétence est élevé et elle est affectée à un automaticien débutant (A18) lorsque le niveau requis de compétence est plus faible. Cela montre que l'indicateur de similarité utilisé pour sélectionner des acteurs donne un résultat raisonnable lors de l'affectation.

Dans le coté droit du tableau (TAB. 4.17-b), nous avons constaté que la solution d'affectation est sensible à la définition des taux de salaire des salariés (expert, confirmé, débutant) et des tuteurs. Par exemple, si les taux de salaire entre expert et débutant ne sont pas très différents, le programme va choisir les experts pour éviter le surcoût venant du débutant. S'il y a un grand écart entre les deux, le programme va choisir de former des débutants (de 13% à 61%).

4.2.7 Synthèse

Dans ce chapitre, nous avons proposé une démarche d'aide à l'affectation. Cette démarche propose de caractériser des tâches et des acteurs en fonction d'un ensemble de connaissances, de calculer un indicateur de similarité entre tâches et acteurs, et de résoudre un problème d'affectation mono-période.

Nous avons testé la démarche proposée sur un jeu de données construit à partir d'un projet de conception et nous avons mené une étude de sensibilité des résultats en fonction du niveau de compétence requis par les tâches et du taux de salaire des acteurs.

Le tableau (TAB. 4.18) donne une estimation qualitative de la satisfaction des qualités attendues.

Le modèle d'affectation proposé permet d'affecter des tâches à des ressources moins compétentes en tenant compte d'un surcoût supplémentaire. Le modèle cherche à trouver un compromis entre le choix des experts ayant un taux de salaire élevé et le choix des débutants ayant un surcoût de formation. L'indicateur de similarité prenant en compte des écarts fournit des solutions d'affectation cohérentes.

Néanmoins, l'évolution des compétences n'a pas été pris en compte, le modèle ne nous permet d'avoir une vision à long terme (par exemple, une proposition qui envisage de former des débutants au début et de les réaffecter ultérieurement aux tâches plus complexes). Cela fera l'objet de la partie suivante de ce chapitre, l'affectation multi-périodes avec évolution des compétences.

Exigences	Estimation	Limites de la validation
E1 : Affecter des tâches aux acteurs en minimisant le coût salarial lié aux niveaux de compétence	++	Test sur un cas d'école
E2 : Proposer un indicateur de similarité entre tâches et acteurs	++	
E3 : Prendre en compte de l'évolution des compétences dans l'affectation des tâches	-	Non modélisée

TAB. 4.18 : Estimation qualitative de la méthode

4.3 Affectation multi-périodes avec évolution des compétences

Dans la suite de cette partie, nous identifions d'abord les qualités attendues de la méthode proposée pour une affectation multi-périodes. Ensuite, nous formulons des hypothèses de modélisation et la construction des données supplémentaires concernant le cas multi-périodes. Nous présentons la structuration globale du modèle présentant une démarche générale pour affecter des tâches à des acteurs en considérant le cas multi-périodes et multi-projets, avec prise en compte de l'évolution des compétences. Puis, nous proposons une loi permettant de modéliser l'évolution des compétences. Par la suite, une formulation mathématique comprenant les fonctions objectifs et les contraintes liées à l'affectation seront présentées. Nous testons le principe de la méthode par un exemple d'application construit à partir de deux projets de conception de micro-produits.

4.3.1 Qualités attendues de la méthode

Les qualités attendues d'une telle méthode sont les suivantes :

- E1 : Affecter des tâches dans le cas multi-périodes et multi-projets en tenant compte d'une loi d'évolution des compétences
- E2 : Minimiser le coût salarial lié aux niveaux de compétence des acteurs
- E3 : Minimiser les pénalités financières dues à la non-satisfaction des objectifs de compétence

4.3.2 Hypothèses de modélisation et construction des données

Nous présentons dans cette partie les hypothèses de modélisation. Ensuite, nous présentons les données supplémentaires du cas d'étude qui nous permettent d'illustrer la méthode proposée.

4.3.2.1 Hypothèses de modélisation

Nous définissons un ensemble de tâches génériques (T_i), indexées par i ($i=1, \dots, M$), un ensemble de projets à réaliser, indexés par l , ($l=1, \dots, P$), un ensemble de périodes, indexées par k , ($k=1, \dots, K$), un ensemble d'acteurs (A_j), indexés par j , ($j=1, \dots, N$) et un ensemble de

connaissances (C_c), indexées par c , ($c=1, \dots, O$), un ensemble de tâches particulières ($T_i^{k,l}$), indexées par i, j, k

Comme dans le chapitre précédent, nous distinguons deux types de tâches dans la planification : les tâches génériques T_i (tâches qui couvrent une classe de tâches appelant les mêmes types de connaissances) et les tâches particulières $T_i^{k,l}$ (tâches qui sont des instances des tâches génériques et qui sont spécifiques pour une période k et un projet l)³².

Les tâches que nous allons affecter dans le cas multi-périodes sont des tâches particulières ($T_i^{k,l}$). Elles mobilisent les mêmes types de connaissances que la tâche générique T_i mais elles peuvent nécessiter un niveau de compétence requis différent. Ainsi, la tâche $T_i^{k,l}$ représente une tâche ressemblant à la tâche générique de type i , appartenant au projet l . Elle a été planifiée pour être réalisée pendant la période k . La valeur $k = 0$ représente la date initiale avant l'affectation des tâches, et la valeur $k=1$ correspond à la première période, et donc la date de la première affectation.

Nous avons considéré que le découpage du projet en différentes tâches et la planification de ces tâches (sur l'ensemble de périodes k) sont des **données d'entrée** du problème. Dit autrement, l'ensemble de tâches à affecter à chaque période k est connu. La figure (FIG. 4.5) présente un exemple de planification des tâches dans le cas de deux projets et de trois périodes.

Tâche générique (T_i)	k=1	k=2	k=3
T1	$T_1^{1,1}$	$T_1^{2,1}$	
T2	$T_2^{1,1}$	$T_2^{2,1}$	
T3 Projet ($l=1$)			$T_3^{3,1}$
T4			
T5			$T_5^{3,1}$
T6			
T7			
T1			
T2			
T3 Projet ($l=2$)	$T_3^{1,2}$	$T_3^{2,2}$	
T4			$T_4^{3,2}$
T5			
T6			$T_6^{3,2}$
T7			

FIG. 4.5 : Exemple d'une planification des tâches provenant de 2 projets sur 3 périodes

Nous supposons que les tâches planifiées sur une période n'ont pas de contraintes de précédence et peuvent se dérouler en simultané (peuvent se réaliser sur une seule période). Ceci nécessite de choisir la durée d'une période et définir des tâches, de façon appropriée. Une tâche dont la durée est supérieure à une période sera découpée sur plusieurs périodes. Dans ce

³² Avec cette notation, nous supposons qu'une tâche générique n'a qu'une seule occurrence sur une période donnée et sur un projet donné mais peut avoir plusieurs occurrences à des périodes d'affectation différentes ou sur des projets différents.

cas, nous n'avons pas imposé que l'acteur qui commence une tâche doive la poursuivre. Nous supposons que plusieurs acteurs peuvent travailler sur une même tâche. Dans ce cas, nous n'avons pas modélisé de coût supplémentaire de coordination. Nous supposons qu'un acteur peut avoir à réaliser plusieurs tâches sur la même période tant que la charge affectée ne dépasse pas la charge que l'acteur peut supporter. Nous n'avons pas modélisé un temps de travail en heures supplémentaires plus coûteuses que les heures normales.

4.3.2.2 Construction des données

Les données principales du jeu de test (concernant la liste des tâches, la liste des acteurs, la liste des connaissances, la MI tâche-connaissance et la MI acteur-connaissance) ont été déjà présentées dans la partie concernant l'affectation mono-période (§4.2.5). Dans cette partie, nous allons présenter une planification des tâches sur plusieurs périodes et sur plusieurs projets.

Pour l'affectation multi-périodes, nous avons défini deux projets semblables ($p=2$; p est le nombre de projets), structurés à partir de ce projet. Le projet 1 est un projet de capsule endoscopique. Le projet 2 concerne également la conception de capsule endoscopique, avec des technologies de capteur différentes.

Le tableau (TAB. 4.19) présente la planification des tâches de ces deux projets. Le niveau requis de compétence et la durée de réalisation de chaque tâche particulière sont également définis dans la planification. Le projet 1 se décompose en 13 tâches particulières réparties sur 3 périodes ($k=3$; k est le nombre de périodes). Le projet 2 se décompose en 10 tâches réparties également sur 3 périodes.

Comme c'est souvent le cas dans la réalité, nous avons aussi représenté le fait que les deux projets ne sont pas engagés en même temps. Ainsi, les tâches T1, T2 du projet 2 sont supposées déjà réalisées.

Un exemple de la différence entre les deux projets concerne les tâches T5 et T15. La tâche T15 consiste en la réalisation d'un capteur micro-optique, alors que la tâche T5 du projet 1 s'intéresse à la réalisation d'un capteur biochimique.

Le choix de la construction de la planification des tâches est dicté par les constatations suivantes :

- Nous avons choisi ces deux projets d'affectation multi-périodes parce que les informations connues sur ces 2 projets sont suffisantes pour modéliser le cas multi-projets. La simulation pendant 3 périodes permet de mettre en évidence et de suivre l'évolution des compétences. L'horizon d'affectation est supposé duré un an (3*4 mois). Il est cohérent avec la durée des projets considérés.
- Comme dans le cas mono-période, nous avons choisi une charge maximale égale à 3,5 mois en considérant qu'une partie du temps des acteurs est indisponible pour les projets, par exemple, ils doivent rester libres pour participer à des activités hors projet.
- Pour définir le niveau de compétence concernant la tâche ($rq_i^{k,l}$), nous considérons les hypothèses suivantes :
 - Dans un projet R&D, l'innovation ne porte que sur certaines tâches. Ainsi, on peut classer des tâches génériques en deux catégories : "tâches innovantes" et tâche d'ingénierie. Les tâches innovantes sont orientées recherche (par exemple : caractérisation des matériaux) et les "tâches d'ingénierie" sont des tâches

relativement routinières ou des tâches d'amélioration (par exemple : conception de circuit intégré, de batterie, etc.).

- Ces deux types de tâche ne présentent pas le même degré de difficulté dans la conception. Les "tâches innovantes" exigent souvent un niveau de compétence plus élevé que les "tâches d'ingénierie" (0,9 pour les tâches innovantes en moyenne et 0,6 pour les tâches d'ingénierie en moyenne). Cependant, cela est à nuancer car il peut y avoir des situations intermédiaires où on demande de réaliser les tâches routinières avec une nouvelle technologie (tâches d'amélioration).
- Le niveau de compétence requis peut varier selon la "période d'affectation". Le début du projet requiert souvent l'engagement d'experts (haut niveau de compétence) pour orienter et définir les concepts principaux, qui seront ensuite affinés par la seconde phase. Cette seconde phase peut être confiée à un ingénieur "moins expérimenté" maîtrisant les outils de modélisation et la justification des choix.

		<i>k=1</i>		<i>k=2</i>		<i>k=3</i>	
Tâche		$rq_i^{k,l}$	Durée (mois)	$rq_i^{k,l}$	Durée (mois)	$rq_i^{k,l}$	Durée (mois)
		<i>p=1</i>	T1	0,9	• (2)	-	-
	T2	0,9	• (2)	-	-	-	-
	T3	-	-	-	-	0,9	• (1)
	T4	0,9	• (1)	0,9	• (1)	0,9	• (1)
	T5	-	-	0,9	• (3)	0,7	• (2)
	T6	-	-	0,9	• (2)	0,7	• (1)
	T7	-	-	0,9	• (2)	0,7	• (1)
	T8	-	-	0,9	• (2)	-	-
	T9	-	-	0,7	• (2)	0,7	• (1)
	T10	-	-	0,7	• (2)	0,7	• (1)
	T11	-	-	0,9	• (2)	-	-
	T12	-	-	0,9	• (2)	0,7	• (1)
	T13	-	-	0,9	• (2)	-	-
	T14	-	-	-	-	-	-
	T15	-	-	-	-	-	-
		<i>k=1</i>		<i>k=2</i>		<i>k=3</i>	
Tâche		$rq_i^{k,l}$	Durée (mois)	$rq_i^{k,l}$	Durée (mois)	$rq_i^{k,l}$	Durée (mois)
		<i>p=2</i>	T1	-	-	-	-
	T2	-	-	-	-	-	-
	T3	-	-	0,7	• (1)	-	-
	T4	0,9	• (1)	0,9	• (1)	-	-
	T5	-	-	-	-	-	-
	T6	0,7	• (3)	-	-	-	-
	T7	-	-	-	-	-	-
	T8	0,7	• (2)	-	-	-	-
	T9	0,7	• (2)	-	-	-	-
	T10	0,7	• (3)	-	-	-	-
	T11	0,9	• (2)	-	-	-	-
	T12	0,7	• (2)	-	-	-	-
	T13	-	-	-	-	-	-
	T14	0,9	• (3)	0,7	• (2)	-	-
	T15	0,9	• (3)	0,7	• (2)	-	-

TAB. 4.19: Planification multi-périodes et multi-projets des tâches

4.3.3 Structure générale

Pour décrire la démarche du problème d'affectation multi-période, nous adaptons la méthodologie d'affectation déjà proposée dans le cas mono-période au cas multi-périodes.

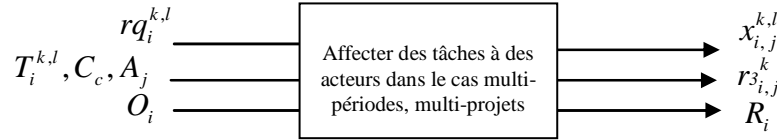


FIG. 4.6 : Vue externe du problème d'affectation multi-périodes, multi-projets

Ce diagramme représente la démarche globale d'affectation dans le cas multi-périodes et multi-projets avec prise en compte de l'évolution des compétences. L'activité "affecter des tâches aux acteurs" est considérée comme la fonction principale. A l'entrée, le système nécessite des données telles que :

- La planification des tâches particulières ($T_i^{k,l}$) sur plusieurs périodes et sur plusieurs projets.
- La liste des acteurs dans l'entreprise (A_j). Ces acteurs sont susceptibles d'être sélectionnés pour travailler sur les tâches des projets.
- La liste des connaissances (C_c). Ces connaissances permettent de caractériser chaque tâche (générique et particulière) et chaque acteur.
- Le niveau requis de compétence concernant la tâche particulière de type i ($r q_i^{k,l}$).
- L'objectif de compétences (O_i) défini pour chaque tâche générique (T_i). L'objectif est défini par le nombre d'acteurs compétents pour chaque tâche générique T_i que l'entreprise souhaite posséder à la fin de l'horizon d'affectation (après K périodes). Nous supposons qu'un seuil du niveau de compétence a été défini par le management stratégique et permet de distinguer si un acteur est jugé compétent ou non.

A la sortie, le système renvoie des résultats tels que :

- Les variables réelles de décision d'affectation $x_{i,j}^{k,l}$ (ou la contribution des acteurs aux tâches), $0 \leq x_{i,j}^{k,l} \leq 1$.
- L'ensemble des valeurs permettant de montrer l'évolution après chaque période k du niveau de compétence, acquis par l'acteur j sur les tâches génériques ($r_{i,j}^k$), $k=1, \dots, K$.
- Le vecteur représentant le nombre d'experts à la fin de l'horizon d'affectation (R_i). L'écart entre ce vecteur et l'objectif fixé au départ représente le nombre d'experts manquants permettant d'estimer un coût supplémentaire (pénalité) dû à la non-satisfaction des objectifs de compétence.

La vue interne de la démarche est présentée sur la figure (FIG. 4.7). Elle se compose de 5 phases différentes.

- La première phase a pour but de caractériser chaque tâche particulière ($r_{i,c}^{k,l}$) et les niveaux de maîtrise des connaissances initiaux de l'acteur ($r_{c,j}^{k=0}$), de façon assez semblable à la démarche déjà présentée dans le cas mono-période.

- La deuxième phase concerne le calcul de l'évolution du niveau de maîtrise de la connaissance acquis par l'acteur ($r_{2,c,j}^k$). Une loi d'augmentation et une loi de régression sont modélisées et prises en compte dans le calcul.
- La troisième phase vise à trouver la proximité entre tâches et acteurs en utilisant le calcul de similarité. Nous obtenons l'indicateur de similarité a priori entre tâches particulières et acteurs ($v_{3,i,j}^{k,l}$) qui compare les niveaux de maîtrise des connaissances requis par la tâche particulière et ceux acquis par l'acteur à la fin de la période précédente. Cette comparaison permet de savoir si l'acteur est compétent pour les tâches à affecter lors de la période k . Cet indicateur est utilisé pour calculer le coefficient correcteur ($\gamma_{i,j}^{k,l}$) qui permet d'estimer le coût salarial lié aux niveaux de compétence.
- La quatrième phase recherche une solution optimale. Cette phase nous permet d'obtenir les variables de décision d'affectation ($x_{i,j}^{k,l}$) de chaque acteur et la version finale de l'ensemble des valeurs concernant l'évolution du niveau de maîtrise dans la *connaissance* acquis par l'acteur ($r_{2,c,j}^k$) une fois les affectations optimisées.
- La cinquième phase est destinée à calculer le niveau de similarité a posteriori (ou le niveau de compétence de l'acteur j concernant la *tâche générique* i ($r_{3,i,j}^k$)). Cette étape calcule la proximité entre tâche (générique)-acteur, quand les affectations de la période k ont été réalisées.

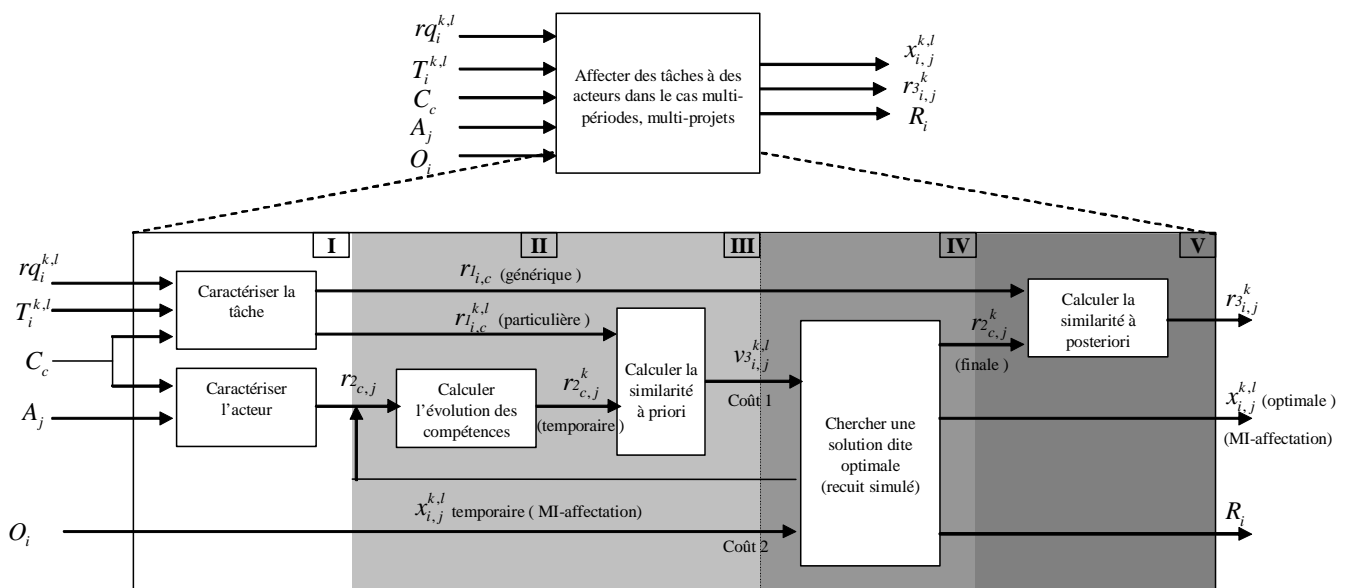


FIG. 4.7 : Démarche d'affectation multi-périodes des tâches à des acteurs

Les cinq phases que nous venons de présenter sont succinctement développées ci-après. Comme la modélisation des lois d'évolution des compétences et la formulation du problème méritent être détaillées, ces deux phases qui sont au cœur de notre contribution seront approfondies en (§4.3.4) et (§4.3.5).

4.3.3.1 Phase I : Caractérisation de la tâche et de l'acteur

Comme dans le cas mono-période (§4.2.3.1), les variables de compétences peuvent être caractérisées sur la tâche et sur l'acteur. Deux niveaux de granularité des compétences (compétence et connaissance) sont modélisés.

Caractérisation de la tâche

Cette étape vise à définir les variables de compétence à la fois requis par la tâche générique et requis par la tâche particulière.

- Tâche générique (T_i)
 - En ce qui concerne la modélisation au niveau de la "compétence", il s'agit d'une variable (rq_i) représentant le niveau requis de compétence concernant la tâche générique de type i .
 - En ce qui concerne la modélisation au niveau des "connaissances", il s'agit d'une variable ($r_{i,c}$) représentant le niveau de maîtrise de la connaissance c requis par la tâche générique de type i .
- Tâche particulière ($T_i^{k,l}$)
 - En ce qui concerne la modélisation au niveau de la "compétence", il s'agit d'une variable ($rq_i^{k,l}$) représentant le niveau requis de compétence concernant la tâche particulière de type i .
 - En ce qui concerne la modélisation au niveau des "connaissances", il s'agit d'une variable ($r_{i,c}^{k,l}$) représentant le niveau de maîtrise de la connaissance c requis par la tâche particulière de type i , planifiée à la période k et appartenant au projet l .

Nous faisons l'hypothèse que le niveau de maîtrise de la connaissance, requis par la tâche particulière ($r_{i,c}^{k,l}$) peut être calculé par le produit du niveau requis de compétence dans la réalisation de la tâche particulière ($rq_i^{k,l}$) et du niveau de maîtrise de la connaissance requis par la tâche générique de type i ($r_{i,c}$), voir (Eq- 4-8).

$$r_{i,c}^{k,l} = rq_i^{k,l} \times r_{i,c}$$

Eq- 4-8

Caractérisation de l'acteur

Cette étape vise à estimer le niveau de maîtrise de la connaissance acquis par l'acteur ($r_{c,j}^{k=0}$) avant l'affectation ($k = 0$).

Dans le tableau (TAB. 4.20), nous avons résumé les variables liées à la caractérisation de la tâche générique, de la tâche particulière et de l'acteur.

Type d'entités caractérisé	Niveau de modélisation (granularité)	
	Compétence	Connaissance
Tâche générique (T_i)	$rq_i = 1$ (niveau de référence)	$r_{1_{i,c}}$ (niveau de référence)
Tâche particulière ($T_i^{k,l}$)	$rq_i^{k,l} = 1$	$r_{1_{i,c}}^{k,l} = r_{1_{i,c}}$
	$0 \leq rq_i^{k,l} < 1$	$r_{1_{i,c}}^{k,l} = rq_i^{k,l} \times r_{1_{i,c}}$
Acteur (A_j)	$r_{3_{i,j}}^k$	$r_{2_{j,c}}^k$

TAB. 4.20 : Caractérisation des tâches et des acteurs (multi-périodes)

4.3.3.2 Phase II : Calcul de l'évolution du niveau de maîtrise acquis dans la connaissance

L'objectif de cette phase est de calculer l'évolution du niveau de maîtrise dans la connaissance c acquis par l'acteur j à la période k ($r_{2_{c,j}}^k$). Les évolutions du niveau de connaissance sont basées sur les décisions d'affectation. Une loi permettant d'estimer l'évolution du niveau de maîtrise de la connaissance $r_{2_{c,j}}^k$ sera présentée ultérieurement (§4.3.4).

Ces niveaux de maîtrise dans les connaissances permettent de calculer par la suite l'indicateur de similarité "a priori" entre les tâches génériques et les acteurs $v_{3_{i,j}}^{k,l}$ (phase III) et l'indicateur de similarité "a posteriori" entre les tâches particulières et les acteurs $r_{3_{i,j}}^k$ (phase V).

4.3.3.3 Phase III : Calcul de l'indicateur de similarité "a priori"

L'indicateur de similarité a priori $v_{3_{i,j}}^{k,l}$ a pour but de mesurer la distance entre tâches et acteurs avant la $kième$ affectation de la tâche particulière $T_i^{k,l}$. On cherche à mesurer à quel point cet acteur correspond à cette tâche particulière. Au moment où l'on calcule cet indicateur, la tâche $T_i^{k,l}$ n'a pas encore été affectée, l'acteur j possède toujours le niveau de maîtrise acquis dans la connaissance à la période précédente ($k-1$). Ainsi, cet indicateur compare la similarité entre le vecteur des niveaux de maîtrise (noté $r_{1_{i,\bullet}}^{k,l}$) dans les connaissances, requis par la tâche particulière $T_i^{k,l}$ et le vecteur de niveau de maîtrise (noté $r_{2_{\bullet,j}}^{k-1}$) dans les connaissances, acquis par l'acteur j de la période précédente.

$$v_{3_{i,j}}^{k,l} = r_{1_{i,\bullet}}^{k,l} \circ r_{2_{\bullet,j}}^{k-1}$$

$$v_{i,j}^{k,l} = 1 - \frac{\sum_{c=1}^O \max(0, r_{i,c}^{k,l} - r_{c,j}^{k-1})}{\sum_{c=1}^O r_{i,c}^{k,l}}$$

$$0 \leq v_{i,j}^{k,l} \leq 1$$

Eq- 4-9

- Le symbole \circ est l'opération d'agrégation définie en (Eq- 4-3, loi de composition).
- Le **vecteur des niveaux de maîtrise dans les connaissances requis par la tâche particulière** ($r_{i,\bullet}^{k,l}$) est constitué des niveaux de maîtrise requis dans les connaissances, $r_{i,\bullet}^{k,l} = [r_{i,c}^{k,l} ; c = 1, \dots, O]$.
- Le **vecteur des niveaux de maîtrise dans les connaissances acquis par l'acteur j à la période $k-1$** ($r_{\bullet,j}^{k-1}$) est constitué des niveaux de maîtrise acquis dans les connaissances, $r_{\bullet,j}^{k-1} = [r_{c,j}^{k-1} ; c = 1, \dots, O]$.

4.3.3.4 Phase IV : Recherche d'une solution dite optimale

Le but de cette étape est de fournir une solution d'affectation prenant en compte les critères liés aux niveaux de compétence et liés à l'évolution des compétences. Pour optimiser la solution, nous élaborons un modèle mathématique, puis nous analysons le type de problème (linéaire ou non linéaire) et nous choisissons une méthode de résolution.

Un modèle mathématique ainsi que les critères utilisés et les contraintes associés à ce problème seront détaillés en (§4.3.5). La méthode de résolution sera approfondie en (§4.3.6).

4.3.3.5 Phase V : Calcul du niveau de similarité a posteriori

Contrairement à l'indicateur de similarité a priori $v_{i,j}^{k,l}$, l'**indicateur de similarité a posteriori** $r_{i,j}^k$ a pour but de mettre à jour le niveau de compétence acquis par l'acteur j dans la réalisation de la tâche i à la fin de la $kième$ affectation.

$$r_{i,j}^k = r_{i,\bullet} \circ r_{\bullet,j}^k$$

Cet indicateur mesure la distance entre le vecteur des niveaux de maîtrise dans les connaissances requis par la **tâche générique** ($r_{i,\bullet}$) et celui acquis par l'acteur ($r_{\bullet,j}^k$). Nous avons choisi une comparaison avec la tâche générique parce que le niveau de maîtrise dans la connaissance requis par la tâche générique $r_{i,c}$ ne dépend pas ni du projet, ni de la période (niveau de référence). D'autre part, ce calcul est en accord avec notre définition de la compétence qui correspond à la capacité de réaliser une classe de tâche. Le niveau de compétence s'estime donc au niveau de la tâche générique.

4.3.4 Contribution à l'évolution des compétences

Cette partie a pour but de comprendre les mécanismes modélisés permettant de faire évoluer les niveaux de compétence (définir des lois d'évolution).

Dans cette partie, nous détaillons le calcul du niveau de maîtrise dans la connaissance $r_{c,j}^k$ (phase II) et le calcul du niveau de compétence $r_{c,j}^k$ (phase V). Pourquoi nous intéressons nous à ces deux variables ? Ces deux variables représentent les niveaux de compétence chez les acteurs (compétence et connaissance) et permettent de suivre leurs évolutions.

D'abord, nous présentons les types de modélisation des compétences (granularité), les démarches possibles de modélisation de l'évolution des compétences (ascendante ou descendante), puis nous expliquons la démarche retenue. Nous présentons ensuite les différentes possibilités de l'évolution de compétences (augmentation, stabilité et diminution). Enfin, nous présentons en détail les étapes pour remettre à jour les niveaux de compétence.

4.3.4.1 Type de modélisations des compétences

Comme dans le cas mono-période (§4.2.3.1), les compétences peuvent être modélisées sur deux entités, la tâche et l'acteur. Les variables de compétence peuvent être décomposées en deux niveaux de modélisation (granularité) : au niveau de la "compétence" et au niveau des "connaissances".

Le tableau (TAB. 4.21) présente les variables utilisées pour la tâche et l'acteur.

		Niveau de modélisation (granularité)	
		Compétence	Connaissance
Type d'entités caractérisé	Tâche	r_{q_i} (Fixé)	$r_{i,c}$ (Fixé)
	Acteur	$r_{i,j}^k$ (Evolutif)	$r_{c,j}^k$ (Evolutif)

TAB. 4.21 : Variables de la modélisation des compétences

- Les variables caractérisant la tâche, quel que soit le niveau de modélisation (compétence, connaissance), sont supposées non-évolutives (fixées). Cette hypothèse signifie par exemple qu'il n'y a pas de changement de technologies important au cours d'une période nécessitant la modification des connaissances nécessaire à la tâche.
- Les variables caractérisant l'acteur, quel que soit le niveau de modélisation, sont supposées évolutives. Les variables de compétence liées à l'acteur contiennent donc l'indice k ($k=1, \dots, K$) permettant de modéliser l'évolution de ses connaissances et compétences.
- Nous supposons que l'affectation d'une tâche engendre une amélioration sur le niveau de maîtrise des connaissances et sur le niveau de compétence concernant la tâche affectée, mais aussi sur d'autres tâches nécessitant les mêmes connaissances.

4.3.4.2 Démarche de modélisation de l'évolution des compétences

La démarche de modélisation de l'évolution des compétences correspond à une remise à jour à chaque période de temps des niveaux de compétence *acquis par un acteur*.

Dans la figure (FIG. 4.8), nous avons défini deux démarches possibles de modélisation de l'évolution des compétences des acteurs : démarche ascendante et démarche descendante. L'affectation des tâches étant réalisée, les niveaux de compétence de l'acteur évoluent. La question qu'on peut se poser est alors comment remettre à jour le niveau d'une compétence suite à une affectation ? Dans une approche descendante, on devrait définir une loi d'évolution de la compétence au niveau global et une loi de déploiement de cette évolution sur les connaissances. Dans une approche ascendante, on devrait définir une loi d'évolution permettant d'estimer le nouveau niveau de maîtrise dans les connaissances suite à des affectations et puis calculer les nouveaux niveaux de compétence.

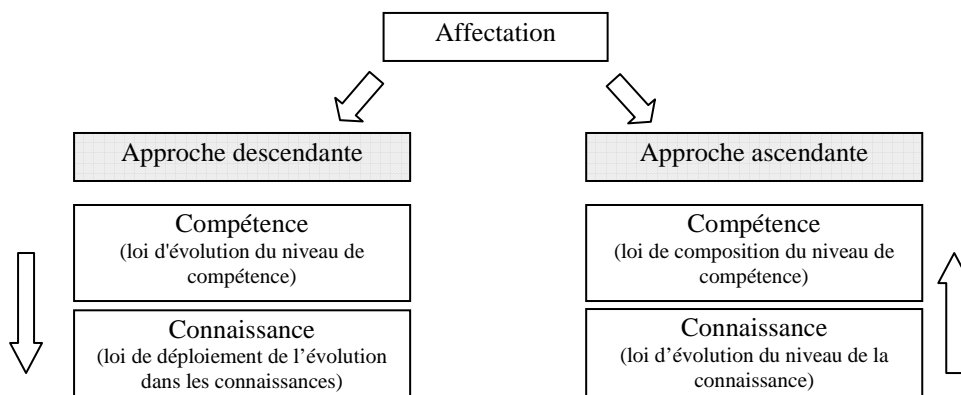


FIG. 4.8 : Démarches "descendante" et "ascendante" de l'évolution des compétences

La modélisation de l'évolution des compétences retenues dans ce document repose sur une approche "ascendante". Nous avons choisi cette démarche car l'indicateur de similarité proposé précédemment est pertinent pour ce type de démarche. Il permet de propager la mise à jour du niveau "connaissance" sur le niveau "compétence".

Selon l'approche ascendante, nous faisons l'hypothèse que lorsque l'on s'améliore dans une compétence, d'autres compétences peuvent en profiter et être aussi améliorées. C'est le cas où certaines connaissances sont transversales à plusieurs compétences. Toutefois, la faiblesse de cette hypothèse tient au fait que le contexte d'application n'étant pas le même, l'amélioration du niveau de maîtrise dans la connaissance peut ne pas être identique.

Pour atténuer l'effet transversal des connaissances, il est possible de considérer une connaissance de type plan d'actions qui serait spécifique à chaque tâche générique. Il est à noter que d'une façon générale dans la littérature, ce type de connaissance, pourtant crucial pour expliquer la combinaison des connaissances, n'est pas modélisé.

4.3.4.3 Structure de la démarche

Nous présentons les étapes détaillées de la démarche ascendante de l'évolution des compétences (FIG. 4.9).

A partir d'une affectation des tâches à des acteurs :

- Étape 1 (FIG. 4.9-n°1) : Calcul, pour chaque tâche réalisée, de l'augmentation du niveau de maîtrise dans la *connaissance* c . Nous choisissons ensuite l'augmentation maximale dans la connaissance c appartenant à l'acteur j à la fin de la k ème affectation, appelé ($DIF_{c,j}^k$).
- Étape 2 (FIG. 4.9- n°2) : Calcul du nouveau niveau de maîtrise de la *connaissance* ($r_{2,c,j}^k$). Cette étape vise à faire évoluer le niveau acquis par un acteur lors la période précédente $r_{2,c,j}^{k-1}$ au niveau de la période actuelle $r_{2,c,j}^k$.
- Étape 3 (FIG. 4.9-n°3) : Calcul du nouveau niveau de compétence ($r_{3,i,j}^k$). Une fois le nouveau niveau de maîtrise dans la connaissance ($r_{2,c,j}^k$) calculé, nous calculons $r_{3,i,j}^k$ par la loi de composition proposée par l'équation (§4.3.3.5).

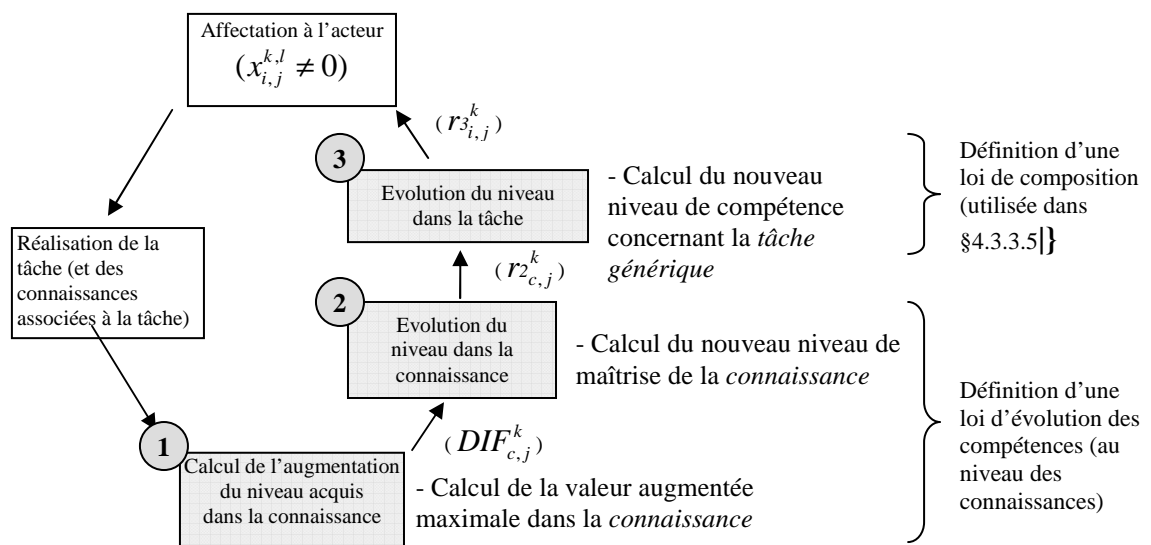


FIG. 4.9 : Démarche ascendante et lois d'évolution des compétences

En résumé, les étapes (1) et (2) présentent la définition d'une loi d'évolution au niveau des "connaissances". L'étape (3) concerne l'évolution au niveau de la "compétence".

Les trois étapes que nous venons de présenter sont développées ci-après.

Nous faisons les hypothèses suivantes :

- Toutes les tâches sont réalisées avec succès,
- La réalisation d'une tâche pour laquelle l'acteur est compétent (le niveau de compétence acquis est supérieur au niveau requis par la tâche) lui permet de conserver son niveau de compétence antérieur. Cette hypothèse a cependant certaines limites puisque, si un acteur n'accomplit que des tâches qui lui sont routinières, il risque de voir son niveau de compétence se dégrader à moyen terme.
- La réalisation d'une tâche pour laquelle l'acteur n'est pas complètement compétent (le niveau de compétence acquis est inférieur au niveau requis par la tâche) lui permet de développer son niveau de compétence. Ce dernier augmente au niveau requis par la tâche. Cette hypothèse a aussi ses limites. Dans le cas d'écart de compétence élevés, l'acteur peut ne pas réussir son apprentissage et par la suite, échouer dans la réalisation de la tâche. Pour limiter les affectations nécessitant de forts apprentissages, nous les

pénalisons relativement fortement. Nous introduisons un coefficient correcteur de la durée nominale de la tâche qui sera fonction de l'écart de compétence (il n'y aura pas de correction pour un acteur compétent). Nous considérons que l'acteur est accompagné par un tuteur pour apprendre en situation à réaliser la tâche. De ce fait, affecter une tâche à un acteur non compétent coûte le travail de l'acteur (calculé sur la base d'une durée réelle estimée avec la durée nominale augmentée par le coefficient correcteur) et le travail d'un tuteur.

4.3.4.3.a Etape 1 : Calcul de l'augmentation du niveau de maîtrise de la connaissance c .

Cette étape vise à calculer l'augmentation maximale du niveau de maîtrise de la connaissance, ou $DIF_{c,j}^k$.

Nous calculons cette valeur selon les étapes suivantes :

Etape 1.1 : Calculer l'augmentation du niveau de maîtrise de la connaissance c acquis par l'acteur j $(\Delta C_{i,l})_{c,j}^k$ au cours de la période k

Nous avons défini la variable $(\Delta C_{i,l})_{c,j}^k$ comme suit. La valeur $(\Delta C_{i,l})_{c,j}^k$ est obtenue à partir de la différence entre les niveaux requis et acquis $(r_{1,i,c}^{k,l} - r_{2,c,j}^{k-1})$ multipliée par la variable de décision de l'affectation $(x_{i,j}^{k,l})$. Nous avons déjà calculé le niveau de maîtrise requis dans la connaissance de la tâche particulière $(r_{1,i,c}^{k,l})$ lors de la phase 1 de la démarche, et le niveau de maîtrise acquis dans la connaissance $(r_{2,c,j}^{k-1})$ de la période $k-1$ est connu.

$$(\Delta C_{i,l})_{c,j}^k = (r_{1,i,c}^{k,l} - r_{2,c,j}^{k-1})x_{i,j}^{k,l}$$

Notons que

$$-1 \leq (\Delta C_{i,l})_{c,j}^k \leq 1, \quad \forall i = 1, \dots, M, \forall l = 1, \dots, P$$

Eq- 4-10

Pendant la période k , l'acteur j peut être alloué à plusieurs tâches de types différents (indice i) et appartenant à des projets différents (indice l).

La matrice $\Delta C_{c,j}^k$ se compose des augmentations du niveau de maîtrise de la connaissance c acquis par l'acteur j à la période k . A titre d'exemple,

$$\Delta C_{c,j}^k = (\Delta C_{i,l})_{c,j}^k = \begin{matrix} & P_1 & P_2 \\ \begin{matrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{matrix} & \begin{bmatrix} \Delta C_{1,1} & \Delta C_{1,2} \\ \Delta C_{2,1} & \Delta C_{2,2} \\ \Delta C_{3,1} & \Delta C_{2,3} \\ \Delta C_{4,1} & \Delta C_{2,4} \end{bmatrix} \end{matrix}$$

Les valeurs $(\Delta C_{i,l})_{c,j}^k$ peuvent être positives, négatives ou nulles selon la valeur de $(r_{i,c}^{k,l} - r_{c,j}^{k-1})$ et la valeur de $x_{i,j}^{k,l}$.

- Valeur de $(\Delta C_{i,l})_{c,j}^k$ positive : cela signifie qu'il s'agit de la mise en œuvre de cette connaissance (l'acteur j est alloué à des tâches liées à la connaissance considérée) et que le niveau de maîtrise de la connaissance c acquis est inférieur au niveau requis (acteurs sous-compétents).
- Valeur de $(\Delta C_{i,l})_{c,j}^k$ nulle : cela signifie que, soit il s'agit de la mise en œuvre de la connaissance mais les niveaux de maîtrise dans les connaissances requis et acquis sont égaux, soit qu'il n'y a pas eu la mise en œuvre de cette connaissance ($x_{i,j}^{k,l} = 0$).
- Valeur de $(\Delta C_{i,l})_{c,j}^k$ négative : cela représente le fait qu'il s'agit de la mise en œuvre de cette connaissance mais le niveau de maîtrise de la connaissance c acquis est supérieur au niveau requis (acteurs compétents).

Etape 1.2 : Choisir la valeur $DIF_{c,j}^k$

Nous faisons l'hypothèse que $DIF_{c,j}^k$ est l'augmentation maximale des $(\Delta C_{i,l})_{c,j}^k$, c'est-à-dire la plus forte valeur de toutes les augmentations du niveau de maîtrise de la connaissance c par l'acteur j pendant la période k (cela correspond à la tâche particulière ayant nécessité de la part de l'acteur j le plus fort apprentissage de la connaissance c pendant la période k).

$$DIF_{c,j}^k = \max_{i,l} [0; (\Delta C_{i,l})_{c,j}^k],$$

$$i \in [1, \dots, M], l \in [1, \dots, P],$$

$$\forall c = 1, \dots, O, \forall j = 1, \dots, N, \forall k = 1, \dots, K$$

Eq- 4-11

Nous illustrons cette idée sous la forme graphique ci-dessous.

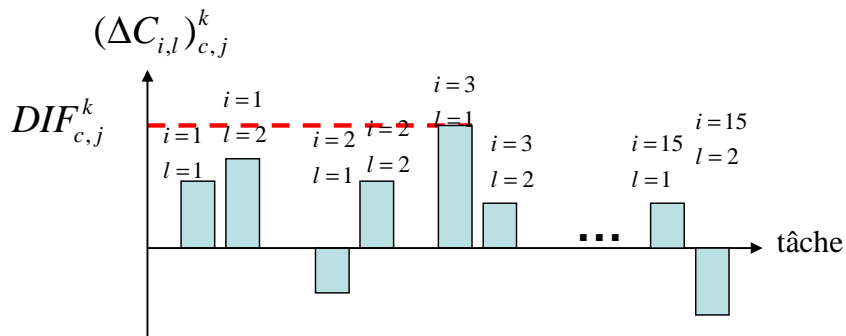


FIG. 4.10 : Augmentation maximale ($DIF_{c,j}^k$)

Exemple d'illustration

L'exemple de la matrice $\Delta C_{c,j}^k$ présenté ci-dessous représente les possibilités d'augmentation dans la connaissance de type ($c=1$) acquis par l'acteur ($j=1$) à la période ($k=1$). Cette matrice contient plusieurs valeurs d'augmentation provenant de différentes tâches (T_i) et de différents projets (p_l).

$$\Delta C_{c,j}^k = (\Delta C_{i,l})_{1,1}^1 = \begin{matrix} & P_1 & P_2 \\ \begin{matrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{matrix} & \begin{bmatrix} 0.05 & 0.03 \\ -0.03 & 0 \\ -0.24 & 0 \\ 0 & 0.04 \end{bmatrix} \end{matrix}$$

L'augmentation maximale $DIF_{c,j}^k$ (quand $c=1, j=1, k=1$) est égale à 0,05. Nous pouvons constater que les valeurs ($DIF_{c,j}^k$) peuvent être également positives, négatives ou nulles.

4.3.4.3.b Etape 2 : Calcul du nouveau niveau de maîtrise de la connaissance

Trois possibilités de changements du niveau de maîtrise acquis dans la connaissance peuvent être considérées : augmentation, stabilité, diminution.

1. Le niveau de maîtrise acquis dans la connaissance c (à la période k) va *augmenter* si une tâche nécessitant la connaissance c est affectée à l'acteur et que le niveau de maîtrise acquis dans la connaissance c (à la période $k-1$) est inférieur au niveau de maîtrise de la connaissance c requis par la tâche (affectée à la période k).
2. Le niveau de maîtrise acquis dans la connaissance c (à la période k) reste *stable* dans le cas où le niveau de maîtrise acquis dans la connaissance c (à la période $k-1$) est supérieur ou égal au niveau de maîtrise requis dans la connaissance c de la tâche (affectée à la période k).
3. Le niveau de maîtrise acquis dans la connaissance c (à la période k) va *diminuer* dans le cas où il n'y a pas d'affectation de tâche à cet acteur qui nécessite la mise en œuvre de cette connaissance.

Sachant qu'il y a trois possibilités de changement du niveau de maîtrise dans la connaissance, il s'agit d'identifier des critères permettant de définir le cas de changement. Nous proposons un chemin logique facilitant la compréhension.

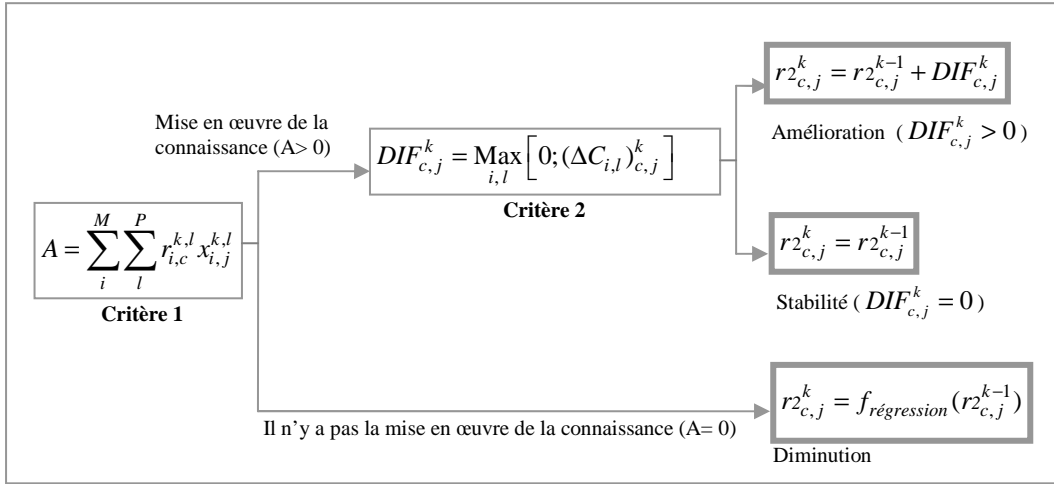


FIG. 4.11 : chemin logique dans le choix de changement du niveau dans la connaissance

Critères utilisés

Ce chemin logique est structuré par deux critères (FIG. 4.11) : relatif à la mise en œuvre de la connaissance (critère 1) et à l'augmentation maximale de la connaissance (critère 2).

Nous fixons les indices i, j, k en considérant ces critères.

- Critère de mise en œuvre de la connaissance. Le premier critère vérifie si la connaissance a été mise en œuvre par l'acteur au cours de la période considérée. Ce critère se traduit par la somme des produits $r_{i,c}^{k,l} \times x_{i,j}^{k,l}$ (le niveau de maîtrise requis de la connaissance c multiplié la variable de décision), présentée dans l'équation (Eq- 4-12). Nous appelons cette somme (A).

$$A = \sum_i^M \sum_l^P r_{i,c}^{k,l} x_{i,j}^{k,l},$$

$$i \in [1, \dots, M], l \in [1, \dots, P],$$

$$\forall c = 1, \dots, O, \forall j = 1, \dots, N, \forall k = 1, \dots, K$$

Eq- 4-12

- Si la connaissance n'a pas été mise en œuvre ($A=0$). Cela amène directement au cas de diminution (Eq- 4-13). Le nouveau niveau de maîtrise de la connaissance c de l'acteur j à la période k est défini ci-dessous (Eq- 4-14). Cette fonction de régression ($f_{\text{régression}}$) basée sur une courbe de type exponentiel est expliquée et justifiée dans l'annexe 5.

$$A = \sum_i^M \sum_l^P r_{i,c}^{k,l} x_{i,j}^{k,l} = 0$$

Eq- 4-13

$$r_{c,j}^k = f_{\text{régression}}(r_{c,j}^{k-1})$$

Eq- 4-14

- Cependant, si ($A > 0$), cela signifie que la connaissance a été mise en œuvre (Eq-4-15). Cependant, nous ne savons pas si c'est le cas d'augmentation ou de stabilité. Nous considérons alors le deuxième critère.

$$A = \sum_i^M \sum_l^P r_{i,c}^{k,l} x_{i,j}^{k,l} > 0;$$

$$\exists (i_1, l_1) \text{ tel que } r_{i_1,c}^{k,l_1} \neq 0 \text{ et } x_{i_1,j}^{k,l_1} \neq 0$$

Eq- 4-15

- Critère d'augmentation maximale de la connaissance ($DIF_{c,j}^k$). Le deuxième critère permet d'identifier si c'est le cas d'augmentation ou le cas de stabilité. Il s'agit de tester la valeur DIF. Cette valeur obtenue par l'étape 1 est toujours positive (voir Eq- 4-11). Si la valeur DIF est supérieure à zéro ($DIF_{c,j}^k > 0$), il y aura une amélioration dans la connaissance. Si la valeur DIF est égale à 0 ($DIF_{c,j}^k = 0$), il n'y aura aucune augmentation du niveau de maîtrise de la connaissance (cas de stabilité).
 - Dans le cas d'augmentation ($DIF_{c,j}^k > 0$), le nouveau niveau de maîtrise dans la connaissance s'écrit comme suit :

$$r_{c,j}^k = r_{c,j}^{k-1} + DIF_{c,j}^k$$

- Dans le cas de stabilité ($DIF_{c,j}^k = 0$), le nouveau niveau de maîtrise dans la connaissance s'écrit simplement³³ :

$$r_{c,j}^k = r_{c,j}^{k-1}$$

4.3.4.3.c Etape 3 : Calcul du nouveau niveau de compétence

- Cette étape est la dernière étape dans la démarche d'évolution de compétences. Elle vise à calculer le nouveau niveau de compétence acquis concernant la tâche. Cette étape correspond à la phase V de la démarche décrite dans la partie (§4.3.3.5).

4.3.5 Modèle mathématique de l'affectation multi-périodes

Le modèle mathématique sera expliqué en 5 parties décrivant :

- les temps de travail corrigés,
- les contraintes liées à la charge de réalisation de la tâche,
- les contraintes liées à la charge des acteurs,
- les critères étudiés,
- le modèle mathématique global.

³³ La formule précédente reste vraie puisque $DIF_{c,j}^k = 0$

4.3.5.1 Temps de travail corrigés

Comme dans le cas mono-période, nous résumons quelques principes. Il s'agit de trois types de temps de travail dans la réalisation des tâches:

- Temps de travail théorique ($L_i^{k,l}$)
- Temps de travail effectué par l'acteur ($\gamma_{i,j}^{k,l} L_i^{k,l}$)
- Temps de formation effectué par le tuteur ($(\gamma_{i,j}^{k,l} - 1)L_i^{k,l}$)

Le **coefficient correcteur** a pour but de calculer la durée corrigée due au cas de sous-compétence dans la réalisation de la tâche $T_i^{k,l}$ par l'acteur j . Le coefficient correcteur $\gamma_{i,j}^{k,l}$ est calculé par l'équation suivante.

$$\gamma_{i,j}^{k,l} = 1 + const \times (1 - v_{3_{i,j}}^{k,l})$$

De même que dans le cas mono-période, nous avons fixé la valeur de $const$ égale à 1. Ainsi, la valeur du coefficient correcteur appartient à l'intervalle $1 < \gamma_{i,j}^{k,l} \leq 2$. La variable $v_{3_{i,j}}^{k,l}$ est un indicateur de similarité a priori entre tâches particulières et acteurs.

$$\gamma_{i,j}^{k,l} = 2 - v_{3_{i,j}}^{k,l}, \quad const = 1$$

$$\gamma_{i,j}^{k,l} = \begin{cases} 1 & \text{si } v_{3_{i,j}}^{k,l} = 1 \\]1, 2] & \text{si } 0 \leq v_{3_{i,j}}^{k,l} < 1 \end{cases}$$

Eq- 4-16

4.3.5.2 Contraintes liées à la charge de réalisation de la tâche

Nous distinguons deux types de durée liés à la tâche : la durée nominale du travail pour la réalisation de la tâche générique i (L_i) et la durée du travail nécessaire à la réalisation de la tâche particulière ($L_i^{k,l}$). La contrainte liée à la charge de travail assure que les tâches particulières sont entièrement affectées aux ressources. Elle est définie ainsi :

$$\sum_{j=1}^N L_i^{k,l} x_{i,j}^{k,l} = L_i^{k,l}, \quad \forall i = 1, \dots, M, \forall l = 1, \dots, P, \forall k = 1, \dots, K$$

Eq- 4-17

4.3.5.3 Contraintes liées à la charge des acteurs

Notons $LM_{j,k}$, la charge maximum que chaque acteur peut supporter (en unité de temps, par exemple, en jours ou en mois). Nous venons de montrer dans la partie précédente que la durée de travail pour réaliser la tâche $T_i^{k,l}$ par l'acteur j est égal à ($\gamma_{i,j}^{k,l} \times L_i^{k,l}$). La contrainte liée à la charge des acteurs peut être formulée comme suit :

$$\sum_{i=1}^M \sum_{l=1}^P \gamma_{i,j}^{k,l} L_i^{k,l} x_{i,j}^{k,l} \leq LM_{j,k}, \forall j = 1, \dots, N, \forall k = 1, \dots, K$$

Eq- 4-18

4.3.5.4 Critères étudiés

Il s'agit de trouver un compromis entre deux critères : le critère de coût du travail lié aux niveaux de compétences et le critère lié à la non-satisfaction des objectifs de compétence.

- Critère lié au coût du travail. Il s'agit de deux coûts complémentaires :
 - Minimiser le coût provenant du temps de travail effectué par l'acteur (fonction objectif N°1).
 - Minimiser le coût provenant du temps de formation effectué par le tuteur (fonction objectif N°2).
- Critère lié à la non-satisfaction des objectifs de compétence. Il s'agit de pénaliser le coût global du projet si l'objectif de compétences n'est pas atteint, c'est-à-dire, si le nombre d'acteurs qualifiés est insuffisant (fonction objectif N°3).

1) Coût provenant de l'acteur ($C1_{i,j}^{k,l}$)

La fonction objectif (Eq- 4-19) a pour but de minimiser le coût salarial.

$$\text{Minimiser } \sum_{k=1}^K \sum_{l=1}^P \sum_{i=1}^M \sum_{j=1}^N C1_{i,j}^{k,l} \times x_{i,j}^{k,l}$$

avec

$$C1_{i,j}^{k,l} = \left[(\gamma_{i,j}^{k,l} \times L_i^{k,l}) \times S_j \right]$$

Eq- 4-19

- le coefficient correcteur ($\gamma_{i,j}^{k,l}$)
- la charge induite de la tâche particulière ($L_i^{k,l}$), (en unité de temps)
- le taux de salaire de l'acteur j (S_j), (en €/unité de temps)
- la contribution de l'acteur j à la tâche particulière, ($x_{i,j}^{k,l}$)

La fonction de coût provenant de l'acteur est le produit du coefficient de coût salarial ($C1_{i,j}^{k,l}$) et de la contribution de l'acteur ($x_{i,j}^{k,l}$). Ce coefficient ($C1_{i,j}^{k,l}$) représente le coût d'utilisation de l'acteur si l'acteur est entièrement alloué à la tâche i ($x_{i,j}^{k,l} = 1$).

2) Coût provenant du tuteur ($C2_{i,j}^{k,l}$)

Le coût d'accompagnement du tuteur est représenté par le produit du coefficient lié au coût du tuteur ($C2_{i,j}^{k,l}$) et de la contribution de l'acteur j à la tâche particulière ($x_{i,j}^{k,l}$), (Eq- 4-20).

Le coefficient $C2_{i,j}^{k,l}$ est un produit de la durée estimée du travail du tuteur, $(\gamma_{i,j}^{k,l} - 1) \times L_i^{k,l}$ et du salaire du tuteur, (ST) qui est supposé d'être le même pour tous les tuteurs (Eq- 4-21). Ce coefficient représentera le coût d'accompagnement du tuteur si la tâche particulière entière est affectée à un seul acteur ($x_{i,j}^{k,l} = 1$).

$$\text{Minimiser } \left[\sum_{k=1}^K \sum_{l=1}^P \sum_{i=1}^M \sum_{j=1}^N C2_{i,j}^{k,l} \times x_{i,j}^{k,l} \right]$$

Eq- 4-20

$$C2_{i,j}^{k,l} = \left[(\gamma_{i,j}^{k,l} - 1) \times L_i^{k,l} \times ST \right], \quad \gamma_{i,j}^{k,l} \geq 1$$

Eq- 4-21

3) Critère lié à la non-satisfaction des objectifs de compétence

Cette fonction objectif a pour but d'évaluer si la solution d'affectation proposée atteint l'objectif de compétences.

La question qu'on peut se poser est comment modéliser un objectif de compétence ? Nous supposons que le directeur R&D ou le chef de projet peut définir un **nombre objectif d'acteurs qualifiés** pour chaque tâche générique i à la fin de l'affectation, O_i ($i=1, \dots, M$). La variable (O_i) est une donnée d'entrée définie dans le but de piloter les compétences des acteurs dans les projets de conception.

A la fin de l'affectation ($k=K$), une variable R_i ($i=1, \dots, M$) indique le nombre d'acteurs qualifiés obtenus à la fin de la simulation, appelée **nombre obtenu d'acteurs qualifiés**.

Les variables O_i et R_i sont des variables en nombres entiers. En comparant le nombre d'acteurs qualifiés objectif et celui obtenu, calculé à la fin de la simulation, nous pouvons estimer des pénalités financières.

Nous définissons un **seuil de performance**, $seuil_i$ ($i=1, \dots, M$) pour chaque tâche générique i . Ce seuil représente un niveau de compétence qui est utilisé pour juger si un acteur est "qualifié" ou non. Cela veut dire qu'à la fin de la simulation, nous recensons pour chaque tâche i les acteurs dont le niveau de compétence ($r_{i,j}^{k=K}$) dépasse le seuil de performance correspondant. Par exemple, si le seuil de performance pour la tâche i est défini à 0,8 ($seuil_i = 0,8$), un acteur j possédant $r_{i,j}^{k=K} = 0,89$ est considérée comme qualifié. On prend alors en compte l'acteur j dans la variable (R_i) qui recense les acteurs qualifiés.

Si le critère n'est pas satisfait ($O_i > R_i$), on pénalise l'affectation avec un coût supplémentaire par acteur qualifié manquant. Nous posons l'hypothèse que la pénalité est une fonction linéaire de la somme des écarts.

D'après l'équation (Eq- 4-22), le coût est calculé par le produit du nombre des acteurs qualifiés manquants (représenté par $O_i - R_i$) et du taux de pénalité. Ce taux est défini par une constante φ_i (unité exprimée en euros par acteur manquant). La variable φ_i dépend de la tâche et permet de représenter le fait que certaines compétences peuvent être plus difficiles à acquérir sur le marché de l'emploi ou sous forme de prestations extérieures que d'autres. On peut considérer aussi que la variable φ_i permet d'homogénéiser les coûts.

$$\text{Minimiser } \sum_{i=1}^M \varphi_i (O_i - R_i)$$

Eq- 4-22

4.3.5.5 Modèle mathématique global

Nous résumons ci-dessous les principales hypothèses que nous avons posées pour les choix d'affectation :

- Les acteurs sous-compétents, c'est-à-dire un niveau de compétence inférieur au niveau requis par la tâche ($v_{3_{i,j}}^{k,l} < 1$), pourront être sélectionnés mais seront pénalisés par un surcoût lié à un coefficient correcteur (durée de réalisation supplémentaire, sollicitation d'un expert pour des demandes d'informations et de conseils). Les acteurs compétents seront sélectionnés sans surcoût s'ils sont "compétents" ($v_{3_{i,j}}^{k,l} = 1$) pour la tâche demandée.
- Les projets seront pénalisés si l'objectif de compétences n'est pas atteint, c'est-à-dire si le nombre d'acteurs qualifiés est insuffisant. Pour diminuer ces pénalités financières, on suppose qu'affecter des tâches aux acteurs sous-compétents va permettre de les former pour qu'ils deviennent compétents.
- Les taux de salaire des acteurs et des tuteurs sont supposés constants durant la simulation.

Nous résumons les variables utilisées pour ce modèle dans le tableau ci-dessous.

- Données au départ

φ_i	Taux de pénalité (en euros par acteur manquant)
O_i	Nombre objectif de compétences, fixé au début de l'horizon d'affectation, décrivant le nombre d'acteurs compétents, demandés pour chaque tâche i à la fin de l'horizon d'affectation.
R_i	Nombre obtenu à la fin de l'affectation (à la K ème affectation, $k=1, \dots, K$) et décrivant le nombre d'acteurs compétents pour chaque tâche i
$L_i^{k,l}$	Charge induite de la tâche particulière $T_i^{k,l}$
$LM_{j,k}$	Charge maximale que peut réaliser l'acteur j pendant la période k .
S_j	Taux de salaire de l'acteur j (euros/ unité de temps).
ST	Taux de salaire des tuteurs (euros/ unité de temps).

- Variables calculées intermédiaires

$\gamma_{i,j}^{k,l}$	Coefficient correcteur
----------------------	------------------------

- Variable de décision

$x_{i,j}^{k,l}$	Variable de décision en nombres réels $x_{i,j}^{k,l} \in]0,1]$ si la tâche i est affectée à l'acteur j $x_{i,j}^{k,l} = 0$, sinon.
-----------------	--

En utilisant les notations ci-dessus, nous pouvons maintenant formuler notre problème d'affectation par un modèle mathématique de la manière suivante.

$$\text{Minimiser } \left\{ \left[\sum_{k=1}^K \sum_{l=1}^P \sum_{i=1}^M \sum_{j=1}^N (C1_{i,j}^{k,l} + C2_{i,j}^{k,l}) x_{i,j}^{k,l} \right] + \sum_{i=1}^M \varphi_i (O_i - R_i) \right\}$$

Eq- 4-23

$$C1_{i,j}^{k,l} = [(\gamma_{i,j}^{k,l} \times L_i^{k,l}) \times S_j]$$

Eq- 4-24

$$C2_{i,j}^{k,l} = [(\gamma_{i,j}^{k,l} - 1) \times L_i^{k,l} \times ST]$$

Eq- 4-25

sous les contraintes :

$$\sum_{j=1}^N L_i^{k,l} x_{i,j}^{k,l} = L_i^{k,l}, \forall i = 1, \dots, M, \forall l = 1, \dots, P, \forall k = 1, \dots, K$$

Eq- 4-26

$$\sum_{i=1}^M \sum_{l=1}^P \gamma_{i,j}^{k,l} L_i^{k,l} x_{i,j}^{k,l} \leq LM_{j,k}, \forall j = 1, \dots, N, \forall k = 1, \dots, K$$

Eq- 4-27

$$0 \leq x_{i,j}^{k,l} \leq 1, \forall i, j, k, l, x_{i,j}^{k,l} \in R^+$$

L'équation (Eq- 4-23) représente la fonction objectif globale de l'affectation multi-périodes. Elle est constituée de trois coûts : le coût provenant du temps de travail effectué par l'acteur, le coût provenant de la participation du tuteur et les pénalités financières liées à la non-satisfaction des objectifs de compétence. La contrainte (Eq- 4-26) assure que toutes les tâches seront affectées. La contrainte (Eq- 4-27) assure que tous les acteurs n'ont pas été affectés au delà de leurs capacités.

Selon l'équation (Eq- 4-24) et (Eq- 4-25), les variables $C1_{i,j}^{k,l}$ et $C2_{i,j}^{k,l}$ dépendent du coefficient correcteur $\gamma_{i,j}^{k,l}$, qui lui-même dépend de la variable de décision $x_{i,j}^{k,l}$. Tenant compte du caractère non linéaire du problème, nous avons choisi de résoudre ce problème par une méta-heuristique, le recuit simulé.

4.3.6 Principe et mise en œuvre de la méthode du recuit simulé

Nous présentons le principe de la méthode du recuit simulé et la mise en œuvre de cette méthode.

4.3.6.1 Principe de la méthode du recuit simulé

En métallurgie, le recuit simulé consiste à amener un solide à basse température après l'avoir élevé à forte température. Lorsque le solide est à une forte température, chaque particule possède une très grande énergie et peut effectuer de grands déplacements aléatoires. Au fur et à mesure que la température est abaissée, chaque particule perd de l'énergie et sa capacité de déplacement se réduit. Par analogie, une technique d'optimisation a été proposée. Pour utiliser cette caractéristique au problème d'optimisation, les déplacements aléatoires vont être dépendants d'une variable de température (T).

Le recuit simulé est un algorithme basé sur une heuristique permettant la recherche de solution d'un problème donné. Il permet notamment de ressortir de minima locaux. La méthode du recuit simulé a été conçue par S. Kirkpatrick *et al.* (1983). Elle fait partie des **méthodes de recherche locale** ou **méthodes de trajectoire** qui tentent itérativement d'améliorer une solution. Le recuit simulé construit une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. Elle fait évoluer la configuration courante en la remplaçant par une autre issue de son voisinage. [Suman et Kumar, 2006] présente un

état de l'art sur l'utilisation de la méthode du recuit simulé pour les problèmes d'optimisation (mono-objectif et multi-objectifs). Le recuit simulé permet d'approcher "de près" la solution optimale du problème plus rapidement qu'une exploration exhaustive dans l'espace de recherche.

Le principe du recuit simulé peut être résumé de la façon suivante. Considérons un système composé de N éléments et une fonction énergétique (f) à optimiser. On recherche la configuration du système qui minimise cette fonction. On commence par une configuration initiale admissible, éventuellement aléatoire. On tire au sort une modification de la configuration qui change l'énergie du système d'une quantité ΔE . Si l'énergie diminue, on effectue le changement. Si l'énergie augmente, on effectue le changement avec une probabilité $\exp(-\Delta E / T)$. L'itération continue tant que l'énergie du système diminue. Lorsque l'énergie reste stationnaire, on diminue un peu la température du système et l'on reprend le processus de décroissance de l'énergie. On arrête lorsque les diminutions de température restent inefficaces.

Algorithme :

1. choisir une solution $s \in S$ ainsi qu'une température initiale T .
2. tant qu'aucun critère d'arrêt n'est satisfait faire
3. choisir aléatoirement $s' \in N(s)$;
4. générer un nombre réel aléatoire r dans $[0,1]$;
5. si $r \leq p(T, s, s')$ alors poser $s := s'$
6. mettre à jour T ;
7. Fin du tant que

La fonction $p(T, s, s')$ est généralement choisie comme étant égale à la distribution de Boltzmann $e^{-\frac{f(s)-f(s')}{T}}$.

- Si $f(s') \leq f(s)$ alors $e^{-\frac{f(s)-f(s')}{T}} \geq 1$, ce qui signifie que $r \leq p(T, s, s')$ et on accepte donc s'
- Si $f(s') > f(s)$ et si T a une très grande valeur alors $e^{-\frac{f(s)-f(s')}{T}} \cong 1$, et on est donc presque sûr d'accepter s' .
- Si $f(s') > f(s)$ et si T a une très petite valeur alors $e^{-\frac{f(s)-f(s')}{T}} \cong 0$ et on va donc probablement refuser s' .

La méthode du recuit simulé est une méthode sans mémoire. On peut facilement l'améliorer en ajoutant une mémoire à long terme qui stocke la meilleure solution rencontrée. En effet, l'algorithme du recuit simulé peut converger vers une solution sf^* en ayant visité auparavant une solution s^* de valeur $f(s^*) < f(sf^*)$. Il est donc logique de mémoriser la meilleure solution rencontrée s^* durant le processus de recherche.

Le fait de partir d'un niveau de température élevée donne à l'algorithme une bonne capacité d'exploration et un refroidissement assez lent évite que la recherche s'arrête sur un minimum local.

4.3.6.2 Justification de l'utilisation du recuit simulé

Lorsque l'on tient compte du caractère non linéaire de l'évolution des compétences et du caractère multi-périodes du projet, le problème ne peut plus être résolu par une méthode exacte connue. Nous proposons de rechercher une solution au problème en développant une heuristique basée sur la méthode du recuit simulé.

Notre choix pour la méthode du recuit simulé est justifié par la simplicité de la programmation, par la facilité d'obtention d'une solution initiale (par programmation linéaire) et par la relative simplicité du processus de génération des voisins dans le problème traité.

Elle permet d'optimiser le problème dans sa globalité, c'est-à-dire, de prendre en compte des critères étudiés et des contraintes imposées sur tout horizon d'affectation.

Parmi les autres méta-heuristiques, les algorithmes génétiques ou la méthode taboue auraient pu être utilisés sur ce type de problème d'optimisation combinatoire. Cependant, nous avons écarté l'utilisation des algorithmes génétiques en raison des contraintes du problème, qui rendent difficile la génération d'une population initiale d'individus, qui aurait ensuite pu être recombinaisonnée ("crossover"). De plus, les individus produits pourraient être fréquemment non admissibles (risque fort d'une durée d'optimisation très longue). La méthode taboue n'a pas été retenue aussi en raison de la taille des matrices manipulées, la maintenance d'une liste taboue de voisins aurait été très coûteuse en mémoire.

4.3.6.3 Mise en œuvre de la méthode du recuit simulé

La méthode du recuit simulé est une méthode simple à utiliser mais nécessite un réglage minutieux de ses paramètres. Nous tenons compte des éléments suivants dans la mise en œuvre de la méthode :

- Construction d'une solution initiale
- Voisinage
- Critères étudiés et probabilité d'acceptation
- Processus de décroissance de la température

Construction d'une solution initiale

La méthode du recuit simulé nécessite une solution initiale pour lancer l'algorithme. La figure ci-dessous présente le principe de construction d'une solution initiale.

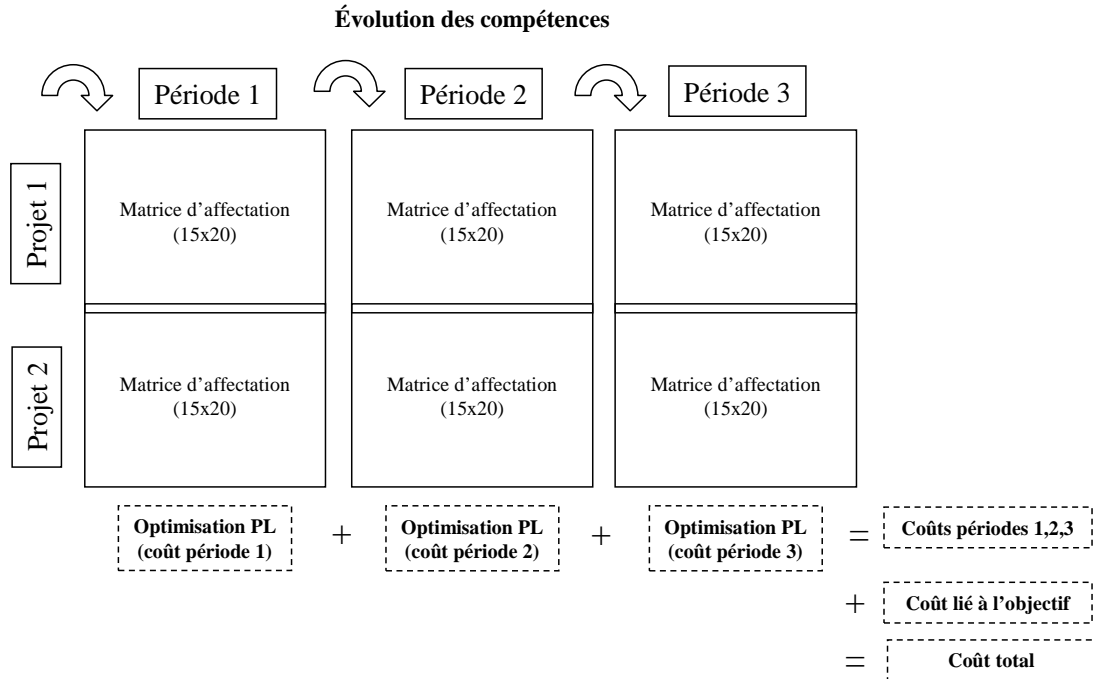


FIG. 4.12 : Optimisation par morceaux par la Programmation Linéaire (PL)

Nous pouvons remarquer que si chaque période est considérée séparément et si on ne prend pas en compte le critère de non-satisfaction des objectifs de compétence, alors le problème se ramène à un problème de programmation linéaire (PL) en nombres réels qui se résout facilement. Nous obtenons ainsi une solution admissible initiale de l'affectation. En effet, en optimisant par PL sur chaque période, nous obtenons une solution d'affectation qui respecte les contraintes de charges de l'acteur et de la tâche.

Les niveaux de compétence obtenus à la fin de chaque période seront les niveaux de départ pour la période suivante. Le problème est décomposé en K sous-problèmes locaux d'optimisation mono-périodes (K nombre de périodes). Ceci permet d'affecter des tâches à des ressources en tenant compte de l'évolution de compétences mais il ne permet pas de considérer le problème dans sa globalité.

En utilisant l'optimisation PL (pour les K sous-problèmes), nous obtenons le coût provenant du temps de travail effectué par l'acteur et le coût provenant de la participation du tuteur pour les trois périodes ("coûts périodes 1, 2, 3"). Cependant, les pénalités financières liées à la non-satisfaction des objectifs de compétence ("coût lié à l'objectif") ne sont pas optimisées car c'est le critère qui doit prendre en compte toutes les périodes en même temps. Elles sont donc calculées à part (sans les optimiser), une fois l'optimisation de toutes les périodes terminée. Enfin, le coût total ("coût total") est la somme de ces trois coûts.

Voisinage

La notion de voisinage est la transformation d'une configuration en une autre configuration qui correspond à une solution voisine (si cette configuration est admissible). Dans notre travail, les solutions voisines ont été générées par une procédure d'échange, qui est définie comme suit :

- Etape 1 : Sélectionner une tâche (ligne) de façon aléatoire
- Etape 2 : Sélectionner deux acteurs (colonne) aléatoirement selon la tâche sélectionnée de l'étape précédente. Ensuite, changer le pourcentage de contribution de ces deux acteurs.

Nous avons fixé la valeur d'échange à chaque perturbation à 0,2, sous réserve que la solution nouvelle vérifie toujours les contraintes. Par exemple, $V(x_{i,j}^{k,l})$ est un voisin de $x_{i,j}^{k,l}$, si on sélectionne la ligne i aléatoirement, on trouve deux acteurs j_1 et j_2 aléatoirement, ensuite on ajoute 0,2 à l'acteur j_1 en enlevant 0,2 de l'acteur j_2 , $V(x_{i,j_1}^{k,l}) = x_{i,j_1}^{k,l} + 0,2$ et $V(x_{i,j_2}^{k,l}) = x_{i,j_2}^{k,l} - 0,2$.

Nous avons défini des pas de 0,2 au lieu d'un nombre aléatoire, pour que l'affectation des tâches ne soit pas trop fractionnée (pas plus de 5 acteurs par tâche) et que les solutions voisines générées aient de fortes chances d'être admissibles.

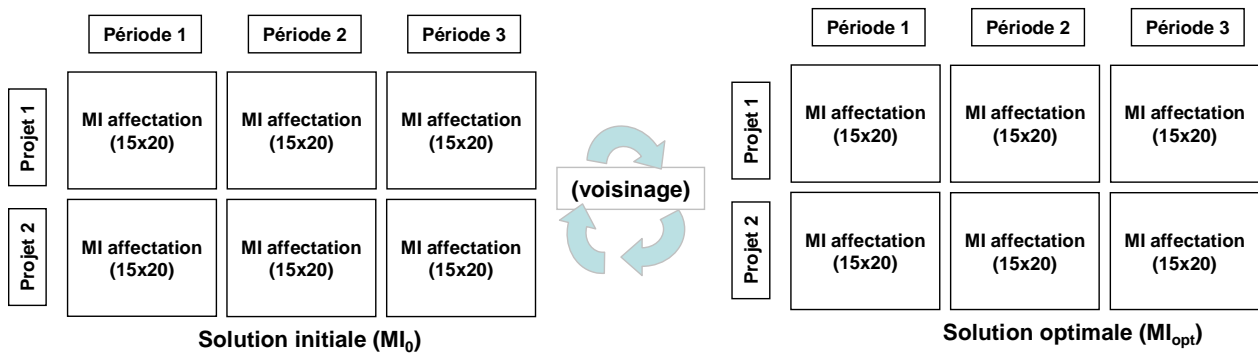


FIG. 4.13 : Optimisation par le recuit simulé

Critères étudiés et probabilité d'acceptation

La fonction objectif permet de mesurer la qualité de la solution obtenue. La fonction objectif est composée des trois critères présentés dans (4.3.5.4).

À chaque itération de l'algorithme, une modification de la solution courante sera effectuée. Cette modification entraîne une variation ΔE de l'énergie du système (toujours calculée à partir de la fonction objectif (F) que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité.

$$P(MI_1, MI_2, T_m) = \exp\left(\frac{\Delta E}{T}\right) = \exp\left[\frac{-(F(MI_2) - F(MI_1))}{T_m}\right]$$

Eq- 4-28

La matrice MI représente l'ensemble des matrices d'affectation. La probabilité dépend de la solution courante MI_1 , de la solution modifiée MI_2 , du coût total de la solution courante $F(MI_1)$, du coût total de la solution modifiée $F(MI_2)$, et de la température T_m .

Processus de décroissance de température

La probabilité d'acceptation est en partie déterminée par la décroissance de la température. Cela est considérée comme un point important dans le recuit simulé.

En général, la méthode la plus utilisée dans le processus de décroissance de la température consiste à définir $T_{k+1} = \alpha T_k$ où α est un nombre réel choisi entre 0 et 1 (typiquement $\alpha=0.95$). Pour converger plus rapidement vers un état stable, nous avons défini une variable simple, appelée *unsurT* (ou $1/T$). Le fait de diminuer la température (T) est équivalent à augmenter la valeur de $(1/T)$. La variable *unsurT* est indexée par un indice ascendant m , ($m=1, \dots, M$). Le début du processus de décroissance commence quand $m=1$ et la fin du processus arrive quand $m= M$; M est un nombre entier.

Le nombre de paliers est représenté par la variable *Palier*, indexée par n , ($n=0, \dots, N$). Cette valeur *Palier* doit être suffisamment grande pour permettre de générer un nombre d'échanges suffisamment grand dans la matrice MI.

L'algorithme de recuit simulé utilisé s'écrit comme suit.


```

-----
1.Initialisation des paramètres
1.1 Le compteur de température,  $m=0$ ;
1.2 Le compteur de palier,  $n = 0$ ;
1.3 Initialise le nombre de décroissances de température  $M$ 
1.4 Initialise le nombre de paliers  $N$ 
1.5 Générer la solution initiale  $MI_0$ ,  $MI_1 := MI_0$ ;
1.6 Calculer la fonction objectif  $F(MI_1)$ ;
1.7  $MI_{opt} := MI_1$ ;  $F_{opt} := F(MI_1)$ ;
2. Tant que ( $m < M$ )
     $m := m + 1$ ;
     $n := 0$ ;
    Tant que ( $n < N$ )
        (i) Effectuer  $MI_2 \in V(MI_1)$ 
        (ii) Calculer la probabilité d'acceptation  $P(MI_1, MI_2, T_m)$ 
        (iii) Accepter ou rejeter le candidat  $MI_2$ 
            Si ( $F(MI_2) \leq F(MI_1)$ ) Alors
                accepter la solution ( $MI_1 := MI_2$ );
                accepter la fonction coût ( $F(MI_1) := F(MI_2)$ );
            Si ( $F(MI_1) \leq F(MI_{opt})$ ) Alors
                accepter la solution optimale ( $MI_{opt} := MI_1$ );
                accepter la fonction coût optimale ( $F_{opt} := F(MI_1)$ );
            Finsi
             $random := rand()$ ;
            Si ( $random < P(MI_1, MI_2, T_m)$ ) Alors
                accepter la solution ( $MI_1 := MI_2$ );
                accepter la fonction coût ( $F(MI_1) := F(MI_2)$ );
            Finsi
        Finsi
     $n = n + 1$ ;
    Fin Tant que
     $m = m + 1$ ;
    Fin Tant que
Afficher  $MI_{opt}$ ;  $F_{opt}$ ;
-----

```

FIG. 4.14 : Algorithme du recuit simulé utilisé

4.3.7 Test du principe

Nous avons créé un cas d'école suffisamment réaliste pour nous permettre d'en tirer des éléments de validation. Ce cas d'école, à partir des données d'entrée (la liste des tâches, des

connaissances, des acteurs, la planification des tâches, etc.), présentées aux (§4.2.5 et §4.3.2.2), nous permet de vérifier la pertinence de notre démarche.

4.3.7.1 Solution finale

En utilisant la méthode du recuit simulé, nous obtenons une nouvelle solution optimale³⁴ que nous présentons par la suite. Cette solution comprend 6 matrices d'affectations optimisées (pour chaque période et pour chaque projet). A cause de la taille importante de ces matrices, nous les avons présentées dans l'annexe 3.

Cependant, nous avons créé le tableau (TAB. 4.22) récapitulant les informations importantes concernant ces 6 matrices d'affectation. Ce tableau met en évidence les informations suivantes : les profils de compétence (électronicien, mécatronicien, etc.), les niveaux d'expertise (E-Expert, C-Confirmé et D-Débutant) et les pourcentages d'affectation pour chaque période.

³⁴ Evidemment, ceci est un abus de langage puisque nous n'avons pas la garantie, ni démontrer que la solution finale du recuit simulé est optimale.

	Tâche	k=1	k=2	k=3
p=1	T1 (spécification)	analyse produit, A1	-	-
	T2 (architecture)	électronicien-E, A5 (0,2), architecte-E, A19 (0,1) intégrateur-C, A20 (0,7)	-	-
	T3 (intégration)	-	-	intégrateur-C, A20
	T4 (management)	intégrateur-C, A20	intégrateur-C, A20	intégrateur-C, A20
	T5 (capteur)	-	électronicien-C, A6 (0,5) mécatronicien-C, A13 (0,5)	automaticien-D, A18
	T6 (vision)	-	automaticien-C, A16	automaticien-D, A18
	T7 (manipulation)	-	mécatronicien-E, A12 (0,2) mécatronicien-C, A14 (0,2) mécatronicien-D, A15 (0,4) intégrateur-C, A20 (0,2)	mécatronicien-C, A13 (0,2) mécatronicien-C, A14 (0,2) mécatronicien-D, A15 (0,4) intégrateur-C, A20 (0,2)
	T8 (locomotion)	-	mécatronicien-C, A14	-
	T9 (circuit)	-	électronicien-C, A6 (0,5) électronicien-D, A8(0,3) mécatronicien-C, A13(0,2)	électronicien-D, A8
	T10 (contrôle)	-	automaticien-C, A17	automaticien-C, A17
	T11 (énergies)	-	mécatronicien-E, A12 (0,2) mécatronicien-C, A13 (0,8)	-
	T12 (communication)	-	électronicien-D, A8	électronicien-D, A8
	T13 (navigation)	-	électronicien-D, A8 (0,2) automaticien-C, A16 (0,2) automaticien-C, A17 (0,6)	-
	T14 (capteur nano)	-	-	-
	T15 (capteur optique)	-	-	-
p=2	Tâche	k=1	k=2	k=3
	T1 (spécification)	-	-	-
	T2 (architecture)	-	-	-
	T3 (intégration)	-	intégrateur-C, A20	-
	T4 (management)	intégrateur-C, A20	intégrateur-C, A20	-
	T5 (capteur)	-	-	-
	T6 (vision)	automaticien-C, A16 (0,9) automaticien-C, A17 (0,1)	-	-
	T7 (manipulation)	-	-	-
	T8 (locomotion)	mécatronicien-C, A13 (0,2) mécatronicien-C, A14 (0,8)	-	-
	T9 (circuit)	électronicien-C, A6 (0,6) électronicien-C, A7 (0,2) électronicien-D, A8 (0,2)	-	-
	T10 (contrôle)	automaticien-C, A17	-	-
	T11 (énergies)	mécatronicien-C, A13	-	-
	T12 (communication)	électronicien-D, A8	-	-
	T13 (navigation)	-	-	-
	T14 (capteur nano)	électronicien-C, A6 (0,2) mécatronicien-E, A12 (0,4) mécatronicien-D, A15 (0,4)	mécatronicien-D, A15 (0,4) automaticien-D, A18(0,6)	-
T15 (capteur optique)	électronicien-D, A8 (0,1) automaticien-D, A18 (0,9)	automaticien-D, A18	-	

TAB. 4.22: Solution d'affectation optimale (recuit simulé)

Le tableau (TAB. 4.23) représente l'évolution du nombre d'acteurs compétents du début à la fin de la simulation. Le nombre d'acteurs compétents est considéré par rapport à l'objectif et au seuil de compétence demandé pour chaque tâche (colonnes 2 et 3). Dans ce tableau, nous comparons l'état initial ($k=1$) à l'état final ($k=K$) en fonction de la satisfaction des objectifs de

compétence. Chaque état (initial ou final) présente les informations suivantes : nombre d'acteurs compétents, état de satisfaction (colonne S/NS ; S=objectif Satisfait, NS=Non Satisfait) et acteurs sélectionnés. La dernière ligne du tableau présente le nombre total d'acteurs compétents manquants. Cette valeur sera utilisée pour estimer les pénalités financières liées à la non-satisfaction des objectifs de compétence.

Tâche	Objectif		R3 initiale (k=1)			R finale (k=K)		
	Nb.	Seuil	Nb.	S/NS ³⁵	Acteur	Nb.	S/NS	Acteur
T1	1	0,8	1	S	A1	1	S	A1
T2	2	0,7	4	S	A2,A12,A19,A20	2	S	A19,A20
T3	2	0,7	3	S	A12,A19,A20	2	S	A12,A20
T4	2	0,8	2	S	A19,A20	2	S	A19,A20
T5	3	0,7	4	S	A4,A5,A6,A13	5	S	A5,A6,A13,A18,A20
T6	4	0,7	2	NS	A16,A17	3	NS	A16,A17,A18
T7	4	0,7	6	S	A2,A9,A12,A13,A19,A20	6	S	A9,A12,A13,A14,A15,A20
T8	3	0,8	1	NS	A12	2	NS	A12,A14
T9	3	0,8	2	NS	A5,A6	3	S	A5,A6,A13
T10	1	0,8	1	S	A12	2	S	A17,A20
T11	2	0,7	1	NS	A12	2	S	A12,A13
T12	3	0,8	2	NS	A16,A17	3	S	A8,A16,A17
T13	3	0,7	2	NS	A16,A17	3	S	A8,A16,A17
T14	4	0,8	2	NS	A5,A12	4	S	A5,A12,A13,A20
T15	4	0,8	0	NS	-	1	NS	A18
Manque	-	-	14p	-	-	5p	-	-

TAB. 4.23: Evolution du nombre des acteurs compétents

A partir de ces deux tableaux, nous retenons les points suivants :

- Nous constatons que le nombre total d'acteurs non compétents a diminué. Cela montre le bon fonctionnement de la loi d'évolution des compétences proposée. Elle confirme également l'hypothèse selon laquelle l'affectation peut être considérée comme un outil permettant d'améliorer les compétences des acteurs selon les objectifs fixés.
- L'algorithme du recuit simulé fournit une solution d'affectation qui améliore le compromis entre la formation des acteurs sous-compétents (dans le but d'atteindre les objectifs de compétence) et la minimisation du surcoût de salaire lié aux niveaux de compétence des acteurs choisis. Prenons l'exemple de la tâche T7 (module de traitement et manipulation des tissus). Nous constatons dans le tableau (TAB. 4.22) que l'acteur, A15 un débutant, a été formé (affecté) à la tâche T7 pendant les périodes 2 et 3. Le programme a choisi de former le débutant A15, qui n'est pas qualifié au début, pour devenir un acteur qualifié à la fin (voir TAB. 4.23). En revanche, l'acteur A19 qui n'a pas été alloué sur cette tâche diminue sa compétence et sort de la liste des qualifiés.

Le tableau (TAB. 4.24) présente l'évolution des compétences des acteurs alloués à la réalisation de la tâche T7. Les niveaux d'expertise des acteurs (E-Expert, C-Confirmé et D-Débutant) sont également présentés. Nous constatons que le niveau de compétence de l'expert A19 sur la tâche T7 est inférieur au seuil de compétence. Il devient sous-compétent par rapport à cette tâche à la fin de l'affectation. Cependant, ce choix d'affectation peut s'expliquer par le

³⁵ S=Satisfait, NS=Non satisfait

coût de salaire élevé de l'acteur A19 qui est un expert. Il est alors préférable de choisir d'autres acteurs débutants (moins coûteux).

T7	A9	A12	A13	A14	A15	A19	A20
(seuil =0,7)	E	E	C	C	D	E	C
période (k=0)	0,82	0,91	0,77	0,68	0,54	0,77	0,77
période (k=1)	0,79	0,91	0,81	0,73	0,61	0,79	0,77
période (k=2)	0,75	0,92	0,79	0,81	0,73	0,73	0,81
période (k=3)	0,71	0,90	0,81	0,82	0,76	0,67	0,81

TAB. 4.24 : Niveau de compétence acquis concernant la tâche T7, ($r_{i,j}^k$)

Le programme est capable de sélectionner les débutants qui ont un niveau de compétence "suffisant" pour être formé dans la tâche tout en trouvant un compromis satisfaisant entre les coûts du travail de l'acteur et du tuteur. Aussi, nous avons constaté que la contrainte de charge de certains débutants (A8 et A18) est saturée à certaines périodes d'affectation.

Acteur	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
Niveau	D	C	D	E	E	C	C	D	E	C	D	E	C	C	D	C	C	D	E	C
k=1	2,2	0,0	0,0	0,0	0,6	2,0	0,4	3,5	0,0	0,0	0,0	1,3	3,1	1,9	1,7	2,9	3,4	3,5	0,3	3,5
k=2	0,0	0,0	0,0	0,0	0,0	2,6	0,0	3,4	0,0	0,0	0,0	1,0	3,5	2,9	2,0	2,6	3,4	3,5	0,0	3,5
k=3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,1	0,0	0,0	0,0	0,0	0,2	0,2	0,4	0,0	1,0	3,3	0,0	2,3

TAB. 4.25 : Calcul de la charge des acteurs

4.3.7.2 Axe de compétence

Les résultats de l'évolution des compétences chez les acteurs sont présentés en deux parties. La première partie montre l'évolution du niveau de maîtrise dans la "connaissance" et la deuxième partie présente l'évolution du niveau de "compétence".

Evolution du niveau de maîtrise dans les connaissances

La figure (FIG. 4.15) présente l'évolution du niveau de maîtrise dans les connaissances. Elle montre le niveau de maîtrise acquis par l'acteur débutant A15 après avoir passé 3 périodes d'affectation. Nous avons pu constater que ce niveau varie en fonction des décisions d'affectation des tâches qui nécessite la mise en œuvre des connaissances considérées. Deux courbes de niveau de maîtrise de 23 connaissances de cet acteur (A15) sont représentées : celles acquises à la période initiale ($k=0$) et celles acquises après la troisième affectation ($k=3$). Des courbes semblables sont obtenues pour les autres acteurs, mais non représentées.

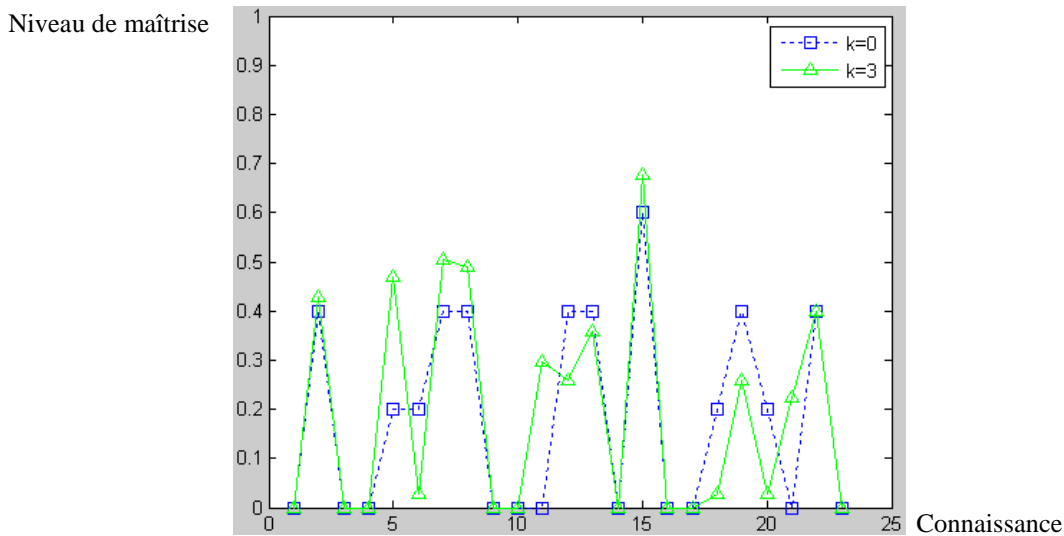


FIG. 4.15 : Evolution du niveau de maîtrise acquis dans la connaissance de l'acteur A15

Nous obtenons que, selon le type de connaissance considéré, le niveau de maîtrise a pu diminuer (par exemple, C12, C18, C19, C20), augmenter (par exemple, C5, C7, C8, C11, C15, C21) ou rester stable (par exemple, C1, C4).

Après ces affectations, A15 a augmenté ses connaissances en matériaux (C5), en électronique (C7), en automatique (C8), en biomédecine (C11), en Matlab (C15), en ingénierie système (C21) ce qui correspond aux cœurs de connaissances d'un mécatronicien dans le biomédical.

Evolution du niveau de compétence

Le niveau de maîtrise acquis dans les connaissances ayant évolué, le niveau de compétence de la tâche a également changé. La figure (FIG. 4.16) présente le niveau de compétence acquis par l'acteur A15 concernant les 15 tâches. Quatre courbes de variations ont été tracées : celles acquises de la période initiale ($k=0$), celles acquises après la première affectation ($k=1$), celles acquises après la deuxième affectation ($k=2$) et celles acquises à la période finale ($k=3$). Des courbes semblables sont obtenues pour 18 acteurs. Les niveaux de compétence acquis par les acteurs ont augmenté après affectations de façon plus ou moins significative.

Ainsi, si on fixe une variation significative qui est au dessus de 0,2, l'acteur A15 a augmenté ses compétences correspondant aux tâches T5, T7 et T14. Par contre, ses compétences sur T2, T3, T4, T6, T8, T9, T10, T12, T13, R15 n'ont pratiquement pas évolué. Or il a été seulement affecté à T7 et T14.

Son niveau de compétence sur T5 a donc augmenté en raison de l'augmentation du niveau de connaissances. Une analyse plus détaillée des connaissances mises en œuvre pour la réalisation de T5 montre que A15 a amélioré ses niveaux de maîtrise de 4 connaissances sur 6 (dont C7 et C11, les plus importantes). Cela explique comment A15 a amélioré indirectement son niveau de compétence sur T5. Etant donné que ce phénomène indirect n'est pas significatif pour A15, et après vérifications pour les autres acteurs, nous pensons que nos modèles d'évolution des compétences s'avèrent relativement cohérents (loi d'augmentation et loi de régression). Toutefois, si on souhaitait diminuer l'impact d'une augmentation indirecte des compétences, il

est possible d'ajouter des connaissances de type "plan d'action" qui seraient spécifiques pour chaque tâche.

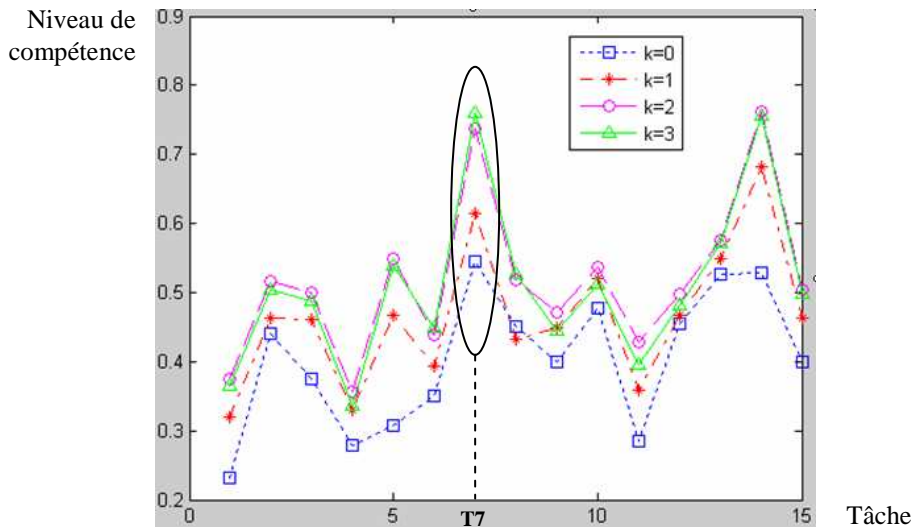


FIG. 4.16 : Axe compétence - Evolution du niveau de compétence acquis par l'acteur A15

4.3.7.3 Axe financier

Ci-dessous se trouvent des courbes représentant des variations de coûts (coût total, coût lié au salaire de l'acteur, coût lié au salaire du tuteur et coût lié à l'objectif de compétence).

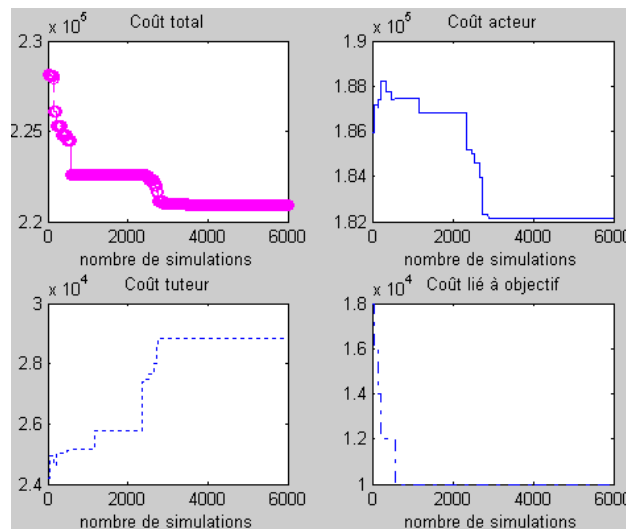


FIG. 4.17 : Axe financier – Variation des coûts

Nous avons constaté une stagnation de l'échange des coûts à partir de 2000 simulations (sur 6000 simulations), malgré le grand nombre de variables de décision. Cela peut s'expliquer par le fait que la solution initiale admissible lors du lancement du processus de recuit simulé est une solution déjà relativement satisfaisante (solution obtenue par la programmation linéaire). Cela permet au processus de recuit simulé de trouver une solution optimale plus rapidement que si la solution initiale était tirée aléatoirement. Il est à noter que le tirage aléatoire d'une solution initiale aurait été difficile en raison des nombreuses contraintes à respecter pour que la solution soit admissible.

Néanmoins, nous sommes conscients que le problème contient un grand nombre de variables (de l'ordre 1800) et de solutions potentielles (de l'ordre de 10^{15}). Même si nous avons utilisé un nombre de simulations assez élevé, nous ne pouvons pas garantir que la solution dite optimale obtenue n'appartient pas à un optimum local.

Nous avons présenté dans la figure (FIG. 4.18) les 200 premières simulations pour observer les échanges entre le coût de l'acteur, le coût du tuteur et le coût objectif. Le programme cherche à diminuer le coût lié à l'objectif de compétences. Le fait que l'on affecte les tâches aux acteurs sous-compétents amène à des coûts supplémentaires pour le projet. Nous avons donc constaté une augmentation des coûts liés aux salaires de l'acteur et du tuteur

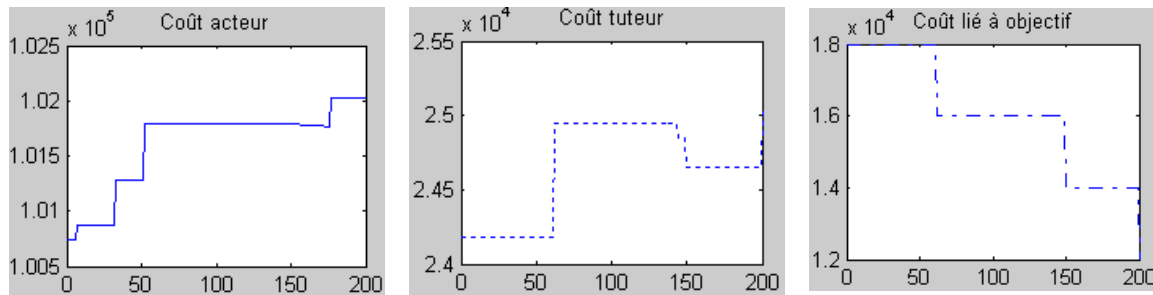


FIG. 4.18 : Axe financier – Compromis entre les coûts liés au salaire, à l'apprentissage et à l'objectif de compétences

Concernant le coût lié à l'objectif de compétences, nous rappelons qu'il est calculé à partir de l'écart entre le nombre d'acteurs qualifiés que l'on espère obtenir à la fin (objectif de compétences), et celui obtenu réellement à la fin de simulation. L'objectif de compétences, ses seuils et le nombre des acteurs compétents au début de la simulation sont des données d'entrée.

Nous remarquons que l'augmentation du nombre d'acteurs qualifiés dépend également du nombre de tâches dans la planification. Le fait que tous les objectifs de compétence ne sont pas satisfaits peut être expliqué par le nombre de tâches insuffisant dans la planification, ce qui ne permettrait pas aux acteurs sous-qualifiés d'améliorer leurs compétences. En même temps, l'objectif de compétences est peut-être trop ambitieux sur l'horizon d'affectation.

Le tableau ci-dessous présente une comparaison des gains entre la solution initiale et la solution proposée.

	Solution initiale (PL)	Solution proposée (Recuit simulé)
Coût salarial lié aux niveaux de compétence	210 136€	210 947€ (+811€)
Coût pénalisant lié à la non-satisfaction des objectifs de compétence (nombre d'acteurs formés)	18 000€ (9 acteurs)	10 000€ (5 acteurs)
Coût total	228 136€	220 947€

TAB. 4.26 : Analyse des résultats

Concernant la solution initiale, elle est obtenue sans prendre en compte l'objectif de compétences. Le tableau (TAB. 4.26) montre que cette solution n'a réussi à former que 5 personnes par rapport à l'objectif fixé de 14 personnes : il manque 9 acteurs compétents.

Quant à la solution obtenue par le recuit simulé, elle permet de former des acteurs sous-compétents même si l'apprentissage de ces acteurs engendre directement une augmentation dans le coût salarial (+811 €), mais une diminution plus importante dans les pénalités financières liées à la non-satisfaction des objectifs de compétence (-8000 €). Cette solution, obtenue après 3 périodes de simulation, a formé 9 acteurs compétents mais il manque 5 acteurs compétents par rapport au nombre objectif d'acteurs qualifiés.

4.3.8 Eléments de validation et synthèse

La méthode proposée fournit une aide à l'affectation multi-périodes dans le but de minimiser les coûts des projets et de tenir compte d'objectifs sur les compétences des acteurs. Le test de cette méthode a été réalisé à travers un cas d'école prenant en compte un objectif chiffré de développement de compétences.

Le modèle proposé pourrait être amélioré sur les points suivants :

- Une limite du cas d'étude que nous avons construit concerne en particulier la surcapacité des acteurs. Il serait intéressant de tester la méthode sur des cas avec des charges plus élevées.
- On pourrait imposer que tout acteur ait un domaine de compétence à la fin de la simulation. Par exemple, l'acteur qui n'est pas affecté à son domaine de compétences pourrait maintenir son niveau de compétences concernant ce domaine.
- Le modèle proposé repose sur l'hypothèse que les acteurs ne développent pas leurs connaissances en dehors de ces projets, ce qui peut être discutable.
- Nous avons supposé que les acteurs sont capables de développer leurs compétences quel que soit leur niveau de compétence initial. Il pourrait être utile (et facile) d'introduire un niveau de compétence minimum à partir duquel un apprentissage par tutorat est envisageable.
- Nous n'avons pas traité le problème de motivation du personnel dans ce travail. Cette extension du problème pourrait être envisagée.

Le (TAB. 4.27) donne une estimation qualitative sur les qualités attendues.

Exigences	Estimation	Limites de la validation
E1 : Affecter des tâches dans le cas multi-périodes et multi-projets en tenant compte d'une loi d'évolution des compétences	++	Test sur un cas d'école Analyse de sensibilité par rapport à $r_{2,j,c}$
E2 : Minimiser le coût salarial lié aux niveaux de compétence	++	
E3 : Minimiser les pénalités financières dues à la non-satisfaction des objectifs de compétence	++	

TAB. 4.27: Estimation qualitative de la méthode

4.4 Conclusion

Après la formulation du modèle d'affectation mono-période dans la première partie du chapitre, nous avons présenté une deuxième contribution sur l'affectation multi-périodes et multi-projets de tâches de conception à des ressources humaines, à l'aide de la méthode du recuit simulé. La difficulté scientifique de ce problème réside d'une part dans sa formulation (choix de lois d'évolution des compétences des acteurs, choix de la fonction objectif, choix d'un indicateur de similarité) et dans sa résolution (problème non linéaire, NP-difficile).

Une autre difficulté est liée aux modalités de test, voire de validation, de cette démarche. N'ayant pas eu l'opportunité de l'expérimenter directement, nous avons construit un jeu de données, suffisamment réaliste, pour nous permettre de tester les hypothèses formulées³⁶.

L'indicateur de similarité entre tâches et acteurs est introduit dans le but de calculer la compatibilité entre tâches et acteurs. Cet indicateur a pour but d'estimer le coût du travail d'un acteur en fonction de son niveau de compétence.

Le modèle d'évolution de compétences nous permet de calculer et de suivre les niveaux de compétences chez les acteurs au cours de la réalisation des tâches. La modélisation des compétences est réalisée sur deux niveaux de granularité : niveaux "compétence" et niveau "connaissance".

Le choix de l'algorithme du recuit simulé comme méthode d'optimisation dans la deuxième contribution (multi-périodes) est cohérent avec le caractère non linéaire et combinatoire du problème ainsi que la possibilité de trouver une solution initiale du problème.

La mise en œuvre de la méthode d'affectation multi-périodes sur l'exemple d'une capsule endoscopique a montré sa possible utilisation et a permis une première validation. Rappelons à cet effet que l'objectif principal de cette méthode est d'aider les managers du projet à appréhender la complexité du pilotage des compétences en leur permettant de simuler et d'analyser les solutions d'affectation et l'évolution de compétences. On peut imaginer qu'à la suite de cette étape le manager du projet puisse être en mesure d'adapter la solution d'affectation proposée.

On peut considérer aussi qu'à la suite de l'application de cette démarche, le manager doit identifier le regroupement des acteurs pour chaque projet. Il pourrait dans ce cas identifier

³⁶ Cette approche est assez classique en ordonnancement.

l'architecture des acteurs en utilisant la solution d'affectation proposée pour chacun des projets. C'est en réponse à cette dernière problématique que nous exposons dans le chapitre suivant une méthode d'identification des architectures des acteurs.

5 Contribution à la constitution d'équipes de conception couplant la structuration du projet et le pilotage des compétences

Ce chapitre est organisé en deux parties.

- La première partie présente une proposition de méthode d'optimisation du couplage entre les architectures de deux domaines. Cette méthode est basée sur un travail au sein de l'équipe permettant de propager des choix d'architecture d'un domaine sur un autre. La particularité de la nouvelle méthode proposée repose sur l'application d'une technique d'optimisation qui permet de réduire le nombre d'interactions au sein d'un projet, nécessaires à la conception d'un produit, en tenant compte des choix d'architecture sur les tâches. Le résultat obtenu correspond à l'architecture des acteurs. Nous testons ce principe sur un exemple de conception de boîte de vitesse robotisée.
 - La seconde partie présente une démarche de constitution d'équipes permettant à la fois de proposer une architecture des acteurs et de piloter les compétences des acteurs à travers l'affectation des tâches à des acteurs. L'identification de l'architecture des acteurs est fondée sur la méthode d'optimisation, proposée dans la première partie. L'affectation repose sur la démarche d'affectation multi-périodes et multi-projets proposée dans le chapitre précédent. Les difficultés scientifiques résident alors dans le couplage de ces deux sous-problèmes et dans la méthode de résolution. Nous appliquons et testons la démarche proposée sur les projets de conception de capsule endoscopique.
-

5.1 Problématique

Nous nous plaçons dans le cadre de la conception de systèmes complexes qui se caractérise par un grand nombre de composants et d'interactions entre ces composants et par un grand nombre d'acteurs interdépendants.

Le problème de la constitution d'équipes de conception n'est pas seulement un problème d'affectation. C'est aussi une étape dans la conception d'une architecture des acteurs puisqu'il s'agit de fournir le découpage en équipes des acteurs du projet. L'architecture des acteurs est primordiale pour faire émerger des compétences collectives ou individuelles jugées stratégiques pour le projet.

En conception préliminaire, l'architecte élabore une architecture préliminaire du produit et le chef de projet doit structurer le projet de développement en tenant compte des interactions nécessaires entre les acteurs pour "bien" concevoir le produit. La phase de développement suit la phase de conception préliminaire. Son but est de réaliser la définition détaillée des composants du produit, en fonction de l'architecture définie préalablement. De ce fait, la phase de développement nécessite beaucoup plus d'acteurs que la phase de conception préliminaire.

Le chef de projet a donc besoin de méthodes d'aide au regroupement des acteurs dans des équipes, au sein desquelles les acteurs pourront collaborer fortement et entre lesquelles les efforts de coordination, coûteux en temps et en budget (frais de déplacement, frais de réunion, ...) pourront être limités. Nous avons déjà présenté, dans la partie 2.1, des travaux qui traitent ce problème en modélisant les interactions entre les acteurs par des matrices DSM et en appliquant sur ces matrices des algorithmes de clustering permettant de minimiser les coûts de coordination entre les équipes en maintenant le maximum d'interactions au sein des équipes. Cependant, ces travaux ne tiennent pas compte de l'architecture du produit. Or, de nombreux auteurs réputés en management de la conception ("engineering management") ou en théorie des organisations ont souligné que l'architecture du produit a un impact fort sur l'architecture des acteurs de conception, c'est-à-dire sur les regroupements d'acteurs en équipes. En effet, les équipes en charge des composants (ou des modules, ensembles de composants) entretiennent des interactions entre elles pour définir les interfaces entre les composants (ou des modules) et assurer leur intégration et validation.

La compréhension de la relation entre l'architecture du produit et l'architecture des acteurs du projet est une nécessité importante dans la structuration du projet de conception. En général, les équipes de conception d'un produit complexe sont organisées de façon empirique selon l'architecture du produit. Le couplage de l'architecture des acteurs avec l'architecture du produit est plutôt laissé aux bons sens du chef de projet, lorsqu'il constitue des équipes, et des acteurs, lorsqu'ils réalisent le projet. Certains travaux [Sosa *et al.*, 2004], [Browning, 2001], [Harmel, 2007], [Baron, 2005], [Gutiérrez-Estrada, 2007], [Robin *et al.*, 2007] s'intéressent aux couplages entre l'architecture du produit et l'architecture des acteurs (structuration du projet). Ces travaux sont très récents et très peu développent une méthode formelle permettant de propager les contraintes d'un domaine sur un autre [Bonjour *et al.*, 2009a].

Partant de constats réalisés en entreprise, dans les travaux qui suivent, nous ferons l'hypothèse qu'à chaque composant ou fonction du produit est associée une unique tâche de définition de ce composant ou de cette fonction. Cette hypothèse permet de relier directement le domaine des acteurs au domaine du produit, ce qui revient à considérer comme identique une matrice DSM produit et une matrice DSM tâche. Cependant, dans le cas où la DSM tâche est déjà connue, cette hypothèse n'est plus utile.

Le schéma (FIG. 5.1) présente le problème que nous allons traiter. Il s'agit de concevoir une organisation de projets de conception durablement performante, en formulant et résolvant des problèmes d'affectation qui tiennent compte des compétences des acteurs, de leur évolution au cours des projets et des choix préliminaires d'architecture des produits. Pour cela, la démarche globale de constitution d'équipes que nous proposons couple les aspects tâche-acteur-compétence.

L'aspect compétence-acteur (sous-problème 1) a pour but de piloter les compétences des acteurs en satisfaisant au mieux des objectifs de développement de compétences. Cet aspect est représenté par l'affectation des tâches à des acteurs en modélisant l'évolution des compétences, déjà traitée en section 4.3.

L'aspect tâche-acteur (sous-problème 2) a pour but de prendre en compte les contraintes de l'architecture des tâches pour identifier une architecture satisfaisante (de l'organisation) des acteurs.

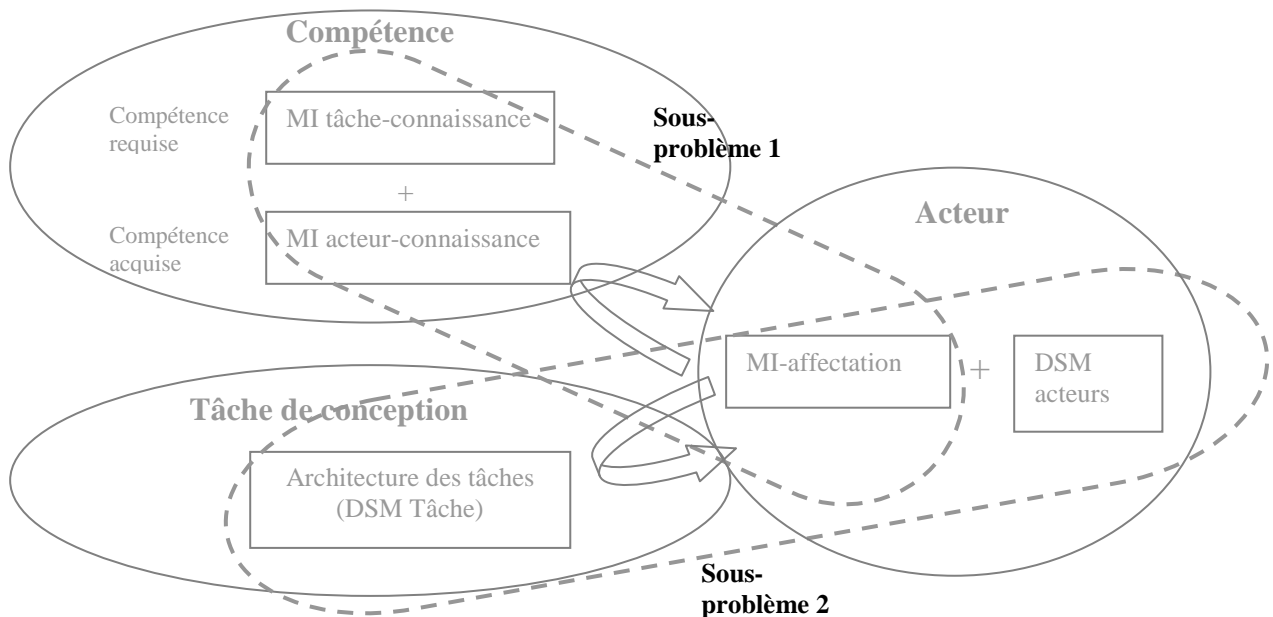


FIG. 5.1: Couplage des aspects tâche-acteur-compétence dans la constitution d'équipes

Les décisions d'affectation obtenues par la résolution du sous-problème 1 sont représentées par une MI-Affectation. Les échanges potentiels d'informations entre les acteurs identifiés par la résolution du sous-problème 2 sont représentés par une DSM Acteurs. Ces informations sont complémentaires pour la **constitution d'équipes** de conception.

En résumé, dans une approche de constitution d'équipes visant à assurer les performances durables des projets de développement, nous proposons d'intégrer, en une méthode unique et globale, la méthode d'affectation multi-périodes avec prise en compte des objectifs de développement des compétences (qui a été présentée au chapitre 4) avec la méthode d'identification d'une architecture des acteurs (qui sera présentée en section 5.2).

5.2 Méthode d'identification d'une architecture par optimisation du couplage entre deux domaines

En collaboration avec S. Chrétien, maître de conférences en mathématiques, et J.Y. Brandelet dans le cadre d'un stage de Master à l'Université de Franche-Comté, nous avons travaillé sur une méthode d'optimisation des couplages entre deux domaines A et B. Nous avons appliqué cette approche sur le problème d'identification de l'architecture des acteurs (domaine B) en tenant compte des interdépendances entre les tâches (domaine A).

La problématique de recherche abordée ici revient à identifier les interactions entre les acteurs (ou efforts de coordination) qui sont nécessaires pour définir au mieux les interfaces entre les tâches (ou composants) et définir les équipes de conception de façon appropriée. Cette problématique peut s'exprimer comme la recherche de minimisation des efforts de coordination tout en garantissant que les interfaces entre les tâches (représentées dans une DSM Tâches) seront bien étudiées. Nous décomposons cette problématique en deux sous-problèmes :

- SP1 : minimiser le nombre d'interactions (donc les efforts de coordination) entre les acteurs tout en garantissant que les interfaces entre les tâches seront bien définies,
- SP2 : regrouper les acteurs en équipes pour minimiser les efforts de coordination (on considère que les coûts d'interaction sont plus élevés entre les équipes qu'au sein d'une même équipe, en raison de la proximité géographique et culturelle des membres d'une même équipe).

Le problème SP2 (qui est un problème classique de clustering) a déjà été traité dans la littérature (partie 2.1). Le problème SP1 n'a pas de solution satisfaisante à ce jour, d'après notre connaissance de la littérature.

Le problème SP1 peut lui-même se décomposer en deux sous-problèmes :

- SP1.1 : minimiser le nombre d'interactions entre les acteurs,
- SP1.2 : garantir que les interfaces entre les tâches seront bien définies si les interactions entre les acteurs sont connues.

Pour apporter une réponse à SP1.2, nous avons comparé, au sein de notre équipe de recherche [Bonjour *et al.*, 2009b], deux méthodes différentes existant dans la littérature. La première méthode, que nous appellerons méthode floue³⁷, a été développée par G. Harmel dans le cadre de son mémoire de thèse [Harmel, 2007], [Bonjour *et al.*, 2009c]. Elle est basée sur une représentation matricielle et utilise un traitement flou en raison du caractère imprécis des données à manipuler (estimation approximative par un expert d'une intensité d'interactions sur une échelle de 0 à 10). La seconde méthode, que nous appellerons méthode matricielle, a été utilisée par [Danilovic et Browning, 2007]. Elle est basée sur un produit matriciel.

Connaissant les choix liés à l'architecture d'un domaine (DSM A connue) et à la projection du domaine A sur un autre domaine B (MI A-B connue), ces méthodes proposent de propager les choix d'architecture du domaine A à travers sa projection sur le domaine B afin de simuler (ou estimer) les interactions nécessaires entre les acteurs (FIG. 5.2). Cela revient à générer une DSM B à partir d'une MI A-B et d'une DSM A.

³⁷ Cette méthode a été utilisée pour la construction de l'architecture des acteurs à partir de l'architecture du produit lors de la phase de conception préliminaire. Elle contient des règles de construction d'une DSM et est basée sur un traitement flou.

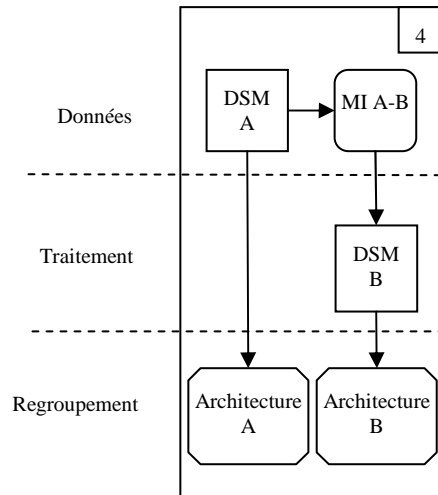


FIG. 5.2 : Schéma de la génération d'une DSM par une méthode de propagation

La méthode floue est la traduction de règles de construction qui permettent de lier les interactions inter-domaines (DSM A) et intra-domaines (MI A-B) pour estimer les interactions dans la matrice DSM B. La méthode proposée offre des résultats satisfaisants (en concordance avec la réalité) et peu sensibles [Bonjour *et al.*, 2009a]. Cependant, elle n'est pas facile à utiliser dans le cadre d'un problème d'optimisation en raison du caractère non linéaire du traitement réalisé.

La méthode matricielle permet d'obtenir des résultats semblables à la méthode floue [Bonjour *et al.*, 2009b], même si une analyse de sensibilité a montré un léger avantage pour la méthode floue. Pour pallier la difficulté liée à l'estimation des intensités d'interaction et par analogie avec la méthode QFD³⁸ utilisée en conception de produits, nous avons préconisé d'utiliser des valeurs discrètes : 0, 3, 6, 9.

Cependant, le but de ces méthodes de propagation n'est pas de rechercher un optimum, c'est-à-dire de s'assurer que la DSM B propagée est minimaliste. Dit autrement, le nombre d'interactions non nulles dans la matrice B obtenue par ces méthodes peut être élevé, sans que l'on puisse s'assurer que les interactions générées ne sont pas redondantes. Nous proposons donc d'améliorer l'identification des architectures (calcul de la DSM B) en proposant une nouvelle méthode d'optimisation.

Nous allons utiliser la méthode matricielle de propagation pour simuler les valeurs des intensités d'interaction. Puis nous allons construire une fonction coût intégrant un critère de fidélité entre la DSM Tâches, construite par le chef de projet, et la DSM Tâches simulée, calculée avec la méthode matricielle de propagation. Comme dans tout traitement de données, nous posons l'hypothèse ici que le chef de projet a fourni des données d'entrée correctes. Nous vérifierons la cohérence des résultats avec le chef de projet à la fois en jugeant la pertinence de l'architecture des acteurs et en comparant l'architecture des tâches simulée et l'architecture construite par le chef de projet.

En résumé (FIG. 5.3), la méthode que nous allons proposer dans cette partie génère une architecture des acteurs optimale à partir de l'architecture des tâches (DSM Tâche connue) et des décisions d'affectation (MI-Affectation tâches-acteurs connue).

³⁸ La méthode QFD préconise d'utiliser quatre valeurs de pondération : 0, 1, 3, 9.

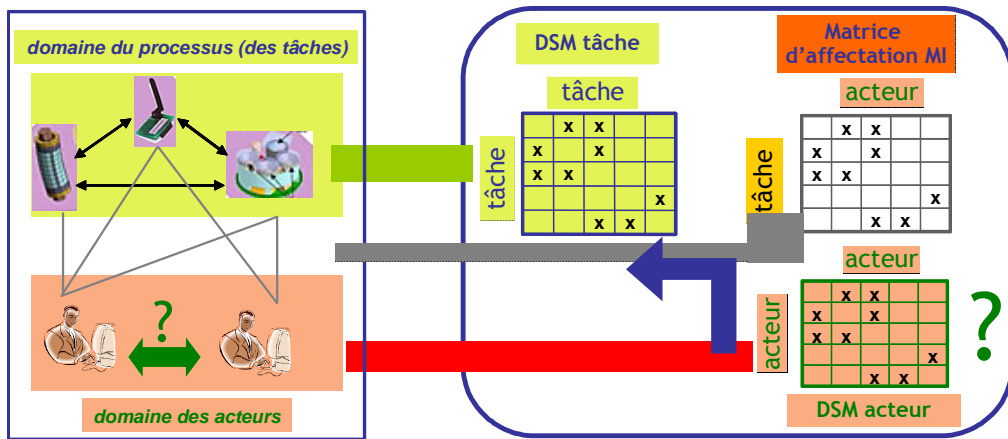


FIG. 5.3 : Principe de la méthode d'identification des interactions entre des acteurs

5.2.1 Qualités attendues

Les qualités attendues sont les suivantes :

- Garantir que les interfaces entre les tâches seront bien étudiées grâce à des interactions appropriées entre les acteurs du projet (la DSM acteur doit permettre de couvrir chaque interaction entre les tâches existant dans la DSM tâche).
- Minimiser les besoins d'interactions entre les acteurs dans le projet de conception (la DSM acteur doit contenir le plus de zéro possible, ce qui signifie que les efforts de communication et de coordination seront minimisés sur le projet).

5.2.2 Formule matricielle représentant le couplage de deux domaines

Notons que nous allons utiliser deux domaines : domaine A (composé d'élément A_i) et domaine B (composé d'élément B_j) et trois matrices : la DSM A représente la relation entre des éléments du domaine A (interactions $A_i - A_i$), la DSM B représente la relation entre des éléments du domaine B (interactions $B_j - B_j$) et la MI A-B représente la relation inter-domaine des domaines A et B (interactions $A_i - B_j$).

Les deux éléments (B_j et $B_{j'}$) peuvent s'avérer interdépendants par l'intermédiaire de trois relations : l'interaction entre l'élément $B_{j'}$ et l'élément A_i , l'interaction entre l'élément A_i et l'élément A_i et l'interaction entre l'élément A_i et l'élément B_j . Cette dépendance est représentée dans la figure (FIG. 5.4).

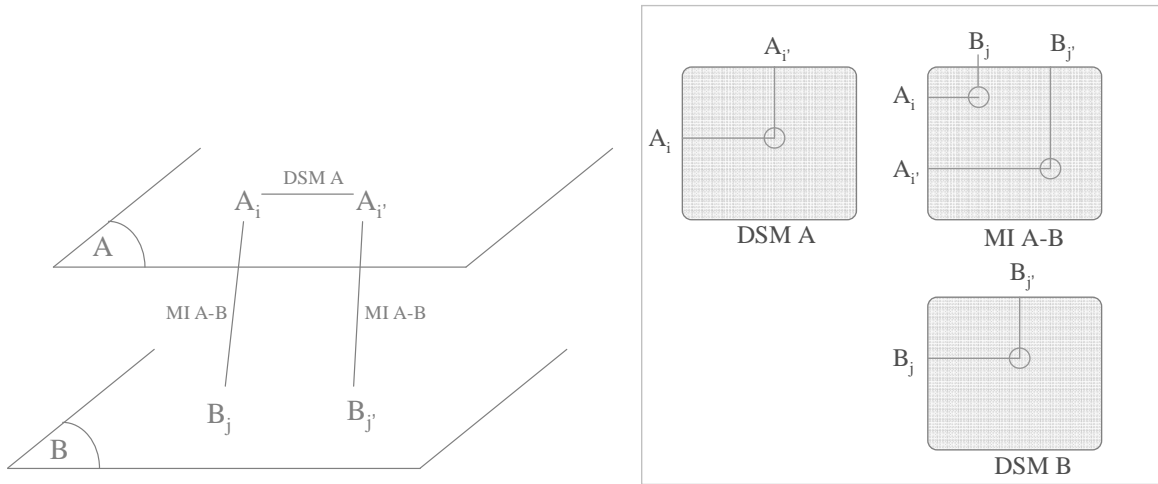


FIG. 5.4: Identification d'une interaction du domaine B à travers le domaine A

Cette relation permet d'identifier l'intensité entre deux éléments quelconques de B (par exemple l'intensité entre B_j et $B_{j'}$). Cette relation peut s'interpréter par une fonction linéaire de la manière suivante (Eq- 5-1).

$$DSM(B_j - B_{j'}) = \mu \sum_{i=1}^M \sum_{i'=1}^M \left[MI(B_j - A_i) \times DSM(A_i - A_{i'}) \times MI(A_{i'} - B_{j'}) \right],$$

$$\forall j = 1, \dots, N, \forall j' = 1, \dots, N$$

Eq- 5-1

A noter que μ est un facteur permettant de normaliser la matrice $DSM(B)$ de sorte que ses valeurs valent entre $[0;1]$. Cette formule peut se traduire sous forme de produit matriciel (Eq- 5-2).

$$DSM(B) = \mu \times MI(A-B) \times DSM(A) \times MI(A-B)$$

Eq- 5-2

De façon symétrique, nous faisons l'hypothèse d'une combinaison linéaire des matrices $DSM(B)$ et $MI(A-B)$ pour simuler les couplages entre tout A_i et $A_{i'}$. Nous pouvons montrer que

$$DSM_{sim}(A) = \frac{MI(A-B) \times DSM(B) \times MI(A-B)'}{\beta}$$

Eq- 5-3

Cette équation (Eq- 5-3) présente une relation matricielle qui est symétrique de l'équation (Eq- 5-2). Cette nouvelle relation matricielle permet d'identifier les interactions dans la matrice $DSM(A)$ simulée (appelé $DSM_{sim}(A)$) dans le sens inverse à partir des interactions inter-domaine ($DSM(B)$) et intra-domaines ($MI(A-B)$). L'équation (Eq- 5-3) nous permettra

de vérifier si les interactions dans la matrice $DSM_{sim}(A)$ restent cohérentes avec les valeurs dans la $DSM(A)$ d'origine.

Notons que, en calculant la matrice $DSM_{sim}(A)$ par cette combinaison linéaire (Eq- 5-3), la valeur $MI(A-B) \times DSM(B) \times MI(A-B)'$ peut dépasser 1. Le paramètre β permet de normaliser la matrice $DSM_{sim}(A)$ (valeurs inférieures à 1) et de faire en sorte que ses valeurs soient proches de la $DSM(A)$ d'origine.

Le choix de ces paramètres (μ et β) est estimé au cas par cas. Ils sont obtenus "par réglage" à partir d'un certain nombre de simulations.

5.2.3 Méthode d'optimisation du couplage pour l'identification de l'architecture des acteurs

Le chef de projet doit identifier le regroupement des acteurs pour chaque projet. Il pourrait dans ce cas identifier l'architecture des acteurs, en utilisant la solution d'affectation obtenue par la démarche d'affectation multi-périodes et l'architecture des tâches de conception qu'il aurait préalablement construite. C'est en réponse à cette problématique que nous proposons dans cette partie une méthode d'identification des architectures des acteurs.

Nous appliquons la formule matricielle présentée précédemment pour simuler les interactions entre les tâches (DSM tâche) dans le cas où l'on connaît la DSM acteur et la MI tâche-acteur. La figure ci-dessous présente graphiquement la démarche adoptée (Eq- 5-5).

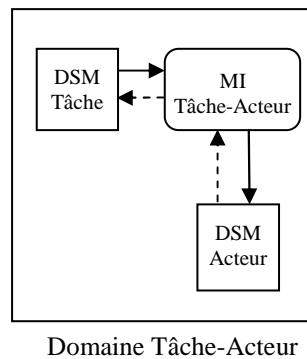


FIG. 5.5 : Couplage des matrices

En utilisant l'équation (Eq- 5-3) et en adaptant les notations, nous obtenons la $DSM_{sim}(T)$ de la façon suivante.

$$DSM_{sim}(T) = \frac{MI(T-A) \times DSM(A) \times MI'(T-A)}{\beta}$$

L'hypothèse qui est ainsi posée est que l'intensité des interactions entre les tâches dans la matrice $DSM_{sim}(T)$ s'explique essentiellement par les couplages existant entre les acteurs ($DSM(A)$) et par la contribution de ces acteurs dans la réalisation des tâches ($MI(T-A)$).

5.2.3.1 Formulation mathématique du problème

La formulation mathématique que nous allons présenter cherche à optimiser deux critères : critère de conservation de l'architecture des tâches (DSM tâche) et critère permettant de maximiser les interactions entre les acteurs (DSM acteur).

- Le critère de conservation de l'architecture des tâches garantit que les interfaces entre les tâches sont bien étudiées grâce à des interactions appropriées entre les acteurs. Ce critère cherche à minimiser la différence terme à terme entre la DSM tâche construite par le chef de projet (DSM(T)) et la DSM tâche simulée ($DSM_{sim}(T)$). Le fait de minimiser la différence revient à maximiser la similitude entre ces deux matrices, et donc d'en assurer la cohérence. Ce critère s'écrit comme suit (Eq- 5-4).

$$\sum_{i=1}^M \sum_{i'=1}^M \left(DSM_{sim}(T)_{i,i'} - DSM(T)_{i,i'} \right)^2$$

Eq- 5-4

avec

$$DSM_{sim}(T) = \frac{MI(T-A) \times DSM(A) \times MI'(T-A)}{\beta}$$

Eq- 5-5

La DSM tâche simulée ($DSM_{sim}(T)$) est obtenue par combinaison linéaire. Elle se décrit par l'équation (Eq- 5-5). Le paramètre β est utilisé comme paramètre de normalisation de la DSM tâche simulée pour qu'elle soit comparable avec la DSM tâche initiale. Normalement, les valeurs dans la DSM tâche initiale sont comprises entre 0 et 1. En calculant DSM tâche simulée par cette fonction linéaire, la valeur $MI(T-A) \times DSM(A) \times MI'(T-A)$ peut dépasser la valeur 1. Donc le paramètre β sera utilisé pour ramener la DSM Tâche simulée dans cet intervalle [0 et 1]. Le choix de ce paramètre est estimé au cas par cas.

- Le critère de minimisation des besoins d'interactions entre les acteurs (ou maximisation du nombre de 0 dans la DSM acteur, ce qui permet de rendre la DSM acteur la plus éparse possible). [Candes et Tao, 2005] se sont intéressés au fait de maximiser le nombre de 0 d'un vecteur ou d'une matrice. Ils ont cherché à construire la matrice la plus éparse possible. Ils ont démontré un théorème disant que le fait de maximiser le nombre de 0 d'une matrice est équivalent au fait de minimiser la somme des valeurs de la matrice. En tenant compte de cette conclusion de Candes-Tao, nous définissons ce critère par la minimisation de la somme des interactions entre les acteurs de la matrice DSM acteur. L'intérêt du critère de minimisation des valeurs dans la matrice est qu'elle est linéaire et convexe alors que la maximisation du nombre de 0 est une fonction non linéaire. Par ailleurs, elle s'interprète favorablement comme l'effort global de coordination dans le projet.

$$\sum_{j=1}^N \sum_{j'=1}^N DSM(A)_{j,j'}$$

L'équation (Eq- 5-6) présente la fonction objectif du problème prenant en compte ces deux critères.

$$\text{Minimiser} \left[\sum_{i=1}^M \sum_{i'=1}^M \left(\text{DSM}_{\text{sim}}(\text{T})_{i,i'} - \text{DSM}(\text{T})_{i,i'} \right)^2 + \lambda \sum_{j=1}^N \sum_{j'=1}^N \text{DSM}(\text{A})_{j,j'} \right]$$

Eq- 5-6

Sous les contraintes

$$\text{DSM}(\text{A})_{j,j'} = 0, \quad [j = j', (\forall j, \forall j' = 1, \dots, N)]$$

Eq- 5-7

$$0 \leq \text{DSM}(\text{A})_{j,j'} \leq 1, \quad j \neq j', \quad \forall j, j' = 1, \dots, N$$

Eq- 5-8

$$\text{DSM}_{\text{sim}}(\text{T})_{i,i'} \geq \text{DSM}(\text{T})_{i,i'}, \quad (\forall i, \forall i' = 1, \dots, M)$$

Eq- 5-9

La première partie de la fonction objectif minimise le critère de conservation de l'architecture des tâches dans la DSM tâche et la deuxième partie de la fonction objectif vise à minimiser le nombre d'interactions entre les acteurs du projet dans la DSM acteur, c'est-à-dire à réduire le coût lié aux efforts de coordination entre acteurs. La première contrainte (Eq- 5-7) impose que les valeurs sur la diagonale de DSM acteur soient nulles. En effet, nous avons considéré qu'il était possible qu'un acteur travaille sur plusieurs composants, mais que ceux-ci ne sont pas obligatoirement en interaction. Cette contrainte permet donc d'obtenir la DSM acteur simulée proche de la DSM acteur initiale (favoriser le premier critère). La deuxième contrainte (Eq- 5-8) impose que les autres valeurs soient comprises entre 0 et 1. La dernière contrainte (Eq- 5-9) impose que la valeur dans la DSM tâche simulée soit toujours plus forte que la valeur dans la DSM tâche initiale afin de ne pas perdre d'interactions entre les tâches.

Dans la méthode d'optimisation de l'architecture des acteurs, nous avons défini le paramètre d'homogénéisation λ et le paramètre de normalisation β . Pour garantir que les interfaces entre les tâches soient bien prises en compte, le paramètre β est utilisé comme paramètre de normalisation de la DSM tâche simulée. Celle-ci est alors comparable avec la DSM tâche initiale. Pour permettre à l'algorithme de minimiser les deux coûts, le paramètre λ est utilisé pour équilibrer l'importance des deux critères dans la fonction objectif.

En raison du premier critère, nous avons construit un problème de type programmation quadratique. Nous avons alors transformé la formulation du problème pour le mettre sous la forme matricielle suivante :

$$\text{Min}_{\mathbf{x} \in \mathcal{R}^n} (0.5 * \mathbf{x}' * \mathbf{H} * \mathbf{x} + \mathbf{f}' * \mathbf{x}) \quad \text{sous la contrainte : } \mathbf{A} * \mathbf{x} \leq \mathbf{B}$$

où \mathbf{x} est un vecteur colonne correspondant à la concaténation des colonnes de la matrice DSM A.

Nous avons démontré que, dans le cas présent, H est une matrice définie positive stricte, ce qui nous permet de conclure à l'unicité de la solution à ce problème. Par ailleurs, la fonction objectif et les contraintes étant convexes, nous avons utilisé le solveur de programmation quadratique convexe que propose la toolbox d'optimisation de Matlab (fonction Quadprog). La démonstration de la convexité du problème se trouve dans l'annexe 6.

5.2.3.2 Test du principe

Nous avons testé cette formulation de problème sur un projet de conception d'une boîte de vitesse robotisée. Les données ont été recueillies à partir d'interviews réalisés auprès des membres du projet et consolidées par le chef de projet.

La description de l'exemple est présentée ci-dessous. Le tableau (TAB. 5.1) résume les principales abréviations utilisées dans les matrices que nous avons construites.

FS VOL	Fonction Système "Volumes fonctionnels"	PAM	Chef de projet
FS TPU	Fonction Système "Transmission de Puissance"	PMIV	Responsable Intégration et validation
FS ELU	Fonction Système reprise d'Effort et Lubrification	AF	Architecte fonctionnel
FS COU	Fonction Système Couplage	CdP	Chargé de projet de conception
FS COM	Fonction Système Commutation		
CDI	Commande Interne		
SYN	Synchroniseur		
EMB	Embrayage		
CIE	Commande Interne d'Embrayage		
DIFF	Différentiel		
MEI	Mécanique Interne		
CART	Carter		
ACT	Actionneur		

TAB. 5.1. Liste des éléments utilisés et leurs abréviations

Ci-dessous se trouvent les DSM tâches et MI tâches-acteurs.

	FS VOL	FS TPU	FS ELU	FS COU	FS COM	CDI	SYN	EMB	CIE	DIFF	MEI	CART	ACT
FS VOL	1	0,7	0,8	0,7	0,9	0,6	0,7	0,9	0,6	0,4	0,7	0,8	0,8
FS TPU	0,7	1	0,7	0	0	0	0	0	0	0	0,8	0,9	0
FS ELU	0,8	0,7	1	0,7	0,8	0	0,5	0,5	0,6	0	0	0,9	0,9
FS COU	0,7	0	0,7	1	0,5	0	0	0,5	0,9	0,7	0	0	0
FS COM	0,9	0	0,8	0,5	1	0,7	0,7	0,8	0	0	0	0	0
CDI	0,6	0	0	0	0,7	1	0,9	0,8	0	0	0	0	0,6
SYN	0,7	0	0,5	0	0,7	0,9	1	0	0	0	0	0,8	0
EMB	0,9	0	0,5	0,5	0,8	0,8	0	1	0	0,7	0	0	0,8
CIE	0,6	0	0,6	0,9	0	0	0	0	1	0,8	0	0,6	0
DIFF	0,4	0	0	0,7	0	0	0	0,7	0,8	1	0	0	0,5
MEI	0,7	0,8	0	0	0	0	0	0	0	0	1	0,7	0,8
CART	0,8	0,9	0,9	0	0	0	0,8	0	0,6	0	0,7	1	0,9
ACT	0,8	0	0,9	0	0	0,6	0	0,8	0	0,5	0,8	0,9	1

FIG. 5.6 : DSM tâches (13x13)

	PAM	PMIV	AF VOL	AF COM	CdP TRI	CdP SYN	CdP CDI	AF TPU	CdP DIFF	CdP MEI	AF COU	CdP EMB	CdP CIE	AF ELU	CdP CART	CdP LUB
FS VOL	0,5	0,5	0,9	0,8	0	0	0	0,7	0	0	0,7	0	0	0,7	0	0
FS TPU	0,5	0,5	0	0	0	0	0	0,9	0,8	0,8	0	0	0	0,5	0	0
FS ELU	0,5	0,5	0,7	0,5	0	0	0	0	0	0	0	0	0	0,9	0,8	0,8
FS COU	0,5	0,5	0	0	0,5	0	0	0	0	0	0,9	0,7	0,7	0	0	0
FS COM	0,5	0,5	0,7	0,9	0,9	0,8	0,8	0	0	0	0	0	0	0	0	0
CDI	0,5	0,6	0	0,8	0,5	0,5	0,9	0	0	0	0	0	0	0	0	0
SYN	0,5	0,6	0	0,8	0	0,9	0,5	0	0	0	0	0	0	0	0	0
EMB	0,9	0,9	0	0,9	0,9	0	0,5	0	0	0	0,3	0	0,5	0	0	0
CIE	0,5	0,6	0	0	0	0	0	0	0	0	0,7	0,9	0,5	0	0	0
DIFF	0,5	0,6	0	0	0,7	0	0	0	0	0	0,7	0,5	0,9	0	0	0
MEI	0,5	0,6	0	0	0	0	0	0,7	0,9	0,5	0	0	0	0	0	0
CART	0,5	0,6	0	0	0	0	0	0,7	0,5	0,9	0	0	0	0	0	0,8
ACT	0,5	0,6	0	0	0,5	0	0	0	0	0,5	0	0	0	0,9	0,9	0,8

FIG. 5.7: MI tâches-acteurs (13x16)

Soit $DSM_{opt}(A)_{(\lambda)}$ la DSM acteur optimale obtenue pour une valeur λ .

Nous avons d'abord fixé $\lambda = 1$, et nous avons trouvé une valeur appropriée pour β , $\beta = 8$. Nous obtenons comme résultat une $DSM_{opt}(A)_{(\lambda=1)}$ d'une densité de 36% (FIG. 5.8- coté gauche).

Nous avons ensuite augmenté la valeur de λ de façon à favoriser le second critère. Nous avons fixé $\lambda = 50$ pour obtenir une densité de 33%, et nous obtenons l'architecture de $DSM_{opt}(A)_{(\lambda=50)}$ (FIG. 5.8- coté droit).

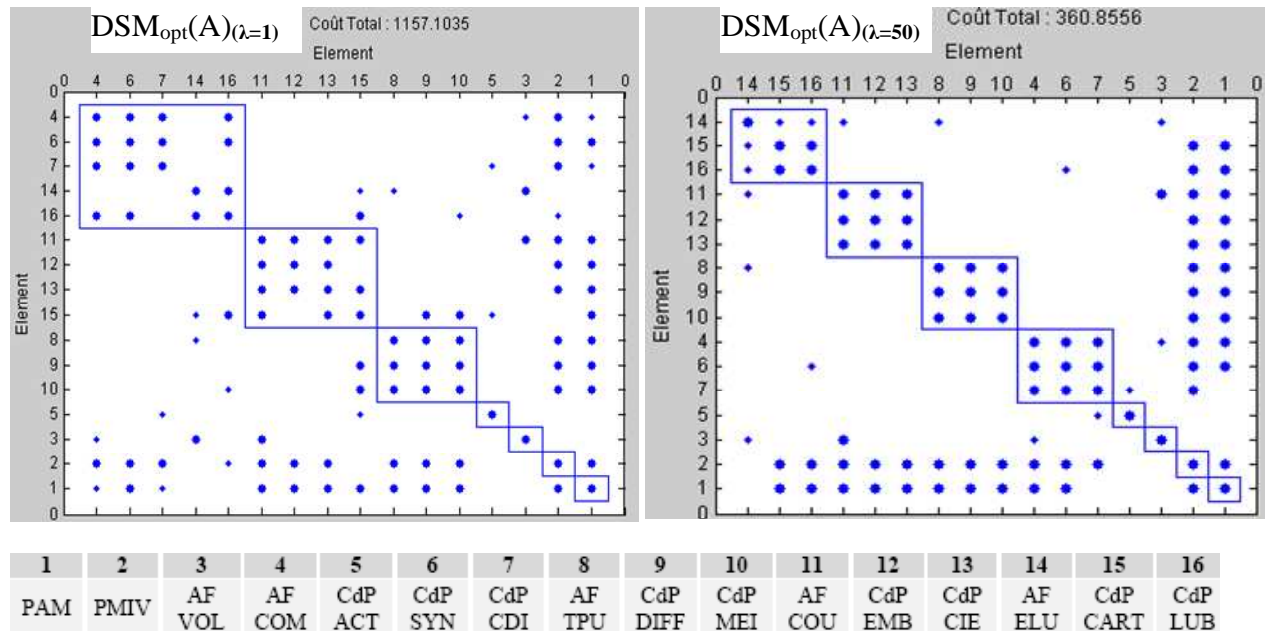


FIG. 5.8: Architecture des acteurs après clustering des $DSM_{opt}(A)_{(\lambda)}$

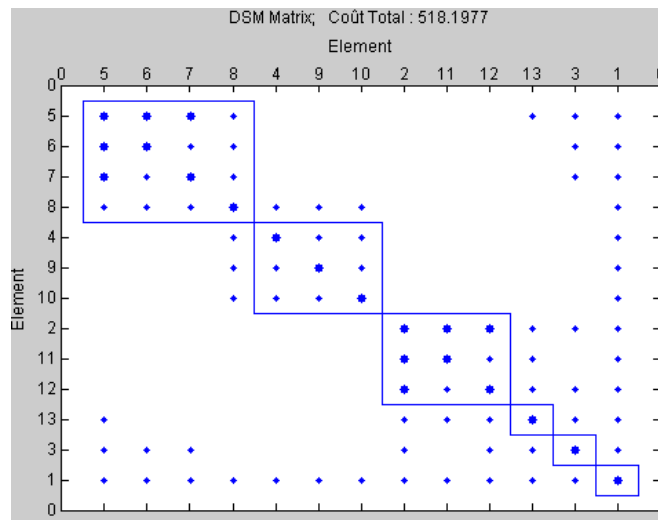
Notons la relative stabilité de l'architecture obtenue avec $\lambda = 1$ ou $\lambda = 50$. En comparant les deux architectures, nous observons que la principale différence provient de la position de l'acteur 15. L'intérêt de la seconde solution est de proposer des modules de plus petite taille et

mieux découplés. Nous constatons que, pour la $DSM_{opt}(A)_{(\lambda=50)}$, la solution optimale est obtenue en désignant l'acteur 5 (chargé de projet de l'actionneur) comme un acteur isolé, couplé avec l'acteur 7 (chargé de projet Commande Interne). Cependant, dans la réalité, il paraît intéressant de l'affecter dans le module composé de 4, 6 et 7. Nous trouvons ici que les acteurs 1 et 2 jouent un rôle d'intégrateur car ils sont en interaction avec un grand nombre de concepteurs. Les interactions de l'acteur 3 (Architecte fonctionnel Volumes) avec d'autres architectes fonctionnels (acteurs 4, 11, 14) sont aussi cohérentes.

5.2.3.3 Eléments de validation et synthèse

Cette architecture est satisfaisante par rapport à la réalité. Elle correspond aussi à l'architecture obtenue par [Harmel, 2007] en utilisant la méthode basée sur un processus flu. La densité est passée de 52% dans le cas de l'utilisation du processus flu à une densité de 33%, ce qui fait un gain de 35% sur le nombre d'interactions.

Nous avons vérifié que la matrice $DSM_{sim}(T)$ restait cohérente avec la réalité et avec les résultats obtenus par le processus flu (FIG. 5.9). Il apparaît que l'architecture de cette matrice simulée est identique à la matrice $DSM(T)$ construite par l'architecte et que les valeurs sont proches (dispersion faible).



1	2	3	4	5	6	7	8	9	10	11	12	13
FS VOL	FS TPU	FS ELU	FS COU	FS COM	CDI	SYN	ACT	EMB	CIE	DIFF	MEI	CART

FIG. 5.9 : Architecture des tâches après clustering de $DSM_{sim}(T)_{(\lambda=50)}$

En synthèse, nous présentons dans le tableau (TAB. 5.2) une estimation qualitative de la satisfaction des exigences décrites au (§5.2.1).

Exigences	Estimation qualitative	Travail supplémentaire
E1 : Garantir que les interfaces entre les tâches seront bien étudiées grâce à des interactions appropriées entre les acteurs du projet.	++	Tester la méthode sur des matrices denses (architectures intégrales)
E2 : Minimiser les besoins d'interactions entre les acteurs dans le projet de conception	++	Réaliser une analyse de sensibilité

TAB. 5.2 : Estimation qualitative de l'optimisation de l'architecture des acteurs

5.2.4 Conclusion

Nous avons présenté dans cette partie (§5.2) une méthode permettant l'optimisation de la matrice des acteurs connaissant la DSM tâches et la matrice d'affectation tâches-acteurs. Cette méthode est générique et permet d'optimiser les couplages entre deux domaines.

Pour tester le principe de la méthode, nous avons appliqué la méthode d'optimisation dans le cas d'un projet de conception d'une nouvelle boîte de vitesse robotisée. Nous avons trouvé des résultats satisfaisants. Cette méthode est générique et nous allons l'appliquer dans le cas de projets de conception de produits mécatroniques.

5.3 Constitution d'équipes contribuant à la structuration du projet et au pilotage des compétences

Dans la seconde partie de ce chapitre, nous présentons une démarche de constitution d'équipes se basant sur l'optimisation de l'architecture des acteurs (§5.2) et sur l'affectation multi-périodes avec prise en compte de l'évolution des compétences (chapitre 4).

Le but de cette méthode est de traiter à la fois le problème de structuration d'un projet (identification de l'architecture des acteurs) et le problème de pilotage des compétences (satisfaction d'objectifs de développement de compétences). Ce travail vise à fournir à des directeurs de R&D ou des directeurs de programme un outil qui pourrait être interfacé à une gestion de projet classique (fournissant une planification des tâches par période) pour proposer des affectations de tâches à des acteurs et une constitution d'équipes qui minimisent globalement les coûts de projets en R&D.

5.3.1 Qualités attendues

Les qualités attendues sont les suivantes :

- E1 : Affecter des tâches multi-périodes et multi-projets en tenant compte d'une loi d'évolution des compétences
- E2 : Minimiser le coût salarial lié aux niveaux de compétence
- E3 : Minimiser les pénalités financières dues à la non-satisfaction des objectifs de compétence
- E4 : Garantir que les interfaces entre les composants du produit seront bien étudiées grâce à des interactions appropriées entre les acteurs du projet.

- E5 : Minimiser les besoins d'interactions entre les acteurs dans le projet de conception

5.3.2 Structure générale

La vue externe représentée dans la figure (FIG. 5.10) met en évidence les données d'entrée et de sortie suivantes :

- Les données d'entrée : tâche ($T_i^{k,l}$), acteur (A_j), connaissance (C_c), niveau de compétence requis concernant la tâche particulière ($rq_i^{k,l}$), objectif de compétences (O_i), DSM tâche (DSM (T)).
- Les données de sortie : MI-affectation ($x_{i,j}^{k,l}$), évolution des compétences ($r_{i,j}^{k,l}$), nombre d'acteurs compétents après l'affectation (R_i), architecture des acteurs du projet (DSM acteur regroupé).

En comparaison avec la démarche proposée en (§4.3.3), il y a une nouvelle donnée d'entrée, la DSM tâche et un nouveau livrable à la sortie, l'architecture des acteurs. Concernant la DSM tâche, nous supposons que les interactions entre les tâches de cette matrice sont de nature statique³⁹. La dépendance entre deux tâches peut être représentée par la fréquence ou le débit du flux d'information ou la criticité de ce flux. Les valeurs de l'intensité entre deux tâches varient entre 0 et 1, où 1 représente un lien de dépendance extrêmement fort.

La vue interne du système est présentée ci-dessous. Quelques étapes ont été rajoutées (présentées en lignes pointillées) par rapport à la démarche d'affectation multi-périodes présentée dans le chapitre 4.

Les différentes étapes sont les suivantes :

- Phase I : Caractérisation de la tâche et de l'acteur (recueil de données)
- Phase II : Calculs de l'évolution du niveau de maîtrise de la connaissance et de l'indicateur de similarité a priori
- Phase III : Optimisation de l'architecture des acteurs
- Phase IV : Recherche de la solution optimale
- Phase V : Calcul de la similarité a posteriori permettant d'estimer l'évolution des compétences à la fin de l'horizon
- Phase VI : Construction de l'architecture des acteurs (algorithme de clustering)

Les phases concernant le pilotage des compétences (I, II, V) ont été déjà présentées dans le chapitre 4 (§4.3.3.1-4.3.3.3, §4.3.3.5). Nous détaillerons dans la partie suivante (§5.3.3) seulement les phases III, IV et VI concernant l'optimisation de l'architecture des acteurs et l'optimisation globale du problème.

La phase III permet de préparer la DSM tâche et la MI-affectations pour l'optimisation de l'architecture des acteurs du projet. La phase IV correspond à la formulation mathématique et l'optimisation du problème en global. Cette étape permet d'obtenir les résultats principaux de la constitution d'équipes (matrice d'affectation et DSM acteur). La phase VI présentera le

³⁹ En général, les DSM tâche ou processus peuvent être temporelles ou statiques, dans ce cas nous avons adopté les DSM –tâche statique

regroupement des acteurs (DSM acteur obtenue par la phase IV) en équipes (architecture des acteurs) en se basant sur un algorithme de clustering.

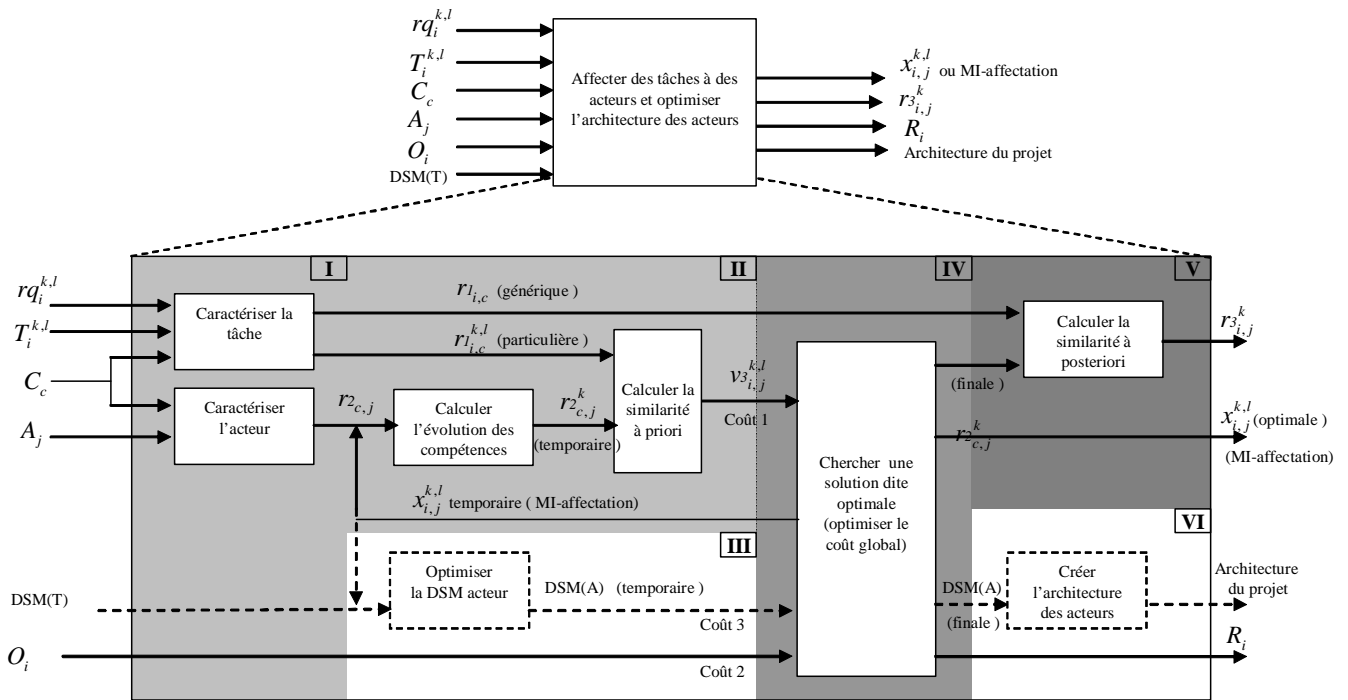


FIG. 5.10 : Démarche globale de la constitution d'équipes

5.3.3 Mise en œuvre et test de la méthode

Nous présentons les différentes étapes constituant cette nouvelle méthode et leur mise en œuvre sur deux projets de conception d'une capsule endoscopique (projets européens). Le projet 1 est un projet de capsule endoscopique pour la thérapie des tumeurs intestinales. Le projet 2 concerne également la conception d'une capsule endoscopique mais avec un capteur de technologie différente.

Nous rappelons que le problème que nous traitons (§4.3.2.2) est constitué par deux projets qui se déroulent simultanément (projet 1 et projet 2) sur trois périodes d'affectation ($k=1, k=2, k=3$).

5.3.3.1 Phase III : Optimiser l'architecture des acteurs

5.3.3.1.a Principe

Dans le cas multi-projets, chaque projet a une architecture des acteurs (DSM acteurs) qui lui est propre. Pour identifier cette architecture, nous avons besoin de construire une DSM tâche et une matrice d'affectation. Pour ce faire, il s'agit de prendre certaines précautions pour construire ces deux matrices d'entrée.

Nous distinguons dans cette partie trois étapes pour l'optimisation de l'architecture des acteurs :

- Etape de préparation des DSM tâches,
- Etape de préparation des MI-affectations,
- Etape d'optimisation permettant de générer la DSM acteur en utilisant la méthode d'optimisation de couplage proposée dans la partie (§5.2).

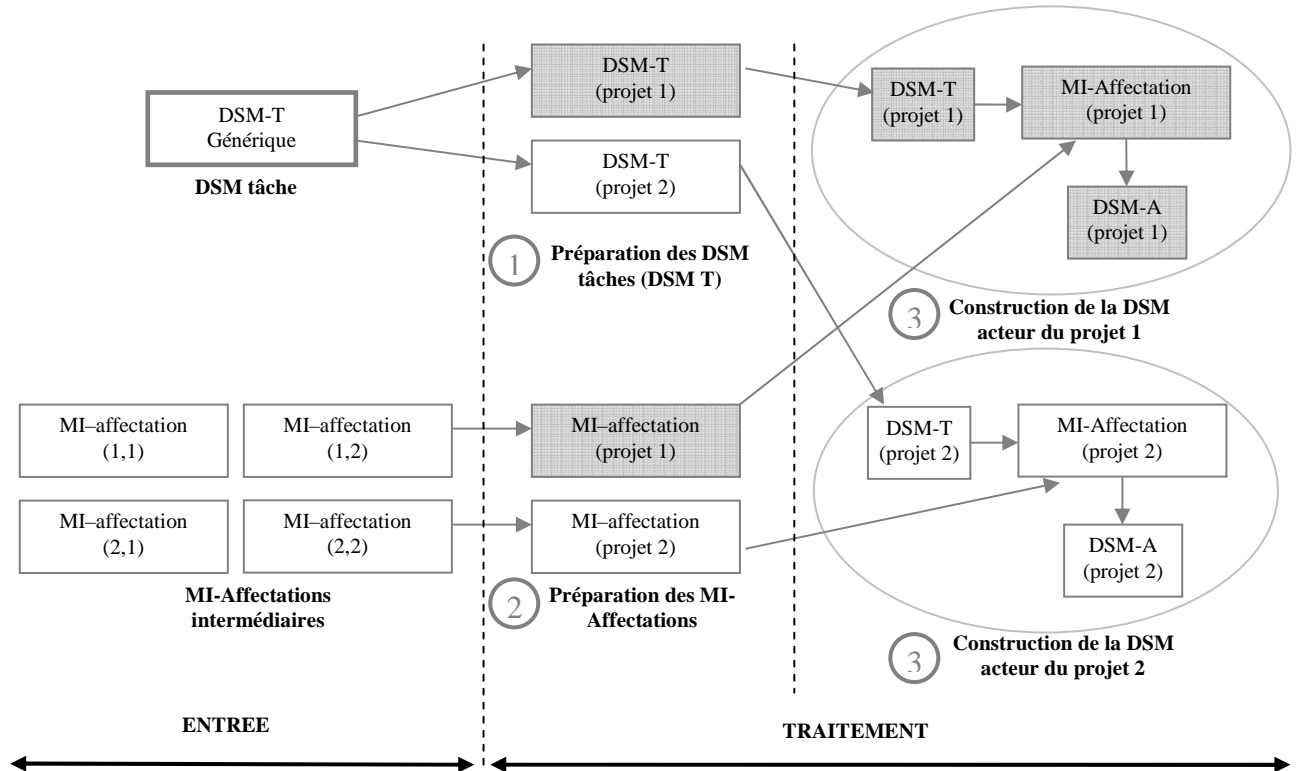


FIG. 5.11 : Préparation des DSM tâches et des MI-affectations

Nous détaillons chacune des étapes ci-dessous :

- Etape 1: préparation des DSM tâches (FIG. 5.11- au centre) : Pour créer la DSM tâche de chaque projet (DSM tâche du projet 1 et DSM tâche du projet 2),
 - nous retirons de la DSM Tâche les tâches qui n'existent pas dans le projet. Nous les filtrons par la valeur du niveau requis par la tâche ($rq_i^{k,l}$). Si la valeur $rq_i^{k,l}$ de la tâche considérée est égale à 0, cette tâche n'existe pas dans le projet.
 - nous enlevons les tâches de management et d'intégration du système (par exemple, définition des spécifications du système, conception de l'architecture du système, intégration des sous-systèmes) qui correspondent à des interactions de coordination et validation du projet. De plus, ces tâches ont tendance à tout coupler car elles sont en interaction avec toutes les autres. Par ailleurs, la nature des interactions entre les autres tâches est différente puisqu'elles sont liées à la conception technique et nécessitent des interactions techniques.
- Etape 2 : préparation des MI-affectations (FIG. 5.11- au centre) : partant du résultat intermédiaire d'affectation (obtenu avec la démarche d'affectation proposée au chapitre précédent), nous préparons la MI-affectation pour chaque projet par les étapes suivantes :

- Faire correspondre la liste des tâches dans la MI-affectation avec celle de la DSM tâche.
- Agréger en une seule matrice, appelée matrice d'affectation agrégée, les K MI-Affectations (une matrice pour chaque période) en pondérant les contributions par la charge de chaque tâche i . Cette matrice représente la contribution agrégée des acteurs j à chaque projet l sur l'ensemble de l'horizon d'affectation.

$$MI_{\text{agrégée}}(T-A)_{i,j}^l = \frac{\sum_{k=1}^K [MI(T-A)_{i,j}^{k,l} \times L_i^{k,l}]}{\sum_{k=1}^K L_i^{k,l}},$$

$$\forall i = 1, \dots, M, \forall j = 1, \dots, N, \forall l = 1, \dots, P$$

Eq- 5-10

Par exemple, dans la figure (FIG. 5.12), nous considérons le projet numéro "1". Il s'agit de deux matrices d'affectations de deux périodes ($k=1$ et $k=2$). Les tâches particulières $T_i^{k,l}$ de type 1 ont été affectées aux deux acteurs, A1 et A2. La charge induite ($L_i^{k,l}$) de la tâche particulière planifiée à la période 1 est de 3 mois ($L_1^{1,1} = 3$) et de celle planifiée à la période 2 est de 2 mois ($L_1^{2,1} = 2$). Considérons l'acteur A1, les valeurs de contribution $x_{i,j}^{k,l}$ de cet acteur pour réaliser la tâche planifiée à la période $k=1$ et la période $k=2$ sont à 0,4 et 0,2 respectivement. En utilisant la formule (Eq- 5-10), nous obtenons la valeur de contribution agrégée de cet acteur de 0,32 (ou $[0,4*3+0,2*2]/5$).

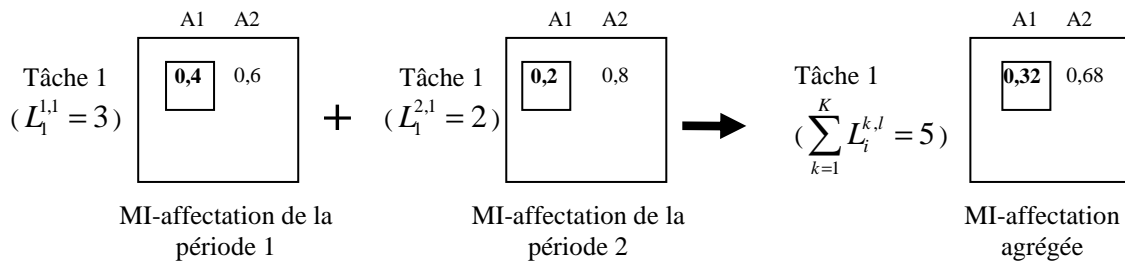


FIG. 5.12 : Exemple de l'agrégation des MI-affectations

- Etape 3 : optimisation (FIG. 5.11- coté droit) : quand la préparation de la DSM tâche et de la MI-affectation agrégée pour chaque projet est réalisée, la DSM acteur pourra être optimisée selon la méthode d'optimisation de l'architecture des acteurs proposée dans la partie (§5.2).

5.3.3.1.b Mise en œuvre

Ci-dessous, nous présentons la mise en œuvre de ce principe.

- Préparation de la DSM tâche

Tout d'abord, nous avons créé une matrice DSM des tâches génériques. Cette matrice se compose de toutes les tâches de référence des projets considérés. Dans le cas de la capsule endoscopique, la DSM tâche comprend 15 tâches génériques.

Nous distinguons deux types de flux d'information modélisés : flux de coordination (présents essentiellement dans les tâches d'intégration et les tâches de management, T1-T4) et flux de conception technique (présents dans les tâches de conception, T5-T15).

L'intensité des interactions entre les tâches est estimée à partir de l'importance des flux d'information. La valeur de l'intensité des interactions est un nombre réel comprise entre 0 et 1. Si le flux d'information entre deux tâches est fort alors la valeur est proche de 1. Dans le cas contraire, elle est proche de 0. Dans le tableau ci-dessous, nous avons défini quatre niveaux d'importance de flux d'information (0 ou "cas vide"=non existant ; 0,3=faible ; 0,6=moyen ; 0,9=important).

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
T1	1	0,6	0,3	0,3											
T2	0,6	1	0,9	0,3											
T3	0,3	0,9	1	0,3											
T4	0,3	0,3	0,3	1											
T5					1					0,3					
T6						1				0,3		0,6			
T7					0,9		1			0,6		0,3	0,3	0,9	0,9
T8								1		0,6	0,6				
T9									1	0,9					
T10					0,6	0,3	0,6	0,3	0,9	1	0,3	0,6		0,6	0,6
T11								0,6		0,3	1				
T12					0,6	0,9					0,3	1	0,6	0,6	0,6
T13								0,6		0,3			1		
T14										0,3				1	
T15										0,3					1

FIG. 5.13 : DSM tâche générique (15x15)

Nous construisons ensuite la DSM tâche pour chaque projet selon la procédure précisée dans la section 5.3.3.1.a. Nous obtenons la DSM tâche du projet "1" qui se compose de 9 tâches de conception et la DSM tâche du projet "2" qui contient 8 tâches (FIG. 5.14).

	T5	T6	T7	T8	T9	T10	T11	T12	T13
T5	1	0	0	0	0	0,3	0	0	0
T6	0	1	0	0	0	0,3	0	0,6	0
T7	0,9	0	1	0	0	0,6	0	0,3	0,3
T8	0	0	0	1	0	0,6	0,6	0	0
T9	0	0	0	0	1	0,9	0	0	0
T10	0,6	0,3	0,6	0,3	0,9	1	0,3	0,6	0
T11	0	0	0	0,6	0	0,3	1	0	0
T12	0,6	0,9	0	0	0	0	0,3	1	0,6
T13	0	0	0	0,6	0	0,3	0	0	1

DSM tâche du projet 1 (9x9)

DSM tâche du projet 2 (8x8)

FIG. 5.14 : DSM tâche pour chaque projet

- Préparation de la MI-affectation

Il s'agit d'utiliser la liste de tâches de la "DSM tâche" pour construire la MI-affectation.

Dans ce problème d'optimisation, les MI-affectation initiales sont obtenues par programmation linéaire (solution admissible) de la même façon que dans la partie (§4.3.6.2). Les contributions des acteurs contenues dans les MI-affectation sont ensuite agrégées selon la procédure décrite dans la partie 5.3.3.1.a. Les MI-affectations agrégées "initiales" des projets "1" et "2" sont représentées ci-dessous.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T5						0,31						0,29					0,40			
T6																0,67		0,33		
T7												0,54								0,46
T8														1,00						
T9						0,57		0,43												
T10																	1,00			
T11												1,00								
T12								1,00												
T13								0,39									0,61			

FIG. 5.15 : MI-affectation agrégée (initiale) du projet 1 ($l=1$)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T6																0,93	0,07			
T8													0,36	0,64						
T9						0,79		0,21												
T10																	1,00			
T11												1,00								
T12								1,00												
T14													0,76						0,24	
T15								0,09											0,91	

FIG. 5.16 : MI-affectation agrégée (initiale) du projet 2 ($l=2$)

- Optimisation de la DSM acteur

En utilisant la méthode d'optimisation de couplage des architectures proposées, nous obtenons ci-dessous les DSM acteurs pour chaque projet. Néanmoins, ces DSM acteurs sont provisoires car elles sont construites par rapport à la MI-affectation initiale. Les DSM acteurs finales seront obtenues par le couplage entre la MI-affectation finale et la DSM tâche. La DSM tâche reste inchangée durant la simulation.

	A6	A8	A12	A13	A14	A16	A17	A18	A20
A6		0,6	0,3	0,5				0,7	0,2
A8	0,6					0,1	0,6	0,1	
A12	0,3			0,3				0,4	0,6
A13	0,5		0,3					0,6	0,2
A14									
A16		0,1						0,6	
A17		0,6							
A18	0,7	0,1	0,4	0,6		0,6			0,3
A20	0,2		0,6	0,2				0,3	

	A6	A8	A12	A13	A14	A16	A17	A18
A6		0,6						0,1
A8	0,6							0,6
A12								0,6
A13					0,6			
A14				0,6				
A16							0,6	
A17	0,1					0,6		
A18		0,6	0,6					

(a) DSM acteur (projet 1)

(b) DSM acteur (projet 2)

FIG. 5.17 : DSM acteurs initiales

5.3.3.2 Phase IV : Recherche de la solution optimale

Cette phase vise à optimiser le coût de la réalisation des tâches.

Critères étudiés

Nous avons modélisé les critères suivants.

- Critères liés aux niveaux de compétence, noté "coût 1" :
 - Minimiser le coût provenant du temps de travail effectué par des salariés.
 - Minimiser le coût provenant de la participation des tuteurs.
- Critère liés à la non-satisfaction des objectifs de compétence, noté "coût 2":
 - Minimiser les pénalités financières dues à l'écart sur l'objectif de compétences.
- Critères liés à l'optimisation de l'architecture des acteurs, noté "coût 3":
 - Minimiser l'écart entre les valeurs dans la DSM tâche et la DSM tâche simulée (garantir que les interfaces entre les composants du produit seront bien étudiées).
 - Minimiser les besoins d'interactions techniques entre les acteurs dans le projet.

Modèle

$$F1 = \left[\sum_{k=1}^K \sum_{l=1}^P \sum_{i=1}^M \sum_{j=1}^N (C1_{i,j}^{k,l} + C2_{i,j}^{k,l}) x_{i,j}^{k,l} \right]$$

$$F2 = \left[\sum_{i=1}^M \varphi_i (O_i - R_i) \right]$$

$$F3 = \left[\sum_{i=1}^M \sum_{i'=1}^M \left(\text{DSM}_{\text{sim}}(T)_{i,i'} - \text{DSM}(T)_{i,i'} \right)^2 + \lambda \sum_{j=1}^N \sum_{j'=1}^N \text{DSM}(A)_{j,j'} \right]$$

Minimiser $\omega_1 F1 + \omega_2 F2 + \omega_3 F3$

$$C1_{i,j}^{k,l} = (\gamma_{i,j}^{k,l} \times L_i^{k,l}) \times S_j$$

$$C2_{i,j}^{k,l} = (\gamma_{i,j}^{k,l} - 1) \times L_i^{k,l} \times ST$$

$$DSM_{sim}(T) = \frac{MI(T-A) \times DSM(A) \times MI(T-A)}{\beta}$$

Sous les contraintes

$$\sum_{j=1}^N L_i^{k,l} x_{i,j}^{k,l} = L_i^{k,l}, \forall i = 1, \dots, M, \forall l = 1, \dots, P, \forall k = 1, \dots, K \quad (1)$$

$$\sum_{i=1}^M \sum_{l=1}^P \gamma_{i,j}^{k,l} L_i^{k,l} x_{i,j}^{k,l} \leq LM_{j,k}, \forall j = 1, \dots, N, \forall k = 1, \dots, K \quad (2)$$

$$0 \leq x_{i,j}^{k,l} \leq 1, \forall i, j, k, l, x_{i,j}^{k,l} \in R^+ \quad (3)$$

$$DSM(A)_{j,j'} = 0, \text{ pour } j = j', (\forall j, j' = 1, \dots, N) \quad (4)$$

$$0 \leq DSM(A)_{j,j'} \leq 1, \text{ pour } j \neq j', (\forall j, j' = 1, \dots, N) \quad (5)$$

$$DSM_{sim}(T)_{i,i'} \geq DSM(T)_{i,i'}, (\forall i, i' = 1, \dots, M) \quad (6)$$

$$\omega_1, \omega_2, \omega_3 \in R^+ \quad (7)$$

Cette fonction objectif vise à optimiser un compromis entre trois coûts : le coût salarial lié aux niveaux de compétence, le coût lié à la non-satisfaction des objectifs de compétence et le coût lié à l'optimisation de l'architecture des acteurs du projet. Pour que ces trois coûts soient comparables et rendus homogènes, nous avons défini des poids (ω_i). Aucune règle permettant de fixer la valeur des poids n'a été définie⁴⁰. Ils sont fixés arbitrairement, de façon à équilibrer les coûts, c'est-à-dire qu'ils aient un ordre de grandeur semblable⁴¹.

Les contraintes (1) et (2) assurent que la charge des tâches et la capacité des acteurs sont respectées. Les contraintes (4), (5) et (6) sont associées à l'optimisation de l'architecture du projet.

Principe de l'optimisation

Pour permettre d'optimiser tous les critères, l'algorithme va procéder en trois étapes de la manière suivante :

A une MI-affectation initiale :

- Première étape : Calculer les coûts 1 et 2

Cette étape vise à optimiser le coût salarial lié aux niveaux de compétence (FIG. 5.10-coût-1) et le coût lié à la non-satisfaction des objectifs de compétence (FIG. 5.10-coût-2). Ces coûts sont calculés avec la méthode présentée dans le chapitre 4, basée sur la méthode du recuit simulé.

⁴⁰ Il est courant de définir des pondérations dont la somme est égale à 1.

⁴¹ Nous avons choisi une approche par pondération des critères d'une part pour sa simplicité de mise en œuvre et d'autre part en raison de la taille des matrices solutions manipulées qui aurait été coûteuse en mémoire dans le cas de la recherche d'un front pareto.

- Deuxième étape : Optimiser le coût 3

Cette étape vise à minimiser le coût lié à l'architecture des acteurs du projet (FIG. 5.10-coût-3) par la méthode d'optimisation de couplage (Eq- 5-6). Nous avons utilisé le solveur de programmation quadratique convexe que propose la toolbox d'optimisation de Matlab (fonction Quadprog).

- Troisième étape : Calculer le coût global et utiliser la méthode de recuit simulé

Le coût global est obtenu en sommant les coûts précédemment calculés. Ce coût global est provisoire.

- Quatrième étape : Générer et tester des solutions voisines

La méthode de recuit simulé génère et teste des solutions d'affectations voisines (MI-affectation) et l'algorithme répète les trois étapes présentées précédemment jusqu'à l'obtention du coût global minimal.

L'algorithme de recuit simulé utilisé s'écrit comme suit.

1. Initialisation des paramètres

- 1.1 Le compteur de température, $m=0$
- 1.2 Le compteur de palier, $n = 0$;
- 1.3 Initialise le nombre de décroissances de température M
- 1.4 Initialise le nombre de paliers N
- 1.5 Générer la solution initiale, $MI_1 := MI_0$; (par programmation linéaire)
- 1.6 Générer la DSM acteur initiale, $DSM-A_1 := DSM-A_0$;
- 1.7 Calculer la fonction objectif $F(MI_1)$;
- 1.8 $MI_{opt} := MI_1$; $DSM-A_{opt} := DSM-A_1$; $F_{opt} := F(MI_1)$;

2. Tant que ($m < M$)

$m := m + 1$; $n := 0$;

Tant que ($n < N$)

- (i) Effectuer $MI_2 \in V(MI_1)$ (génération d'un voisin)
- (ii) Calculer $coût_1$ et $coût_2$ (coûts liés au pilotage des compétences)
- (iii) Optimiser la DSM- A_2 ; Calculer $coût_3$; (coût lié à la DSM acteur)
- (iv) Calculer le coût global $F(MI_2) := coût_1 + coût_2 + coût_3$;
- (v) Calculer la probabilité d'acceptation $P(MI_1, MI_2, T_m)$
- (vi) Accepter ou rejeter le candidat MI_2

Si ($F(MI_2) \leq F(MI_1)$) Alors

accepter la solution ($MI_1 := MI_2$);

accepter la fonction coût ($F(MI_1) := F(MI_2)$);

accepter la DSM acteur ($DSM-A_1 := DSM-A_2$);

Si ($F(MI_1) \leq F(MI_{opt})$) Alors

accepter la solution optimale ($MI_{opt} := MI_1$);

accepter la fonction coût optimale ($F_{opt} := F(MI_1)$);

accepter la DSM acteur ($DSM-A_{opt} := DSM-A_1$);

Finsi

$random := rand()$;

Si ($random < P(MI_1, MI_2, T_m)$) Alors

accepter la solution ($MI_1 := MI_2$);

accepter la fonction coût ($F(MI_1) := F(MI_2)$);

Finsi

Finsi

$n = n + 1$;

Fin Tant que

$m = m + 1$;

Fin Tant que

Afficher MI_{opt} ; $DSM-A_{opt}$; F_{opt} ;

FIG. 5.18: Algorithme du recuit simulé utilisé

Mise en œuvre

- *MI-affectation optimale*

En utilisant l'heuristique d'optimisation présentée précédemment, nous obtenons les MI d'affectations optimales se composant de 6 sous-matrices (une matrice pour chaque projet et chaque période). La version complète de ces matrices est présentée dans l'annexe 4.

<i>p=1</i>	Tâche	<i>k=1</i>	<i>k=2</i>	<i>K=3</i>
	T1 (spécification)	analyse produit, A1	-	-
	T2 (architecture)	électronicien-E, A5 (0,2), architecte-E, A19 (0,1) intégrateur-C, A20 (0,7)	-	-
	T3 (intégration)	-	-	intégrateur-C, A20
	T4 (management)	intégrateur-C, A20	intégrateur-C, A20	intégrateur-C, A20
	T5 (capteur)	-	électronicien-C, A6 (0,5) mécatronicien-C, A13 (0,5)	automaticien-D, A18
	T6 (vision)	-	automaticien-C, A16	automaticien-D, A18
	T7 (manipulation)	-	mécatronicien-E, A12 (0,8) intégrateur-C, A20 (0,2)	mécatronicien-C, A13 (0,2) intégrateur-C, A20 (0,8)
	T8 (locomotion)	-	mécatronicien-C, A14	-
	T9 (circuit)	-	électronicien-C, A6 (0,7) électronicien-D, A8(0,3)	électronicien-D, A8
	T10 (contrôle)	-	automaticien-C, A17	automaticien-C, A17
	T11 (énergies)	-	mécatronicien-E, A12 (0,2) mécatronicien-C, A13 (0,8)	-
	T12 (communication)	-	électronicien-D, A8	électronicien-D, A8
	T13 (navigation)	-	électronicien-D, A8 (0,2) automaticien-C, A16 (0,4) automaticien-C, A17 (0,4)	-
	T14 (capteur nano)	-	-	-
	T15 (capteur optique)	-	-	-
<i>p=2</i>	Tâche	<i>k=1</i>	<i>k=2</i>	<i>k=3</i>
	T1 (spécification)	-	-	-
	T2 (architecture)	-	-	-
	T3 (intégration)	-	intégrateur-C, A20	-
	T4 (management)	intégrateur-C, A20	intégrateur-C, A20	-
	T5 (capteur)	-	-	-
	T6 (vision)	automaticien-C, A16 (0,9) automaticien-C, A17 (0,1)	-	-
	T7 (manipulation)	-	-	-
	T8 (locomotion)	mécatronicien-C, A13 (0,2) mécatronicien-C, A14 (0,8)	-	-
	T9 (circuit)	électronicien-C, A6 (0,8) électronicien-C, A7 (0,2)	-	-
	T10 (contrôle)	automaticien-C, A17	-	-
	T11 (énergies)	mécatronicien-C, A13	-	-
	T12 (communication)	électronicien-D, A8	-	-
	T13 (navigation)	-	-	-
	T14 (capteur nano)	mécatronicien-E, A12	mécatronicien-D, A12 (0,4) automaticien-D, A18(0,6)	-
T15 (capteur optique)	électronicien-D, A8 (0,1) automaticien-D, A18 (0,9)	automaticien-D, A18	-	

TAB. 5.3 : Solution d'affectation finale (recuit simulé)

Ce tableau met en évidence les informations suivantes : les profils de compétence (électronicien, mécatronicien, etc.), les niveaux d'expertise (E-Expert, C-Confirmé et D-Débutant) et les pourcentages d'affectation pour chaque période.

Nous présentons ci-dessous les matrices d'affectation optimales (agrégées) pour le projet 1 (FIG. 5.19) et projet 2 (FIG. 5.20). Ces matrices agrégées sont obtenues lors de l'optimisation de l'architecture des acteurs.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T5						0,31							0,29					0,40		
T6																0,67		0,33		
T7												0,54	0,07							0,39
T8														1,00						
T9						0,44		0,56												
T10																	1,00			
T11												0,20	0,80							
T12								1,00												
T13								0,19								0,40	0,41			

FIG. 5.19 : MI-affectation optimale (agrégée) du projet 1 ($l=1$)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T6																0,93	0,07			
T8													0,16	0,84						
T9						0,79	0,20	0,01												
T10																	1,00			
T11													1,00							
T12								1,00												
T14													0,76						0,24	
T15								0,09											0,91	

FIG. 5.20 : MI-affectation optimale (agrégée) du projet 2 ($l=2$)

- *DSM acteur et architecture des acteurs optimales*

En partant des matrices DSM tâches présentées (FIG. 5.14) et des MI-Affectations agrégées obtenues (FIG. 5.19 et FIG. 5.20), nous générons les DSM acteurs pour chaque projet. Ces DSM acteurs sont présentées dans la figure (FIG. 5.21). Nous avons fixé les paramètres de normalisation à 0,4 ($\beta = 0,4$) et d'homogénéisation à 6.

	A6	A8	A12	A13	A14	A16	A17	A18	A20
A6		0,6	0,2	0,5				0,7	0,2
A8	0,6					0,4	0,4		
A12	0,2			0,6				0,3	0,6
A13	0,5		0,6					0,6	0,2
A14									
A16		0,4					0,8	0,6	
A17		0,4				0,8			
A18	0,7		0,3	0,6		0,6			0,2
A20	0,2		0,6	0,2				0,2	

(a) DSM acteur (projet 1)

	A6	A7	A8	A12	A13	A14	A16	A17	A18
A6		0,6						0,1	
A7	0,6								
A8									0,6
A12									0,6
A13						0,6			
A14				0,6					
A16								0,6	
A17	0,1						0,6		
A18			0,6	0,6					

(b) DSM acteur (projet 2)

FIG. 5.21 : DSM acteurs optimales

L'analyse des DSM acteurs nous permet de dresser les constats suivants :

- Une analyse visuelle des valeurs des DSM acteurs montre qu'elles sont cohérentes avec celles des DSM tâches (FIG. 5.14) et des MI-affectations (FIG. 5.19 et FIG. 5.20).
- La MI-affectation a plus d'impact sur les valeurs résultantes que la DSM tâche. Cela peut être lié à la formule utilisée ($DSM(A)=\mu \times MI(T-A)' \times DSM(T) \times MI(T-A)$) qui prend en compte deux fois les valeurs dans la MI-affectation. Ainsi, deux acteurs étant alloués pour travailler sur la même tâche auront une forte valeur de couplage dans les DSM acteurs résultantes. Deux acteurs travaillant sur des tâches qui sont interdépendantes (selon le lien de DSM(T)) seront couplés mais les valeurs dans la matrice résultante seront moins fortes.
- Comme nous avons enlevé les acteurs qui travaillent sur les tâches d'ingénierie système et de coordination du projet, ces acteurs n'ont pas été présentés dans cette architecture. Rappelons qu'en réalité, ces acteurs sont membres d'une équipe dont le rôle est d'assurer la coordination entre les équipes et l'intégration globale du système.

5.3.3.3 Phase VI : Construction de l'architecture des acteurs (algorithme de clustering)

Principe

Cette étape présente le regroupement des acteurs sélectionnés (DSM acteur optimisée) en équipes de projet (architecture des acteurs). La matrice DSM acteur sera regroupée selon l'algorithme de regroupement afin de faire apparaître l'architecture des acteurs.

Notre objectif dans cette phase n'étant pas de développer un algorithme de regroupement, nous avons utilisé un algorithme existant, ayant déjà prouvé son efficacité, pour identifier l'architecture des acteurs [Harmel, 2007]. Nous avons donc adopté un algorithme de référence proposée par [Idicula, 1995], amélioré par Fernandez [1998] et Thebeau [2001] pour faire ressortir des équipes dans le projet permettant de minimiser les efforts de coordination entre équipes.

Mise en œuvre

- Architecture des acteurs

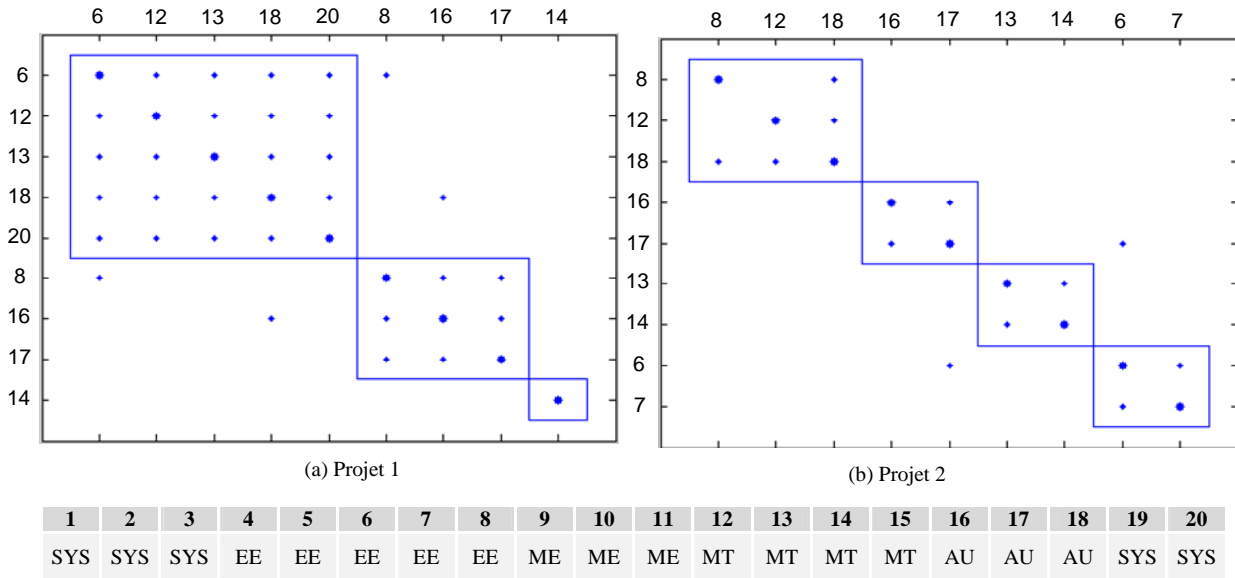


FIG. 5.22 : Architecture des acteurs simulée

La figure (FIG. 5.22) présente l'architecture des acteurs simulée pour chacun des projets. La signification des légendes du profil des acteurs est présentée dans le tableau (TAB. 4.10) dans le chapitre 4.

Nous analysons les résultats obtenus de la manière suivante :

- L'algorithme de clustering a identifié deux équipes et un acteur autonome dans le projet 1. Dans le projet 2, l'algorithme a défini quatre équipes.
- Concernant l'architecture des acteurs obtenue, nous retrouvons globalement les équipes de conception qui sont organisées selon les interactions de la matrice d'affectation.
- Le premier projet est constitué de trois équipes. L'équipe 1 comprend les acteurs A6, A12, A13, A18 et A20. L'acteur A6 travaille sur les tâches T5, T9, l'acteur A12 travaille sur les tâches T7, T11, l'acteur A13 travaille sur les tâches T5, T7 et T11, l'acteur A18 travaille sur les tâches T5, T6. Nous constatons que ces acteurs font partie de la même équipe parce qu'ils sont alloués pour travailler sur certaines tâches communes (T5, T7, T11). Concernant la deuxième équipe du projet, elle comprend les acteurs A8, A16, A17. L'acteur A8 travaille sur les tâches T9, T12, T13, l'acteur A16 réalise les tâches T6, T13 et l'acteur A17 est alloué aux tâches T10, T13. Ces acteurs sont amenés à travailler ensemble sur la réalisation de la tâche T13.
- Le deuxième projet est constitué de quatre équipes. Les équipes sont aussi formées en fonction des interactions dans la MI-affectation. Prenons l'exemple de la première équipe, elle est constituée des acteurs A8, A12, A18. L'acteur A8 travaille sur les tâches T9, T12, T15, l'acteur A12 réalise la tâche T14 et l'acteur A18 réalise les tâches T14 et T15. Ces acteurs sont donc amenés à travailler ensemble à la réalisation des tâches (T14 et T15).

- Certains acteurs ont des interactions avec des acteurs appartenant à des équipes différentes. Cependant, ces acteurs ne sont pas en interaction avec suffisamment de membres d'équipes différentes pour être considérés comme étant des acteurs intégrateurs.
- Globalement, les résultats obtenus sont donc satisfaisants.

5.3.4 Eléments de validation

Nous analysons maintenant les intérêts de la méthode proposée au regard des qualités attendues, spécifiées au (§5.3.1).

Minimiser le coût de réalisation des tâches

Ci-dessous se trouvent des courbes représentant les variations de différents coûts (coût total, coût salaire, coût tuteur et coût lié à l'objectif, coût de conservation de l'architecture des tâches et coût DSM acteur). Cette figure montre les valeurs et les ordres de grandeur des coûts utilisés dans l'optimisation. Les valeurs des poids permettant d'équilibrer les coûts ont été définies comme suit, $\omega_1 = 1$, $\omega_2 = 1$ et $\omega_3 = 100$.

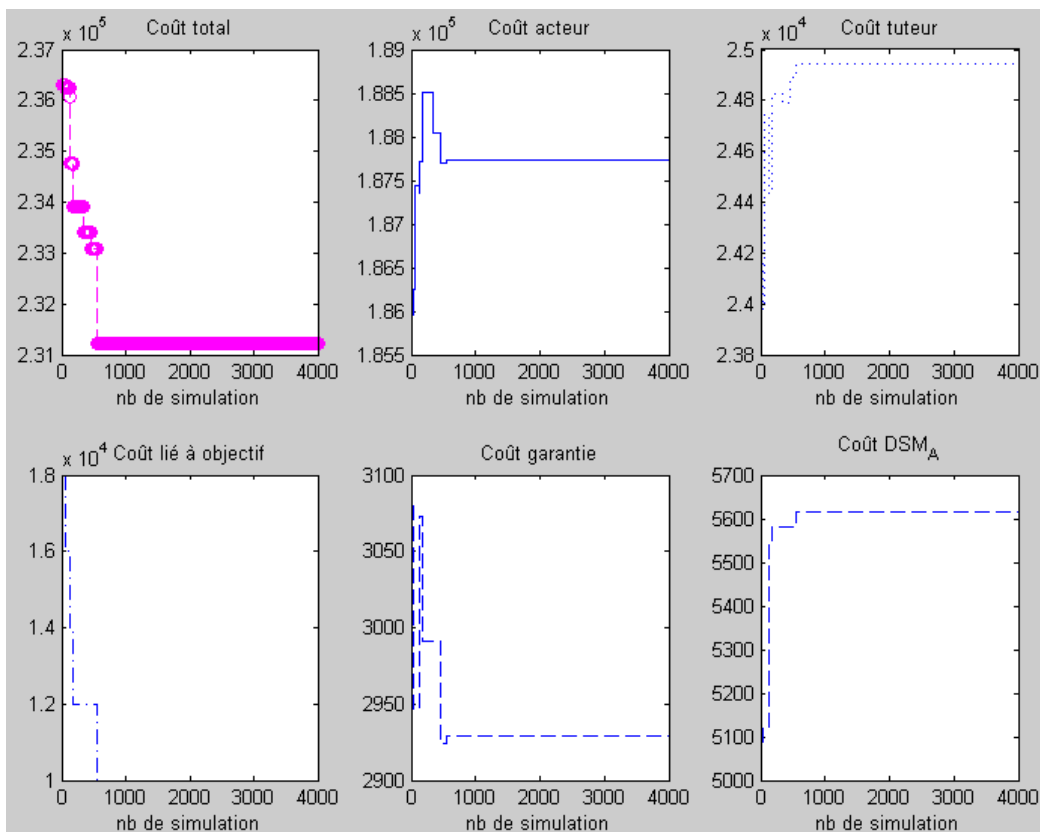


FIG. 5.23 : Axe financier – variations des coûts

- Après avoir utilisé l'algorithme proposé, nous avons constaté que les coûts ont été équilibrés globalement. Nous avons obtenu un gain économique de 6% sur le coût global par rapport à la solution initiale.

- Un grand nombre de simulations ont été réalisées, de façon à obtenir une stabilisation du coût global. Les courbes de coûts sont stables à partir de 25% du nombre de simulations total. Néanmoins, nous sommes conscients que le problème contient un grand nombre de solutions potentielles (de l'ordre de 10^{15}). Même si nous avons utilisé un nombre de simulations assez élevé, nous ne pouvons pas garantir que la solution obtenue n'est pas en réalité un optimum local. Nous avons lancé plusieurs fois la résolution de ce problème et nous avons retrouvé les mêmes résultats.
- Nous avons constaté que la programmation linéaire en nombre réel propose une solution initiale admissible de manière rapide (en 2-3 secondes). L'algorithme basé sur le recuit simulé améliore les résultats obtenus mais utilise un temps de simulation important (en 9 heures pour 4000 simulations).

Affecter des tâches multi-périodes et multi-projets en tenant compte d'une loi d'évolution des compétences et minimiser le coût lié à l'objectif de compétences

Le tableau ci-dessous présente le résultat sur la satisfaction en termes d'objectif de compétences et de nombre d'acteurs compétents qui ont été formés.

Tâche	Solution finale							
	Objectif		R3 initiale (k=0)			R finale (k=K)		
	Nb.	Seuil	Nb.	S/NS ⁴²	Acteur	Nb.	S/NS	Acteur
T1	1	0,8	1	S	A1	1	S	A1
T2	2	0,7	4	S	A2, A12, A19, A20	2	S	A19, A20
T3	2	0,7	3	S	A12, A19, A20	2	S	A12, A20
T4	2	0,8	2	S	A19, A20	2	S	A19, A20
T5	3	0,7	4	S	A4, A5, A6, A13	6	S	A5, A6, A12, A13, A18, A20
T6	4	0,7	2	NS	A16, A17	3	NS	A16, A17, A18
T7	4	0,7	6	S	A2, A9, A12, A13, A19, A20	5	S	A9, A12, A13, A14, A20
T8	3	0,8	1	NS	A12	2	NS	A12, A14
T9	3	0,8	2	NS	A5, A6	3	S	A5, A6, A13
T10	1	0,8	1	S	A12	2	S	A17, A20
T11	2	0,7	1	NS	A12	2	S	A12, A13
T12	3	0,8	2	NS	A16, A17	3	S	A8, A16, A17
T13	3	0,7	2	NS	A16, A17	3	S	A8, A16, A17
T14	4	0,8	2	NS	A5, A12	4	S	A5, A12, A13, A20
T15	4	0,8	0	NS	-	1	NS	A18
Manque	-	-	14p	-	-	5p	-	-

TAB. 5.4: Evolution du nombre des acteurs compétents

Ce tableau nous permet de tirer les conclusions suivantes :

- Nous remarquons que les objectifs de compétences à la fin de simulation ont été globalement mieux satisfaits par rapport à la période initiale. Le nombre d'acteurs qualifiés manquants est diminué de 14 à 5 personnes. Cela signifie que l'algorithme travaille correctement puisqu'il permet d'allouer des acteurs aux tâches selon la loi d'évolution des compétences.

⁴² S=Satisfait, NS=Non satisfait

- La solution initiale est la même que celle présentée dans le cas multi-période, mais la solution finale est différente car cet algorithme cherche en plus à diminuer le coût d'interaction entre des acteurs (DSM acteur). Nous comparons et analysons dans la partie suivante les différentes solutions proposées dans ce mémoire.
- Les objectifs de compétence non satisfaits par l'affectation devront faire l'objet de plans d'actions particuliers en dehors des projets (formation, recrutement, ...). Le coût de ces actions correspond aux pénalités financières ajoutées au coût des projets.

5.3.5 Comparaison des solutions proposées

Nous présentons dans le tableau (TAB. 5.5), un comparatif entre 3 méthodes de résolution du problème, proposées dans les chapitres 4 et 5. Tout d'abord, rappelons les critères pris en compte et les techniques de résolution proposées.

- La première résolution prend en compte uniquement le critère visant à minimiser le coût salarial lié aux niveaux de compétence. Les critères liés à la non-satisfaction des objectifs de compétence et à l'interaction des acteurs n'ont pas été pris en compte. Néanmoins, la loi d'évolution des compétences a été appliquée. Le problème est résolu par la programmation linéaire en nombres réels. Cette résolution est présentée dans la partie (§4.3.6.2).
- La deuxième résolution prend en compte le critère lié aux niveaux de compétence et le critère lié la non-satisfaction des objectifs de compétence. Le critère d'interaction des acteurs n'a pas été pris en compte. La loi d'évolution des compétences a été appliquée. Le problème est résolu par la méthode du recuit simulé. Elle est présentée dans la partie (§4.3.7.1).
- La troisième résolution prend en compte tous les critères : critère lié aux niveaux de compétence, critère lié à la non-satisfaction des objectifs de compétence, critère de conservation de l'architecture des tâches et critère de minimisation des interactions entre les acteurs. La loi d'évolution des compétences a été appliquée. Elle est résolue par un algorithme comprenant l'optimisation quadratique et la méthode de recuit simulé. Elle est présentée dans la partie (§5.3.3.2).

Le tableau (TAB. 5.5) présente les résultats principaux obtenus par ces trois résolutions⁴³.

⁴³ Les coûts présentés dans ce tableau sont déjà multipliés par les poids (ω_i).

	Résolution (1)	Résolution (2)	Résolution (3)
Coût salarial lié aux niveaux de compétence ($\omega_1 = 1$)	210 136€	210 947€	212 684€
Coût de pénalité financière liée à la non-satisfaction des objectifs de compétence ⁴⁴ (nombre d'acteurs formés) ($\omega_2 = 1$)	18 000€ (9 acteurs)	10 000€ (5 acteurs)	10 000€ (5 acteurs)
Coût de conservation de l'architecture des tâches ($\omega_3 = 100$)	3 000€	3 100€	2 900€
Coût d'interaction entre les acteurs ($\omega_3 = 100$)	5 100€	10 600€	5 600€
Nombre d'interactions dans la MI-affectation	28	37	31
Nombre d'acteurs alloués ⁴⁵	17 acteurs	20 acteurs	18 acteurs

TAB. 5.5 : Analyse des résultats

Le tableau de comparaison ci-dessus suggère les remarques suivantes :

- La résolution (1) propose comme solution une équipe minimisant le coût salarial lié aux niveaux de compétence. Ce coût salarial est moins élevé que celui pouvant être obtenu par les autres solutions. L'algorithme donne la priorité aux acteurs compétents dans l'affectation car les acteurs ont besoin de peu d'accompagnement. Le critère de la non-satisfaction des objectifs de compétence n'étant pas pris en compte, nous pouvons constater logiquement que le coût de pénalité financière est élevé (le nombre d'acteurs formés est peu élevé).
- La résolution (2) propose une solution d'équipe de conception répondant à la fois au critère lié aux niveaux de compétence et au critère lié à la non-satisfaction des objectifs de compétence. L'algorithme donne alors la priorité à des acteurs moins compétents. Cela engendre directement une augmentation dans le coût salarial, mais une diminution des pénalités financières liées à la non-satisfaction des objectifs de compétence (nous pouvons constater que le nombre d'acteurs compétents a augmenté).
- Dans la résolution (2), nous constatons que le fait de sélectionner et de former des acteurs sous-compétents a entraîné une augmentation du nombre d'acteur et du nombre d'interactions entre les acteurs. Cela veut dire que l'algorithme cherche à affecter les tâches au maximum à des acteurs sous-compétents pour atteindre les objectifs de compétence fixés. Cela a augmenté le nombre d'acteurs dans les projets et

⁴⁴ Le taux pénalisant utilisé est 2000 euros par un acteur compétent manquant

⁴⁵ Ce sont des acteurs qui travaillent uniquement aux tâches de conception (les tâches d'ingénierie système ne sont pas considérés), le même acteur peut être compté plusieurs fois s'ils contribuent à plusieurs projets.

le nombre d'interactions entre les acteurs. Nous pouvons donc constater un coût d'interaction élevé.

- La résolution (3) propose une équipe de conception répondant aux quatre critères (coût salarial, objectif de compétences, conservation de l'architecture et minimisation des interactions entre les acteurs). Nous remarquons que l'algorithme cherche à former les acteurs sous-compétents comme dans la résolution (2). Cette résolution apporte en plus par rapport à la solution (2), une diminution du coût d'interaction entre les acteurs, de 10 600 à 5 600 euros. Cette diminution peut être expliquée par la diminution du nombre d'interactions dans la MI-affectation et du nombre d'acteurs participant au projet. Sans ce critère, il y a un "éparpillement" des contributions dans l'affectation.

La résolution (3) nous apparaît comme une solution pour obtenir une équipe de conception prometteuse permettant de prendre en compte tous les critères définis.

5.4 Conclusion et synthèse

Nous avons dans une première partie, présenté une méthode d'optimisation du couplage entre les architectures de deux domaines. Nous l'avons illustrée sur un projet de conception dans le but d'obtenir l'architecture des acteurs. Dans la seconde partie, nous avons présenté la démarche de la constitution d'équipes comprenant l'optimisation de l'architecture des acteurs et de l'affectation multi-périodes et multi-projets en tenant compte de l'évolution des compétences. Les enjeux de cette méthode sont de piloter les compétences des acteurs et de proposer des équipes de conception en tenant compte des contraintes liées au produit.

Nous avons appliqué cette démarche dans le cas d'un projet de conception d'une capsule endoscopique. Il est certain que cette méthode devrait être validée par un cas industriel réel et que d'autres applications devraient être réalisées afin d'en confirmer l'intérêt.

En synthèse, nous présentons dans le tableau (TAB. 5.6) une estimation qualitative de la satisfaction des exigences spécifiées au (§5.3.1).

Exigences	Estimation qualitative	Travaux supplémentaires
E1 : Affecter des tâches multi-périodes et multi-projets en tenant compte d'une loi d'évolution des compétences	++	<ul style="list-style-type: none"> - Test sur un cas réel - Analyse de sensibilité - Autres algorithmes d'optimisation
E2 : Minimiser le coût salarial lié aux niveaux de compétence	++	
E3 : Minimiser les pénalités financières dues à l'écart sur l'objectif de compétences	+ +	
E4 : Garantir que les interfaces entre les composants du produit seront bien étudiées	++	
E5 : Minimiser les besoins d'interactions entre les acteurs dans le projet de conception	+ +	

TAB. 5.6: Estimation qualitative de la démarche de constitution d'équipes

6 Conclusion et perspectives

Nous avons insisté tout au long de la première partie de ce mémoire sur la place et le rôle prépondérants de l'homme en tant que ressource principale dans les projets de conception. La réalisation d'outils d'aide aux décisions pour les problèmes de constitution d'équipes n'est pas un processus achevé. Les chefs de projets et les directeurs de R&D ont besoin d'outils qui pourraient compléter les logiciels existants en gestion de projet, afin de mieux prendre en compte des objectifs de développement des compétences.

Ce travail s'inscrit au niveau tactique lors du pilotage des compétences des ressources. Son originalité majeure réside dans une démarche de la constitution d'équipes contribuant à la fois à la structuration du projet et au pilotage des compétences (problème d'affectation). Cette thèse est organisée en deux parties, répartie sur 5 chapitres. Dans les trois premiers chapitres, nous avons approfondi les études sur la structuration du projet et le pilotage des compétences permettant de préciser notre problématique. Les deux derniers chapitres présentent les contributions scientifiques de notre travail.

Dans un premier temps, nous porterons notre attention sur notre contribution à la problématique traitée. Ensuite, nous tirerons un bilan de nos travaux. Nous dégagerons finalement des perspectives de recherche.

Etats de l'art et contributions

Dans le premier chapitre, nous avons exposé une présentation globale de deux thèmes de recherche relatifs à la constitution d'équipes : la structuration du projet et le pilotage des compétences. Ces deux thèmes sont les axes de recherche que nous avons approfondis tout au long de mémoire. D'une part, dans le cadre de la structuration du projet, la conception des architectures des acteurs est considérée comme une manière de constituer des équipes. Ces travaux ont pour but de concevoir des équipes cohérentes qui améliorent la communication entre concepteurs. D'autre part, dans le cadre du pilotage des compétences, l'affectation des tâches à des ressources prenant en compte la mobilisation des compétences des acteurs est considérée comme une autre manière de constituer des équipes. En conclusion de ce chapitre, un premier positionnement de notre recherche a été présenté comme une articulation entre ces deux thèmes.

Le deuxième chapitre a présenté des états de l'art relatifs à ces deux thèmes. Concernant la structuration du projet, nous avons montré l'importance de la prise en compte des contraintes liées à l'architecture du produit (ou du processus) sur la conception de l'architecture des acteurs. Le nombre de travaux de recherche proposant une approche formelle de couplage est encore limitée. Concernant le pilotage des compétences, nous avons étudié les travaux de la constitution d'équipes dans le domaine de la conception. Il est important de rappeler que pour développer leurs compétences, il est nécessaire que les acteurs mobilisent leurs connaissances

ou leurs compétences dans la réalisation des tâches correspondantes. Une bonne mobilisation des compétences des acteurs permet d'avoir un pilotage efficace des compétences. Ainsi, le problème d'allocation des ressources aux tâches (ou affectation des tâches à des ressources) est considéré comme un sous-problème important dans la constitution d'équipes. Nous retenons que les travaux d'affectation multi-périodes avec évolution de compétence présentent un intérêt majeur mais sont peu traités dans la littérature. Enfin, nous avons constaté qu'il n'y a aucun travail qui traite le problème d'affectation avec évolution de compétence et la conception de l'architecture des acteurs en même temps.

Dans le troisième chapitre, nous avons approfondi des travaux concernant l'affectation des tâches à des ressources prenant en compte le caractère évolutif des compétences. Le but de cette analyse était de mettre en évidence les limites et les intérêts des modèles existants. Malgré l'importance de l'évolution des compétences dans le pilotage des compétences, nous avons constaté que peu de travaux se sont intéressés à celle-ci. Cela est dû au fait que la définition d'une loi d'évolution est complexe.

Dans le quatrième chapitre, nous avons présenté une contribution à l'affectation multi-périodes et multi-projets des tâches aux ressources avec prise en compte de l'évolution des compétences. Pour éviter la situation où les écarts négatifs compensent les écarts positifs, nous avons proposé un indicateur de similarité entre tâches et acteurs prenant en compte ces différences. Un challenge scientifique a été de modéliser une loi d'évolution des compétences entre deux périodes k et $k+1$. Cette loi d'évolution des compétences nous permet d'anticiper l'évolution du niveau de compétences des acteurs. Nous avons supposé que cette loi dépend des choix d'affectation réalisés à la période k . En raison du caractère non linéaire de la loi d'évolution des compétences, le problème est non linéaire. Nous avons donc choisi de le résoudre par une technique de résolution de type méta-heuristique, le recuit simulé.

Le dernier chapitre de ce document est consacré à la constitution d'équipes contribuant à la fois à l'affectation multi-périodes et à la conception de l'architecture des acteurs du projet. Dans la première partie de ce chapitre, nous avons proposé une méthode d'optimisation de couplage entre les architectures de deux domaines permettant d'anticiper l'impact des choix concernant l'architecture du produit (ou du processus) sur les efforts de coordination qui seront nécessaires entre les équipes. Dans la seconde partie de ce chapitre, nous avons combiné les méthodes d'optimisation de l'architecture des acteurs et de l'affectation multi-périodes pour proposer une démarche globale de constitution d'équipes.

La démarche de la constitution d'équipes que nous avons présentée dans ce mémoire est liée au contexte industriel. Notre premier exemple d'application (jeu de données tiré d'un projet de conception de micro-produit) nous a permis de confirmer la validité du "principe de solution" pour résoudre ce type de problématique. Néanmoins, nous sommes conscients que les apports de ce travail sont à consolider et que des améliorations doivent être envisagées pour le rendre applicable. D'autres applications sont à prévoir pour affiner les modèles proposés et confirmer l'intérêt des méthodes proposées.

Améliorations et perspectives

Notre approche présente certaines limites et plusieurs perspectives de recherche sont envisageables.

Améliorations de la méthode d'évolution des compétences proposée

Une amélioration de la méthode proposée pourrait être liée au calcul du niveau de compétence selon l'approche "ascendante". Cette approche modélise le fait que les acteurs

peuvent augmenter leurs niveaux de compétence dans la réalisation de tâches qu'ils n'ont pas réalisés. Cela est dû au changement du niveau de maîtrise dans une "connaissance" qui a été répercuté sur toutes les tâches associées à cette connaissance. La répercussion systématique pourrait engendrer des augmentations inappropriées : les acteurs compétents "théoriquement" seraient aussi compétents que les acteurs compétents "en pratique". Pour pallier cet éventuel problème, nous pensons qu'il est intéressant de caractériser et de modéliser les plans d'actions comme un type de connaissance avec éventuellement un poids d'importance. Ces plans d'actions représentent la capacité de mettre en œuvre et de combiner l'ensemble des connaissances nécessaires à la réalisation d'une mission (en pratique). Néanmoins, l'application réalisée a montré que cet inconvénient peut être mineur.

La loi d'évolution de compétence proposée calcule l'augmentation du niveau de maîtrise dans la connaissance en fonction de la charge affectée. D'autres types de lois d'évolution de compétences peuvent être envisagés, par exemple une loi dépendant de la fréquence d'affectation (nombre d'affectations). Les courbes d'apprentissage peuvent être également personnalisées selon le type de compétences (chaque compétence présente un niveau de difficulté différent) et le niveau de compétences des concepteurs (chaque acteur possède une capacité d'apprentissage différente).

La constitution d'équipes est une activité importante dans le pilotage d'un projet de conception. Elle met l'accent non seulement sur la prise en compte de la performance technique mais également sur la prise en compte des compétences psycho-sociologiques facilitant la collaboration entre membres d'une même équipe. Il pourrait être intéressant de compléter la démarche par l'ajout d'autres contraintes relatives aux compétences interpersonnelles ou à la capacité à collaborer afin d'assurer une meilleure performance des équipes du projet.

Vers une amélioration de l'évolution des compétences

Une perspective paraissant intéressante concerne le développement d'une démarche aidant les directeurs de R&D à fixer des objectifs de développement de compétences (types et niveaux de compétences). Elle pourrait s'appuyer sur des actions de type benchmarks ou veille technologique (par rapport à leurs concurrents, fournisseurs, clients, etc.) pour identifier les compétences qui peuvent potentiellement améliorer les performances de l'entreprise.

Vers une amélioration de la méthode de constitution d'équipes

Pour créer des équipes de conception, nous avons proposé une démarche qui procède d'abord à l'affectation et ensuite qui regroupe les acteurs en équipes. A l'inverse, on pourrait envisager une démarche qui commence par le regroupement des tâches avant de les affecter aux acteurs. Ce regroupement des tâches permettrait d'assurer la cohérence des modules des tâches en termes de contraintes de précedence (approche classique avec les DSM tâches) et de proximité entre les compétences requises. Ce dernier critère devrait permettre d'affecter plusieurs tâches aux mêmes acteurs afin de réduire la taille des équipes.

Nous avons rencontré dans l'exemple certaines tâches nécessitant plusieurs types de connaissances (mécanique, automatique, électronique, etc.) en même temps. Il est difficile de trouver un seul acteur compétent pour ce type de tâche. Souvent les acteurs qui ne sont compétents que dans certaines connaissances ne seront pas sélectionnés parce qu'ils ne sont pas assez compétents globalement pour la tâche. Il est donc intéressant d'envisager une démarche de constitution d'équipes permettant de choisir des membres dans l'équipe de manière à compléter des connaissances manquantes.

Par ailleurs, nous sommes conscients que les interactions entre individus n'ont pas seulement un coût mais peuvent aussi être créatrices d'innovation. Une extension de nos

travaux concerne une amélioration des modèles proposés pour intégrer une dimension collaborative favorable à l'émergence de compétences collectives.

Une autre perspective concerne la prise en compte de la planification des tâches qui nécessite de modéliser des contraintes de précedence entre tâches. Nous pensons que différentes configurations de l'ordre des tâches peuvent éventuellement proposer différents plans d'apprentissage des acteurs, tout en permettant de mieux répondre à l'objectif de compétences fixé.

Ce travail est évalué uniquement en termes de coût du projet. Les risques du projet n'ont pas été pris en compte. Il est intéressant de lier la notion de risques avec la notion de compétences. Dans la gestion des risques, on cherche à définir des critères permettant d'évaluer la réussite potentielle du projet résultant d'une exposition aux risques. On peut considérer des critères basés sur les niveaux de compétence pour estimer les risques (probabilité de réussite) du projet. Ces critères permettraient de comparer en particulier l'impact de différentes solutions d'affectation sur la réussite potentielle. Les contraintes de précedence permettant de définir le chemin critique du projet pourraient éventuellement être intégrées dans la démarche pour évaluer la réussite potentielle.

Jusqu'à présent, nous avons fait l'hypothèse d'évolution déterministe des compétences (selon une loi connue à l'avance) et d'une réussite systématique des tâches du projet, quels que soient les niveaux de compétence. Avec cette hypothèse, la recherche d'optimisation par méta-heuristique est possible. Nous pourrions imaginer de générer aléatoirement un taux d'apprentissage et un taux de réussite des tâches. En cas d'échec, une tâche devrait être re-planifiée à la période suivante. Avec ce type de problème plus complexe, il pourrait être envisagé d'utiliser des outils de simulation pour évaluer, de façon statistique, les performances globales des projets.

Les taux de salaire doivent être déterminés de manière évolutif afin d'assurer la pertinence des résultats d'affectation. Dans le modèle d'affectation proposé, nous avons supposé que le taux de salaire des acteurs est statique durant la simulation, ce qui est raisonnable dans le cas où les projets considérés ne durent pas très longtemps. Dans le cas des projets de longue durée, nous pensons qu'il est utile de prendre cette évolution en compte.

Par ailleurs, nous avons résolu les problèmes formulés par une méta-heuristique : recuit simulé. Nous avons justifié le choix et l'utilisation du recuit simulé. Cependant, il faudrait comparer les performances de cet algorithme avec celles d'autres techniques de résolution de type méta-heuristiques.

Vers le développement d'un outil informatique

Une extension de ce travail est le développement d'un outil informatique supportant les différentes phases de la méthode (caractérisation des tâches et des acteurs, calcul de similarité, affectation, etc.) pour aider les chefs de projet à la conception des organisations dans le projet de conception des systèmes micro-mécatroniques.

L'interfaçage de ce modèle à des outils de gestion de projet pourrait être envisagé. Ces outils peuvent faciliter l'utilisation du modèle proposé en fournissant des informations concernant la planification des tâches, la gestion des compétences des acteurs, les travaux réalisés par des acteurs (le suivi), etc. L'alimentation de ces informations permettrait aux directeurs en R&D de disposer des nouvelles propositions d'équipes répondant à d'éventuelles évolutions de compétence (à court et long terme).

7 Bibliographie

- [Acuna et Juristo, 2004] Acuna S.T., Juristo N., "Assigning people to roles in software projects", *Software Practice and Experience*, 2004, No.34, pp. 675-696, 2004.
- [Acuna *et al.*, 2006] Acuna S.T., Juristo N., Moreno A.M., "Emphasizing Human Capabilities in Software Development", *IEEE Software*, Vol. 23, No. 2, pp. 94-101, Mar./Apr. 2006.
- [Acuna *et al.*, 2009] Acuña S.T., Gómez M., Juristo N., "How do personality, team processes and task characteristics relate to job satisfaction and software quality?", *Information and Software Technology*, Vol. 51, pp. 627–639, 2009.
- [Alliger *et al.*, 1993] Alliger G.M., Feinzig S.L., Janak E.A., "Fuzzy sets and personnel selection: Discussion and an application", *Journal of Occupational and Organizational Psychology*, Vol. 66, pp. 163-169, 1993.
- [Altus *et al.*, 1996] Altus S., Kroo I., Gage, P., "A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems", *ASME Journal of Mechanical Design*, Vol. 118, No.4, pp. 486-489, 1996.
- [Baker et Salas, 1992] Baker D. P., Salas E., "Principles for measuring teamwork skills", *Human Factors*, Vol. 34, No.4, pp. 369-475, 1992.
- [Baptiste *et al.*, 2005] Baptiste P., Giard V., Haït A., Soumis F., "Gestion de production et ressources humaines", Montréal, Presses internationales, 2005.
- [Baron, 2005] Baron C., "L'évaluation dans la conception système – Vers une aide à la conduite de projet", HDR, Institut National Polytechnique de Toulouse, 2005
- [Barreto *et al.*, 2008] Barreto A., Barros M.O., Werner C.M.L., "Staffing a software project: A constraint satisfaction and optimization-based approach", *Computers & Operations Research*, Vol. 35 , No. 10, pp. 3073-3089, October 2008.
- [Barrick *et al.*, 1998] Barrick M.R., Stewart G.L., Neubert M., Mount M.K., "Relating member ability and personality to work team processes and team effectiveness", *Journal of Applied Psychology*, Vol. 83, pp.377-391, 1998.
- [Belbin, 1981] Belbin R.M., "Management Teams: Why they Success or Fail", Heinemann, London, 1981.
- [Belkadi, 2006] Belkadi F., "Contribution au pilotage des compétences dans les activités de conception : De la modélisation des situations à la caractérisation des compétences", Thèse de doctorat, Université de franche-comté, 2006
- [Belkadi *et al.*, 2007] Belkadi F., Bonjour E., Dulmet M., "Competency characterisation by means of work situation modelling", *Computers in Industry*, Vol. 58, No. 2, pp. 164-178, 2007.
- [Bellenguez-Morineau, 2006] Bellenguez-Morineau O., "Méthode de résolution pour un problème de gestion de projet multi-compétence", Thèse de doctorat, Université de François Rabelais, 2006.
- [Berio et Harzallah, 2007] Berio G., Harzallah M., "Towards an integrating architecture for competence management", *Computers in Industry*, Vol. 58, pp. 199–209, 2007.
- [Bennour, 2004] Bennour M., "Contribution à la modélisation et à l'affectation des ressources humaines dans les processus", Thèse de doctorat, Université Montpellier II, 2004
- [Bennour *et al.*, 2008] Bennour M., Crestani D., Crespo O., "Une méthode d'affectation des ressources humaines aux processus industriels", *Journal Européen des Systèmes Automatisés*, Vol 42, No. 5, pp. 541-577, 2008.
- [Boh *et al.*, 2007] Boh W. F., Slaughter S., Espinosa J.A., "Learning from Experience in Software

- Development: A Multi-Level Analysis", *Management Science*, Vol.53, No.8, pp. 1315-1331, 2007.
- [Bonjour *et al.*, 2007] Bonjour E., Micaëlli JP, Hlaottinun O, Deniaud I, (2007) "*Design And Core Competency, The Missing Links*", 4th International Federation of Automatic Control Conference, on Management and Control of Production and Logistics, Sibiu, Romania, 27-30 September 2007.
- [Bonjour, 2008a] Bonjour E., "Contributions à l'instrumentation du métier d'architecte système : de l'architecture modulaire du produit à l'organisation du système de conception", HDR, Université de Franche-Comté, 2008.
- [Bonjour, 2008b] Bonjour E., "Du retour d'expérience au développement des compétences", in "Retour et capitalisation d'expériences", (Editeurs : J. Renaud, E. Bonjour, B. Chebel-Morello, B. Fuchs, N. Matta), chapitre 5, pp 155-198, AFNOR, 2008.
- [Bonjour *et al.*, 2009a] Bonjour E., Deniaud S., Dulmet M., Harmel G., "A fuzzy method for propagating functional architecture constraints to physical architecture", *Transactions of ASME, Journal of Mechanical Design*, Vol. 131, No. 6, 12p, June, 2009.
- [Bonjour *et al.*, 2009b] Bonjour E., Dulmet M., Deniaud S., Micaëlli JP., "Propagating product architecture decisions onto the project organization: a comparison between two methods", *International Journal of Design Engineering*, à paraître en 2009, 21p
- [Bonjour *et al.*, 2009c] Bonjour E., Harmel G., Micaëlli J-P., Dulmet M., "Simulating change propagation between product architecture and development organization", *International Journal of Product Development*, accepté pour publication en 2009, 20p
- [Bonjour et Dulmet, 2006] Bonjour E., Dulmet M., "Pilotage des activités de conception par l'Ingénierie Système", *Ingénierie de la conception et cycle de vie des produits* (sous la direction de Roucoules L., éd.), Hermès, pp. 85-105, janvier, 2006.
- [Boucher *et al.*, 2006] Boucher X., Bonjour E., Grabot B., "Formalisation and use of competencies for industrial performance optimisation: A survey", *Computers in Industry*, Vol. 58, No. 2, pp. 98-117, 2006.
- [Boucher et Burlat, 2003] Boucher X., Burlat P., "Vers l'intégration des compétences dans le pilotage des performances de l'entreprise", *Journal européen des systèmes automatisés (JESA)*, Vol. 37, No. 3, pp. 363-390, 2003.
- [Bourgeon, 2007] Bourgeon L., "Staffing approach and conditions for collective learning in project teams: The case of new product development projects", *International Journal of Project Management*, Vol. 25, pp. 413-422, 2007
- [Braha, 2002] Braha D., "Partitioning tasks to product development teams", in *Proceedings of DETC'02 ASME 2002 International Design Engineering Technical Conferences*, Montreal, Canada, 2002.
- [Browning, 1998] Browning T.R., "Use of dependency structure matrices for product development cycle time reduction", *Proceedings of the 5.fth ISPE international conference on concurrent engineering: research and applications*, Tokyo, Japan; 1998.
- [Browning, 1999] Browning, T.R., "Designing system development projects for organizational integration", *Journal of Systems Engineering*, Vol. 2, No. 4, pp. 217-225, 1999
- [Browning, 2001] Browning T.R., "Applying the Design Structure Matrix to System Decomposition and Integration Problems", *IEEE Transactions on Engineering Management*, Vol.48, pp. 292-306, 2001.
- [Brusco et Johns, 1998] Brusco M.J., Johns T.R., "Staffing a multiskilled workforce with varying levels of productivity: An analysis of cross-training policies", *Journal of Decision Sciences*, Vol. 29, No. 2, pp. 499-515, June 2007.
- [Campbell et Diaby, 2002] Campbell G.M., Diaby M., "Development and evaluation of an assignment heuristic for allocating cross-trained workers", *European Journal of Operational Research*, Vol. 138, pp. 9-20, 2002.
- [Campbell, 1999] Campbell G.M., "Cross-Utilization of Workers Whose Capabilities Differ",

- Management Science, Vol. 45, No. 5, pp. 722-732, May, 1999.
- [Candes et Tao, 2005] Candes E.J., Tao T., "Decoding by linear programming", Information Theory, IEEE Transactions, Vol. 51, No.12, December 2005.
- [Canos et Liern, 2004] Canos L., Liern V., "Some fuzzy models for human resource management", International Journal of Technology Policy and Management, Vol. 4, No. 4, pp. 291-308, 2004.
- [Caron *et al.*, 1999] Caron G., Hansen P., Jaumard B., "The assignment problem with seniority and job priority constraints", Operations Research, Vol. 47, No. 3, pp. 449-454, March 1999.
- [Chan, 2002] Chan Y.C., "La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes", Thèse de doctorat, Université de Grenoble, 2002.
- [Chen et Askin, 2009] Chen J., Askin R.G., "Project selection, scheduling and resource allocation with time dependent returns", European Journal of Operational Research, Vol. 193, pp. 23-34, 2009.
- [Chen, 2005] Chen S.J., "An Integrated Methodological Framework for Project Task Coordination and Team Organization in Concurrent Engineering", Concurrent Engineering and Research and Applications, Vol.13, No. 3, pp. 185-197, 2005.
- [Chen et Lin, 2003] Chen S.J., Lin L., "Decomposition of interdependent task group for concurrent engineering", Computer & Industrial Engineering , Vol. 44, pp. 435-459, 2003.
- [Chen et Lin, 2004] Chen S.L., Lin L., "Modeling team member characteristics for the formation of a multifunctional team", IEEE Transactions on Engineering Management, Vol.51, No. 2, pp.111-124, 2004.
- [Cheng *et al.*, 2009] Cheng M.Y., Tsai H.C., Lai Y.Y., "Construction management process reengineering performance measurements", International Journal of Automation in Construction, Vol. 18, No. 2, pp. 183-193, March 2009.
- [Cho et Eppinger, 2001] Cho S., Eppinger S.D., "Product Development Process Modeling Using Advanced Simulation", Proceedings of DETC'01 ASME 2001 Design Engineering Technical Conferences Pittsburgh, Pennsylvania September 9-12, 2001
- [Cleland, 1994] Cleland D.I., "Project Management: Strategic Design and Implementation", 2nd ed., McGraw-Hill, New York, NY, pp. 391, 1994.
- [Coates *et al.*, 2004] Coates G., Duffy A.H.B., Whitfield I., Hillis W., "Engineering management: operational design coordination", Journal of Engineering Design, Vol. 15, No. 5, pp. 433-446, October 2004.
- [Coates *et al.*, 2007] Coates G., Duffy A.H.B., Hillis W., Whitfield I., "A preliminary approach for modelling and planning the composition of engineering project teams", Proceedings of the I MECH E Part B, Journal of Engineering Manufacture, Vol 221, No. 7, pp. 1255-1265, 2007.
- [Corominas *et al.*, 2006] Corominas A., Pastor R., Rodriguez E., "Rotational allocation of tasks to multifunctional workers in a service industry", International Journal of Production Economics, Vol. 103, pp.3-9, 2006.
- [Danilovic et Browning, 2007] Danilovic M., Browning T.R., "Managing complex product development projects with design structure matrices and domain mapping matrices", International Journal of Project Management, Vol. 25, pp. 300-314, 2007.
- [David *et al.*, 2002] David M., Idelmerfaa Z., Richard J., "Organization method for complex cooperative design projects", IEEE International Conference on Systems, Man and Cybernetics, Hammamet, 2002.
- [De Korvin *et al.*, 2002] De Korvin A., Shipley M.F., Kleyle R., "Utilizing fuzzy compatibility of skill sets for team selection in multi-phase projects", Journal of Engineering and Technology Management, Vol. 19, No.3-4, pp. 307-319, September 2002.

- [De Montmollin, 1986] De Montmollin M., "L'intelligence de la tâche, éléments d'ergonomie cognitive", Peter Lang, Berne, 1986.
- [Drezet et Baillaut, 2008] Drezet L.-E., Baillaut J.-C., "A project scheduling problem with labour constraints and time-dependent activities requirements", *International Journal of Production Economics*, Vol 112, pp. 217–225, 2008.
- [Eiselt et Marianov, 2008] Eiselt H.A., Marianov V., "Employee positioning and workload allocation", *Computers & Operations Research*, Vol. 35, No. 2, pp. 513-524, 2008.
- [Eppinger *et al.*, 1994] Eppinger S., Whitney D., Smith R., Gebala D., "A Model-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, Vol. 6, No. 1, pp.1-13, 1994.
- [Eppinger et Salminen, 2001] Eppinger S., Salminen V., "Patterns of product development interactions", *International Conference On Engineering Design, ICED*, Glasgow, August 21-23, Vol. 1, pp. 283-290, 2001
- [Fan *et al.*, 2009] Fan Z.-P., Feng B., Jiang Z.-Z., Fub N., "A method for member selection of R&D teams using the individual and collaborative information", *Expert Systems with Applications*, Vol. 36, pp. 8313–8323, 2009.
- [Fernandez, 1998] Fernandez C., "Integration analysis of product architecture to support effective team co-location", *Master of Science Thesis, MIT*, 1998.
- [Fitzpatrick et Askin, 2005] Fitzpatrick E.L., Askin R.G., "Forming effective worker teams with multi-functional skill requirements", *Journal of Computers & Industrial Engineering*, Vol. 48, pp. 593–608, 2005.
- [Fowler *et al.*, 2008] Fowler J.W., Wirojanagud P., Gel E.S., "Heuristics for workforce planning with worker differences", *European Journal of Operational Research* Vol.190, 2008, pp. 724–740
- [Galbraith, 1977] Galbraith, J. R., "Organization design", Reading, MA: Addison-Wesley, 1977.
- [Giard, 1991] V. Giard, "Gestion de projet", *Economica*, Paris, 1991.
- [Grabot, 2008] Grabot B., "Prise en compte des contraintes liées à la gestion des compétences en ordonnancement et simulation", *Conférence internationale MOSIM'08*, Paris 2008.
- [Gronau *et al.*, 2006] Gronau N., Fröming J., Schmid S., Rüssbüldt U., "Approach for requirement oriented team building in industrial processes", *Computers in Industry*, Vol.58, No. 2, pp. 179-187, 2006.
- [Gulati et Eppinger, 1996] Gulati R., Eppinger S., "The Coupling of Product Architecture and Organizational Structure Decisions", *Massachusetts Institute of Technology, Working Paper Number 3906*, 1996.
- [Gutiérrez – Estrada, 2007] Gutiérrez-Estrada C., "Méthodes et outils de la conception système couplée à la conduite de projet", *Thèse de doctorat, INSA Toulouse*, 06 Février 2007.
- [Gutjahr *et al.*, 2008] Gutjahr W.J., Katzensteiner S., Reiter P., Stummer C., Denk M., "Competence-driven project portfolio selection, scheduling and staff assignment", *Central European Journal of Operations Research*, Vol. 16, No. 3, pp. 281-306, September 2008.
- [Hadj-Hamou et Caillaud, 2004] Hadj-Hamou K., Caillaud E., "Cooperative design : A framework for a competency-based approach", *5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering- IDMME'04*, University of Bath, avril 2004.
- [Harmel, 2007] Harmel G., "Vers une conception conjointe des architectures du produit et de l'organisation du projet dans le cadre de l'Ingénierie Système", *Thèse de doctorat, Université de Franche-Comté*, 2007
- [Hartigan, 1975] Hartigan J., "Clustering Algorithms", *Wiley & Sons*, New York, NY, 1975.
- [Harzallah et Vernadat, 2008] Harzallah M., Vernadat F., "IT-based competency modelling and management:

- 2002] from theory to practice in enterprise engineering and operations", *Computers in Industry*, Vol. 48, pp. 157-179, 2002.
- [Hendriks *et al.*, 1999] Hendriks M.H.A., Voeten B., Kroep L., "Human resource allocation in a multi-project R&D environment : Resource capacity allocation and project portfolio planning in practice", *International Journal of Project Management*, Vol. 17, No. 3, pp. 181-188, 1999
- [Henry et Stevens, 1999] Henry S.M., Stevens K.T., "Using Belbin's leadership role to improve team effectiveness: An empirical investigation", *Journal of Systems and Software*, Vol. 44, No. 3, pp. 241-250, January 1999.
- [Hlaoittinun *et al.*, 2007a] Hlaoittinun O., Bonjour E., Dulmet M., "Méthodologie pour la constitution d'équipes de projet de conception", *Congrès International de Génie Industriel*, Trois-rivières, Québec, Canada, 5-8 June 2007.
- [Hlaoittinun *et al.*, 2007b] Hlaoittinun O., Bonjour E., Dulmet M., "A Fuzzy Array-Based Clustering Method In Team Building", 4th International Federation of Automatic Control Conference (IFAC), on Management and Control of Production and Logistics, Sibiu, Romania, 27-30 September 2007.
- [Hlaoittinun *et al.*, 2007c] Hlaoittinun O., Bonjour E., Dulmet M., "A team building approach for competency development", *IEEE International Conference on Industrial Engineering and Engineering Management*, Singapore, 2-5 December 2007
- [Hlaoittinun *et al.*, 2008] Hlaoittinun O., Bonjour E., Dulmet M., "A multidisciplinary team building method based on competency modeling in design project management", *International Journal of Management Science and Engineering Management*, Vol. 3, N° 3, 2008, pp.163-175, 2008.
- [Hlaoittinun *et al.*, 2009] Hlaoittinun O., Bonjour E., Dulmet M., "Affectation multi-périodes de tâches de conception en fonction de l'évolution des compétences", *Congrès International de Génie Industriel*, Bagnères de Bigorre, France, 10-12 juin 2009
- [Hopp et Van Oyen, 2004] Hopp W.J., Oyen M., "Agile Workforce Evaluation: A Framework for Cross-training and Coordination", *IIE Transactions*, Vol. 36, No.10, October, pp. 919-940, 2004.
- [Huang *et al.*, 2009] Huang D.K., Chiu H.N., Yeh R.H., Chang J.H., "A fuzzy multi-criteria decision making approach for solving a bi-objective personnel assignment problem", *Computers & Industrial Engineering*; Vol. 56, pp. 1–10, 2009.
- [Huth, 2008] Huth T., "Organizing Cross-Functional New Product Development: The Phase-Specific Effects of Organizational Antecedents", *Gabler Edition Wissenschaft*, pp. 7-48, 2008.
- [Idicula, 1995] Idicula J., "Planning for concurrent engineering", Thesis draft, Nanyang Technology University, 1995.
- [IEEE 1220, 2005] IEEE Std 1220™ - 2005, "Standard for application and Management of the Systems Engineering Process", *IEEE Computer Society*, Revision of IEEE Std 1220-1998.
- [Jia, 1998] Jia T., "Vers une meilleure gestion des performances d'un groupe autonome de fabrication", Thèse de doctorat, Université de Tours, 1998
- [Jin et Levitt, 1996] Jin, Y. and R.E. Levitt, "The Virtual Design Team: A Computational Model of Project Organizations", *Journal of Computational and Mathematical Organization Theory*, Vol. 2, No. 3, pp. 171-195, September 1996.
- [Kichuk et Wiesner, 1997] Kichuk S.L., Wiesner W.H., "The Big Five personality factors and team performance : implications for selecting successful product design teams", *Journal of Engineering and Technology Management*, Vol. 14, pp. 195-221, 1997.
- [King, 1980] King J.R., "Machine-Component Group Formation in Group Technology", *OMEGA Journal of Management Science*, Vol. 8, No. 2, pp. 193-199, 1980.
- [King et Nakornchai, King J.R., Nakornchai V., "An interactive data-clustering algorithm",

- 1986] International journal of Flexible manufacturing system, pp. 285-291, 1986
- [Kirkpatrick *et al.*, 1983] Kirkpatrick S., Gelatt C.D., Vecchi M.P., "Optimization by simulated annealing", *Science*, Vol. 220, No. 4598, pp. 671–680, May 1983.
- [Krackhardt et Carley, 1998] Krackhardt D., Carley K.M., " A PCANS model of structure in Organizations", *Proceedings of the 1998 international symposium on command and control research and technology*, Monterey, CA, June 1998.
- [Levy-Leboyer, 1996] Levy-Leboyer C., "La gestion des compétences", éd. organisation, 1996.
- [Ley et Albert, 2003] Ley T., Albert D., "Identifying Employee Competencies in Dynamic Work Domains: Methodological Considerations and a Case Study", *Journal of Universal Computer Science*, Vol. 9, No. 12, pp. 1500-1518, 2003.
- [Li, 2003] Li M.-L., "The algorithm for integrating all incidence matrices in multi-dimensional group technology", *International Journal of Production Economics*, Vol. 86, No. 2, pp. 121-131, November 2003.
- [Lopez et Roubellat, 2001] Lopez P., Roubellat F., "Ordonnancement de la production", Hermès Science Europe Ltd, 2001.
- [Lucia et Lepsinger, 1999] Lucia A.D., Lepsinger R., "The art and science of competency models: Pinpointing critical success factors in Organizations", *Jossey-Bass* 1999.
- [Malmqvist, 2002] Malmqvist, J., "A classification of matrix based methods for product modelling", *Design 2002*, 14-17, Cavtat-Dubrovnik, Croatia, 2002
- [McCord et Eppinger, 1993] McCord K., Eppinger S., "Managing the Integration Problem in Concurrent Engineering", M.I.T. Sloan School of Management, Cambridge, MA, Working Paper No.3594, 1993.
- [McCormick *et al.*, 2002] McCormick W.T., Schwitzer P.J., White T.W., "Problem decomposition and data reorganization by a clustering technique", *Operations Research*, Vol. 20, pp. 993-1009, 2002.
- [Meinadier, 2002] Meinadier J-P., "Le métier d'intégration de systèmes", Hermès Sciences Publications, 2002.
- [Miller et Franz, 1996] Miller J.L., Franz L.S., "A binary-rounding heuristic for multi-period variable-task-duration assignment problems", *Computers and Operations Research*, Vol. 23, No. 8, pp. 819-828, August 1996.
- [Morelli *et al.*, 1995] Morelli M., Eppinger S., Gulati R., "Predicting technical communication in product development organizations", *IEEE Trans. Eng. Management*, Vol. 42, pp. 215-222, Aug. 1995.
- [Nembhard, 2001] Nembhard D.A., "Heuristic approach for assigning workers to tasks based on individual learning rates", *International Journal of Production Research.*, Vol. 39, No. 9, pp. 1955-1968, 2001.
- [Nembhard et Uzumeri.,2000] Nembhard D.A., Uzumeri M.V., "Experiential learning and forgetting for manual and cognitive tasks", *International Journal of Industrial Ergonomics*, Vol. 25, No. 4, pp. 315-326, May 2000.
- [Oosterman, 2001] Oosterman B. "Improving Product Development Projects by Matching Product Architecture and Organization", PhD thesis, Groningen University, The Netherlands, ISBN: 90-72591-99-2, 2001.
- [Otero *et al.*, 2009] Otero L.D., Centeno G., Ruiz-Torres A.J., Otero C.E., "A systematic approach for resource allocation in software projects", *Computers & Industrial Engineering*, Vol. 56, pp. 1333–1339, 2009.
- [Pentico, 2007] Pentico D.W., "Assignment problems: A golden anniversary survey", *European Journal of Operational Research*, Vol. 176, No. 2, pp. 774-779, 2007.
- [Peters et Zelewski, 2007] Peters M.L., Zelewski S., "Assignment of employees to workplaces under consideration of employee competences and preferences", *Journal of Management Research News*, Vol. 30, pp. 84-99, 2007.

- [Pimmler et Eppinger, 1994] Pimmler T., Eppinger S., "Integration Analysis of Product Decompositions", Proc. ASME Design Theory and Method Conference (DTM'94), Vol. 68, pp. 343-351, 1994.
- [Plaza et Rohlf, 2008] Plaza M., Rohlf K., "Learning and performance in ERP implementation projects: A learning-curve model for analyzing and managing consulting costs", International Journal of Production Economics, Vol. 115, pp. 72– 85, 2008.
- [Plaza *et al.*, 2009] Plaza M., Ngwenyama O.K., Rohlf K., "A comparative analysis of learning curves: Implications for new technology implementation management", European Journal of Operational Research, 2009.
- [PMI, 2004] Project Management Institute, "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)" - Third Edition, ISBN: 193069945X, 388p., 2004.
- [Robin *et al.*, 2007] Robin V., Rose B., Girard Ph., "Modelling collaborative knowledge to support engineering design project manager", Computers in Industry, Vol. 58, No. 2, pp. 188-198, 2007.
- [Rose *et al.*, 2006] Rose B., Robin V., Caillaud E., Girard Ph., "Comment répondre aux challenges de la gestion des compétences en conception collaborative de produits", Revue Française de Gestion Industrielle, Vol. 26, No. 2, pp. 10-15, 2006.
- [Sanchez and Mahomey, 1996] Sanchez R., Mahomey J.T., "Modularity, flexibility and knowledge management in product and organization design", Strategic Management Journal, Vol. 17, pp. 63-76, 1996.
- [Sayin et Karabati, 2007] Sayin S., Karabati S., "Assigning cross-trained workers to departments: A two-stage optimization model to maximize utility and skill improvement", European Journal of Operational Research, Vol. 176, No. 3, pp. 1643–1658, 2007.
- [Sharifi et Pawar, 1996] Sharifi S., Pawar K.S., "Product design as a means of integrating differentiation", Technovation, Vol. 16, No. 5, pp. 255-264, 1996.
- [Shipley et Johnson, 2009] Shipley M.F., Johnson M., "A fuzzy approach for selecting project membership to achieve cognitive style goals", European Journal of Operational Research, Vol.192, pp. 918–928, 2009.
- [Sommerville et Dalziel, 1998] Sommerville J., Dalziel S., "Project Teambuilding the applicability of Belbin's team role self-perception inventory", International Journal of Project Management, Vol 16, No. 3, pp. 165–171, 1998.
- [Sosa *et al.*, 2000] Sosa, M.E., Eppinger, S.D. et Rowles C.M., 'Designing modular and integrative systems', Proceedings of DETC '00: ASME 2000 International Design Engineering Technical Conferences, 2000.
- [Sosa *et al.*, 2003] Sosa M., Eppinger S., Rowles C., "Identifying modular and integrative systems and their impact on design team interactions", Transactions of the ASME, Vol. 125, pp. 240-252, 2003
- [Sosa *et al.*, 2004] Sosa, M.E., Eppinger, S.D. et Rowles C.M., "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development", Management Science, Vol. 50, No. 12, December 2004, pp. 1674–1689, 2004
- [Sosa, 2007] Sosa M., "Aligning Process, Product, and Organizational Architectures in Software Development", Int. Conf. Engineering Design, ICED'07, Paris, 2007.
- [Stickley et Grabot, 1994] Stickley A., Grabot B., "Fuzzy logic in team design", Conference IFIP -- Benchmarking Theory and Practice, Trondheim, June, 1994.
- [Suman et Kumar, 2006] Suman B., Kumar P., "A survey of simulated annealing as a tool for single and multiobjective optimization", Journal of the Operational Research Society, Vol. 57, No. 10, pp. 1143-1160, October 2006.
- [Tang *et al.*, 2000] Tang D., Zheng L., Lia Z., Lib D., Zhang S., "Re-engineering of the design process for concurrent engineering", International Journal of Computers &

- Industrial Engineering, Vol. 38, pp 479-491, 2000.
- [Thebeau, 2001] Thebeau R., "Knowledge management of system interfaces and interactions for product development processes", Master of Science Thesis, MIT, 2001.
- [Tsai *et al.*, 2003] Tsai H-T., Moskowitz H., Lee L-H., "Human resource selection for software development projects using Taguchi's parameter design", *European Journal of Operational Research*, Vol. 151, 2003, pp. 167-180.
- [Tseng *et al.*, 2004] (Bill) Tseng T-L., Huang C-C., Chu H-W., Gung R.R., "Novel approach to multi-functional project team formation", *International Journal of Project Management*, Vol. 22, No. 2, pp. 147-159, 2004.
- [Tuckman, 1965] Tuckman B.W., "Developmental Sequence in Small Groups", *Psychological Bulletin*, Vol. 63, No. 6, pp. 384-99, 1965.
- [Ulrich et Tung, 1991] Ulrich K., Tung K., "Fundamentals of product modularity", *Proceeding of ASME Winter Annual Meeting Symposium on Design and Manufacturing Integration*. pp.73-79. Atlanta, GA. November 1991.
- [Ulrich, 1995] Ulrich K., "The role of product architecture in the manufacturing firm", *Research Policy*, Vol. 24, pp. 419-440, 1995.
- [Vairaktarakis *et al.*, 2002] Vairaktarakis G.L., Cai X., Lee C-Y., "Workforce planning in synchronous production systems", *European Journal of Operational Research*, Vol. 136, pp. 551-572, 2002.
- [Warfield, 1973] Warfield J., "Binary Matrices in System Modeling", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, pp. 441-449, 1973.
- [Wheelwright et Clark, 1992] Wheelwright S., Clark K., "Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality", Free Press, New York, NY, 1992.
- [Whitfield *et al.*, 2005] Whitfield R.I., Duffy A., Gartzia-Etxabe L., "Identifying and evaluating parallel design activities using Design Structure Matrix", *Proc. ICED 05 Melbourne*, August 15-18, 2005
- [Wi *et al.*, 2009] Wi H., Oh S., Muna J., Jung M., "A team formation model based on knowledge and collaboration", *Expert Systems with Applications*, Vol. 36, pp. 9121-9134, 2009.
- [Wright, 1936] Wright T., "Factors Affecting the Cost of Airplanes", *Journal of Aeronautical Science*, Vol. 4, No. 4, pp. 122-128, 1936.
- [Wu, 2007] Wu Y.K., "On the manpower allocation within matrix organization: A fuzzy linear programming approach", *European Journal of Operational Research*, Vol. 183, pp. 384-393, 2007.
- [Yassine et Braha, 2003] Yassine A., Braha D., "Complex concurrent engineering and Design Structure Matrix method", *Concurrent Engineering: Research and Applications*, Vol. 11, No. 3, pp. 165-176, 2003.
- [Yassine et Wissman, 2007] Yassine A., Wissmann L., "The Implications of Product Architecture on the Firm", *Journal of Systems Engineering*, Vol. 10, No. 2, pp.118-137, Summer 2007.
- [Zakarian et Kusiak, 1999] Zakarian A., Kusiak A., "Forming teams: an analytical approach", *IIE Transaction*, Vol. 31, pp. 85-97, 1999.
- [Zülch *et al.*, 2004] Zülch G., Rottinger S., Vollstedt T., "A simulation approach for planning and re-assigning of personnel in manufacturing", *International Journal of Production Economics*, Vol. 90, pp. 265-277, 2004.
- [Zülch et Rottinger, 2007] Zülch G., Rottinger S., "Approach for personnel development planning based on the technology calendar concept", *International Journal of Production Economics*, Vol. 105, pp. 273-281, 2007.

TABLE DES FIGURES

FIG. 1.1 : THEMES DE RECHERCHE LIES A LA CONSTITUTION D'EQUIPES	10
FIG. 1.2 : DSM ACTEURS (1) AVANT ET (2) APRES CLUSTERING	16
FIG. 1.3 : POSITIONNEMENT GLOBAL DE NOTRE RECHERCHE	24
FIG. 2.1 : CATEGORIES DE LIENS DE COMPETENCES DANS LE RESEAU SEMANTIQUE [GRONAU ET AL, 2006].....	31
FIG. 2.2 : DEMARCHE DE CONSTITUTION D'EQUIPES [TSENG ET AL., 2004].....	37
FIG. 3.1 : EXEMPLE D'UN GRAPHE DE PRECEDENCE ET UNE SOLUTION OPTIMALE DE RCPSP (GoTHA).....	51
FIG. 3.2 : EXEMPLE D'UN GRAPHE DE PRECEDENCE ET UNE SOLUTION OPTIMALE DE L'ORDONNANCEMENT [BELLENGUEZ-MORINEAU, 2006]	52
FIG. 4.1 : VUE EXTERNE DE LA DEMARCHE D'AFFECTION MONO-PERIODE DES TACHES AUX ACTEURS	68
FIG. 4.2: DEMARCHE D'AFFECTION MONO-PERIODE DES TACHES AUX ACTEURS.....	69
FIG. 4.3 : NIVEAU DE MAITRISE DANS LA CONNAISSANCE DE LA TACHE PARTICULIERE.....	72
FIG. 4.4 : TEMPS LIES A LA REALISATION D'UNE TACHE.....	78
FIG. 4.5 : EXEMPLE D'UNE PLANIFICATION DES TACHES PROVENANT DE 2 PROJETS SUR 3 PERIODES	91
FIG. 4.6 : VUE EXTERNE DU PROBLEME D'AFFECTION MULTI-PERIODES, MULTI-PROJETS	95
FIG. 4.7 : DEMARCHE D'AFFECTION MULTI-PERIODES DES TACHES A DES ACTEURS.....	96
FIG. 4.8 : DEMARCHES "DESCENDANTE" ET "ASCENDANTE" DE L'EVOLUTION DES COMPETENCES	101
FIG. 4.9 : DEMARCHE ASCENDANTE ET LOIS D'EVOLUTION DES COMPETENCES	102
FIG. 4.10 : AUGMENTATION MAXIMALE ($DIF_{c,j}^k$).....	104
FIG. 4.11 : CHEMIN LOGIQUE DANS LE CHOIX DE CHANGEMENT DU NIVEAU DANS LA CONNAISSANCE.....	106
FIG. 4.12 : OPTIMISATION PAR MORCEAUX PAR LA PROGRAMMATION LINEAIRE (PL).....	116
FIG. 4.13 : OPTIMISATION PAR LE RECUIT SIMULE.....	117
FIG. 4.14 : ALGORITHME DU RECUIT SIMULE UTILISE.....	119
FIG. 4.15 : EVOLUTION DU NIVEAU DE MAITRISE ACQUIS DANS LA CONNAISSANCE DE L'ACTEUR A15.....	124
FIG. 4.16 : AXE COMPETENCE - EVOLUTION DU NIVEAU DE COMPETENCE ACQUIS PAR L'ACTEUR A15	125
FIG. 4.17 : AXE FINANCIER – VARIATION DES COUTS.....	125
FIG. 4.18 : AXE FINANCIER – COMPROMIS ENTRE LES COUTS LIES AU SALAIRE, A L'APPRENTISSAGE ET A L'OBJECTIF DE COMPETENCES	126
FIG. 5.1: COUPLAGE DES ASPECTS TACHE-ACTEUR-COMPETENCE DANS LA CONSTITUTION D'EQUIPES	133
FIG. 5.2 : SCHEMA DE LA GENERATION D'UNE DSM PAR UNE METHODE DE PROPAGATION	135
FIG. 5.3 : PRINCIPE DE LA METHODE D'IDENTIFICATION DES INTERACTIONS ENTRE DES ACTEURS.....	136
FIG. 5.4: IDENTIFICATION D'UNE INTERACTION DU DOMAINE B A TRAVERS LE DOMAINE A.....	137
FIG. 5.5 : COUPLAGE DES MATRICES	138
FIG. 5.6 : DSM TACHES (13x13).....	141
FIG. 5.7: MI TACHES-ACTEURS (13x16).....	142
FIG. 5.8: ARCHITECTURE DES ACTEURS APRES CLUSTERING DES $DSM_{OPT}(A)_{(A)}$	142
FIG. 5.9 : ARCHITECTURE DES TACHES APRES CLUSTERING DE $DSM_{SIM}(T)_{(A=50)}$	143
FIG. 5.10 : DEMARCHE GLOBALE DE LA CONSTITUTION D'EQUIPES	146
FIG. 5.11 : PREPARATION DES DSM TACHES ET DES MI-AFFECTATIONS	147
FIG. 5.12 : EXEMPLE DE L'AGREGATION DES MI-AFFECTATIONS	148
FIG. 5.13 : DSM TACHE GENERIQUE (15x15).....	149
FIG. 5.14 : DSM TACHE POUR CHAQUE PROJET	149
FIG. 5.15 : MI-AFFECTATION AGREGEE (INITIALE) DU PROJET 1 ($L=1$)	150
FIG. 5.16 : MI-AFFECTATION AGREGEE (INITIALE) DU PROJET 2 ($L=2$)	150
FIG. 5.17 : DSM ACTEURS INITIALES	151
FIG. 5.18: ALGORITHME DU RECUIT SIMULE UTILISE.....	154
FIG. 5.19 : MI-AFFECTATION OPTIMALE (AGREGEE) DU PROJET 1 ($L=1$).....	156
FIG. 5.20 : MI-AFFECTATION OPTIMALE (AGREGEE) DU PROJET 2 ($L=2$).....	156
FIG. 5.21 : DSM ACTEURS OPTIMALES.....	157
FIG. 5.22 : ARCHITECTURE DES ACTEURS SIMULEE	158
FIG. 5.23 : AXE FINANCIER – VARIATIONS DES COUTS.....	159

TABLE DES TABLEAUX

TAB. 1.1 : USAGES DES DSM.....	14
TAB. 1.2 : TYPOLOGIE DES THEMES DU PILOTAGE DES COMPETENCES [BOUCHER <i>ET AL.</i> , 2006]	23
TAB. 2.1 : SYNTHESE DE L'ETAT DE L'ART SUR LA CONSTITUTION D'EQUIPES	38
TAB. 3.1 : CLASSIFICATION DES PROBLEMES D'AFFECTION	62
TAB. 3.2 : REPARTITION DES TRAVAUX DE CONSTITUTION D'EQUIPES A CONTRAINTES DE COMPETENCES.	64
TAB. 4.1 : VARIABLES DANS L'AFFECTION MONO-PERIODE	68
TAB. 4.2 : EXEMPLE DU NIVEAU DE MAITRISE DE LA CONNAISSANCE C REQUIS PAR LA TACHE GENERIQUE I.....	70
TAB. 4.3 : EXEMPLE DU NIVEAU DE MAITRISE DE LA CONNAISSANCE REQUIS PAR LA TACHE PARTICULIERE	71
TAB. 4.4 : CARACTERISATION DES TACHES ET DES ACTEURS	73
TAB. 4.5 : INDICATEURS DE SIMILARITE ENTRE LA TACHE GENERIQUE ET LES ACTEURS	75
TAB. 4.6 : INDICATEUR DE SIMILARITE ENTRE LA TACHE PARTICULIERE ET LES ACTEURS	75
TAB. 4.7 : TEMPS DE TRAVAIL CONSIDERES	78
TAB. 4.8: LISTE DES TACHES GENERIQUES	81
TAB. 4.9: LISTE DES CONNAISSANCES (REQUISES)	82
TAB. 4.10: ACTEURS ET TAUX DE SALAIRE	83
TAB. 4.11 : MI TACHE-CONNAISSANCE.....	85
TAB. 4.12: MI ACTEUR-CONNAISSANCE.....	85
TAB. 4.13 : CHARGE INDUITE PAR LA TACHE	86
TAB. 4.14 : TAUX DE SALAIRE DES ACTEURS	86
TAB. 4.15 : MATRICE DE COMPETENCE ET MATRICE DE COEFFICIENT CORRECTEUR	87
TAB. 4.16 : MATRICE D'AFFECTION $x_{i,j}, (rq_i = 1)$	87
TAB. 4.17: RESULTATS DE L'AFFECTION.....	88
TAB. 4.18 : ESTIMATION QUALITATIVE DE LA METHODE.....	90
TAB. 4.19: PLANIFICATION MULTI-PERIODES ET MULTI-PROJETS DES TACHES	94
TAB. 4.20 : CARACTERISATION DES TACHES ET DES ACTEURS (MULTI-PERIODES)	98
TAB. 4.21 : VARIABLES DE LA MODELISATION DES COMPETENCES	100
TAB. 4.22: SOLUTION D'AFFECTION OPTIMALE (RECUIT SIMULE).....	121
TAB. 4.23: EVOLUTION DU NOMBRE DES ACTEURS COMPETENTS.....	122
TAB. 4.24 : NIVEAU DE COMPETENCE ACQUIS CONCERNANT LA TACHE T7, $(r_{i,j}^k)$	123
TAB. 4.25 : CALCUL DE LA CHARGE DES ACTEURS.....	123
TAB. 4.26 : ANALYSE DES RESULTATS	127
TAB. 4.27: ESTIMATION QUALITATIVE DE LA METHODE	128
TAB. 5.1. LISTE DES ELEMENTS UTILISES ET LEURS ABREVIATIONS	141
TAB. 5.2 : ESTIMATION QUALITATIVE DE L'OPTIMISATION DE L'ARCHITECTURE DES ACTEURS	144
TAB. 5.3 : SOLUTION D'AFFECTION FINALE (RECUIT SIMULE).....	155
TAB. 5.4: EVOLUTION DU NOMBRE DES ACTEURS COMPETENTS.....	160
TAB. 5.5 : ANALYSE DES RESULTATS	162
TAB. 5.6: ESTIMATION QUALITATIVE DE LA DEMARCHE DE CONSTITUTION D'EQUIPES.....	164

8 Annexe

LISTE DES ANNEXES

8.1	ANNEXE 1 : REPRESENTATIONS MATRICIELLES	181
8.2	ANNEXE 2 : AFFECTATION MONO-PERIODE.....	186
8.3	ANNEXE 3 : AFFECTATION MULTI-PERIODES.....	188
8.4	ANNEXE 4 : CONSTITUTION D'EQUIPES	199
8.5	ANNEXE 5 : FONCTION DE REGRESSION DES COMPETENCES.....	205
8.6	ANNEXE 6 : CONVEXITE DU PROBLEME QUADRATIQUE	214

8.1 Annexe 1 : Représentations matricielles

[Malmqvist, 2002] distingue deux classes d'outils matriciels : les matrices intra-domaine (Design Structure Matrix) et les matrices inter-domaines (matrice d'incidence).

Représentation des architectures par l'outil DSM

L'utilisation des matrices dans le domaine de la modélisation des systèmes remonte aux travaux de Warfield dans les années 70 [Warfield, 1973] et de Steward dans les années 80 [Steward, 1981]. Cependant, ce n'est que dans les années 90 que cette méthode a reçu une attention de plus en plus grande et une diffusion importante et ce, sous l'impulsion principale des chercheurs du MIT qui ont utilisé cette méthode dans le domaine de la modélisation des architectures en conception.

Les matrices binaires de Warfield et Steward sont des matrices d'adjacence (existence ou non d'un arc entre deux sommets dans un graphe) dans lesquelles les éléments sur la diagonale n'ont aucune signification (les boucles réflexives sont interdites). La Figure 1 montre l'architecture d'un système représenté par une DSM.

	A	B	C	D	E	F	G	H	I
A		x		x		x		x	
B	x		x	x				x	
C		x		x			x		x
D	x	x	x					x	
E						x		x	
F	x				x		x	x	
G			x			x			x
H	x	x		x	x	x			x
I			x				x	x	

Figure 1: Exemple de représentation d'une architecture

La classification des DSM est présentée ci-dessous (Figure 2). Elle met en évidence deux types de DSM : statique et temporelle.

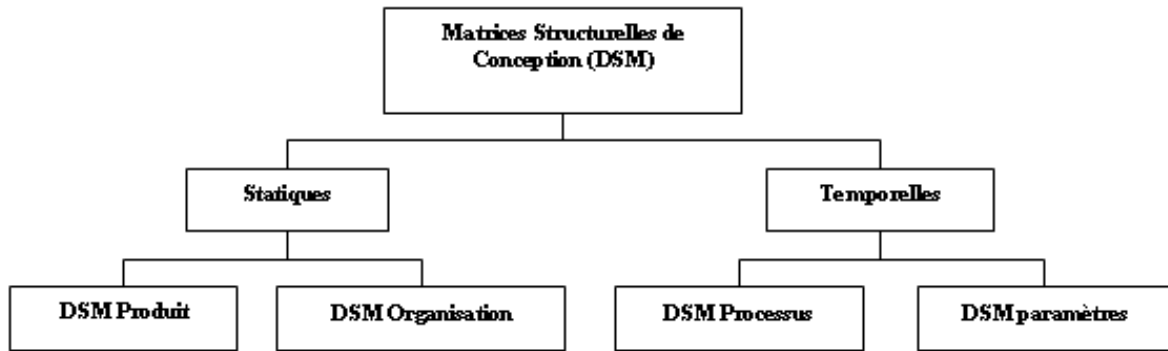


Figure 2: Classification des DSM

DSM statiques

Les DSM statiques tirent leur nom du fait qu'elles sont utilisées pour matérialiser les interactions entre les éléments d'un système, indépendamment du temps. Deux types de DSM statiques existent : les DSM Produit et les DSM Organisation (ou acteurs).

Une DSM Produit modélise les interactions entre les constituants d'un produit. Pimmler et Eppinger [1994] proposent une taxonomie permettant de différencier les différents types d'interactions. Historiquement, les DSM statiques binaires ont été choisies pour modéliser l'architecture des produits et modéliser les interactions qui lient leurs composants. Ces DSM permettent d'identifier l'existence ou non d'un type d'interaction étudiée.

Une DSM "Organisation"⁴⁶ (ou Acteur) permet de modéliser les interactions et efforts de coordination entre les acteurs dans une équipe de conception. Parmi les travaux les plus récents utilisant les DSM Organisation binaires, on citera ceux de [Sosa *et al.*, 2004], qui proposent une taxonomie des interactions entre acteurs.

DSM temporelles

Les DSM temporelles permettent de représenter les systèmes dont les éléments sont liés par des relations de causalité ou de précédence. L'objectif général lorsqu'on analyse des DSM temporelles est de réorganiser les éléments en ligne et en colonne de manière à réduire les retours en arrière. On distingue les DSM Processus et les DSM Paramètres.

Les DSM Processus permettent de modéliser l'enchaînement des tâches des processus de l'entreprise. Les deux grands domaines d'application des DSM temporelles sont : optimisation des processus du point de vue coût et délais et optimisation des processus pour la conception des équipes de conception.

Les DSM Paramètres ont été initialement utilisées pour modéliser les relations de précédence et les contraintes entre les paramètres et variables en conception. Elles permettent d'optimiser les boucles de conception entre groupes de paramètres

Représentation des couplages entre domaines par une Matrice d'Incidence

⁴⁶ Terme utilisé dans les articles anglo-saxons, il pourrait être ambigu dans notre travail où nous considérons l'organisation comme l'architecture des processus, l'affectation de tâches à un acteur et l'architecture des échanges entre acteurs. Aussi, nous avons choisi de le remplacer par le terme Acteurs.

Les matrices inter-domaines permettent de représenter les couplages existant entre deux domaines. Nous les appelons dans ce mémoire Matrice d'Incidence (MI). La Figure 3 présente un exemple de MI représentant les couplages entre les éléments de deux domaines A et B. D'autres auteurs parlent de matrices d'allocation et de traçabilité [IEEE 1220, 2005] ou de projection [Meinadier, 2002] ou de "Domain Mapping Matrix (DMM)" [Danilovic et Browning, 2007] ou encore de "affiliation matrix" [Sosa, 2007].

Avec les MI et les DSM, on modélise un système complexe composé de plusieurs sous-domaines (Figure 4). Elles sont ainsi complémentaires puisque l'outil DSM représente les couplages internes à tout sous-domaine et les MI, les couplages externes entre éléments appartenant à des sous-domaines différents. Il est alors possible de construire une chaîne de matrices DSM-MI pour relier de multiples domaines.

	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉
B ₁	x		x		x		x	x	
B ₂	x	x							x
B ₃	x			x		x			x
B ₄				x				x	
B ₅		x	x		x		x		
B ₆				x				x	
B ₇	x		x				x		x

Figure 3 : Exemple de matrice d'incidence

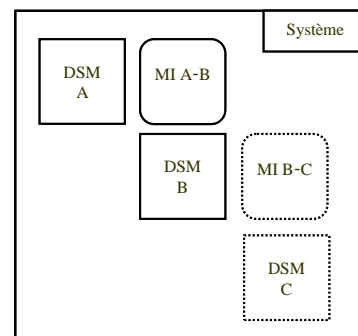


Figure 4 : Modélisation multi-domaines

Dans le cadre de la modélisation du projet de conception à travers des chaînes DSM-MI, [Danilovic et Browning, 2007] introduisent deux systèmes à modéliser : système-produit et système-projet.

Système-produit (Figure 5a) : la modélisation du produit couple un niveau abstrait (une fonction du produit) avec un niveau concret (sous-systèmes, paramètres etc.). Les auteurs proposent la modélisation d'un produit d'une part, par les couplages internes de cinq domaines différents (au moyen de DSM), et d'autre part, par les couplages externes appartenant à ces cinq domaines (au moyen de MI). Les différentes DSM sont : DSM Exigence du client, DSM Fonction, DSM Paramètre, DSM Spécification et DSM produit (composant). La synchronisation des informations inter-domaines (MI) du projet permet aux chefs du projet d'avoir une vision plus globale sur le système à concevoir.

Système-projet (Figure 5b) : Les auteurs proposent de modéliser un projet par les couplages internes de cinq domaines (DSM Objectif, DSM Produit, DSM Processus, DSM Organisation⁴⁷ et DSM Outil) et par les couplages externes (MI) appartenant à ces cinq domaines. Parmi ces cinq DSM, trois DSM (produit, processus du projet et organisation) sont utilisés traditionnellement. La modélisation par des MI permet d'avoir des informations supplémentaires. Par exemple, une MI couplant la DSM Processus du projet et la DSM Organisation est interprétée comme une matrice d'affectation des tâches à des ressources humaines. Une MI couplant la DSM Produit et la DSM Organisation est une matrice d'affectation des composants du produit aux ressources humaines.

⁴⁷ DSM Organisation est ce que nous appelons aussi DSM acteurs

[Krackhardt et Carley, 1998] modélisent trois domaines : tâche (T), acteur (I) et ressource matérielle (R). La modélisation de ces trois domaines résulte de six types de représentations matricielles (DSM et MI) sous l'acronyme «PCANSS»: Precedence (P), Commitment (C), Affectation (A), Network (N), Skill (S), Substitut (S). Les six types de représentations sont les suivantes :

- Matrice de précédence (P) est représentée par une DSM mettant en relation des tâches, $P=T \times T$. Elle correspond à la DSM tâche (processus).
- Matrice d'allocation des ressources matérielles aux tâches (C) est représentée par une MI mettant en relation des ressources matérielles et des tâches, $C=T \times R$.
- Matrice d'affectation des tâches aux acteurs (A) est représentée par une MI mettant en relation tâches-acteurs, $A=I \times T$.
- Matrice d'organisation (N) est représentée par une DSM mettant en relation des acteurs, $N=I \times I$. Elle correspond à la DSM organisation.
- Matrice de compétence (S) est représentée par une MI mettant en relation acteurs-ressources matérielles, $S=I \times R$.
- Matrice de substitution des ressources matérielles (S) est représentée par une DSM mettant en relation des ressources matérielles, $S=R \times R$.

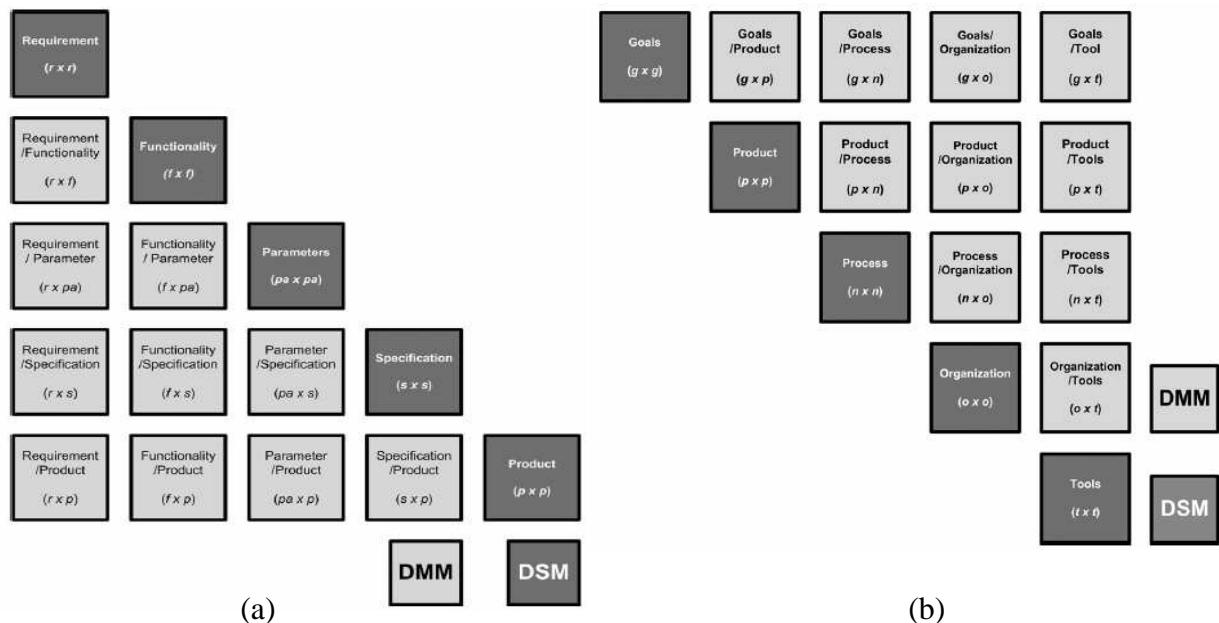


Figure 5: DSM et MI dans la modélisation du système-produit et du système-projet [Danilovic et Browning, 2007]

Il s'agit des algorithmes de *clustering* qui portent directement sur des matrices d'incidence. Une caractéristique commune à ces algorithmes est qu'ils réorganisent consécutivement les lignes et les colonnes selon un indice de similarité jusqu'à l'obtention de blocs diagonaux. A l'origine, ces algorithmes ont été développés pour le regroupement de machines en cellules de production (Group Technology) capables de produire un ensemble homogène de pièces. Les exemples d'algorithmes sont : ROC (Rank Order Clustering) [King, 1980], ROC2 [King et Nakornchai, 1986], BEA (Bond-Energy Algorithm) développé par [McCormick *et al.*, 2002]. Ces algorithmes utilisent une matrice d'incidence de type machine-composant, où un élément

$a_{i,j}$ dans la matrice A, est égale à "1" si la machine i est utilisée pour transformer le composant j, ou à "0" sinon. Ces algorithmes réarrangent les lignes et les colonnes pour former la matrice finale, où les machines qui transforment le même groupe de composants sont proches les unes des autres. En général, les valeurs non nulles dans A sont rassemblées autour de la diagonale de la matrice. Cet arrangement facilite l'identification des cellules manufacturières. Cependant, une fois la matrice A formée, l'identification des groupes reste visuelle.

8.2 Annexe 2 : Affectation mono-période

La formulation mathématique du modèle d'affectation mono-période dans MATLAB est présentée ci-après. Notons qu'il s'agit de l'utilisation de la fonction `linprog` permettant de résoudre le problème linéaire.

Fichier : `binary.m`

```
function[Iopt,cout_total,F_obj,F01,F02,A,Aeq,b,beq]=binary(Delta1,L1,LM,S1,ST)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Formulation de la programmation linéaire
%
%           min z = c*x
%           avec   : Ax <= b
%                   x >= 0
%
% La fonction objectif optimise deux coûts
% 1) coût des acteurs
% 2) coût du tuteur
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la fonction objectif (f)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m=size(Delta1,1);
n=size(Delta1,2);
F01=zeros(m,n);
for i=1:m
    for j=1:n
        F01(i,j)=L1(i)*S1(i,j)*Delta1(i,j);
    end
end
F02=zeros(m,n);
for i=1:m
    for j=1:n
        F02(i,j)=(Delta1(i,j)-1)*L1(i)*ST;
    end
end
end

% Coût total
F_obj=F01+F02;

% Mettre en une seule ligne
i=1;j=1;
g=F_obj(1,1:n);
for i=2:m
    g=[g F_obj(i,1:n)];
end
f = g';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Construction de la matrice contrainte A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% - Contrainte liée à la capacité des acteurs
% - Equation : somme(toutes les tâches)*d(i,j)<= capacité de chaque acteur
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=zeros(n,n);
for k=1:m
    for i=1:n
        A(i,i)=L1(k)*Delta1(k,i);
    end
    if k==1
        A1=A;
    else
        A1=[A1 A];
    end
end
end
A=A1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la matrice contrainte Aeq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% - Contrainte liée à la charge de la tâche
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B=zeros(m,n);
for k=1:m
    for i=1:m
        if i==k
            B(i,:)=L1(i)*ones(1,n);
        end
    end
end
```

```

        else
            B(i,:)=zeros(1,n);
        end
    end
    if k==1
        C1=B;
    else
        C1=[C1,B];
    end
end
Aeq=C1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création du vecteur (b)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% correspond à la capacité maximale des acteurs
b=LM*ones(n,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création du vecteur (beq)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% correspond à la charge des tâches
beq=L1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la borne inférieure (lb)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lb=zeros(m*n,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la borne supérieure (ub)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ub=ones(m*n,1);
[Xopt,cout_total,exitflag,output,lambda]= linprog(f,A,b,Aeq,beq,lb,ub);
disp(['Le coût salaire est de : ', num2str(cout_total)]);

Iopt=zeros(m,n);
for i=1:m
    Iopt(i,:)=Xopt(1+(n*(i-1)):i*n);
end
for i=1:m
    for j=1:n
        if Iopt(i,j)<0.01
            Iopt(i,j)=0;
        end
    end
end
end

```

8.3 Annexe 3 : Affectation multi-périodes

La programmation de recuit simulé pour résoudre l'affectation multi-périodes sous MATLAB est présentée ci-après.

Fichier : recuit.m

```
Function[Iopt,cout_tt,cout_total,cout_sa,cout_salaire,cout_tut,cout_tuteur,cout_pns,cout_penal
isant]=recuit(I0,arret_simul,RQ,L,LM,task_disp,Rinit)

%% Initialisation des paramètres
unsurT=0;
nbechec=0;
ordre=I0;
load W % W est le poids d'importance
[x,y,z,w]=f(ordre,RQ,L,LM,task_disp,Rinit);
c11=W1*y;c12=W2*z;c13=W3*w; % coûts liés au salaire
f1=c11+c12+c13; % f1 est le coût total initial
fopt=f1; % fopt sauvegarde le meilleur cout
Iopt=ordre; % Iopt sauvegarde la meilleure matrice I
cb1=c11;cb2=c12;cb3=c13; % cb1 sauvegarde le meilleur coût

%% Programme principal
for stop = 0 : arret_simul % arret_simul est nombre de décroissance de température
    unsurT = unsurT+0.1; % augmente unsurT revient à diminuer la température
    Palier=2000; % nombre d'échanges
    n=0;
    while (n<Palier)
        [ordre2,nbechec]=voisin(ordre,RQ,L,LM,task_disp);
        [f2,x,y,z]=f(ordre2,RQ,L,LM,task_disp,Rinit);
        c21=W1*x;c22=W2*y;c23=W3*z;
        f2=c21+c22+c23;
        if (f2<=f1)
            ordre=ordre2;
            f1=f2;
            c11=c21;c12=c22;c13=c23;
            if (f1<=fopt)
                Iopt=ordre;
                fopt=f1;
                cb1=c11;cb2=c12;cb3=c13;
            end
        else
            f1_f2=(c11-c21)+(c12-c22)+(c13-c23);
            proba=exp(-(f1_f2)*unsurT);
            random=rand();
            if(random<proba)
                ordre=ordre2;
                f1=f2;
            end
        end
        n=n+1;
        disp(stop)
        disp(n)
        cout_total(stop+1,n)=fopt;
        cout_salaire(stop+1,n)=cb1;
        cout_tuteur(stop+1,n)=cb2;
        cout_penalisant(stop+1,n)=cb3;
        save('couts','cout_salaire','cout_tuteur','cout_total','cout_penalisant')
    end
    stop=stop+1;
end
```

Les résultats tirés de MATLAB

A. Données de départ

- MI-affectations initiales ou solution admissible par la programmation linéaire, (3x2 matrices)

B. Données de sortie

- MI-affectations optimales par la méthode du recuit simulé (3x2 matrices)
- Indicateur de similarité à priori, V3, (3x2 matrices)
- Indicateur de similarité à posteriori, R3, (4 matrices)
- Evolution du niveau de maîtrise dans la connaissance, R2, (4 matrices)

Matrices d'affectation initiales (PL)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,3	0,7
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI (k=1, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,9	0,1	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0,4	0,6	0	0	0	0	0	0
T9	0	0	0	0	0	0,8	0	0,2	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0	0	0,9	0	0

MI (k=1, p=2)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0,5	0	0	0	0	0	0	0,5	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0	0	0	0	0	0	0,2
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
T9	0	0	0	0	0	0,9	0	0,1	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,6	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI (k=2, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0,4	0	0	0	0	0,6	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

MI (k=2, p=2)

Matrices d'affectation initiales (PL) (suite)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI (k=3, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI (k=3, p=2)

Matrices d'affectation finales (recrut simulé)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,7
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=1, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,9	0,1	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0,2	0,8	0	0	0	0	0	0	0
T9	0	0	0	0	0	0,6	0,2	0,2	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0,2	0	0	0	0	0,4	0	0	0,4	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0	0,9	0	0	0

MI opt (k=1, p=2)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0,5	0	0	0	0	0	0	0,5	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0,2	0	0,2	0,4	0	0	0	0	0,2
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
T9	0	0	0	0	0	0,5	0	0,3	0	0	0	0	0,2	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0,2	0,8	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0,2	0	0	0	0	0	0	0	0,2	0,6	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=2, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,4	0	0	0,6	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

MI opt (k=2, p=2)

Matrices d'affectation finales (recuit simulé) (suite)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0,2	0,2	0,4	0	0	0	0	0,2
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=3, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=3, p=2)

Indicateur de similarité à priori, (V3) (suite)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
T2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
T3	0,4	0,6	0,3	0,5	0,6	0,5	0,4	0,3	0,3	0,2	0,2	0,8	0,7	0,5	0,5	0,5	0,6	0,5	0,7	0,9
T4	0,6	0,6	0,4	0,4	0,5	0,4	0,2	0,1	0,4	0,2	0,1	0,5	0,4	0,3	0,4	0,4	0,5	0,3	0,9	1,0
T5	0,3	0,4	0,3	0,8	1,0	1,0	0,8	0,7	0,3	0,2	0,1	0,8	1,0	0,5	0,7	0,6	0,5	0,9	0,6	0,8
T6	0,2	0,5	0,3	0,5	0,6	0,5	0,5	0,7	0,4	0,3	0,1	0,6	0,6	0,7	0,5	1,0	1,0	0,9	0,5	0,6
T7	0,3	0,7	0,4	0,5	0,7	0,6	0,3	0,4	0,9	0,6	0,4	1,0	0,9	0,9	0,9	0,6	0,6	0,7	0,8	0,9
T8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
T9	0,1	0,4	0,3	0,8	1,0	1,0	1,0	0,9	0,1	0,3	0,1	0,7	1,0	0,7	0,6	0,4	0,5	0,8	0,4	0,5
T10	0,3	0,7	0,4	0,4	0,7	0,6	0,4	0,5	0,3	0,3	0,1	0,9	0,8	0,7	0,7	0,7	1,0	0,7	0,8	1,0
T11	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
T12	0,2	0,4	0,4	0,6	0,4	0,4	0,7	1,0	0,2	0,2	0,1	0,4	0,4	0,4	0,5	1,0	1,0	0,7	0,4	0,4
T13	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
T14	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
T15	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

V3 (k=3, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

V3 (k=3, p=2)

Evolution du niveau de maîtrise dans la connaissance (R2)

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
A1	0	0	0	0	0	0	0	0	0	0	0,2	0	0	0	0	0	0	0	0	0,4	0,6	0,6	0,8
A2	0	0,4	0	0	0,4	0	0,4	0,6	0	0	0,2	0	0,4	0	0,4	0	0,4	0,4	0	0,4	0,8	0,6	0,4
A3	0	0,2	0	0	0	0	0,2	0,4	0	0	0,2	0	0,2	0	0,2	0	0	0	0,2	0,4	0,4	0,4	0,2
A4	0	0	0	0	0,4	0,8	1	0,2	0,6	0	0,4	0	0	0,4	0,4	0	0	0	0	0	0,4	0,6	0
A5	0	0	0	0	0,4	0,6	0,8	0,6	0	0	0,4	0	0	0,8	0,6	0	0	0	0	0	0,4	0,6	0
A6	0	0	0	0	0	0,6	0,6	0,4	0	0	0,4	0	0,4	0,6	0,4	0	0	0	0	0	0,4	0,4	0
A7	0	0	0	0	0	0,4	0,6	0,4	0,4	0	0	0,4	0	0,6	0,2	0	0	0	0	0	0,4	0,4	0
A8	0	0	0	0	0	0,4	0,4	0,4	0,4	0	0	0,4	0	0,4	0,4	0	0	0	0	0	0,2	0,2	0
A9	0,8	0,8	0,6	0,6	0,6	0	0	0	0	0	0,4	0	0,8	0	0,8	0,8	0	0	0	0	0,4	0,6	0
A10	0,4	0,6	0,6	0,6	0,4	0,4	0	0	0	0	0,2	0	0,4	0	0,4	0,6	0	0	0	0	0,4	0,4	0
A11	0,4	0,4	0,4	0,4	0,4	0	0	0	0	0	0	0	0,4	0	0,2	0,4	0	0	0	0	0	0,4	0
A12	0	0,8	0,2	0	0,4	0,6	0,8	0,8	0	0	0,4	0	0,6	0	0,8	0	0	0,4	0	0,4	0,6	0,4	0
A13	0	0,6	0,2	0	0,4	0,4	0,6	0,8	0	0,4	0,2	0	0,4	0,6	0,4	0	0	0,4	0	0,2	0,4	0,6	0
A14	0	0,6	0	0	0,2	0,4	0,2	0,6	0	0,4	0	0,2	0,6	0,4	0,6	0	0	0,6	0	0,2	0,4	0,2	0
A15	0	0,4	0	0	0,2	0,2	0,4	0,4	0	0	0	0,4	0,4	0	0,6	0	0	0,2	0,4	0,2	0	0,4	0
A16	0	0	0	0	0	0	0,4	0,8	0,8	0,6	0,4	0,2	0	0	0,8	0	0	0	0	0	0,4	0,6	0
A17	0	0	0	0	0	0	0,4	0,8	0,6	0,6	0	0,4	0,4	0	0,8	0	0	0,4	0	0,4	0,4	0,6	0
A18	0	0,4	0	0	0	0	0,4	0,4	0,4	0,4	0	0,4	0,4	0,2	0,6	0	0	0,2	0	0,4	0,2	0,2	0
A19	0	0,4	0	0	0,4	0	0,4	0,6	0	0	0,4	0	0,6	0	0,4	0	0,4	0,6	0,4	0,4	0,8	1	0,4
A20	0	0,4	0	0	0,4	0	0,4	0,6	0	0	0,4	0	0,4	0	0,4	0	0,4	0,4	0,6	0,6	0,8	0,8	0,2

R2 (k=0)

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
A1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,6	0,9
A2	0,0	0,4	0,0	0,0	0,4	0,0	0,4	0,6	0,0	0,0	0,1	0,0	0,4	0,0	0,4	0,0	0,4	0,4	0,0	0,4	0,8	0,6	0,4
A3	0,0	0,1	0,0	0,0	0,0	0,0	0,1	0,4	0,0	0,0	0,1	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1	0,4	0,4	0,4	0,1
A4	0,0	0,0	0,0	0,0	0,4	0,8	1,0	0,1	0,6	0,0	0,4	0,0	0,0	0,4	0,4	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,0
A5	0,0	0,0	0,0	0,0	0,4	0,6	0,8	0,6	0,0	0,0	0,4	0,0	0,0	0,8	0,6	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,0
A6	0,0	0,0	0,0	0,0	0,0	0,6	0,6	0,4	0,0	0,0	0,4	0,0	0,4	0,6	0,4	0,0	0,0	0,0	0,0	0,0	0,4	0,4	0,0
A7	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,4	0,4	0,0	0,0	0,4	0,0	0,6	0,1	0,0	0,0	0,0	0,0	0,0	0,4	0,4	0,0
A8	0,0	0,0	0,0	0,0	0,0	0,4	0,4	0,4	0,7	0,1	0,0	0,4	0,0	0,4	0,4	0,0	0,0	0,0	0,0	0,0	0,2	0,2	0,0
A9	0,8	0,8	0,6	0,6	0,6	0,0	0,0	0,0	0,0	0,0	0,4	0,0	0,8	0,0	0,8	0,8	0,0	0,0	0,0	0,0	0,4	0,6	0,0
A10	0,4	0,6	0,6	0,6	0,4	0,4	0,0	0,0	0,0	0,0	0,1	0,0	0,4	0,0	0,4	0,6	0,0	0,0	0,0	0,0	0,4	0,4	0,0
A11	0,4	0,4	0,4	0,4	0,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,0	0,1	0,4	0,0	0,0	0,0	0,0	0,4	0,0	0,0
A12	0,0	0,8	0,1	0,0	0,4	0,6	0,8	0,8	0,0	0,0	0,5	0,0	0,6	0,0	0,8	0,0	0,0	0,4	0,0	0,4	0,6	0,4	0,0
A13	0,0	0,6	0,5	0,0	0,4	0,9	0,6	0,8	0,0	0,4	0,5	0,0	0,5	0,6	0,4	0,0	0,0	0,5	0,0	0,1	0,4	0,6	0,0
A14	0,0	0,7	0,0	0,0	0,3	0,4	0,1	0,6	0,0	0,4	0,0	0,1	0,6	0,4	0,6	0,0	0,0	0,6	0,0	0,1	0,4	0,3	0,0
A15	0,0	0,4	0,0	0,0	0,1	0,1	0,4	0,4	0,0	0,0	0,0	0,4	0,4	0,0	0,6	0,0	0,0	0,1	0,4	0,1	0,0	0,4	0,0
A16	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,8	0,8	0,7	0,4	0,1	0,0	0,0	0,8	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,0
A17	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,8	0,6	0,6	0,0	0,4	0,4	0,0	0,8	0,0	0,0	0,4	0,0	0,6	0,4	0,6	0,0
A18	0,0	0,4	0,0	0,0	0,0	0,0	0,5	0,5	0,4	0,8	0,0	0,4	0,4	0,6	0,6	0,0	0,0	0,1	0,0	0,4	0,2	0,2	0,0
A19	0,0	0,4	0,0	0,0	0,4	0,0	0,4	0,6	0,0	0,0	0,4	0,0	0,6	0,0	0,4	0,0	0,4	0,6	0,4	0,5	0,8	1,0	0,4
A20	0,0	0,4	0,0	0,0	0,4	0,0	0,4	0,7	0,0	0,0	0,4	0,0	0,4	0,0	0,4	0,0	0,4	0,5	0,6	0,7	0,9	0,9	0,3

R2 (k=1)

Evolution du niveau de maîtrise dans la connaissance (R2) (suite)

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
A1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,6	0,9
A2	0,0	0,3	0,0	0,0	0,3	0,0	0,3	0,5	0,0	0,0	0,1	0,0	0,3	0,0	0,3	0,0	0,3	0,3	0,0	0,3	0,8	0,5	0,3
A3	0,0	0,1	0,0	0,0	0,0	0,0	0,1	0,3	0,0	0,0	0,1	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1	0,3	0,3	0,3	0,1
A4	0,0	0,0	0,0	0,0	0,3	0,8	1,0	0,1	0,5	0,0	0,3	0,0	0,0	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,0
A5	0,0	0,0	0,0	0,0	0,3	0,5	0,8	0,5	0,0	0,0	0,3	0,0	0,0	0,8	0,5	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,0
A6	0,0	0,0	0,0	0,0	0,0	0,6	0,7	0,4	0,0	0,0	0,5	0,0	0,3	0,6	0,3	0,0	0,0	0,0	0,0	0,0	0,4	0,4	0,0
A7	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,3	0,3	0,0	0,0	0,3	0,0	0,5	0,1	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,0
A8	0,0	0,0	0,0	0,0	0,0	0,4	0,5	0,5	0,9	0,1	0,0	0,5	0,0	0,5	0,5	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,0
A9	0,8	0,8	0,5	0,5	0,5	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,8	0,0	0,8	0,8	0,0	0,0	0,0	0,0	0,3	0,5	0,0
A10	0,3	0,5	0,5	0,5	0,3	0,3	0,0	0,0	0,0	0,0	0,1	0,0	0,3	0,0	0,3	0,5	0,0	0,0	0,0	0,0	0,3	0,3	0,0
A11	0,3	0,3	0,3	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,1	0,3	0,0	0,0	0,0	0,0	0,0	0,3	0,0
A12	0,0	0,8	0,1	0,0	0,7	0,5	0,8	0,8	0,0	0,0	0,5	0,0	0,6	0,0	0,8	0,0	0,0	0,3	0,0	0,3	0,6	0,4	0,0
A13	0,0	0,6	0,5	0,0	0,4	0,9	0,7	0,8	0,0	0,3	0,5	0,0	0,4	0,6	0,3	0,0	0,0	0,5	0,0	0,1	0,4	0,6	0,0
A14	0,0	0,9	0,0	0,0	0,5	0,5	0,1	0,6	0,0	0,3	0,0	0,1	0,7	0,3	0,5	0,0	0,0	0,5	0,0	0,1	0,4	0,4	0,0
A15	0,0	0,3	0,0	0,0	0,1	0,1	0,3	0,3	0,0	0,0	0,0	0,3	0,3	0,0	0,5	0,0	0,0	0,1	0,3	0,1	0,0	0,3	0,0
A16	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,8	0,8	0,9	0,3	0,1	0,0	0,0	0,8	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,0
A17	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,8	0,8	0,6	0,0	0,5	0,3	0,0	0,8	0,0	0,0	0,4	0,0	0,6	0,4	0,6	0,0
A18	0,0	0,3	0,0	0,0	0,1	0,0	0,5	0,5	0,3	0,8	0,3	0,3	0,3	0,6	0,6	0,0	0,0	0,1	0,0	0,3	0,2	0,2	0,0
A19	0,0	0,4	0,0	0,0	0,4	0,0	0,4	0,6	0,0	0,0	0,4	0,0	0,6	0,0	0,4	0,0	0,4	0,6	0,3	0,5	0,8	1,0	0,4
A20	0,0	0,4	0,0	0,0	0,5	0,0	0,6	0,7	0,0	0,0	0,4	0,0	0,4	0,0	0,5	0,0	0,4	0,5	0,6	0,7	0,9	0,9	0,3

R2 (k=2)

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
A1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,5	0,9
A2	0,0	0,3	0,0	0,0	0,3	0,0	0,3	0,5	0,0	0,0	0,0	0,0	0,3	0,0	0,3	0,0	0,3	0,3	0,0	0,3	0,7	0,5	0,3
A3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,3	0,0
A4	0,0	0,0	0,0	0,0	0,3	0,7	1,0	0,0	0,5	0,0	0,3	0,0	0,0	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,0
A5	0,0	0,0	0,0	0,0	0,3	0,5	0,7	0,5	0,0	0,0	0,3	0,0	0,0	0,7	0,5	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,0
A6	0,0	0,0	0,0	0,0	0,0	0,6	0,6	0,4	0,0	0,0	0,4	0,0	0,3	0,6	0,3	0,0	0,0	0,0	0,0	0,0	0,4	0,4	0,0
A7	0,0	0,0	0,0	0,0	0,0	0,3	0,5	0,3	0,3	0,0	0,0	0,3	0,0	0,5	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,0
A8	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,5	0,9	0,0	0,0	0,5	0,0	0,6	0,4	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,0
A9	0,7	0,7	0,5	0,5	0,5	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,7	0,0	0,7	0,7	0,0	0,0	0,0	0,0	0,3	0,5	0,0
A10	0,3	0,5	0,5	0,5	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,3	0,5	0,0	0,0	0,0	0,0	0,3	0,3	0,0
A11	0,3	0,3	0,3	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,3	0,0
A12	0,0	0,8	0,0	0,0	0,6	0,5	0,8	0,8	0,0	0,0	0,5	0,0	0,5	0,0	0,8	0,0	0,0	0,3	0,0	0,3	0,6	0,4	0,0
A13	0,0	0,6	0,5	0,0	0,4	0,9	0,6	0,8	0,0	0,3	0,5	0,0	0,4	0,5	0,3	0,0	0,0	0,5	0,0	0,0	0,4	0,6	0,0
A14	0,0	0,9	0,0	0,0	0,5	0,5	0,0	0,6	0,0	0,3	0,0	0,0	0,7	0,3	0,5	0,0	0,0	0,5	0,0	0,0	0,4	0,3	0,0
A15	0,0	0,3	0,0	0,0	0,0	0,0	0,3	0,3	0,0	0,0	0,0	0,3	0,3	0,0	0,5	0,0	0,0	0,0	0,3	0,0	0,0	0,3	0,0
A16	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,8	0,8	0,9	0,3	0,0	0,0	0,0	0,8	0,0	0,0	0,0	0,0	0,0	0,4	0,6	0,0
A17	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,8	0,8	0,5	0,0	0,4	0,3	0,0	0,8	0,0	0,0	0,4	0,0	0,6	0,4	0,6	0,0
A18	0,0	0,3	0,0	0,0	0,0	0,0	0,6	0,6	0,4	0,8	0,4	0,3	0,3	0,6	0,6	0,0	0,0	0,0	0,0	0,3	0,3	0,3	0,0
A19	0,0	0,3	0,0	0,0	0,3	0,0	0,3	0,6	0,0	0,0	0,3	0,0	0,5	0,0	0,3	0,0	0,3	0,5	0,3	0,4	0,8	1,0	0,3
A20	0,0	0,4	0,0	0,0	0,6	0,0	0,7	0,7	0,0	0,0	0,5	0,0	0,4	0,0	0,6	0,0	0,3	0,4	0,6	0,7	0,9	0,9	0,3

R2 (k=3)

8.4 Annexe 4 : Constitution d'équipes

La programmation du recuit simulé et la programmation quadratique pour résoudre le problème de constitution d'équipes sont présentées ci-après, dans deux fichiers MATLAB :

- recuit_MO.m
- pb_quadratique.m

Fichier : recuit_MO.m

```
Function[Iopt,Dopt,cout_tt,cout_total,cout_sa,cout_salaire,cout_tut,cout_tuteur,cout_pns,cout_penalisant,cout_gar,cout_garantie,cout_DSM,cout_DSM_A]=recuit_MO(I0,arret_simul,RQ,L,LM,task_disp,C,Rinit)

%% Initialisation des paramètres
unsurT=0;
nbehec=0;
ordre=I0;
load W
[x,y,z,w]=f(ordre,RQ,L,LM,task_disp,Rinit);
ORDRE=I_integree(L,ordre);
[D,q1,q2]=pb_quadratique_1(ORDRE,C);
c11=W1*y;c12=W1*z;c13=W2*w;c14=W3*q1;c15=W3*q2;
f1=c11+c12+c13+c14+c15;
fopt=f1;
Iopt=ordre;
Dopt=D; % Dopt sauvegarde des meilleures équipes
cb1=c11;cb2=c12;cb3=c13;cb4=c14;cb5=c15;

%% Programme principal
for stop = 0 : arret_simul
    unsurT = unsurT+0.1;
    Palier=1000;
    n=0;
    while (n<Palier)
        [ordre2,nbehec]=voisin(ordre,RQ,L,LM,task_disp);
        [x,y,z,w]=f(ordre2,RQ,L,LM,task_disp,Rinit);
        ORDRE2=I_integree(L,ordre2);
        [D2,q1,q2]=pb_quadratique_1(ORDRE2,C);
        c21=W1*y;c22=W1*z;c23=W2*w;c24=W3*q1;c25=W3*q2;
        f2=c21+c22+c23+c24+c25;
        if (f2<=f1)
            ordre=ordre2;
            D=D2;
            f1=f2;
            c11=c21;c12=c22;c13=c23;c14=c24;c15=c25;
            if (f1<=fopt)
                Iopt=ordre;
                Dopt=D;
                fopt=f1;
                cb1=c11;cb2=c12;cb3=c13;cb4=c14;cb5=c15;
            end
        else
            f1_f2=(c11-c21)+(c12-c22)+(c13-c23)+(c14-c24)+(c15-c25);
            proba=exp(-(f1_f2)*unsurT);
            random=rand();
            if(random<proba)
                ordre=ordre2;
                D=D2;
                f1=f2;
            end
        end
        n=n+1;
        disp(stop)
        disp(n)
        cout_total(stop+1,n)=fopt;
        cout_salaire(stop+1,n)=cb1;
        cout_tuteur(stop+1,n)=cb2;
        cout_penalisant(stop+1,n)=cb3;
        cout_garantie(stop+1,n)=cb4;
        cout_DSM_A(stop+1,n)=cb5;
        save('couts','cout_salaire',
'cout_tuteur','cout_total','cout_penalisant','cout_garantie','cout_DSM_A')
    end
    stop=stop+1;
end
```

Fichier : pb_quadratique.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               écriture quadratique                               %
%   min  sum((IDI')/beta-C)^2 +(lambda/beta^2)*sum(D(i,j))                       %
%   sous les contraintes IDI'>=beta*C, D(i,i)=0, 0<=D(i,j)<=1                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [cout_total,cout_fidelite,cout_interaction,Dopt]=pb_quadratique(I,C)
beta=0.6;
% initialiser des paramètres
m=size(I);
n=m(1);
d=m(2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****                               création du vecteur b ( =C(i,j))           %*****
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b=zeros(n^2,1);
for j=1:n
    for i=1:n
        b(i+(j-1)*n)=C(i,j);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****                               construction de la matrice de contrainte A   %*****
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=zeros(n^2,d^2);
I1=I';
for j=1:n
    for i=1:n
        A1=zeros(d,d);
        for k=1:d
            for l=1:d
                A1(l,k)=(I(i,k))*(I1(l,j));
            end
        end
%concaténation
        A11=zeros(1,d^2);
        for p=1:d
            for q=1:d
                A11(q+(p-1)*d)=A1(q,p);
            end
        end
        A(i+(j-1)*n,:)=A11;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****                               construction de la contrainte D(i,i)=0     %*****
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Aeq=zeros(d,d^2);
for k=1:d
    Aeq(k,(k+(k-1)*d))=1;
end
beq=zeros(d,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****                               création des contraintes 0<=D(i,j)<=1       %*****
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lb=zeros(d^2,1);
ub=ones(d^2,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****                               création de la contrainte Dopt(i,j)=Dopt(j,i) %*****
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P2=eye(d^2,d^2);
for j=1:d-1
    for i=j+1:d
        P2(i+(j-1)*d,j+(i-1)*d)=-1;
    end
end
for i=1:d^2
    c=0;
    for j=1:d^2
        c=c+P2(i,j);
    end
    if c==1;
        P2(i,:)=0;
    end
end
end

```

```

%*****
%permettre de supprimer la redondance des contraintes
for i=d^2:-1:1
    if P2(i,:)==0
        P2(i,:)=[];
    end
end
%*****
%création du vecteur qui l'accompagne
s=size(P2);
p2=zeros(s(1),1);
Aeq=[Aeq;p2];
beq=[beq;p2];
%*****
%***** définition de la fonction objectif *****
%***** la fonction quadprog donne *****
%***** min (1/2 X'HX +f'X) *****
%*****
e=ones(1,d^2);
lambda_1=1;
lambda_2=lambda_1/beta^2;
H=lambda_1*2*(A'*A)/beta^2;
f=(lambda_1*(-2)*b'*(A/beta)) + (lambda_2*e);
f=f';
%*****
%*** utilisation de la fonction quadprog pour trouver l'optimum *****
%*****
opts = optimset('LargeScale','off','MaxIter',1500);
[Xopt,cout,exitflag,output] = quadprog(H,f,-A/beta,-b,Aeq,beq,lb,ub,[],opts);
disp(exitflag)
disp(output)
%*****
%***** création de la matrice Dopt *****
%*****
Dopt=zeros(d,d);
for j=1:d
    for i=1:d
        Dopt(i,j)=Xopt(i+(j-1)*d);
    end
end
%*****
%***** calcul du coût de fidélité *****
%*****
diff=(I*Dopt*I')/beta-C;
diff=diff.*diff;
cout_fidelite=0;
for j=1:n
    for i=1:n
        cout_fidelite=cout_fidelite+diff(i,j);
    end
end
cout_fidelite=lambda_1*cout_fidelite;
%*****
%***** calcul du cout total *****
%*****
cout_total=cout+lambda_1*(b'*b);
%*****
%***** calcul du cout d'interaction *****
%*****
v=0;
for j=1:d
    for i=1:d
        v=v+Dopt(i,j);
    end
end
cout_interaction=lambda_2*v;

```

Matrices d'affectation finales (chapitre 5)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,7
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=1, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,9	0,1	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0,2	0,8	0	0	0	0	0	0	0
T9	0	0	0	0	0	0,8	0,2	0,0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0	0,9	0	0	0

MI opt (k=1, p=2)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0,5	0	0	0	0	0	0	0,5	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0,8	0	0	0	0	0	0	0	0,2
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
T9	0	0	0	0	0	0,7	0	0,3	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0,2	0,8	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0,2	0	0	0	0	0	0	0	0,4	0,4	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=2, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0,4	0	0	0	0	0	0,6	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

MI opt (k=2, p=2)

Matrices d'affectation finales (chapitre 5-suite)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0,2	0	0	0	0	0	0	0,8
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=3, p=1)

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MI opt (k=3, p=2)

8.5 Annexe 5 : Fonction de régression des compétences

Concernant l'évolution des compétences, nous proposons deux lois d'évolution : loi d'augmentation et loi de régression. La loi d'augmentation a été présentée dans le chapitre 4 (§4.3.4), cette loi a été définie en fonction des décisions d'affectation.

Cette annexe a pour but de proposer une courbe de régression permettant d'estimer la diminution du niveau de maîtrise dans la connaissance acquis par l'acteur ($r_{c,j}^k$).

8.5.1 Hypothèses

Les hypothèses utilisées dans cette courbe de régression sont les suivantes :

- Le niveau de maîtrise dans la connaissance diminue si l'acteur n'est pas alloué aux tâches associées à cette connaissance. La diminution du niveau de maîtrise dans les connaissances dépend du nombre de périodes durant lesquelles l'acteur n'est pas alloué à cette connaissance.
- Plus longtemps une personne étudie une connaissance, plus cette personne peut retenir ce qu'elle a appris.
- La courbe de régression est la même pour toutes les natures des connaissances.
- Pour faciliter le choix de la courbe de régression, nous supposons que la courbe de régression peut être définie à partir de la courbe d'apprentissage. C'est-à-dire, si la courbe d'apprentissage est présentée par une forme (X), nous supposons que la courbe de régression peut être obtenue en utilisant une "forme inverse" (1-X) de cette courbe d'apprentissage.

Sachant que la démarche que nous avons choisie pour définir la courbe de régression des compétences est basée sur l'expérimentation des courbes d'apprentissage (ou courbe d'augmentation) existantes dans la littérature, nous allons donc dans un premier temps présenter ces courbes d'apprentissages (en section 2). Puis, nous analysons ces courbes à travers des expérimentations (en section 3). Enfin, nous proposons une courbe de régression et le réglage de ses paramètres (en section 4).

8.5.2 Introduction de la courbe d'apprentissage

La théorie de l'apprentissage, appliquée à la firme signifie qu'avec la répétition de tâches identiques, le personnel d'une entreprise devient de plus en plus expérimenté et permet à celle-ci de développer des gains de productivité. Les observations faites dans l'aéronautique américaine dans les années 30 par l'officier Wright ont montré qu'avec le temps et notamment en fonction de la production cumulée d'avions, le nombre d'heures de travail nécessaires pour chaque nouvelle unité produite décroissait régulièrement : à chaque doublement de la production cumulée correspondait une économie de temps de 20%. Dans les "courbes d'expérience", la baisse des coûts unitaires de production n'est pas expliquée par le seul effet d'apprentissage direct, elle dépend aussi des économies d'échelles et de l'introduction de l'innovation.

Dans les activités où le volume de production croît rapidement, la courbe d'expérience est un outil d'analyse stratégique pertinent : il s'agit de descendre le plus vite possible le long de

la courbe d'expérience pour avoir les coûts unitaires de production les plus bas possibles. Il est vrai que la diminution des coûts en raison de l'effet d'expérience constitue une barrière à l'entrée de nouveaux producteurs qui doivent, dans un premier temps, accepter des coûts plus élevés et donc une rentabilité moindre.

Le terme "courbe d'apprentissage" est proche du terme "courbe d'expérience", ces deux courbes expriment la relation entre l'expérience et l'efficacité. Quand des individus et/ou des organisations acquièrent plus d'expérience sur une tâche, ils deviennent plus efficaces pour réaliser cette tâche.

Le processus d'apprentissage et le processus de régression de compétence sont liés aux décisions d'affectation des tâches et ont souvent été étudiés dans le domaine de production [Nembhard et Uzumeri, 2000], [Nembhard, 2001], dans la conception de logiciel [Boh *et al.*, 2007], ou encore dans le management de projet [Plaza et Rohlf, 2008], [Plaza *et al.*, 2009]. Ces travaux ont pour but d'aider l'ordonnancement à définir les temps standards, à estimer les coûts opérationnels ou à optimiser les programmes d'apprentissage.

On peut distinguer deux types d'apprentissage : au niveau de l'organisation et au niveau des individus. L'apprentissage au niveau de l'organisation est étudié au niveau des unités organisationnelles de l'entreprise. L'apprentissage des individus concerne l'étude de la courbe d'apprentissage pour comprendre ou estimer les mécanismes de chaque individu.

Ci-après, nous présentons deux courbes d'apprentissage : l'une proposée par Wright (1936) et l'autre proposée par Nembhard et Uzumeri, (2000).

8.5.2.1 Courbe d'expérience de Wright

Wright (1936) a étudié le phénomène de l'apprentissage des individus dans la production. Il propose une courbe logarithmique qui s'explique par le fait que l'employé va utiliser moins de temps pour la tâche suivante (répétition). Wright fait l'hypothèse que le temps de production se réduit d'un pourcentage constant (ex. 90%, 80%, etc.) à chaque fois que la quantité produite cumulée est doublée. Wright propose alors le modèle suivant :

$$Y = aX^b$$

Équation 1

Où Y est le temps moyen cumulé (ou coût) par unité de production, X est le nombre d'unités produites cumulées, a est le temps (coût) requis pour produire la première unité, b est la pente de la fonction quand la courbe est tracée dans un repère log-log. La valeur b est le logarithme du taux d'apprentissage.

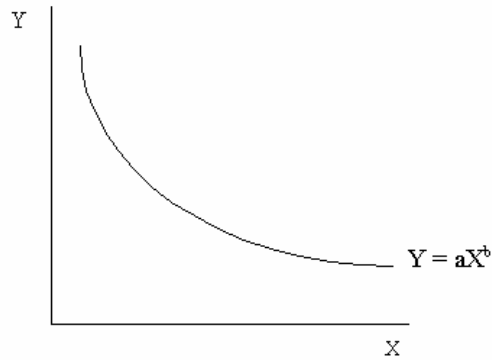


Figure 6: Courbe d'apprentissage de (Wright, 1936)

8.5.2.2 Modèle de Nembhard et Uzumeri

A partir des données dans la gestion des moyens de production, [Nembhard et Uzumeri, 2000] proposent un modèle d'apprentissage et de décroissance de compétence dans le but de mesurer la productivité et le taux d'apprentissage des employées. Les facteurs pris en compte dans leurs expérimentations sont les types d'employés (nouveau embauche, apprenti), les types de tâches (manuelle ou cognitive). On montre qu'il existe une similarité entre les deux types de tâches. En ce qui concerne la corrélation entre le taux d'apprentissage (apprenti rapide ou apprenti lent) et le taux de dépréciation, les employés qui apprennent plus rapidement ont une tendance à oublier plus facilement pendant la période d'interruption. Il a été prouvé que cet effet se produit à la fois dans les tâches manuelles et dans les tâches cognitives.

Basé sur la courbe parabolique, [Nembhard et Uzumeri, 2000] proposent un modèle d'apprentissage et de décroissance de compétence. Leur modèle d'apprentissage est utilisé pour mesurer la capacité d'apprentissage des individus en prenant en compte leurs niveaux d'expériences anciennes. La valeur y_x est un taux de productivité correspondant à l'unité cumulative du travail (x). Le paramètre k est utilisé pour estimer le taux de productivité (en unité du temps) à l'état stationnaire dans la courbe asymptotique. Le paramètre r est une quantité cumulative de production x pour atteindre la moitié de la valeur k . La valeur k est estimée par rapport à la réalisation des tâches. Le paramètre p représente les anciennes expériences du personnel consacrées à une tâche. Ce paramètre est estimé par des experts et des travaux effectués avant. La forme de la courbe d'apprentissage individuelle est déterminée par le paramètre r . La valeur r est le nombre nécessaire d'unités cumulatives du travail pour atteindre le taux de productivité de $k/2$. La courbe individuelle part de zéro unité cumulative du travail ($x=0$).

$$y_x = k \left(\frac{x+p}{x+p+r} \right)$$

Équation 2

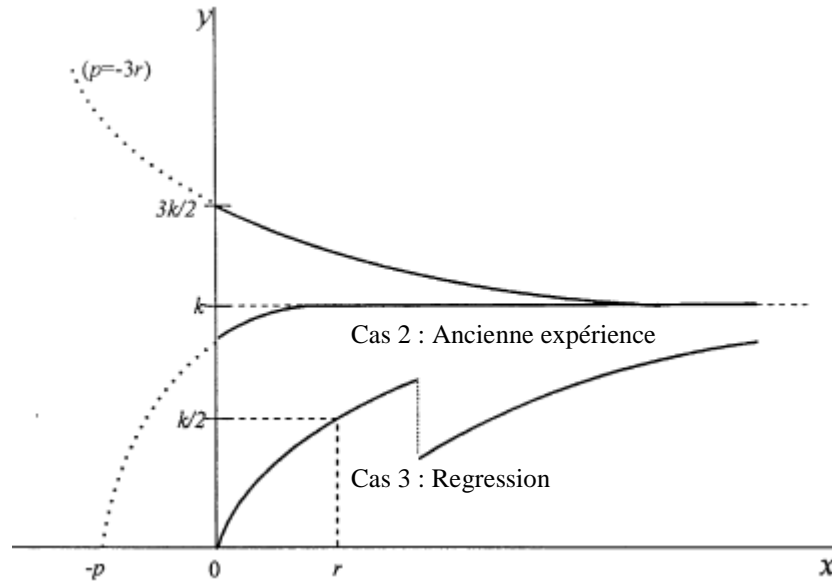


Figure 7: Modèle d'apprentissage [Nembhard et Uzumeri, 2000]

8.5.3 Expérimentation de la courbe de régression

Nous procédons à deux expérimentations : premier essai avec la courbe de [Nembhard et Uzumeri, 2000] et deuxième essai avec la courbe de Wright (1936).

8.5.3.1 Premier essai avec la courbe de Nembhard et Uzumeri

En générale, la courbe d'apprentissage de [Nembhard et Uzumeri, 2000] commence en 0 et est croissante presque à l'asymptote. Nous testons une courbe de régression utilisant une "forme inverse" de la courbe d'apprentissage de [Nembhard et Uzumeri, 2000].

Cette forme inverse est présentée dans l'équation 3. La courbe part du niveau maximal de performance dans la connaissance 1 et décroît jusqu'à 0.

$$y = 1 - k \left(\frac{x}{x+r} \right)$$

Équation 3

- La variable x représente le nombre de périodes pendant lesquelles un acteur n'est pas affecté à une tâche.
- La variable y représente le niveau de maîtrise dans la connaissance.
- La valeur Δx représente le nombre de périodes pendant lesquelles le niveau de maîtrise dans la connaissance diminue de niveau y_0 à y_1 .
- La valeur Δy représente le niveau de régression du niveau de maîtrise dans la connaissance pendant le temps (Δx).
- La valeur r est la durée non affectée (Δx) qui fait descendre le niveau de maîtrise dans la connaissance diminue de 1 à 0.5.

- La valeur k est la valeur asymptote, $k=1$.

Nous avons lancé une simulation de 1000 périodes consécutives pendant lesquelles un acteur n'est pas affecté à une tâche et tracé cette courbe de régression (Figure 8 (a)). La Figure 8 (b) est un *zoom* de la courbe de régression précédente. Elle met en évidence la valeur r (où le niveau de maîtrise dans la connaissance diminue à la moitié). Nous avons constaté que si le niveau de maîtrise dans la connaissance a diminué à la moitié (0.5), le nombre de périodes consécutives non affectées est égale à r , ($x=r=24$).

Cette courbe implique que le niveau de maîtrise dans la connaissance diminue très vite au début et diminue lentement à la fin. Par exemple, pour diminuer de 1 à 0.5, il faut passer 24 périodes de non affectation. Quant à la diminution de 0.5 à 0.1, il faut passer 192 périodes.

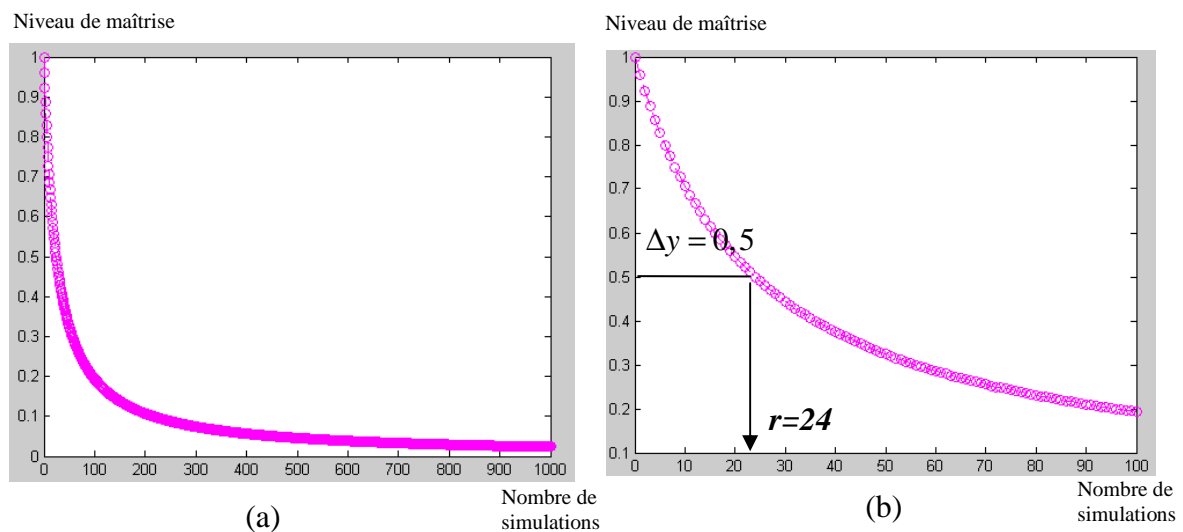


Figure 8 : Courbe de régression de connaissance, $k=1$, $r=24$

Une traduction possible de cette courbe est que les personnes qui possèdent un haut niveau de maîtrise perdent la connaissance plus vite que les personnes qui possèdent un bas niveau de maîtrise. Cette caractéristique ne correspond pas à la dégradation du niveau des compétences des concepteurs. Puisque, d'une façon générale dans le domaine de la conception, les experts (possèdent le niveau de compétence >0.8 par exemple) perdent la connaissance moins vite que les débutants. Nous n'avons pas choisi cette courbe pour représenter la courbe de régression cherchée.

8.5.3.2 Deuxième essai avec la courbe de régression inspirée par Wright, (1936)

Les résultats obtenus (ci-dessus) nous ont incités à chercher une expression permettant de changer "la concavité" de la courbe de régression. Nous avons adapté la courbe d'apprentissage de Wright, (1936) qui est basée sur une courbe exponentielle ($y = kx^b$). En adaptant la courbe de Wright en forme "inverse", la fonction de régression retenue est présentée comme suit.

$$y = 1 - kx^b$$

Équation 4

Notations :

- La valeur x représente le nombre de périodes.
- La valeur y représente le niveau de maîtrise dans la connaissance.
- La valeur Δx représente le nombre de périodes pendant lesquelles le niveau de maîtrise dans la connaissance diminue de y_0 à y_1 .
- La valeur Δy représente le niveau de régression du niveau de maîtrise dans la connaissance pendant une durée non affectée (Δx).
- La valeur r est la durée non affectée de Δx qui fait descendre le niveau de maîtrise dans la connaissance diminue de 1 à 0.5.
- La valeur b est une valeur constante de régression
- La valeur k est une valeur constante de la courbe

La courbe de régression obtenue (Figure 9) paraît satisfaisante. Nous observons "la concavité" de la courbe de régression. Cette concavité permet au niveau de maîtrise de la connaissance de diminuer très peu au début et de diminuer plus rapidement à la fin. Ainsi, la connaissance des experts diminue plus lentement que la connaissance des débutants.

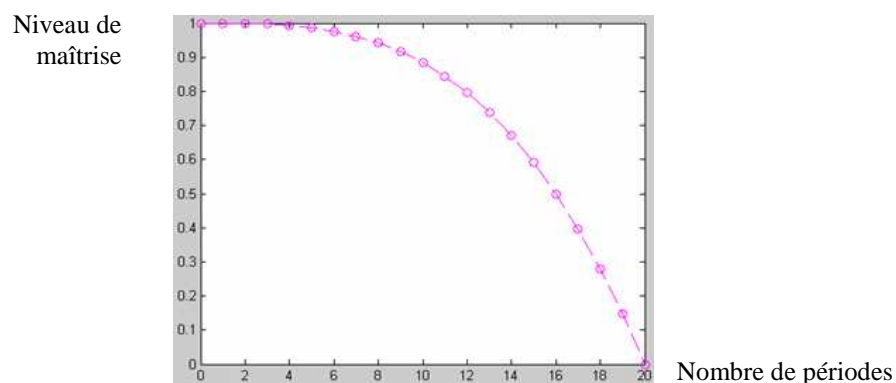


Figure 9 : Courbe de régression inspirée par Wright (1936)

8.5.3.3 Conclusion

Dans un premier temps, nous avons réalisé des essais avec la courbe proposée par [Nembhard et Uzumeri, 2000]. Nous n'avons pas choisi cette courbe parce que, selon cette courbe, les experts perdent la connaissance plus vite que les débutants. Cela ne reflète pas la nature souhaitée de l'apprentissage des concepteurs.

Dans un second temps, nous avons testé une nouvelle courbe de régression inspirée de la courbe de Wright, (1936). Cette courbe présente des caractéristiques satisfaisantes. Elle montre que les experts perdent la connaissance moins vite que les débutants. Autrement dit, plus longtemps un individu est affecté à un sujet, plus il retient ce qu'il a appris. Nous avons adopté et utilisé cette courbe dans la simulation de la régression de compétence.

8.5.4 Proposition et paramétrage de la courbe de régression proposée

Le but de cette partie est de tracer la courbe de régression qui sera utilisée dans l'évolution des compétences et de définir la fonction de régression en termes du niveau de maîtrise dans la connaissance $r_{c,j}^k$.

8.5.4.1 Tracer la courbe de régression

Notre courbe de régression, inspirée par la courbe de Wright, est présentée par l'équation ci-dessous.

$$y = \begin{cases} 1 - kx^b & , x \leq R \\ 0 & , x > R \end{cases}$$

Équation 5

Certaines coordonnées sont à considérer pour tracer cette courbe.

- La coordonnée $(y = 1 ; x = 0)$ est traduite par le fait que l'expert possède de niveau 1 à la période initiale.
- La coordonnée $(y = 0 ; x = R)$ signifie que le niveau de maîtrise dans la connaissance de l'expert va diminuer de 1 jusqu'à 0 lorsque cet expert n'a pas été affecté pendant R périodes.
- La coordonnée $(y = 0,5 ; x = r)$ signifie qu'il faut r périodes non affectées pour qu'un expert (niveau de maîtrise 1) ait le niveau divisé par 2.

8.5.4.1.a Fixer les paramètres b et k

Pour fixer les paramètres b et k , nous définissons les paramètres suivants :

- une période de temps dans la simulation dure 3 mois.
- $R = 20$; le concepteur perd tout le niveau de maîtrise dans la connaissance ($y=0$) s'il n'a pas été affecté pendant 20 périodes consécutives. Cette durée est équivalente à 5 ans ($20 \times 3/12$).
- $r = 16$; le niveau de maîtrise de l'expert diminue de moitié s'il n'a pas été affecté pendant 16 périodes consécutives (soit 4 ans).

On remarque que le choix des valeurs R et r est délicat et dépend du type de compétence. Le réglage de ces paramètres nécessite l'expérience des experts.

Pour fixer les constantes b et k , nous avons considéré deux points de repère :

Au premier point $(y = 0,1 ; x = 20)$, nous avons obtenu

$$0,1 = 1 - k(20)^b$$

Au deuxième point $(y = 0,5 ; x = 16)$, nous avons obtenu

$$0,5 = 1 - k(16)^b$$

Avec les deux équations, les constantes b et k seront calculées.

$$b = \frac{\log(1/0,5)}{\log(20/16)} = 3,1$$

$$k = 1 / (20)^b = 0,0001$$

8.5.4.1.b Tracer la courbe de régression

En fixant ($R = 20$) et ($r = 16$), nous avons obtenu les valeurs b et k ($b=3,1$, $k=0,0001$). Nous pouvons tracer la courbe dans la figure ci-dessous.

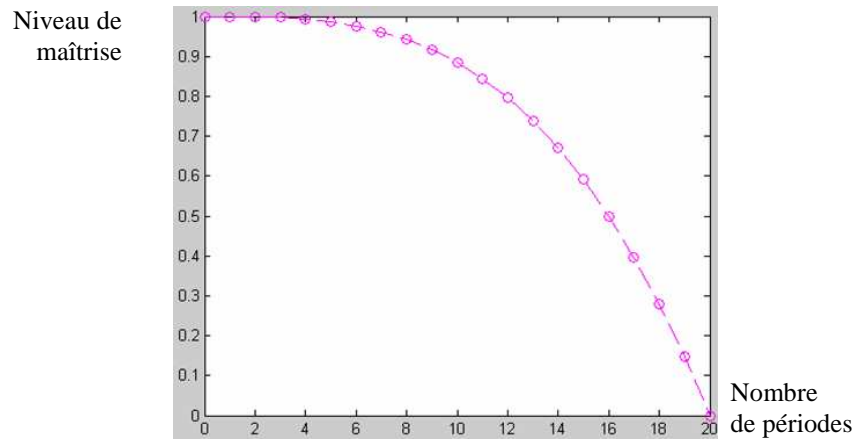


Figure 10 : Courbe de régression proposée

8.5.4.2 Fonction de régression en termes de $r_{2,c,j}^k$ et $r_{2,c,j}^{k-1}$

Notons que :

- y_1 représente le niveau de maîtrise dans la connaissance de la période actuelle.
- y_0 est le niveau de maîtrise dans la connaissance de la période précédente.
- x_1 représente la période actuelle
- x_0 est la période précédente.

A partir de la courbe retenue, nous avons défini une fonction (Équation 6) permettant de calculer le nouveau niveau de maîtrise acquis dans la connaissance (y_1) pendant une seule période non affectée ($\Delta x = x_1 - x_0 = 1$). Cette fonction s'écrit comme suit.

$$y_1 = f_{régression}(y_0)$$

Équation 6

Pour trouver un nouveau niveau de maîtrise dans la connaissance (y_1) pendant une seule période non affectée ($\Delta x = x_1 - x_0 = 1$), nous détaillons les étapes suivantes:

- 1) Chercher la valeur x_0

Ayant la valeur y_0 comme une donnée d'entrée, la valeur x_0 peut être obtenue à partir de l'équation $y = 1 - kx^b$ ou encore.

$$x_0 = 10^{\{(1/b) \times [(\log(1-y_0)) - \log(k)]\}}$$

2) Ajouter une période supplémentaire à x_0 ,

$$x_1 = x_0 + 1$$

$$x_1 = \left[10^{\{(1/b) \times [(\log(1-y_0)) - \log(k)]\}} + 1 \right]$$

3) Nous avons trouvé la fonction entre y_1 et y_0 que nous appelons $f_{régression}$.

$$y_1 = 1 - kx_1^b$$

$$y_1 = 1 - k \left\{ \left[10^{\{(1/b) \times [(\log(1-y_0)) - \log(k)]\}} + 1 \right]^b \right\}$$

$$y_1 = f_{régression}(y_0)$$

Nous pouvons écrire cette équation en fonction des niveaux de maîtrise dans la connaissance de la période actuelle ($r_{c,j}^k$) et de la période précédente ($r_{c,j}^{k-1}$).

$$\begin{aligned} r_{c,j}^k &= f_{régression}(r_{c,j}^{k-1}) \\ &= 1 - k \left\{ \left[10^{\{(1/b) \times [(\log(1-r_{c,j}^{k-1})) - \log(k)]\}} + 1 \right]^b \right\} \end{aligned}$$

Équation 7

8.6 Annexe 6 : Convexité du problème quadratique

Le problème à considérer est le suivant :

$$\left\| \frac{DI'}{\beta} - C \right\|_2^2 = (AX - c)^2 = X' A' A X - 2cA' X + c' c = X' H X - 2cA' X + c' c$$

Un problème quadratique classique de la forme $x'Hx + f'x$ admet une solution si la matrice H est définie positive ou nulle.

Ici, nous avons $H=A'A$, donc H est bien une matrice définie positive et donc il existe au moins une solution à ce problème.

De plus, la solution est unique si H est définie positive stricte, ce que nous allons démontrer.

Soit x une solution minimisant le problème

On a donc $f(y) \geq f(x) \forall y \in S$ (espace solution convexe)

Supposons qu'il existe $y \in S$ tel que $f(y) = f(x)$

Posons $z_t = x + t(y - x) \forall t \in [0,1]$. Par convexité de S , $z_t \in S$

On a donc $f(x) \leq f(z_t) \leq tf(y) + (1-t)f(x) = f(x)$

Donc s'il existe une solution y telle que $f(y)=f(x)$, alors pour tout $t \in [0,1]$ $z_t = x + t(y - x)$ est également solution.

Nous venons de montrer que $f(z_t)$ était constant en t , donc si on dérive f en z_t dans la direction $(y-x)$, cette dérivée doit être nulle. On a donc :

$$\begin{aligned} f'(z_t; y-x) = 0 &\Leftrightarrow \langle f'(z_t, A'Az_t - 2A'c, y-x), y-x \rangle \\ &\Leftrightarrow \langle A'Az_t - 2A'c, y-x \rangle \\ &\Leftrightarrow (y-x)' (A'Az_t - A'c_2) \text{ avec } c_2 = 2c \end{aligned}$$

Maintenant remplaçons z_t par sa valeur, c'est-à-dire par $x + t(y - x)$.

On obtient:

$$\underbrace{t(y-x)' A' A (y-x)}_{\text{non constant (dépend de t)}} + \underbrace{(y-x)' (A' A x - A' c_2)}_{\text{constant (ne dépend pas de t)}} = 0$$

La seule façon pour que cette équation soit vérifiée est d'avoir $(y-x)=0$, c'est-à-dire $(y-x) \in \text{Ker}(H)$.

Donc H est définie positive stricte,

Donc la solution de ce problème est unique.

Contribution à la constitution d'équipes de conception couplant la structuration du projet et le pilotage des compétences

Résumé

Lors de l'engagement d'un projet, un chef de projet doit structurer le projet de conception. Il s'agit de définir une architecture préliminaire du produit, de structurer le processus de conception et de regrouper les acteurs du projet en équipes interdépendantes (le regroupement est appelé "architecture des acteurs"). Ces trois activités sont fortement inter-reliées et critiques lors de la phase de conception préliminaire, car les décisions prises auront un impact fort sur les performances de l'ensemble du projet. Cependant, peu de méthodes formelles ont été proposées pour les considérer conjointement. Par ailleurs, lors de l'affectation des tâches du processus aux acteurs, le chef de projet doit s'assurer que les acteurs sont compétents pour réaliser une tâche. Il doit aussi prendre en compte des besoins de futurs projets qui pourraient nécessiter des apprentissages dans les projets en cours. Dans la réalité, la complexité de l'organisation d'un projet empêche d'avoir une vision globale des conséquences des choix d'affectation, qui plus est, sur plusieurs périodes et sur plusieurs projets. Peu de travaux se sont intéressés à ce type de problème. Or aujourd'hui, les tâches et les compétences nécessaires dans les projets de conception évoluent rapidement en lien avec la forte évolution des technologies et des besoins du marché. Le développement des compétences des ressources humaines devient une des priorités de l'entreprise qui ne les considère plus comme des ressources statiques (comme des machines) mais comme des ressources dynamiques auxquelles il faut attacher des objectifs de développement.

Dans ce mémoire, nous proposons une méthode d'aide à la constitution d'équipes permettant de coupler la structuration du projet et le pilotage des compétences. Ce travail consiste alors à optimiser l'affectation des tâches aux acteurs et le regroupement des acteurs en équipes, en tenant compte à la fois des contraintes liés à l'architecture du produit et des objectifs de développement de compétences, sur de multiples projets et de multiples périodes. Ce problème étant complexe, ce mémoire de thèse présente des contributions à des sous-problèmes (affectation mon-période, affectation multi-périodes avec prise en compte du développement des compétences, architecture des acteurs avec prise en compte de l'architecture du produit), avant de proposer une formulation du problème global et une méthode de résolution en deux phases intégrant un algorithme de recuit simulé.

Mots clés – constitution d'équipes, structuration du projet, structure organisationnelle, pilotage des compétences, développement des compétences, affectation des tâches, allocation de ressources

Contribution to the design team building coupling the project organization and the competency development

Abstract

At the moment of the commitment of a project, project managers have to organize their design projects. For doing this, they have to define a preliminary product architecture, to organize design processes and to group the assigned actors into independent teams. These three activities are strongly related and critical for the entire project. However, few formal methods have been proposed for considering these activities simultaneously. In addition, during the task assignment, project managers have to ensure that the actors are competent enough for doing the tasks. They have to consider the future needs in term of competency development, which could be obtained by training the actors during the ongoing projects. In reality, the complexity of the project organization can prevent the managers from having the global vision of the consequences of their assignment decisions, which are sometimes made related to many periods and many projects. Furthermore nowadays, the competencies that are necessary in the design project rapidly evolve due to quick changes in technologies and customers' needs. The development of competencies of human resources becomes one of the key insights of the firms. They don't consider these competencies as being static but dynamic resources. These resources have to reach specific goals of competency development.

In this thesis, we propose a method to build project teams. This work consists in optimizing the "tasks-human resources" assignment decisions and the project organization, coupling the constraints of product architecture and goals of competency development. This problem is formulated for complex design situations with multiple design projects and multiple periods. Being complex, this work presents contributions in several sub-problems (mono-period assignment, multi-period assignment with the competency development, organizational architecture of the problem by taking the product architecture into account), before proposing a formulation of the global problem. This problem is solved by integrating an algorithm of simulated annealing.

Keywords – team building, project organization, organizational architecture, organization structure, task assignment, competency management, competency development, resource allocation