

**Faculté des Sciences Exactes & de l'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique**

**MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : Ingénierie des Systèmes d'Information**

**Implémentation d'une méthode d'optimisation des coûts
dans une chaîne logistique.**

Présenté par :

- **MEDJAHED Belkacem**
- **HADRA Djamaia**

Encadré par:

- **Mme AIT SI LARBI OULD BRAHIM E.**

Remerciements

A nos chers parents,

Que nulle dédicace ne puisse exprimer ce que nous leurs devons, pour leur bienveillance, leur affection et leur soutien... Trésors de bonté, de générosité et de tendresse, en témoignage de notre profond amour et notre

grande reconnaissance

« Que Dieu vous garde ».

A notre encadreur,

Toute notre gratitude est adressée à notre encadreur Mme Ait Si Larbi Ould Brahim Et Fasmine qui nous a soutenus, et encouragés par des orientations et conseils riches de bon sens.

A nos chers frères et sœurs,

En témoignage de nos sincères reconnaissances pour les efforts qu'ils ont consenti pour l'accomplissement de nos études.

Nous leurs dédions ce modeste travail en témoignage de notre grand amour et notre gratitude infinie

Sommaire

Introduction générale.....	5
Chapitre I : Les chaînes logistiques.....	6
Introduction	7
1. La chaîne logistique	7
1.1 Les flux de la chaîne logistique	7
1.1.1 Le flux physique :.....	7
1.1.2 Le flux de données :	7
1.1.3 Le flux financier :	8
1.2 Les acteurs de la chaîne logistique	8
1.3 Les processus	8
2. Gestion de chaînes logistiques (Supply Chain Management).....	9
2.1 Définitions	9
3. Modélisation décentralisée de chaîne d'approvisionnements.....	10
3.1 La modélisation des problèmes des chaînes logistiques.....	10
4. Gestion distribuée de chaîne d'approvisionnements.....	11
4.1 Optimisation du réseau logistique	11
4.2 Définition générale de l'optimisation distribuée.....	11
4.3 Utilisation de l'optimisation distribuée dans SCM.....	13
Conclusion.....	14
Chapitre II : Les méthodes d'optimisation	15
1. Introduction	16
2. L'optimisation	16
2.1 La recherche opérationnelle (RO)	16
2.1.1 Définition	16
2.1.2 Principe de résolution d'un problème en recherche opérationnelle	16
3. Les problèmes d'optimisation	17
4. Les éléments d'optimisation.....	17
5. Méthodes de résolution d'optimisation.....	18
5.1 Les méthodes de résolution exactes.....	18
5.1.1 Le Simplexe.....	19
5.2 Les méta-heuristiques	19
5.2.1 Les algorithmes de colonies de fourmis	20
5.3 Optimisation du réseau logistique	22

6.	Définition générale de l'optimisation distribuée.....	23
7.	Utilisation de l'optimisation distribuée dans SCM	24
8.	Conclusion.....	25
Chapitre III : Les systèmes multi agents		26
1.	Introduction	27
2.	Les agents.....	27
2.1	Définition.....	27
2.2	Les Propriété d'un agent.....	28
2.2.1	Autonome	28
2.2.2	Proactif	28
2.2.3	Flexible.....	28
2.2.4	Social.....	28
2.2.5	Situé.....	28
2.3	Communication entre agents	28
2.3.1	Types de communication	28
2.3.1.1	Communications directes (ou explicites)	28
2.3.1.2	Communications indirectes (ou implicites).....	29
2.4	Classification	29
2.4.1	Les agents réactifs	29
2.4.2	Les agents cognitifs	30
2.4.3	Différence entre agent réactifs et agents cognitifs	30
2.4.4	Les agents hybrides	31
3.	Les systèmes Multi-Agents (SMA).....	31
3.1	Définition.....	32
3.2	Conception d'un SMA	32
3.3	Caractéristiques de Systèmes Multi-Agents	32
3.3.1	Ouvert [Vercouter (2000)]	32
3.3.2	Fermé.....	32
3.3.3	Homogène	32
3.3.4	Hétérogène	33
3.4	Interactions entre les agents.....	33
3.5	La coordination entre agents.....	33
3.6	La coopération entre agents	33
3.7	La négociation dans le système multi-agent.....	33
4.	Intérêt des agents pour la modélisation et la simulation de la chaîne logistique.....	34

5.	Plates-formes de développement de systèmes multi agents.....	34
5.1	La plateforme JADE.....	35
Chapitre VI : Conception et implémentation		37
1.	Environnement de développement	38
1.1	Netbeans	38
2.	Langage de programmation utilisé.....	38
2.1	Le langage JAVA	38
2.1	Le langage SQL	38
3	Conception de la base de données.....	39
3.1	Dictionnaire de données	39
3.2	Modèle conceptuel des données	41
3.3	Modèle logique de données	42
4	Fonctionnement globale	42
5	Schéma de fonctionnement de l'application	43
6	Schéma de fonctionnement du système logistique.....	44
7.	Modélisation du système logistique	45
7.1	Les agents	45
7.1.1	Agent client :	45
7.1.2	Agent producteur :	45
7.1.3	Agent Fournisseur :	45
7.1.4	Agent distributeur :	45
7.2	Diagramme de séquence	46
7.3	Diagramme de classes	46
8.	Modélisation des données du système logistique.....	47
8.1	Structure de données.....	47
8.1.1	Structure des commandes.....	48
8.1.2	Structure des offres de vente de la MP.....	48
8.1.3	Structure des offres de transport.....	48
9.	Modélisation de l'algorithme OCF	48
9.1	Graphe du problème	49
9.1.1	Structuration de graphe du problème.....	49
9.2	Résolution par l'algorithme d'OCF	51
9.2.1	Paramétrages de l'algorithme :	51
9.2.2	Résolution.....	51
10.	Implémentation de l'outil	52

10.1 Les interfaces de l'application	52
10.1.1 Fenêtre principale	52
10.1.2 Fenêtre des clients	52
10.1.3 Fenêtre des produits	53
10.1.4 Fenêtre des matières premières.....	54
10.1.5 Fenêtre tableau de bord.....	55
10.1.6 Fenêtre simulateur	56
10.1.7 Fenêtre des résultats.....	57
Conclusion générale	58
Références bibliographiques	59

Tables des figures

Figure 1 Les flux de la chaîne logistique.....	8
Figure 2 Eléments essentiels pour l'optimisation.....	17
Figure 3 Les différents types de méthodes d'optimisation.....	18
Figure 4 L'expérience du pont à double branche.....	21
Figure 5 Modèle d'un agent réactif.....	29
Figure 6 Modèle d'un agent cognitif.....	30
Figure 7 Architecture hybride (Fischer et al 1995a).....	31
Figure 8 Architecture logiciel de la plateforme JADE.....	36
Figure 9 Modèle conceptuel des données.....	41
Figure 10 Schéma de fonctionnement de l'application.....	43
Figure 11 Schéma de fonctionnement du système logistique.....	44
Figure 12 Diagramme de séquence.....	46
Figure 13 Diagramme de classes.....	47
Figure 14 Fenêtre principale.....	52
Figure 15 Fenêtre des clients.....	53
Figure 16 Fenêtre nouveau client.....	53
Figure 17 Fenêtre des produits.....	54
Figure 18 Nouveau produit.....	54
Figure 19 Fenêtre des matières premières.....	55
Figure 20 Nouvelle matière première.....	55

Figure 21 Fenêtre tableau de bord.....	56
Figure 22 Fenêtre simulateur.....	57
Figure 23 Fenêtre des résultats	57

Introduction générale

Les chaînes d'approvisionnements ou Supply Chain en Anglais (SC) peuvent être considérées de nos jours adaptatives et intelligentes, réseaux comportant des dispositifs de collaboration et de visibilité entre les acteurs pour traiter la dynamique de l'environnement.

Les SC sont des entités économiques séparées et indépendantes, intéressées plus par leurs objectifs locaux que l'exécution de système global, l'approche centralisée de gestion, où un seul acteur optimise l'exécution globale, devient alors moins réaliste. Celle-ci peut être remplacée par des approches décentralisées, où la collaboration entre les acteurs peut améliorer l'exécution individuelle et globale. La question clé est d'aligner les objectifs des acteurs et de coordonner leurs décisions pour optimiser l'exécution des SC.

La gestion de SC ou Supply Chain Management en Anglais (SCM) cherche à déterminer les niveaux des stocks dans les centres logistiques en fonction de la demande, ... La concordance entre les centres de prise de décision dans la chaîne peut être facilement accomplie par un cadre de multi-agents. Plusieurs travaux de recherche se sont consacrés à l'utilisation des SMA pour résoudre des problèmes d'optimisation liés aux chaînes logistiques [Aitsilarbi, 2013], cela dit, la recherche dans ce domaine a abordé principalement l'interaction entre les agents, les solutions d'optimisation sont généralement résolues par des règles. Cependant, ces méthodes ne sont pas suffisantes pour traiter la complexité des problèmes réels. Pour avoir une efficacité de SC, les protocoles d'échanges devraient permettre l'échange d'informations pendant le processus d'optimisation, afin de permettre aux agents de réagir à toute perturbation.

Ce document présente une méthodologie de gestion de SC s'appuyant sur la notion du multi-agent et basée sur la description de la SC comme étant un ensemble de problèmes distribués, Cette approche emploie la méta-heuristique d'Optimisation par Colonie de Fourmis (OCF) pour réaliser la coopération entre différents acteurs d'un SC.

Afin de présenter au mieux notre travail, le premier chapitre aborde les chaînes logistiques, le second chapitre concerne les méthodes d'optimisation appliquées en particulier à des systèmes distribués, le troisième chapitre montre les aspects essentiels qui concernent les systèmes multi agents et le dernier chapitre dévoile la conclusion issue de ce travail et quelques perspectives.

Chapitre I : Les chaînes logistiques

Introduction

La chaîne logistique n'est pas un concept moderne. Depuis des décennies, les matières premières ou les produits finis traversent les continents pour arriver aux clients. Cependant, le développement et le progrès important des technologies et plus précisément celles de l'information et de la communication ont intensifié le besoin de formaliser les interactions entre les acteurs et d'intégrer des nouvelles pratiques de gestion pour piloter et optimiser le fonctionnement de tout le réseau. Cette intégration a complexifié la structure organisationnelle et la prise de décision dans le but de réguler les flux dans la chaîne logistique.

1. La chaîne logistique

Le terme « chaîne logistique » vient de l'Anglais Supply Chain qui signifie littéralement « chaîne d'approvisionnement ». Il existe plusieurs définitions de la « chaîne logistique ». Une des définitions trouvées dans la littérature :

Définition : "La chaîne logistique peut être considérée comme le réseau d'entreprises qui participent, en amont et en aval, aux différents processus et activités qui créent de la valeur sous forme de produits et de services apportés au consommateur final. En d'autres termes, une chaîne logistique est composée de plusieurs entreprises, en amont (fourniture de matières et composants) et en aval (distribution), et du client final." [Cristopher, 1992].

1.1 Les flux de la chaîne logistique

Les flux correspondent à toute entité, matérielle ou non, circulante entre les maillons de la SC. Nous distinguons le flux physique, le flux de données et le flux financier.

1.1.1 Le flux physique :

Entité qui circule au niveau de la SC de l'amont vers l'aval (du fournisseur vers le client) pour fournir de la valeur ajoutée au client final. Nous avons ici les produits, matières premières, ..

1.1.2 Le flux de données :

Représente les données qui circulent au niveau de la SC dans les deux sens. Les données sont utilisées par les acteurs afin de coordonner leurs activités et planifier et prévoir les demandes futures,

L'analyse des définitions dans le domaine de SC, nous a permis de classer les données en :

- **Données informationnelles:** Sont les données de gestion (valeurs, ratios, prix, capacité, etc.) et les données informatiques (statiques, dynamiques ou historiques).

- **Données décisionnelles** : Données qui caractérisent une décision prise par les acteurs de la chaîne à long, moyen et court terme (plan de production, etc.).
- **Métriques** : sont les indicateurs et les mesures qui permettent de piloter la chaîne logistique et de mesurer sa performance à long, moyen et court terme.

1.1.3 Le flux financier :

Ou flux monétaire, circule dans le sens inverse du flux physique. Représente la valeur totale de ventes et d'achats dans une période comptable.

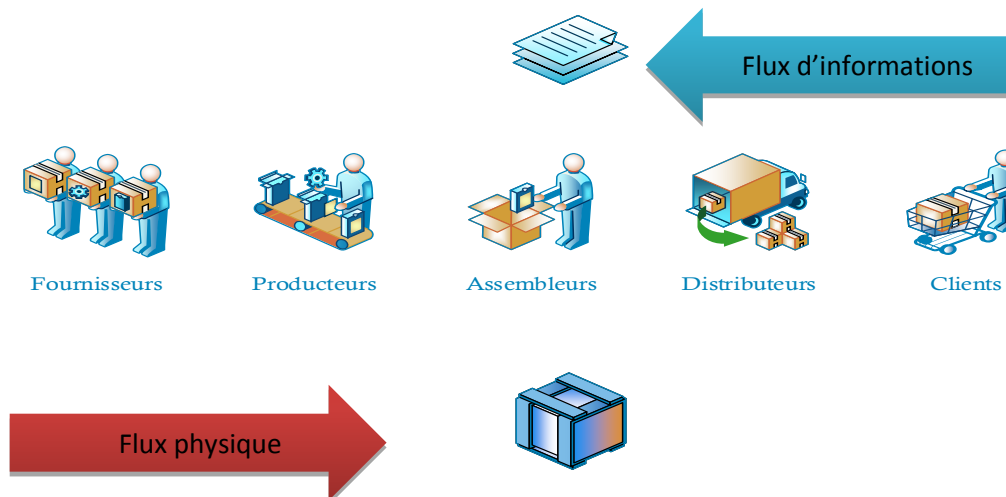


Figure 1 Les flux de la chaîne logistique.

1.2 Les acteurs de la chaîne logistique

La chaîne logistique contient des acteurs, juridiquement indépendants, stratégiquement différents, qui se coordonnent (coopèrent) dans la réalisation d'un ensemble d'activités ayant pour objectif la satisfaction d'un client en termes de coûts, de qualité, de délai et de services associés au produit [Christopher, 1992]. Ces acteurs sont en général les :

- Fournisseurs,
- Producteurs, plus assembleurs,
- Transporteurs,
- Détaillants, plus les clients.

1.3 Les processus

Un processus est défini comme étant un ensemble d'activités corrélées ou interactives qui transforment des éléments d'entrée en éléments de sortie. Dans une SC, plusieurs processus se combinent et se complètent afin de fournir un bien au consommateur final. Ces processus diffèrent selon les secteurs d'activité de l'entreprise. [Cooper et al., 1997] et [Lambert, 2006] ont identifié les processus clés de la SC qui peuvent être catégorisés en cinq macro processus :

- **La planification** : est un processus ayant pour but l'organisation des autres processus de la chaîne logistique. Il porte généralement sur trois activités fondamentales : la prévision de la demande, la gestion du stock et la planification de la production.
- **L'approvisionnement** : est défini à travers les activités nécessaires pour récupérer de la matière première afin de fabriquer le produit.
- **La production** : représente l'ensemble des activités nécessaires pour réaliser le produit, le fabriquer et le stocker. Il se base essentiellement sur la conception du produit et la gestion de la production et des services.
- **La distribution** : englobe toutes les activités prenant en charge les commandes clients et leur livraison. Il inclut la gestion de la commande (entrée de commande et traitement), la gestion du transport et la livraison aux clients.
- **La gestion des retours** : processus récent dans le modèle prenant en compte toutes les activités nécessaires pour gérer le retour du produit par un maillon du réseau.

2. Gestion de chaînes logistiques (Supply Chain Management)

2.1 Définitions

De nombreuses définitions de la « Gestion de Chaînes Logistiques » (ou de « Supply Chain Management » dans la littérature anglo-saxonne) ont été proposées.

De manière générale on peut définir la gestion de chaînes logistiques comme les décisions qui permettent d'optimiser les performances de la chaîne logistique.

[MENT & al 01] proposent la définition du « Supply Chain Management » suivante : « *supplychain management peut être défini comme la coordination systémique, stratégique des fonctions opérationnelles classiques et de leurs tactiques respectives à l'intérieur d'une même entreprise et entre partenaires au sein de la chaîne logistique, dans le but d'améliorer la performance à long terme de chaque entreprise membre et de l'ensemble de la chaîne* ».

3 Modélisation décentralisée de chaîne d'approvisionnements

3.1 La modélisation des problèmes des chaînes logistiques

La modélisation des chaînes logistiques et la réévaluation permanentes de leur capacité à réagir aux changements et aux éléments extérieurs sont des activités essentielles pour le management des entreprises.[SIMC& al, 07]

L'objectif principal du SCM est de maximiser la rentabilité des entités du réseau de la SC. Ceci ne peut être réalisé que si toutes les entités ont la volonté d'optimiser la chaîne d'approvisionnements dans son ensemble. Notons que plusieurs objectifs sont souvent contradictoires les uns avec les autres. Ainsi, le problème de SC devrait être considéré comme problème d'optimisation multi objectifs.

Il y a souvent des contraintes qui violent les solutions optimales, ainsi les contraintes de la SC doivent être satisfaites, d'où les problèmes résolus par satisfaction de contrainte (CSP). Un autre aspect à considérer est que le problème est logiquement physiquement distribué. Par conséquent, le problème de SC a besoin de techniques de résolution de problèmes distribués. A partir de cela, nous pouvons définir le problème de SC comme étant un problème d'optimisation multi objectifs et de satisfaction de contraintes distribué.

En général, nous pouvons considérer un modèle générique de SC ayant trois systèmes : un système logistique, ses fournisseurs et les distributeurs. Le système de fournisseur-logistique-distribution proposé dans ce travail utilise l'optimisation distribuée en considérant tous les niveaux de la chaîne. Le système logistique rassemble les commandes des clients et programme les composants fournis par les fournisseurs comme commandes. Ce système est aussi responsable du pilotage de la coopération entre les acteurs du réseau et pour la communication entre les clients et la SC. La Fig. 3 montre une représentation schématique d'un système multi-échelons (niveaux).

L'approche de modélisation proposée dans ce document décrit les sous-systèmes de la SC par un problème d'optimisation : le système logistique est décrit par le problème général d'optimisation de tâches [C.A. Silva, 2005]; la SC par le problème d'optimisation de planification de fabrication ; et le système de distribution par le problème de cheminement de véhicule (VRP) [G.B. Dantzing, 1959].

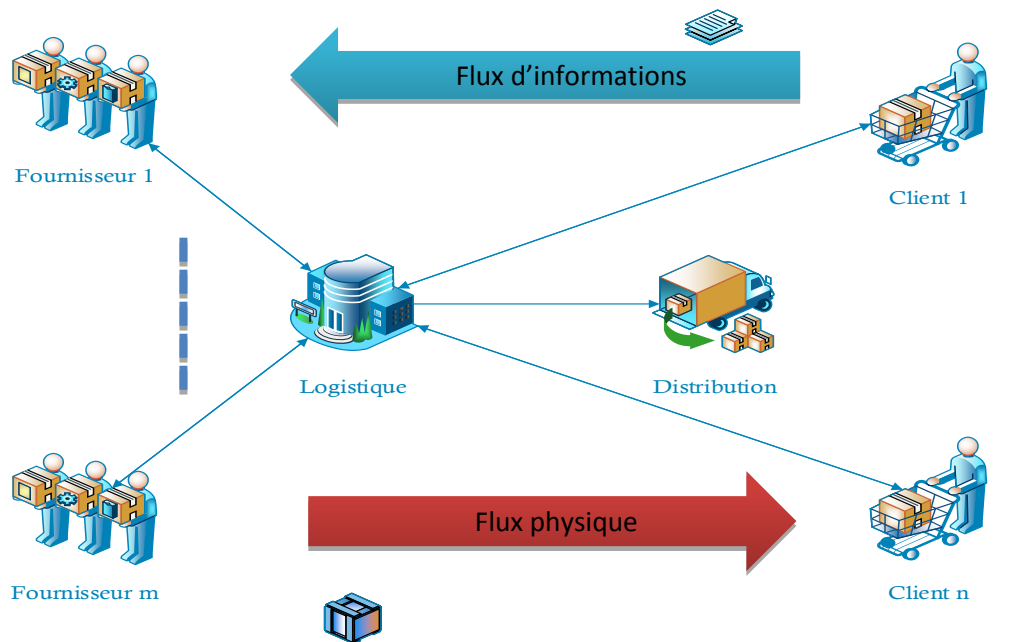


Figure 2 - Système générique de chaîne d'approvisionnements de multi-échelons

4 Gestion distribuée de chaîne d'approvisionnements

La gestion de SC décentralisée peut être définie en tant que mise en œuvre des mécanismes qui favorisent l'adoption des actions locales d'un mode coordonné par chaque membre individuel tout en optimisant leur propre performance qui conduisent à l'optimisation de la SC globale. Cependant, les problèmes de système local et les paramètres de l'information sont des problèmes très spécifiques, et le mécanisme de coopération doit être en général suffisant pour permettre un échange d'informations efficace à travers tous les systèmes.

4.1 Optimisation du réseau logistique

Dans le contexte concurrentiel de la nouvelle économie, il devient important que les entreprises ajustent leur réseau logistique afin que celui-ci soit en ligne avec la réalité. La détermination optimale d'un réseau logistique doit répondre à des questions:

- Combien devrait-il y avoir d'usines et d'entrepôts dans le réseau ?
- Quelles technologies de production, de manutention et de stockage devrait-on adopter et quelle capacité devrait-on avoir ? De quelle façon se fait l'affectation des clients aux entrepôts, des entrepôts aux usines et des usines aux fournisseurs ?

- Quel fournisseur doit servir quel producteur ?

4.2 Définition générale de l'optimisation distribuée

L'idée fondamentale de la programmation dynamique distribuée est qu'il est possible d'optimiser une fonction définie dans un espace de dimension n en utilisant n centres de calcul.

Chacun optimise la fonction sur sa propre direction et échange l'information périodiquement avec les autres centres, tels que la convergence à un optimum global peut être réalisée dans certaines conditions. Un des problèmes d'optimisation à l'origine proposés dans [Bertsekas, 1982] était l'optimisation d'un chemin sur un graphe basé sur ce concept, nous définissons le paradigme distribué d'optimisation :

Définition 1 : Traiter un problème d'optimisation $X = \{X_1, \dots, X_n\} \in x^n$ se composant d'un ensemble de n sous-problèmes X_k , avec $k = 1, \dots, n$, et $f = f_1 \circ \dots \circ f_n$: soit la fonction de coût du problème X , où chacune des n fonctions de coût f_k correspond au sous-problème X_k et \circ est un certain opérateur d'agrégation. L'optimisation distribuée est une méthodologie où les n processus d'optimisation $\min[f_k]$ s'exécutent en parallèle, résolvant les fonctions de coût f_k pour chaque sous-problème X_k . L'optimisation globale est définie comme :

$$\min[f(t)] = \diamond_{k=1}^n \min[f_k(t - l_k)] = \min[f_1(t - l_1) \diamond \dots \diamond \min[f_n(t - l_n)] \quad (5)$$

Où t décrit l'itération actuelle d'optimisation, l_k décrit les n différentes itérations d'optimisation précédentes et \diamond décrit l'échange de l'information entre les différents sous-systèmes. Le problème global X , défini sur l'espace de n dimension x^n se compose d'un ensemble de n problèmes locaux X_k , où chaque problème résout sa propre fonction de coût local f_k .

Les sous-espaces \mathcal{X}_k où chaque problème X_k est définie, ne sont pas disjonctifs, ainsi l'optimisation de la fonction de coût f_k du problème X_k peut dépendre de la solution de la fonction de coût f_h du problème X_h et vice-versa, avec $h \neq k$ et $h = 1, \dots, n$. Par conséquent, la fonction de coût $f = f_1 \circ \dots \circ f_n$ consiste une agrégation fortement non triviale \circ de différentes fonctions locales de coût f_k .

Définition 2 : Considérer le processus d'optimisation globale

$$\min[f(t)] = \min[f_1(t - l_1) \diamond \dots \diamond \min[f_n(t - l_n)]$$

Si $\forall_{k=1, \dots, n} l_k = 0$ alors l'optimisation distribuée serait synchrone. Si $\exists_{k=1, \dots, n} l_k \neq 0$ alors l'optimisation distribuée serait asynchrone. Ceci signifie qu'à chaque itération dans l'optimisation synchrone, les méthodes accèdent à l'information réelle de toutes les autres méthodes. Dans la méthode asynchrone, à chaque itération chaque méthode d'optimisation peut accéder à l'information des itérations précédentes à n'importe quels moments l_k du temps. La méthode asynchrone est employée pour éviter des problèmes de convergence de chaque problème d'optimisation $\min[f_k]$.

4.3 Utilisation de l'optimisation distribuée dans SCM

Le problème de chaîne d'approvisionnements présenté peut être décrit comme problème de gestion $X_{SCM} = \{X_L, X_{M_1}, \dots, X_{M_m}, X_D\}$, où X_L est le problème de planification logistique, $X_{M_i}, i = 1, \dots, m$ est le problème de distribution.

Nous considérons que la fonction de coût de chaîne d'approvisionnements est :

$$f_{SCM} = f_{M_1} \circ \dots \circ f_{M_m} \circ f_L \circ f_D \quad (6)$$

Où f_L, f_{M_i} sont les expressions proposées dans (1) - (3), respectivement.

Observer que la fonction de coût globale de chaîne d'approvisionnements f_{SCM} peut-être l'index de performance p_{scm} défini dans (4), si l'opérateur d'agrégation \circ est tout simplement une opération de somme. Noter que la gestion de chaîne d'approvisionnements est faite à deux instants distincts :

- D'abord, les fournisseurs décident au sujet de leur propre politique de planification, ce peut être influencé ou pas par le système logistique avant qu'il reçoive les nouveaux composants au centre de croix-amarrage. Cependant, le système de distribution n'est pas lié à ce processus de décision, puisqu'il n'y a aucun article à distribuer.
- Puis, le processus logistique choisit les commandes à livrer après qu'il reçoive les nouveaux composants, sous l'influence ou pas du système de distribution. Cependant, les composants sont déjà produits et la SC n'est pas liée à ce processus de décision.

Ainsi, le problème de gestion quotidien de SC $\min[f_{SCM}]$ est divisé en deux problèmes séquentiels de coopération à chaque jour, qui sont le problème fournissances logistique $\min[f_{SL}]$ et le problème de distribution logistique :

$$\min[f_{SCM}] = \min[f_{SL}] \rightarrow \min[f_{LD}] \quad (7)$$

Où l'opérateur \rightarrow indique l'ordre chronologique.

Le problème de fournissances logistique est défini comme :

$$\min[f_{SL}] = \min[f_{M_1}] \diamond \dots \diamond \min[f_{M_m}] \diamond \min[f_L] \quad (8)$$

Et le problème de distributions logistique est défini comme :

$$\min[f_{LD}] = \min[f_L] \diamond \min[f_D] \quad (9)$$

Cette formulation permet une séparation totale entre les fournisseurs et les systèmes de distribution. Il est possible que le système logistique utilise la technique d'optimisation de distribution avec un des associés et emploie un mécanisme complètement différent de coordination avec l'autre associé. Puisque le problème global d'optimisation de SC peut être décrit comme ensemble de différents problèmes distribués, le processus d'optimisation permet à chaque problème d'employer des résultats intermédiaires d'optimisation des autres problèmes, à

la fin, la solution globale est optimisée selon les solutions optimales de tous les problèmes locaux, bien que la solution globale finale ne soit pas nécessairement optimale.

La prochaine section montre comment cette procédure peut être facilement mise en application en utilisant l'algorithme d'optimisation par colonie de fourmis.

Conclusion

L'environnement économique est en perpétuelle mutation induisant ainsi une instabilité et une concurrence mondiale auxquelles les entreprises doivent faire face. Cette mutation est principalement due au renversement du rapport entre l'offre et la demande, à la globalisation des marchés économiques mais aussi aux mutations techniques et technologiques.

Ainsi, la survie des entreprises dépend de leur réactivité et de leur capacité d'adaptation aux changements. Dans cette optique, les entreprises créent de plus en plus d'alliances pour pallier aux limites et aux faiblesses individuelles face aux pressions de l'environnement mondial. Ces alliances ont donné lieu à de nouvelles formes organisationnelles et relationnelles dans lesquelles s'inscrivent les SC.

Cela dit, le fonctionnement de ces organisations doit être optimisé afin d'arriver aux buts souhaités, le chapitre suivant montre un aperçu sur les méthodes d'optimisation qui existent pour arriver aux colonies de fourmis, méthode utilisée dans notre apport.

Chapitre II : Les méthodes d'optimisation

1. Introduction

L'optimisation peut être définie comme la sélection du meilleur élément (optimum) parmi des éléments autorisés (espace de recherche), en fonction d'un critère de comparaison.

En général, l'optimisation d'un problème revient à minimiser ou maximiser la valeur d'une fonction mathématique (appelée fonction objectif) dans un espace de recherche.

Historiquement, les premières méthodes d'optimisation provenaient de résolutions mathématiques. Des méthodes algorithmiques itératives ont été proposées par Newton et Gauss. Les premières méthodes dites de "programmation linéaire" ont été introduites par Kantorovich [Kantorovich 60] peu de temps avant qu'elles ne soient redécouvertes et améliorées par Dantzig [Dantzig 63]. Ces méthodes étaient dédiées à la planification de programmes militaires pendant la seconde guerre mondiale. D'où le terme "programmation", qui a fait référence aux termes planification et ordonnancement. Ces méthodes ont ensuite été étendues à l'optimisation linéaire.

De nos jours, les applications peuvent concerner l'optimisation dans le domaine de la logistique, en particulier, dans la gestion des chaînes logistiques, afin d'optimiser essentiellement les coûts et le temps.

Dans ce chapitre, nous allons présenter la notion d'optimisation ainsi que ses méthodes d'une part, et l'optimisation des SC d'autre part.

2. L'optimisation

2.1 La recherche opérationnelle (RO)

2.1.1 Définition

La RO peut être vue comme un ensemble de méthodes mathématico informatiques destiné à l'optimisation des processus de production et de diffusion des produits.

Selon Robert FAURE (1918-1982), pionnier de la RO en France, la RO peut être définie comme étant l'ensemble des méthodes et techniques rationnelles d'analyse et de synthèse des phénomènes d'organisation utilisables pour élaborer de meilleures décisions. ("Précis de recherche opérationnelle", ed. Dunod, 2000).

2.1.2 Principe de résolution d'un problème en recherche opérationnelle

Afin de résoudre un problème à l'aide de la RO, les étapes suivantes doivent être suivies :

- 1) Détection du problème ;
- 2) Formulation du problème ;
- 3) Elaboration d'un modèle ;
- 4) Collecte de données ;

- 5) Résolution du modèle ;
- 6) Validation du modèle ;
- 7) Prise de décisions et implémentation de la solution.

3. Les problèmes d'optimisation

La résolution des problèmes d'optimisation est utilisée dans un grand nombre de domaines. A l'origine, ce sont les militaires qui se sont intéressés à ces questions pendant la seconde guerre mondiale (avec la notion de logistique). Le développement de l'informatique a ouvert de nouveaux horizons à la résolution de ces problèmes, et a permis un élargissement des champs d'application de ces techniques. Aussi, la résolution d'un problème d'optimisation est un problème complexe, car de nombreux facteurs interviennent et interagissent entre eux.

4. Les éléments d'optimisation

L'optimisation est une des mathématiques consacrée à l'étude des minimums/maximums d'une fonction à une ou plusieurs variables sur un certain domaine de définition, de l'étude de leur existence à leur détermination, en général par la mise en œuvre d'un algorithme et par suite un programme. Pour bien mener une opération, des éléments sont indispensables et conditionnent la solution trouvée. La figure suivante présente les quatre éléments essentiels à la résolution d'un problème d'optimisation.

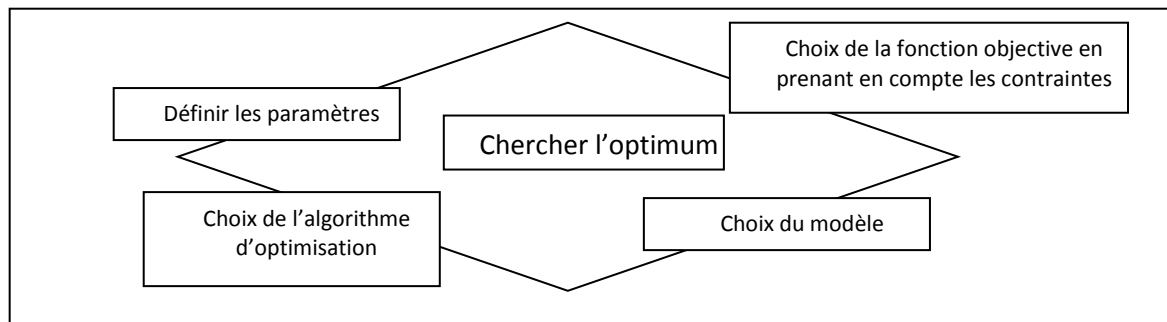


Figure 2 Eléments essentiels pour l'optimisation.

En général, beaucoup de paramètres sont indispensables. Certains paramètres ont une influence sur la fonction choisie. Les paramètres les plus influents sont :

a) Une fonction objectif : définit l'objectif à atteindre. Le problème est formulé en un problème d'optimisation par l'intermédiaire de la fonction objectif. C'est cette fonction qui est au centre de l'optimisation.

b) Un modèle du système étudié : Doit être précis, robuste et malléable du système. Ce modèle doit être utilisable sur un domaine d'étude le plus largement possible.

c) Un algorithme d'optimisation : permet de trouver la solution. Différentes méthodes d'optimisation existent et sont présentées dans la section suivante.

5. Méthodes de résolution d'optimisation

Les méthodes d'optimisation peuvent être classées en fonction du problème posé, comme le montre la figure 3 :

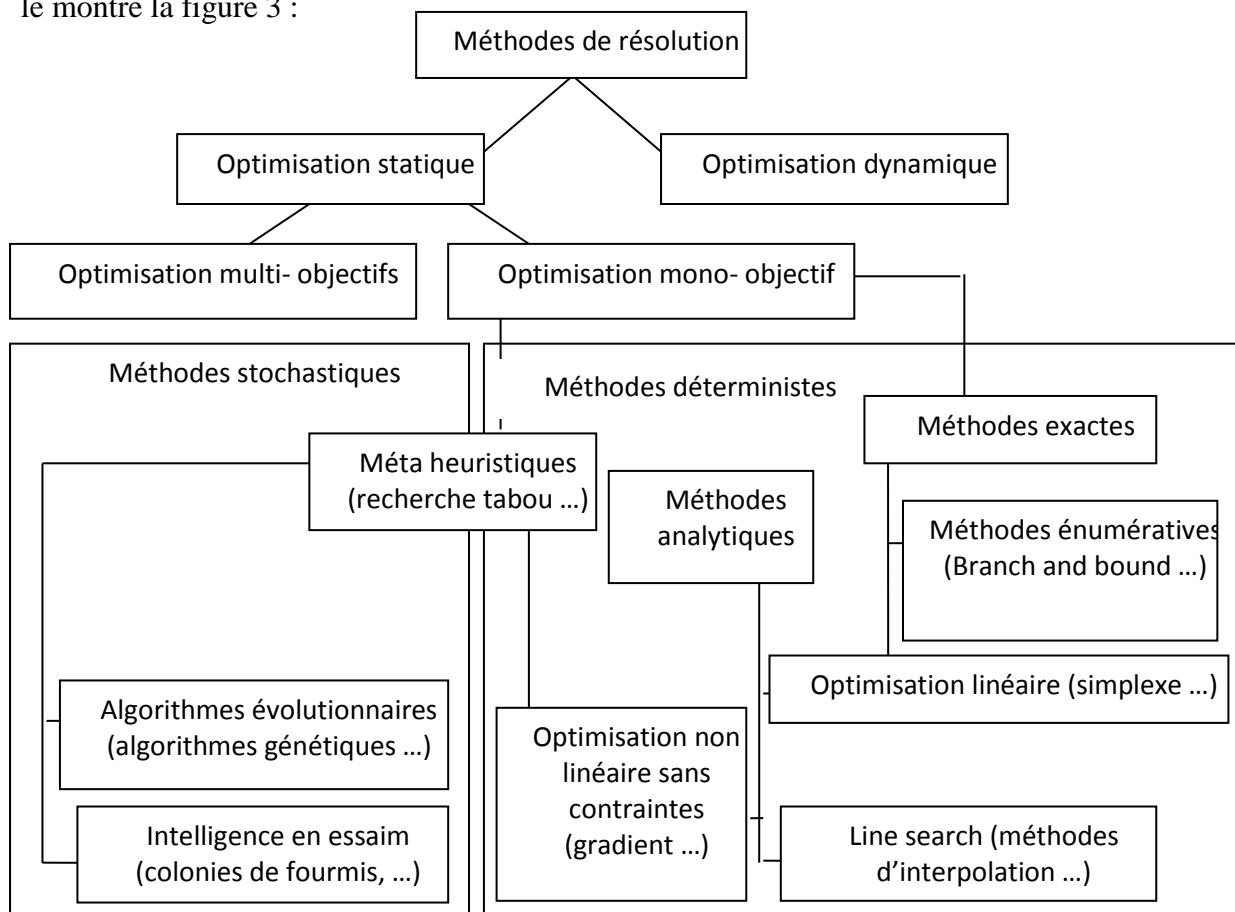


Figure 3 Les différents types de méthodes d'optimisation.

5.1 Les méthodes de résolution exactes

Les méthodes exactes [Dudzinski 87] [Kohl 95] sont des méthodes permettant d'obtenir la solution optimale d'un problème, en un temps "raisonnable". Elles s'opposent aux heuristiques permettent d'obtenir théoriquement la solution optimale (à un epsilon près en raison de la précision numérique des calculs), et non une solution approchée. Dans cette famille d'algorithmes, nous pouvons citer par exemple les méthodes plus mathématiques, telles que la programmation linéaire (méthode du simplexe de Dantzig [Dantzig 63]).

D'autres méthodes exactes reposent sur le principe "diviser-et-régner" : Le problème est décomposé en sous-problèmes, plus faciles à résoudre, les résultats sont combinés jusqu'à arriver à résoudre le problème initial. Notons que ce principe ne s'applique que dans le cadre de l'hypothèse de Markov [Howard 60], qui suppose qu'à chaque étape de décomposition du problème on ne puisse pas remettre en cause la solution optimale de chaque sous problème.

Les méthodes qui en découlent sont nombreuses, nous pouvons citer la méthode par séparation et évaluation (Branch and bound) [Land 60] et ses dérivées (Branch and cut, etc.) et la programmation dynamique. Pour être efficaces, ces méthodes exactes d'énumération implicite utilisent néanmoins des heuristiques (pour choisir les branchements, pour calculer des bornes, etc...). Nous pouvons également citer l'algorithme A* [Hart 68], qui permet de trouver un chemin entre deux sommets d'un graphe, et qui est très souvent utilisé dans le domaine de l'intelligence artificielle. De par leur nature, les méthodes exactes ne peuvent donc s'appliquer qu'à des problèmes spécifiques [Sprecher 94] [Wolf 99].

5.1.1 Le Simplexe

Constitue la méthode la plus utilisée en RO. A été mis au point en 1947 par G. Dantzig pour la résolution des programmes linéaires continus sur l'allocation optimale des ressources dans la production de biens manufacturés. L'algorithme a été publié dans l'article « Programming in Linear Structure » du journal *Econometrica* en 1949. Il sert depuis à la résolution de modèles linéaires pour résoudre des problèmes de transport, d'affectation, etc.

5.2 Les méta-heuristiques

Ces algorithmes permettent généralement d'obtenir une solution de très bonne qualité pour des problèmes issus des domaines de la RO ou de l'ingénierie dont on ne connaît pas de méthodes efficaces pour les traiter ou bien quand la résolution du problème nécessite un temps élevé ou une grande mémoire de stockage.

Nous pouvons distinguer les méta-heuristiques qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales (La recherche Tabou et le Recuit Simulé). L'optimisation par essaim de particules et les algorithmes de colonies de fourmis travaillent avec une population.

5.2.1 Les algorithmes de colonies de fourmis

Les algorithmes de colonies de fourmis ont été proposés par Colomni, Dorigo et Maniezzo en 1992 (Colomni et al. 1992b) et appliqués la première fois au problème du voyageur de commerce. Ce sont des algorithmes itératifs à population où tous les individus partagent un savoir commun qui leur permet d'orienter leurs futurs choix et d'indiquer aux autres individus des choix à suivre ou à éviter.

Le principe de cette méta-heuristique repose sur le comportement particulier des fourmis, elles utilisent pour communiquer une substance chimique volatile particulière appelée *phéromone* grâce à une glande située dans leur abdomen.

En quittant leur nid pour explorer leur environnement à la recherche de la nourriture, les fourmis arrivent à élaborer des chemins qui sont fréquemment les plus courts pour aller du nid vers une source de nourriture. Chaque fourmi dépose alors une quantité de phéromone sur ces pistes qui deviendront un moyen de communication entre elles, les fourmis choisissent ainsi avec une probabilité élevée les chemins contenant les plus fortes concentrations de phéromones à l'aide des récepteurs situés dans leurs antennes.

La figure 3 montre et confirme ce constat, une expérience a été faite par Gauss et Deneubourg en 1989 appelée expérience du pont à double branche, les fourmis qui retournent au nid rapidement, après avoir visité la source de nourriture, sont celles qui ont choisi la branche courtée, les fourmis empruntant cette branche faisant plus d'aller-retour, et par conséquent la quantité de phéromones déposée sur la plus courte branche est supérieure à celle présente sur la plus longue branche.

Les fourmis sont donc plus attirées vers les pistes de plus grande concentration en phéromones, la branche courte sera alors la plus empruntée par la majorité des fourmis. Cette méta-heuristique a permis de résoudre différents problèmes d'optimisation combinatoire à forte complexité, comme le problème du voyageur de commerce (Dorigo et al. 1996), le problème de coloration de graphe (Costa et Hertz 1997) et le problème de routage de véhicules (Mazzeo et Loiseau 2004), etc.

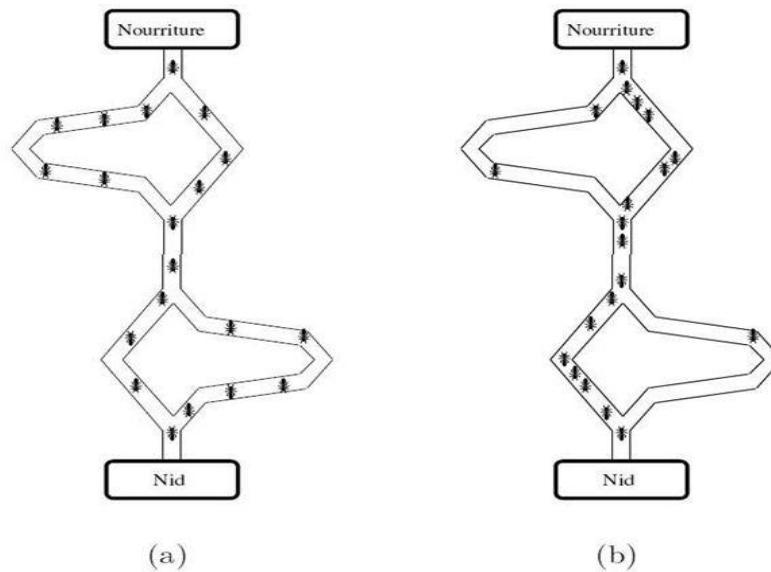


Figure 4 L'expérience du pont à double branche.

a) Principe de l'algorithme

Dans ce paragraphe, nous décrivons l'implémentation d'un algorithme de colonie de fourmis dit 'Ant System' (AS) (Colomi et al. 1992a), mis en œuvre pour résoudre le problème de voyageur de commerce (TSP), ce problème consiste à trouver le plus court cycle hamiltonien dans un graphe, où chaque sommet du graphe représente une ville. La distance entre les villes i et j est représentée par d_{ij} , le couple (i, j) représente l'arête entre ces deux villes.

La quantité de phéromone est d'abord initialisée sur les arêtes à $\tau_{ij} > 0$, chaque fourmi parcourt le graphe et construit un trajet complet (une solution). A chaque étape de la construction des solutions, la fourmi doit décider à quel sommet elle va se déplacer, cette décision est prise d'une manière probabiliste fondée sur les valeurs de phéromone et d'une information statistique qui permet notamment de trouver une bonne solution.

La probabilité pour qu'une fourmi k se déplace du sommet i au sommet j , qui appartient à un ensemble de sommets qui ne sont pas encore visités par la fourmi k noté S_i^k , est :

$$p_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in S_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta}$$

α et β sont deux paramètres qui influencent sur l'importance de l'intensité de phéromone τ_{ij} , et l'information statistique dite visibilité η_{ij} . Cette valeur guide le choix des fourmis vers des villes proches, et éviter le plus possible celles trop lointaines ($\eta_{ij} = \frac{1}{d_{ij}}$). Pour $\alpha = 0$, on prend en

compte juste la visibilité c'est-à-dire que le choix sera tombé à chaque fois sur la ville la plus proche. Si $b = 0$, seules les pistes de phéromones jouent sur le choix. Pour éviter une sélection trop rapide d'un chemin, un compromis convenable entre ces deux paramètres est obligatoire.

b) Mise à jour de phéromones

Quand toutes les fourmis ont construit une solution, une quantité de phéromones $\Delta\tau_{ij}^k$ est déposée par chaque fourmi k sur son trajet. Pour toute itération t , si le chemin (i, j) est dans la tournée de la fourmi k la quantité de phéromones déposée sur ce trajet est :

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L^k(t)}$$

Où $L^k(t)$, la longueur totale de la tournée de la fourmi k , et Q est une constante. Donc l'ajout de la quantité de phéromones dépend certainement de la qualité de la solution obtenue c'est-à-dire plus la tournée parcourue est petite plus l'ajout de la quantité de phéromones est important. Dans le but de ne pas négliger toutes les mauvaises solutions obtenues, et ainsi éviter la convergence vers des optima locaux de mauvaise qualité, le concept d'évaporation des pistes de phéromones est simulé à travers un paramètre r appelé le taux d'évaporation ($0 < r < 1$) comme suit :

$$\tau_{ij}(t+1) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Où, $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$, t représente une itération donnée et m le nombre de fourmis.

L'optimisation des chaînes logistiques

5.3 Optimisation du réseau logistique

Comme nous l'avons mentionné plus haut, dans le contexte actuel, il devient primordial que les entreprises ajustent leur réseau logistique afin que celui-ci soit en ligne avec la réalité. Pour répondre aux questions posées plus haut, le gestionnaire doit identifier l'ensemble des coûts liés à son réseau de distribution: les coûts de production liés à chacune des usines, les coûts de transport, ... etc.

Le gestionnaire doit recueillir les données suivantes :

- Positionnement des différents acteurs ;
- La capacité de production et d'entreposage des usines et la capacité d'entreposage des entrepôts ;

- Les matrices de distance entre les fournisseurs et les usines, les usines et les entrepôts et les entrepôts et les clients.

Remarquons que dans un projet d'optimisation d'un réseau logistique, la collecte d'informations est l'étape la plus laborieuse du projet. L'étape suivante consiste à modéliser l'ensemble des informations recueillies de façon à représenter parfaitement l'ensemble des coûts et des contraintes du problème.

6. Définition générale de l'optimisation distribuée

L'idée fondamentale de la programmation dynamique distribuée est qu'il est possible d'optimiser une fonction définie dans un espace de dimension n en utilisant n centres de calcul. Chacun optimise la fonction sur sa propre direction et échange l'information périodiquement avec les autres centres, tels que la convergence à un optimum global peut être réalisée dans certaines conditions. Un des problèmes d'optimisation à l'origine proposés dans [Bertsekas, 1982] était l'optimisation d'un chemin sur un graphe. Basé sur ce concept, nous définissons le paradigme distribué d'optimisation :

Définition 1 : Traiter un problème d'optimisation $X = \{X_1, \dots, X_n\} \in \mathcal{X}^n$ se composant d'un ensemble de n sous-problèmes X_k , avec $k = 1, \dots, n$, et $f = f_1 \circ \dots \circ f_n$: soit la fonction de coût du problème X , où chacune des n fonctions de coût f_k correspond au sous-problème X_k et \circ est un certain opérateur d'agrégation. L'optimisation distribuée est une méthodologie où les n processus d'optimisation $\min[f_k]$ s'exécutent en parallèle, résolvant les fonctions de coût f_k pour chaque sous-problème X_k . L'optimisation globale est définie comme :

$$\min[f(t)] = \diamond_{k=1}^n \min[f_k(t - l_k)] = \min[f_1(t - l_1) \diamond \dots \diamond \min[f_n(t - l_n)] \quad (5)$$

Où t décrit l'itération actuelle d'optimisation, l_k décrit les n différentes itérations d'optimisation précédentes et \diamond décrit l'échange de l'information entre les différents sous-systèmes. Le problème global X , défini sur l'espace de n dimension \mathcal{X}^n se compose d'un ensemble de n problèmes locaux X_k , où chaque problème résout sa propre fonction de coût local f_k .

Les sous-espaces \mathcal{X}_k où chaque problème X_k est définie, ne sont pas disjonctifs, ainsi l'optimisation de la fonction de coût f_k du problème X_k peut dépendre de la solution de la fonction de coût f_h du problème X_h et vice-versa, avec $h \neq k$ et $h = 1, \dots, n$. Par conséquent, la fonction de coût $f = f_1 \circ \dots \circ f_n$ consiste une agrégation fortement non triviale \circ de différente fonction locale de coût f_k .

Définition 2 : Considérer le processus d'optimisation globale

$$\min[f(t)] = \min[f_1(t - l_1) \diamond \dots \diamond \min[f_n(t - l_n)]$$

Si $\forall_{k=1, \dots, n} l_k = 0$ alors l'optimisation distribuée serait synchrone. Si $\exists_{k=1, \dots, n} l_k \neq 0$ alors l'optimisation distribuée serait asynchrone. Ceci signifie qu'à chaque itération dans l'optimisation synchrone, les méthodes accèdent à l'information réelle de toutes les autres méthodes. Dans la méthode asynchrone, à chaque itération chaque méthode d'optimisation peut accéder à l'information des itérations précédentes à n'importe quels moments l_k du temps. La méthode asynchrone est employée pour éviter des problèmes de convergence de chaque problème d'optimisation $\min[f_k]$.

7. Utilisation de l'optimisation distribuée dans SCM

Le problème de chaîne d'approvisionnements présenté peut être décrit comme problème de gestion $X_{SCM} = \{X_L, X_{M_1}, \dots, X_{M_m}, X_D\}$, où X_L est le problème de la planification logistique, $X_{M_i}, i = 1, \dots, m$ est le problème de distribution.

Nous considérons que la fonction de coût de chaîne d'approvisionnements est :

$$f_{SCM} = f_{M_1} \circ \dots \circ f_{M_m} \circ f_L \circ f_D \quad (6)$$

Où f_L, f_{M_i} sont les expressions proposées dans (1) - (3), respectivement.

Observer que la fonction de coût globale de chaîne d'approvisionnements f_{SCM} peut-être l'index de performance p_{scm} défini dans (4), si l'opérateur d'agrégation \circ est tout simplement une opération de somme. Noter que la gestion de chaîne d'approvisionnements est faite à deux instants distincts :

- D'abord, les fournisseurs décident au sujet de leur propre politique de planification, ce peut être influencé ou pas par le système logistique avant qu'il reçoive les nouveaux composants au centre de crois-amarrage. Cependant, le système de distribution n'est pas lié à ce processus de décision, puisqu'il n'y a aucun article à distribuer.
- Puis, le processus logistique choisit les commandes à livrer après qu'il reçoive les nouveaux composants, sous l'influence ou pas du système de distribution. Cependant, les composants sont déjà produits et la SC n'est pas liée à ce processus de décision.

Ainsi, le problème de gestion quotidien de SC $\min[f_{SCM}]$ est divisé en deux problèmes séquentiels de coopération à chaque jour, qui sont le problème fournissances logistique $\min[f_{SL}]$ et le problème de distribution logistique :

$$\min[f_{SCM}] = \min[f_{SL}] \rightarrow \min[f_{LD}] \quad (7)$$

Où l'opérateur \rightarrow indique l'ordre chronologique.

Le problème de fournissances logistique est défini comme :

$$\min[f_{SL}] = \min[f_{M_1}] \diamond \dots \diamond \min[f_{M_m}] \diamond \min[f_L] \quad (8)$$

Et le problème de distributions logistique est défini comme :

$$\min[f_{LD}] = \min[f_L] \diamond \min[f_D] \quad (9)$$

Cette formulation permet une séparation totale entre les fournisseurs et les systèmes de distribution. Il est possible que le système logistique utilise la technique d'optimisation de distribution avec un des associés et emploie un mécanisme complètement différent de coordination avec l'autre associé. Puisque le problème global d'optimisation de SC peut être décrit comme ensemble de différents problèmes distribués, le processus d'optimisation permet à chaque problème d'employer des résultats intermédiaires d'optimisation des autres problèmes, à la fin, la solution globale est optimisée selon les solutions optimales de tous les problèmes locaux, bien que la solution globale finale ne soit pas nécessairement optimale.

8. Conclusion

Si les méthodes de résolution exactes permettent d'obtenir une solution dont l'optimalité est garantie, dans certaines situations, on peut cependant chercher des solutions de bonne qualité, sans garantie d'optimalité, mais au profit d'un temps de calcul plus réduit. Pour cela, on applique des méthodes appelées méta-heuristiques, adaptées à chaque problème traité, avec cependant l'inconvénient de ne disposer en retour d'aucune information sur la qualité des solutions obtenues.

Les heuristiques ou les méta-heuristiques exploitent généralement des processus aléatoires dans l'exploration de l'espace de recherche pour faire face à l'explosion combinatoire engendré par l'utilisation des méthodes exactes. En plus de cette base stochastique, les méta-heuristiques sont le plus souvent itératives, ainsi le même processus de recherche est répété lors de la résolution. Leur principal intérêt provient justement de leur capacité à éviter les minima locaux en admettant une dégradation de la fonction objective au cours de leur progression.

Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable et rencontrent généralement des difficultés face aux applications de taille importante. En revanche les méthodes approchées ne garantissent pas de trouver une solution exacte, mais seulement une approximation.

Nous avons abordé dans ce chapitre l'optimisation distribuée, nous allons montrer dans ce qui suit la notion de systèmes multi agents pour la prise en compte de l'aspect distribué vu la nécessité de communication, de prise de décision distribuée ...

Chapitre III : Les systèmes multi agents

1. Introduction

Les systèmes multi-agents font partie d'Intelligence Artificielle Distribuée (IAD), qui, à la différence d'IA classique, s'intéresse aux comportements intelligents, résultant de l'activité coopérative de plusieurs agents [Lab, 93].

Les SMA sont à la connexion de plusieurs domaines en particulier de l'intelligence artificielle, des systèmes informatique distribués et du génie logiciel.

Ils se caractérisent par des comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents, que ces interactions tournent autour de la coopération, de la concurrence ou de la coexistence entre ces agents.

2. Les agents

2.1 Définition

Un agent est une entité réelle ou virtuelle dont le comportement est autonome, évoluant dans un environnement qu'il est capable de percevoir et sur lequel il est capable d'agir, et d'interagir avec les autres agents ([Ferber (1995)], [Demazeau et Costa (1996)]).

A partir de cette définition, [Ferber (1995)] définit un agent comme une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement
- qui peut communiquer directement avec d'autres agents
- qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser)
- qui possède des ressources propres
- qui est capable de percevoir (mais de manière limitée) son environnement
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune)
- qui possède des compétences et offre des services
- qui peut éventuellement se reproduire
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

2.2 Les Propriété d'un agent

2.2.1 Autonome

Agir sans l'intervention directe d'un humain (ou d'un autre agent) en contrôlant ses actions et de son état interne.

2.2.2 Proactif

Capable d'avoir un comportement opportuniste, dirigé par ses buts ou sa fonction d'utilité, et prendre des initiatives au moment approprié.

2.2.3 Flexible

Capable de répondre à temps.

2.2.4 Social

Capable d'interagir avec les autres agents (artificiels ou humains)

2.2.5 Situé

Recevoir des entrées sensorielles provenant de son environnement et ainsi effectuer des actions qui sont susceptibles de changer cet environnement.

2.3 Communication entre agents

- Un agent doit être capable de communiquer avec les autres agents.
- Les agents doivent avoir des capacités à manipuler un langage commun.

2.3.1 Types de communication

2.3.1.1 Communications directes (ou explicites)

L'échange direct est réalisé volontairement en direction d'un individu ou groupe d'individus

- communication par partage d'informations
- les agents lisent et déposent une information sur une zone de données commune
- communication par envoi de messages
- communication point à point (téléphone)
- communication par diffusion (broadcast)

2.3.1.2 Communications indirectes (ou implicites)

Les agents laissent des traces (signaux) de leur présence ou de leur action qui sont perçues par d'autres agents

- l'environnement propage (et éventuellement déforme) les signaux déclenchés par la réalisation d'une action ; cela entraîne des types d'échanges limités et permet de ne pas avoir à déterminer précisément le rôle de chaque individu dans le traitement collectif (ex les objets dans l'environnement émettent des signaux ou des champs de potentiels guidant les agents)

2.4 Classification

Dans un Système Multi-Agents, les agents ont des rôles, perçoivent leur environnement souvent à travers des messages. Leurs actions correspondent à des choix de comportement qui dépendent de l'environnement tel qu'il est perçu.

Deux types d'agents ont été distingués. Nous avons les agents réactifs, les agents cognitifs. Les agents cognitifs et réactifs qui peuvent se combiner pour distinguer troisième types d'agents hybrides.

2.4.1 Les agents réactifs

Les agents réactifs possèdent une architecture décisionnelle simple. Comme leur nom l'indique, ils sont munis d'une intelligence réactive. Ulrich (*Ulrich, 1998*) définit ce type d'intelligence comme une base de connaissances régie par une série de règles ou un arbre de décision sensible aux stimuli de l'environnement. Le processus de recherche de solution apporte une solution adéquate pour la situation actuelle et consiste en une procédure de recherche dans la base de connaissances individuelle. En d'autres termes, un agent réactif perçoit son environnement et agit en conséquence des stimuli reçus.

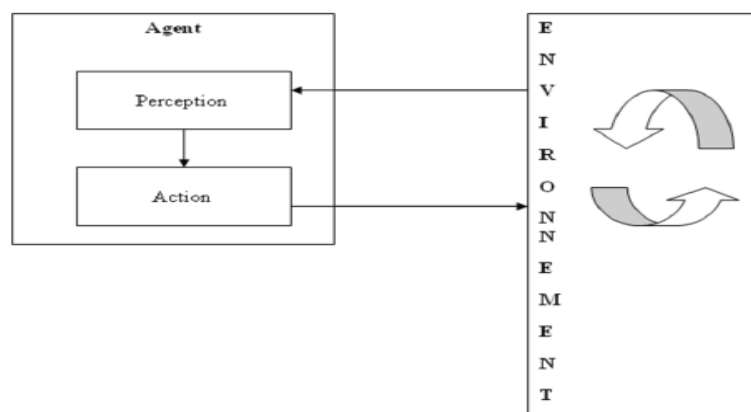


Figure 5 Modèle d'un agent réactif.

2.4.2 Les agents cognitifs

Un agent cognitif possède une représentation explicite et symbolique de son environnement (de soi, du groupe, des autres, de la tâche) dans sa base de connaissances. Cette base de connaissances lui sert de support pour raisonner et aussi prédire des évènements futurs. Ainsi, un agent cognitif possède une intelligence cognitive qui est dirigée par des buts et qui lui permettent de choisir l'action adéquate afin de faire face à une situation complexe. Une intelligence cognitive est décrite par Ulrich (Ulrich,1998) comme une intelligence plus exigeante dans la résolution des problèmes. Elle se base sur des procédures qui sélectionnent des méthodes prédéfinies et prêtes à l'emploi selon l'objectif à réaliser. Ces méthodes sont mises en place selon des critères pertinents qui tendent à fournir la meilleure solution.

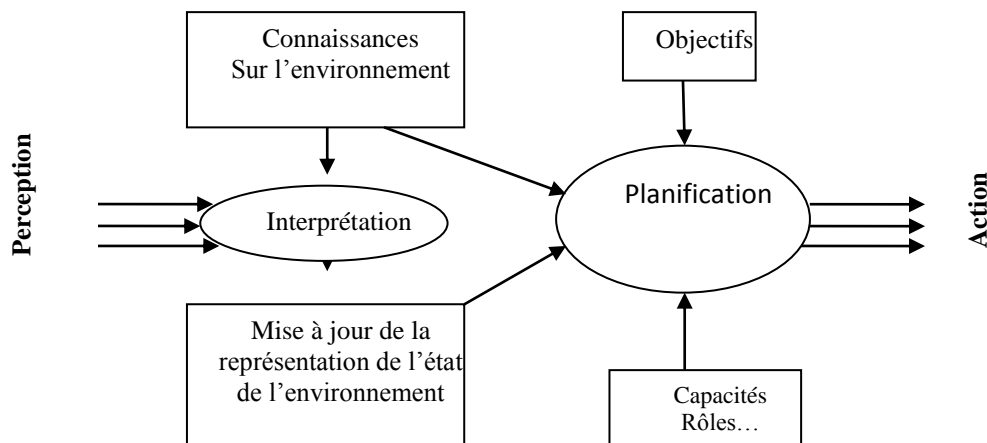


Figure 6 Modèle d'un agent cognitif.

2.4.3 Différence entre agent réactifs et agents cognitifs

Système d'agents cognitifs	Système d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agent complexes	Fonctionnement Stimulus/action
Petit nombre d'agent	Grand nombre d'agents

Tableau 1. Comparaison entre agents cognitifs et agents réactifs

2.4.4 Les agents hybrides

Un agent hybride possède un croisement entre le comportement réactif et le comportement cognitif. Dans ce type d'architecture, les agents sont conçus de niveaux hiérarchiques qui interagissent entre eux. Chaque niveau gère un aspect du comportement de l'agent (O'Hare et Jennings, 1994) :

- Bas niveau : couche réactive qui est en relation directe avec l'environnement et qui raisonne suivant les informations brutes de ce dernier.
- Niveau intermédiaire : couche mentale qui fait abstraction des données brutes et qui se préoccupe d'un aspect de l'environnement plus évolué. Elle se tourne vers les connaissances extraites de l'environnement.
- Niveau supérieur : couche sociale qui tient compte de l'interaction avec les autres agents.

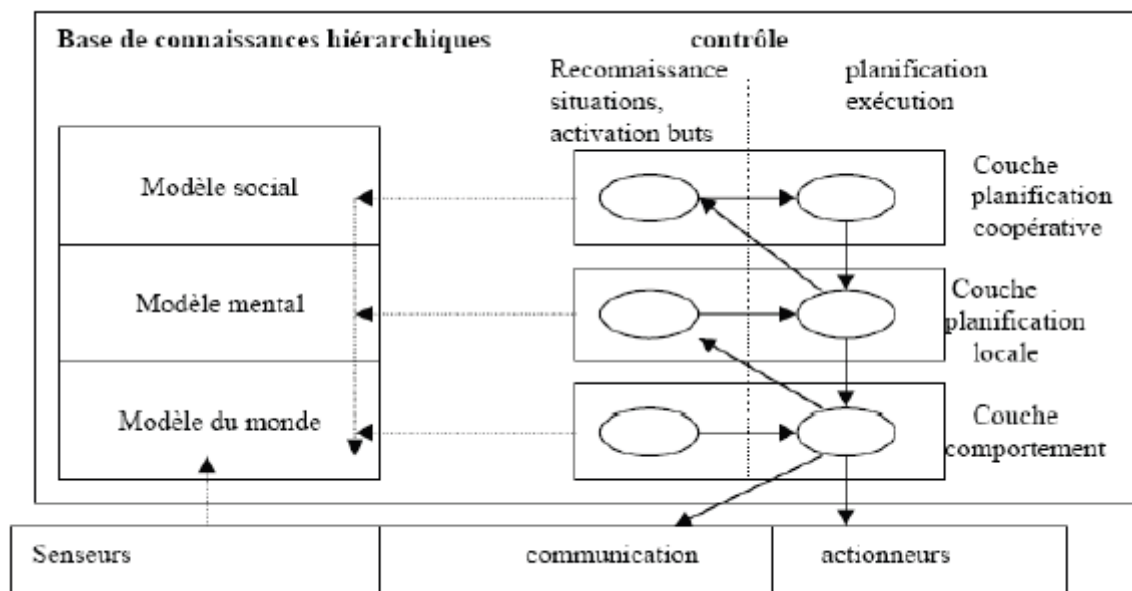


Figure 7 Architecture hybride (Fischer et al 1995a).

Architecture InteRRaP (Integration of ReactiveBehavior And Rational Planning) (Fischer et al., 1995a) qui s'inspire des concepts BDI est un exemple de ce type d'agent (Figure 3). Elle a été utilisée dans des applications telles que le transport routier (Fischer et al., 1995a, b) et la robotique (d'Inverno et al., 1997).

3. Les systèmes Multi-Agents (SMA)

Historiquement, les systèmes multi-agents se positionnent au carrefour de la programmation (logiciels), de l'intelligence artificielle (autonomie de décision), et des systèmes répartis (décentralisation). Le domaine des systèmes multi-agents a vu le jour pendant les années 80 sous

le nom d'Intelligence Artificielle Distribuée (Huhns, 1987) aux Etats Unis et d'Intelligence Artificielle Décentralisée (Demazeau et Müller, 1990) ou systèmes multi-agents en Europe.

3.1 Définition

Un SMA est un système composé d'entités informatiques, appelées des agents, qui évoluent et interagissent dans un environnement commun. La notion d'interaction entre agents est essentielle car chacun d'eux est impliqué dans une dynamique commune, au lieu d'évoluer parallèlement et indépendamment aux autres.[BRIOT & all, 01]

3.2 Conception d'un SMA

Impose une vision locale et décentralisée

- **Locale:** chaque agent est responsable de ses connaissances (encapsulation) et de ses actions (autonomie), mais également de l'organisation qu'il met en place avec d'autres agents. Aucun agent n'a de vue globale du SMA tout entier.
- **Décentralisée:** on s'efforce d'éliminer tout contrôle central. Les tâches à réaliser et les compétences pour le faire sont distribuées sur les agents.

3.3 Caractéristiques de Systèmes Multi-Agents

L'importance de l'approche Système Multi-Agents s'est accrue ces dernières années. Cela peut être expliqué par le fait que les systèmes d'information que l'on cherche à modéliser sont de plus en plus distribués, ouverts, à grande échelle et hétérogènes. C'est pourquoi les Systèmes Multi-Agents présentent souvent les caractéristiques suivantes :

3.3.1 Ouvert [Vercouter (2000)]

Les agents d'un Système Multi-Agents ouvert n'ont pas la possibilité d'avoir une représentation complète de l'environnement. Ce système doit être modulaire et extensible. Les agents y entrent et en sortent librement (ex: un café)

3.3.2 Fermé

L'ensemble d'agents reste le même (ex: un match de football)

3.3.3 Homogène

Deux agents dans un système homogène sont identiques du point de vue de leur modèle et leur architecture.

Tous les agents sont construits sur le même modèle (ex: une colonie de fourmis)

3.3.4 Hétérogène

Des agents de modèles différents, de granularité différentes (ex: l'organisation hospitalière).

3.4 Interactions entre les agents

Un Système Multi-agents est composé d'une collection d'agents indépendants ; ces agents interagissent dans le but d'atteindre conjointement un objectif particulier. L'interaction signifie un type d'action collective où une entité effectue une action ou prend une décision qui est influencée par une autre entité [Hussein Joumaa et Vincent (2008)].

Les SMA ont surtout l'avantage de faire intervenir des schémas d'interaction sophistiqués. Ils peuvent ainsi coexister, être en compétition ou coopérer.

- S'ils ne font que coexister, alors chaque agent ne considère les autres agents que comme des composantes de l'environnement.
- S'ils sont en compétition, alors le but de chaque agent est de maximiser sa propre satisfaction,
- S'ils sont en coopération, alors le but des agents n'est plus seulement de maximiser sa propre satisfaction mais Les agents travaillent ensemble à la résolution d'un problème

3.5 La coordination entre agents

On distingue deux composantes fondamentales de la coordination entre agents, ce sont :

- **L'allocation de ressources rares** : Pour l'allocation des ressources partagées, les agents doivent être capables de faire des transferts de ressources (micro, balles, meubles).
- **La communication de résultats intermédiaires** : les agents doivent être capables de communiquer entre eux de façons à pouvoir échanger les résultats intermédiaires.

3.6 La coopération entre agents

La coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents, il existe plusieurs décompositions possibles, le processus de décomposition doit donc tenir compte des ressources disponibles et des compétences des agents.

3.7 La négociation dans le système multi-agent

La négociation dans le système multi-agent intervient lorsque les agents tous les agents décident de prendre une décision commune alors qu'ils poursuivent des buts différents. On a les deux grands types de négociation :

- Négociation compétitive : les agents d'intérêts différents tentent un choix de groupe sur des alternatives bien définies

- Négociation coopérative : les agents ont un but commun, sont collaboratifs et coopèrent.

4. Intérêt des agents pour la modélisation et la simulation de la chaîne logistique

Critère de comparaison	Chaîne logistique (SC)	Système multi-agent
La multiplicité des entités agissantes	multiples entités aux rôles et compétences différents pour la réalisation des tâches communes.	
Les propriétés des entités	les acteurs de la chaîne possèdent des objectifs, moyens et compétences nécessaires à l'exécution des tâches, et suivent pour cela un ensemble de règles de gestion.	Les agents possèdent des objectifs, compétences, rôles et capacités de raisonnement, qu'ils mettent en œuvre selon différents modes décisionnels complexes
L'habilité sociale des entités	les prises de décision se font au travers de méthodes de coordination et/ou de négociation entre les acteurs	Les agents sont autonomes, sensibles aux modifications de l'environnement, proactifs (initient les prises de décisions et d'actions) et disposent de capacités sociales
Les capacités décisionnelles	apprentissage et raisonnement sont nécessaires à la prise de décision individuelle et collective pour les acteurs	Capacités de raisonnement, d'acquisition ou de modification de connaissances par interaction avec l'environnement possible pour les agents
La coordination entre entités	coordination des acteurs de la SC par le partage de flux matériels, informationnels, monétaires ou décisionnels	Coordination des activités des agents par interaction avec les autres agents
La distribution et l'incomplétude de l'information	un acteur de la SC accède à des informations incomplètes, partagées le long des frontières du système	Les agents possèdent des informations incomplètes, les partages d'informations et / ou des connaissances s'effectuent par échanges de messages
La répartition des tâches	Les tâches des acteurs de la SC peuvent être décomposées et données en responsabilités à d'autres acteurs	Les agents peuvent déléguer des tâches ou en partager pour la résolution de problèmes complexes
L'évolutivité des systèmes	Structures dynamiques, les acteurs s'inscrivent ou quittent la chaîne logistique	Les agents peuvent rejoindre le système et d'autres peuvent être détruits

Tableau 2 : Analogie entre la chaîne logistique et le système multi-agent [Yuan et al., 2002]

5. Plates-formes de développement de systèmes multi agents

Il existe des outils ou plates-formes de développement pour la conception de systèmes reposant sur l'utilisation du paradigme agent. Parmi les plates-formes les plus connues, nous pouvons citer SWARM, JADE, VOYAGER, DIMA, Zeus, MADKIT, etc.

- **SWARM** est une plate-forme de développement qui a une orientation vers le développement d'applications de simulation à événements discrets de systèmes complexes.
- **JADE** (Java Agent Development Framework) est une plate-forme où les agents sont des coquilles auxquelles il faut ajouter des comportements implémentant des services/fonctionnalités ; elle offre une possibilité de communication entre plusieurs plates-formes JADE. Les communications utilisent le standard ACL.
- **VOYAGER** est une plate-forme de développement commerciale. Elle fonctionne en environnement distribué et repose sur le standard CORBA. Son domaine d'application est orienté vers le commerce électronique et les applications WEB.
- **DIMA** (Développement et implémentation de systèmes multi agents) est une plate-forme dans laquelle l'architecture des agents choisie pour le développement de celle-ci est une architecture modulaire (module de supervision, module de communication, module de perception, module de raisonnement).
- **ZEUS** est une plate-forme de développement de SMA générique, personnalisable et peut être augmentée par l'adjonction de nouveaux composants. De par sa généricité, elle peut être utilisée dans la conception d'un grand nombre d'applications mais en retour nécessitera des augmentations assez nombreuses et spécifiques à chaque type d'application.
- **MADKIT** est une plate-forme de développement et d'exécution de SMA fondés sur des critères organisationnels.

5.1 La plateforme JADE

JADE (Java Agent DEvelopment Framework) est un logiciel libre distribué par le CSELT (Centre Studi E Laboratori Telecomunicazioni) de l'université de Parma. Cet environnement de développement d'applications à base d'agents, JADE, est implémenté entièrement en Java. JADE est conforme aux normes et spécifications de FIPA (Foundation For Intelligent Physical Agents) (FIPA, 2002), elle hérite alors son architecture, ses protocoles, ses services et ses mécanismes de transport et de communication. JADE est compatible avec la plupart des configurations matérielles et logiciels (Windows, Unix. L'agent JADE peut être implémenté en Java ou en Jess (Java Expert System Shell) (Jess, 2001). Ce dernier est un système expert qui permet d'implémenter des architectures d'agents à base de règles (Jess, 2001). Les agents JADE communiquent en utilisant le langage ACL (Agent Communication Language) (FIPA, 2000), dans le cas où ces agents sont hétérogènes, ils coopèrent par négociation.

La plateforme JADE peut être répartie sur plusieurs serveurs, une seule application Java pour une machine virtuelle est exécutée (JVM) sur chaque serveur. Chaque JVM est un conteneur d'agents qui fournit un environnement complet pour l'exécution d'agent et permet à plusieurs agents de s'exécuter en parallèle. La plate-forme JADE est aussi basée sur le protocole IIOP conforme à la spécification FIPA (FIPA, 2002) pour connecter plusieurs plates-formes agents.

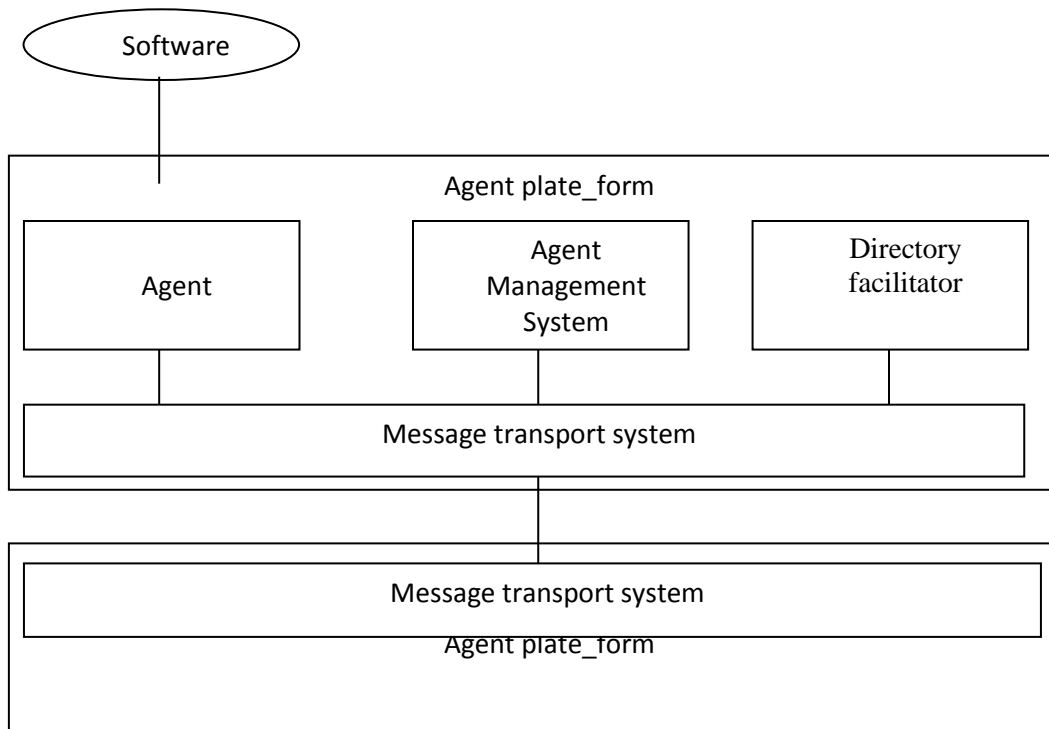


Figure 8 Architecture logiciel de la plateforme JADE.

Le but de JADE est de simplifier le développement des systèmes multi-agents en conformité avec la norme FIPA pour réaliser des systèmes multi-agents inter opérables.

Conclusion

Nous avons montré dans ce chapitre les principes notion des systèmes Multi Agent et défini quelques termes de ce concept et d'autre part, voir comment les agents peuvent avoir une importance à être utilisés dans la gestion d'une SC.

Le chapitre suivant traite la conception et l'implémentation de notre outil de gestion de chaînes logistiques en utilisant les SMA.

Chapitre VI : Conception et implémentation

Introduction :

L'étape de la réalisation du projet est la partie cruciale du développement pour atteindre les objectifs fixés au départ. Lors de cette étape, nous allons préciser les outils utilisés et l'architecture appliquée pour implémenter ce projet. En dernier point, nous allons montrer les résultats finaux de la conception et les études effectuées.

1. Environnement de développement

1.1 Netbeans

Netbeans est un environnement de programmation visuel orienté objet. Il augmente la puissance de l'outil de développement rapide d'application (RAD), comme il comprend plusieurs applications spécialement conçues pour faciliter la productivité.

1.2 MySQL

MySQL est un gestionnaire de base de données relationnelles (SGBDR : Système de gestion de base de données relationnelles) qui permet de décrire, manipuler, traiter, emmagasiner l'ensemble de données formant une base.

2. Langage de programmation utilisé

2.1 Le langage JAVA

Le langage JAVA est un langage de programmation orienté objet qui permet d'implémenter des applications informatiques de haute performance.

2.1 Le langage SQL

Le langage SQL (*StructuredQueryLanguage*) est un langage informatique qui permet d'interroger les bases de données, en d'autres termes, il permet d'effectuer des opérations sur les bases de données telles que les opérations de recherche, adjonction, modifications et suppression des données.

3 Conception de la base de données

3.1 Dictionnaire de données

Le dictionnaire de données est un tableau qui regroupe toutes les données du système d'information, où pour chaque donnée on précise sa désignation, sa longueur, son type et son code d'attribut.

Le dictionnaire des données associé à notre système d'optimisation des coûts dans la chaîne logistique est le suivant :

Entité	Données	Type	Taille	Désignation
Clients	Id_clt	Integer	4	Identifiant du client.
	Nom	Varchar	50	Nom du client ou de la société.
	Tel	Varchar	10	Le numéro de téléphone du client.
	Tel_prt	Varchar	10	Le numéro du téléphone portable du client.
	Fax	Varchar	10	Le numéro du fax du client.
	Mail	Varchar	50	L'adresse électronique du client.
	Site	Varchar	60	L'URL du site internet du client.
	Id_loc	Integer	4	Identifiant de la localisation du client.
	Id_frn	Integer	4	Identifiant du fournisseur.
	Nom	Varchar	50	Nom du fournisseur ou de la société.
	Tel	Varchar	10	Le numéro de téléphone du fournisseur.
	Tel_prt	Varchar	10	Le numéro du téléphone portable du fournisseur.
	Fax	Varchar	10	Le numéro du fax du fournisseur.
	Mail	Varchar	50	L'adresse électronique du client.
	Site	Varchar	60	L'URL du site internet du fournisseur.
	Id_loc	Integer	4	Identifiant de la localisation du fournisseur.
Distributeurs	Id_dst	Integer	4	Identifiant du distributeur.
	Nom	Varchar	50	Nom du distributeur ou de la société.
	Tel	Varchar	10	Le numéro de téléphone du distributeur.
	Tel_prt	Varchar	10	Le numéro du téléphone portable du distributeur.

	Fax	Varchar	10	Le numéro du fax du distributeur.
	Mail	Varchar	50	L'adresse électronique du distributeur.
	Site	Varchar	60	L'URL du site internet du distributeur.
	Id_loc	Integer	4	Identifiant de la localisation du distributeur.
Produits	Id_prd	Integer	4	Identifiant du produit.
	Libellé	Varchar	50	Le nom du produit.
	Désignation	Varchar	Text	La désignation du produit.
	Longueur	Double	4.2	La longueur du produit.
	Largeur	Double	4.2	La largeur du produit.
	Hauteur	Double	4.2	La hauteur du produit.
	Poids	Double	4.2	Le poids du produit.
	Temps_prd	Time	-	Le temps de la production du produit.
	Prix	Double	10.2	Le prix du produit.
Matières	Id_mp	Integer	4	Identifiant de la matière première.
	Libellé	Varchar	50	Le nom de la matière première.
	Désignation	Varchar	Text	La désignation de la matière première.
	Longueur	Double	4.2	La longueur de la matière première.
	Largeur	Double	4.2	La largeur de la matière première.
	Hauteur	Double	4.2	La hauteur de la matière première.
	Poids	Double	4.2	Le poids de la matière première.
	Temps_prd	Time	-	Le temps de la production de la matière première.
	Prix	Double	10.2	Le prix de la matière première.
Entrepôts	Id_entrp	Integer	4	Identifiant de l'entrepôt.
	Libellé	Varchar	50	Le nom de l'entrepôt.
	Capacité	Double	4.2	La capacité de stockage en mètres cube de l'entrepôt.
	Id_loc	Integer	4	Identifiant de la localisation de l'entrepôt.
Véhicules	Id_vec	Integer	4	Identifiant du véhicule.
	Libellé	Varchar	50	Nom de la marque du véhicule.
	Capacité	Double	4.2	La capacité de la charge du véhicule.
	Puissance	Double	4.2	La puissance de véhicule.
Commandes	Id_cmd	Integer	4	Identifiant de la commande.
	Désignation	Varchar	Text	Désignation de la commande.
	Date_cmd	Date	-	Date de la commande.

	Date_liv	Date	-	Date de livraison de la commande.
--	----------	------	---	-----------------------------------

3.2 Modèle conceptuel des données

Le modèle conceptuel des données (MCD) est un diagramme qui permet de schématiser la structure et l'organisation logique des données du système étudié. Le MCD relative à ce système d'optimisation est le suivant :

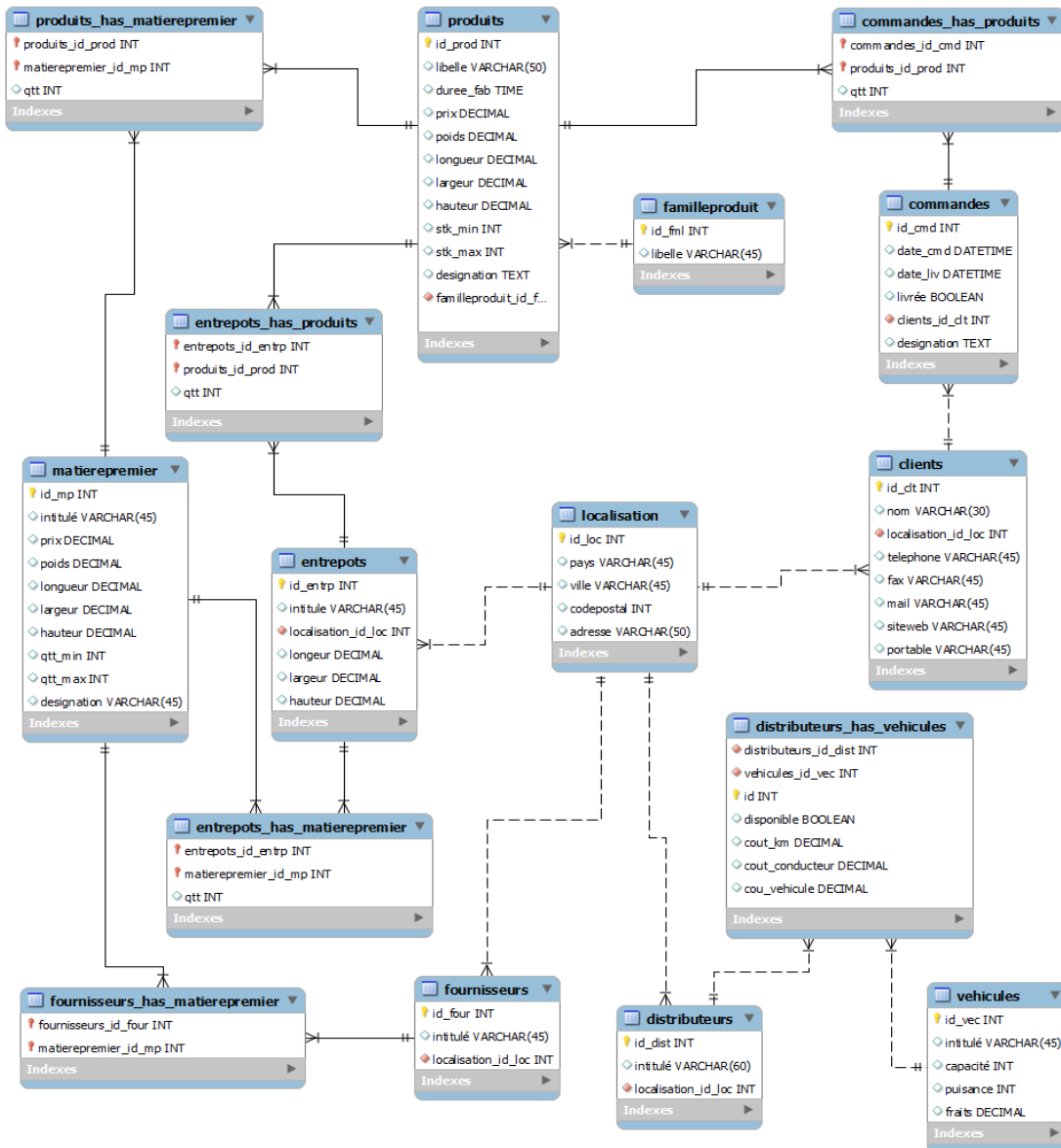


Figure 9 Modèle conceptuel des données

3.3 Modèle logique de données

Le modèle logique de données (MLD) est une représentation des données dans un formalisme compris par la machine, en d'autres termes c'est la traduction du modèle conceptuel de données qui ne prend en considération aucune contrainte organisationnelle, à un autre modèle dont le système l'exige. Cette traduction se fait comme suit :

- Tout objet du MCD se convertit en table dans le MLD.
- Toutes les propriétés de l'objet deviennent les attributs de la table.
- L'identifiant de l'objet devient la clé de la table.

D'où le MLD qui convient au MCD précédent est le suivant :

Clients (**id_clt**, Nom, Tel, Tel_prt, Fax, Mail, Site, id_loc*).

Fournisseurs (**id_frn**, Nom, Tel, Tel_prt, Fax, Mail, Site, id_loc*).

Distributeurs (**id_dst**, Nom, Tel, Tel_prt, Fax, Mail, Site, id_loc*).

Produits (**id_prd**, Libellé, Désignation, Longueur, Largeur, Hauteur, Poids, Temps_prd, Prix).

Matières (**id_mp**, Libellé, Désignation, Longueur, Largeur, Hauteur, Poids, Temps_prd, Prix).

Commandes (**id_cmd**, Désignation, Date_cmd, Date_liv, id_clt*).

Entrepôts (**id_entrp**, Libellé, Capacité, id_loc*).

Véhicules (**id_vec**, Libellé, Capacité, Puissance).

Commandes_Produits (**id_cmd, id_prd, qtt**).

Produits_Matières (**id_prd, id mp, qtt**).

Fournisseurs_Matières (**id_frn, id mp, prix**).

Distributeurs_Véhicules (**id, id_dist***, id_vec*, disponible).

4 Fonctionnement globale

Cette application permet à son utilisateur (responsable de l'affectation de projets) de s'occuper de l'activité d'optimisation des coûts dans sa chaîne logistique. Elle lui facilite cette tâche d'optimisation en lui offrant un système d'aide à la décision qui permet de la gérer d'une manière intelligente, robuste, efficace et rapide.

Pour utiliser cette application, l'utilisateur doit effectuer les tâches suivantes :

- Saisir : la liste des clients, fournisseurs, distributeurs, produits, et matières premières.
- Entrer les commandes des clients.

Au final il peut lancer l'opération d'optimisation des coûts. Cette opération permet à l'utilisateur de décider quelle commande doit être livrée, quel fournisseur et quel distributeur doivent être contactés pour les coûts d'achats des matières premières (fournie par le fournisseur choisi) et

les coûts de transport doivent être optimal. L'opération d'optimisation est réalisée par l'algorithme de colonie de fourmis (OCF).

5 Schéma de fonctionnement de l'application

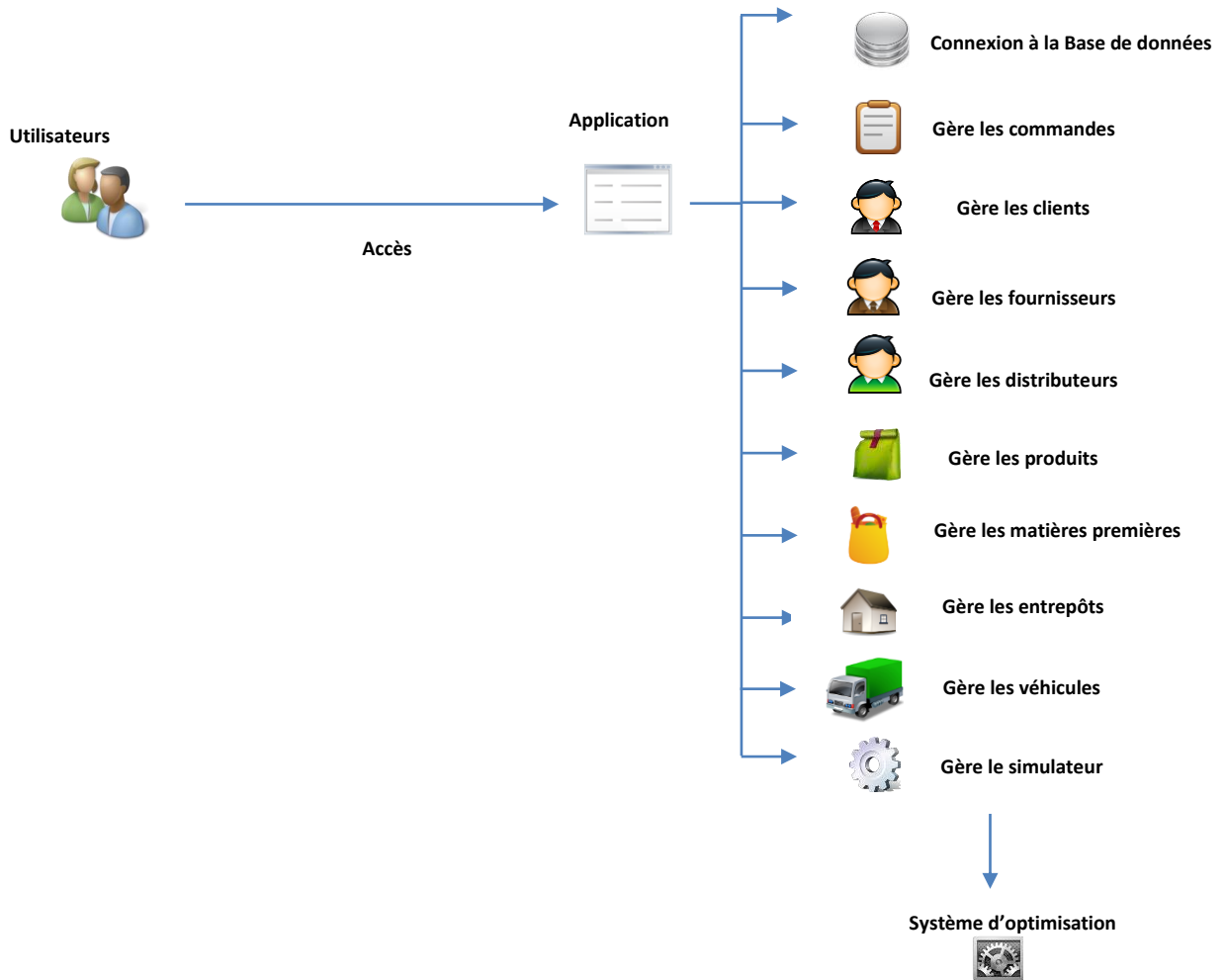


Figure 10 Schéma de fonctionnement de l'application

6 Schéma de fonctionnement du système logistique

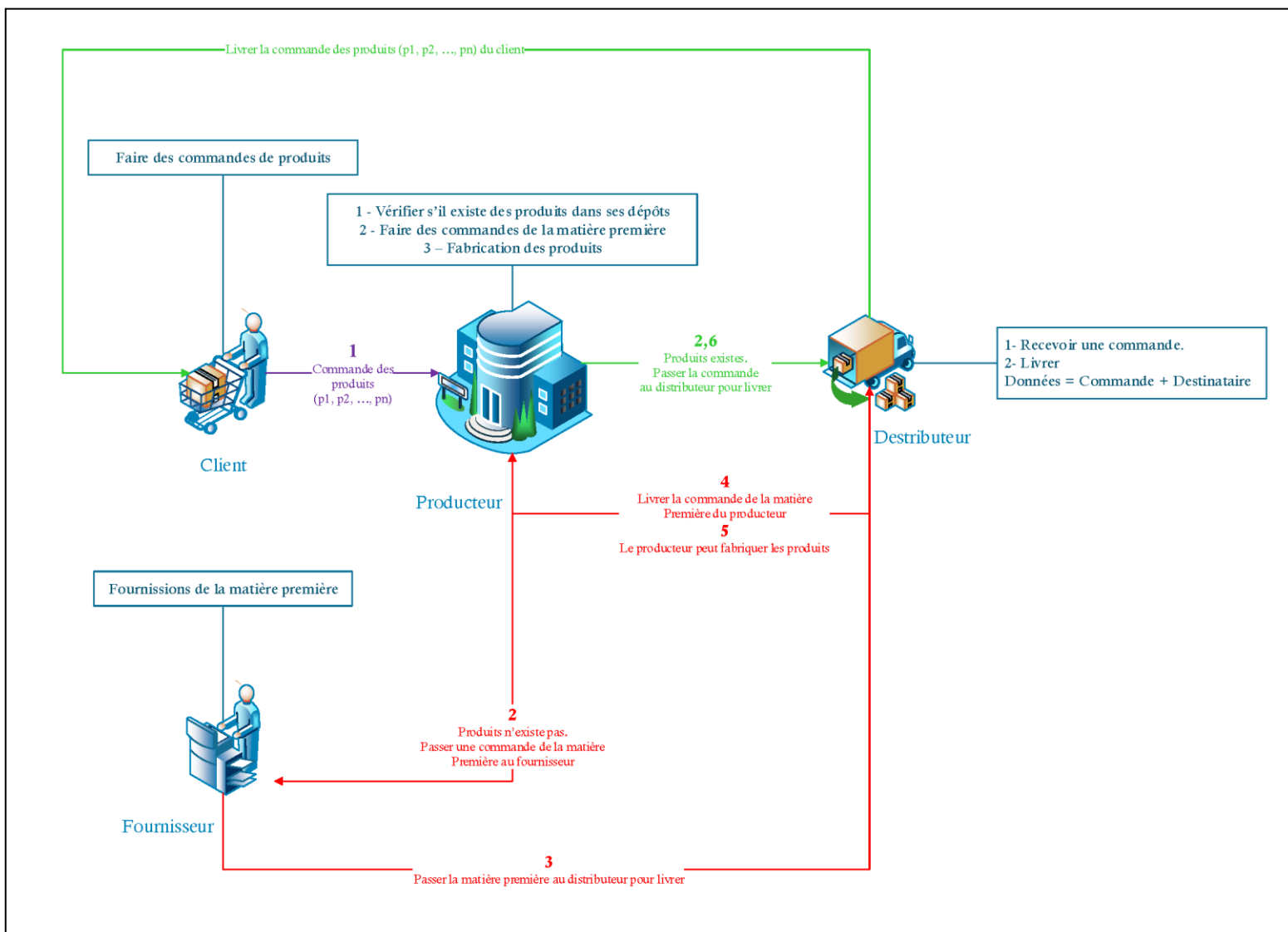


Figure 11 Schéma de fonctionnement du système logistique.

1. Le client fait une commande de produits.
2. Le producteur vérifie son stock de produits :
 - 2.1 Si les produits existent alors passer la commande du client au distributeur pour livrer.
 - 2.2 Si les produits n'existent pas alors le producteur vérifie son stock de la matière première nécessaire pour la production.
 - 2.2.1 Si la matière première existe alors le producteur produit les produits et faire l'étape 2.1.
 - 2.2.2 Si la matière première n'existe pas alors le producteur fait une commande de la matière première voulu au fournisseur.
 - 2.2.2.1 Quand le fournisseur reçoit la commande de la matière première issue de producteur il passe la matière première de commandée au distributeur pour livrer au producteur.

2.2.2.2 Quand le producteur reçoit la matière première il fait l'étape 2.2.1

7. Modélisation du système logistique

La chaîne logistique contient des acteurs qui coopèrent dans la réalisation d'activités ayant pour objectif la satisfaction des clients en termes de coûts, de qualité, de délai et de services associés au produit. De cet environnement, la modélisation adéquate de système est la modélisation en agent où chacun simule les activités d'un agent réel.

7.1 Les agents

Il existe quatre agents principaux dans la chaîne logistique : Client, Producteur, Fournisseur et Distributeur.

7.1.1 Agent client : L'agent client a comme objectifs :

- Envoie des commandes de produit fini à l'agent producteur.
- Recevoir sa commande depuis l'agent distributeur (cas d'une livraison indirecte).
- Recevoir sa commande depuis l'agent producteur (cas d'une livraison directe).

7.1.2 Agent producteur : L'agent producteur assure les tâches suivantes :

- Recevoir les commandes des clients.
- Traitement des commandes.
- Envoie des demandes de livraison à l'agent distributeur.
- Envoie des commandes de la matière première à l'agent fournisseur.
- Production (transformation de la matière première en produits finie).

7.1.3 Agent Fournisseur : L'agent fournisseur a comme tâches :

- Recevoir des commandes de matière première issue du l'agent producteur.
- Envoie des demandes de livraison à l'agent distributeur.

7.1.4 Agent distributeur : En générale, l'agent distributeur joue le rôle d'un lieur entre l'agent producteur et l'agent fournisseur, et entre l'agent producteur et l'agent client. Il assure les objectifs suivants :

- Réception des demandes de livraison.
- Transporter les commandes aux clients.

7.2 Diagramme de séquence

Nous précisons que nous avons élaboré nos diagrammes UML en utilisant l’outil StarUML, open source flexible et extensible.

Le diagramme de séquence est une représentation graphique de la chronologie des échanges de messages avec le système ou au sein du système. Ces principes :

- Interactions entre objets dans une séquence temporelle.
- Aspect chronologique ne rendant pas compte explicitement du contexte.
- Permet de bien montrer l’opération à exécuter dans une interaction.
- Description de scénarios typiques et des exceptions.

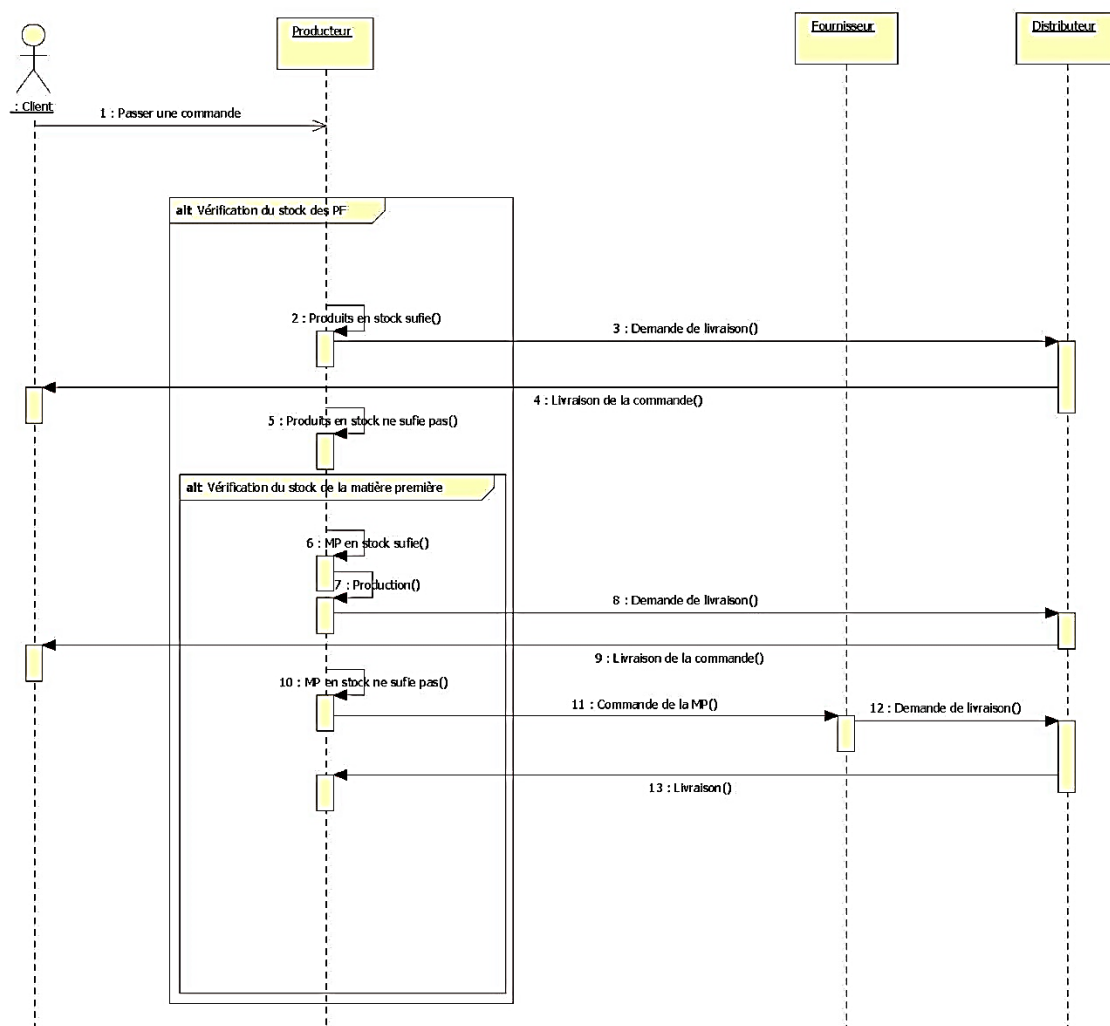


Figure 12 Diagramme de séquence

7.3 Diagramme de classes

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations. En d’autres termes, il est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce

diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

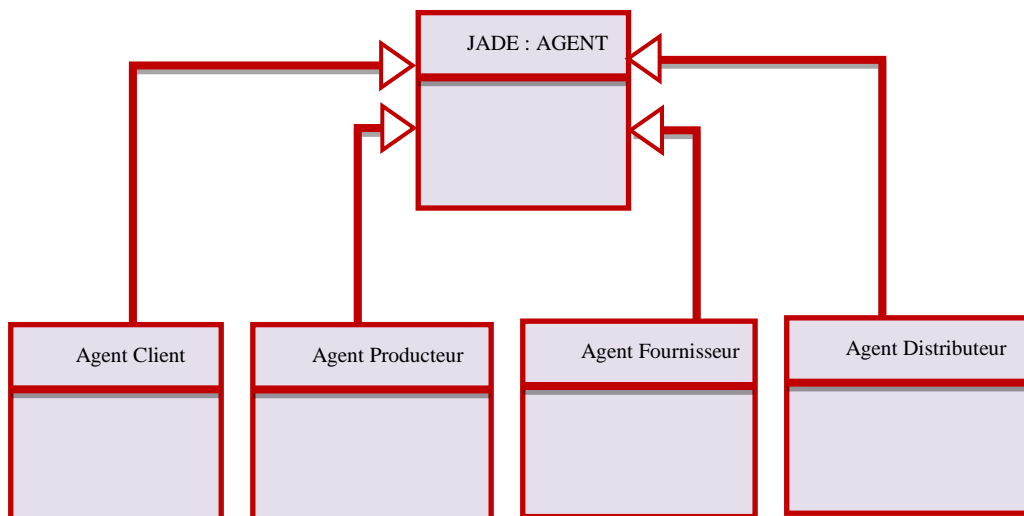


Figure 13 Diagramme de classes

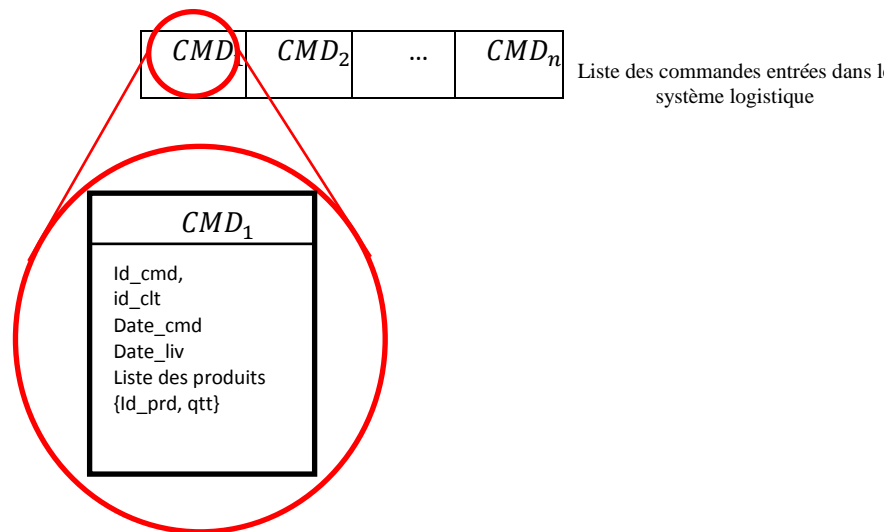
8. Modélisation des données du système logistique

Chaque jour, le système logistique (producteur) a une liste de n commandes en attente pour être livrées. Une commande est un ensemble de l différents produits, et chaque produit à une quantité. Pour que le producteur puisse de traiter chaque commande arrivée, il doit tout d'abord lancer des activités de production. Ces activités sont des ensembles d'opération de transformation de la matière première en produits finie. Cela pose au producteur des problèmes de décision .c.-à-d. le producteur doit trouver un fournisseur de la matière première et un distributeur dont le coût d'achat et le coût de transport sera optimal. Alors l'objectif de l'algorithme d'OCF est d'aider le producteur à trouver pour chaque commande un fournisseur et un distributeur qui optimisent ses coûts d'achat et celui de transport.

8.1 Structure de données

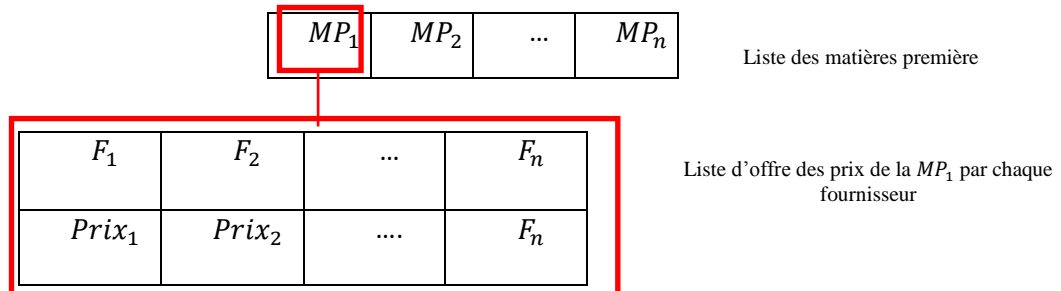
Le système logistique possède tous les informations sur la chaîne logistique et ces acteurs sous forme des enregistrements stockés dans la base de données. L'ensemble de ces informations forme les données du problème à optimiser. Pour cela le système transforme ces informations en structure de données bien définie comme suit :

8.1.1 Structure des commandes



Liste des commandes entrées dans le système logistique

8.1.2 Structure des offres de vente de la MP



Liste des matières première

Liste d'offre des prix de la MP_1 par chaque fournisseur

8.1.3 Structure des offres de transport

	D_1	...	D_n
F_1	PT_{11}	...	PT_{1n}
...
F_m	PT_{m1}	...	PT_{mn}

Matriced'offredes prix de transport (PT_{ij}) entre le $Fournisseur_i$ et le Producteur en utilisant le $Distributeur_j$

9. Modélisation de l'algorithme OCF

Les algorithmes d'OCF sont très intéressants pour résoudre les problèmes de recherche du coût minimum d'un graphe, particulièrement quand les coûts de raccordement peuvent changer avec le temps, c.-à-d., quand les problèmes sont dynamiques. Le cadre général de l'algorithme

peut être décrit comme suit : Considérer un problème qui se compose de trouver le chemin avec le coût minimum sur un graphe, pour une certaine fourmi k placée dans le nœud u .

Alors pour optimiser les coûts d'achats et les coûts de transport, il faut formaliser le problème d'optimisation sous forme d'un graphe.

9.1 Graphe du problème

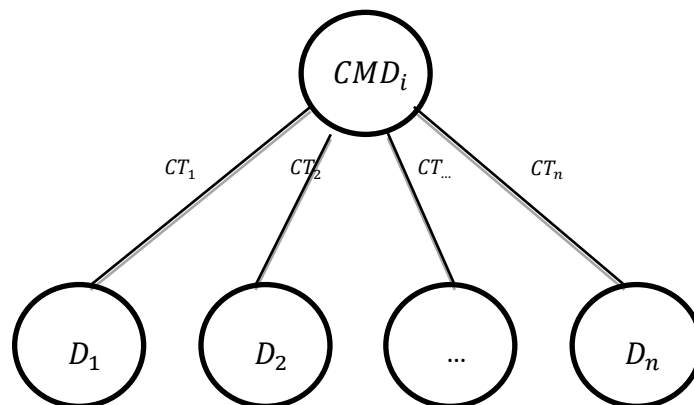
Le graphe du problème est un graphe qui résume tous les informations nécessaire à l'optimisation. En d'autres termes c'est la structure principale de données dont l'algorithme d'OCF se déroulera.

9.1.1 Structuration de graphe du problème

Supposant que le système logistique veut traiter une commande i qui contient n produits. Le traitement de cette commande peut être effectué dans l'un de ces cas suivant :

Cas 1 : Quantité de produit fini en stocke est suffi pour livrer la commande.

Dans ce cas le producteur doit choisir un distributeur qui permet de livrer la commande au client avec un coût moins cher. Ce cas sera traduit par un graphe qui illustre le problème d'optimisation.



CMD_i : La commande i à traiter.

CT_i : Coût de transport fixé par le distributeur D_i pour livrer la commande du producteur vers le client concerné.

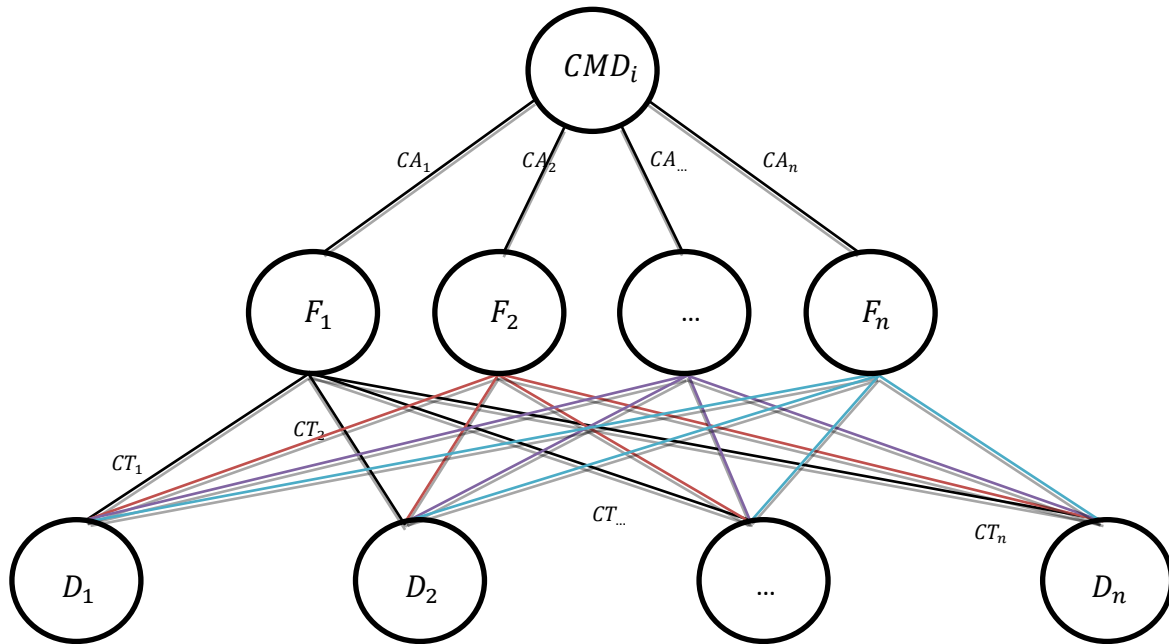
D_i : Un distributeur.

Cas 2 : Quantité de produit fini en stocke est ne suffit pas pour livrer la commande, mais la quantité de la matière première suffit pour fabriquer des nouveaux produits.

Dans ce cas le producteur va lancer une opération de la production pour qu'il puisse livrer la commande en attente de livraison. Après la production, la quantité des produits finie sera suffi pour livrer la commande, dans cette étape en revient au cas 1. Dans ce cas la livraison de la commande peut être retardée.

Cas 3: Quantité de produit fini en stocke est ne suffit pas pour livrer la commande, et la quantité de la matière première ne suffit pas pour fabriquer des nouveaux produits.

Dans ce cas le producteur va passer une commande de matière première au fournisseur. Après l'arrivée de la commande demandée, le producteur sera trouvé dans le cas 2 (matière première suffit pour la production). Ca cas sera traduit par un graphe qui illustre le problème d'optimisation :



CMD_i : La commande i à traiter.

CT_i : Coût de transport fixé par le distributeur D_i pour livrer la commande du fournisseur vers le producteur.

D_i : Un distributeur.

F_i : Un Fournisseur.

Ce graphe illustre le processus d'achat de la matière première.

9.2 Résolution par l'algorithme d'OCF

9.2.1 Paramétrages de l'algorithme :

L'algorithme d'OCF a besoin de trois paramètres :

Nb_Frm : C'est le nombre de fourmi dans la colonie.

Qtt_Phr : C'est la quantité de la phéromone à poser dans chaque itinéraire trouvé par la fourmi.

Ev_Coef : C'est le coefficient d'évaporation de phéromone posée.

9.2.2 Résolution

Prenant le graphe du problème du cas 3 précédent.

D'après ce graphe on constate que pour traiter la commande i , le système logistique doit contacter le système fournisseur (approvisionnement) et le système de distribution (traitement par deux niveaux). Le rôle de la colonie dans ce cas est de trouver un $Fournisseur_i$ et un $Distributeur_j$ pour acheter la matière première nécessaire à la production avec un coût optimal.

La démarche de résolution est comme suit:

- 1- La colonie de Nb_Frm fourmis sera placée dans la racine du graphe.
- 2- Initialiser chaque arête du graphe avec une quantité de phéromone fixé au départ.
- 3- La colonie propage dans le premier niveau du graphe (choisir un fournisseur).

Dans ce cas chaque fourmi choisit un fournisseur d'une manière aléatoire. Une fois qu'elle a choiit le fournisseur, elle propagera une autre fois dans le deuxième niveau pour qu'elle choisira un distributeur d'une manière aléatoire, et ici la fourmi a trouvé une solution. Cette solution est la somme total d'achat et de transport nécessaire pour acheter la matière première souhaitée, le fournisseur et le distributeur choisit. Quand la fourmi trouve la solution, elle enrichira la connaissance collective de la colonie en posant une quantité de la phéromone sur les arêtes qu'elle a passées par eux. Dans ce moment il passe un évènement qui s'appelle l'évaporation de phéromone, dans cet évènement une quantité de phéromone égale à Ev_Coef sera perdu de la quantité de phéromone qui se trouve sur les arêtes du graphe.

Les fourmis considèrent les coûts qui étiquettent les arêtes du graphe comme distance.

Donc toute arête étiquette par un coût minimal à une quantité de phéromone plus élevé.

- 4- A la fin de l'algorithme, on analyse chaque solution trouvée par chaque fourmi. On prend la solution de la fourmi qui a trouvé un coût optimal.
- 5- Le même principe est appliqué sur les autres cas.

10. Implémentation de l'outil

L'outil que nous avons implémenté est une application qui contient un ensemble d'interface graphique (GUI : Graphical User Interface) qui permet aux utilisateurs de communiquer avec l'application et sa base de données. Ces interfaces sont de type MDI (Multiple Document Interface). En d'autres termes, il est une application multifenêtre où chaque composante gère une partie de l'application.

10.1 Les interfaces de l'application

10.1.1 Fenêtre principal

La fenêtre principale de l'application donne l'accès à toutes les fonctionnalités de l'application.

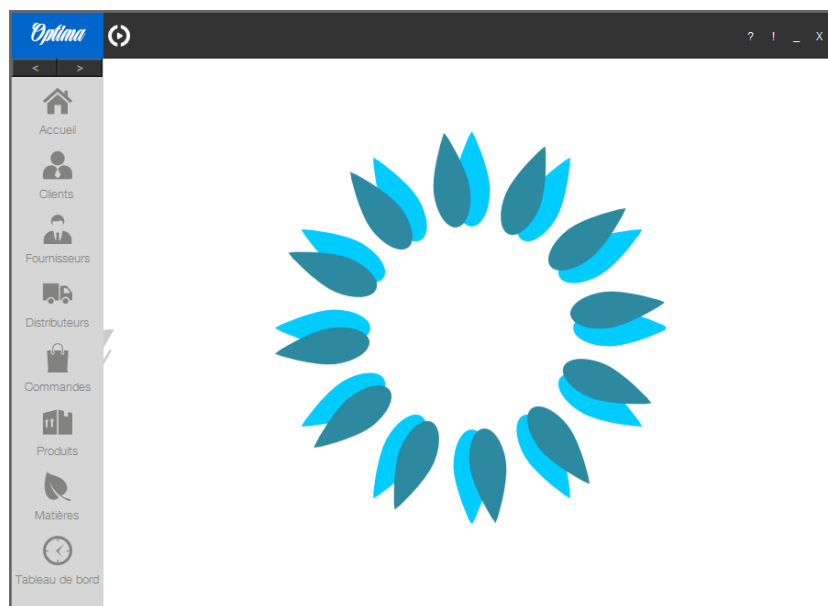


Figure 14 Fenêtre principal

10.1.2 Fenêtre des clients

La fenêtre des clients permet à l'utilisateur de saisir sa liste des clients en entrant tous les informations nécessaires pour la gestion.

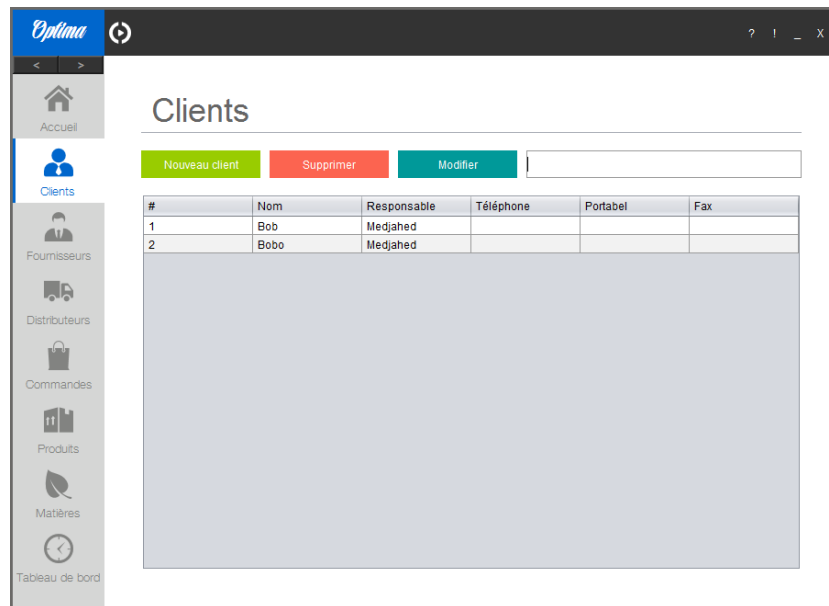


Figure 15 Fenêtre des clients

Cette fenêtre inclut un formulaire pour ajouter des nouveaux clients, elle inclut aussi d'autres fonctionnalités telles que la modification, suppression et la recherche d'un client.

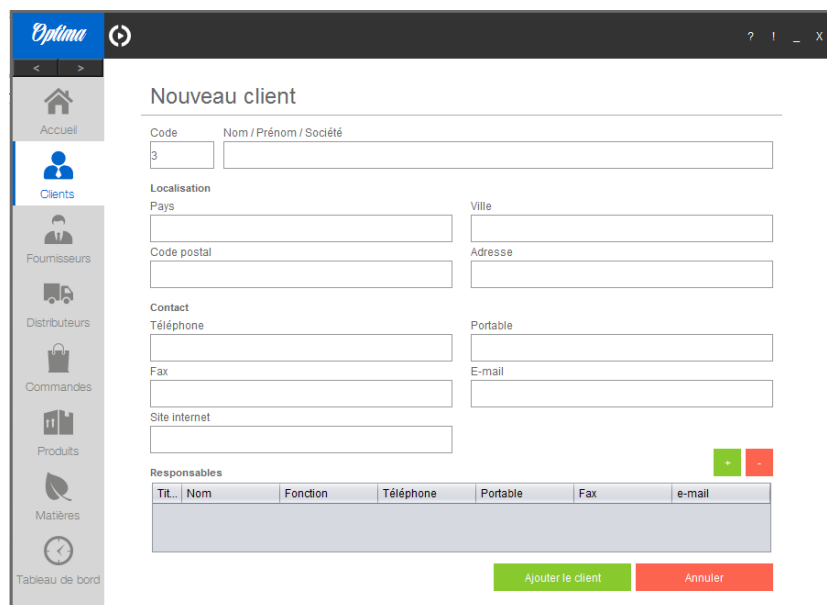


Figure 16 Fenêtre nouveau client

10.1.3 Fenêtre des produits

La fenêtre des produits c'est point d'accès à la liste des produits. Cette fenêtre inclut toutes les fonctionnalités nécessaires pour ajouter, modifier, supprimer ou rechercher un produit.

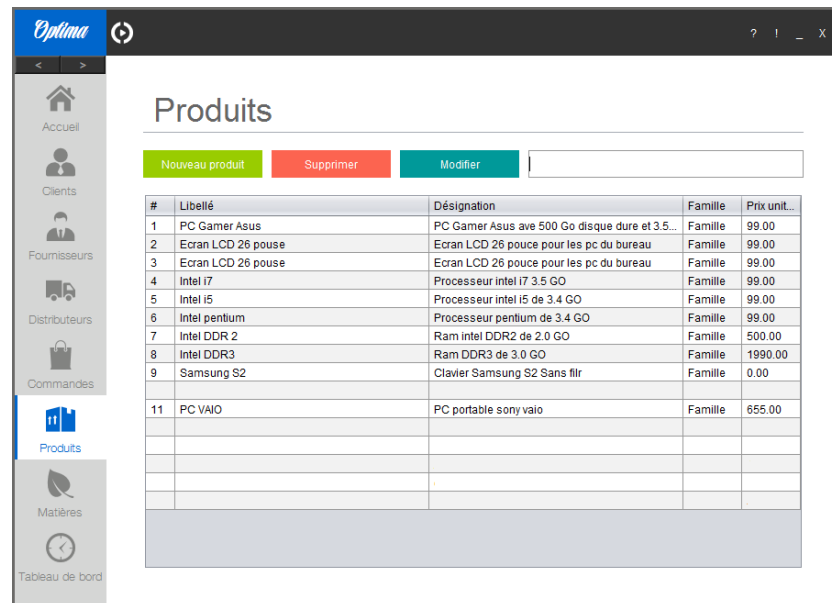


Figure 17 Fenêtre des produits

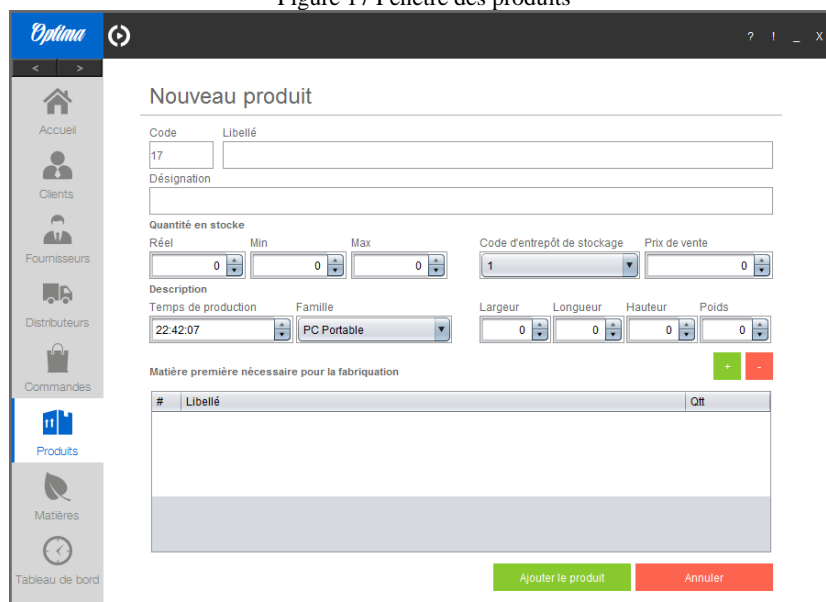


Figure 18 Nouveau produit

10.1.4 Fenêtre des matières premières

Cette fenêtre donne un accès à la liste des matières première. Elle permet d'ajouter, modifier, supprimer ou rechercher une matière première.

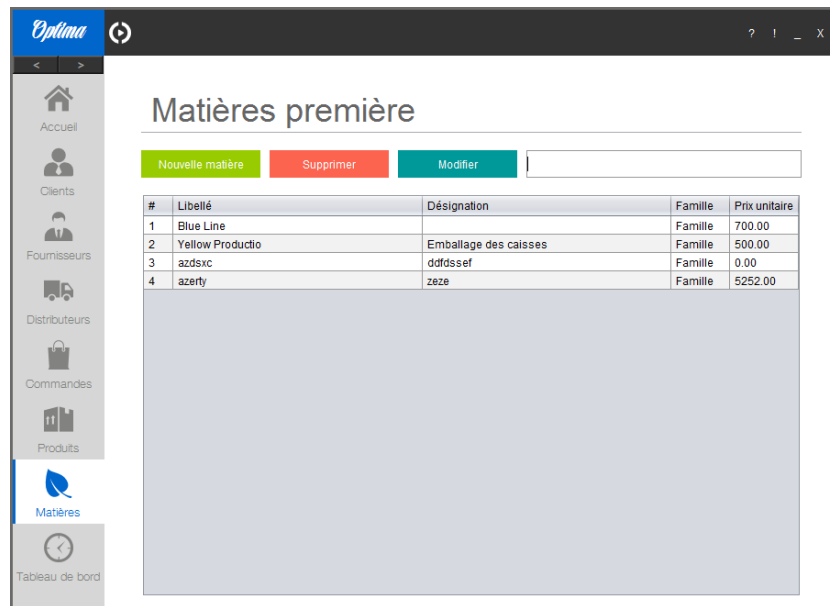


Figure 19 Fenêtre des matières premières



Figure 20 Nouvelle matière première

10.1.5 Fenêtre tableau de bord

Cette fenêtre donne un résumé graphique sur tous les données stockées, ainsi que les dernières nouvelles opérations effectuée.

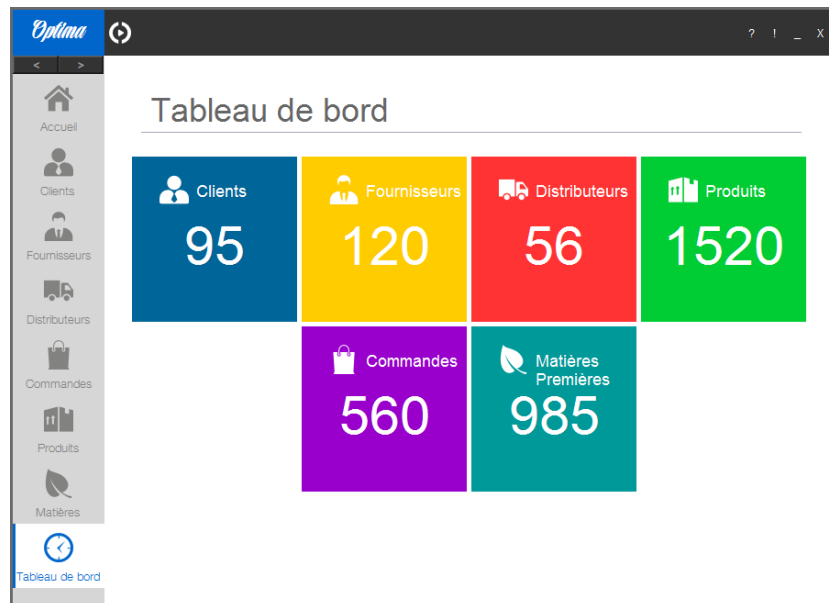


Figure 21 Fenêtre tableau de bord

10.1.6 Fenêtre simulateur

La fenêtre simulateur donne un accès à l'interface de configuration et de lancement du simulateur. Ce simulateur c'est un programme qui mixe deux systèmes multi agents (SMA). Le premier SMA simule les activités réelles des acteurs de la chaîne logistique, le deuxième système simule le comportement de la colonie de fourmi, ce dernier implémente l'algorithme d'OCF.

Cet outil contient d'autres fenêtre telle que :

- La fenêtre des distributeurs : Permet d'ajouter des fournisseurs et ces véhicules, et de mettre à jour la liste.
- La fenêtre des distributeurs : Donne un accès à la liste des distributeurs et ces listes de véhicules. Elle permet la mise à jour de la liste.
- La fenêtre des entrepôts de stockage et la fenêtre des véhicules.

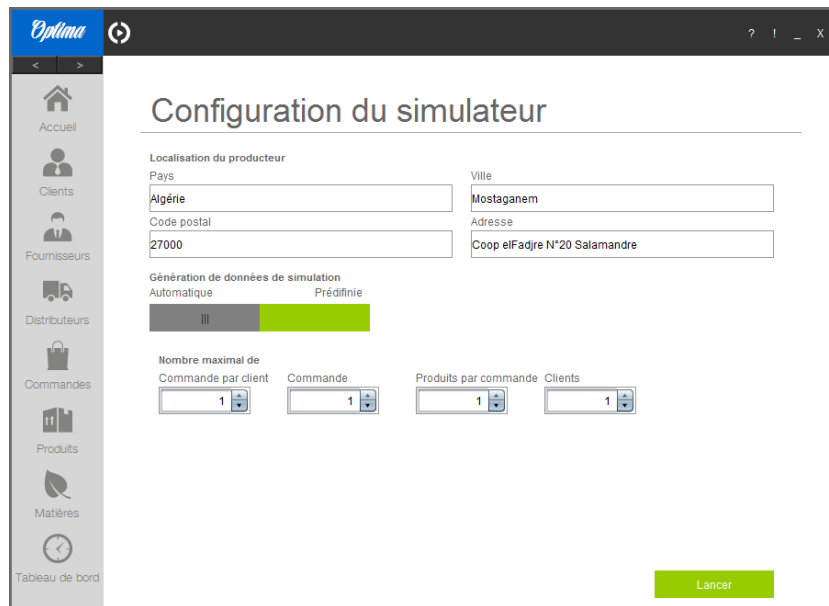


Figure 22 Fenêtre simulateur.

10.1.7 Fenêtre des résultats

La fenêtre des résultats donne un affichage global du résultat donnée par le simulateur. Elle permet à l'utilisateur de consulté la décision qu'il doit appliquer pour optimiser ses coûts logistique.

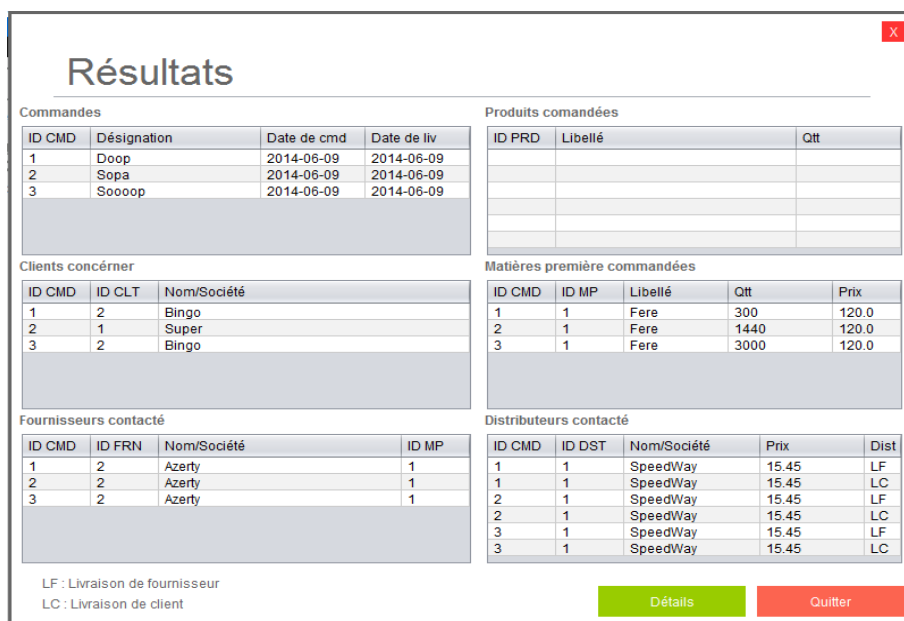


Figure 23 Fenêtre des résultats

Conclusion :

Dans ce chapitre, nous avons présenté la conception de la base de données, les modèles UML de notre système ainsi que le déroulement de notre application.

Conclusion générale

Ce document a présenté une technique de gestion des activités opérationnelles d'une SC générique, incluant les fournisseurs, les acteurs de logistique et de distribution. La méthodologie consiste à modéliser les acteurs comme un problème d'optimisation combinatoire et optimiser chaque problème en utilisant l'algorithme d'optimisation par colonie de fourmis. Cet algorithme utilise une matrice de phéromone pour garder une trace de l'information au cours de la procédure d'optimisation. Il est alors possible d'échanger des informations entre les processus d'optimisation fonctionnant en parallèle et réaliser un mécanisme de coopération. L'objectif principal a été de livrer aux clients, dans les temps, des produits de qualité au meilleur prix.

Très peu d'auteurs décrivent les méthodes et les outils qu'ils utilisent et qu'ils emploient pour optimiser la logistique ou la fonction logistique considérée. Il s'agit d'optimiser la gestion des flux au niveau global, de l'approvisionnement en matières premières au consommateur final. Il s'agit de coordonner tous les processus pour atteindre le niveau de performance qui convient à l'entreprise, la logistique, l'approvisionnement, la production, les stocks, la distribution, le système d'information.

Enfin ce travail nous a initié à l'optimisation distribuée des coûts au sein des SC, et aussi nous a donné une idée qui a contribué à comprendre le fonctionnement d'une gestion des chaînes logistiques en utilisant les méthodes d'optimisation. Cela a enrichi nos compétences dans les différents domaines tels que la gestion des chaînes logistiques et l'application des méthodes d'optimisation. Ce projet nous a exigé un investissement personnel et professionnel qui nous permettra dans un prochain travail à réaliser une application mettant en pratique les méthodes étudiées dans cette contribution, en effet, nous comptons nous inspirer de cette contribution pour présenter une nouvelle approche pour la modélisation par colonies de fourmis.

Références bibliographique

- **[Aitsilarbi, 2013]** AIT SI LARBI El Yasmine, BEKRAR Abdelghani, TRENTESAUX Damien, BELDJILALI Bouziane : Supplychain management using multi agent systems in agri foodindustry. SOHOMA 2013 : Workshop on Service Orientation in Holonic and Multi Agent Manufacturing Control. Valenciennes, Juin 2013.
- **[Bellman 57]** :R. E. Bellman. Dynamic programming. Princeton UniversityPress, 1957.
- **[Bertsekas, D. (1982)]** :Distributed dynamic programming. IEEE Transactions on Automatic Control 27 (3) , 610–616.
- **[BRIOT 01]**:J. P. Briot et Y. Demazeau « Agent et systèmes multiagents » chez Hermès en 2001
- **[C.A. Silva, J. S. (2005)]** :Soft computing optimization methods applied to logistic processes,. International Journal of ApproximatedReasoning40 , 280 - 301.
- **[Colorni, M. Dorigo, et V. Maniezzo]** : Distributed optimization by ant colonies. invarela, f. and bourgine, p., editors, proceedings of ECAL'91. First EuropeanConference on Artificial Life, Paris, France, pages 134–142,1992a. (Cité page 22.)
- **[Costa et A. Hertz. Ants can color graphs]** :Journal of the Operational Research Society, 48 :295–305, 1997. (Cité page 22.)
- **[Cristopher, M. (1992)]** :Logistics & Supply Chain Management: Creating Value-Adding Networks. UK: Pearson.
- **[Dantzig 63]** :G. Dantzig. Linear programming and extensions. Princeton UniversityPress, 1963.
- **[Demazeau et A. R. Costa]** :Populations and organizations in open multiagent systems. In Proceedings of the First National Symposium on Parallel and Distributed AI (PDAI'96), Hyderabad, 1996.
- **[Demazeau et Müller, 1990]** :Demazeau Y., et Müller J.P, 1990. Decentralized Artificial Intelligence. Ed. North-Holland.
- **[D'Inverno et al., 1997]** :D'Inverno M., Kinny D., Luck M. et Wooldridge M., 1997. A Formal Specification of dMARS. In Proceedings of the Fourth International Workshop on Agent Theories, Architectures, and Languages, ATAL'97, Novembre.

- **[Dorigo, A. Colorni, et V. Maniezzo. The ant system]** Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26, No.1 :29–41, 1996. (Cité pages 22 et 23.)
- **[Dudzinski 87]** : K. Dudzinski& S. Walukiewicz. Exact methods for the knapsack problem and its generalizations. European Journal of OperationalResearch, vol. 28, no. 1, pages 3_21, 1987.
- **[Dudzinski 87]** :K. Dudzinski& S. Walukiewicz. Exact methods for the knapsack problem and its generalizations. European Journal of OperationalResearch, vol. 28, no. 1, pages 3_21, 1987.
- **[Ferber J., Gutknecht O., Michel F., From Agents toOrganizations]:** an Organizational View of MultiAgent Systems, in Agent-Oriented Software Engineering (AOSE) IV, P. Giorgini, Jörg Müller, James Odell, eds, Melbourne, July 2003, LNCS 2935, pp. 214-230, 2004
- **[Fischer et al., 1995a]** : Fischer K., Müller J.P et Pischel M., 1995a. Unifying control in a layered agent architecture” IJCAI95, Agent Theory, Architecture and Language Workshop 95, pp. 240-252.
- **[Fischer et al., 1995b]** : Fischer K., Müller J.P et Pischel M., 1995b. Cooperative transportation scheduling: an application domain for DAI. Journal of Applied Artificial Intelligence (Special Issue on Intelligent Agents – Eds. M. J. Wooldridge & N. R. Jennings).
- **[Hart 68]** : P. E. Hart, N. J. Nilsson & B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pages 100_107, 1968.
- **[Howard 60]** : R.A. Howard. Dynamic programming and markov process. The MIT Press, 1960.
- **[Huhns, 1987]** : Huhns M.N, 1987. DistributedArtificial Intelligence. Ed. Pitman.
- **[Kantorovich 60]** : L. V. Kantorovich. Mathematical Methods of Organizing and Planning Production. Management Science, vol. 6, no. 4, pages 366_422, 1960.
- **[Kohl 95]** :N. Kohl. Exact methods for time constrained routing and related scheduling problems. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard PetersensPlads, Building 321, DK-2800 Kgs. Lyngby, 1995.[Howard 60] R.A. Howard. Dynamic programming and markov process. The MIT Press, 1960.

- **[Land 60]** :A. H. Land & A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, vol. 28, no. 3, pages 497_520, 1960.
- **[MENT & AL 2001]** : Mentzer J.T., Dewitt W., Keebler J.S., Min S., Nix N.W., Smith C.D., Zacharia Z.G., *Journal of business logistics Management*. Vol 22 n°2, 2001.
- **[M. Dorigo et L. M. Gambardella. Ant colony system, 1997]**: A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1) :53–66, 1997. (Cité page 24.)
- **[M. Gambardella, E. D. Taillard, et M. Dorigo]** : Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50(2) :167–176, 1999. (Cité page 22.)
- **[O’Hare et Jennings, 1994]** :O’Hare G.M.P et Jennings N.R., 1994. *Foundation of distributed Artificial Intelligence*. John Wiley & Sons, Inc., New York, USA.
- **[SIMC& al, 2007]** : David Simchi-Levi, Philip Kaminsky and Edith Simchi-Levi, McGraw Hill, Extraittiré de l’ouvrage « *Designing and Managing the Supply Chain* »pour *Physical Supply Chains*, 3^{ème} édition, 2007.
- **[Ulrich, 1998]** : Ulrich H., 1998. A simulation platform for multiagent systems in logistics. *Operations ResearchProceedings*, pp 261-268.
- **[Wolf 1999]** :D. Wolf, P. Keblinski, S. R. Phillpot& J. Eggebrecht. Exact method for the simulation of Coulombic systems by spherically truncated, pairwise r summation. *The Journal of chemicalphysics*, vol. 110, no. 17, pages 8254_8283, 1999.
- **[Wooldridge , 1999]** : Wooldridge M.J, 1999. *Intelligent Agents*. Chapitre 1 dans *Multiagent Systems: a modern approach to distributed modern approach to artificial intelligence*, Gerhard Weiss edition, The MIT Press, Cambridge, Massachusetts, London, Angleterre.
- **[Yuan et al., 2002]** : Yuan Y., Liang T.P et Zhang J.J, 2002. Using agent technology to support supply chain management: potentials and challenges. Mc Master University.