

# Une Approche multi-agents pour la modélisation et l'optimisation des Systèmes de gestion de transport maritime

Elfazziki A. \*, Nejeoui A.\*\* , Sadgal M. \*

Dépt. Informatique, Faculté des Sciences Semlalia, Bd Pr. My Abdellah B.P 2390  
Marrakech 40000 Maroc

\* Professeur de l'enseignement supérieur au dépt. Informatique, faculté des Sciences Semlalia (FSSM)  
Université Cadi Ayyad Marrakech

\*\* Etudiant en thèse dépt. Informatique FSSM  
{a.elfazziki, a.nejeoui, m.sadgal}@ucam.ac.ma

## 1. Introduction

L'avènement du conteneur a totalement changé le monde du transport maritime. En effet, en très peu de temps, les compagnies ont changé de méthode de travail et de matériel pour investir dans de nouveaux navires coûteux et dans des flottes de porte-conteneurs. La tendance actuelle est à la concentration, une concentration financière qui se traduit par des fusions-absorptions ou des prises de contrôle et une concentration technique pour éviter une surcapacité des porte-conteneurs. Les alliances et fusions entre les acteurs du transport maritime sont induites par le phénomène de globalisation. Une compagnie à qui l'on demande une prestation de transport maritime peut l'organiser même si ses propres navires ne desservent pas cette destination. Partout, les réseaux de transport maritime doivent restructurer leurs systèmes de gestion pour faire face aux changements imposés par la globalisation de l'économie afin de pouvoir répondre à la demande des entreprises aux réseaux globaux émanant de la fusion des multinationales. Cette restructuration des SGTM impose une remise en cause de l'approche managériale de la compagnie aux niveaux stratégique, organisationnel et opérationnel. Dans ce travail, on s'intéresse aux problèmes, liés à la planification des tâches, rencontrés au niveau opérationnel.

D'abord, en tant que premier objectif de ce travail nous projetons de modéliser le domaine de transport maritime par un système multi-agents afin d'identifier et d'utiliser les protocoles de négociation qui seront utilisés lors de la distribution et la décomposition des tâches entre les agents transporteurs [1]. Pour cela, nous préconisons une modélisation des SGTM basée sur une classification de leurs différentes tâches en trois sous-systèmes. Le premier regroupe toutes les tâches se rapportant à la planification dans un SGTM : **Sous-Système de planification**. Le deuxième regroupe toutes les tâches relatives à la supervision d'une flotte de bateaux dans un SGTM : **Sous-système de supervision**. Le troisième regroupe toutes les tâches relatives à la gestion des bateaux-comandants dans un SGTM : **Sous-système Ergonomique** (aide à la navigation et assistance au commandant du bateau). En suite, nous allons développer une stratégie de négociation entre les agents basée sur le protocole Contract-Net.

Le deuxième objectif de ce travail est de mettre en place une approche originale pour le calcul des plans d'actions pour le routage et la répartition des missions des bateaux. Il concerne l'élaboration de nouveaux modèles pour l'interprétation, la négociation et la planification.

L'approche de résolution que nous préconisons est basée sur les techniques de la programmation génétique (AG) améliorée par l'utilisation simultanée, de l'algorithme  $\lambda$ -interchange et de la méthode de Recherche Tabou afin de générer des tournées satisfaisant des contraintes d'opération (capacité maximale ou des fenêtres de temps pour le service) et générer une solution en temps réel.

Le troisième objectif est la résolution du problème d'optimisation de chargement/déchargement des porte-conteneurs. La résolution proposée repose sur l'utilisation d'un réseau de neurones multi-couches avec apprentissage supervisé et correction d'erreurs par l'algorithme du rétropropagation du gradient.

Le reste de ce papier s'articule autour de six sections. La section 2 donne une brève définition des agents et des systèmes multi-agents et explique l'apport de l'approche agent par rapport aux autres paradigmes de modélisation cités dans la littérature. La section 3 décrit l'approche préconisée : structuration d'un SGTM et sa modélisation par les agents. Dans la quatrième section, nous présentons la gestion des appels d'offre et le mécanisme de négociation adopté. La section 5 présente la planification des tâches dans un SGTM et donne la description des différents algorithmes de base proposés. La section 6 décrit l'algorithme général des AG améliorés par la méta heuristique du Recherche Tabou déployé pour résoudre le problème des tournées des bateaux. La section 7, présente la problématique de l'optimisation du chargement/déchargement des marchandises et la description de la modélisation de ce problème par les réseaux de neurones. Enfin, nous terminons ce papier par une conclusion et les perspectives de ce travail.

## **2. Modélisation des systèmes de gestion de transport maritime par des systèmes multi-agents**

### **2.1 Agent et Systèmes multi-agents**

Par agent, on entend généralement, la conception d'une entité informatique qui soit capable de raisonner et qui soit la plus autonome possible. Une telle entité est également capable de communiquer, d'échanger des points de vue, de négocier [2,3,4] et de collaborer avec les autres entités de son environnement. Chaque entité est caractérisée par :

- L'autonomie : l'agent agit sans l'intervention d'humains ou d'autres intervenants et dispose d'un certain contrôle sur ses actions et ses états internes [5].
- L'habilité sociale : l'agent interagit avec d'autres agents (pouvant être des êtres humains) à l'aide d'un langage de communication d'agent.
- La réactivité : l'agent perçoit son environnement (pouvant être un monde physique, un utilisateur via une interface graphique, un ensemble d'autres agents ou encore tous ces éléments combinés) et répond de manière opportuniste aux changements qui y surviennent [6].
- La pro-activité : l'agent n'agit pas simplement aux stimuli de son environnement, il est aussi capable de démontrer des comportements dirigés par des buts en prenant des initiatives.

Un système multi-agents (SMA) [7,8] est un environnement, qui comporte un ensemble d'agents, capables d'interagir entre eux. En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre l'environnement et les agents, soit par perception, soit par communication.

Les principales caractéristiques d'un SMA sont :

1. l'hétérogénéité des agents : les messages doivent être mutuellement compréhensibles.
2. l'échange de savoir : un agent coopératif au sein du SMA doit pouvoir exprimer ses différents types de connaissances.
3. le contrôle local : les agents doivent être autonomes ; c'est-à-dire, leur comportement ne doit dépendre ni d'un planificateur central ni d'interactions pré-définies.

### 3. Approche préconisée

La modélisation et la simulation sont devenues des outils indispensables pour comprendre et contrôler des systèmes artificiels ou naturels de plus en plus complexes. On a donc vu fleurir des multitudes d'outils mathématiques et informatiques afin de construire des modèles représentant au mieux la réalité. A titre d'exemple, on peut citer les automates à états finis ou à événements finis, les réseaux de Petri, les « state charts » d'Harel, les modèles à base de files d'attente, les modèles stochastiques, les modèles fonctionnels, les équations différentielles et les modèles à base d'agents.

Dans notre approche, nous préconisons une modélisation des SGTM par des systèmes multi-agents basé sur la négociation et la coopération inter-agents. Le premier objectif de notre travail est la mise en place d'une infrastructure multi-agents en se basant sur une classification du SGTM suivant ses différentes tâches. Cette classification permet de regrouper toutes les tâches fortement liées dans un sous-système et celles n'ayant pas ou peu de relations entre elles dans deux sous-systèmes différents. Le résultat de cette classification est un ensemble de sous-systèmes caractérisés par une forte cohésion et un faible couplage [9]. Ainsi, toutes les tâches se rapportant à la gestion du fret et de chargement de marchandises dans un sous-système, **les tâches se rapportant** à la gestion navires-commandants dans un sous-système et **les tâches se rapportant** à la gestion de la flotte de navires dans un sous-système. Donc un SGTM sera composé des trois sous-systèmes suivants (cf. figure 1) :

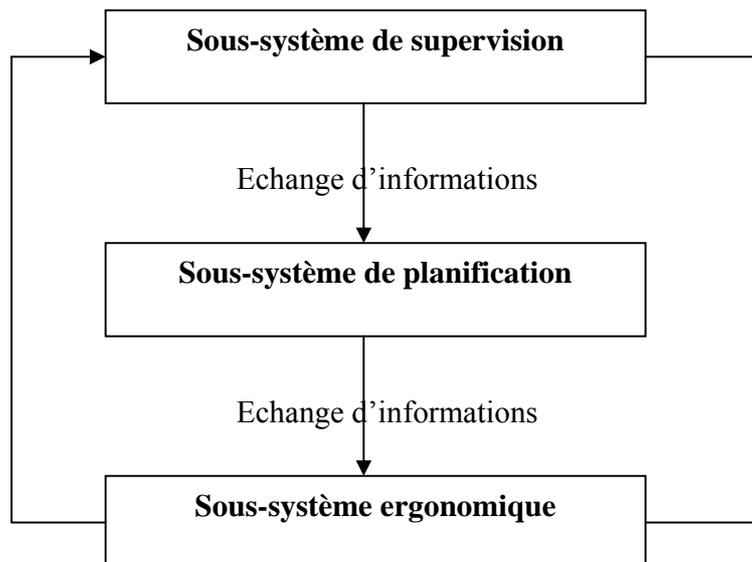
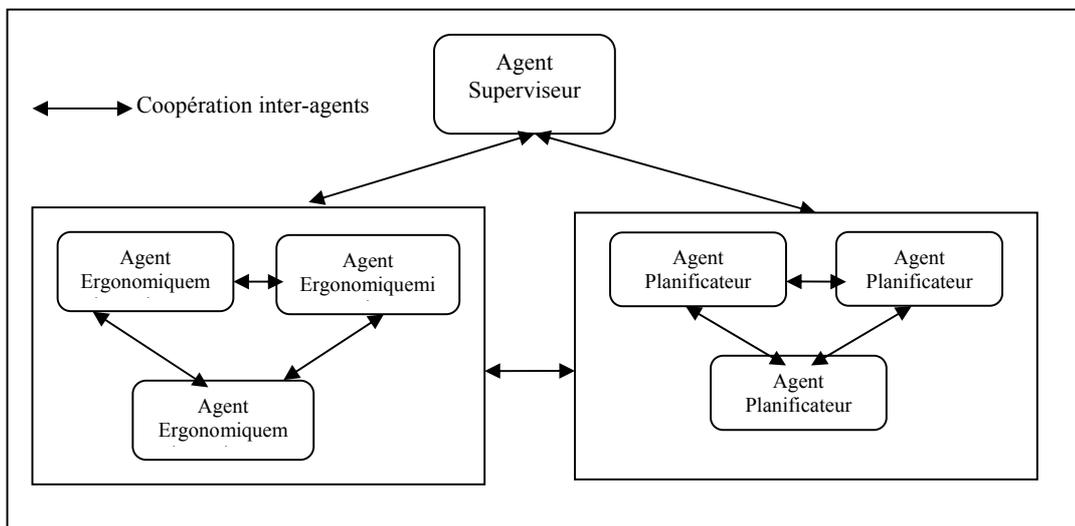


Figure 1 : Classification d'un SGTM

- Un **sous-système de planification** chargé de la planification et la gestion du fret et du chargement/déchargement des marchandises.
- Un **sous-système Ergonomique** chargé de la gestion de navire-commandant.
- Un **sous-système de supervision** chargé de gestion de la flotte de navires.

### 3.1 Modélisation du SGTM par un SMA

Généralement, le domaine de transport maritime comporte 3 entités physiques distinctes: les compagnies de transport, les commandants et les navires. Les compagnies de transport disposent d'une flotte de navires et sont amenées à allouer à leur flotte, un ensemble de commandes leur arrivant de manière asynchrone et dynamique, tout en respectant un ensemble de contraintes. Dans notre approche, nous proposons la modélisation suivante :



**Figure 2. Structure d'un Système multi-agents de transport**

- Les compagnies seront considérées comme des agents (Agents superviseurs).
- Les navires avec leurs commandants seront considérés comme des agents : (Agents planificateurs).
- Les tâches se rapportant à la gestion des navires et des commandants comme des agents : (Agents Ergonomiques).

Dans notre approche le système appartenant au domaine du transport intelligent que nous proposons sera composé de trois sous-systèmes :

- Un sous-système de planification composé d'un ensemble d'entités (agents planificateurs). Chacun de ces agents fret sera chargé de la gestion du fret, calcul de tournée et planification du chargement/déchargement de marchandises d'un navire.
- Un sous-système de gestion navires-commandant composé d'un ensemble d'entités (agents Ergonomiques). Chacun de ces agents sera chargé de la gestion des tâches et fonctions se rapportant à un navire et son commandant.

- **Un sous-système de supervision** qui constituera le composant élémentaire d'un SGTM (Agent superviseur) qui sera chargé de la supervision de toutes les fonctions du SGTM.

### **3.2 Agent Planificateur**

La principale tâche d'un agent planificateur est l'élaboration de ses plans locaux. Il doit être capable d'effectuer des livraisons de commandes dans différents ports, générer des plans locaux et négocier avec les autres agents planificateurs afin de minimiser le coût du plan commun. Chaque agent reçoit des commandes sous la forme: "transporter  $a_1$  unités de la marchandise  $G_1$  de la ville  $V_1$  à la ville  $V_2$ . Après avoir livré toutes les commandes, les agents doivent retourner à leur ville de départ.

Un plan de chargement peut constituer une autre fonction importante. Il consiste à établir, selon l'affectation optimale de l'ordre des chargements et déchargements, la disposition optimale du fret dans les conteneurs des différents navires et à répercuter cette information au moment des chargements en vue du positionnement des marchandises à destination. Cette fonction suppose évidemment que la planification des itinéraires ait été assurée préalablement. En plus, il assure un ensemble de fonctions basiques telles que la gestion des ordres et des documents et la mise à jour du déroulement des opérations de transport. Il prendra en charge la gestion des opérations relatives aux chargements telles que : La vérification de l'adéquation des marchandises avec les documents qui s'y réfèrent par la confrontation des documents avec les unités logistiques au moyen d'outils d'identification automatique. L'enregistrement de cette vérification dans les messages de remontée de l'information est également pris en charge ainsi que l'accessibilité simultanée des partenaires tiers à l'information. La transmission de ces données en temps réel et le contrôle de la conformité de celles-ci avec les normes de sécurité permettent d'assurer le tracking-tracing des chargements.

### **3.3 Agent Ergonomique**

Il est chargé de la mise en oeuvre des connaissances relatives à un navire et aux itinéraires nécessaires. Il s'occupe aussi de gérer les infrastructures matérielles et logicielles disponibles pour que les missions de transport puissent être accomplies avec le maximum de confort, de sécurité et d'efficacité.

Les autres fonctions qui seront associées à un agent Ergonomique peuvent également être assez variées telles que :

- ❖ La vérification, avant le départ, de la bonne maintenance du navire conformément aux interventions planifiées par l'applicatif.
- ❖ La vérification de la présence à bord de tous les accessoires nécessaires et des documents réglementaires électroniques ou non.
- ❖ Le contrôle automatique de l'état du navire et notamment des fonctions électromécaniques et électromotrices et la mémorisation des valeurs dès que le navire entre dans une zone d'alerte.

La gestion du commandant prend en compte les fonctions de contrôle des documents présents à bord sous forme électronique ou papier facilitant une consultation de la part des autorités compétentes. La gestion du temps (temps d'arrêts,...) constitue une application prioritaire mémorisée et permet l'identification du commandant (par association). Le calcul de la rémunération peut également être réalisé par une application appropriée connectée avec le

système de gestion des temps. Ces données sont donc accessibles, en temps réel, à la direction et au commandant.

L'aide à la conduite par la présence d'une série d'outils (sonar, caméra et analyse d'images,...) destinés à saisir et à analyser les données relatives à l'environnement dans le but de détecter des anomalies (détection d'obstacles, navigation...).

L'assistance au commandant regroupe des outils et services importés temporairement (via Internet ou par chargement à bord) tels que des cartes digitales muettes, des programmes pédagogiques et des programmes de mémorisation des caractéristiques personnelles des commandants.

### **3.4 Agent Superviseur**

L'agent superviseur est d'une part chargé de la coopération avec d'autres agents superviseurs. D'autre part, il supervise la gestion des appels d'offre :

- Emission d'un appel d'offres avec une description du travail à faire aux agents planificateurs.
- Attente et collecte des réponses (jusqu'au déclenchement d'un timeout).
- Tri des réponses reçues pour ne garder que la ou les meilleures offres.
- Attribution du contrat aux agents ayant fourni les meilleures offres.
- Attente des résultats.

Il peut contribuer à la planification opérationnelle des itinéraires des agents planificateurs et répartit les missions individuelles. Il s'agit par là principalement de déterminer de manière dynamique les itinéraires optimaux, la programmation des chargements et le suivi de la position des navires. Il assure l'échange d'information avec la base.

### **3.5 Apport de l'approche agent par rapport aux autres approches de modélisation.**

Les différentes approches de modélisation des systèmes informatiques se résument en trois grandes approches à savoir l'approche fonctionnelle, l'approche objet et l'approche agent qui émerge.

L'approche fonctionnelle s'applique généralement au domaine des systèmes d'information d'entreprises en traitant à la fois le point de vue données et traitements et les aspects allant de la conception à la réalisation.

En approche objet, concevoir un système revient à le décomposer en unités indépendantes, chaque unité dispose de ses propres caractéristiques (attributs) et des opérations qu'elle sait effectuer (méthodes) [10]. Contrairement à un agent, un objet ne réagit qu'à un appel de méthode et toutes les situations auxquelles il va être confronté doivent être prise en compte par le concepteur. En plus les objets ne peuvent avoir des buts ni recherche de satisfaction et le mécanisme d'envoi de messages se résume en un simple appel de procédure. Il n'y a pas de langage de communication à proprement dit. Les mécanismes d'interactions sont à la charge du programmeur. En revanche, les agents disposent des buts qui leurs donnent une autonomie de décision vis à vis des messages reçues [11].

L'approche agent revient à déterminer des éléments générateurs du comportement d'un système complexe de transport [12], qui va se réorganiser continuellement selon différentes échelles grâce à la réactivité des agents. Le comportement global du système résultera de celui des relations entre ses différents agents et ce comportement sera parfois régulier, parfois chaotique, dans tous les cas non linéaire. Le modèle comportemental doit donc prendre en

compte des possibilités de non régularité du système. Cela distingue bien l'approche par agents de celles retenues avec les modèles fonctionnels et les modèles à objets.

En conclusion, les modèles fonctionnels et les modèles à objet ne sont pas suffisants pour modéliser les systèmes de transport intelligents. Car elles ne prennent pas en compte les caractéristiques d'autonomie et de sociabilité de chaque composante de ce système ainsi que leurs caractéristiques de complexité et d'évolution. L'idée de l'utilisation de l'approche agent découle du constat que l'utilisation d'un transport intelligent devient impérative pour les SGTMs. Il est donc nécessaire d'aborder ces systèmes par de nouvelles approches tant pour la planification des tâches, la supervision et l'ergonomie.

## **4. Communication et négociation inter-agents**

### **4.1 Généralités**

Pour modéliser des agents en coopération [13], on est souvent obligé de définir une suite d'échange des messages, c'est-à-dire une conversation d'agents. Cet échange de message suit une certaine séquence attendue. De telles conversations sont définies par leurs protocoles appelés « protocoles de coopération » ou « protocole de conversation », qui indiquent les séquences autorisées de messages. Une grande variété de protocoles existe : A titre d'exemples le protocole d'appel d'offre « Contract net protocol », les divers protocoles dans les travaux de [Demazeau] et les protocoles génériques proposés par FIPA [14] dont on cite « Dutch auction protocol » et « iterated contract net protocol ».

Dans notre approche nous proposons l'utilisation du protocole Contract-Net [15] pour la négociation inter-agents. C'est un mécanisme de négociation que nous considérons comme un processus de prise de décision commun entre deux ou plusieurs agents, durant lequel chacun essaie d'atteindre son objectif. Actuellement les systèmes informatiques deviennent de plus en plus autonomes et commencent davantage à prendre des décisions. Un grand nombre de ces décisions sont effectuées de concert avec d'autres systèmes. Dès lors, il est important de se pencher sur les protocoles de négociation entre systèmes de manière à rendre plus efficace ce type de coopération.

### **4.2 Protocole Contract-Net**

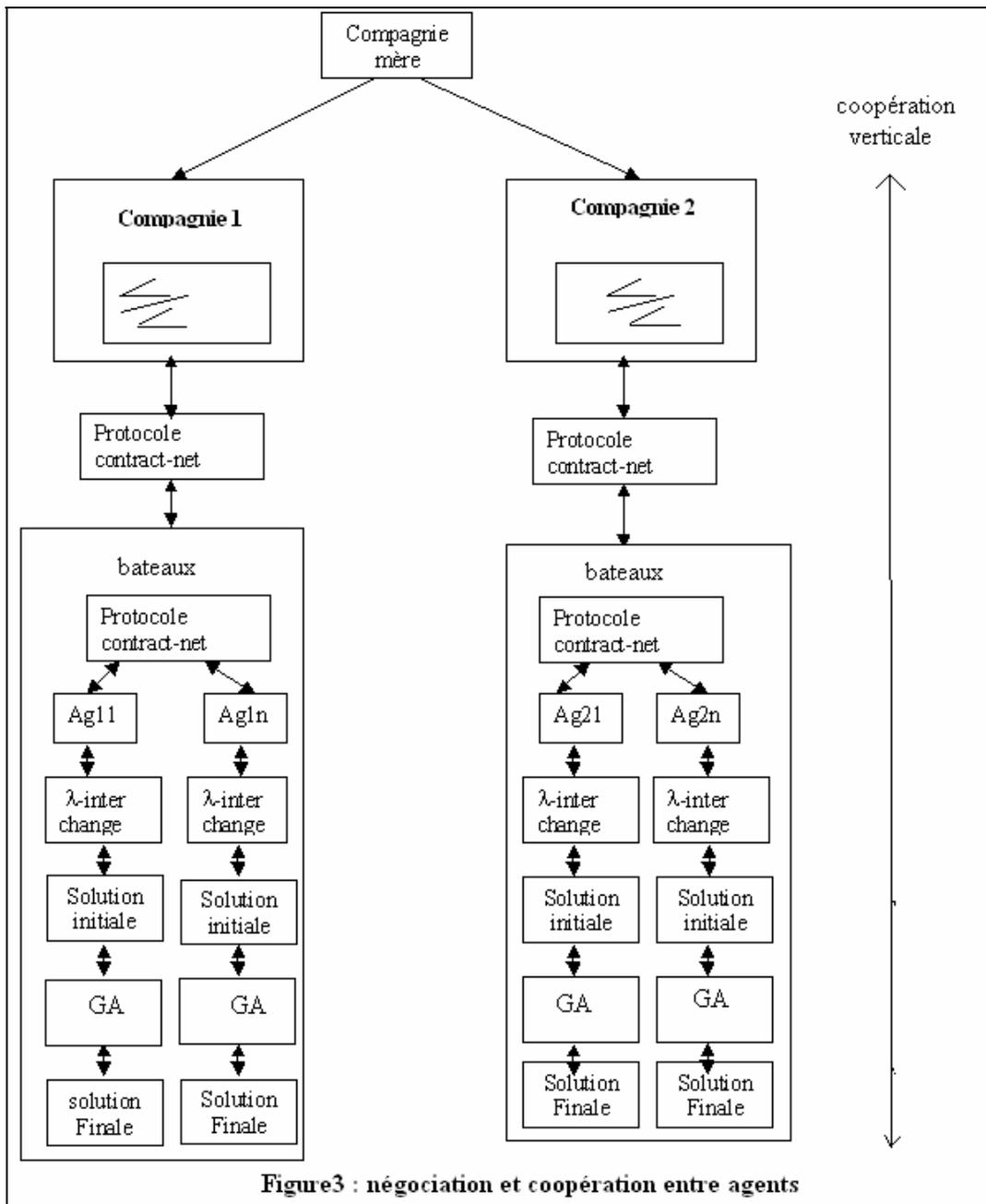
Le Protocole Contract-Net [16] est un mécanisme de négociation entre deux types d'agents : contractant et gestionnaire. Il permet à un gestionnaire suite à quelques échanges avec un groupe d'agents de retenir les services d'un agent appelé contractant pour l'exécution d'une tâche contrat. Ce protocole est qualifié de type 'sélection mutuelle' puisque pour signer un contrat l'agent choisi doit s'engager envers le gestionnaire pour l'exécution de la tâche et le gestionnaire de ça part ne sélectionne que l'agent ayant fourni la proposition la plus avantageuse. La version originale du protocole décrite dans ce que suit comporte trois étapes principales : l'appel d'offre, la soumission des propositions et l'attribution de contrat.

Etant donné une tâche, un gestionnaire et un groupe d'agents :

1. Le gestionnaire envoie un message de type « annonce-tâche » à un groupe d'agents ou fait un "broadcast".
2. Chaque agent évalue la tâche annoncée à l'aide d'une fonction locale "évalue-annonce".
3. L'évaluation précédente permet à certains agents de soumettre une proposition à l'aide d'une "soumission-tâche" au gestionnaire.
4. Si une proposition est jugée satisfaisante (à l'aide de la fonction "évalue-soumission"), alors le gestionnaire envoie un message de type "acceptation" à celui dont la proposition est retenue. Il envoie également un message de type "refus" aux autres agents dont les propositions n'ont pas été retenues.
5. Le gestionnaire peut mettre fin à la période d'acceptation de proposition si le temps d'expiration est dépassé.
6. Si aucune proposition n'a été retenue, alors le gestionnaire fait parvenir à tous les agents non retenus un message de type "refus" pour indiquer le rejet de chacune des propositions. Il peut alors se retirer de la négociation, retenir la proposition la plus acceptable, redémarrer un nouvel appel d'offre (nouveau "annonce-tâche") ou prolonger le temps d'expiration de la période d'acceptation de proposition.
7. L'agent ayant obtenu le contrat, remet un rapport d'exécution lorsque la tâche est complétée.

## **5. La planification des tâches dans les SGT**

La planification des tâches dans le transport maritime constitue un problème crucial. Elle consiste à mettre au point des algorithmes et des stratégies performants pour la résolution des problèmes de programmation mathématique et d'optimisation combinatoire que l'on retrouve au cœur des modèles développés dans le domaine du transport en général et particulièrement le problème d'affectation du trafic dans le transport maritime. Dans ce papier, nous proposons les algorithmes génétiques améliorés par la méta heuristique du Recherche Tabou pour la résolution de ce problème (cf.figure3).



### 5.1 Les Algorithmes Génétiques (AG)

Les AGs ont été développés par John Holland [17] en 1975, pour imiter les phénomènes d'adaptation des êtres vivants. L'application aux problèmes d'optimisation a été développée ensuite par Goldberg. On part d'une population initiale de N solutions, chacune étant codée par une chaîne de caractères appelée chromosome. Holland recommande un codage sous forme de chaînes de bits, mais ce n'est pas obligatoire. Chaque chromosome est muni d'une mesure d'adaptation (fitness). Dans notre cas, il s'agit tout simplement de la distance totale traversée par un bateau. Un algorithme génétique choisit des paires de chromosomes-parents, en favorisant les plus adaptés (ceux de meilleur fitness) et engendre de nouvelles solutions (enfants) en appliquant

des opérateurs de croisement (crossover) et de mutation. On espère ainsi que les bonnes solutions vont échanger par croisement leurs caractéristiques et engendrer des solutions encore meilleures. Pour exploiter ce type d'algorithme, il convient de définir les cinq concepts suivants :

- Définir l'encodage et le décodage d'une solution, ainsi que sa valeur.
- Définir une population initiale
- Définir une méthode adéquate de sélection des individus (chromosomes)
- Définir des opérateurs adéquats de reproduction
- Définir des opérateurs adéquats de mutation

### 5.1.1 Le codage

Le premier concept à définir avant l'utilisation des AGs est le codage des entités qu'ils manipulent. Dans le domaine du transport maritime, nous considérons une population constituée de chromosomes qui représentent les plans communs de commandes. Un plan commun représente la liste de ports qui doivent être visités par les bateaux. Nous attachons à chaque chromosome  $x$  une valeur d'adaptation ou fitness  $f(x)$  qui indique la qualité de la solution qui sera calculée en fonction de la distance totale à parcourir par les bateaux. Dans notre cas, le codage adopté est le suivant :

- Un gène correspond à un port.
- Un chromosome correspond à une suite de ports (chaîne d'entiers) de longueur  $M$  où  $M$  est le nombre total des ports.
- Une Population correspond à un ensemble de  $N$  chromosomes.

### 5.1.2 Création de population initiale

Le choix de la population initiale conditionne fortement la rapidité et l'efficacité de l'algorithme. Si la position de la solution optimale est totalement inconnue, il est naturel de générer aléatoirement les chromosomes. Dans le problème du transport, nous utilisons une génération de ce type pour créer une partie de la population initiale, le reste de la population sera généré par la méthode  $\lambda$ -inter change (sec 5.1.2.1) afin d'accélérer la convergence.

#### 5.1.2.1 La méthode $\lambda$ -inter change

C'est une méthode de recherche locale introduite par Osman et Christofides [18]. Etant donnée une solution faisable de problème d'optimisation des itinéraires des bateaux représentée par l'ensemble  $S = \{R_1, \dots, R_p, \dots, R_q, \dots, R_k\}$  où  $R_p$  représente l'ensemble des ports qui doivent être visités par un bateau  $P$  et  $k$  représente le nombre total des bateaux. Une  $\lambda$ -inter change entre deux itinéraires  $R_p$  et  $R_q$  est un remplacement d'un sous-ensemble  $S_1 \subseteq R_p$  de cardinalité  $|S_1| \leq \lambda$  ( $\lambda$  est un entier qui détermine le nombre maximum de ports qui peuvent être échangés entre deux itinéraires) avec un autre sous-ensemble  $S_2 \subseteq R_q$  de cardinalité  $|S_2| \leq \lambda$ , pour obtenir deux nouveaux itinéraires  $R'_p = (R_p - S_1) \cup S_2$  et  $R'_q = (R_q - S_2) \cup S_1$  et une solution voisine  $S' = \{R_1, \dots, R'_p, \dots, R'_q, \dots, R_k\}$ .

Le voisinage  $N_\lambda(S)$  d'une solution  $S$  est l'ensemble de tous les voisins  $\{S'\}$  de  $S$  générés par la méthode  $\lambda$ -interchange pour une valeur d'un  $\lambda$  donnée, il est exploré en appliquant  $\lambda$ -interchange pour toutes les combinaisons possibles de couples  $(R_p, R_q)$ .

### 5.1.2.2 Génération de la population initiale

Nous proposons d'abord la génération d'une solution initiale  $S_0$  à l'aide de la méthode Push-Forward Insertion Heuristic PFIH introduite par Solomon [19] ; ensuite, une première partie d'une population par rapport à cette solution est créée par  $N_\lambda(S_0)$  « voisinage de  $S_0$  généré par la méthode  $\lambda$ -inter change ». Le reste de la population est totalement produit d'une manière aléatoire ou indépendante de  $S_0$ . La raison d'avoir cette population hétérogène est évidente : une population d'individus entièrement du même voisinage ne peut aller trop loin de ce voisinage et par conséquent manque l'occasion d'explorer d'autres solutions. La proportion de chromosomes générés par  $\lambda$ -inter change et de chromosomes aléatoires est régie par un paramètre TAUX\_Hasard qui indique le taux de la population initiale généré d'une manière aléatoire. Plus ce rapport est grand, plus la population initiale devient diverse. Ce paramètre reflète également le niveau de confiance de l'utilisateur par rapport à la solution initiale obtenue par la méthode PFIH. S'il y a une forte chance que l'optimum global soit situé dans le voisinage  $N_\lambda(S_0)$ , dans ce cas c'est intéressant d'avoir un petit TAUX\_Hasard pour forcer la population à converger vers l'optimum plus rapidement. Dans notre étude nous avons limité la taille de chaque population à  $N=100$  chromosomes et le nombre de générations entre 500 et 1000. La solution obtenue s'améliore au fur et à mesure que nous explorons plus de générations.

### 5.1.3 La sélection

Après la création d'une population initiale, nous aurons besoin d'un mécanisme de sélection de parents pour la reproduction. Dans notre approche, nous proposons le mécanisme de sélection par tournois [20]. Dans ce mécanisme, deux copies identiques  $POP_1$  et  $POP_2$  d'une population de taille  $N$  sont prises à chaque génération. Pour la population  $POP_1$ , chaque couple de chromosomes adjacents (d'indices  $2i$  et  $2i+1$ ) sont comparés, celui, ayant une plus petite valeur de fitness, sera qualifié pour être le parent potentiel  $P_i$ . Une fois tous les couples sont comparés dans  $POP_1$ , nous aurons  $N$  "pères", à savoir  $\{P_0, P_1, \dots, P_{N-1}\}$ . Le même processus étant appliqué à  $POP_2$ , nous obtenons ainsi un ensemble de "mères"  $\{m_0, m_1, \dots, m_{N-1}\}$ . Ensuite, père  $P_i$  est croisé à une mère  $m_i$ . La figure 4 illustre bien ce processus.

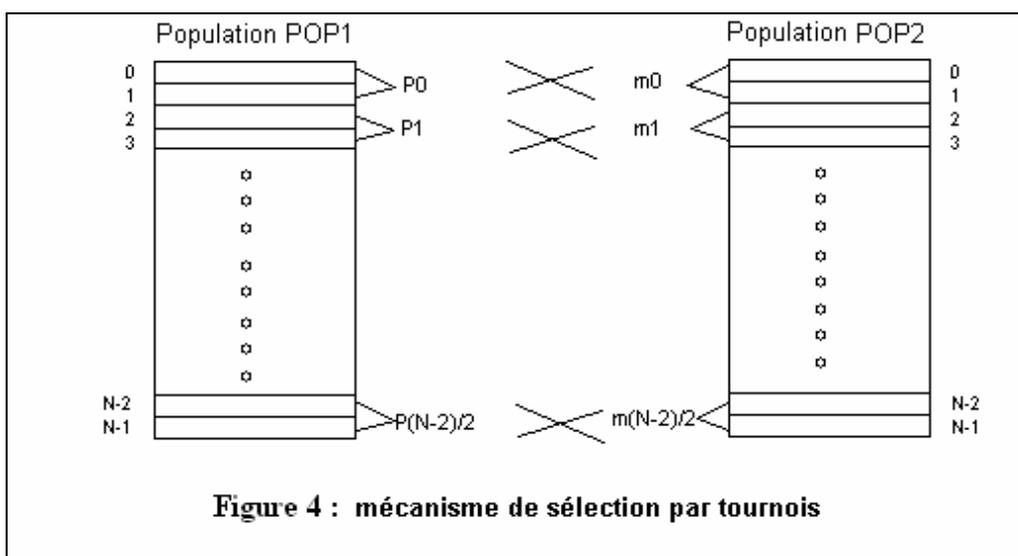


Figure 4 : mécanisme de sélection par tournois

### 5.1.4 La reproduction

La reproduction est un mixage aléatoire des gènes de deux chromosomes, donnant naissance à deux chromosomes enfants ayant une empreinte génétique nouvelle, héritée des parents. L'évolution normale d'une espèce, d'une génération à la suivante, est alors une simple recombinaison obtenue par croisement aléatoire des chromosomes appariés. La phase de reproduction dans les algorithmes génétiques se traduit par les mécanismes de croisement et mutation.

#### 5.1.4.1 Le croisement:

C'est une opération recombinaison de deux chromosomes parents pour produire deux nouveaux chromosomes qui héritent des caractéristiques parentales. Cet opérateur a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Généralement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Plusieurs techniques de croisement sont proposées dans la littérature [21,22,23,24]. Dans le contexte du problème de routage des bateaux où chaque entier apparaît seulement une fois dans chaque solution, la plupart de ces techniques produisent inévitablement des enfants inadmissibles qui peuvent contenir des entiers dupliqués. Pour éviter ce problème de duplication, nous définissons un opérateur de croisement adapté à notre problème.

L'opérateur de croisement que nous proposons est un opérateur qui procède en choisissant deux points de coupure au hasard.

Si on considère l'exemple suivant :

Parent 1 : 3 25 41

Parent 2 : 2 14 53

La section entre deux points de coupure définit une série de permutations à exécuter sur le deuxième parent (voir figure 5). Dans le cas de notre exemple, nous permutoons 2 avec 1 et 5 avec 4, nous obtenons ensuite l'enfant suivant :

Enfant 1 : 3 14 52

En effectuant la même permutation sur le premier parent on obtient l'enfant suivant :

Enfant 2 : 1 25 43

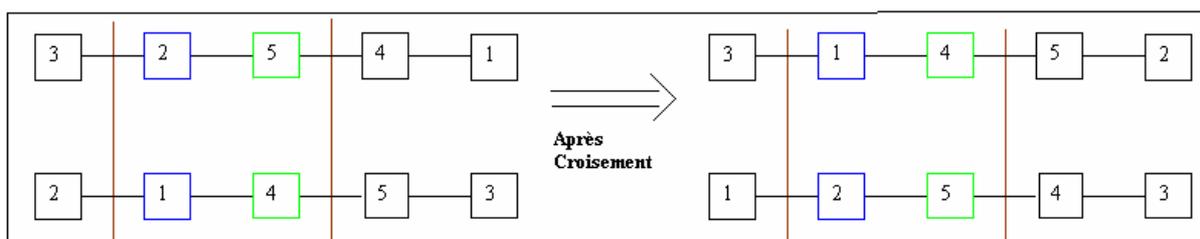
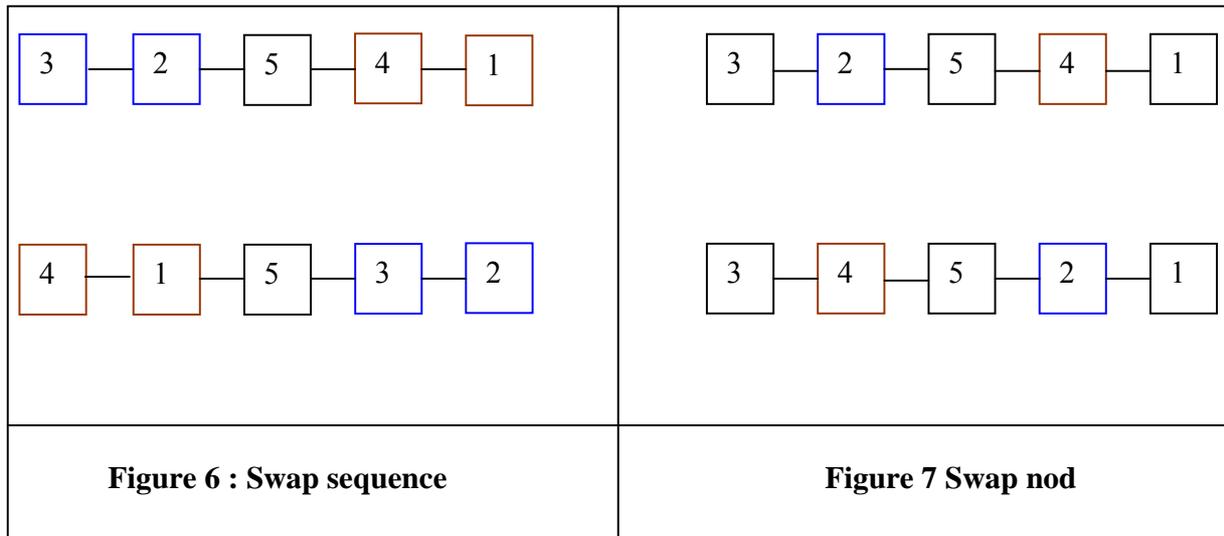


figure 5: Opérateur de Croisement

#### 5.1.4.2 La mutation :

La mutation est une opération qui permet de diversifier l'espace de recherche et de s'échapper aux minima locaux, elle consiste à tirer aléatoirement un gène dans un chromosome et à le remplacer par une valeur aléatoire.

Plusieurs opérateurs de mutation ont été proposés dans la littérature [25,26]. Puisque dans notre problème les solutions sont de longueur fixe, seulement l'opérateur d'échange de gène (Voir figure 7) ou échange de séquence de gènes (voir figure 6) seront utilisés.



Il y a un paramètre important lié aux opérations de mutation. C'est la probabilité de la mutation ( $P_{\text{mutation}}$ ), le taux qui précise la probabilité avec laquelle les chromosomes seront mutés. Les premières recherches ont prouvé qu'une  $P_{\text{mutation}}$  excessive entraîne une convergence prématurée; une petite  $P_{\text{mutation}}$  produit le résultat opposé : convergence intolérablement lente.

## 6. Les algorithmes génétiques améliorés par la méthode de Recherche Taboue

### Principe de la méthode de Recherche Taboue

La Recherche Taboue est une stratégie de recherche basée sur l'exploitation de la mémoire, originellement proposée par F.Glover [27] pour guider les méthodes de recherche locale à surmonter les problèmes des optima locaux. L'algorithme de Recherche Taboue garde une liste de solutions déjà explorées. Cette liste, connue sous le nom de Liste Taboue, a la structure d'une file d'attente de taille fixe. Le rôle de cette liste est d'éviter l'exploration de nouveau des solutions déjà archivées.

### La liste taboue T

Glover [27] a montré qu'une liste taboue T de taille  $NT = 7$  à 20 suffit en pratique pour empêcher l'algorithme de boucler en revenant sur une solution déjà visitée. T fonctionne donc comme une sorte de mémoire. A chaque itération, la  $NT^{\text{ième}}$  solution de T (la plus ancienne) est écrasée par la dernière solution examinée. En pratique, T se gère simplement avec une structure de données de type file d'attente.

La mémoire peut être utilisée à court terme ou à long terme. En court terme, la Liste Taboue de taille NT enregistre les NT dernières opérations que l'algorithme a rencontrées et les considère comme Taboues. La mémoire à long terme, complète celle à court terme en fournissant des informations additionnelles sur le nombre de fois que ces opérations ou solutions Taboues ont été rencontrées. La liste Taboue mémorise deux sortes d'informations :

- Les opérations les plus récentes ; ce qui permet d'éviter de refaire les mêmes croisements ou mutations dans un futur proche.

- Les solutions rencontrées au cours de la recherche d'itinéraires optimaux par les AGs. Chaque solution rencontrée est codée sous forme de chaîne d'entiers et stockée dans la liste Taboue. En plus des informations de routage, nous attachons aussi le coût total de la solution à la chaîne d'entiers. La durée de vie des opérations taboues ainsi que les solutions taboues dans la liste est gouvernée par la taille de la liste taboue (TLT). Plus cette liste est longue, plus ces opérations et solutions y restent comme tabous. Mais il s'est avéré qu'une longue liste taboue restreint le processus de recherche et lui cause une convergence prématurée. Dans notre cas, nous fixons TLT à 10.

La liste taboue enregistre aussi des informations de fréquence à savoir combien de fois une solution ou une opération a été rencontrée pendant la recherche ; ce genre d'informations est important pour déterminer le statut du processus de recherche. Une haute fréquence qui dépasse un seuil  $N_0$ , assignée à une opération ou à une solution signifie que la recherche est capturée dans un optimum local ce qui nécessite la diversification de la population.

### **Les algorithmes génétiques améliorés par la méthode de Recherche Taboue**

Dans ce qui suit nous présentons une description générale des Algorithmes Génétiques améliorés par la méthode de Recherche Taboue :

Générer une solution initiale  $S_0$  avec la procédure PFIH

Générer une population initiale de  $N$  chromosomes avec un Taux-Hasard=0.6 ( 40% à partir de  $N_2(S_0)$  et 60 % aléatoirement)

Tant que le nombre maximum des générations n'est pas atteint faire :

Evaluer la fitness  $f(x)$  pour chaque chromosome  $x$  de la population, calculer la valeur moyenne de la fitness et la probabilité de mutation ( $P_{\text{mutation}}$ );

Sélectionner le meilleur chromosome et le placer dans la liste des solutions taboues ;

Mettre l'opération génératrice dans la liste des opérations taboues ;

// création d'une nouvelle génération //

Tant qu'une nouvelle population de même taille n'est pas créée faire:

Répéter

Sélectionner un couple de chromosomes parents ( $P,m$ ) de la population à l'aide de la méthode de Tournoi ;  
Jusqu'à ce que ( $P,m$ ) ne figure plus dans la liste taboue

Croiser les parents pour former deux nouveaux enfants ;  
Placer cette opération sur la liste des opérations taboues ;

Répéter

Sélectionner au hasard un chromosome  $C$  de la population ;  
Jusqu'à ce que  $C$  ne figure plus sur la liste taboue

Muter le chromosome à des points au hasard ;

Placer cette opération dans la liste des opérations taboues ;

Placer les nouveaux enfants dans une nouvelle population ;

Remplacer les 4% des faibles chromosomes dans la nouvelle population avec les 4% des meilleurs chromosomes de l'ancienne population ;

Fin tant que.

Remplacer l'ancienne population par la population générée ;

Fin tant que

Exécuter l'algorithme  $\lambda$ -interchange ( $\lambda=2$ ) pour raffiner la solution trouvée et obtenir une solution améliorée ;

## **7. Optimisation de chargement/déchargement)**

### **7.1 Le problème du chargement /déchargement**

Un porte-conteneurs fait escale dans différents ports pour charger et décharger des conteneurs. Les conteneurs sont rangés par piles. Ainsi lors de la tournée, il est parfois nécessaire de déplacer des conteneurs (sans les débarquer) car ils se trouvent au-dessus d'autres conteneurs à

décharger. L'objectif est donc de charger le bateau de façon à minimiser le nombre de ces déplacements inutiles et coûteux. Devant l'impossibilité de calculer parfaitement le meilleur remplissage, on est contraint d'utiliser des stratégies basées sur l'expérience et le bon sens. Le but de ce travail est de définir des objectifs pour le remplissage et développer différentes méthodes pour atteindre nos objectifs.

Après avoir assigné à chaque bateau un parcours bien précis, on va aborder le problème de chargement/déchargement qui consiste à optimiser la façon dont les conteneurs sont entreposés dans le bateau, sachant que ces conteneurs sont stockés en pile, et que l'accès à un conteneur qui n'est pas au sommet nécessite le dégagement des conteneurs qui le couvrent. On cherche bien sûr à limiter autant que possible le nombre de manutentions effectuées au cours d'un parcours. Pour cela, nous proposons une modélisation de ce problème par les réseaux de neurones.

## 7.2 Les réseaux de neurones

Les réseaux de neurones (RN) constituent une branche de l'Intelligence Artificielle [28] qui procède à l'analyse logique des tâches relevant de la cognition humaine et tente de les simuler par des programmes. Un réseau de neurones doit tout d'abord passer par une phase d'apprentissage durant laquelle il adapte ses coefficients synaptiques au problème étudié. Nous nous limitons, pour la résolution du problème de chargement/déchargement, à l'apprentissage supervisé à partir d'une base d'exemples qui est bien adapté pour l'apprentissage à partir de données complexes. Les entrées et sorties sont représentées par des attributs à valeurs réelles. Ce type d'apprentissage est tolérant au bruit et aux erreurs. Le temps d'apprentissage peut être long, par contre, après apprentissage, le calcul des sorties à partir d'un vecteur d'entrée est rapide. La critique principale est que le résultat de l'apprentissage, c'est-à-dire le réseau de neurones calculé par l'algorithme d'apprentissage, n'est pas interprétable par l'utilisateur : on ne peut donner d'explication au calcul d'une sortie sur un vecteur d'entrée. Ce qui explique l'appellation de «boîte noire».

Le problème du chargement /déchargement introduit beaucoup des contraintes à satisfaire et ne peut être considéré totalement sous forme d'un problème d'optimisation. Devant la difficulté de traduction de ce problème en fonction d'évaluation, nous l'abordons ici d'abord en tant que satisfaction de contraintes en exprimant les priorités au déchargement. L'objectif, bien évidemment, est de minimiser le temps global en agençant au mieux les emplacements lors du chargement. Nous pouvons définir le problème comme suit :

Il s'agit de mettre N conteneurs dans un espace (habitable) 3D.

Deux contraintes essentielles doivent être respectées :

- 1- Les conteneurs les plus lourds sont placés en bas.
- 2- Les conteneurs les plus prioritaires doivent être placés à l'arrière.

Pour cela nous tenons compte de trois caractéristiques pour chaque conteneur : Poids (P), Volume (V) et priorité de déchargement (Pr).

Nous pouvons subdiviser l'espace de placement (l'habitable) en zones utiles. La précision sur les zones dépend de l'importance qu'on peut donner aux notions : « lourd » « léger » et « prioritaire » d'une part et à la nature (petites ou grandes tailles par exemple) des conteneurs d'autre part.

Pour illustrer notre propos, nous considérons quatre zones utiles dans l'habitable : Haut, Bas, Avant et Arrière et nous considérons les règles suivantes :

Les plus lourds sont placés en bas, les plus prioritaires sont placés en arrière, les plus légers en haut, et les moins prioritaires en avant.

Pour affiner ce placement « trop grossier », on peut procéder de la même façon d'une manière récursive par subdivision de chacune des zones.

Nous avons modélisé cette procédure à l'aide d'un Réseau de Neurones (RN) qui nous permettra d'avoir pour chaque conteneur une liste ordonnée de zones possibles (figure 8).

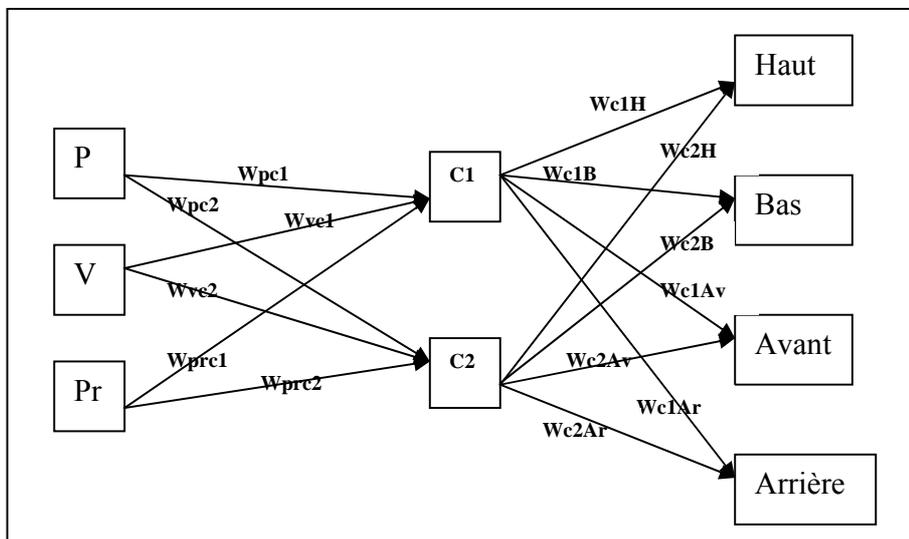
Ce RN possède trois Couches : Entrée. Sortie et une Couche cachée.

L'entrée du RN est le vecteur : (P, V, Pr).

La sortie est le vecteur zone : (Haut, Bas, Avant, Arrière).

La couche Cachée constituée de deux neurones C<sub>1</sub> et C<sub>2</sub>.

$W = (W_{ij})$  est le vecteur des coefficients synaptiques du réseau.



**Figure 8**

Pour un conteneur donné, ce réseau doit assigner une classe parmi les quatre.

L'apprentissage est supervisé et utilise un grand nombre d'exemples de chargement/déchargement.

Un exemple est modélisé par un vecteur d'entrée correspondant à (P, V, Pr) et un vecteur de sortie correspondant à une zone possible pour le conteneur.

On peut spécifier les valeurs d'entrée dans un ensemble fini de valeurs :

Par exemple :

(1,2,4,5) correspond à (léger, moyen, lourd, très lourd).

(1,2,4,5) correspond à (petit, moyen, volumineux, très volumineux)

(1,3,4,5,6,7) correspond à l'ordre de priorité

Un exemple d'entrée  $e_1 = (2,3,1)$  : un conteneur de poids moyen, volumineux et prioritaire.

Sortie est un vecteur  $s = (o_1, o_2, o_3, o_4)$   $o_i$  binaire correspond à la sortie  $i$  selon l'ordre des classes de la figure. Ainsi :

Le vecteur (1,0,0,0) signifie la classe Haute.

Le vecteur (0,1,0,0) signifie la classe Basse.

Le vecteur (0,0,1,0) signifie la classe Avant.

Le vecteur (0,0,0,1) signifie la classe Arrière.

Un exemple d'apprentissage  $Ex=(e_1,s_1)=((2,3,1),(1,0,0,1))$  : le conteneur  $e_1$  à placer en haut et en arrière. L'apprentissage consiste à introduire un certain nombre d'exemples pour calculer le vecteur des coefficients synaptiques  $W$  qui minimise l'erreur globale entre les sorties désirées et les sorties calculées.

Pour cela, nous avons utilisé l'algorithme de retropropagation du gradient [29] :

L'erreur considérée :  $E(W)=\frac{1}{2}\sum e^{(T_e-O_e)^2}$ , la fonction d'activation à la sortie de chaque cellule est la sigmoïde :  $\sigma(x)=1/(1+e^{-x})$ .

$T_e$  étant la sortie désirée

$O_e$  étant la sortie calculée

L'algorithme de la retropropagation est itératif et effectue la correction des coefficients synaptiques à chaque itération ( $W_{ij} \leftarrow W_{ij} + \Delta W_{ij}$ ) par propagation des erreurs de la sortie au sein des autres cellules.

Après un entraînement suffisant de ce réseau, nous l'appliquerons pour le classement des conteneurs. Le conteneur est présenté par ses caractéristiques (P,V,Pr) au réseau qui calculera sa sortie  $s = (a_1,a_2,b_1,b_2)$ . Celle-ci est interprétée de la manière suivante :

Nous retenons de  $s$  deux valeurs :  $v_1=\max(a_1,a_2)$  et  $v_2=\max(b_1,b_2)$  pour associer les deux classes correspondantes au conteneur ( Haut ou Bas) et ( Avant ou Arrière). En effet, dans notre exemple, une seule classe n'est pas suffisante pour placer un conteneur, il faut donc spécifier les combinaisons : Haut-Avant, Haut-Arrière, Bas-Avant et Bas-Arrière.

Après la présentation de tous les conteneurs, la procédure peut continuer sur le sous-ensemble des conteneurs de chacune des zones en la subdivisant à son tour en quatre zones (Haut, Bas, Avant et arrière).

## Conclusion

Dans ce papier nous avons modélisé les SGTMs par un SMA afin d'identifier et employer les protocoles de négociation qui doivent nous servir pendant la distribution et la décomposition des tâches parmi les agents transporteurs. La raison principale pour laquelle nous avons proposé l'utilisation de cette approche est que les aspects distribués et adaptatifs des systèmes multi-agents permettent de bien prendre en compte les particularités du transport maritime (autonomie, alliance, coordination, négociation et coopération). En outre, nous avons développé une stratégie de négociation parmi les agents basés sur le protocole Contract-Net et déployé les algorithmes génétiques améliorés par la méthode de Recherche Tabou pour la résolution des problèmes de routage des bateaux. Pour chaque bateau nous avons résolu le problème de chargement/déchargement des conteneurs en tant que problème de satisfaction de contraintes à l'aide d'un réseau de neurones multicouches.

## Remerciements :

Ce travail est partiellement supporté par le centre national de recherche scientifique et technique marocain (CNRST). Les auteurs tiennent également à remercier le comité scientifique de RIST pour ces remarques utiles.

## **Bibliographie**

- [1] Elfazziki, A., Nejeoui, A., Ait Ouahman, A.- Agent framework for modelling management systems of maritime transport. -In : Proceedings of the 2004 AMSE international conference on modelling and simulation (ms'2004-France), 2-4 July 2004, Lyon France
- [2] Kurbel, K. and Loutchko, I.- Multi-agent negotiation under time constraints on an agent-based market place for personnel acquisition.-In : Proceedings of the 3rd International Symposium on Multi-Agent Systems : Large Complex Systems, and E-Business (MALCEB2002), October 2002, Erfurt Germany, pp.566-579.
- [3] Wooldridge, M. and Parsons, S.- Languages for negotiation.-In : Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI), 2000, Berlin Germany .
- [4] Kraus, S.- Negotiation and cooperation in multi-agent environments. -In : Artificial Intelligence, Vol.94, 1997.-PP.79-98.
- [5] Guessoum, Z. and al. -Adaptive replication of large-scale multi-agent systems: towards a fault-tolerant multi-agent platform, ACM SIGSOFT Software Engineering Notes. -In : Proceedings of the fourth international workshop on Software engineering for large-scale multi-agent systems (SELMAS '05), Vol.30, N°4, May 2005.
- [6] Liau, C. -A modal logic framework for multi-agent belief fusion. -In : ACM Transactions on Computational Logic (TOCL), Vol.6, N°1, January 2005.
- [7] Chaib-draa, B. and al. -Systèmes multi-agents : principes généraux et applications. -In : J .P. Briot and Y. Demazeau, editors, Principes et architectures des systèmes multi-agents. Collection IC2, Hermes Science Publication, 2002, Paris France.
- [8] Guessoum, Z. and al. -Monitoring and Organizational-Level Adaptation of Multi-Agent Systems. -In : Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems – Vol.2, July 2004.
- [9] Nejeoui, A., Elfazziki, A., Sadgal, M., aitouahman, A. -A Hopfield Neural Network to deal with tasks planning in a multi agent system of transport. À paraître dans WSEAS Transactions on System Vol.5, N°1, Janvier 2006.
- [10] Booch, G. -Conception Orientée Objet ET Application. -In : Addison Wesley. 1992.
- [11] Abdelaziz Elfazziki. -Une approche orientée agent pour l'analyse et la conception des systèmes de contrôle des processus industriels.-2002. Thèse de doctorat d'état en informatique. Faculté des sciences Semlalia Marrakech : septembre 2002.

- [12] Casteran, J., Glezes, M. -Des méthodologies Orientées multi-agent. –In : 8ième Journées francophones d’Intelligence Artificielle Distribuée et Syst. multi-Agents. Edition Hermès 2000.
- [13] Guessoum, Z., Ziane, M. and Faci, N. -Monitoring and Organizational-Level Adaptation of Multi-Agent Systems. –In : Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol.2, July 2004.
- [14] Xueguang, C. and Haigang, S. -Agents interactions, mobility and systems (AIMS): Further extensions of FIPA Contract Net Protocol: threshold plus DoA. –In : Proceedings of the 2004 ACM Symposium on Applied Computing, March 2004.
- [15] Reid, G. -The contract net protocol: High-level communication and control in a distributed problem solver. -In : IEEE Transactions on Computers, Vol.29 N°12, PP.1104-1113, December 1980.
- [16] Juhasz, Z. and Prasenjit, P. -Scalability Analysis of the Contract Net Protocol. -In : Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID). 2002.
- [17] Holland, J. -Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Harbor, 1975.
- [18] Osman and al. -Simulated Annealing and Descent Algorithms for Capacitated Clustering Problem. –In : Research Report, Imperial College, University of London, 1989.
- [19] Solomon, M. -Algorithms for Vehicle Routing and Scheduling Problems with Time Window Constraints. –In : Operations Research Vol.35 , N° 2, 1987 .
- [20] Eshelman, L. -A mathematical analysis of tournament selection : Genetic Algorithms. –In : Proceedings of 6th International Conference (ICGA95), Morgan Kaufmann, San Francisco, 1995.
- [21] Hong, T., Wang, H., Lin, W., Lee, W. -Evolution of Appropriate Crossover and Mutation Operators in a Genetic Process. –In : Applied Intelligence, Vol.16 , N°1, November 2001.
- [22] Hunter, A. -Crossing Over Genetic Algorithms: The Sugal Generalised GA. -In : Journal of Heuristics, Vol.4 , N°2. July 1998.
- [23] Poli, R. -Exact Schema Theory for Genetic Programming and variable-Length Genetic Algorithms with One-Point Crossover. –In : Genetic Programming and Evolvable Machines, Vol. 2 , N°2. June 2001.

- [24] Sudholt, D. -Crossover is provably essential for the Ising model on trees. –In : Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO'05), June 2005.
- [25] Hong, T., Wang, H. and Chen, W. -Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms. –In : Journal of Heuristics, Vol.6, N°4. 2000.
- [26] Muehlenbein, H. -How genetic algorithms really work: Mutation and hill-climbing. –In : Parallel Problem Solving from Nature, Vol.2, 1992, PP.15-26.
- [27] Glover, F. and Laguna, M. -Tabu Search. Kluwer Academic Publishers, 1997.
- [28] Patterson, D. -Artificial Neural Networks. Singapore: Prentice Hall, 1996.
- [29] Hertz, J., Krogh, A., Lautrup, B. and Lehmann, T. -Non-Linear Back-propagation: Doing Back-Propagation without Derivation of the activation function. –In : IEEE Transactions on Neural Networks, Vol.8, N°6, PP.1321-1327, November 1997