2005 WSEAS Int. Conf. on DYNAMICAL SYSTEMS and CONTROL, Venice, Italy, November 2-4, 2005 (pp398-404)

Solving combinatorial optimisation problems in transport multi-agent systems using Hopfield-Neural network

A. NEJEOUI, A. ELFAZZIKI, M. SADGAL, A. AITOUAHMAN
Department of Computing Science
University Cadi Ayyad, Faculty of Sciences Semlalia
Bd. Prince My. Abdellah BP 2390 Marrakech 4000
MOROCCO
http://www.ucam.ac.ma

Abstract: The tasks planning in the transport domain is a difficult problem which requires the use of analytical techniques and modelling methods resulting from the operational research, the distributed Artificial Intelligence (multi-agents systems), the decision analysis, and many other disciplines. Our contribution to this problem consists on one hand of proposing a modelling of the transport system by a multi-agents system (MAS) based on a classification model of agents to manage our different agent Subsystems (supervision subsystem, planning subsystem, ergonomic subsystem) at the same time, while keeping a global structure; and on the other hand of deploying a Fuzzy Hopfield-Neural Network model to solve the routing and scheduling problems within our planning subsystem. The computational experiments were carried out on an extended set of 300 Routing problems with 21 customers. The results demonstrate that the connexionist approach is highly competitive in term of computing time, providing the best solutions to 56% of all test instances within reasonable computing times. The power of our algorithm is confirmed by the results obtained on 21 customer problems from the literature.

Key-Words: Combinatorial optimization, vehicle routing, Hopfield-Neural Network, Simulated Annealing, Multi Agent Systems, Tasks Planning, Transport.

1 Introduction

The globalisation of the economy has completely changed the world of transport. In no time, the transport companies had to change working methods and materials in order to invest in new expensive vehicles' fleet. The tendency is under concentration:

A financial concentration, which results in fusion-takeover, and a technical concentration to avoid an overloading capacity of vehicles. Alliances and fusions within the actors of the transport are one of the globalisation phenomenon results: a company which is required for a transport service can organize it, even if its own vehicles do not serve this destination. Generally, the transport firms must restructure their management systems to meet the changes imposed by the globalisation of the economy, in order to be able to satisfy the needs of the multinational companies. This reorganization of the management systems of transport imposes a radical change of the management approach of the company in the strategic, organisational and operational levels. In this work we are interested particularly in the tasks planning problems involved in the operational level.

First we will model the transport domain by a multi-agents system based on a classification model of agents to manage our different agent Subsystems (supervision subsystem, planning subsystem, ergonomic subsystem) at the same time, while keeping a global structure [1]. This classification will enable us to identify and use the optimisation algorithms which will be useful to the planner agents during the distribution and the decomposition of tasks among the Ergonomic agents [2]. A second objective of this work is to set up an original technique for vehicles' routing problem and vehicle' distribution within the planning subsystem. It aims to develop new models for interpretation, negotiation and planning. The resolution approach that we recommend is based on a Hopfield-neural network model in order to generate solutions satisfying multiple side constraints (maximum capacity, time windows…) in real time.

The remainder of this paper is articulated around four sections. Section 2 gives a short definition of agents and multi-agents systems and describe the various subsystems of our multi agents framework. Section 3 presents the tasks planning in the transport domain, gives a description of the vehicle routing problem and an overview of the related works, in the fourth section we will present our Hopfield-Neural Network model proposed to solve the VRP within the planning subsystem; and discuss the experimental results carried out on an extended set of 300 Routing problems with 21 customers. Finally we finish this paper by a conclusion and the prospects for this work.

2 Modelling of the transport systems domain by MAS

2.1 Agent and multi-agents systems

We mean by agent, the design of an autonomous computing entity capable of arguing. Such an entity is also capable of communicating, exchanging points of view, negotiating [3] and collaborating with the other entities of its environment. Each entity is characterized by:

- Autonomy: the agent acts without the intervention of humans or other entities, and has a certain control on its internal actions and its states.
- Social ability: the agent interacts with other agents (automated agents or humans) using an agent language communication.
- Reactivity: the agent perceives its environment (which can be a physical world, a user via a graphic interface, a whole of other agents, or all these elements combined), and reacts in an opportunist way to all changes, which can occur in that environment.
- Pro-activity: the agent does not act simply with the stimuli of its environment; but it is also able to show behaviours directed by goals.

A multi-agents system (MAS) [4] is an environment, which contains several agents able to interact among them, In general, the interactions are implemented by a transfer of information among agents or the environment and the agents, either by perception, or by communication.

The principal characteristics of MAS are:

1. The heterogeneity of agents: the messages must be mutually comprehensible.
2. The exchange of knowledge: a co-operative agent within the MAS must be able to express its various kinds of knowledge.
3. Local control: the agents must be autonomous. In other word their behaviour does not have to depend neither on a central planner nor of predefined interactions.

2.2 A multi-agent framework model for the transport domain

In the transport domain we can distinguish two different kinds of physical entities: transport companies and vehicles, which can be modelled by agents. Transport companies (supervisor agents) have a fleet of vehicles (Ergonomic agent) and are brought to assign to their fleet a set of orders channelled to them in an asynchronous and dynamic way while respecting a set of constraints. Our multi agent framework is based on a classification model of agents to manage different agent Subsystems at the same time, while keeping a global structure [1]. This classification makes it possible to gather all freight management tasks and Vehicle fleet scheduling tasks, loading and unloading operations in a subsystem (Planning Subsystem); tasks relating to the transport of merchandises, in a subsystem (Ergonomic Subsystem); and tasks relating to supervision and management of different agent groups in a subsystem (Supervision Subsystem). Therefore a transport domain will be composed of the three following subsystems (see figure 1):

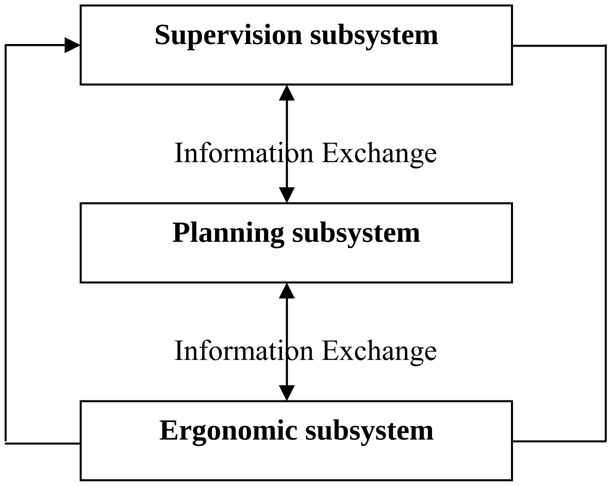

Fig.1: Classification of a Transport domain

- A supervision subsystem deals with the management of the planner agents and ergonomic agents
- A planning subsystem deals with the planning and management of freight and vehicles' fleet, and loading unloading operations; each planner agent receives orders (messages) from the supervisory agent to serve a set of customers (solve a VRP), it deals with the optimisation of itineraries (using Hopfield Neural network algorithm), it affects to each Ergonomic agent a specific tour in order to serve all customers while minimizing the total cost.
- An ergonomic subsystem (made up of vehicles with their drivers) deals with the management of pick-up and delivery tasks; it is capable of making deliveries of orders in various cities, of generating local plans, and of negotiating among the other agents to minimize the cost of the common plan. Each ergonomic agent A_k receives orders (messages) from the planner agents in the form : " transport u_i units of the goods G from the depot to the customer C^k_i. After having delivered all of orders, the ergonomic agents must turn back to the Depot of departure.

3 The tasks planning in transport domain

The tasks' planning in the transport domain constitutes a crucial problem. It consists in developing algorithms and powerful strategies for the resolution of mathematical programming problems and combinatorial optimisation problems which we can generally find in the models developed in the transport domain, and particularly the traffic assignment problem in the transport domain. The daily management of transport system involves making decisions regarding three main aspects: request clustering, vehicle routing and vehicle scheduling. Request clustering consists of creating groups of requests to be served by the same ergonomic agent because of their spatial and temporal proximity. Given these groups, vehicle routing consists of deciding the order in which the associated pickup and delivery locations should be visited by each ergonomic agent.

Finally, vehicle scheduling specifies the exact order in which each location should be visited. These decisions are obviously tightly intertwined and a proper management of the transport system calls for their optimization using real time heuristics.

In this paper, we propose a Hopfield-Neural Network heuristic to deal with vehicles' routing and scheduling problems encountered by each planner agent.

3.1 Problem description and related works

The Vehicle Routing Problem (VRP) concerns the transport of merchandises between depot and customers by means of a fleet of vehicles. The VRP can be instantiated to many real world transport domains, examples are the milk distribution, mail delivery, school bus routing, parcel pick-up and delivery and many others. In general, solving a VRP means to find the best route to service all customers using a fleet of vehicles. The solution must ensure that all customers are served, respecting the operational constraints, such as vehicle capacity and the driver's maximum working time, and minimising the total transportation cost.

3.1.1 The mathematical formulation of VRP

Given a set of n customers denoted by 1,…,n with demands D_i, i=1,…,n respectively, the nodes 0 and n+1 represent the depot. All routes start at 0 and end at n+1. A cost d_{ij} is associated with each pair of customers (i, j). The set of vehicles (Ergonomic Agents) is denoted by V={1,2,…,M}. Each ergonomic agent has a given capacity q. The decision variable X^k_{ij} (defined $\forall$ i,j = 0,…,n+1) is equal to 1 if The ergonomic agent k drives from node i to node j, and 0 otherwise.

The VRP can be stated mathematically as:

$$Minimize \;\; C(X) = \sum_{k=1}^{M}\sum_{ij=0}^{n+1} d_{ij} X_{ij}^{k} \qquad (1)$$

$$subject \;\; to \;\; :$$

$$\sum_{k=1}^{M}\sum_{j=0}^{n+1} X_{ij}^{k} = 1 \qquad \forall i = 1,\ldots,n \qquad (2)$$

$$\sum_{i=1}^{n} D_i \sum_{j=0}^{n+1} X_{ij}^{k} \le q \qquad \forall k = 1,\ldots,M \qquad (3)$$

$$\sum_{j=0}^{n+1} X_{0j}^{k} = 1 \qquad \forall k = 1,\ldots,M \qquad (4)$$

$$\sum_{i=0}^{n+1} X_{i,n+1}^{k} = 1 \qquad \forall k = 1,\ldots,M \qquad (5)$$

$$X_{ij}^{k} \in \{0,1\} \quad \forall k = 1,\ldots,M \quad \forall i,j = 0,\ldots,n+1 \qquad (6)$$

The VRP is a practical problem that has been studied widely in the literature. Several papers were published in the field in general we can distinguish between exact algorithm and approximation algorithms (heuristics).

3.1.2 Exact algorithm

The first paper proposing an exact algorithm for solving the VRP was published back in 1987 in [5]. Since then a number of papers have been published and almost all the algorithms use one of three principles:

1. Dynamic Programming.
2. Lagrange Relaxation-based methods.
3. Column Generation.

Most of the approaches rely on the solution of a shortest paths problem with additional constraints.

A different approach is described in the Ph.D. thesis [6] by Kontoravdis. The research on the VRP has been surveyed in the papers [7,8].

3.1.3 Approximation algorithms and heuristics

The field of non-exact algorithms for the VRP problem has been very active far more active than that of exact algorithms. A long series of papers has been published over the recent years. In the field of approximation algorithms and heuristics one sometimes classifies an algorithm as sequential or parallel. In a sequential algorithm one route at a time is constructed, while a parallel algorithm may build more routes at the same time. Heuristic algorithms that build a set of routes from scratch are typically called route-building heuristics, while an algorithm that tries to produce an improved solution on the basis of an already available solution is denoted route-improving.

Route-building heuristics

The first paper on route-building heuristics for the VRP is [9]. Their algorithm is an extension of the legendary Savings heuristic of Clark and Wright for the VRP problem.

Route-improving heuristics

The basis of almost every route-improving heuristic is the notion of a neighbourhood. The neighbourhood of a solution S is a set N(S) of solutions that can be generated with a single "modification" of S.

Neighbourhoods for the VRP

One of the most used improvement heuristics in routing and scheduling is the r-Opt heuristic. Here r arcs are removed and replaced by r other arcs.

In [10], Potvin and Rosseau present two variants 2-Opt and Or-Opt that maintain the direction of the route.

Simulated Annealing

Simulated Annealing was one of the first metaheuristics developed. When using simulated annealing one does not search for the best solution in the neighbourhood of the current solution. Instead one simply draws at random a solution from the neighbourhood. If the solution is better it is always accepted as a new current solution, but if the solution is worse than the present current solution it is only accepted with a certain probability [11].

Tabu Search

Just as simulated annealing, the Tabu Search heuristic is one of the "old" metaheuristics. It was introduced by Glover [12] in two papers from 1989 and 1990. At each iteration the neighbourhood of the current solution is explored and the best solution in the neighbourhood is selected as the new current solution. In order to allow the algorithm to "escape" from a local optimum the current solution is set to the best solution in the neighbourhood even if this solution is worse than the current solution. To prevent cycling visiting recently selected solutions is forbidden.

The Genetic Algorithm

Genetic Algorithms is an iterative procedure that maintains a population of K candidates (solutions). The population members can be seen as entities of artificial chromosomes (of fixed length with binary values). Each chromosome has a fitness value describing the "goodness" of the solution. Variation into the population is introduced by cross-over and mutation. In previous work [13] we have implemented a genetic based approach to deal with the VRP within a multi agent system of maritime transport.

Competitive Neural Networks

In [14] a special type of neural network called competitive neural network is used to select the seed customers. Competitive neural network is frequently used to cluster or classify data. For every vehicle we have a weight vector. Initially all weight vectors are placed randomly close to the depot. Then we select one customer at a time. For each cluster we calculate the distance to all weight vectors. The closest weight vector is updated by moving it closer to the customer. This process is repeated for all customers a number of times, each time the process is restarted the update of the weight vector becomes less sensitive.

4 Neural Networks

Neural networks grew out of research in Artificial Intelligence; specifically, attempts to mimic the fault-tolerance and capacity to learn of biological neural systems by modelling the low-level structure of the brain [15].

Neural networks have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, logistics, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, optimization, classification or control, neural networks are being introduced. This sweeping success can be attributed to a few key factors:

Power: Neural networks are very sophisticated modeling techniques capable of modeling extremely complex functions.

Ease of use: Neural networks learn by example. The neural network user gathers representative data, and then invokes training algorithms to automatically learn the structure of the data. Although the user does need to have some heuristic knowledge of how to select and prepare data, how to select an appropriate neural network, and how to interpret the results, the level of user knowledge needed to successfully apply neural networks is much lower than would be the case using (for example) some more traditional nonlinear statistical methods.

4.1 Hopfield-Neural Network

The Hopfield network is a recurrent neural network with no hidden units that belong to penalty methods for solving optimisation problems, where the weights are symmetric ($W_{ij}=W_{ji}$). The processing element is an adder followed by a threshold nonlinearity. The model can be extended to continuous units. The processing elements are updated randomly, one at a time, with equal probability (synchronous update is also possible). The condition of symmetric weights is fundamental for studying the information capabilities of this network. It turns out that when this condition is fulfilled the neurodynamics are stable in the sense of Lyapunov, which means that the state of the system approaches an equilibrium point. With this condition Hopfield [16] was able to explain that neural network' input puts the system in a point in its state space, and then the network dynamics (created by the recurrent connections) will necessarily relax the system to the nearest equilibrium point.

Now if the equilibrium points were pre-selected (for instance by hard-coding the weights), then the system could work as an associative memory. The final state would be the one closest (in state space) to that particular input. We could then classify the input or recall it using content addressable properties. In fact, such a system is highly robust to noise, also displaying pattern completion properties.

Most Hopfield net applications are in optimisation [17], where a mapping of the energy function to the cost function of the user's problem must be established and the weights pre-computed. The weights in the Hopfield network can be computed using Hebbian learning, which guarantees a stable network. Recurrent backpropagation can also be used to compute the weights, but in this case, there is no guarantee that the weights are symmetric (hence the system may be unstable). In our approach we implement the Hopfield net and train it with fixed-point learning.

4.2 Implementation

In this section we will describe The Algorithm that governs the behavior of the planner agents while they are distributing tasks among the ergonomic agents selected according to their strategic situations. The particularities of the MAS make the choice of the adequate algorithm among those cited in the literature (see paragraph 2) a difficult task due to the hard combinatorial characteristics of the VRP.

Our approach consists in implementing a new Hopfield-Neural Network model that solves the Euclidean Vehicle Routing Problem encountered by each planner agent (William Wolfe [17], and Lipo Wang [18] have applied this model to solve The well known Traveling Salesman Problem). The problem is represented as an nxn array of neurons (n is the total number of customers to be served), where states that have one maximal neural activation in each row and column correspond to a tour. The neural activations are initialized to very small random values, and then the network dynamics will drive the network into a state that corresponds to a permutation matrix.

Convergence Criteria:

If there is a unique neuron in each row and column with activation above the threshold (in most simulation runs we set threshold=0.8), then we halt the simulation and declare the corresponding permutation matrix to be the output.

It is still possible that the activations will not satisfy these criteria. So, we have set an absolute maximum number of iterations (10000 iterations). If most simulation runs reach the maximum number of iterations without satisfying the convergence criteria, then the algorithm will automatically lower the threshold to 0.60 and try again. The lower threshold may cause an occasional premature convergence, but it also make the network run faster (fewer iterations). In other words, with a higher threshold we get better tours, but it will cost a huge run time.

Fuzzy Read Out: To read the network activations at every iteration (i.e.: convert the activations into a tour even when the activations are very small and when there is no clear winner in each row and column) we used the Fuzzy Read Out (pseudo-code below) introduced by W.J. Wolfe [17]. This fuzzy approach is used to explain the networks behavior. The fuzzy interpretation consists of computing the center of mass of the positive activations in each row. This produces real numbers along the time line, one for each customer, and defines a tour in a natural way (referred to as a fuzzy tour). A fuzzy tour is computed at each iteration, and examination of such tours exposes fine features of the network dynamics. In short, this means finding the center of mass of the positive activations in each row and then ordering the customers accordingly. The center of mass calculation is simplified by finding the maximum activation, and the corresponding column index, and then doing the center of mass calculation for a window of plus or minus "base", where "base" is set to a value of 2 or 3.

```
1.  Assuming the grid indices:
columns (time stops):    i
rows (customers):        j
2. Set: base = 2
3. Center of Mass calculation:
for j = 0 to N-1 /* for each customer  calculate the max
activation and corresponding index*/
max_act = -999999999
i_max = -1
for i = 0 to N-1
    if act(i, j) > max_act then
        max_act = act(i, j)
        i_max = i
    end if
      next
// calculate the center of mass:
sum1 = 0
sum2 = 0
for k = i_max - base to i_max + base
    if a((N+k)%N, j) > 0 then
        sum1 = sum1 + k * a((N+k)%N, j)
```

```
        sum2 = sum2 + a((N+k)%N, j)
    end if
      next
/* now store the center of mass ("value") and the corresponding customer index */
center(j).value = sum1/sum2
// this is the "center of mass"
center(j).customer  = j
// this is the customer associated with that center of mass
      next
// Now SORT center by value (use a sort routine).
// After the sort, center(0).value, center(1).value, ...., center(N-1).value
//should be an increasing sequence of center of mass values,
// And, we should have a corresponding sequence of customers: center(0).customer, //center(1).customer, etc.
// So:
for j = 0 to N-1
      tour(j) = center(j).customer
        next
// That's it.  The sequence of customers in tour(j) specifies the tour.
```

4.3 Experimental results

4.3.1 Simulation Parameters:

We used an nxn grid of neurons, Number of customers: n= 21

n(i,x) is the neuron at the i^{th} column (time stop) and x^{th} row (customer).

a(i,x) is the activation of neuron(i,x).

w(i,x,j,y) is the connection strength between neuron(i,x) and neuron(j,y).

$net(i,x) = \sum_y \sum_j (w(i,x,j,y) * a(j,y))$

Note: external input = 0.

Row connections: if y = x and j != i :

$w(i,x,j,y) = 1/n^2 - 1/n$

Column connections: if y != x and j = i :

$w(i,x,j,y) = 1/n^2 - 1/n$

Self connections: if y = x and j = i :

$w(i,x,j,y) = 1/n^2 - 2/n$

Distance connections (neighboring columns):

if y != x and j = i+1 or j = i-1 :

$w(i,x,j,y) = 1/n^2 - d(x,y)/n$

All other connections: if j != i-1, i, i+1, and y != x : $w(i,x,j,y) = 1/n^2$

Maximum neural activation: M = 1

Minimum neural activation: m = -1/(n-1)

Dynamics: $a(i,x)_{t+1} = a(i,x)_t + step * (a(i,x)_t - m) * (M - a(i,x)_t) * net(i,x)_t$

Step size: step = 1.0

Initial activations: very small random numbers: between 0 and 10^{-10} (different for each neuron)

4.3.2 Results

We used a PC computational environment (x86 Family 6 Model 8 Stepping 6, AT/AT Compatible). We did 50 runs for each instance of the 300 VRPs set test, with different initialization parameters, 44% of the time the algorithm converged to an ambiguous state (no winner in each row/column after reached 10000 iterations) but 56% of the time the algorithm reach best solutions within reasonable computing times (in few iterations), the figure bellow describe the energy of our Neural Network model versus iterations with three different initialisation parameters.

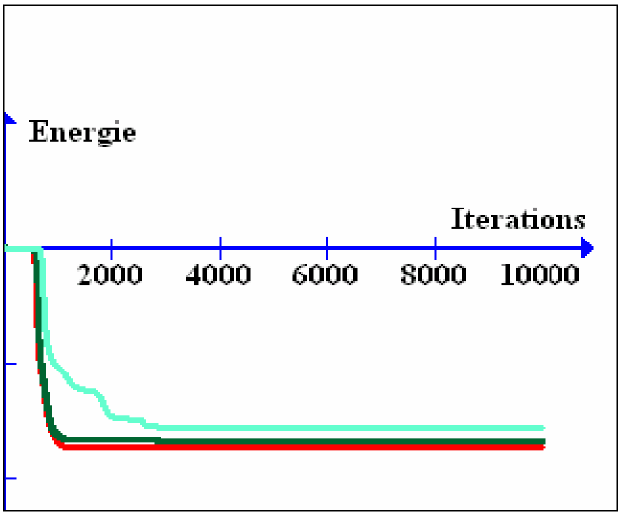

Fig.2 Energy versus Iterations

The comparison of the obtained results with our previous genetic algorithm [13] prove that the Hopfield-Neural Network model is highly competitive in term of computing time providing good results in less than 4000 iterations hence more adapted to real time decision process in multi-agent systems.

Our experiments for big instances of VRP showed that while our model may converge well to valid solutions, it may not converge to good quality solutions due to stability problems encountered in penalty methods in general.

5 Conclusion and Prospects

In this paper we have provided a multi-agent framework modeling for the transport domain based on a classification model of agents to manage the different agent Subsystems (supervision subsystem, planning subsystem, ergonomic subsystem) at the same time, while keeping a global structure and deployed a Hopfield Neural Net to deal with combinatorial optimization problems within the planning subsystem: routing and scheduling of ergonomic agents. Our Neural network model has proved its power by the results obtained on 21 customer problems from the literature providing the best solutions to 56% of all test instances within reasonable computing times, about 44% of the time the algorithm converged to an ambiguous state (no winner in each row/column after reached 10000 iterations), in this case we used the Fuzzy Read Out algorithm to convert the activations into a tour even when the activations are very small.

In further work we will investigate on the implementation of the other subsystems of our multi-agent framework using a multi-agent development kit to provide real-time simulation of agents behavior within the transport domain; and we will improve our algorithm in order to enable the planner agents to deal with big instances of VRP and to overcome the stability problems.

Acknowledgement
The authors would like to thank the WSEAS reviewers for their helpful comments. Also we thank William Wolfe and Lipo Wang for sending their recent publications. This work was partially sponsored by the Moroccan NCSTR (National Center of Scientific and technical Research).

References
[1] A. ElFazziki, M. Sadgal, une méthode d'analyse et de conception orientée agents des systèmes de contrôle de processus industriels, *JESA, Journal Européen des Systèmes Automatisées*, Vol.36, No.10, 2002, pp. 1369-1397.
[2] A. Elfazziki, A. Nejeoui, A. Ait Ouahman, Agent framework for modelling management systems of maritime transport, *Proceedings of the 2004 AMSE international conference on modelling and simulation (ms'2004-France)*, Lyon 2004.
[3] K. Kurbel and a. Loutchko, Multi-agent negotiation under time constraints on an agent-based marketplace for personnel acquisition, *Proceedings of the 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (MALCEB2002)*, Erfurt, Germany, October 2002, pp. 566-579.
[4] Z. Guessoum, M. Ziane, N. Faci, Monitoring and Organizational-Level Adaptation of Multi-Agent Systems, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems*, Vol.2, 2004.
[5] A. Kolen, A. Rinnooy Kaan, and H. Trienekens, Vehicle routing with time windows, *Operations Research*, Vol.35, No.2, March-April 1987, pp. 266-273.
[6] G. Kontoravdis, *The Vehicle Routing Problem with Time Windows*, PhD thesis, The University of Texas at Austin, August 1997
[7] L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews the state of art, *Computers & Operations Research*, Vol.10, No2, 1983, pp. 62-212.
[8] M. Solomon, J. Desrosiers, Time window constrained routing and scheduling problems, *Transportation Science*, Vol.22, No.1, February 1988, pp. 1-13.
[9] K. Baker, R. Schaffer, Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints, *American Journal of Mathematical and Management Science*, vol.6, No.3, 1986, pp. 261-300.
[10] J. Potvin, J. Rousseau, An exchange heuristic for routing problems with time windows, *Journal of the Operational Research Society*, Vol.46, No.12, 1995, pp. 1433-1446.
[11] W. Chiang, R. Russell, Simulated annealing metaheuristics for the vehicle routing problem with time windows, *Annals of Operations Research*, Vol.63, 1996, pp. 3-27.
[12] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997
[13] A. Elfazziki, A. Nejeoui, A. Ait Ouahman, Négociation et planification dans les systèmes de gestion de transport maritime : une approche multi-agents, *Proceedings of the 2005 CIRO international conference on operational research (CIRO'2005-Morocco)*, Marrakech 2005.
[14] J. Potvin, C. Robillard, Clustering for vehicle routing with a competitive neural network, *Neurocomputing*, Vol.8, 1999, pp. 125-139.
[15] D. Patterson, *Artificial Neural Networks*, Singapore Prentice Hall, 1996
[16] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceeding of National Academic Science (USA)*, Vol.79, 1982, pp. 2554-2558.
[17] W. Wolfe, A Fuzzy Hopfield-Tank Traveling Salesman Problem Model, *INFORMS Journal on Computing*, Vol.11, No.4, Fall 1999.
[18] L. Wang and al, A Noisy Chaotic Neural Network for Solving Combinatorial Optimization Problem Stochastic Chaotic Simulated Annealing, *IEEE Transactions on systems man and cybernetics Part B: Cybernetics*, Vol. 34, No.5, October 2004, pp. 2119-2125.