Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE2008
La Manga, Spain, 13-16 June 2008

Dynamic Case Based Planning

Juan F. De Paz, Sara Rodríguez, Juan M. Corchado, Javier Bajo

Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, España
Email: {fcofds, srg, corchado, jbajope}@usal.es

Abstract

This paper presents a CBP-BDI planning model which incorporates a novel artificial neural network. The CBP-BDI model, which is integrated within an agent, is the core of a Multiagent System that allows managing the security in industrial environments. The proposed model uses Self-Organized Maps to calculate optimum routes for the security guards. Besides, some technologies of Ambient Intelligence such as RFID and Wi-Fi are used to develop the intelligent environment that has been tested and analyzed in this paper.

Key words: Multiagent Systems, Case-Based Reasoning, Cased-Based planning, BDI, Ambient Inteligent, Selft-Organized Maps, RFID

1 Introduction

During the last decades, there has been an important evolution in the management of business using Artificial Intelligence techniques. But there are some aspects that still need to be improved, especially in techniques and technology for monitoring the workers activities in more efficient ways. It is necessary to establish security policies to manage risks and control hazardous events, providing better working conditions and an increase in productivity. Implementation of time control systems has a good influence in productivity, since the workers optimize their potential and enhance the process where they collaborate. The remote monitoring is becoming increasingly common in industrial scenarios, where recent studies [1] reveal that at least 3% of working shifts time is spent because of lack of time control system, allowing supervisors to observe the behaviour of remote workers and the state of facilities.

Multi-agent systems have been recently explored as supervision systems, with the flexibility to be implemented in a wide diversity of devices and scenarios, including industrial environments. This has prompted the use of ubiquitous computing [10], which constitutes the most optimistic approach to solve the challenge to create strategies that allow the anticipation and prevention of problems on automated environments [2]. The agents have several capabilities such as autonomy, learning, reasoning. They allow developing applications in dynamic and flexible environments. These capabilities can be modelled in different ways and with different tools [8], with the use of Case Based Reasoning (CBR) systems as a possibility.

This paper focuses on presenting a hybrid CBR-based (Case Based Reasoning) deliberative agent architecture BDI (Beliefs, Desires, Intentions) [13] that incorporates a specialized planning mechanism Case-Based Planning CBP [5] [15] to implement the retrieve, reuse, revise and retain stages of the CBR system. These hybrid architectures will be called CBP-BDI [7] [5]. BDI agents use mental aptitudes as beliefs, desires and intentions to develop intentional processes. A CBR system uses past experiences to resolve new problems and executes a sequential cycle composed of four stages. The integration of CBR systems within BDI agents provides a powerful tool to resolve problems in dynamic environments.

The use of wireless technologies, such as GPRS (General Packet Radio Service), UMTS (Universal Mobile Telecommunications System), RFID [6] (Radio-frequency identification) [3] , Bluetooth, etc., make possible to find better ways to provide mobile services and also give the agents the ability to communicate using portable devices (e.g. PDA's and cellular phones) [4].

In section 2 the basics of CBR systems are presented. Next, in section 3, the CBP-BDI model proposed in this work is explained in detail. Then, in section 4, a case study is presented, describing the main technologies used to schedule and surveillance routes for the security guards on industrial environments and finally, in section 5 results and conclusions are exposed.

2 CBR and CBP

Case-based Reasoning (CBR) is a type of reasoning based on the use of past experiences [9] to resolve new problems. CBR systems solve new problems by adapting solutions that have been used to solve similar problems in the past, and learn from each new experience. The primary concept when working with CBR's is the concept of case. A case can be defined as a past experience, and is composed of three elements: A problem description, which describes the initial problem; a solution, which provides the sequence of actions carried out in order to solve the problem; and the final state, which describes the state achieved once the solution was applied. A CBR manages cases (past experiences) to solve new problems. The way cases are managed is known as the CBR cycle, and consists of four sequential phases: retrieve, reuse, revise and retain.

Case-based planning (CBP) is a variation of CBR which consists of the idea of planning as remembering [5]. In CBP, the solution proposed to solve a given problem is a plan, so this solution is generated taking into account the plans applied to solve similar problems in the past. The problems and their corresponding plans are stored in a plans memory. In practice, what is stored is not only a specific problem with a specific solution, but also additional information about how the plans have been derived. The formal description of a case-based planner can be formalized as a 3-tuple <*I,G,Op*>:

- *I* is a set of formulae describing the initial state.
- *G* is a set of formulae describing the goal specification.
- *Op* is the set of operators (also called actions) that can be applied in a plan. Every action $a \in Op$ is described in terms of pre-conditions *Ca* (what has to be fulfilled in order to the action can be executed) and port-conditions *Ea* (what has to be fulfilled after the execution of the action).

A plan *P* is a tuple <*S,B,O,L*>:

- *S* is the set of plan actions. There are two special actions: t_I, those whose effects are *I*, that is, the initial state; and t_G, those actions whose pre-conditions are *G*, that is, the goal specification.
- *O* is an ordering relation on *S* allowing to establish an order between the plan actions. t_I is always the first action and t_G is the last action. If the ordering relation is total, P is a linear plan, whereas if it is a partial-order relation, *P* is a non-linear plan.
- *B* is a set that allows describing the bindings and forbidden bindings on the variables appearing in *P*.
- *L* is a set of casual links of the form $s \xrightarrow{p} s'$, where $s, s' \in S$, $p \in Es$ and $p \in Cs'$. That is, relations allowing to establish a link between plan actions.

A plan *P* constitutes the solution generated to solve a planning problem when for each action $s \neq t_I$, for each $p \in Cs$ there exists a causal link $s \xrightarrow{p} s'$ and for each action $s \neq t_G$ there exists at least a causal link $s \xrightarrow{q} s''$. In the case that the planner is interested in retaining the failures or unexpected situations during the plan, these failures or situations are represented as a set of formulae *F*.

3 CBP-BDI

Agents with BDI architecture have their origins in the *practical reasoning* of the traditional philosophy. These agents are supposed to be able to decide in each moment what action to execute according to their objectives. The practical reasoning undergoes two phases: in the first one the goals are defined and in the second one it is defined how to achieve such goals A representation based on an action requires an agent architecture in which the way to acquire and process the knowledge of the world, at the reasoning stage, is closely related to the way in which plans are constructed and used, in the phase of execution. In this section, we show how such a requirement can be achieved through a BDI agent model [7][13]. The terminology used is the following.

- The environment M and the changes that are produced within it, are represented from the point of view of the agent. Therefore, the world can be defined as a set of variables that influence a problem faced by the agent

$$M = \{\tau_1, \tau_2, \cdots, \tau_s\} \text{ with } s < \infty \quad (1)$$

- The beliefs are vectors of some (or all) of the attributes of the world taking a set of concrete values

$$B = \{b_i / b_i = \{\tau_1^i, \tau_2^i, \cdots, \tau_n^i\}, n \le s \quad \forall i \in N\}_{i \in N} \subseteq M \quad (2)$$

- A state of the world $e_j \in E$ is represented for the agent by a set of beliefs that are true at a specific moment in time t.
  Let $E = \{e_j\}_{j \in N}$ set of status of the World if we fix the value of t then

$$e_j^t = \{b_1^{jt}, b_2^{jt}, \cdots b_r^{jt}\}_{r \in N} \subseteq B \quad \forall j, t \quad (3)$$

- The desires are imposed at the beginning and are applications between a state of the current world and another that it is trying to reach

$$d : \underset{e_0}{E} \underset{\rightarrow}{\rightarrow} \underset{e^*}{E} \quad (4)$$

- Intentions are the way that the agent's knowledge is used in order to reach its objectives. A desire is attainable if the application i, defined through n beliefs exists:

$$i : \underset{(b_1, b_2, \cdots\cdots\cdots, b_n, e_0)}{\overset{n)}{BxBx \cdots xBxE}} \underset{\rightarrow}{\rightarrow} \underset{e^*}{E} \quad (5)$$

  In our model, intentions guarantee that there is enough knowledge in the beliefs base for a desire to be reached via a plan of action.
- We define an agent action as the mechanism that provokes changes in the world making it change the state,

$$a_j : \underset{e_i}{E} \underset{\rightarrow}{\rightarrow} \underset{a_j(e_i) = e_j}{E} \quad (6)$$

- Agent plan is the name we give to a sequence of actions that, from a current state e_0, defines the path of states through which the agent passes in order to reach the other world state.

$$p_n : \underset{e_0}{E} \underset{\rightarrow}{\rightarrow} \underset{p_n(e_0) = e_n}{E}$$
$$p_n(e_0) = e_n = a_n(e_{n-1}) = \cdots = (a_n \circ \cdots \circ a_1)(e_0) \quad p_n \equiv a_n \circ \cdots \circ a_1 \quad (7)$$

Below, the attributes that characterise the plans in the case base are presented, which allow us to relate BDI language with the interest parameters within a CBP. A constraint satisfaction problem (CSP) planning problem is considered in order to lend the model generality. These kinds of problems do not only search for solutions but also have to conform to a series of imposed restrictions. Based on the theory of action, the set of objectives for a plan and the resources available are selected as a variable upon which the CSP problems impose the restrictions. A plan *p* is expressed as *p=<E, O, O', R, R'>,* where:

E is the environment, but it also represents the type of problem faced by the agent, characterised by $E = \{e_0, e^*\}$, where e_0 represents the starting point for the agent when it begins a plan, and e^* is the state or states that it is trying to attain.
O indicates the objectives of the agent and O' are the results achieved by the plan. R are the total resources and R' are the resources consumed by the agent. Table 1 shows the indicators derived from the attributes described above, used to identify and contrast the quality of the different plans (# means cardinal of a set).

Indicator	Formulae
Efficacy of the plan: relationship between objectives attained and objectives proposed	$E_f = \frac{\#(O' \cap O)}{\#O}$
Cost of the plan: relationship between the resources used and the resources available	$C = \frac{\#R'}{\#R}$
Efficiency of the plan: relationship between the objectives attained and the resources consumed	$E_{ff} = \frac{\#(O' \cap O)}{\#R'}$

Table 1. Indicators of plan quality.

If a problem $E= \{e_0, e_1\}$ has been defined, a plan p to solve the problem can be characterised by the relationships between the objectives reached and the resources consumed between both states. The general functioning process is derived by following the typical phases of a case based system [14](eliminating the revision phase, since it can be external to the system needing the intervention of an expert). The reasoning process of this kind of system carries out the following four sequential stages (noticing that the revision stage has been eliminated: it usually is carried out by an expert and is external to the system):

- **Retrieval:** Given a state of the perceived world e_0 and the desire that the agent encounters in a state $e_0 \neq e^*$, the system searches in the case base for plans that have resolved similar problems in the past.
- **Adaptation/Reuse:** From the previous phase, a set of possible solutions for the agent $\{p_1,\ldots, p_n\}$ is obtained. In this phase, in accordance with the planning model G, the system uses the possible solutions to propose a solution p^* (8).

$$G(e_0, p_1, \cdots, p_n) = p^* \quad (8)$$

- **Learning/Retain:** The plan proposed may achieve its objective or fail in its development. The information on the quality of the final plan in the $w_f(p^*)$ cycle is stored for the future and is directly proportional (i) to the initial value of $w_i(p^*)$, and (ii) to the "rate of use" $\alpha(N)$, where N is the number of times that the plan has been used in the past.

$$w_f(p^*) = w_i(p^*)\alpha(N) \quad (9)$$

The model proposed conforms to the conditions required in order to obtain a representation and reasoning based on the action [16]. The capabilities of the hybrid system restrict what kind of plans can be generated. Plans structure and world representation can be easily adapted to a wide range of problems.

4 Case Study

A multi-agent system has been developed to provide control over the activities performed by the staff responsible for overseeing the industrial environments. The agents in the system calculate the surveillance routes for the security guards depending on the working shifts, the distance to be covered in the facilities and the security guards available. Considering this latter feature, the system has the ability to re-plan the routes automatically. A supervisor can set the possible routes, defining the areas that must be supervised. It is also possible to track the workers activities (routes completion) over the Internet.

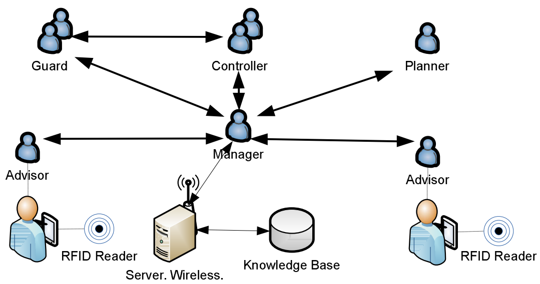

Figure 1. System Structure

Radiofrequency Identification (RFID) is a key technology in this development. It can be used to electronically identify, track, and store information about products, items, components or people. Once defined the system structure, shown on Figure 1, it is possible to define the five different kinds of agents:

- Guard Agent. It is associated to each PDA. Manages the portable RFID readers to get the RFID tags information on every control point. Communicates with Controller Agents to check the accomplishment of the assigned surveillance routes, to obtain new routes, and also to send the RFID tags information via Wi-Fi.
- Manager Agent. Controls the rest of agents in the system. Manages the connection and disconnection of Guard Agents to determine the available security guards available. The information is sent to the Planner Agent to generate new surveillance routes. It also receives incidences (omitted control points, route changes, new security guard connected/disconnected, security guards notifications, etc.) from the Controller Agents and Guard Agents and, depending on its priority, informs the Advisor Agent. Manager Agent stores all the system information (incidences, time, data, control points, route status, etc.) into a database.
- Planner Agent. Generates automatically the surveillance routes which are sent to the Manager Agent to distribute them among the security guards.
- Controller Agent. Monitors the security guards activities by means of the control points checked. Once a surveillance route is generated by the Planner Agent, the average time to reach each control point is calculated.

The Controller Agent also handles the associated route incidences and sends them to the Manager Agent.

- Advisor Agent. Administers the communication with the supervisors (person). Receives from the Manager Agent the incidences, and decides if is sent to the supervisor. Incidences can be sent via Wi-Fi, SMS or GPRS.

The agents of the system react to the events in the environment. The most important agent in the system is the Planner agent, which incorporates the CBP-BDI model. Table 2 shows the structure for a plan.

Task Field	Field Type
taskList	ArrayList of Route
numberAgents	Integer

Table 2. Task structure

The information stored for each route is shown in Table 3.

Task Field	Field Type
taskList{Position, Estimated Arrive, Arrive}	ArrayList of Task

Table 3. Route

The variation of the agent plan $p_A(t)$ will be provoked essentially by: the changes that occur in the environment and that force the initial plan to be modified, and the knowledge from the success and failure of the plans that were used in the past, and which are favoured or punished via learning. The planning is carried out through a neural network based on the Kohonen network [12]. Each of the phases of the CBP-BDI planner are explained in detail in the following sub-sections:

4.1 Retrive

In this phase the most similar plan resolved in the past including all the control points indicated in the new problem is recovered. The information of the plan is given for the following record.

$$< T = \{x_i / x_i = (x_{i1}, x_{i2}), i = 1...n\}, g > \qquad (10)$$

Being x_i the control point i that it will be visited, (x_{i1},x_{i2}) the coordinates of point i and g the number of security guards. The routes r_i recovered follow the equation.

$$R = \{r_i\}\ i = 1...g \text{ where } r_i \subseteq T, r_i \cap r_j = \phi \forall i \neq j\ j = 1...g \qquad (11)$$

4.2 Reuse

If $R = \{\phi\}$ or the user establishes that he wishes to make a new distribution of the routes, the system will create a new allocation for the control points among routes. The following algorithm allows distributing the points. For surveillance routes calculation, the system takes into account the time and the minimum distance to be covered. So it is necessary a proper control points grouping and order on each group. The planning mechanism uses Kohonen SOM (Self Organizing Maps) neural networks with the k-means learning algorithm [11] to

calculate the optimal routes and assign them to the available security guards. Neural networks allow the calculus of variable size data collections, and reduce the time and distances to be covered. The distribution of the control points must follow the equation. In addition, the control points can be changed on each calculation, so the surveillance routes are dynamic, avoiding repetitive patterns.
Once the distribution of the points among routes r_i has been made, the CBP-BDI starts spreading the control points among the available security guards. Then, the optimal route for each one is calculated using a modified SOM neural network. The modification is done through a FYDPS neural network, changing the neighbourhood function defined in the learning stage of the Kohonen network. The new network has two layers: IN and OUT. The IN layer has two neurons, corresponding the physical control points coordinates. The OUT layer has the same number of control points on each route [10]. Be $x_i \equiv (x_{i1}, x_{i2}) \quad i = 1, \cdots . N$ the i control point coordinates and $n_i \equiv (n_{i1}, n_{i2}) \quad i = 1, \cdots, N$ the i neuron coordinates on $\Re^2$, being N the number of control points in the route. So, there are two neurons for the IN layer and N neurons for the OUT layer. The weight actualization formula is defined by the following equation:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t) g(k,h,t)(x_i(t) - w_{ki}(t)) \qquad (12)$$

Be w_{ki} the weight that connect the IN layer i neuron with the OUT layer k neuron. t represents the interaction; $\eta(t)$ the learning rate; and finally, $g(k,h,t)$ the neighbourhood function, which depends on three parameters: the winner neuron, the actual neuron, and the interaction.
A decreasing neighbourhood function is considered with the number of interactions and the winner neuron distance.

$$g(k,h,t) = Exp\left[\left(-\frac{|k-h|}{N/2} \right)^{\frac{\underset{\substack{i,j \in \{1,\cdots,N\} \\ i \neq j}}{Máx} \{f_{ij}\} - \sqrt{(n_{k1}-n_{h1})^2 + (n_{k2}-n_{h2})^2}}{\underset{\substack{i,j \\ i \neq j}}{Máx} \{f_{ij}\}}} - \lambda \frac{|k-h| t}{\beta N} \right] \qquad (13)$$

λ and β are determined empirically. The value of λ is set to 1 by default, and the values of β are set between 5 y 50. t is the current interaction. Its value is obtained by means of β N. $Exp[x] = e^x$, where N is the number of control points. f_{ij} is the distance between two points i and j. Finally, $Max\{f_{ij}\}$ represents the maximum distance that joins those two points.
To train the neural network, the control points groups are passed to the IN layer, so the neurons weights are similar to the control points coordinates. When all the process concludes, there is only one neuron associated to each control point. To determine the optimal route, the i neuron is associated with the i+1 neuron, from i=1, 2, …, N, covering all the neurons vector. A last interval is added to complete the route, associating the N neuron with the i neuron. The learning rate depends on the number of interactions, as can be seen on the following equation:

$$\eta(t) = Exp\left[-4\sqrt{\frac{t}{\beta N}}\right] \qquad (14)$$

The neurons activation function is the identity. When the learning stage ends, the winner neuron for each point is determined, so each point has only one neuron associated. The optimal route is then calculated following the weights vector. This vector is actually a ring, where the n_1 neuron is the next n_N neuron. Initially considering a high neighbourhood radius, the weights modifications affect the nearest neurons. Reducing the neighbourhood radius, the number of neurons affected decrease, until just the winner neuron is affected.

The initial number of interactions is $T_1 = \beta N$ in the first stage. When $t = \beta N$, the weights of the possible couple of neurons are changed from the neurons ring obtained. If the distance is optimized, the number of interactions is reduced to continue the learning. In the Z phase, the total number of interactions is:

$$T_Z = T_{Z-1} - \frac{T_{Z-1}}{Z} \qquad (15)$$

The objective of these phases is to avoid the crossings. Once all interactions are concluded, the distance obtained is analyzed to determine if it is the optimal distance. So, the recoil in the number of interactions is reduced each time, obtaining a maximum number of interactions, although the value is variable. Figure 2 shows the routes calculated for one and two security guards.

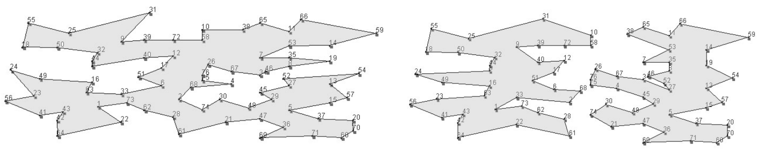

Figure 2. Planned routes for one (left) and two (right) security guards

4.3 Retain and Revise

In this case study, the final routes for the users were stored when they were successfully completed.

5 Results and Conclusions

The system presented on this paper has been implemented and tested over experimental and controlled scenarios. Simulations have been done to calculate surveillance routes and monitor the accomplishment of each one. The results obtained have shown that it is possible to find out the necessary number of security guards depending on the surveillance routes calculated by the system.

To evaluate the system efficiency, a comparison after and before the prototype implementation was done, defining multiple control points sets and just one security guard. The results of times and distances calculated by the users and the system are shown on Figure 3.

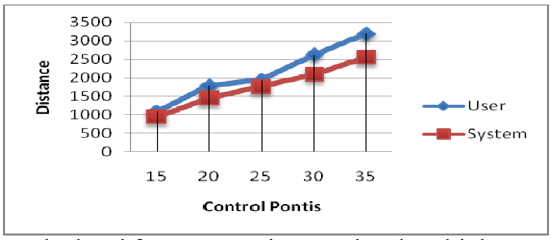

Figure 3.Distance calculated for one security guard and multiple control points sets

The system provides optimized calculations, so the time and distance are reduced. A complete working day shift can be fixed according the system results, for example, if the route calculated is too long or the time exceeds eight working hours, a new guard must be incorporated. The usage of a CBP-BDI agent allows the system to increase its performance since the ANN facilitates automatic route's calculation. The plans are more suitable to the user's skills because the planner takes into account their profiles and the results obtained in previous experiences, so they have a more realistic estimation of times to go between control points. Moreover, the CBP-BDI allows reducing the number of preplanning in the system. The planner with ANN only calculates the routes when it is necessary to replan or the system doesn't have any similar case in the memory. The administrator is able to redistribute the control points whenever he wishes. In this way the system avoids over-learning. In Figure 4b it is possible to see how the percentage of variation for the routes related to the increase the weeks. In Figure 4a shows the average number of estimated security guards needed to cover an entire area, which consisted on a mesh from 20 to 100 control points, with an increment of 5 control points. The results are clear, for example, for 80 control points, the users estimated 4 security guards, but the system recommended only 3.

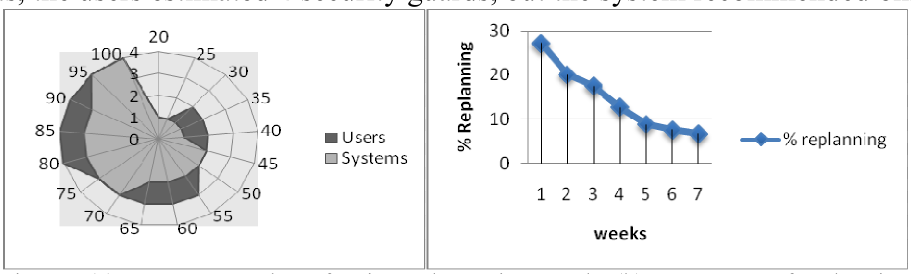

Figure 4(a). Average number of estimated security guards. (b) Percentage of replanning

The results obtained so far are positive. It is possible to determine the number of security guards needed to cover an entire area and the loops in the routes, so the human resources are optimized. In addition, the system provides the supervisors relevant information to monitor the workers activities, detecting incidences in the surveillance routes automatically and in real-time. The system presented can be easily adapted to other scenarios with similar characteristics.

Acknowledgements

This development has been supported by the projects MEC TIN2006-14630-C03-03 and JCYL-2002-05 (SA104A05). Special thanks to Telefónica Móviles and Sokymat for the technology provided.

6 References

[1]INOLOGY, PRESS NOTE, *Available: http://www.controldetiempos.com/sala_de_prensa.htm#absentismo.* (2005)

[2]C ANGULO AND R TELLEZ, *Distributed Intelligence for smart home appliances,* Tendencias de la minería de datos en España, Red Española de Minería de Datos, 2004.

[3]SOKYMAT.: SOKYMAT. *http://www.sokymat.com. (2006)*

[4]P RIGOLE, T HOLVOET AND Y BERBERS, *Using Jini to integrate home automation in a distributed software-system,* Computational Sciences Department, Belgium, (2002).

[5]GLEZ-BEDIA, M., CORCHADO, J.M.: *A planning strategy based on variational calculus for deliberative agents.* Computing and Information Systems Journal. Vol.10(1) 2-14. (2002)

[6]S GARFINKEL AND B ROSENBERG, *RFID: Applications, security, and privacy,* Addison-Wesley Professional, (2005), pp. 15-36.

[7]CORCHADO J. M. AND LAZA R.. *Constructing Deliberative Agents with Case-based Reasoning Technology,* International Journal of Intelligent Systems. Vol 18, No. 12, December. (2003) pp.: 1227-1241

[8]M WOOLDRIDGE AND N R JENNINGS, *Agent Theories, Architectures, and Languages: a Survey, In: Wooldridge and Jennings ed.,* Intelligent Agents, Springer-Verlag, (1995), pp. 1-22.

[9]KOLODNER J. *Case-Based Reasoning. Morgan Kaufmann (1993).*

[10] Q. MARTÍN, M T SANTOS AND Y DE PAZ, *Operations research: Resolute problems and exercises,* Pearson, (2005), pp. 189-190.

[11] N JENNINGS AND M WOOLDRIDGE, *Applications of Intelligent Agents, Queen Mary & Westfield College,* University of London, (1998).

[12] K S LEUNG, H D JIN AND Z B XU, *An expanding Self-organizing Neural Network for the Traveling Salesman Problem.* Neurocomputing, Vol. 62, (2004), pp. 267-292.

[13] BRATMAN, M.. *Intention, Plans and Practical Reason. Harvard U.P.,* Cambridge. (1987)

[14] ALLEN, J.F.: *Towards a General Theory of Action and Time. Artificial Intelligence* Vol. 23 (1984) 123-154

[15] SPALAZZI L. A *Survey on Case-Based Planning.* Artificial Intelligence Review, Vol. 16, Issue 1 (2001) 3-36

[16] POLLACK, M.E.: *The Uses of Plans. Artificial Intelligence.* Vol. 57 (1992) 43-68