SCISPACE
formerly Typeset

Open access • Journal Article • DOI:10.1080/00207160802562556

Mathematical model for dynamic case-based planning — Source link

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

Institutions: University of Salamanca

Published on: 01 Nov 2009 - International Journal of Computer Mathematics (Taylor & Francis)

Topics: Case-based reasoning, Multi-agent system, Intelligent environment and Ambient intelligence

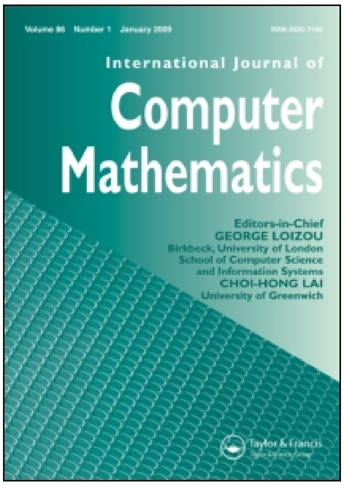
Related papers:

- Multi-agent system to monitor oceanic environments
- An execution time neural-CBR guidance assistant
- Ovamah: Multiagent-based adaptive virtual organizations
- Adaptation of cases for case based forecasting with neural network support
- Case-based reasoning as a decision support system for cancer diagnosis: A case study

Share this paper:

View more about this paper here: https://typeset.io/papers/mathematical-model-for-dynamic-case-based-planning-2co49sld4n

This article was downloaded by: *[B-on Consortium - 2007]*
On: *24 September 2009*
Access details: *Access Details: [subscription number 908038079]*
Publisher *Taylor & Francis*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713455451

Mathematical model for dynamic case-based planning

J. F. De Paz [a]; S. Rodríguez [a]; J. Bajo [a]; J. M. Corchado [a]
[a] Departamento de Informática y Automática, Universidad de Salamanca Plaza de la, Salamanca, Spain

Online Publication Date: 01 October 2009

To cite this Article De Paz, J. F., Rodríguez, S., Bajo, J. and Corchado, J. M.(2009)'Mathematical model for dynamic case-based planning',International Journal of Computer Mathematics,86:10,1719 — 1730

To link to this Article: DOI: 10.1080/00207160802562556

URL: http://dx.doi.org/10.1080/00207160802562556

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

International Journal of Computer Mathematics
Vol. 86, Nos. 10–11, October–November 2009, 1719–1730

Taylor & Francis
Taylor & Francis Group

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

Mathematical model for dynamic case-based planning

J.F. De Paz*, S. Rodríguez, J. Bajo and J.M. Corchado

Departamento de Informática y Automática, Universidad de Salamanca Plaza de la, Salamanca, Spain

(*Received 10 August 2008; revised version received 07 October 2008; accepted 14 October 2008*)

This paper presents a case-based planning and beliefs, desires, intentions (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 multi-agent system that allows managing the security in industrial environments. The BDI model integrates within a CBP engine of reasoning that incorporates artificial neural network-based techniques, and in this way it is possible to adapt past experiences to generate new plans. The proposed model uses self-organized maps to calculate optimum routes for the security guards. Besides, some technologies of ambient intelligence such as radio-frequency identification and Wi-Fi are used to develop the intelligent environment that has been tested and analysed in this paper.

Keywords: multi-agent systems; case-based reasoning; cased-based planning; beliefs, desires, intentions; ambient inteligent; self-organized maps; RFID

2000 AMS Subject Classifications: 68T05; 68T20; 68T27; 68T37; 68W15

1. Introduction

During the last decades, there has been an important evolution in the management of business using artificial intelligence (AI) techniques. However, there are some aspects that still need to be improved, especially in techniques and technology for monitoring the workers activities. Remote monitoring is becoming increasingly common in industrial scenarios, where recent studies reveal that at least 3% of working shifts time is spent because of lack of time control system.

Multi-agent systems (MAS) [3,9] 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, which constitutes the most optimistic approach to solve the challenge to create strategies that allow the anticipation and prevention of problems on automated environments [17]. In these environments the use of wireless technologies, such as general packet radio service (GPRS), universal mobile telecommunications system (UMTS), Radio-frequency identification (RFID) [15], Bluetooth, etc., make it possible to find better ways to provide mobile services and also give the agents the ability to communicate using portable devices (e.g. personal digital assistants (PDA's) and cellular phones) [12].

*Corresponding author. Email: fcofds@usal.es

ISSN 0020-7160 print/ISSN 1029-0265 online
© 2009 Taylor & Francis
DOI: 10.1080/00207160802562556
http://www.informaworld.com

In the field of AI mathematical models are frequently used for determining the environment and actions to carry out depending on the objectives of the system. This paper presents an MAS which incorporates a special type of intelligent agent characterized for an internal structure that integrates a mathematical model based on a symbolic computational model derived from the CBP–BDI [5,6] (case-based planning) [5,14,16] (beliefs, desires, intentions) [10] model. Moreover, the phases of the CBP system incorporate a sub-symbolic model, based on artificial neural networks (ANN), for resolving the problems at a low level of detail.

In Section 2, the CBP–BDI model proposed in this work is explained in detail. Then, in Section 3, 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 Sections 4 and 5 results and conclusions are exposed.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

2. CBP–BDI model for generating routes

The problem of generating routes in industrial environments is a highly dynamic problem that requires intelligent systems with great capacity for learning and adaptation. The case-based reasoning (CBR) systems are based on a model where past experiences are used to solve new problems. In this sense, they are very appropriate to be used in changing environments, since they are able to adapt themselves to changes in the environment using memories.

CBR is a type of reasoning based on the use of past experiences [2] 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 CBRs 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 past experiences to solve new problems. The way cases are managed is known as the CBR cycle, and consists of four phases: retrieve, reuse, revise and retain.

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 memory of plans.

The BDI model proposed in this work integrates within a CBP engine of reasoning that incorporates ANN-based techniques, and, in this way, it is possible to integrate both the symbolic and sub-symbolic models. BDI-based agent are supposed to be able to decide in each moment what action to execute according to their objectives. The terminology used for a BDI agent model [6,10] is the following.

(1) The environment or world M and the changes that are produced, it can be defined as a set of variables that influence a problem faced by the agent

$$M = \{\tau_1, \tau_2, \ldots \tau_n\} \quad \text{with} \quad s < \infty. \tag{1}$$

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

$$B = \left\{ \frac{b_i}{b_i} = \{\tau_1^i, \tau_2^i, \ldots, \tau_n^i\}, n \leq s \forall i \in N \right\}_{i \in N} \subseteq M. \tag{2}$$

(3) 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

$$e_j^t = \{b_1^{jt}, b_2^{jt}, \ldots, b_r^{jt}\}_{r\in N} \subseteq \quad \forall j, t. \tag{3}$$

(4) The desires are the applications between a state of the current world and another that it is trying to reach

$$\begin{aligned} d : E &\longrightarrow E \\ e_0 &\longrightarrow e^*. \end{aligned} \tag{4}$$

(5) 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

$$\begin{aligned} i : \overbrace{BxBx\ldots xB}^{n}xE &\longrightarrow E \\ (b_1, b_2, \ldots, b_n, e_0) &\longrightarrow e^*. \end{aligned} \tag{5}$$

(6) We define an agent action as the mechanism that provokes changes in the world making it change the state

$$\begin{aligned} a_j : E &\longrightarrow E \\ e_i &\longrightarrow a_j(e_i) = e_j. \end{aligned} \tag{6}$$

(7) 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

$$\begin{aligned} p_n : E &\longrightarrow E \\ e_0 &\longrightarrow p_n(e_0) = e_n \end{aligned} \tag{7}$$

$$p_n(e_0) = e_n = a_n(e_{n-1}) = \cdots = (a_n \circ \cdots \circ a_1)(e_0) \quad p_n = a_n \circ \cdots \circ a_1.$$

Below, the attributes that characterise the plans for a CBP–BDI agent in the case base are presented, which allow us to relate BDI model with the interest parameters within a CBP. 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 constraint satisfaction problems impose the restrictions. A plan p is expressed as $p = \langle E, O, O', R, R'\rangle$, where: E is the environment, but it also represents the type of problem faced by the agent, characterized 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.

If a problem $E = \{e_0, e_1\}$ has been defined, a plan p to solve the problem can be characterized 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 [1,4].

(1) 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.
(2) 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

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

solutions to propose a solution p^* (8). To carry out this phase, different AI techniques can be used, but we will focus on the sub-symbolic models. In the case study we will focus on the ANNs.

$$G(e_0, p_1, \ldots, p_n) = p^* \tag{8}$$

(3) Learning/retain: the plan proposed may achieve its objective or fail. The information on the quality of the final plan is represented as $w_{\mathrm{f}}(p^*)$ and is 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.

$$w_{\mathrm{f}}(p^*) = w_i(p^*)\alpha(N) \tag{9}$$

Next, a case study is presented, describing the main technologies used to schedule and monitor security guards surveillance routes on industrial environments.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

3. Case study

An MAS 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. It is possible to track the workers activities (routes completion) over the internet. RFID is a key technology in this development.

The MAS is defined by the five different kinds of agents:

(1) *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 routes, to obtain new routes, and also to send the RFID tags information via Wi-Fi.
(2) *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.
(3) *Planner agent*. Generates automatically the surveillance routes which are sent to the manager agent to distribute them among the security guards.
(4) *Controller agent*. Monitors the security guards activities by means of the control points checked.
(5) *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. In order to adapt the CBP–BDI model to the problem of security in industrial spaces, the environment (1) has been defined through the variables: security guards available, coordinates for every control point, arrival time, initial time, final time and service time. The current state (3) is obtained through the number of available security guards, their corresponding control points at that moment and the time. The desires (4) are represented as the surveillance route that allows to cover all the control points in the minor time bearing in mind the time restrictions. The intentions (5) are given for the neural networks that establish the sequence of states through which the system pass in order to reach the final state in which the surveillance routes have been successfully completed. Equations (10) and (11) show the structure for a plan (7).

The planning is carried out through a neural network based on the Kohonen network [13]. Each of the phases of the CBP–BDI planner are explained in detail in the following sub-sections.

3.1 *Retrieve*

In this phase the most similar plans resolved in the past including all the control points indicated in the new problem are recovered. The information of the plan is given for the following records (10) and (11).

$$\left\langle T = \left\{ \frac{(x_i, a_i)}{x_i} = (x_{i1}, x_{i2}), i = 1, \ldots, n \right\}, g \right\rangle \tag{10}$$

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, a_i arrival time. The routes r_i recovered follow the equation.

$$R = \{r_i\} \quad i = 1, \ldots, g \quad r_i \subseteq T, r_i \cap r_j = \phi \quad \forall i \neq j \quad j = 1, \ldots, g \tag{11}$$

3.2 *Reuse*

In this phase, those retrieved routes, represented as R, are adapted to the temporal restrictions stated in the problem description. The problem description of the problem is given by the control points to visit and the initial and final time to arrive to reach them (22). If $R = \{\}$, or the user establishes that he wishes to make a new distribution of the routes, or the time restrictions are incompatible with the experiences previously stored, the system will create a new allocation for the control points among routes. If $\#R < g$, then only one new distribution is generated for those control points not associated with any of the routes, and this is done for the $g - \#R$ pending routes. 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 self organizing maps (SOM) [11] neural networks with the k-means learning algorithm to calculate the optimal routes and assign them to the available security guards. The inputs of the SOM are $x_i = (x_{i1}, x_{i2})$ $i = 1, \ldots, N$, the i control point coordinates and N the number of control points in the route, W_{kj} is the weight of the neuron k of the output layer that connects with the neuron j in the input layer. Once the input and output are established, the k-means algorithm is carried out to create a new allocation of the control point:

(1) Establish the number k of initial groups. Initiate the weights in the output layer with the k initial patterns. $W_{ij} = x_{ij}$.
(2) Establish for each of the patterns the nearest neuron of the output layer and associate the pattern with it. The distance used is euclidean. Q_k represents the set of input patterns associated with the neuron of the output layer k.

$$Q_k = \left\{ \frac{x_i}{d(W_k, X_i)} \leq d(W_\mathrm{r}, X_i) \forall k \neq r \right\} \quad d(W_\mathrm{r}, X_i) = \|W_\mathrm{r} - X_i\| \tag{12}$$

(3) Calculate the new centroids of the neurons of the hidden layer as the average of the input associated patterns.

$$W_{kj} = \frac{1}{\#Q_k} \sum x_{\mathrm{s}j} \quad \text{with} \quad x_\mathrm{s} \in Q_k \tag{13}$$

(4) Repeat from step 2 until the modification of the centroids will be minor than α.

$$\sum \Delta W_k = \sum \|W_k(t) - W_k(t-1)\| < \alpha \tag{14}$$

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

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. There exits another methods which allow calculating optimal routes, among them we can enumerate: genetic algorithms (GA), integer linear programming, Lin Kernighan Heuristic [8]. However, it is difficult to take into account time restriction in these heuristics, only the GAs are easily adaptable to this situation.

The 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 [7]. Be $x_i \equiv (x_{i1}, x_{i2})$ $i = 1, \ldots, N$ the i control point coordinates and $n_i \equiv (n_{i1}, n_{i2})$ $i = 1, \ldots, N$ the i neuron coordinates on $\Re^2$, being N the number of control points in the route. The weight actualization formula is defined by the following equation:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t) g(k, h, t)(x_i(t) - w_{ki}(t)) \tag{15}$$

where w_{ki} is the weight that connects 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 current neuron and the interaction. A decreasing neighbourhood function is considered with the number of interactions and the winner neuron distance.

$$g(k, h, t) = \mathrm{Exp}\left[-\frac{|k-h|}{N/2} \frac{\sqrt{(n_{k1}-n_{h1})^2 + (n_{k2}-n_{h2})^2}}{\mathrm{Max}_{i,j \in (1,\ldots,N) i \neq j}\{f_{ij}\}} - \lambda \frac{|k-h| t}{\beta N} \right] \tag{16}$$

α and β and are determined empirically. The value of α is set to 1 by default, and the values of β are set between 5 and 50, t is the current interaction. Its value is obtained by means of βN, N is the number of control points, f_{ij} is the distance between two points i and j and $\mathrm{Max}\{f_{ij}\}$ 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. To determine the optimal route, the i neuron is associated with the $i+1$ neuron, from $i = 1, \ldots, N$, covering all the neurons vector. The learning rate depends on the number of interactions, as can be seen on the following equation:

$$\eta(t) = \mathrm{Exp}\left[-\left(\frac{t}{BN} \right)^{1/4} \right]. \tag{17}$$

The neurons activation function is the identity. 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:

$$T_z = T_{z-1} - \frac{T_{z-1}}{Z}. \tag{18}$$

The objective of these phases is to avoid the crossings. Figure 1 shows the routes calculated for one and two security guards.

To allow resolving optimization problems according to the temporal restrictions imposed by the supervisor, it is necessary to modify the previously explained SOM networks. The restrictions that must be considered are: service time employed for a security guard in checking a control point, initial time (if security guard arrives before this time he will wait for this time) and final

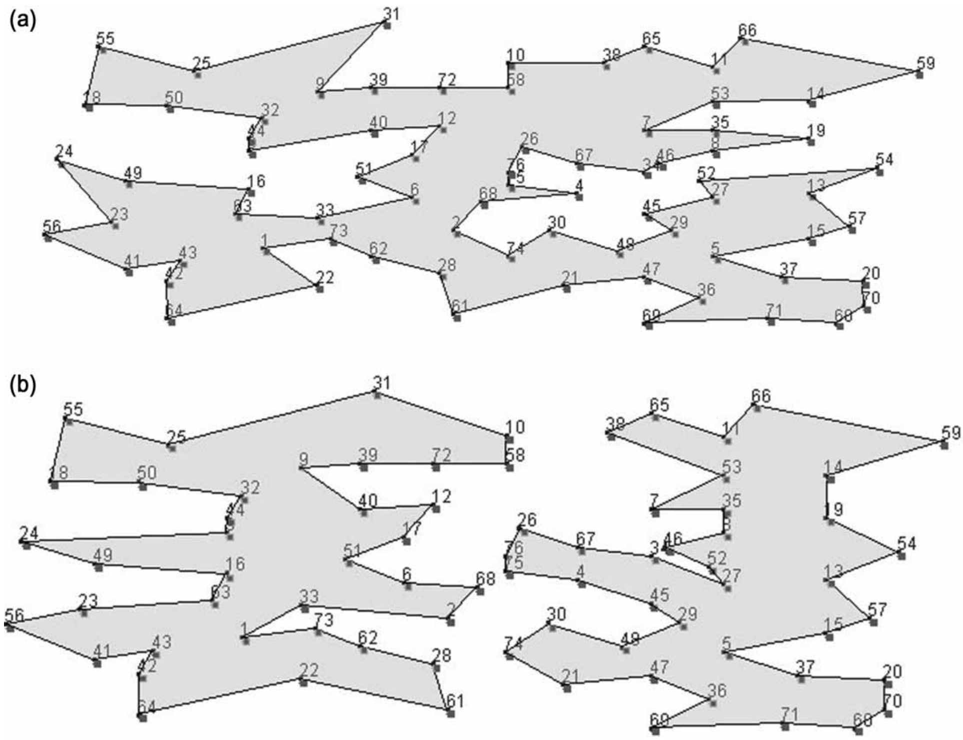


Figure 1. Planned routes for one (a) and two (b) security guards.

time limit arrival hour. The coordinates have been scaled so that the space journey through time unit is also a unit. This is because it is necessary that the units are comparable to the input layer of the ANN. The information available to the input layer will be: coordinates, initial time, final time and service time.

The modification of the values corresponding to the weights of the links between neurons will be made in the same manner as with the previously explained network (15), defining a new neighbourhood function. Moreover, a new distance function will be defined. It will be called temporal distance and it replaces the previously used Euclidean distance in the neighbourhood function. The new function is:

$$\mathrm{d}t_{ij} \equiv \mathrm{d}t(x_i, x_t) = \mathrm{Max}\{f_{ij} + t_i, b_j\} \tag{19}$$

where t_i accumulated time to arrive to control point i plus the service time, b_j initial time, f_{ij} distance between neurons i and j and n_{ij} coordinate j of the neuron i. Therefore the neighbourhood function will be:

$$g(k,h,t) = \mathrm{Exp}\left[\left(-\frac{|k-h|}{N/2}\right)\frac{\mathrm{d}f_{kh}}{\mathrm{Max}_{i,j}\{d^*_{ij}\}} - \lambda\frac{|k-h|t}{\beta N}\right] \tag{20a}$$

$$\mathrm{d}f_{kh} = \begin{cases}\sqrt{(n_{k1}-n_{h1})^2 + (n_{k2}-n_{h2})^2} & \text{if } c_k - \mathrm{d}t_{0k} < d^*_{kh}\\ 0 & eoc\end{cases} \tag{20b}$$

where $d^*_{ij} \equiv f_{ij} + s_j$ with s_j being the service time for the control point ij, c_k being the closing time of the neuron k. f_{ij} is de distance between i and j.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

The use of the new distance $\mathrm{d}f_{kh}$ allows the neurons to be swapped with their neighbours if the temporal restrictions have not been overcome; nevertheless, this method does not guarantee that the system can achieve a valid solution.

3.3 *Revise and retain*

The revise phase is carried out by the security guard, who provides a report. If the security guard provides a positive assessment, then the complete plan is stored. This plan contains the sequence of states together with the values of believes for each of them. That is, the sequence of control points and their corresponding times. The quality of the route comes determined by the number of replannings.

$$w_{\mathrm{f}}(p) = w_i(p) \cdot \ln(t) \cdot \frac{(N-n)}{N} \tag{21}$$

where t is the number of times that the plan has been used, n the number of replannings and N the number of control points.

The information stored in the memory of plans follows the expression (10) and (11). If the problem includes time restrictions, this information is added to the rest of the plan information. In this way, the plan will contain the follow information:

$$\left\langle T = \left\{ \frac{(x_i, a_i, s_i, e_i, t_i)}{x_i = (x_{i1}, x_{i2})}\ i = 1, \ldots, n \right\}, g \right\rangle \tag{22}$$

where x_i position (x, y) of every control point, a_i arrival time, s_i initial time, e_i final time and t_i service time.

4. Results

The system presented in 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. In a first step, the operation of the sub-symbolic model applied in the reuse phase of CBP–BDI was checked. The model was implemented through sub-symbolic ANNs. In Table 1 and Figure 2(a), (b) are represented the plans scheduled by the neural network RPTW (Routing Problems with Time Windows) once the division of the checkpoints had been previously done. In Table 1, it is shown the description of an example of surveillance route for a security guard. Table 1 shows (O) the order of the points in the route, (CP) identification of the control point to visit, the location of the control point (coordinates), the distance between the current and the following control point, the accumulated time from the initial control point, (IT) the initial time represents the lower time required to check the control point, that is the control point cannot be checked before this time, (FT) the final time represents the maximum time allowed for the arrival and finally (ST) the service time. The default upper limit is determined as the half of the working shift, 14,400 (4*3600). To simplify the results, it has been established that the speed at which the guards move is 1 m/s, so that there is an equivalence between distance and time. Thus, the results can be interpreted in an easiest way.

The final distance obtained is 2795.6938 with temporal restrictions and 1908.1222 without restrictions. It should be borne in mind that in the example shown in Table 1, there is a restriction for the start time of 2500, so that in neither case could end before that time.

Figure 2(a) shows the graphical representation of the route covered by the security guard previously studied in Table 1. Figure 2(a), (b) shows the route followed by the security guard when

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

Table 1. Route followed by a security guard under nine time restrictions.

O	CP	Position	Distance	Arrival	IT	FT	ST
0	0	(140.0, 250.0)	22.36068	0.0	0.0	100.0	5.0
1	**23**	**(130.0, 230.0)**	**151.20847**	**27.36068**	**1.0**	**500.0**	**5.0**
2	**18**	**(38.0, 110.0)**	**111.8034**	**183.56915**	**1.0**	**2000.0**	**5.0**
3	**7**	**(148.0, 90.0)**	**314.42966**	**300.37256**	**120.0**	**500.0**	**5.0**
4	**8**	**(427.0, 235.0)**	**106.06602**	**619.80225**	**1.0**	**700.0**	**5.0**
5	**10**	**(442.0, 130.0)**	**316.58963**	**730.8683**	**1.0**	**1500.0**	**5.0**
6	**14**	**(140.0, 225.0)**	**95.33625**	**1052.4579**	**100.0**	**1200.0**	**5.0**
7	11	(148.0, 320.0)	114.14027	1152.7942	1.0	14400.0	5.0
8	24	(126.0, 432.0)	82.0975	1271.9344	1.0	14400.0	5.0
9	2	(130.0, 350.0)	173.10402	1359.032	1.0	14400.0	5.0
10	20	(136.0, 177.0)	38.600517	1537.136	1.0	14400.0	5.0
11	**3**	**(147.0, 140.0)**	**125.57468**	**1580.7365**	**1600.0**	**2300.0**	**5.0**
12	17	(135.0, 15.0)	97.128784	1730.5747	1.0	14400.0	5.0
13	**21**	**(232.0, 20.0)**	**7.81025**	**1832.7035**	**1.0**	**2500.0**	**5.0**
14	16	(238.0, 25.0)	85.09406	1845.5138	1.0	14400.0	5.0
15	13	(242.0, 110.0)	120.9504	1935.6079	1.0	14400.0	5.0
16	15	(344.0, 45.0)	70.028564	2061.5583	1.0	14400.0	5.0
17	12	(342.0, 115.0)	95.12623	2136.587	1.0	14400.0	5.0
18	9	(435.0, 135.0)	18.681541	2236.7131	1.0	14400.0	5.0
19	6	(453.0, 130.0)	120.20815	2260.3948	1.0	14400.0	5.0
20	19	(538.0, 215.0)	206.32256	2385.603	1.0	14400.0	5.0
21	**5**	**(350.0, 130.0)**	**16.763054**	**2596.9255**	**2500.0**	**3000.0**	**5.0**
22	22	(334.0, 125.0)	109.38464	2618.6885	1.0	14400.0	5.0
23	1	(228.0, 152.0)	85.86617	2733.0732	1.0	14400.0	5.0
24	4	(250.0, 235.0)	111.01801	2823.9395	1.0	14400.0	5.0
			2795.6938	**2939.9575**			

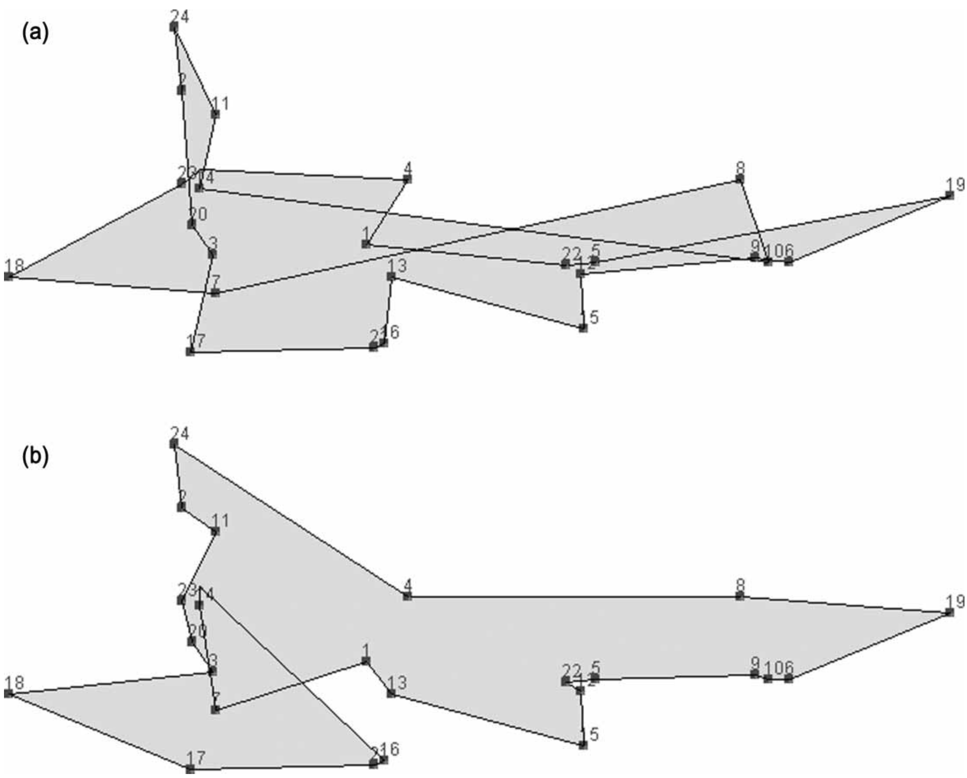


Figure 2. Distance calculated for one security guard with nine and four time restrictions.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

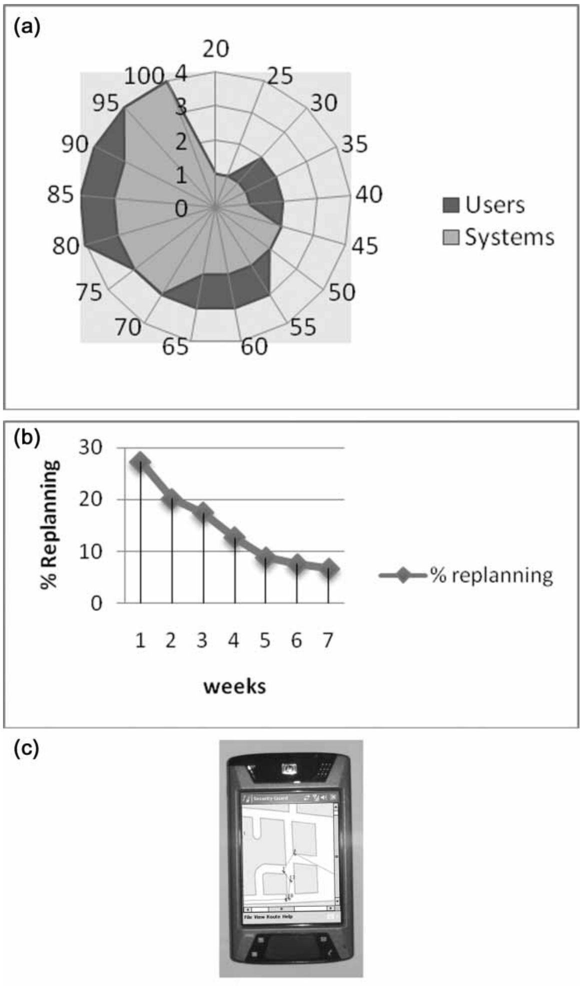


Figure 3. Average number of estimated security guards. Percentage of replanning. Route in the PDA.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

the system takes into account four temporal restrictions and the final distance is 2173.7756. In this case, it is possible to observe that the number of crossings notably increases compared to the example shown in Figure 2(b). This effect is due to the temporal restrictions imposed to the guard.

In Figure 3(b) it is possible to see how the percentage of variation for the routes related to the increase of the weeks. Figure 3(a) 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 five control points. The results are clear, for example, for 80 control points, the users estimated four security guards, but the system recommended only three. Figure 3(c) shows the PDA with the route that must follow the security guard.

5. Conclusions

The usage of a CBP–BDI agent allows the system to increase its performance since the ANN facilitates automatic route's calculation. Moreover, the CBP–BDI allows reducing the amount of preplanning in the system. 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.

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 mathematical AI model provides the supervisors with relevant information to monitor the workers activities, detecting incidences in the surveillance routes automatically and in real-time.

In this work, we have presented a novel CBP–BDI mathematical model, based on combining different AI techniques: MAS, CBR systems and neural networks. The CBP–BDI model has been successfully applied to a concrete scenario in the construction sector, for planning and monitoring surveillance routes. The promising results obtained as well as the characteristics of the CBP–BDI model, such as high learning and adaptation capabilities, let us conclude that the model can be very appropriated to be applied to similar environments, as shopping malls, health care scenarios, tourism, etc. The CBP–BDI model presented within this work is specially suitable to satisfy the needs of the emerging ambient intelligence (AmI). AmI is a new field with an important growth in the last years, and has the aim of developing intelligent environments where people are surrounded by technologies automatically adaptable to their personal needs. The basic concepts of the AmI are the ubiquitous computation, ubiquitous communication and intelligent interfaces. We think that the CBP–BDI model can provide efficient solutions for ubiquitous computation and can facilitate the construction of intelligent environments. That is our next challenge.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009

Acknowledgement

This work was supported in part by the projects MEC THOMAS TIN2006-14630-C03-03, IMSERSO 137/07 and JCYL SA071A08.

References

[1] J. Allen, *Towards a general theory of action and time*, Artif. Intell. 23 (1984), pp. 123–154.
[2] D. Apostolou, G. Mentzas, B. Klein, A. Abecker, and W. Maass, *Interorganizational knowledge exchanges*, IEEE Intell. Syst. 23 (2008), pp. 65–74.
[3] J. Bajo, V. Julián, J.M. Corchado, C. Carrascosa, Y. Pazade, V. Bottib, and J.F. Pazde, *An execution time planner for the artis agent architecture*, IEEE Intell. Syst. 21 (2008), pp. 769–784.
[4] P. Chang and C. Lai, *A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting*, Expert Syst. Appl. 29 (2005), pp. 183–192.
[5] J. Corchado, J. Bajo, and A. Abraham, *Gerami: improving the delivery of health care (special issue on ambient intelligence)*, IEEE Intell. Syst. 3 (2008), pp. 19–25.
[6] J. Corchado and R. Laza, *Constructing deliberative agents with case-based reasoning technology*, Int. J. Intell. Syst. 18 (2003), pp. 1227–1241.
[7] J. Créput and A. Koukama, *A memetic neural network for the euclidean traveling salesman problem*, Neurocomputing 72 (2009), pp. 1250–1264.
[8] D. Gamboa, C. Rego, and F. Glover, *Implementation analysis of efficient heuristic algorithms for the traveling salesman problem*, Comput. Oper. Res. 33 (2006), pp. 1154–1172.
[9] F. Gandon, A. Poggi, G. Rimassa, and P. Turci, *Multi-agent corporate memory management system*, IEEE Intell. Syst. 16 (2002), pp. 699–720.
[10] M. Guerini, O. Stock, and M. Zancanaro, *A taxonomy of strategies for multimodal persuasive message generation*, Appl. Artif. Intell. 21 (2007), pp. 99–136.
[11] M. Kianga and D. Fisherc, *Selecting the right MBA schools an application of self-organizing map networks*, Expert Syst. Appl. 35 (2008), pp. 946–955.
[12] O. Kwona, J.M. Shinb, and S.W. Kimc, *Context-aware multi-agent approach to pervasive negotiation support systems*, Expert Syst. Appl. 31 (2006), pp. 275–285.
[13] K. Leung, H. Jin, and Z. Xu, *An expanding self-organizing neural network for the traveling salesman problem*, Neurocomputing 62 (2004), pp. 267–292.
[14] H.M. Avilanoz and M. Cox, *Case-based plan adaptation: An analysis and review (special issue on ambient intelligence)*, IEEE Intell. Syst. 23 (2008), pp. 75–81.

[15] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita, *Interactive humanoid robots for a science museum*, IEEE Intell. Syst. 22 (2007), pp. 25–32.
[16] L. Spalazzi, *A survey on case-based planning*, Artif. Intell. Rev. 16 (2001), pp. 3–36.
[17] P. Vrba, F. Macůrek, and V. Mařík, *Using radio frequency identification in agent-based control systems for industrial applications*, Eng. Appl. Artif. Intell. 21 (2008), pp. 331–342.

Downloaded By: [B-on Consortium - 2007] At: 10:51 24 September 2009