Guiding Clients in a Shopping Mall using Case-based Planning and FYDPS Neural Networks

Juan F. de Paz, Juan M. Sánchez, Sara Rodríguez, Angélica González and Juan M. Corchado

Departamento Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{fcofds, darksnaider, srg, agl, corchado}@usal.es

Abstract. This paper presents a deliberative architecture based on the concept of CBP-BDI agent. A CBP-BDI agent is a BDI agent that incorporates a CBP reasoning engine. The work here presented focuses in the development of the CBP internal structure. The planning mechanism has been implemented by means of a novel FYDPS neural network. The system has been tested and this paper presents the results obtained.

1. Introduction

In this article we present a novel planning system based on the combination of neuronal networks with CBP (Case-based planning) systems [8]. Case-based planning allows us to retrieve past experiences when a new plan is created which lends the system a large capacity for learning and adaptation [8]. The neuronal networks proposed within this research framework are self-organised, based on Kohonen [11] networks, but which present certain improvements (FYDPS neural Neural Network) [13]. These improvements allow the network to reach a solution much more rapidly. Besides, once a solution has been reached, it is possible to make new modifications taking restrictions into account (in this study, specifically time restrictions).

Case-based planning is based on the way through which a new plan is generated through experiences acquired in the past (after the creation and execution of plans to resolve similar problems to the current one). Case-based planning is carried out through a CBP cycle [2], [3], [8]. The CBP cycle is formed by four sequential stages: retrieve, reuse, revise and retain. In the retrieve stage past experiences are recuperated with a description of the problem similar to that of the current problem. In the reuse stage, solutions used in the past are adapted to create a new solution. In the revise stage the results attained after executing a new plan are evaluated. Lastly, in the retain stage, lessons are learnt from the new experience. Each one of the stages of the CBP cycle may be implemented in various ways, using different algorithms. In this article we present a novel model that allows the integration of the planning based on cases from FYDPS networks. This model offers greater speed for obtaining the solutions that Kohonen networks, and incorporated restrictions in the network.

The planning system developed has been applied to an existent multiagent system, developed for guiding and advising users in Shopping Centres (also known as shopping malls) [2], [3]. A shopping centre is a dynamic environment, in which shops change, promotions appear and disappear continuously, etc. The proposed system helps users to identify a shopping or leisure plan as well as to identify other users within a given shopping mall. A CBP-BDI agent is a deliberative agent that works at a high level with the concepts of Believe, Desire, Intention (BDI) [2], [9]. The CBP-BDI agent has learning and adaptation capabilities, which facilitate its work in dynamic environments. A CBP-BDI agent is therefore a particular type of CBR-BDI agent [6], which uses case-based reasoning as a reasoning mechanism, which allows it to learn from initial knowledge, to interact autonomously with the environment as well as with users and other agents within the system, and to have a large capacity for adaptation to the needs of its surroundings. The multiagent system used a system of planning based on geodesic calculus [2], [3]. The results obtained with the planning system proposed in this study are compared with those obtained with the previous planning system and with a classic planning system.

Section two presents the shopping mall wireless multiagent system, then section three introduces the planning strategy and section four presents the novel FYDPS neural network model finally, the system is evaluated and the conclusions discussed.

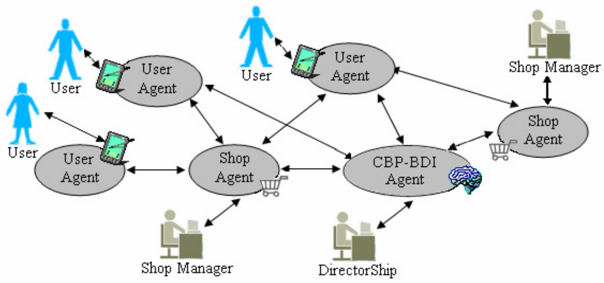

Fig. 1. Shopping Mall multiagent system: CBP-BDI agent, Shop agents and User agents.

2. Shopping Mall Multiagent System

This paper presents a distributed architecture whose main characteristics are the use of a CBP-BDI guiding agent, wireless agents and RFID technology [2], [3]. The CBP-BDI agent incorporates a reasoning Case Based Planning (CBP) engine which allows the agent to learn from initial knowledge, to interact autonomously with the environment and users, and allows it to adapt itself to environmental changes by discovering knowledge “know how”. The aim of this work is to obtain a model for recommending plans in dynamic environments. The proposal presented has been used to develop a guiding system for the users of a shopping mall that helps them to identify bargains, offers, leisure activities, etc. An open wireless system has been developed, which is capable of incorporating agents that can provide useful guidance and advice services to the users not only in a shopping centre, but also in any other similar environment such as the labour market, educational system, medical care, etc. Users (clients in the mall) are able to gain access to information on shops and sales and on leisure time activities (entertainment, events, attractions, etc) by using their mobile phone or PDA. Mechanisms for route planning when a user wants to spend time in the mall are also available. Moreover, it provides a tool for advertising personalized offers (a shop owner will be able to publicise his offers to the shopping mall users), and a communication system between management, the commercial sector or shoppers. Figure 1 shows the shopping mall multiagent system structure that has been developed and that is explained in detail in previous works [2], [3].

3. New Neural Network-based Plannig System

The purpose of case-based reasoning (CBR) is to solve new problems by adapting solutions that have been used to solve similar problems in the past [1]. The CBP is a variation of the CBR which is based on the generation of plans from cases. The deliberative agents, proposed in the framework of this investigation, use this concept to gain autonomy and improve their guiding capabilities. The relationship between CBP systems and BDI agents can be established by implementing cases as beliefs, intentions and desires which lead to the resolution of the problem. In a CBP-BDI agent, each state is considered as a belief; the objective to be reached may also be a belief. The intentions are plans of actions that the agent has to carry out in order to achieve its objectives [9], so an intention is an ordered set of actions; each change from state to state is made after carrying out an action (the agent remembers the action carried out in the past, when it was in a specified state, and the subsequent result). A desire is any of the final states reached in the past (if the agent has to deal with a situation, which is similar to one in the past, it will try to achieve a similar result to the one previously obtained). Below, the CBP guiding mechanism, used by the CBP-BDI guiding agent, is presented: Let $E = \{e_0,...,e_n\}$ the set of the possible interesting places to visit and shop at. These places are selected from the cases memory following the user preferences (shop type, product type or time available). Once the user indicates his preferences, the shops are selected, taking into account those that correspond with the user preferences. This selection is filtered using the user profile and is sent to the user. The user selects the shops, establish temporal restrictions and request a route generation. If the time available has been indicated, then the system chooses among the existing sub-routes in the cases memory those of route time lower than the available time, and shows the best alternatives. Finally the user selects one of the alternatives and the route is generated.

$$a_j : \underset{e_i}{E} \quad \underset{\rightarrow}{\rightarrow} \quad \underset{a_j(e_i)=e_j}{E} \tag{1}$$

An Agent plan is the name given to a sequence of actions (1) that, from a current state e_0, defines the path of states through which the agent passes in order to offer to the user the better path according to each user's characteristics. Below, in (2), the dynamic relationship between the behaviour of the agent and the changes in the environment is modelled. The behaviour of agent A can be represented by its action function $a_A(t)\ \forall t$, defined as a correspondence between one moment in time t and the action selected by the agent,

$$Agent\ A = \{a_A(t)\}_{t \in T \subseteq N} \tag{2}$$

From the definition of the action function $a_A(t)$ a new relationship that collects the idea of an agent's action plan (3) can be defined,

$$p_A : \underset{(t, a_A(t))}{TxA} \underset{\rightarrow}{\rightarrow} \underset{p_A(t)}{A} \tag{3}$$

in the following way,

$$p_A(t_n) = \sum_{i=1}^{n} a_{iA}(t_i - t_{i-1}) \tag{4}$$

Given the dynamic character desired for the agent, the continuous extension of the previous expression (4) is proposed as a definition of the agent plan (5).

$$p_A(t_n) = \int_{t_0}^{t_n} a_A(t)dt \tag{5}$$

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 [10]. The neurons are organized in a two-layer unidirectional architecture. The learning method is presented as follows: (The equations are presented in the order in that they should be executed).

- To present the input vector $X^p = (X_1^p,...,X_i^p,...X_N^p)^T$ in the input layer.
- The weightings initially take random weightings in (0,1).
- Calculate the intensity of the neurons of the output layer. Euclidean distance:

$$y_k = \sqrt{\sum_{i=1}^{N}(x_i - w_{ki})^2} \tag{6}$$

- To determine the winning neuron: that of smaller Euclidean distance.
- To upgrade the weights of the neurons that connect the input layer with the output neuron:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t)g(k,h,t)(x_i(t) - w_{ki}(t)) \tag{7}$$

$$g(k,h,t) = e^{\frac{-|k-h|^2}{2R(t)^2}} \tag{8}$$

(Gaussian function), therefore the formula is the following:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t) e^{\frac{-|k-h|^2}{2R(t)^2}} (x_i(t) - w_{ki}(t)) \quad (9)$$

Where w_{ki} is the weight of the connection between the input neuron i and the output neuron k; t is the iterations; η is the learning rate; h is the position of the winning neuron; k is the neuron of the output layer; and i is the neuron of the input layer. In k-h a Euclidean distance is calculated between the neurons.

4. Defining Self-Organising Neural Network in a novel Way: FYDPS Neural Network

The basic Kohonen network [11] cannot be used to resolve our problem since it attempts to minimise distances without taking into account any other type of restriction, such as time limits. In the present study a planner is described that is based on Kohonen networks but with a number of improvements (FYDPS Neural Network) [13] that allow us to reach a solution far more rapidly. Furthermore, once a solution has been reached, it is re-modified in order to take restrictions into account. Neural networks are heuristics trying to achieve a solution among all possible solutions, closest to the optimum solution.

4.1 Objective 1: Reach a Solution more Rapidly

As such, for this modification of the basic algorithm (FYDPS), we are aiming to make the solution search more agile and in order to achieve this, the basic vicinity function used in the Kohonen network is modified and the number of neurones in the output layer corresponds to the places that the subject wishes to visit. The topology of the neural network being considered is described below. The input layer is formed by two neurons, each one of those receives one of the coordinates of the shop presented as input. A vector of neurons is used of size the same as the number of places to visit by clients of the problem in the output layer, as in [10], [12]. The number of neurons in the output layer isn't modified. Let $x_i \equiv (X_{i1}, X_{i2})$ $i=1,\ldots,N$ the coordinates from the shop i and $n_i \equiv (n_{i1}, n_{i2})$ $i=1,\ldots,N$ the coordinates the neurons i in R^2. N shops will be visited by the fixed client. Then there will be: Two neurons in the input layer and N neurons in the output layer. It will be considered a vicinity function decreasing with the number of iterations.

$$g(k,h,t) = Exp\left[\left(-\frac{|k-h|}{N/2} \right) \frac{\underset{\substack{i,j \in \{1,\cdots,N\} \\ i \neq j}}{Máx} \{f_{ij}\} - \sqrt{(n_{k1}-n_{h1})^2 + (n_{k2}-n_{h2})^2}}{\underset{\substack{i,j \\ i \neq j}}{Máx} \{f_{ij}\}} - \lambda \frac{|k-h|t}{\beta N} \right] \quad (10)$$

λ and β are determined empirically, their respective values are: 5 and 50. t is the current iteration. $Exp[x] = e^x$. N is the number of the rooms that are visited by a fixed nurse and f_{ij} is the distance given by the Floyd Algorithm [14].

The radius of final vicinity should be similar to 0 so that the winner is only upgraded. Iteratively the group of shops will be presented, so that the weights of the neurons approach the coordinates of the shops. When concluding the process, there will be a neuron associated to each shop. To determine the route to follow, we will leave the shop associated to the neuron *i* to the associated to *i+1*, for *i=1, 2,…, N*, passing the whole vector of neurons. To close the road, the last tract will be given by the route that joins the shop in the commercial centre associated to the neuron *N* with the associated to the neuron *1*. The distance of the road will be given by the sum of the distances between the successive couples of shops of the road. The learning rate is a decreasing function:

$$\eta(t) = Exp\left[-\sqrt[4]{\frac{t}{\beta N}}\right] \tag{11}$$

The function of activation of the neurons is the identity. When the system stops, the route to continue will be given by the weights of the neurons that will be very closer to the coordinates of the shops. To know which the following shop in the journey, is we pass to the following element of the vector of neurons. The neurons are stored in a vector that contains the weights of each one in the current instant. So that the vector defines a ring, the neuron n_1 is the following to the n_N considered.

With a big radius of vicinity, in the first iterations of the algorithm the victory of a neuron affects great part of the map, so that a global self-organization takes place. If the radius decreases, the effect of a victory affects every time a part smaller in the map, so that the the criterion to stop the learning of the network is that the distance among shops cannot be optimized more. The initial number of total iterations is of $T_1 = \beta N$ (first phase). When $t = \beta N$, all the couples of possible neurons are exchanged (exchanging their weights) in the obtained ring of neurons, if the distance is optimized then the learning isn't finished.

In general, in the phase *Z*, the total number of interactions to carry is:

$$T_Z = T_{Z-1} - \frac{T_{Z-1}}{Z} \tag{12}$$

The aim of these phases is to eliminate the crossings. Concluded the iterations of each phase is proven if the distance is already optimized, in such a way that in the phase that stops the learning, the distance is minimum. For example, if we take the coordinates that represent European cities (TSPLIB is a library of sample instances for the TSP (and related problems) from various sources and of various types) from: http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/tsp/. In table 1, it is possible to observe that the primary objective - to achieve artificial neural networks that are faster than basic Kohonen networks, applied to the problems that the basic networks resolve - has been achieved. In the apparatus below the necessary modifications are introduced into the algorithm so that the network can take restrictions into account, and therefore be able to resolve other problems that cannot be resolved by basic Kohonen networks.

Table 1. Comparison between the Kohonen and FYDPS netowrks.

Problem	Number of cities	Optimum distance	Modified model		Basic model	
			Euclidean distance	Time (sg.)	Euclidean distance	Time (sg.)
ATT48	48	33523.7085	33831.73	6	36638.643	26
EIL51	51	426	433.753	5.25	453.698	29
PR2392	2392	378032	441413.72	14796	465678.89	39875

4.2 Objective 2: Taking Time Restrictions into Account

The number of neurons in the hidden layer is 5, two for the coordinates and three for the opening time, closing time and service time. Instead of using the Euclidean distance, a different distance is used that we call "temporal distance". Without loosing any generality it can be supposed that a distance unit is equivalent to a time unit.

$$dt_{ij} \equiv dt\left(x_i, x_j\right) \square\; Máx\left\{f_{ij} + t_i, b_j\right\} \tag{13}$$

Where t_i is the time hended to get to shop neuron *"i"* from the previous shop neuron plus the time taken on tasks to be carried out within shop neuron *"i"* (in other words, the service time in *"i"*) and b_j is the time limit for carrying out the tasks in shop neuron *"j"*. In this way, the vicinity function of the network modified from the FYDPS network is:

$$g(k,h,t) = Exp\left[\left(-\frac{|k-h|}{N/2}\right)\frac{\underset{\substack{i,j\in\{1,\cdots,N\}\\ i\neq j}}{Máx}\left\{d_{ij}^{*}\right\} - \sqrt{\left(n_{k1}-n_{h1}\right)^{2} + \left(n_{k2}-n_{h2}\right)^{2}}}{\underset{\substack{i,j\\ i\neq j}}{Máx}\left\{d_{ij}^{*}\right\}} - \lambda\frac{|k-h|t}{\beta N}\right] \tag{14}$$

$$d_{ij}^{*} \equiv d^{*}(x_i, x_j) \square\; \frac{f_{ij} + t_i}{dt_{ij}} \tag{15}$$

When the network obtains as a result the optimum plan p^{*}. Figure 2 shows the CBP cycle. If the plan p^* is not interrupted, the agent will reach a desired state $e_j \equiv e^*$. In the learning phase, a weighting $w_f(p)$ is stored. With the updating of weighting $w_f(p^*)$, the planning cycle of the CBP motor is completed. Let's suppose that the agent has initiated a plan p^* but at a moment $t>t_0$, the plan is interrupted due to a change in the environment. The solutions given by the neural netowrk meet the conditions of the Bellman Principle of Optimality [2], in other words, each on of the plan's parts is partially optimum between the selected points. This guarantees that if g_0 is optimum route for interrupted e_0 in t_1, because e_0 changes to e_1, and g_1 optimum route to e_1 that is begun in the state where g_0 has been interrupted, it follows that: $g = g_0 + g_1$ is optimum route to $e = e_0\,(t_1 - t_0) + e_1\,(t_2 - t_1)$.

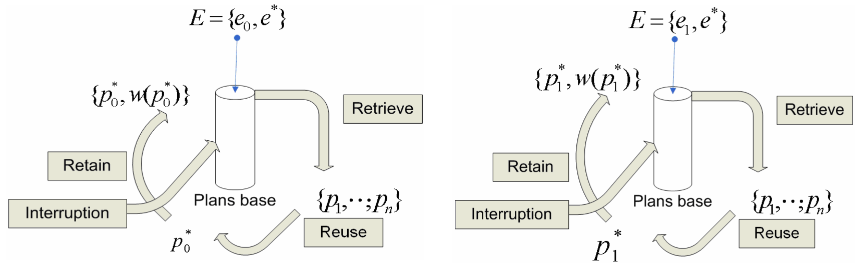

Fig.2. CBP cycle.

5. Results and Conclusions

This paper has presented an innovative guiding tool for clients in a shopping mall or similar environments. The guiding tool has into account the objectives of the client, the possible temporal and economic restrictions, and proposes the optimum route for the client profile and preferences. The proposed route considers opening and closing times for each shop in the shopping mall as well as their locations and the presence of doors, stairs, elevators, etc. The planning model proposed has been integrated within a previously Developed multiagent system [2], [3] and has been tested at the Tormes shoping mall in Salamanca. The users at the commercial centre access the planning service from their mobile devices (telephones or PDA's). An example of its use is illustrated in Figure 3. Figure 3 shows the plan that is suggested when a user arrives at the shopping mall and uses the guidance and suggestions service. Using his PDA, the user communicates his objectives, time available, timetable for carrying out his objectives within the set time frame and money. If the user has more time available than the time estimated to carry out his objectives, and according to his profile, other activities and additional visits are proposed, taking into account current promotions.

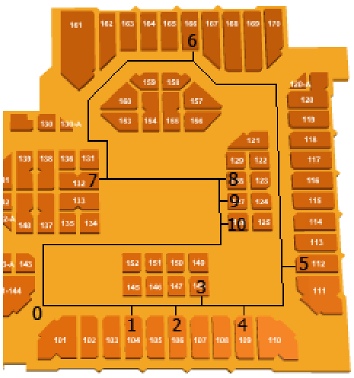

Fig.3. Route to be followed by the user.

In the specific example shown in Figure 3, the user defines his desire to buy music and clothes and to go to a gift shop. The user also indicates his desire to see an action film at the cinema, eat a pizza, and have a drink. The system proposes an additional activity to go to a mobile phone shop. The available time introduced was seven hours and no limit was placed on the money available. The plan proposed to the user can be seen in Figure 3. Table 2 shows the sequence o shops visited, the distance covered, and the time used at each stage of the plan. The user leaves intital point 0 in the Shopping Mall. First, he visit's the gift shop. Then he visit's a mobile phone shop. The third stage consists in a visit to a music shop and during stages 4 and 5 he goes shopping in clothes shops. In stage 6 the user goes to the cinema. At this stage the user has a time restriction to be at the cinema 10 minutes before the film begins (18.50 - 19.00) and to use the service during the subsequent hour and forty minutes. When the user leaves the cinema, he dines at restaurant 7. Lastly, the user visits three video arcades. Clients indicate the beginning and the end of each task since the tasks are presented one by one. Moreover the RFID readers situated in the entrances of each shop allow obtaining aditional informtion.

Table 2. Planning example.

Shop Coordinates	Distance	Arrival Time	End Time
(254, 444)	0	17:00	
(376, 456)	138	17:02	17:08
(436, 456)	84	17:10	17:20
(472, 532)	60	17:21	17:35
(530, 456)	82	17:37	18:00
(602, 390)	148	18:03	18:40
(460, 94)	388	18:47	20:30
(332, 270)	320	20:36	21:00
(604, 270)	172	22:13	22:37
(504, 300)	46	22:38	23:00
(504, 332)	46	23:01	23:27

In the example shown, there was no need to replan since there were no problems encountered with time restrictions or with the user changing his preferences. Before leaving the commercial centre the user gives his opinion on the route proposed, and if satisfactory, the data is stored in the belief base.

Table 3. Comparison of planners.

Planner	Execution Time (Secs)
Classic	134
Geodesic based	39
Proposed planner	27

In terms of the efficacy obtained with the new planning model, the results of the new model have been compared with those of a classic planner and with the prior system. A set of synthetic tests has been developed, proposing 50 cases for generating a plan in each planner. The average times taken by each planner to generate a plan is illustrated in Table 3. The Table shows how the planner proposed in this study significantly improves the time taken over classical planner, and also slightly improves on the time taken by a geodesic based planner.

Although the number of shops the planner proposes needs to be finite, the number can be very large. The service time can be estimated through other similar cases. The plans proposed for each problem don't need to be determinant. If more than one solution exists, the planner will offer different situations. The solutions given to problems are given rapidly. The plan proposed depends on each problem but if the problem has a solution, the planner will always find it. The plans are sequential, but nevertheless they are able to replan. The planning time is used as a restriction but if these time restrictions didn't exist, the planner would also function, with its objective being to minimise the route followed between the shops of the commercial centre, as chosen by the user. The planner has the ability to replan in the event that the optimum plan proposed is interrupted.

References

1. Aamodt A. and Plaza E.: Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7., pp 39-59 (1994)
2. Bajo J., de Paz Y., de Paz J.F., Martín Q. and Corchado J.M.: SMas: A Shopping Mall Multiagent Systems. Proccedings of IDEAL'06, LNAI, vol 4224 pp. 1166-1173, Springer Verlag. (2006)
3. Bajo J., Corchado J.M. and Castillo L.F.: Running Agents in Mobile Devices. Proccedings of IBERAMIA'06, LNAI, vol 4140 pp. 58-67, Springer Verlag. (2006)
4. R.E. Bellman, Dynamic Programming. (Princeton University Press, Princeton, New Jersey, 1957).
5. Bratman, M.E.: Intentions, Plans and Practical Reason (Harvard University Press, Cambridge, M.A., 1987).
6. Corchado J.M., and Laza R.: Constructing Deliberative Agents with Case-based Reasoning Technology, International Journal of Intelligent Systems, 18 (2003) 1227-1241.
7. Corchado J.M., Pavón J., Corchado E. and Castillo L.F.: Development of CBR-BDI Agents: A Tourist Guide Application, in: Proc. ECCBR'04, Lecture Notes in Artificial Intelligence, Vol. 3155 (Springer, Berlin, 2004) 547-559.
8. Cox M.T., Muñoz-Avila, H. and Bergmann R. Case-based planning. The Knowledge Engineering Review, Vol. 00:0, 1–4. c 2005, Cambridge University Press
9. Glez-Bedia M., Corchado J.M., Corchado E. and Fyfe C.: Analytical Model for Constructing Deliberative Agents, Engineering Intelligent Systems, Vol 3 (2002) 173-185.
10. Jin H.D., Leung K.S., Wong M.L., Xu Z.B.: An Efficient Self-Organizing Map Designed by Genetic Algorithms for the Traveling Salesman Problem. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, vol. 33, no 6. (2003). pp 877-888.
11. Kohonen T.: Self-organization and associative memory, Springer Verlag (1984).
12. Leung K.S., Jin H.D., Xu Z.B.: An expanding Self-organizing Neural Network for the Traveling Salesman Problem. Neurocomputing, vol. 62. (2004). pp 267-292.
13. Martín Q., De Paz J.F., De Paz Y., Pérez E., Solving TSP with a modified kohonen network, European Journal of Operational Research, (2007).
14. Martín Q., Santos M.T., De Paz Y., Operations research: Resolute problems and exercises, Pearson, (2005) 189-190.