

Dynamic Distribution of Tasks in Health-Care Scenarios

Carolina Zato, Ana de Luis, and Juan F. De Paz

Abstract. This paper presents a multiagent system that use an autonomous deliberative case-based reasoning agent to design an efficient working day. The system has been developed to plan and distribute tasks in a health care scenario, specifically in geriatric residences. This model generates a planning of tasks, minimizing the resources necessary for its accomplishment and obtaining the maximum benefits. For this purpose, the queuing theory and genetic algorithms have been include in a CBR architecture to obtain an efficient distribution. To evaluate the model, the obtained results have been compared with a previous method of planning based on neural networks.

Keywords: multiagent systems, queuing theory, genetic algorithm, task scheduling, health-care.

1 Introduction

During the last three decades the number of Europeans over 60 years old has risen by about 50%. Today they represent more than 25% of the population and it is estimated that in 20 years this percentage will rise to one third of the population, meaning 100 millions of citizens [1]. This situation is not exclusive to Europe, since studies in other parts of the world show similar tendencies. In the United States of America, people over 65 years old are the fastest growing segment of the population and it is expected that in 2020 they will represent about 1 of 6 citizens totalling 69 million by 2030. Furthermore, over 20% of people over 85 years old have a limited capacity for independent living, requiring continuous monitoring and daily care, for this reason it is important to create a task planner and control system for elderly people.

Carolina Zato · Ana de Luis · Juan F. De Paz

Department of Computer Science and Automation, University of Salamanca

Plaza de la Merced, s/n, 37008, Salamanca, Spain

e-mail: {carol_zato, adeluis, fcofds}@usal.es

Nowadays, a great amount of dynamic problems can be considered as planning problems. In this kind of problems the aim is to obtain a plan, as well as obtaining a scheduling of the actions along the time, assigning the necessary resources and taking into account restrictions like the benefits. Practical planning has been applied in many different fields and has been combined with different artificial intelligence techniques [2] [10]. One of the main challenges has been the development of intelligent systems focused on the improvement of health care sciences that can be applied in hospitals, residences, etc. [3].

The system presented in this paper includes deliberative agents using a case-based reasoning (CBR) architecture, as a way to implement adaptive systems to improve assistance and health care support for elderly and people with disabilities. Agents in this context must be able to respond to events, take the initiative according to their goals, communicate with other agents, interact with users, and make use of past experiences to find the best plans to achieve goals. The planning model is performed by applying CBR systems [5]. The CBR system integrates into various stages of reasoning, techniques to estimate resources based on queuing theory and planning using genetic algorithms.

The planner has been included into the ALZ-MAS multi-agent system [4][7]. ALZ-MAS generates the distribution of tasks among the personal taking into account their abilities and allowing the dynamic distribution of tasks according to their profiles. ALZ-MAS facilitates carrying out tasks in a ubiquitous way, such as plans generation for nurse's and doctors working days, or patient and task assignation. These tasks are executed in a distributed way inside the intelligent environment and are independent of the location of the processing devices due to the use of mobile devices and wireless networks. The previous planner of ALZ-MAS calculated the distribution of tasks using neural networks to establish the final sequence of actions. Now, this model has been improved using the queuing theory and genetic algorithms.

This article is divided as follows: section two presents the proposed model and section three describes the results obtained and the conclusions.

2 Proposed Model

The model proposed in this paper focuses on developing a planning mechanism to coordinate the agents. The roles of the agents are:

1. **Patient role.** It manages the patient's personal data and behaviour (monitoring, location, daily tasks, and anomalies). Every hour validates the patient location, monitors the patient state and sends a copy of its memory base to the manager agent in order to maintain backups.
2. **Personnel role.** It represents the different workers. This role can have different specializations according to the profile of the workers and his capabilities. In this case, the specialized profiles are doctors, nurses and nurse assistants.
3. **Planner role.** Design the overall plan to be implemented by the organization. Sets the number of personnel agents and makes the distribution of tasks depending on the specialized role they play. Replan depending on the size of the input queue or inability to accomplish with a plan.

4. **Distributor role.** Distributing tasks according to its completion by the agents and checks that each task is being processed within time limits to serve the plan.
5. **Manager role.** This agent controls the security (patients location, locks, alarms, etc.) and manages all the information of the system (record database, incidences, etc.).

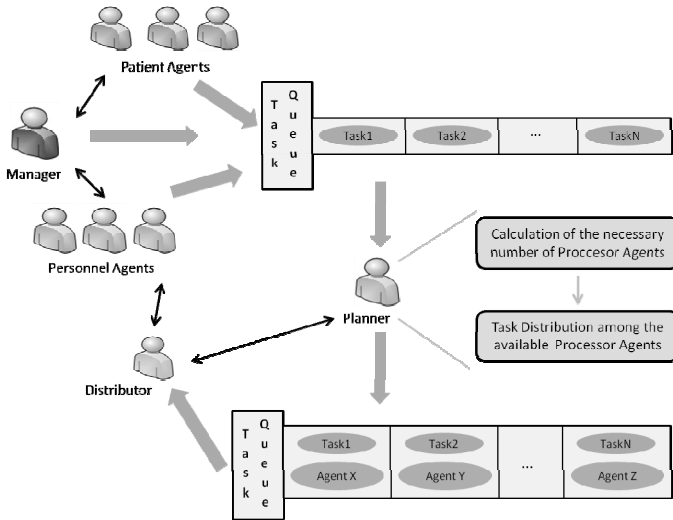


Fig. 1 Outline of proposed model

In the figure 1 the different agents of the system and some of the interactions among them are represented. In the upper corner of the figure appears the task list that store the activities to carry out in the multi-agent system and in the centre of the image, the planner agent that generates the planning.

To carry out the planning process, the agent follows the planning model CBR-BDI. The first point in the definition of a CBR-BDI model is the definition of case (1).

$$C = \{t_i / t_i = (ids, idt, l, f, p, d, a), i = 1..n\} \tag{1}$$

Where t_i represents the task i , ids is the identifier of the task type, idt the process, l localization (x, y, floor), f date of entry, p deadline date, d duration, a the ID of the agent that performed the task. The application of the different stages of reasoning is performed as follows:

Recovery. During this stage the most similar cases to the current case $cn+1$ are retrieved. The most similar cases are those containing more tasks of the same type as the tasks that are currently queued in the system. The number of cases retrieved

is predefined to subsequently implement the adaptation phase. The set of recovered cases are called $C_r \subseteq C$.

Adaptation. During this stage, the recovered information C_r is adapted from cases of memory to generate the entire plan corresponding to the case c_{n+1} . The information recovered is adapted by applying queuing theory and genetic algorithms. On one hand, the recovered information is used to determine the arrival rate and service for each of the tasks and thus using queuing theory we can determine the number of agents needed to run the indicated tasks. Recovered cases are the basis for constructing initial chromosomes in genetic algorithms.

Review. The review stage is performed automatically as the agents are finalizing tasks. The agent updates the duration of the tasks as they are completed and in turn, makes new plans if any notice is received from the processor agents on the inability to complete a plan under time constraints provided.

Learning. The learning phase is limited to store the case when the day is finished. The new case memory C' is defined as follows: $C' = C \cup c_{n+1}$. To limit the size of cases of memory, some are removed from memory if they exceeded a predefined age.

2.1 *Dynamic Planning Roles*

The number of agents that should be available in the system is estimated dynamically. It is intended that the number of agents suits demand to ensure that the system utilization factor ρ is less than 1. This estimate will be done through the use of queuing theory in a model $M/G/s$, where the arrival rate follows a Poisson distribution (the most commonly used in similar work), the exponential service and the existence of multiple servers (agents).

The problem of planning multiple tasks can be reduced to the case of planning for a single task of each type. Thus, for each task a planning is performed independently so as to calculate the average waiting time and average queue length independently. The average waiting time and the overall average length is reduced to calculating the average values calculated for each of the tasks. In the case of the $M/G/s$ model where $s = 1, 2, 3, \dots$ is the number of agents and given an arrival rate $\lambda_n = \lambda = \text{cte}$, the service rate when there are n processes is defined by the following equation (2) [8] [5].

$$\mu_n = \begin{cases} n\mu & n = 1, 2, \dots, s-1 \\ s\mu & n \geq s \end{cases} \quad (2)$$

Where μ represents the average service rate for s available agents. This value depends on both the agents and the machine found.

Assuming that the system is in a stable condition, i.e. it meets the utilization factor $\rho = (\lambda / \mu s) < 1$, the queuing theory [8] allows to calculate the probability that n tasks exists (P_n) in the system, the number of tasks (L) in the system. To determine the optimal number of agents, an estimation that minimizes the cost function is calculated and it depends on the number of agents used and on the waiting time in the queue. The function is defined in a particular way for each service depending on the actual costs of each agent in the system, though the following benefit function is provided (4).

$$f(L, P_0, \dots, P_{s-1}, \mu', \bar{p}, \bar{b}) = f_b(L, \mu', \bar{p}, \bar{b}) - k \cdot s \quad (3)$$

$$f_b(L, \mu', \bar{p}, \bar{b}) = \begin{cases} (\bar{p} / u') \bar{b} \cdot s \cdot (1 - \rho) & \text{si } L \cdot u' > \bar{p} \cdot s \cdot (1 - \rho) \\ L \bar{b} & \text{si } L \cdot u' \leq \bar{p} \cdot s \cdot (1 - \rho) \end{cases} \quad (4)$$

Where k is a constant associated with the cost of having an agent working, \bar{b} the average benefit of performing the task μ' is the average time to complete the task obtained from the service rate, \bar{p} the average time to execute a task. If the conditions of stability are overpass, f_b is counted only up to the utilization factor 1. The utilization factor ρ varies according to the new services added to the queue till it reaches the utilization factor of 1.

Following the cost function given in (4), a global cost function is introduced (5) that takes into account the implementation of the various services.

$$f(f_1, \dots, f_k) = \sum_{j=1}^k f_j \quad (5)$$

Where f_i is calculated from equation (4). Because sometimes you might not be given the stability conditions, it is necessary to calculate the terms in order of benefit depending on the type of the task so that when you reach the utilization factor of 100%, the process ends calculating the summation terms. Once the optimization function defined in (5) the maximum value is calculated iteratively starting with number of agents equal to 1, the fixed value is the first local maximum that corresponds to the global maximum.

2.2 Task Assignment

Once the number of starting agents is considered to minimize costs, an allocation of tasks between the available agents has to be done. The distribution of tasks among agents is performed so as to ensure as far as possible that it can perform assigned tasks in case of delays or the time to perform a task increases. It is performed so as to maximize the following function (6):

$$\min \sum_{i=1}^k \pm \log |1 + x_i - \bar{x}| \quad (6)$$

Where $\log |1 + x_i - \bar{x}|$ takes the sign of the expression $x_i - \bar{x}$, $x_i = t_i - a_{i-1} - c_i$ with t_i the maximum time for completion of the task i , a_{i-1} the accumulative time to perform the tasks $i-1$ above and finally c_i the time to run the task i plus the time of movement from the task $i-1$ calculated by Floyd algorithm [11], which is customized according to the agent selected and calculated from the average value of previously executed tasks. Minimizing the differences get all the tasks to have a uniform distribution of the remaining time so it gets easier to achieve them.

The chromosome encoding is performed so that each gene is composed of the elements listed by t_i identified in (1). The crossover operator is defined similarly to the multi-junction used in other problems such as TSP. The operator is defined as follows:

- Select a partial route
- Exchange of direct segments of tasks where ID matches
- The exchanges define a series of matches that relates each of the genes of a chromosome with which occupies the same position in the other parent

Mutation operators define various modes that will be executed randomly, and just those mutations that improve the aptitude of chromosomes will be selected. The defined mutation operators are: exchange order of tasks, exchange of assigning contiguous tasks and changing the allocation of a task.

Elitism operator is defined to keep the percentage of efficient solutions in every generation of population and population size as a constant this involves the replacement of parents by the children chromosomes in generations with the exception to remain with elite chromosomes. The roulette selection is the criteria chosen for this.

The initiation of chromosomes is based on the received tasks; each chromosome is initiated with the tasks of the new case. For each task of the chromosome a sequential search is made in the case and assigns the task to an existing agent so that this association is maintained for the remaining tasks.

3 Results and Conclusions

The previous planning model of ALZ-MAS calculated the distribution of tasks using neural networks to establish the final sequence of actions. This model was implemented by means of a RPTW (Routing Problems with Time Windows) neural network[4], which allows reaching a solution much more rapidly.

In order to compare both systems, we measure the number of replannings carried out by the planner agent. The first planner of ALZ-MAS was applied in a

working day and the process was repeated the next day using this new proposed planner. This process was repeated during 5 iterations. The image 1 shows the numbers of replannings carried out for each of the 5 nurses. The y axis represents the number of replannings and the x axis the nurses. The thin lines represent the RPTWplanner of ALZ-MAS and the fat lines the new planner. The colour represents the nurses. The number of replannings is reduced because the new planner creates the plans bearing in mind the intervals among the tasks in the final plan.

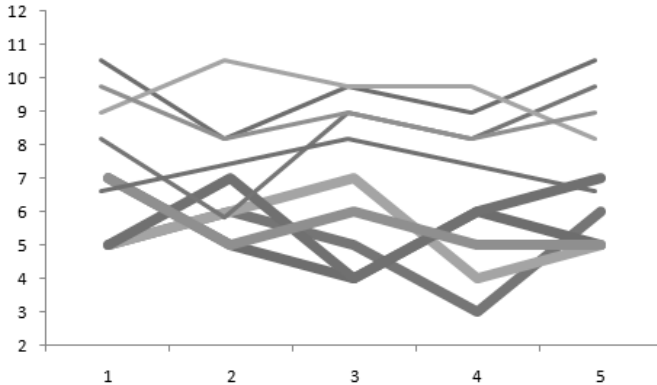


Fig. 1 Numbers of replannings carried out for each of the 5 nurses. Fat lines new planner and thin lines RPTW.

To complete the study the Mann-Whitney U-test [9] was applied. It was a non-parametric test and it is not necessary to make assumptions on the data distribution, as in the t-test. The test determines two values: H_0 and H_1 . H_0 shows if the data in both groups presents the same distribution, whereas H_1 determines if there is difference in number of replannings. The number of replannings for each worker was compared in both planners. The results establish that the number of replannings comes from different statistical distributions for each workerso, we can conclude that the differences are relevant.

The proposed planner allows reducing the number of replannings because it establishes homogeneous intervals among the tasks and calculates the arrival time of tasks using the previous information. Besides, the planner makes predictions of the necessary personal in order to accomplish all the activities.

Acknowledgments. This study has been supported by the Spanish Ministry of Science and Innovation project TRA2009_0096.

References

1. Camarinha-Matos, L., Afsarmanesh, H.: Design of a virtual community infrastructure for elderly care. In: PRO-VE 2002 – 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises, Sesimbra, Portugal (2006)

2. Zülch, G., Rottinger, S., Vollstedt, T.: A simulation approach for planning and reassigning of personnel in manufacturing. *International Journal of Production Economics* 90, 265–277 (2004)
3. Foster, D., McGregor, C., El-Masri, S.: A Survey of Agent-Based Intelligent Decision Support Systems to Support Clinical Management and Research. In: *Proceedings of the First International Workshop on Multi-Agent Systems for Medicine, Computational Biology and Bioinformatics*, pp. 16–35. Utrecht, Netherlands (2005)
4. Bajo, J., De Paz, J.F., De Paz, Y., Corchado, J.M.: Integrating case-based planning and RPTW neural networks to construct an intelligent environment for health care. *Journal Expert Systems Applications* 36(3), 5844–5858 (2009)
5. Martín, Q., Santos, M.T., de Paz, Y.: *Investigacion Operativa: Problemas y Ejercicios*. Pearson Educacion, London (2005)
6. Kolodner, J.: *Case-Based Reasoning*. Morgan Kaufmann, San Francisco (1993)
7. Corchado, J., Tapia, D., Bravo, J., García, Ó., Saavedra, A., Alonso, R., García, I.: ALZ-MAS 2.0; A Distributed Approach for Alzheimer Health Care. In: *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*. AISC, vol. 51, pp. 76–85. Springer, Heidelberg (2008)
8. Martín, Q.: *Investigación Operativa*. Prentice-Hall, Englewood Cliffs (2003)
9. Yue, S., Wang, C.: The influence of serial correlation on the Mann–Whitney test for detecting a shift in median. *Advances in Water Resources* 25(3), 325–333 (2002)
10. Yu, J., Buyya, R.: Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming* 14(3), 217–230 (2006)
11. Hougardy, S.: The Floyd–Warshall algorithm on graphs with negative cycles. *Information Processing Letters* 110(8-9), 279–281 (2010)