

A New Model for the Itinerary Definition of Real-Time Imprecise Mobile Agents

Luciana Rech, Carlos Montez and Rômulo de Oliveira
Department of Automation and Systems Engineering
Programa de Pós Graduação em Engenharia Elétrica - UFSC
P.O.Box 476, CEP 88040-900, Florianópolis, SC, Brazil.
lrech, montez, romulo@das.ufsc.br

Abstract

In information retrieving applications, mobile agents may have to finish their mission before a deadline. Also, there may be some flexibility for the itinerary definition due to optional precedence relations. In this paper, it is described an execution model where real-time imprecise mobile agents do not previously know their itineraries. The mobile agent route must be defined so as to meet a firm deadline while it maximizes the mission benefit.

1. Introduction

Mobile agents are agents that can migrate between physical nodes [1, 2]. Mobile agents reduce the network load, they execute asynchronously and autonomously; and can adapt dynamically [3]. In this work we consider the association of real-time requirements to mobile agents.

Mobile agents may be used to supervise the operation of a production line in factories. The agent has the mission of gathering data in order to form an image of a problem and to decide the next visit. In this case, the mobile agent reaches the fault diagnosis by traveling through several nodes and by collecting data for the decision making.

Information retrieval in distributed environment is another area where mobile agents can be employed [4]. Mobile agents can avoid the transfer of a large amount of data over the network and even operate in situations of discontinuous network connections.

In [5] it was defined a real-time imprecise mobile agent model along with some simple heuristics to guide the mobile agents in the definition of their itinerary. The goal of this paper is to describe a new execution model of mobile agents that are imprecise but have real-time constraints. We use some heuristics describe in [6] along with new algorithms for the dynamic determination of their itinerary.

This text is divided in 6 sections. In section 2 the adopted execution model is described. In section 3 the heuristics used for definition of the itinerary are introduced. Section 4 presents a comparison and the evaluation of those heuristics and section 5 has the conclusions.

2. Execution model

An itinerary I is defined as a route traveled in the distributed system. It is described as an orderly sequence of nodes N_i belonging to set N . An itinerary does not need to include all nodes of the system, and it can include in its defining sequence the same node several times.

The itinerary is gradually known by the mobile agent along the route, because it is defined by the information that the agent accesses at each node. In that way, the mobile agent is defined as myopic, that is to say, at each instant it only knows the possible immediate next nodes. Consequently, the problem described in this paper cannot be solved by using optimization approaches, because at the starting node the resource diagram is not completely known; and during the itinerary the visited nodes have limited processing capacity and they cannot make the calculations necessary for an optimization approach.

There is a benefit associated with each resource based on its functional importance within a particular mission. Each resource has a computing time previously defined. Each node has a set of resources. Instances of the same resource type may exist in different nodes. Figure 1 illustrates a situation where it is possible to observe that instances of the resource $r1$ can be found at node $N2$, $N6$ and $N8$.

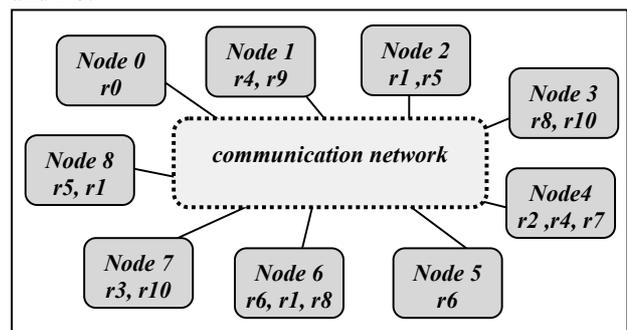


Figure 1. Resources in nodes of a distributed system

All nodes are directly connected to each other, so after each execution of a resource the agent has r possible ways (being r the number of available mission resources).

At each moment, the agent has an estimative for the benefit of each resource, also considering the resources already visited. These values are not fixed, after each visit all these estimates are updated. As soon as a resource is executed, its benefit for that mission becomes zero. This

value will be changed again only if there is a dependence of another resource and that is also executed. For example, the agent executes resource **R1** and its benefit is added to the mission total benefit, the resource benefit is now zero. The agent continues executing other resources. After the agent executes resource **R5** a benefit value of **10** is associated to resource **R1**, so the value of B_1 for this mission will become more than zero again, because of the execution of resource **R5**. The agent must return to resource **R1** in order to collect this additional benefit.

A resource may be visited many times. The benefit acquired at each visit may be different. It will depend on the information the mobile agent has at that moment. For example, benefit B_3 of resource **R3** is **10**, but if when executing **R3** the agent has already executed **R5** (**R5** has information that make resource **R3** to increase its contribution to the mission), then the benefit of **R3** is added by **15**, so, now the benefit of **R3** is **25**. But, if when executing **R3** the agent has not yet executed **R5**, the benefit acquired will be **10**.

The effective benefit of the mission is obtained by the sum of the benefits collected from each resource in the itinerary: $B = \sum B_i(y)$ for each $ri \in R$. (1)

where $B_i(y)$ is the benefit realized by the utilization of resource **ri** that the agent visited in order to fulfill its mission through itinerary **y**, **B** is the benefit generated by the utilization of all resources in the mission.

The importance of the present information in a resource (benefit) depends of which nodes it visited before.

We can classify this dependence in the agent's behavior as a relationship of optional precedence. That is, the execution of the resources is not forced to respect a pre-defined sequence, but depending on the traveled itinerary the agent will receive additional benefits. These values are known by the agent as it is executing each resource, that is, the agent is myopic.

Figure 2 presents examples of optional precedence relations between two resources. These examples are part of the application created to evaluate by simulation the heuristics described in section 3. Figure 2(a) shows that if when executing resource **R2**, resource **R1** is already executed, the benefit of **R2** will be added by 7. Otherwise, it will continue with its initial value. Similar effect is described in (b), (c), (d), (e) and (f).

3. Heuristics

In [5] it was defined another model for real-time imprecise mobile agents. It was also presented some simple heuristics to guide the mobile agent in the definition of its itinerary. In this article, differently of [5], we will consider that the itinerary possibilities are ignored before the mobile agent's departure.

This section presents some heuristics that make their decisions based only on the possible next nodes to be

visited; they do not consider the future unfolding of their short-term decisions. Whenever the agent needs to decide between two routes that, in the perspective of the heuristic used, they are equivalent, a random decision is made. The heuristics used are simple (they require minimum computational effort) and myopic (they work based on a partially known resource diagram).

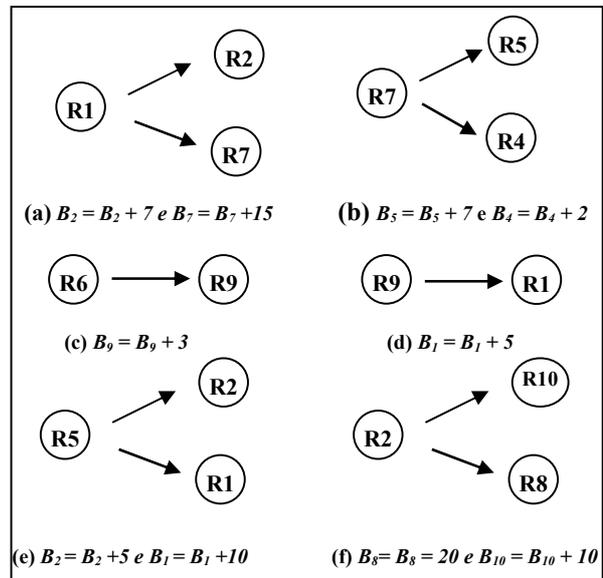


Figure 2. Examples of relationship of optional precedence

Another subject to be considered is the condition of the network and of the nodes (system load). The agent's decision making should take into consideration the benefit obtained by the execution of a certain resource together with the necessary computation time.

In this execution model, the behavior of the heuristics is subject to a great variation due to the great number of itinerary possibilities and also for the constant updates in the individual benefits of the resources. It happens because the relationships of optional precedence.

In [5] the flexibility in the choice of the itinerary was given by the existence of optional resources. In the model described in this paper the precedence relationship is optional and the flexibility in the definition of the itinerary facilitates that, after each executed resource, the agent may redefine its itinerary, that is, the decision making is based on the characteristics of the heuristics used by the agent in every moment of the mission. The termination condition of the mission is based on the time that the agent has for execution of the resources.

Heuristics Lazy, Greedy, Random, Higher Density and the versions with clock were described with details in [6].

Lazy Algorithm - It tries to execute the resources in the fastest possible way (resource that presents the smallest computation time) ignoring its benefits.

Greedy Algorithm - Whenever there is an alternative route, it will be chosen the resource that presents the largest benefit for the mission.

Random Algorithm - This algorithm chooses randomly the next node to be visited by the agent.

Higher Density Algorithm - This algorithm chooses the next resource that one that presents the best benefit/execution time rate. A resource will be executed only if its density is higher than the average density of the mission to that moment.

Versions with Clock - It has a probabilistic behavior. In the beginning of the execution they follow their original behavior; but as the deadline of the mission approaches, these algorithms adopt characteristics similar to the Lazy Algorithm. The algorithms that present this variation are the Clock Greedy and Clock Density Algorithm.

NZ_Random Algorithm - This algorithm has the same characteristics of the Random Algorithm. The difference is in the moment of the decision making, NZ_Random doesn't consider the possibilities where the benefit of the resource is zero.

Late Greedy Algorithm - This heuristic is based on probabilities too. In the beginning of its execution it is decided the behavior that the algorithm will have. Initially the probability of being chosen the Lazy Alg. is larger. But, as the deadline D of the mission approaches, the probability of the Greedy Algorithm to be used increases.

4. Comparison of the heuristics

For the simulations it was used: the described execution model, the resources presented in Figure 1 and the precedence relationships presented in Figure 2. Based on these data a graph was created where each vertex represents a state in the agent's behavior (for example: arrival in the node, execution of the resource) and each edge represents the benefit and the time consumed for the execution of a certain action. The vertices 0 and 1 indicate the points at the beginning and ending, respectively.

For the calculation of the total mission computation time it was considered: (i) computation time of each resource ($t_1=7, t_2=8, t_3=5, t_4=10, t_5=7, t_6=9, t_7=2, t_8=4, t_9=3, t_{10}=4$); (ii) time in the processor queue, uniform distribution between 1 and 3 (light load), or between 1 and 10 (heavy load); and (iii) sending time L between nodes (exponential distribution with average 2).

The quality of the algorithm (global benefit G) is calculated by:
$$G = \sum B_{DA} / NT \quad (3)$$

where B_{DA} is the benefit acquired by the execution of resources with met deadline ($B_1=5, B_2=7, B_3=8, B_4=4, B_5=9, B_6=15, B_7=6, B_8=14, B_9=12, B_{10}=9$); and NT is the number of tries.

4.1. Simulation Results

Figure 3 presents the global benefit G obtained by each algorithm when the system load is respectively light (a) and heavy (b). A maximum potential benefit B_{max} exists

for the mission. How much a mobile agent is capable to accomplish of this maximum potential benefit it will depend on its deadline D . Given enough time, any heuristic will obtain B_{max} .

In Figure 3(b) the characteristic behavior and the results are similar to situations where the load in the nodes is light. It is important to remind that, in this case, the total time to reach the maximum benefit B_{max} of the mission is larger because of the time that the agent waits in local queues to execute the resources in the nodes.

Even with very tight deadline (e.g., $D=35$) the agent, independent on the chosen heuristic and the load conditions of the system and the network, adapts the restrictions of the mission and it acquires some benefit, even if small. That happens due to the termination condition of the heuristics. So, the agent is able of sensing when there is a time slack and it continues executing the resources of the mission to increase the benefit obtained so as to reach B_{max} .

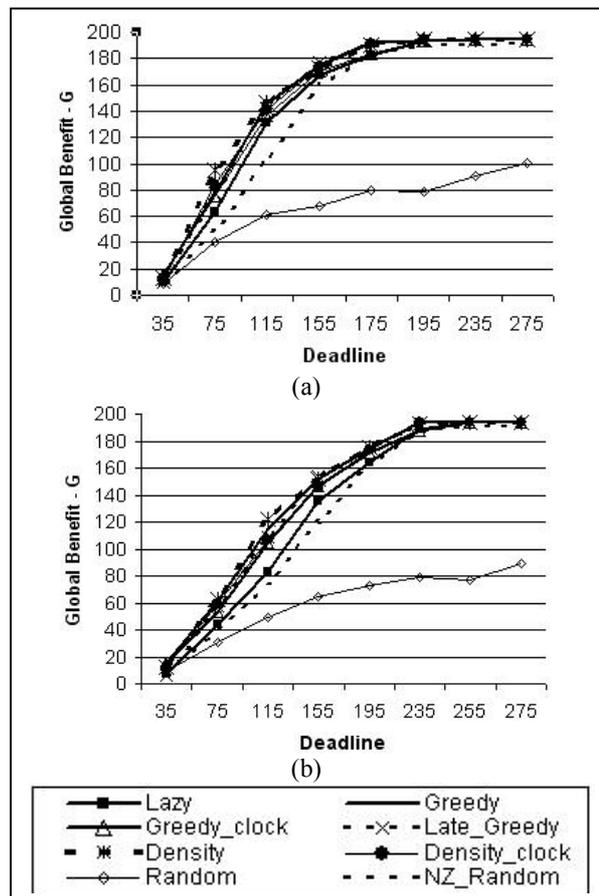


Figure 3. Performance of the heuristics

From a certain moment the global benefit obtained by the algorithms stays stable (Figure 3), independently of the deadline (in this model, $B_{max} = 194$ for all the heuristics).

The behavior of the heuristics is very similar (except for Random). What differentiates them the benefit

obtained at each deadline. For example, if the mission has a very tight deadline ($D=35$) the heuristics that had better performance was Greedy and Clock Greedy Algorithms for both load situations in the system and network.

Random Algorithm presented very low benefits for all the deadlines. The algorithms Greedy, Clock Greedy, Density, Clock Density and Late Greedy had close results. The clock addition to the algorithms Density and Greedy didn't result in significant improvement, these heuristics presented lower values than the original heuristics.

Lazy and NZ_Random Algorithms presented smaller benefits than the other heuristics (with tight and loose deadlines) until they reached B_{max} .

The superior envelop of the global benefit was formed by five different heuristics: the Clock Density, the Density, the Late Greedy, the Clock Greedy and the Greedy. This clearly illustrates the fact that the best heuristic will always depend of how loose or tight is the deadline.

Figure 4 presents a comparison of the Density Algorithm performance with light and heavy load, showing the relation between the response time and the mission benefit for each individual agent. A deadline of 150 was specified for the light load (a) and heavy load (b), 500 mobile agents were used for the mission.

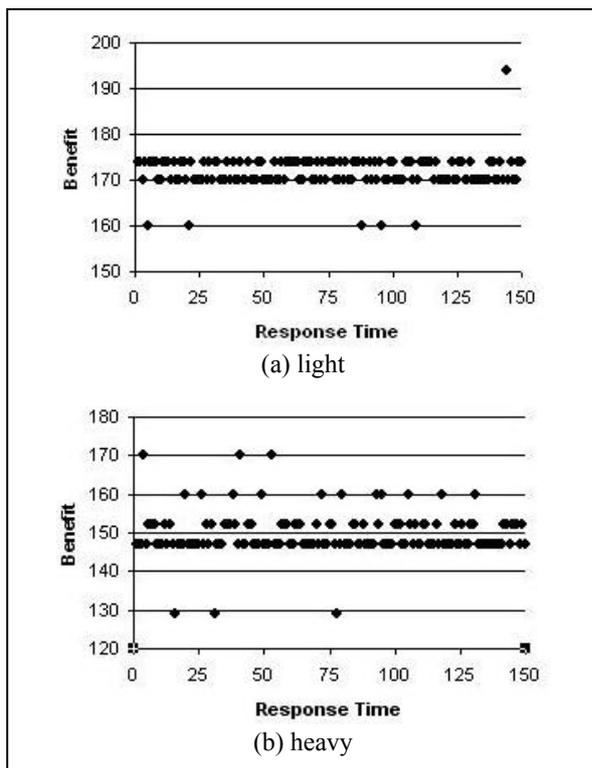


Figure 4. Comparative Graph of the Density Algorithm

In Figure 4 the points generated for the Density agent show the universe of possible results for the diagrams and the utilized simulation conditions. The comparison between graphs 4(a) and 4(b) shows what was expected in

relation to B_{max} . The necessary time to reach this value, besides depending of the used heuristic, will depend on the conditions of the system. For the same deadline ($D=150$), the Density Algorithm obtained larger medium benefit, approximately 170, when the load in the system was light. When the load in the system was heavy, the same heuristic reached for this deadline the medium benefit of 150.

The response time dispersion is consequence of the probabilistic behavior of the network delays and the processor queue delays.

In this paper considering the execution model proposed the heuristics presented a similar behavior (exception Random) differently of [6] where these heuristics were applied using another execution model. In this paper a possible choose might be the heuristic that presents the smallest overhead.

5. Conclusions

This paper presented a new execution model for real-time imprecise mobile agents. They have firm deadlines but some flexibility for the definition of their itinerary. There are timing requirements and the concept of optional resource and precedence added to the technology of mobile agents. It was described several heuristics that can be used for the determination of the mobile agent itinerary.

In the context of information retrieving, it is possible to associate the mission benefit with the confidence level the agent has in its conclusions. Zero may represent no confidence, while 100 represents complete confidence. A benefit that is bigger than 100 may represent over-confidence resulting from data redundancies. Each resource visited by the agent adds a little more confidence to its conclusions.

6. References

- [1] G. P. Picco, "Understanding, Evaluating, Formalizing, and Exploiting Code Mobility" PhD Thesis, Politeco di Torino, Italy, 1998.
- [2] J.W. Baek, G.T. Kim, and H.Y. Yeom, "Timed Mobile Agent Planning for Distributed Information Retrieval" Computation and Engineering School, Seoul National University, Proceedings of AGENTS'01, Montreal, Quebec, Canada, June, 2001.
- [3] R. H. Glitho, E. Olougouna, and S. Pierre, "Mobile Agents and Their Use for Information Retrieval: A Brief Overview and an Elaborate Case Study" IEEE Network, January/February, 2002.
- [4] G. Cabri, and L. F. Zambonelli, "Agents for information retrieval: Issues of mobility and coordination", J Syst Arch 46, 2000, pp. 1419-1433.
- [5] L. Rech, R. de Oliveira, and C. Montez, "Dynamic Determination of the Itinerary of Mobile Agents with Timing Constraint", IAT 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Compiègne, France, September, 2005, pp. 45-50.
- [6] L. Rech, R. de Oliveira, and C. Montez, "A Clone-Pair for the Dynamic Determination of the Itinerary of Imprecise Mobile Agents with Firm Deadlines", ETFA 2006 11th IEEE International Conference on Emerging Technologies and Factory Automation, Sep., 2006. (accepted)