UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DENISE DE OLIVEIRA
PAULO ROBERTO FERREIRA JR.

Aplicação de abordagens baseadas na organização de colônias de insetos sociais para resolução de problemas de organização em Sistemas Multiagentes

Relatório de Pesquisa
RP-344

Profª. Dra. Ana Lucia Cetertich Bazzan
Orientador

Porto Alegre, Abril de 2004

SUMÁRIO

1 INTRODUÇÃO

Um estudo anterior realizado sobre a organização das colônias de insetos sociais e sua aplicação na organização de Sistemas Multiagentes, na disciplina de projeto de pesquisa CMP301[1], mostrou o quão promissora é a aplicação dessa abordagem. Na referida disciplina dois problemas foram tratados por abordagens construídas através da inspiração biológica e baseadas em modelos teóricos já aplicados por outros pesquisadores em problemas correlatos.

O primeiro problema considerado foi o da sincronização de semáforos em uma via, cujo objetivo é garantir um fluxo de veículos distribuído proporcionalmente nas vias que são controladas por estes semáforos. O segundo problema aborda um aspecto da utilização do TÆMS[2]. O TÆMS oferece recursos para representar os meios necessários para os agentes atingirem seus objetivos. Realizar a análise deste processo e determinar um curso apropriado para as ações dos agentes dadas restrições temporais é o papel do escalonador. Em geral o número de escalonamentos possíveis para qualquer estrutura significante é muito grande e é intratável computacionalmente determinar cada um deles.

Motivado pelo aparente sucesso na construção das abordagens concebidas na disciplina CMP301 e pelos interessantes resultados que a experimentação e a comparação com outras soluções podem trazer, é que se buscou trabalhar com mais detalhe cada uma destas abordagens.

Nas seções que seguem cada uma das abordagens é apresentada e os resultados obtidos são discutidos. A este documento estão anexados os dois artigos publicados que foram fruto do trabalho aqui relatado. O primeiro, intitulado “Reducing Traffic Jams with a Swarm-based Approach for Selection of Signal Plans” trata da sincronização de semáforos e o segundo, intitulado “A Swarm Based Approach for Task Allocation in Dynamic Agents Organizations”, trata do escalonamento de tarefas utilizando o TÆMS.

[1]Disciplina CMP301 entitulada “Estudo de insetos sociais e sua aplicação para organização de Sistemas Multiagentes” cujo relatório encontra-se publicado como RP-340

[2]O TÆMS é uma linguagem concebida para descrever a estrutura de tarefas dos agentes.

2 REDUZINDO ENGARRAFAMENTOS COM UMA ABORDAGEM INSPIRADA EM INSETOS SOCIAIS PARA A SELEÇÃO DE PLANOS SEMAFÓRICOS

Muitos trabalhos na área de tráfego buscam reduzir os congestionamentos. Vários destes buscam essa redução através da sincronização de semáforos, permitindo que os veículos trafeguem em uma direção de uma via principal sem ter de parar nas intersecções com vias de menor movimento.

As abordagens clássicas são, em sua maioria, baseadas na priorização de uma direção de forma centralizada e estática, sendo realizada normalmente por um humano especialista. Estas abordagens são pouco eficientes quando se considera a dinamicidade do fluxo de tráfego ou dependem excecivamente de comunicação, o que é bastante custoso. Abordagens mais flexíveis tem sido propostas baseadas, por exemplo, na teoria dos jogos, mas que necessitam de muitos recursos para serem implementadas.

O artigo em anexo referente a esta parte do trabalho propõe uma abordagem onde cada semáforo se comporta como um inseto social. Os planos semafóricos são vistos como tarefas que devem ser realizadas pelos insetos sem um controle central ou mecanismo de alocação de tarefas. Os estímulos a que os insetos são submetidos nesta abordagem dependem do número de carros aguardando para atravessar o cruzamento sinalizado com o semáforo, entre outra coisas.

Esta abordagem foi implementada em um simulador microscópico de tráfego que permite a modelagem de cada objeto individualmente (veículos, semáforos, etc). O cenário utilizado foi um trecho da rede viária de Porto Alegre, de onde foram utilizados dados reais de fluxo e planos semafóricos.

Foram simulados os fluxos de veículos na via principal e nas secundárias em diferentes situações: sem nenhuma sincronização entre os semáforos; com coordenação fixa; e com a abordagem inspirada nos insetos sociais. Em todos os casos foi medida a densidades de veículos na via principal.

Os resultados mostraram que a abordagem inspirada em insetos sociais é mais flexível. Os semáforos se adaptam ao fluxo corrente selecionando o plano semafórico mais apropriado, reduzindo a densidade na via principal.

O artigo denominado “Reducing Traffic Jams with a Swarm-based Approach for Selection of Signal Plans” apresenta com detalhes da abordagem desenvolvida e seus resultados. Este artigo foi submetido para o *Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence* - ANTS 2004 - que acontecerá de 6 a 8 de setembro deste ano em Bruxelas na Bélgica e ainda encontra-se em revisão.

3 UMA ABORDAGEM PARA A ALOCAÇÃO DE TAREFAS EM ORGANIZAÇÕES DINÂMICAS DE AGENTES INSPIRADA EM INSETOS SOCIAIS

Um dos problemas mais estudados na área de Sistemas Multiagentes é como determinar que tarefa será realizada, em que tempo, e por qual agente para atingir os objetivos do sistema. Tais tarefas tem importantes restrições como prazo de execução, dependência de outras tarefas, recursos associados, etc. Os agentes também têm características diferentes que devem ser consideradas como capacidade de realizar a tarefa, motivação, acesso a recursos, etc. Este problema tem sido tratado por diferentes abordagens.

Em ambientes abertos e dinâmicos os agentes tem de ser capazes de se adaptar e alterar seu objetivos, seus recursos disponíveis, sua relação com outros agentes, entre outras coisas. Esta questão é crucial nos sistemas multiagentes e está fortemente relacionada a modelos de aprendizado e adaptação com aqueles também observados sobre os insetos sociais.

O artigo em anexo referente a esta parte do trabalho propõe uma abordagem para a geração, adaptação e mudança na organização do sistema multiagentes dinamicamente utilizando uma abordagem inspirada nos insetos sociais. Esta abordagem é utilizada principalmente para a alocação de tarefas se planejamento ou especificação prévios e sem a necessidade de coordenação explícita.

Os resultados obtidos por esta abordagem mostram que os agentes são capazes de alterar sua organização dinamicamente de forma a atingir seus objetivos tão qualitativamente eficiente quanto se houvesse um planejamento anterior.

O artigo denominado “A Swarm Based Approach for Task Allocation in Dynamic Agents Organizations” apresenta com detalhes da abordagem desenvolvida e seus resultados. Este artigo foi submetido para o *Third International Joint Conference on Autonomous Agents and Multi Agent Systems* - AAMAS 2004 - que acontecerá de 19 a 23 de julho deste ano em Nova Yorque nos EUA. Este artigo será publicado e apresentado no formato de Poster.

APÊNDICE A *REDUCING TRAFFIC JAMS WITH A SWARM-BASED APPROACH FOR SELECTION OF SIGNAL PLANS*

Reducing Traffic Jams with a Swarm-based Approach for Selection of Signal Plans

Denise de Oliveira[1], Paulo Roberto Ferreira Jr.[1], Ana L. C. Bazzan[1], and Franziska Klügl[2]

[1] Instituto de Informática, UFRGS, C.P. 15064, 91501-970, P.Alegre, RS, Brazil
[2] Dept. of Art. Int., Univ. of Würzburg, Am Hubland, 97074 Würzburg, Germany

Abstract. Several approaches tackle the problem of reducing traffic jams. A class of these approaches deals with synchronisation of traffic lights in order to allow vehicles travelling in a given direction to pass an arterial without stopping at junctions. In short, classical approaches, which are mostly based on offline and centralized determination of the priorized direction, are quite unflexible since they cannot cope with dynamic changes in the environment (traffic flow) and/or depend too much on communication which can be costly or unavailable. More flexible approaches have been proposed but can be demanding to realise if based on techniques of game theory, for instance. The present paper proposes an approach where each traffic light behaves like a social insect. The signal plans are seen as tasks to be performed by the insect without any centralised control or task allocation mechanism. The stimulus depends on the number of cars waiting or passing the traffic lights, among other things. We implemented this approach in a microscopic traffic simulator which permits the modelling of each individual object – vehicles, traffic lights, etc. The scenario is taken from the city of Porto Alegre in Brazil, with real flow and signal plan data. We have simulated the flow of vehicles in an arterial and its vicinity under different situations: without any coordination between traffic lights, with fixed coordination, and with our approach. In all cases we have measured the density of vehicles in the arterial. The results show that our swarm-based approach is more flexible: traffic lights adapt to the current flow of vehicles by selecting the appropriate signal plan, thus reducing the density in the arterial.

1 Introduction

Approaches to reducing traffic jams has been proposed in several disciplines like transportation engineering, physics, and artificial intelligence, among others. A classical approach is to coordinate or synchronise traffic lights so that vehicles can traverse an arterial *in one direction*, with a specific speed, without stopping [1]. Thus, coordination here means that if appropriate signal plans are selected to run at the adjacent traffic lights, a "green wave" is built so that drivers do not have to stop at junctions.

This approach works fine in traffic networks with defined traffic flow patterns like for instance morning flow towards downtown and it similar afternoon rush hour. However, in cities where these patterns are not clear, that approach may not be effective. This is clearly the case in big cities where the business centers are no longer located exclusively downtown.

Beside, a priori determination of the appropriate signal plans for the different times of a day is a complex task that requires a lot of knowledge about dynamic traffic flow. Thus, flexible and robust approaches are not only attractive, but necessary. Multiagent systems, and especially swarm intelligence offer more flexible solutions. In [2, 3] a multi-agent based approach is described in which each traffic light is modelled as an agent. Each of them has pre-defined signal plans to coordinate with other agents in the neighbourhood. Different signal plans can be chosen in order to coordinate in a given direction or during a pre-defined period of the day. This approach makes use of techniques of evolutionary game theory: intersections in an arterial are modelled as individually-motivated agents or players taking part in a dynamic process in which, due to the reward, not only their own local goals but also a global one has to be taken into account. Moreover, each agent possesses only information about their local traffic states.

The benefits of this approach are threefold. First, it is not necessary to have a central agent to determine the direction of the coordination. Second, agents can build subgroups of synchronisation which meet their own local and current needs in terms of allowing vehicles to pass in one given direction. Third, it avoids communication between agents when they have to decide which direction to priorise, i.e. there is no explicit communication or negotiation.

However, payoff matrices (or at least the utilities and preferences of the agents) are required, i.e these figures have to be explicitly formalised by the designer of the system. This makes the approach time consuming when many different options of coordination are possible (for example all four directions: south, north, east, and west) and/or the traffic network is complex (for instance, not only a main arterial has to be considered but also many transversal and parallel streets).

Therefore, in order to meet this need, this paper presents an approach in which each crossing with a traffic light behaves like a social insect that grounds its decision-making on mass recruitment mechanisms found in social insects [4, 5]. Henceforth we use the terms crossing, junction, and traffic light indistinguily. This is so because in fact in each crossing or junction only one *signal plan* runs in a set of traffic lights (despite the fact that one sees two or three of these) so that the set of traffic lights must be seen as a single entity.

The signal plans are seen as tasks to be performed by the insect. Thus, in our approach the ability of changing tasks in order to suit the colony needs (both at local and global levels) are located in each crossing or junction. Stimuli are provided by the vehicless that, while waiting for their next green phase, continuously evaporate some “pheromone”. Thus the volume of traffic coming from one direction can be evaluated by the agent, and this may trigger some signal plan switching. No other information is available for the intersection agents.

Our approach was realized on a microscopic traffic simulator. This is necessary in any swarm-based approach since it is desirable that the objects can be modelled at individual level. Thus, the next section presents some basic concepts about the simulator and traffic simulation regarding synchronisation of traffic lights. Section 3 then discusses our swarm-based model of the traffic scenario, while Section 4 presents the scenario we simulated as well as the results of these simulations. Section 5 summarises the contributions and discusses future extensions.

2 Description of the Simulator and Synchronisation of Traffic Lights

We use the Nagel–Schreckenberg model [6] which is a microscopic model for traffic simulation originally based on cellular–automata (CA). In short, each road is divided in cells with a fixed length. This permits the representation of a road as an array where on the discrete positions vehicles may be positioned. Each vehicle travels with a speed which is represented by the number of cells it currently may advance at each time step. The vehicle behavior is expressed by some rules that represent a special form of car following behavior. This simple, yet valid microscopic traffic model can be implemented in such an efficient way that it is good enough for real time simulation and control of traffic.

As for the network representation, each road is described as a composition of nodes representing junctions (also called intersections, crossings) and edges. The expression edge is used to refer to directed edges representing one direction of motion on a road, i.e., one road usually consists of two (oppositely directed) edges.

In the urban traffic scenario, more elements were added such as traffic lights and more complex types of intersections. Thus, the simulation tool we developed consists of different elements like lanes, edges, vehicles, sources and sink (of vehicles), sensors and detectors, traffic lights. The topological configuration and parameter for the simulation dynamics are stored in a database. This database can also be used for save the status of all objects in the simulation.

Basically, the simulator checks the static and dynamic network data read from the database for consistency and initialises the scenario. During the simulation it receives and updates dynamic data like vehicle counts, etc. and handles the simulation output, as well updates the vehicle motions, traffic light, and data for statistics.

More details can be obtained in the paper which describes the structure of the simulator and the database [7]. Here, we focus on the traffic light since it is the main object for the coordination. Each (signalised) junction has an agent which is in charge of deciding which signal plan to run. In this paper we assume that all main junctions have traffic lights.

Signalised intersections are controlled by signal-timing plans which are implemented at traffic signals. A signal-timing plan (henceforth signal plan for short) is a unique set of timing parameters comprising basically the cycle length (the length of time for the complete sequence of the phase changes), and the split (the division of the cycle length C among the various movements or phases).

The criteria for obtaining the optimum signal timing is that it should lead to the minimum overall delay at the intersection. This is usually achieved by using simulation or optimisation programs. Several plans are normally required for an intersection (or set of intersections in the case of a synchronised system) to deal with changes in traffic flow.

The goal of coordinated or synchronised systems is *to synchronise the traffic signals along an arterial* in order to allow vehicles, travelling at a given speed, to cross the arterial without stopping at red lights. Besides the parameters mentioned above, the synchronised plans also need an *offset* i.e. the time between the beginning of the green phase at two consecutive traffic signals (only when they are synchronised).

Well designed signal plans can achieve acceptable results in *un-congested streets in one flow direction.* However synchronisation in two opposing directions of an arterial is difficult to achieve, if not impossible, in almost all practical situations. The difficulty is that the geometry of the arterial is fixed and with it the spacing between adjacent intersections. Only in very special cases the geometry allows progression in opposite directions. Synchronisation in four directions is, for practical purposes, impossible. Therefore an agent at a junction must *select* which plan to carry out, in analogy to a task selection.

As a measure of effectiveness of such systems, one generally seeks to optimise a weighted combination of stops and delays or a measure of the density (vehicles/unit of length) in the road or network. Here we use the latter.

The average density $\bar{\delta_k}$ of a lane k during a given simulation horizon T is thus computed by:

$$\bar{\delta}_k = \frac{\sum\limits_T \sum\limits_L N}{T \times L} \tag{1}$$

where:

L is the length of the lane in number of cells
N is the number of vehicles

If the time horizon T is 1 time step (as it is usually the case), then we do not need to consider the sum over T. Moreover, the density δ is always between 0 and 1 since a cell is occupied by at most one vehicle. Also, an average density value for a set of lanes or for the whole network can be computed by simply weighing each $\bar{\delta}_k$ by each length L_k.

We measure the average density in the network and also the density in some key roads. The former gives the engineer an idea of the whole performance but is of little use because it may compensate heavy loads in some roads with lower ones, giving the false figure that *on average* the flow of vehicles is satisfactory. More details are given in Section 4.1.

3 Model of task allocation in the traffic scenario

Theraulaz et al. [8] present a mathematical model that resembles a hyphothesis of how the division of labour may be organised in colonies of social insects. Interactions among members of the colony and the individual perception of local needs result in a dynamic distribution of tasks. Their model describes the colony task distribution using the stimulus produced by tasks that need to be performed and an individual response threshold related to each task. Each individual insect has a response threshold to each task to be performed. That means, at individual level, each task has an associated stimulus (e.g. the perception of waste as a stimulus for cleaning behavior). The levels of the stimulus increase if tasks are not performed, or not performed by enough individuals, etc. An individual that perceives (e.g. after walking around randomly) a task stimulus higher than its associated threshold, has a higher probability to do this task. This model also includes a simple way of reinforcement learning where individual thresholds decreases when performing some task and increases when not performing. This double reinforcement process leads to the emergence of specialised individuals.

These concepts are used in our approach in the following way: Each traffic light has a social insect behavior. This traffic light has different tendencies to execute one of its signal plans (each signal plan is considered an available task), according to the environment stimulus and particular thresholds. Besides these individuals, this approach also considers that each vehicle leaves a pheromone trace that can be perceived by the traffic light at the junction This metaphor is realistic since many junctions have sensors of type loop induction detectors which detect the counting of vehicles (and sometimes speed).

3.1 Computation of Stimulus

The liberated pheromone dissipates in a pre defined rate in time and its intensity indicates the vehicle flow in the street section. The pheromone trail can be considered as a stigmergic communication among the adjacent traffic lights. The increase of the accumulated pheromone in a certain direction can be seen by the insect as a change in a task selection executed by its neighbour.

Each particular task in the Theraulaz et al. model [8] has one associated stimulus. The intensity of this stimulus can be associated with a pheromone concentration, a number of encounters between individuals performing the task, or any other quantitative cue sensed by individuals. The traffic light stimulus is the average of the accumulated pheromone of all the lanes (incoming and outgoing).

The accumulated pheromone in a lane, $d_{i,t}$, is the pheromone trail accumulated in the lane i at time t. While the vehicles are waiting for the green light they remain releasing pheromone so the amount of pheromone increases.

$$d_{i,t} = \frac{\sum_{i=0}^{n} \beta^{-i}(d_{i,t})}{\sum_{i=0}^{t} \beta^{-i}} \tag{2}$$

where:

n time-window size

β pheromone dissipation rate of the lane

The stimulus s of the plan j is based in a weighted sum of accumulated pheromone in each phase of this plan. Each phase has a time share $((time_{end} - time_{begin})/time_{cycle})$, that indicates how much time the plan spends with a phase. A higher time interval indicates a phase priority in the plan.

$$s_j = \sum_{i=0}^{n} ((1 - \alpha) d_{in_{i,t}} + \alpha d_{out_{i,t}}) \Delta t_i \tag{3}$$

where:

n number of phases of the signal plan j
$d_{in_{i,t}}$ is the accumulated pheromone trail in the input lanes in phase i at time t
$d_{out_{i,t}}$ is the accumulated pheromone trail in the output lanes in phase i at time t
Δt_i is the time fraction of the phase i
α constant employed to set different priorities to the input and output lane densities

3.2 Actual Plan Allocation

Behavioural flexibility of changing plans is a consequence of environmentally induced changes in stimulus and threshold. Every signal plan possess associated stimuli according to the direction towards this signal plan is biased. Individuals may change task because high levels of stimulus related to a direction exceed their response threshold. Equation 4, defines the response function (the probability of chose the plan j as a function of stimulus intensity s_j) of the individual i.

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \tag{4}$$

where:

θ_{ij} is the response threshold for the individual i for executing the task j.
s_j is the stimulus associated with the task j.

3.3 Reinforcement

We use the specialisation model [9], where the threshold is updated in a self reinforced way. Each individual in the model has one response threshold to each task. Those thresholds are updated (increasing or decreasing) according to two different coefficients. The response threshold θ is expressed as units of intensity of stimulus. The response threshold θ_{ij} of an individual i when performing task j during time interval of duration Δt is:

$$\theta_{ij} = \theta_{ij} - \xi \Delta t_{ij} \tag{5}$$

where:

ξ learning coefficient
Δt time interval

The response threshold θ_{ij} of the agent i when not performing method j during time interval of duration Δt is:

$$\theta_{ij} = \theta_{ij} + \rho \Delta t_{ij} \tag{6}$$

where:

ρ forgetting coefficient

According to Gordon [10] the real ants are directly influenced by its success in performing a given task. Successful ants are motivated to stand performing a task and unsuccessful ants are motivated to change or stop performing the task. We have extended the Bonabeau et al. [9] model in order to include a success function as the coefficient that describes learning and forgetting at the same time (when the l is negative the agent is forgetting). Equation 7 defines this extention.

$$\theta_{ij} = \theta_{ij} - l \Delta t \tag{7}$$

where:

l is the learning/forgetting coefficient.
Δt is a normalised discrete time interval.

The success degree of the individual is given by the Equation 8 and Equation 9, where a greater standard deviation of the densities σ (Equation 10, where n is the number of street sections) leads to a smaller degree of success.

$$l = 1 - 2\sigma \tag{8}$$

$$l = 2e^{(-5\sigma)} - 1 \tag{9}$$

σ is the standard deviation of accumulated pheromone trail in the sections.

$$\sigma = \sqrt{\frac{1}{n-1} \sum\nolimits_{i=1}^{n} (d_i - \overline{d})^2} \tag{10}$$

where:

n is the number of street sections.
$\overline{d}$ is the mean accumulated pheromone trail in n sections.

The whole system tends to stay stable and suited to the traffic flow but can change in order to adapt to a new environment situation. Traffic lights in the same street with an intense traffic flow in a certain direction tend to adopt the synchronised plans and give priority for this direction.

4 Description of the Scenario and Results of the Simulations

4.1 Scenario

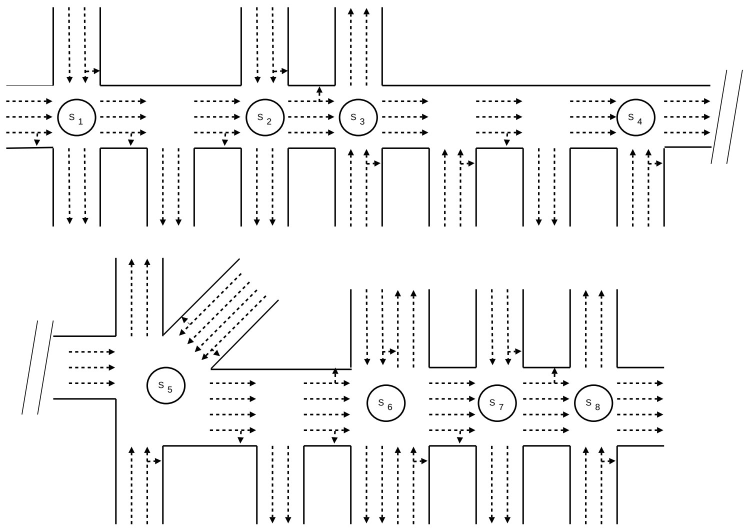

Fig. 1. Network and the traffic light location (numbered circles)

The scenario, showed in Figure 1 in a simplified schema, is part of a real network situated in the city of Porto Alegre (Brazil). This set of streets were chosen due to the high traffic flow and availability of data regarding flow of vehicles and the signal plans.

The main street or arterial has eight traffic lights, each with two possible plans. Signal plan 1 gives priority to the main direction (WE) and it is synchronised with the adjacent traffic lights in this direction. Plan 2 is not synchronised with plans in neighouring junctions, and allocates equal share of green time for each direction, as we can see in Figure 2.

Regarding plan 1, the difference between this kind of plan running at two adjacent junctions is the offset. For instance, junctions S_2 and S_3 in Figure 1 have the same basic plans but S_3 has a 16 second offset. This indicates that vehicles departing at S_2 and travelling with the synchronisation speed V will be able to pass at S_3 16 seconds later without stopping.

Vehicles are inserted in the network by sources located at the borders of it. For instance, vehicles are inserted in the main street from a source located in the left corner (Fig. 1). This insertion happens with different rates in each street. We setup these rates according to real traffic flow information. Similarly, at the network borders, vehicles are removed from the scenario. Besides, each junction

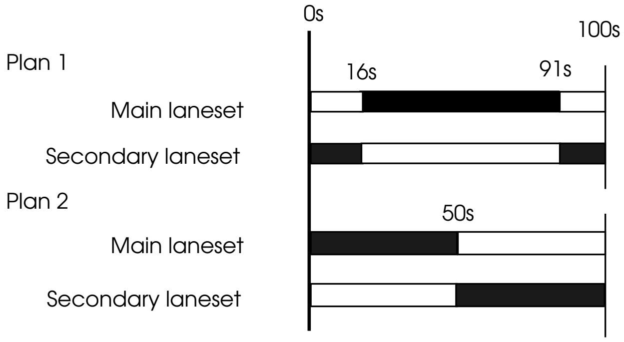

Fig. 2. Basic Signal Plans (the dark strip represents the time of green

has turning probabilies which can be computed from real data. Therefore, each vehicle either stays in the direction it is, or turns to another one.

4.2 Results from Simulations

The simulations presented in this section were generated using the simulator discussed in Section 2 with the scenario presented in Section 4.1. In the beginning of the simulation the network is empty and some time is needed for vehicles to reach all the portions of the network. Thus, in order to have a stable situation in the network, with a representative number of vehicle, we do not consider the initial time window. The average density of the arterial is computed in each simulation step.

The aim of this experiment is to compare the density achieved in the network using both our approach and standard ones. Two situations are evaluated. In the first, there is no synchronisation (i.e. plans like Plan 2 in Figure 2 are used in each junction). In the second case, we compare our approach to the situation in which a synchronisation is present but it is fix, i.e. the designer or engineer decides that all traffic lights are synchronised in a fix way.

In each case, we also evaluate and compare the different possibilities of our extension of the specialization model (the success function). Thus, our approach was simulated in four diferent ways: one do not uses reiforcement (the tresholds do not change during the simulation); one uses the original idea about the threshold, updating the threshold with a learning and forgetting coefficients; one uses the linear function to update the threshold; and the last uses the exponential function to update the threshold.

In this paper we adopt $\alpha = 0.2$, $\beta = 0.5$ and θ starting with 0.5. When changing the threshold using the original idea we adopt $\xi = 0.5$ and $\rho = 0.05$. Our extension uses the linear function presented in Equation 8 and the exponential function presented in Equation 9.

As we can see in Figure 3, the swarm approach achieves the best result when we use our extension on threshold variation using Equation 9 as the success

Fig. 3. Change in densities over time for the simulations.

function. The manual synchronisation shows a slightly better result because, in this scenario, we are not changing significantly the traffic flow in the adjacent streets, so the main street has a more intense traffic. Besides, in the simulation beginning (from step 5,000 to step 6,000), when the main street has a lower traffic flow than the adjacent ones, we can see that the manual synchronisation shows worst results than our approach. It is happen because the traffic flow in the adjacents streets are growing while the main street stays almost empty. Our approach was able to perceive this difference and to adapt the traffic lights to priorize the grater traffic flow. A total lack of synchronisation among the agents shows the highest densities levels, as expected. The fixed threshold curve indicates lower densities than the original model of learning and forgetting and also the success based variation that uses Equation 8 as learning and forgetting coefficient.

5 Conclusions and Outlook

This paper proposes an approach to reduce traffic jams based on a swarm-inspired method of selecting signal plans. We have discussed some approaches to reducing traffic jams, focusing on signal plan selection, either via classical approaches or via more flexible ones like the one proposed in [3]. We also discussed the need for even more flexible approaches in which the preferences of the traffic lights regarding the coordination or synchronisation do not have to be explicitly stated.

The swarm approach is well suited here because it profits from the metaphor of vehicles leaving a pheromone trail when stopped at a junction. This metaphor is used as a kind of stigmergy between adjacent junctions.

The approach was realized in a microscopic traffic simulator, to which models of social insects were added. These insects thus perceive the pheromone trails and act accordingly which in this case means a selection of an appropriate signal plan.

The average density in the arterial was measured in order to compare the following situations: i) the traffic lights are not coordinated; ii) they are coordinated in the classical way, i.e., using a central decision component (normally the traffic engineer) which determines the unique synchronisation for all junctions; iii) they are free to decide, at local level, whether or not to coordinate. This last approach is more flexible and depends only on flow detectors installed at each junction.

Quantitatively, when the agents are free to decide coordinating according to the swarm approach the system behaves almost as if a central decision support was given. Our experiments shows that the agents achieve synchronisation without any management, that indicates a successful swarm based application.

This works foresees some extensions as for instance increasing the set of signal plans an insect has. Additional signal plans can be designed either to coordinate in other directions or to coordinate in the main direction with other shares of green time and offsets. To implement this we depend on the traffic engineer who has to design such plans.

Other possible extensions are the simulation of the enlarged network (which is currently being done and again, depends on the engineers) so to consider parallel streets and so on. The case in which both arterials crossing at junction S_6 (Figure 1 are allowed to coordinate is very interesting because both are important arterials in the city.

References

1. TRANSYT-7F: TRANSYT-7F User's Manual. Transportation Research Center, University of Florida. (1988)
2. Bazzan, A.L.C.: Evolution of coordination as a metaphor for learning in multi–agent systems. In Weiss, G., ed.: DAI Meets Machine Learning. Number 1221 in LNAI. Springer–Verlag, Berlin Heidelberg New York (1997) 117–136
3. Bazzan, A.L.C.: A distributed approach for coordination of traffic signal agents. Autonomous Agents and Multiagent Systems (2004) under review, to appear.
4. Gordon, D.: The organization of work in social insect colonies. Nature **380** (1996) 121–124
5. Robison, G.E.: Regulation of division of labor in insect societies. Annual Review of Entomology **37** (1992) 637–665
6. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. I France **2** (1992) 2221
7. Andriotti, G.K., Bazzan, A.L.C.: An object-oriented microscopic traffic simulator. In: Proceedings of the XXVIII Latin-American Conference on Informatics (CLEI 2002), Montevideo, Uruguay (2002)
8. Theraulaz, G., Bonabeau, E., Deneubourg, J.: Response threshold reinforcement and division of labour in insect societies. In: Proceedings of the Royal Society of London B. Volume 265. (1998) 327–332
9. Bonabeau, E., Thraulaz, G., Dorigo, M.: Swarm Intelligence: From Natural to Artificial Systems. Oxford Univ Press (1999)
10. Gordon, D.: Ants at Work: How an Insect Society is Organized. W.W. Norton Company (2000)

APÊNDICE B *A SWARM BASED APPROACH FOR TASK ALLOCATION IN DYNAMIC AGENTS ORGANIZATIONS*

A Swarm Based Approach for Task Allocation in Dynamic Agents Organizations

Denise de Oliveira*, Paulo R. Ferreira Jr. and Ana L. C. Bazzan
Instituto de Informática, UFRGS
Caixa Postal 15064
91.501-970 Porto Alegre, RS, Brazil
{edenise, prferreiraj, bazzan}@inf.ufrgs.br

Abstract

One of the well studied issues in multi-agent systems is the standard action-selection and sequencing problem where a goal task can be performed in different ways, by different agents. Tasks have constraints such as deadlines and characteristics such as duration, resources, etc. Agents also have different characteristics such as capacity, access to resources, motivations, etc. This class of problems has been tackled under different approaches. At the high-level coordination, the specification of the organizational issues is crucial. However, in open, dynamic environments, agents must be able to adapt to the changing organizational goals, available resources, their relationships to another agents, and so on. This problem is a key one in multi-agent systems and relates to models of learning and adaptation, such as those observed among social insects. The present paper tackles the process of generating, adapting, and changing multi-agent organization dynamically at system runtime, using a swarm inspired approach. This approach is used here mainly for task allocation with low need of pre-planning and specification, and no need of explicit coordination. The results of our approach and another quantitative one are compared here and it is shown that in dynamic domains, the agents adapt to changes in the organization, just as social insects do.

1. Introduction

The organizational structure of multi-agent systems (MAS) is one of the most significant aspect for its success [10]. The agent's organization depends on the system needs to achieve the goals, to perceive the environment and to configure the agent's activities and its interactions. One problem is to define which organization form fits those needs best.

A simple way to solve this problem is to define the organization statically, that means to find the system needs and design an appropriate organization. Once this is made off-line the advantages of a well defined organization turn into disadvantages in an unstable environment. As MAS are used in dynamic problems, static organizational structures with rigid definitions become inefficient.

MAS need to manage the problems dynamics such as variation in the number of agents, changes in environment and in the system's goals. The question is how to derive such specific organizational structure given a particular situation. Most of the works in this area focus on adapting some specific aspects of the organization or on structure generation. Each of these approaches shows good results in their specific scenarios, but they are not general solutions to the problem. Recently an approach that intends to be more general, based on the TÆMS/GPGP/DTC framework plus system self-diagnosis was proposed [8]. This is a high-level coordination framework based on specification (of the organization goals, etc.), planning, and scheduling (which we call task allocation here). This framework also shows good results but some questions about their general efficiency are still open: regarding communication issues, especially in large organizations, how efficient are the resulting organizations?

The process of generating, adapting and changing organization dynamically at system runtime in MAS is usually called self-organization. Some authors prefer to call this process *organizational adaptation* [9] or *organization self-design*. Some critics to each of this definitions can be found in [12]. In these work we use *organizational adaptation* because our approach is based in organizational self-adaptation according to the dynamic changes in organizational needs.

* **Student Paper**

This motivation for studying organizational adaptation in MAS can also be found in some biological entities such as the social insects. Social insect colonies (also called swarms) show evidences of ecological success due to their *organization* which is observed in division of labor, specialization, collective regulation, etc. [2, 3]. The needs of the colony organization change over time. These changes are associated with the phase of colony development, time of year, food availability, predation pressure, and climatic conditions. Despite this drastic variations in colony's conditions, social insects do have ecological success [16].

A social insect colony operates without any explicit coordination. An individual worker cannot assess the needs of the colony, it just has a fairly simple local information, and no one is in charge of coordination [7]. From individual workers aggregation, the colony behavior emerges without any type of explicit coordination or planning. The key feature of this emergent behavior is the plasticity in division of labor inside the colony [11]. Colonies respond to changing conditions by adjusting the ratios of individual workers engaged in the various tasks.

In this paper we propose an approach to adapt organization in MAS inspired on social insects colonies organization. Due to its domain-independence, we use TÆMS [5] to represent agents activities. TÆMS is further discussed in Section 2.1 together with GPGP, and the Design-to-Criteria (DTC) scheduler. In our approach, a TÆMS task structure is used to model the necessary activities to achieve the system goal: the task structure is read and our approach determines the allocation of the tasks. This is performed using the theoretical model of task allocation in social insect colonies discussed in Section 2.2.

In Section 3 we present our approach in detail. The target scenarios and the simulations over it are presented in Section 4. Also in Section 4, we discuss the results and the performance of the current version of our approach. Section 5 concludes with further directions for this work.

2. Organization in MAS and Social Insects

Although many approaches to organization in MAS exist, this section focuses only on organizational representation, planning and scheduling (TÆMS/GPGP/DTC framework), and on the swarm based model necessary to explain our approach.

2.1. TÆMS, GPGP, and DTC

TÆMS [5], GPGP [6], and the Design-to-Criteria (DTC) [15] have been used as a domain-independent language for description of tasks associated with agents, planning and scheduling of agent's tasks. TÆMS allows the construction of a task model in which the relation of the actions available to choice is shown, providing ways to model scenarios where tasks have deadlines and some kind of result must be reached.

Agent's activities are represented as a graph in terms of their task groups aiming at achieving agent's goals. The leaves of the graph are called executable methods, which have probability distribution on their characteristics like quality, cost, and duration. The quality of a task group depends on what is executed and when. For example, quality can be accrued by a quality accumulation function (QAF) like *sum()*, which indicates that all tasks in the structure need to be accomplished.

Besides the local effects of the execution of methods on the quality and duration of their supertasks, there exist non-local effects (NLE) such as enables, facilitates, etc. Generally speaking, a task T may enable a method M in the sense that the quality of M cannot be accrued until T is completed, i.e. the earliest start time of M is the finish time of T. Therefore enables is a hard relationship, i.e. it has to be necessarily observed.

By using these tools, it is possible to construct the task structure of a problem-solving situation. The actual structure is called an objective model of the environment, and is inaccessible to agents. However, agents have each a subjective and a conditioned model or view of it, which they use to predict other agents actions'. The subjective view contains tasks and relationships the agent believes to be the complete model of its alternatives.

NLE's which involve more than one agent are called coordination relationships. Coordination mechanisms can recognize the features of the agent's subjective view, such as redundancies and soft and hard relationships. GPGP and DTC perform analysis of the processes modeled in TÆMS, and decides on the commitments and appropriate courses of action for the agent given the constraints (deadline, resources, etc.).

2.2. Swarm-Like Organization

Theraulaz et al. [13] present a model for self-organization inspired on the plasticity of division of labor in colonies of social insects [11]. Interactions among members of the colony and the individual perception of local needs result in a dynamic distribution of tasks.

This model describes the colony task distribution using the stimulus produced by tasks that need to be performed and an individual response threshold related to each task. Each individual insect has a response threshold to each task to be performed. That means, at individual level, each task has an associated stimulus (e.g. the amount of food need to be carried to the nest, if the task is to forage). The levels of the stimulus increase if tasks are not performed, or not performed by enough individuals, etc. An individual that perceives (e.g. after walking around randomly) a task stimulus higher than its associated threshold, has a higher probability to do this task. This model also includes a simple way of reinforcement learning where individual thresholds decreases when performing some task and increases when not performing. This double reinforcement process leads to the emergence of specialized individuals.

Let us assume that there are M tasks to be performed, each denoted by j, and that each of this tasks are associated with a stimulus s_j. Also assume that there are N individuals, each denoted by i, with response thresholds θ_{ij} associated with task j stimulus. An individual i engages in task j with probability:

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \tag{1}$$

where:

s_j stimulus associated with task j

θ_{ij} response threshold of individual i to task j

Each individual in the model has one response threshold to each task. Those thresholds are updated (increase or decrease) according to two different coefficients. The response threshold θ is expressed as units of intensity of stimulus.The response threshold θ_{ij} of an individual i when performing task j during time interval of duration Δt is:

$$\theta_{ij} = \theta_{ij} - \xi \Delta t_{ij} \tag{2}$$

where ξ is the learning coefficient and Δt is the time interval.

The response threshold θ_{ij} of the agent i when not performing method j during time interval of duration Δt is:

$$\theta_{ij} = \theta_{ij} + \rho \Delta t_{ij} \tag{3}$$

where ρ is the forgetting coefficient.

Each particular task in the model has one associated stimulus. The intensity of this stimulus can be associated with a pheromone concentration, a number of encounters between individuals performing the task, or any other quantitative cue sensed by individuals.

Variations in stimulus intensity can result from task performance or natural increase of task's demand. Bonabeau et al. [3] present two distinct ways to model this stimuli variation: performing a given task increases the demand for another tasks; and applying different success rates according to the task performance, changing to each specific task. The equations and results of this approaches are also presented in [3].

3. A Swarm Based Approach for Task Allocation

We use the swarm-based model to allocate insects-like agents to perform specific methods of a TÆMS task structure. This means that each agent deals with a dynamically changing TÆMS task structure and schedule its methods according to the TÆMS semantic.

Next, we discuss how the ideas of social insect organizations are used in order to allocate agents to tasks, and their application in the actual simulated scenarios.

3.1. Stimulus

As mentioned in section 2.1, a method is the element in a TÆMS task structure that represents what the agent can actually do (hereafter we call the insects tasks as methods). All methods in the TÆMS task structure have probability distributions of quality, cost, and duration. This values describes the possible results of the method execution. Therefore methods have quality (q_j), cost (c_j) and duration (d_j) and these are used to compute the stimulus s_j of a method j. The intensity of this stimulus is associated with the results of the methods execution. Each method j have one stimulus s_j:

$$s_j = \varphi * (\alpha * \hat{q_j} - \beta * \hat{c_j} - \gamma * \hat{d_j} + \beta + \gamma) + (1 - \varphi) * x_j \tag{4}$$

where:

$\hat{q_j}$ normalized expected quality of method j.

$\hat{c_j}$ normalized expected cost of method j.

$\hat{d_j}$ normalized expected duration of method j.

x_j stimulus associated with the QAF related to the method j.

$\alpha, \beta, \gamma, \varphi$ constants.

The constants are employed to set different priorities to the quality, cost and duration values (the sum of those constants should be 1). In this paper, these constants have the following values: $\alpha = \beta = \gamma = 1/3$ (in order to give quality, cost, or duration the same weight), and $\varphi = 0.5$.

The stimulus s_j for each method j is recalculated every time one method is performed by an agent (hereafter we call this an iteration). This stimulus updating is performed to model the emergent task succession discussed by Bonabeau et al. [3]. In social insects colonies, performing a given task increases the demand for another related task. For instance, creating a waste pile at the entrance of the nest generates a need for cleaning. In our approach, performing a method influences the stimulus associated with all methods of the same TÆMS task according the tasks' QAF.

Let us assume that there are M methods in the TÆMS task structure perceived by a given agent (only methods that are allowed to be performed in the current interaction). When any method k of M is performed, all methods j, related by a QAF with k, will have the x_j stimulus updated:

$$x_j = x_j + \kappa \tag{5}$$

where κ is the constant related to the QAF, as defined in Table 1.

This influence is recursive to each method of the parents tasks in the task structure tree. A constant κ associated with the QAF is used to model the influence of interrelated methods. We adopt small values for κ ($0 < \kappa \leq 1$) because the stimulus x_j is cumulative (increasing in each iteration) and takes values only between 0 and 1.

In Equation 4 we use the constant φ in order to set different priorities to the stimulus associated with the results of the methods execution (quality, cost and duration) and to the stimulus related to the emergent task succession.

QAF	κ
SeqMax, Max, SeqMin, Min	0
SeqSum, Sum	0.01
ExactlyOne	-0.01

Table 1. QAF related constants

Our approach was developed focusing on dynamic environments where the TÆMS task structure can be modified on the fly: methods can appear or disappear; the number of available agents can change; and the interrelationship among methods can also change. The latter is supported by the stimulus model presented above. However, this stimulus model does not take into account the changes in the number of agents and methods. Bonabeau et al. [3] show that emergent task succession can be achieved using fixed thresholds; however this has only limited applicability. In order to overcome these limitations of the stimulus model, our approach uses a modification of the specialization model (Section 2.2). This modification is discussed next.

3.2. Polyethism

Division of labor, in which a set of workers specialize in different set of tasks, is an important and well-studied aspect of colony behavior [11]. The age oriented specialization are called by the biologists temporal polyethism. In honey bees (*Apis mellifera*) this is the main form of division of labor. Young workers perform tasks within the hive, while older workers perform tasks outside the hive, such as foraging and colony defense.

Theraulaz et al. [13] only suggests an extension to model the temporal polyethism. His original model, without polyethism, uses two constants as learning and forgetting coefficients (Equations 2 and 3). To calculate the response thresholds with polyethism, we modify the Theraulaz et al. [13] specialization model. Our version uses two variables as coefficients of learning and forgetting based on temporal polyethism. The response threshold θ_{ij} of an individual i when performing method j is given by:

$$\theta_{ij} = \theta_{ij} - \frac{a_i}{A_i} * \frac{\mathcal{A} - m_j}{\mathcal{A}} \tag{6}$$

where:

a_i age of the agent i.

A_i maximum age of the agent i.

m_j age of the method j.

$\mathcal{A}$ age of the oldest available method.

The response threshold θ_{ij} of an individual i when *not* performing method j:

$$\theta_{ij} = \theta_{ij} + \frac{a_i}{A_i} * \frac{m_j}{\mathcal{A}} \tag{7}$$

In this specialization model, all agents start with the same θ_{ij} (usually an intermediate value). When a method is performed by an agent, the response threshold changes. For the agent to specialize in selecting a specific method, it is necessary that it selects this method some times. Thus, it is necessary to run the model for several rounds. In each round our approach produces a task allocation for the given task structure.

Given the probabilistic nature of the model, these allocations are not necessarily the same. If the task structure does not change, we consider an allocation of tasks to be the final one when it does not change after a specific number of consecutive rounds.

In equations 6 and 7, we consider the agents' age (proportional to the task deadline) and the methods' age (proportional to the oldest method's age). The age of methods and agents is computed at each iteration. A method's age increases until the method does not finish. An agent can survive a single round or stay alive during several rounds. The agents has higher thresholds regarding old methods and lower thresholds regarding new ones. Besides, a young agent has a lower threshold that an old agent regarding the same method. The idea behind this is to specialize old agents regarding a wider range of methods, and young agents regarding specific methods as it occurs in Nature.

4. Experiments

The simulations presented in this section were generated using a simulation tool developed in JAVA and using the TÆMS API.

In our approach we adopt the Bonabeu et al. tendency (Equation 1). This equation computes a probability distribution over the tendency of an agent to respond to each method's stimulus. As it is the case with social insects that inspire Bonabeu et al. models, with the use of this equation, each methods in the task structure can be performed since each has a *probability* of being carried out by an agent. Therefore, the best way to present and discuss the results is via the use of statistics: the results presented here are averages over 1,000 repetitions of each experiment.

4.1. Scenario I

We use the TÆMS task structure of Figure 1 as basis for our simulations. This task structure can represent, for instance, a typical problem of job scheduling among multi-purpose machines [4] in which the authors associate a small number of machines (2–4) with wasps. This task structure can also be related to the aircraft servicing scenario discussed in [14].

Task T_1 is the first stage of production/servicing. Jobs or aircrafts of type a, b, or c can arrive. If it is of type a, for example, then m_{1a} is allocated to an agent. This enables method m_{2a} and so on. Notice that in this scenario, there are hard relationships of type *enables* which make the task allocation little flexible.

Figure 1 shows only the duration distribution probability of each method. All methods have the same

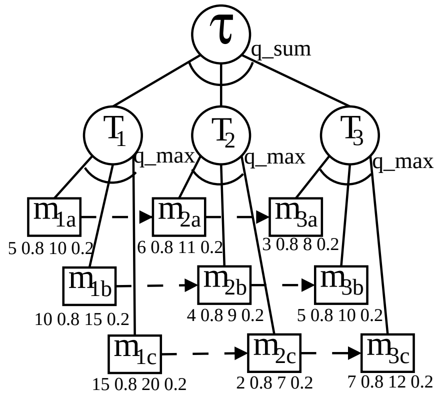

Figure 1. Objective TÆMS task structure.

cost and quality probability distribution: the cost for all methods is (0 0.8 1 0.2). The quality for all methods is (5 0.8 4 0.2). Unless said, we use deadline equal to 25.

We run DTC to compare our task allocation to the standard output produced by DTC, which is shown in Table 2. This table presents the start and end time for each method, qualities, and costs. Not shown there: the total quality is 14.35, the total cost 0.6, and the total duration 17.0.

method	start	finish	quality	cost	duration
m_{1a}	0.0	6.0	4.8	0.2	6.0
m_{2a}	6.0	13.0	4.8	0.2	7.0
m_{3a}	13.0	17.0	4.76	0.2	4.0

Table 2. DTC schedule

Although DTC deals with the probability distributions of quality, cost, and duration coming from the semantic of TÆMS , it does not handle these distributions probabilistically (e.g. using a roulette wheel). Instead it computes an average for each probability distribution. We use both, the probabilistic and the DTC approaches. The latter is useful when we compare results; the former is used otherwise.

Using our approach in a non-probabilistic variant produces the task allocation shown in Table 3. In fact, our approach produces 1000 outputs (as mentioned before), one for each repetition. The one shown in this table is the top one (more frequent) and is produced 32.7% of the time. The total quality equals to 14.4, the total cost 0.6, and the total duration 17.0. That means that our more frequent output is the one which resembles the output of DTC.

The same task structure was scheduled with probabilistic treatment of the quality, cost, and duration distributions (second variant). Table 4 shows the most frequent schedule, produced 22.3% of the 1000 repetitions. The total quality equals to 15, the total cost 0.0, and the total duration 24.0. All the three methods scheduled in the first variant are also present in the second one. However, due to the probabilistic variation of the duration, sometimes more methods can be scheduled within the deadline. In total, 41.5% of the repetitions scheduled at least the 3 methods DTC did, within the deadline.

method	start	finish	quality	cost	duration
m_{1a}	0.0	6.0	4.8	0.2	6.0
m_{2a}	6.0	13.0	4.8	0.2	7.0
m_{3a}	13.0	17.0	4.8	0.2	4.0

Table 3. Top schedule (first variant)

method	start	finish	quality	cost	duration
m_{1a}	0.0	5.0	5.0	0.0	5.0
m_{2a}	5.0	11.0	5.0	0.0	6.0
m_{3a}	11.0	14.0	5.0	0.0	3.0
m_{1b}	14.0	24.0	5.0	0.0	10.0

Table 4. Top schedule (second variant)

Of course, in both variants, the overall results of our approach are not as good as the one DTC computes.

However, our approach is intended not for static environments but for dynamically evolving ones. This means that our agents can adapt to changes in the environment with no need of commitments and communication. Such environments are now presented and discussed.

4.2. Scenario II

In order to measure the performance of our approach in dynamic environments, we schedule four different TÆMS task structures appearing randomly with the same probability. The first task structure (TS_1) is the one depicted in Figure 1. The other three are variants of it: one has no enable relationships between the tasks (TS_2); one has not the task T_3 (TS_3); and the last has the deadline changed to 30 (TS_4). Beside, we have changed the quality distribution: in TS_2 it is (15 0.8 10 0.2), in TS_3 (50 0.8 40 0.2), and in TS_4 (100 0.8 90 0.2).

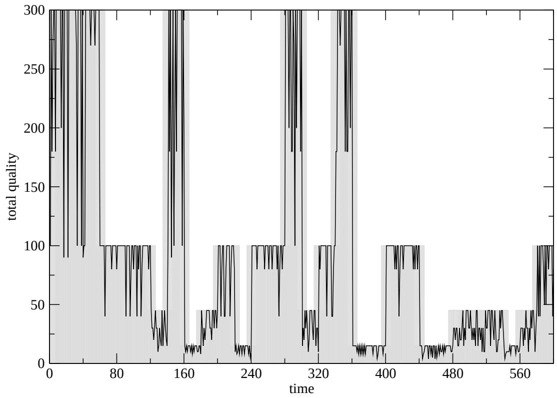

Figure 2. Change in quality over time (black line); Changes in the environment are depicted as shadow boxes.

Of course, we can only compare individual tasks structures to the DTC outputs. In our simulation, these 4 outputs appear 48% of the 1000 repetitions. However, the real power of this approach is when the environment changes dynamically.

Figure 2 shows these changes and how agents adapt to them. We change task structures randomly. In the beginning (first 50 steps), agents are still adapting. The gray shadow in the figure indicates which task structure is the actual one at a given time step. When the shadow goes up to 300, this means TS_4. Remember that the total quality is the sum over three tasks in TS_1, TS_2, and TS_4, and over two tasks in TS_3. When the shadow goes up to 100, 50, and 15, the actual task structure is TS_3, TS_2, or TS_1 respectively.

Ideally, each time a task structure changes, agents should adapt and so the quality would change instantly. Due to the time necessary for agents to adapt their stimuli, etc., there is a small delay in this process which can be seen in Figure 2 (black line does not match the shadow exactly).

Because of the probabilistic aspects of our approach (probabilistic distribution of methods' quality and probabilistic tendency to schedule a method) the black line is not constant, there is a variation in the schedules' quality for the same task structure. However, Figure 2 shows that when the task structure changes, so does the total quality associated with it.

This results show that modifying the task structure on the fly disturbs only slightly the performance of the agents regarding the quality of the schedules produced because each time the task structure changes, agents do adapt to this situation.

4.3. Scenario III

With the same aim of the simulation described above, we now change dynamically the number of agents available to perform the task structure. In this scenario, we also employed the basis task structure of Figure 1. This time three of these task structures are subtasks of a new task group (root task) whose QAF is *sum()*. The new task group has now 3 times as many methods as the basis taks structure, i.e. 27 methods. Therefore we vary the number of agents between 1 and 27. To cope with the probabilistic nature of the problem, we perform 100 repetitions each time we vary the number of agents, totalizing 2700 repetitions.

Figure 3 shows the influence of the number of agents over the number of methods scheduled and also over the quality. Nine methods of those 27 do not have any enable NLE. As we increase the number of agents, the number of this enable-free methods performed increases. When the number of agents is equal to the number of the enable-free methods, that means 9 agents, the number of performed methods stabilizes because even if we put more agents, they cannot perform methods which are not enabled. The same reasoning applies to the quality: the best one is achieved after this stabilization, that means when the number of agents is equal to the number of enable-free methods. The highest possible quality, around 40, is reached when we have 9 agents.

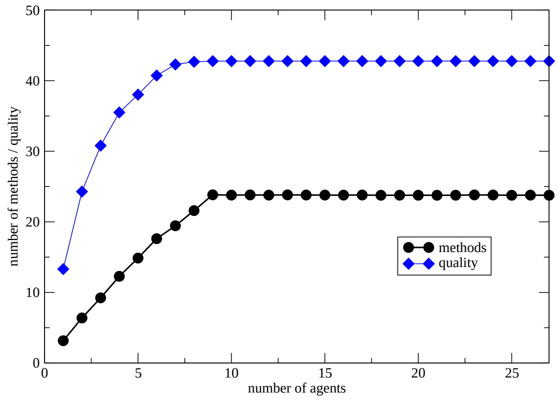

Figure 3. Number of methods and quality, for varying number of agents.

5. Conclusions

The approach presented here deals with the action-selection and sequencing problem. It aims at situations when the environment changes and so demands different organizations of tasks and agents. In other approaches, this adaptation requires a learning component, normally based on explicit coordination and/or communication.

We focus on a paradigm based on colonies of social insects, where there are plenty of evidences of ecological success, despite the apparent lack of explicit coordination. These insects adapt to the changes in the environment and to the needs of the colony using the mechanisms explained here. The key issues are the learning/forgetting specialization and the plasticity in division of labor.

Our aim is to show that such an approach can be used to allocate tasks to agents in MAS, when organizations change dynamically. As already pointed out in [2], there is no standard way of evaluating the performance of algorithms dealing with dynamic environments, because benchmark problems (e.g. travel salesman) are static problems. Therefore, we use the GPGP/DTC/TÆMS framework as comparison, although this is somehow limited to the static cases. When it comes to dynamic situations, we can only discuss the qualitative advantage of our approach. The main one is that it does not need the explicit commitments each time the number of agents changes. This is especially important when it comes to domains with large number of agents. In the scenarios we discussed here, although it takes some time for the agents to adapt, this adaptation is reached and deadlines were kept. Also, in our approach we ensure the synchronization of team members and handle teamwork redundancy. As discussed in [1], GPGP mechanisms support neither synchronization nor handle teamwork redundancy. For instance, in scenario III, more than one agent can be performing the same task but no more than one agent is able to perform the same method.

In summary, there is a tradeoff between explicit coordination leading to highly accurate outputs *versus* implicit coordination via learning and adaptation leading to more relaxed outputs (less quality, higher costs or durations in the scenarios discussed here). Our approach is certainly not the best in static situations, while it is effective in dynamic ones. The efficiency is an issue related to the specific scenarios. For instance, if, besides deadline constraints, there are also constraints related to quality or cost, then in some cases these will not be respected.

In order to tackle these limitations, we intend to work on different parameters of the functions discussed in Section 2.2 and also study new extensions to those equations so that we can accommodate a wider range of types of agents. For instance, we might need agents with shorter life spans than others (this would imply different life probability functions), or different thresholds to the tasks in order to respond faster or slower.

We also intend to compare our results with one deadline with a large number of agents. In our approach, having such a large number of agents is straightforward since they all follow the same basic specialization/plasticity model. Even if we consider the extensions just discussed, having a large number of agents would not be a problem. What makes the comparison difficult now is the lack of such a result in the literature.

Also, resources are not explicitly modelled in our approach. We decided to do this because we are still looking for a suitable model (from the theoretical biology point of view) explaining whether or not insects have a different behavioral model for tasks and resources such as food. Handling resources and increasing the range of non-local relationships are necessary extensions in order to be able to compare other scenarios already used by the GPGP/DTC/TÆMS framework. We intend to do this next. Finally, it would be desirable to have probabilistic definitions of non-local relationships (e.g. an enables exist between T_x and T_y with probability p). In this case, this would have to be extended in TÆMS as well.

References

[1] Abdallah, S., Darwish, N., and Hegazy, O. Monitoring and synchronization for teamwork in GPGP. In *Proceedings of the 2002 ACM symposium on Applied computing* (2002), pp. 288 – 293.

[2] Bonabeau, E., Sobkowski, A., Theraulaz, G., and Deneubourg, J.-L. Adaptive task allocation inspired by a model of division of labor in social insects. In *Biocomputing and Emergent Computation* (1997), D. Lundh, B. Olsson, and A. Narayanan, Eds., World Scientific, pp. 36–45.

[3] Bonabeau, E., Thraulaz, G., and Dorigo, M. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford Univ Press, 1999.

[4] Cicirello, V., and Smith, S. Improved routing wasps for distributed factory control. In *Workshop on Articial Intelligence and Manufacturing: New AI Paradigms for Manufacturing* (august 2001).

[5] Decker, K. S., and Lesser, V. R. Quantitative modeling of complex computational task environments. In *The Eleventh National Conference on Artificial Intelligence* (Washington, 1993), AAAI Press.

[6] Decker, K. S., and Lesser, V. R. Designing a family of coordination algorithms. In *The First International Conference on Multi–Agent Systems* (San Francisco, CA, 1995), AAAI Press.

[7] Gordon, D. The organization of work in social insect colonies. *Nature 380* (1996), 121–124.

[8] Horling, B., Benyo, B., and Lesser, V. Using self-diagnosis to adapt organizational structures. In *Proceedings of the 5th International Conference on Autonomous Agents* (Montreal, June 2001), ACM Press, pp. 529–536.

[9] Horling, B., Lesser, V., Vincent, R., Bazzan, A., and Xuan, P. Diagnosis as an integral part of multi-agent adaptability. In *DARPA Information Survivability Conference and Exposition* (South Carolina, January 2000), IEEE Computer Society, pp. 211–219. see also UMASS CSTR 1999-03.

[10] Lesser, V. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. *Autonomous Agents and Multi-Agent Systems 1* (January 1998), 89–111.

[11] Robison, G. E. Regulation of division of labor in insect societies. *Annual Review of Entomology 37* (1992), 637–665.

[12] Schillo, M., Fley, B., Florian, M., Hillebrandt, F., and Hinck, D. Self-organization in multiagent systems. In *Third International Workshop on Modelling Artificial Societies and Hybrid Organizations (MASHO)* (2002).

[13] Theraulaz, G., Bonabeau, E., and Deneubourg, J. Response threshold reinforcement and division of labour in insect societies. In *Proceedings of the Royal Society of London B* (2 1998), vol. 265, pp. 327–332.

[14] Wagner, T., Guralnik, V., and Phelps, J. A key-based coordination algorithm for dynamic readiness and repair service coordination. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS03)* (2003), ACM Press.

[15] Wagner, T., and Lesser, V. Design-to-Criteria Scheduling: Real-Time Agent Control. In *Proceedings of AAAI 2000 Spring Symposium on Real-Time Autonomous Systems* (Stanford, CA, March 2000), A version also available as UMASS CS Tech Report 99-58, pp. 89–96.

[16] Wilson, E. O. *Sociobiology: The New Synthesis*. Harvard Univ Press, 2000.