Ant-Based Task Allocation Among Teams
(Extended Abstract)

Fernando dos Santos and Ana L. C. Bazzan
PPGC – Universidade Federal do Rio Grande do Sul
Caixa Postal 15064, CEP 91501-970, Porto Alegre, RS, Brazil
{fsantos,bazzan}@inf.ufrgs.br

ABSTRACT

This paper addresses the problem of multiagent task allocation among teams. An approximate algorithm is presented, which is inspired in process of recruitment for cooperative transport observed in ant colonies. It is shown that eXtreme-Ants outperforms other two algorithms regarding communication and computational effort.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms

Keywords

Multiagent Task Allocation, Swarm-Intelligence

1. INTRODUCTION

Large-scale task allocation involves thousands of agents that must somehow coordinate. Scerri *et al.* call these scenarios *extreme teams* [3]. This problem can be formalized as an extended generalized assignment problem (E-GAP) whose solution is the allocation which maximizes a reward measure.

Among social insect (e.g. ants) similar problems arise. These insects are able to perform a division of labor which has been modeled mathematically, based on individual response thresholds and tasks stimuli. Need for simultaneous execution of tasks also appears in social insects, as for instance, in some species of ants, the transportation of large preys. Instead of individually transporting a large prey, groups of ants are formed to cooperatively do the transport. These groups are formed via a process called recruitment[2].

We propose a multiagent approximate task allocation algorithm, called eXtreme-Ants, which is inspired in the division of labor in social insects and in the process of recruitment present in ants. The model of division of labor offers fast and efficient decisions, while the recruitment ensures the allocation of constrained tasks that require simultaneous execution. We evaluate eXtreme-Ants in a domain independent simulator and compared it with two other algorithms for extreme teams: Swarm-GAP [1] and LA-DCOP [3]. In Low-communication Approximate DCOP (LA-DCOP) [3], an agent decides whether or not to allocate a task based both on its capability and on a threshold associated to the task. To deal with inter-task constraints, LA-DCOP uses a potential token. If an agent in LA-DCOP is able to allocate more than one task, it must select the ones that maximize its capability given its resources. This selection is a maximization problem, which can be reduced to a binary knapsack problem, proved to be NP-Complete. Another approximate algorithm which can deal with extreme teams is the Swarm-GAP [1]. An agent in Swarm-GAP decides whether or not to allocate a task based on the model of division of labor used by social insects colonies.

Due to lack of space we refer the reader to [3] and [1] for details about the E-GAP and about the mathematical model for task allocation respectively.

Cite as: Ant-Based Task Allocation Among Teams, (Extended Abstract), Fernando dos Santos, Ana L. C. Bazzan, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1183–1184

2. OUTLINE OF THE ALGORITHM

eXtreme-Ants is an approximate algorithm that facilitates the solution of E-GAPs when sub-sets of tasks must be performed simultaneously in order to produce a reward. Henceforth we refer to these tasks as AND-constrained. The internal threshold $\theta_{a_i \tau_j}$ of an agent a_i regarding a task τ_j corresponds to the inverse of the capability $Cap(a_i, \tau_j)$. Further, each task $\tau_j \in \mathcal{T}$ has an associated stimulus s_{τ_j}. Low stimulus means that a task is allocated only by agents with low thresholds (hence more capable) and vice-versa.

Since in E-GAP each task must be allocated to at most one agent, tokens are used in eXtreme-Ants . An agent that holds a token has the exclusive right to allocate the tasks contained in the token. If the agent does not allocate all the task in a token, it passes it to another agent.

To deal with AND-constrained tasks, agents in eXtreme-Ants reproduce the recruitment process of ants. When an agent notices that it is not capable of allocating all AND-constrained tasks perceived, it recruits other agents to form a group committed with the simultaneous allocation.

Algorithm 1 outlines the main steps of eXtreme-Ants. Each agent a_i reacts to two events: task perception and message arriving. When the agent perceives a set $\mathcal{T}$ of non AND-constrained tasks it creates a token to store the perceived tasks. The agent then decides whether or not to allocate the tasks, given its tendency and available resources. If some task contained in the token remains unallocated, the agent sends the token to a randomly selected agent. The

Algorithm 1: eXtreme-Ants for agent a_i

when perceived *set of tasks* $\mathcal{T}$ **do**
  try to allocate the tasks of $\mathcal{T}$ according to the tendency and respecting $a_i.res$;
  pass on unallocated tasks to randomly selected agents;

when perceived *set of AND- constrained tasks* α_k **do**
  try to allocate all tasks of α_k by itself, according to the tendency and respecting $a_i.res$;
  if *some task of* α_k *remains unallocated* **then**
  - rollbacks the previously allocated tasks of α_k;
  - start the recruitment process for α_k, sending a certain number of recruitment request, each one to a randomly selected agent, and wait for commitments;
  - finish the recruitment when all tasks of α_k have an agent committed to the simultaneous allocation;
  - abort the recruitment when a timeout is reached, releasing the committed agents (to avoid deadlocks);

agent that receives the token repeats this process.

When an agent perceives a set α_k of AND-constrained tasks (composed by τ_{k_j}) it attempts first to allocate all. Upon failing it begins a recruitment process by sending a certain number of recruitment requests for each task $\tau_{k_j} \in \alpha_k$. These requests are sent to randomly selected agents.

When an agent receives a recruitment request for a task τ_{k_j} it checks its tendency to decide whether or not to accept the request. If the request is accepted, the agent commits to the allocation of the task, reserving the amount of resources which are required by the task. It might be that more than one agent commit to the same task. In this case one is probabilistically selected based on agents' capabilities. At this moment the recruitment is finished yielding a group in which each agent is engaged in the allocation of a task $\tau_{k_j} \in \alpha_k$, enabling the simultaneous execution of all AND-constrained tasks in α_k.

3. DISCUSSION AND REMARKS

We have performed several experiments changing number of agents, task stimuli and agents' internal threshold, and other parameters. We report here the main conclusions only. Each experiment consists of 2000 tasks, grouped in five classes. Each agent has a 60% probability of having a non-zero capability for each class. In this case the agent has a randomly assigned capability ranging from 0 to 1. Regarding the AND-constrained tasks, 60% of all tasks are constrained in groups of five tasks.

Changing the number of agents from 500 to 4000, on average, the rewards from eXtreme-Ants are 25.21% $\pm$ 1.98% higher than those from Swarm-GAP and 19.13% $\pm$ 3.55% lower than those from LA-DCOP (t-test, 99% confidence). It was verified that despite the fact that agents in Swarm-GAP use almost the same percentage of resources, the total rewards are worse than those achieved by eXtreme-Ants and LA-DCOP. This is due to the way Swarm-GAP deals with AND-constrained tasks (simultaneous allocation of the AND-constrained tasks is not ensured). Thus, the agents use their resources to allocate tasks, but this allocation does not translates into a reward. Both eXtreme-Ants and LA-DCOP outperform Swarm-GAP regarding the total reward due to the existence of an explicit coordination mechanism to deal with AND-constrained tasks. LA-DCOP achieves better rewards than eXtreme-Ants because each agent maximizes its capability when allocating the tasks, taking into account the available resources. On the other hand, agents in eXtreme-Ants make a simple one-shot decision to allocate tasks. The maximization leads to a better exploitation of the agents' resources. However, there is a tradeoff between the achieved total reward and the communication/computational effort.

Communication is measured here as the sum of messages sent by the agent over all time steps, regardless of message type (e.g. token, recruitment request, etc.). On average, agents in eXtreme-Ants sent fewer messages than those in LA-DCOP and more messages than those in Swarm-GAP.

As for computational effort, this is defined here as the number of evaluated tasks by each agent at each time step. Each time the agent decides whether or not to allocate a task, an internal counter is incremented. The computational effort of Swarm-GAP is, on average, 55.06% $\pm$ 3.40% lower than those from eXtreme-Ants. Since in Swarm-GAP there is no explicit coordination mechanism to deal with AND-constrained tasks, the agents do not have to make additional evaluations regarding the simultaneous allocation of constrained tasks. eXtreme-Ants outperforms LA-DCOP.

It was verified with the experiments that the probabilistic allocation of eXtreme-Ants, based on the model of division of labor, reduces the amount of communication and the computational effort. The reduction in the computational effort is due to the simple one-shot decision, which does not require any local maximization. The low computational effort causes the reduction in the number of messages sent, since in LA-DCOP the tasks which are not selected in the local maximization are sent do other agents. In both eXtreme-Ants and LA-DCOP the presence of an efficient coordination mechanism to deal with inter-task constraints leads to better total rewards regarding Swarm-GAP.

Acknowledgments

This research is partially supported by the Air Force Office of Scientific Research (AFORS) (grant number FA9550-06-1-0517) and by the Brazilian National Council for Scientific and Technological Development (CNPq).

4. REFERENCES

[1] P. R. Ferreira, Jr., F. Boffo, and A. L. C. Bazzan. Using swarm-gap for distributed task allocation in complex scenarios. In N. Jamali, P. Scerri, and T. Sugawara, editors, *Massively Multiagent Systems*, number 5043 in Lecture Notes in Artificial Intelligence, pages 107–121. Springer, Berlin, 2008.

[2] B. Hölldobler, R. C. Stanton, and H. Markl. Recruitment and food-retrieving behavior in *Novomessor* (formicidae, hymenoptera). *Behavioral Ecology and Sociobiology*, 4(2):163–181, 1978.

[3] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Allocating tasks in extreme teams. In *Proc. of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 727–734, New York, USA, 2005. ACM Press.