Logics, local search and resource allocation

Andrea Roli and Paolo Torroni
DEIS, Università di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
{aroli,ptorroni}@deis.unibo.it

The agent metaphor, and models such as the BDI and its evolutions, make it possible to characterize a computational entity as a pro-active system, equipped with (explicit) goals, and with resources and plans to achieve them. In order to execute the actions of a plan, agents may require a number of *resources*. In the general context of a resource bounded environment, negotiation can be used to make the agent plans feasible, in a way that preserves the agent autonomy: agents can estimate the *cost* of their current plan, based on the resources that they need to obtain before they can execute it, and may negotiate in order to obtain such resources.

In [3], the authors show how a *resource reallocation problem* (r.r.p.) can be solved by means of dialogues, proposing a logic-based framework and an operational model where agents are provided with a single plan to achieve a goal. They propose an instance of such framework, called $\mathcal{N}$-system, and prove that it is able to generate dialogues, and that if there exists a solution to the r.r.p. *without needing to change the agents' plans,* then such solution (or an equivalent one) will be found by the system. This leads to defining a *weak* notion of completeness, that is, completeness with respect to solutions that do not require a modification of the agent plans.

In a more general setting, agents may have several alternative plans to achieve a goal. In a system of n agents, each provided with m alternative plans, the solution of a r.r.p. would require the selection of n plans, one for each agent. The size of the search space (m^n) makes any exhaustive search approach absolutely impractical. This work builds on [3] and combines it with metaheuristic techniques [1]. We do not want to make the system complete, but we rather aim to show experimentally how it can be improved with a limited additional computational cost. In our "combined" approach, computational logics serves as a tool to specify negotiation policies and the agent knowledge in a declarative way, and provides a theoretically founded framework for negotiation, while local search overcomes the efficiency limitations of a complete search approach. We map the plan selection problem into a minimization problem on an energy landscape, and apply metaheuristic techniques to tackle it. In particular, we define a correspondence between the cost of the agent plan and its *frustration* measuring the cost of a plan in terms of *number of missing resources*. The r.r.p. is therefore defined as the problem of re-distributing the system resources in such a way that all agents minimize their degree of frustration. A r.r.p. is called *solvable* if it is possible to decrease such degree of frustration to zero. Specifically, we consider the case of *self interested* agents, for which a negotiation process cannot possibly result in a growth of the cost of their current plan (the agents of an $\mathcal{N}$-system are self-interested).

An interesting characteristic of $\mathcal{N}$-systems is that after a set of terminated negotiation dialogue sequences, the system is in one among several possible states, characterized by having all the same frustration value. This result enables to model the plan selection problem into a minimization problem.

If we want to decrease the agents' *degree of frustration* by trying with alternative plans, there are two choices then to be made: *which agents* should change plan (agent selection problem), and *which plan* among the allowed ones has to be selected (plan selection problem). The problem thus turns out as a search problem on frustration landscape. The landscape is defined by states S, frustration values $F(S)$ and the neighborhood structure, which defines the landscape topology. The general LS algorithm is described below:

```
S_0 ← Initial_plan_assignment(); NegotiationPhase(S_0)
F(S_0) ←Eval(S_0) {Eval() counts the number of missing resources}
while Termination condition not met do
   Agent ← ChooseAgent(); ChangePlan(Agent)
   S ← New_state_after_changes(); NegotiationPhase(S) F(S) ←Eval(S)
end while
```

We implemented and tested three algorithms, which differ in the agent selection rule. The guiding heuristic, inspired by *repairing techniques* [2], tries to reduce the global frustration by selecting a frustrated agent to change plan. The choice of the alternative plan is random. The first algorithm (*Most Frustrated*, or *MF* for short) always chooses the most frustrated agent to change the plan. A randomized version (*RF*) of the previous algorithm is obtained by choosing at random a frustrated agent. These two cases represent the extremes of a probabilistic choice, based on a distribution which is a function of the agent frustration. Finally, we experimented also with an algorithm which allows more than one agent to change plan. At each iteration, each frustrated agent has probability τ to change plan, resulting in an average parallelism of $\tau n_{\text{frustrated}}$.

The algorithms have been tested on a benchmark composed of eight randomly generated problem instances with different characteristics, involving 30 agents and 50 resources each, and 5 to 10 available plans per agent. By our experiments, we observe that both *MF* and *RF* strategies quickly decrease the global frustration. Nevertheless, the choice of the most frustrated agent can lead the system to local minima, from which the algorithm is not able to escape. The random choice among frustrated agents shows its effectiveness, as it is able to found paths toward optimal frustration on most of the instances. We believe that the results obtained by simulation, with a single process implementing the local search algorithms, can be used to aid the design of a multi-agent system for negotiation. One possibility could be of course to introduce in the multi-agent system an additional agent, whose task is to "suggest" an agent to change plan. Another option could be to let frustrated agents autonomously decide to change plans, but following a protocol that guarantees that only one agent at a time can actually change plan. The design of such protocol and system architecture is subject for current work. It could be interesting in the future to experiment with other metaheuristic algorithms and introduce heuristics derived by the analysis of landscape properties.[1]

References

[1] E. Aarts and J. Lenstra, eds. *Local Search in Combinatorial Optimization*. Wiley-Interscience, 1997.

[2] S. Minton, M.D. Johnston, A.B. Philips, and P. Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58(1-3):161–205, 1992.

[3] F. Sadri, F. Toni, and P.Torroni. An abductive logic programming approach for negotiation and communication in multi-agent systems. Tech, Rep. DEIS-LIA-02-001, University of Bologna (Italy), 2002.

[1] We would like to thank Paola Mello, Michela Milano, Francesca Toni and Fariba Sadri for their interest in this work and for helpful discussions and suggestions. This work has been partially supported by the EU funded project IST-2001-32530, Societies Of ComputeeS (SOCS).