Pheromone models for numerical planning

M.Baioletti and **P.Furiani** and **A.Milani** and **V.Poggioni** and **F.Rossi**
Dipartimento Matematica e Informatica, Università di Perugia

Abstract

Ant Colony Optimization (ACO) is an optimization technique which has been successfully applied to classical planning problems in order to minimize plan length and plan execution cost. In this paper an ongoing research on the extension of ACOPlan, a planner based on the ACO approach, to numerical planning is described. The extension consists in introducing numerical aspect of domains and problems into two main parts of the ACO model: the pheromone model used by the ant colony to share information about search decisions, and the heuristic function used to guide search and solution evaluation. Two pheromone models for numerical planning are introduced and some preliminary experimental results are also presented and discussed.

1 Introduction

Numerical planning is one of the most studied extensions of the classical planning and it is of great interest from the application point of view. In fact in this model, as in many real world applications, action execution produces modifications on numerical resources (e.g. fuel consumption, available quantity of raw material, money etc.) which can affect the actual actions applicability and/or the problem goals satisfiability. In this framework, planning problems are better defined as optimization problems, where the purpose is not only to find admissible solution plans but to find high quality plans with respect to a given objective function. In fact, in these models different solutions of the same planning problem could have extremely different values of the optimization metric (e.g. finding the solution plan with the lowest energy consumption).

In the last years, several specific approaches and systems have been presented (Hoffmann 2003; Gerevini, Saetti, and Serina 2008; Chen and Hsu 2006; Do and Kambhampati 2003) to solve planning problems in numerical domains. On the other hand, several classical planning systems have been also designed in order to optimize solution plans in terms of plan length or plan execution cost (Hoffmann and Nebel 2001; Gerevini and Serina 2002; Kautz, McAllester, and Selman 1996).

One of these approaches, ACOPlan (Baioletti et al. 2009a; 2009d; 2009c; 2009b), applies Ant Colony Optimization (ACO) techniques to the optimization of plan length/cost. In ACOPlan a colony of planning ants repeatedly builds plans in a forward way by exploiting suitable heuristic function and pheromone values. It is well known that the choice of the pheromone model is a critical point in the ACO approach. In (Baioletti et al. 2010) a comparative study of different planning pheromone models concludes that fuzzy pheromone models seems to be most promising than crisp ones in the planning framework.

In this paper an ACO model for planning with numerical variables is presented. Increasing the planning model expressivity, in order to manage and optimizing numerical variables, constraints and goals, requires a deep modification of the framework with respect to the first ACOPlan proposal (Baioletti et al. 2009a). In particular there is the need for new *pheromone models* and suitable *heuristic* functions which take into account the presence of a numerical part of the problem.

The first requirement has been faced by defining and studying different "numerical" pheromone models which are used together with "logical" pheromone models: in this paper the *Bucket* and the *Weighted Average* models are presented and discussed.

In this paper, an experimental comparison between the current implementation of ACOPlan and the LPG system is presented; the choice of LPG as the first comparison basis is twofold motivated because it is one of the best planners developed in the recent years and it is based on a stochastic approach.

Although experiments and the numerical ACOPlan system are still under development, it can be already stated that the approach is feasible and very promising, since the experimental results are comparable (at least in some domains) with the results obtained with the LPG system. It is also important to point out that the aim of this preliminary comparison is to better understand the role and the contribution of the pheromone model and to preliminary quantifying the gap between ACOPlan and the state of the art.

2 Automated Planning

This paper focuses on the numerical planning model, which consists of the extension of the classical planning model to numerical variables. The numerical planning model is based on a set of propositional variables $\mathbf{V}$, a set $\mathbf{R}$ of numerical variables, and a set of actions $\mathbf{A}$. The variables of

$\mathbf{R}$ can be interpreted as representing the state of consumable/producible numerical resources.

The logical part of a state can be represented as a subset $s \subseteq \mathbf{V}$, while the numerical part consists in a vector $\mathbf{r} \in \mathbb{R}^k$, where k is the number of resources. An action $a \in \mathbf{A}$ is described as a tuple $(p(a), np(a), e^+(a), e^-(a), ne(a))$. The set $p(a) \subseteq \mathbf{V}$ contains the logical preconditions of a, while $np(a)$ are the numerical preconditions. A numerical precondition is an inequality/equality constraint written in terms of rational functions on some variables of $\mathbf{R}$. a is executable in a state $(s, \mathbf{r})$ if all the preconditions are satisfied. The execution of a on the state $(s, \mathbf{r})$ produces the state $(s', \mathbf{r}')$ by means of function $Res((s, \mathbf{r}), a)$ which applies the logical effects $e^+(a), e^-(a)$ to s and the numerical effects $ne(a)$, to the current values of the variables $\mathbf{r}$; each variable can be modified by assigning a new value computed as a rational function on $\mathbf{R}$. A planning problem is described by the tuple $(\mathbf{V}, \mathbf{R}, \mathbf{A}, s_0, \mathbf{r}_0, G, NG)$, where $s_0 \subseteq \mathbf{V}$ is the initial state, $\mathbf{r}_0$ is the initial values of the resources, $G \subseteq \mathbf{V}$ is the goal condition, NG is the numerical goal. A solution to a planning problem is a plan, i.e. a sequence of action $< a_1, \ldots, a_n >$, such that each action is executable in the corresponding state (i.e. a_1 is executable in $(s_0, \mathbf{r}_0)$, a_2 is executable in $(s_1, \mathbf{r}_1) = Res((s_0, \mathbf{r}_0), a_1)$, and so on), and the last state $(s_n, \mathbf{r}_n)$ satisfies G and NG.

A planning problem can also be formulated as an optimization problem. In this case an objective function Ψ has to be minimized or maximized. The function Ψ is a rational function on $\mathbf{R}$ and usually is a linear polynomial, interpreted as a multi–criteria plan quality. A solution plan is optimal if it minimizes/maximizes Ψ.

Since optimization planning problems are very hard to solve, a slightly easier framework is satisficing planning problems, in which the aim is to find very good quality plans, instead of optimal plans.

Many classical heuristic planners have been extended to deal with numerical resources and/or time, like for example FF (Hoffmann 2003), LPG (Gerevini, Saetti, and Serina 2008), SAPA (Do and Kambhampati 2003) or SGPlan (Chen and Hsu 2006). In these systems, heuristics that are sensitive to time and/or resource values have been introduced in order to guide the search process to try to optimize a generic metric function.

3 ACOPlan

Optimal/Satisficing Planning with numerical variables can be seen as a Combinatorial Optimization Problem, hence many techniques used for these problems could be also applied to Planning.

Recently, ACOPlan, a classical planner based on Ant Colony Optimization techniques (Dorigo and Stuetzle 2004), was introduced and described in (Baioletti et al. 2009a; 2009b; 2009d).

ACO is a metaheuristic optimization method inspired by the foraging behaviour of colony of ants. Ants uses pheromone trails as an indirect way of communication when they are looking for food. Indeed they release an amount of pheromone when they move from the nest to the food and vice versa. Moreover, they tend to follow paths which are marked with a stronger quantity of pheromone. The long term behaviour is that the ants very often "converge" to the shortest path from the nest to the food. In fact the first ant which arrives to the food has probably chosen a short path and will use and mark the same path to come back to the nest. Evaporation does the rest, because its overall effect is to penalize longer paths.

ACO is used to find optimal solution to combinatorial optimization problems by simulating a colony of artificial ants. Each ant builds a solution (composed by discrete components) which is evaluated and the pheromone associated to each component is increased by a quantity which depends on the solution quality. The process of solution construction is incremental and each component is randomly chosen with a probability distribution which depends on the corresponding pheromone value. Evaporation is simulated by decreasing all the pheromone values. Finally the search process can also be guided by means of a heuristic function.

The planner ACOPlan exploits the ACO capabilities to optimize the length of solution plans proving that the proposed technique is effective and well performing. Some results about the effectiveness of applying ACO techniques to planning are presented and discussed in (Baioletti et al. 2010). Due to these results and peculiarities of ACO techniques, we think that the ACO capabilities can be more exploited in cases where the optimization is more interesting like in numerical and temporal domains.

Algorithm 1 The algorithm ACOPlan

$\pi_{best} \leftarrow \emptyset$
InitPheromone(c_0)
for $g \leftarrow 1$ to N **do**
  for $m \leftarrow 1$ to n_a **do**
    $\pi_m \leftarrow \emptyset$
    $s \leftarrow s_0$
    $A_1 \leftarrow$ executable actions in s_0
    for $i \leftarrow 1$ to L_{max} **while** $A_i \neq \emptyset$ **and** $G \nsubseteq s$ **do**
      $a \leftarrow$ *ChooseAction*(A_i)
      extend π_m with a
      $s \leftarrow Res(s, a)$
      $A_{i+1} \leftarrow$ executable actions on s
    end for
  end for
  find π_{iter}
  update π_{best}
  UpdatePheromone(π_{best}, π_{iter}, ρ)
end for

In Algorithm 1 the basic algorithm of ACOPlan is described. A colony of n_a planning ants is created and handled for N iteration.

At each iteration, each planning ant builds a plan in a forward way, i.e. starting from the initial state s_0 and trying to reach a state in which G is satisfied. The actions are chosen in a random way, with a probability distribution affected by the pheromone values and a heuristic function.

The function *ChooseAction* chooses the action having the highest probability by means of the "classical" transition probability function used in ACO. The probability is associated to each solution component c by

$$p(c) = \frac{\tau(c)^\alpha \eta(c)^\beta}{\sum_x \tau(x)^\alpha \eta(x)^\beta}, \quad (1)$$

where c is the discrete solution component under evaluation and $\tau(c)$, $\eta(c)$ are respectively the pheromone value and the heuristic value associated to the component c. In the classical version of the planner the heuristic value $\eta(c)$ is computed by means of the FF heuristic function, while in the version with non uniform cost $\eta(c)$ is computed by means of the FFAC heuristic (Baioletti et al. 2009a; 2009d; 2009c; 2009b; 2010). The parameters α and β are used to tune the contribution of pheromone and heuristic respectively. It is important to note that usually a component c comprises the action to be chosen.

At the end of each iteration, the best plan found during the iteration, π_{iter}, is selected and, if it is also the best solution plan ever found, the global best plan π_{best} is updated.

Finally, the component pheromone values are updated and evaporated by the *UpdatePheromone* function that takes into account π_{iter}, π_{best} and the evaporation rate ρ.

ACOPlan for Numerical Planning

ACOPlan can be extended to solve satisficing planning problems with numerical variables. In this extension two different problems must be faced: to find a suitable heuristic function and to define specific pheromone models.

First of all, we have decided to introduce a second kind of pheromone specialized on the numerical part of the problem. This additional information is integrated in the transition probabilities function that becomes

$$p(c) = \frac{\tau(c)^\alpha \eta(c)^\beta \phi(c)^\gamma}{\sum_x \tau(x)^\alpha \eta(x)^\beta \phi(x)^\gamma} \quad (2)$$

where $\phi(c)$ is the contribution given by the pheromone associated to the numerical aspects of the problem tuned by the additional parameter γ. In this case the choice of each solution component is not affected only by the pheromone value $\tau(c)$ and the heuristic value $\eta(c)$ as usually, but also by the value $\phi(c)$, i.e. the pheromone value associated to numerical aspects.

This work focuses on the definition, development and test of new pheromone models that are able to manage the numerical aspects of problems and domains. The analysis proposed in this paper has the aim of understanding the contribution of the "numerical pheromone" as the only guide of the optimization process. In fact the heuristic function used here does not take into account the numerical aspects of the problem: it is just the heuristic FF already used in the classical version of the planner.

4 Pheromone Models for Resources

In this section two different pheromone models to deal with numerical quantities in ACOPlan are introduced. The pheromone values estimate how desirable is to reach a certain combination of values for the variables in $\mathbf{R}$. We assume the hypothesis that the contributions of each numerical variable to the "numerical" pheromone $\phi(c)$ are independent and the overall pheromone value can be computed by averaging the contributions:

$$\phi(c) = \frac{1}{k} \sum_{i=1}^{k} T_i(r_i) \quad (3)$$

where r_i is the value reached by the resource R_i in the state generated by executing the action related to c and $T_i(\cdot)$ is the corresponding pheromone function. Although this idea is supported by intuition, it could be misleading because of its high granularity: it can often cause the phenomenon of pheromone spreading thus resulting in pheromone values becoming too little informative. Side effects deriving from this model which can prevent the colony of ants from finding optimal solutions are mainly two:

- values that are near to a good pheromone value previously reached will have low levels of pheromonone, but they could be potentially good values to reach high quality solutions;
- only few values will have significant pheromone levels and this can cause a premature convergence.

To address these observations, the basic features required for a suitable pheromone model are (i) smoother pheromone distribution, e.g. to assign a pheromone level to all the resource values within a given range, and (ii) to avoid the premature convergence of the solution plans.

Bucket Pheromone Model

This model tries to overcome the problem of high granularity by grouping resource values in intervals (buckets). Since the planning graph has a limited number of levels, we can assume that each resource R_i is bounded in a real interval $[m_i, M_i]$. The basic idea is to divide this interval in some buckets and to associate the pheromone value to each bucket instead of to the single resource values.

The introduction of these buckets arises two main questions about the number of buckets and their width. It is obvious that the ideal number and width strongly depend on the planning problem and they could be different for each resource. Therefore, we have decided to compute dynamically the number N_i of buckets for each resource R_i using the bounds m_i and M_i, and the average change produced by the actions affecting R_i, denoted by Δ_i. We define

$$N_i = \frac{(M_i - m_i) + 1}{\Delta_i}. \quad (4)$$

The value Δ_i can be computed as

$$\Delta_i = \frac{\sum_{a \in \mathcal{C}_i} |eff(a, R_i)|}{|\mathcal{C}_i|}.$$

where $\mathcal{C}_i$ is the set of all actions affecting R_i and $eff(a, R_i)$ is the effect of the action a over the resource R_i.

Once the buckets have been created, the pheromone values are initialized; then the pheromone evaporation and update are performed when each iteration is concluded. From the evaporation point of view, the pheromone for resources behaves exactly as the pheromone for the logical component: it evaporates according to the *evaporation rate* ρ. On the other hand, the pheromone updates are applied on the buckets containing the values of the resources reached by the execution of the π_{iter} and π_{best} plans. The initial value for the pheromone, the evaporation rate and the contributions given by the two solution plans are parameters of the system.

Weighted Average Pheromone Model

In this model the pheromone is deposited to the single value assumed by the numerical resources during the execution of the plans π_{iter} and π_{best}. For each resource R_i, the model can be formalized by a vector Φ_i of pairs (r_i, v_i), where r_i is a value assumed by the resource R_i and v_i is the corresponding pheromone amount.

In the update phase, the best plans π_{iter} and π_{best} are executed, finding all the values reached by each resource in the plan trajectories. When a new value r_i is reached by the resource R_i, i.e. a value never reached before, r_i is added to the corresponding vector Φ_i and its pheromone amount is initialized to a default value. When r_i is a value already inserted in Φ_i, the corresponding pheromone level is increased by an amount, which is usually $\frac{2}{3}\rho$ for π_{iter} and $\frac{1}{3}\rho$ for π_{best}.

In this way, for each resource, the values which are most frequently reached by the best plans will get a greater pheromone value: the idea is to "reward" these values during the plan search phase.

Although the model deposits the pheromone quantity only to the resource values reached by "good" plans, a pheromone value is also implicitly assigned to the resource values which are neighbors of a resource value which has been actually reached. To extract these values, the following function T_i maps any value r of the resource R_i to a pheromone level in $[0, 1]$ obtained as a weighted average, as defined in

$$T_i(r) = \frac{\sum v_i W(r_i - r)}{\sum W(r_i - r)} \tag{5}$$

where $W(x) = exp(-t \cdot x^2)$ and t is a tuning parameter.

5 Experiments

An experimental test plan has been designed in order to evaluate the effectiveness of the proposed system.

The goal of the first set of tests is to compare ACOPlan with some planners that have demonstrated state of the art performances in numeric domains.

The test domains have been taken from domains of the last planning competitions which are most significant from the numerical point of view.

Although the complete test plan is still ongoing, preliminary results are already available for numerical ACOPlan with *bucket pheromone model* compared with LPG system. The test results have been obtained on numerical versions of the domains Depots and Driverlog (from the 3rd IPC). The choice of LPG as the first comparison basis is twofold motivated because both it is one of the best performing planner developed in recent years and it uses a stochastic approach.

Since both ACOPlan and LPG are non deterministic, the results are presented both in terms of minimum quality and average quality computed on 10 runs. The experiments run on a Intel Core 2 Quad 2.40GHz with 4GB RAM, using a time–out of 1200 seconds.

In Fig.1 and Fig.2 some results for the minimum quality and the average best quality for ACOPlan and LPG are shown. Each symbol represents a problem solved by both the systems: if the symbol is above the diagonal, then ACOPlan performed better than LPG and vice versa; the distance from the diagonal is proportional to the performance gap between the two systems. From the figures we can see that LPG performs better in the most cases, but the systems are comparable. Since ACOPlan is still under development, it is possible to deduce that the approach is feasible and promising. Moreover, there is room for improvement because ACOPlan currently is not using a fully informed heuristic function.

The aim of the second set of tests, still ongoing, is to investigate the impact of the new pheromone models, i.e. the Bucket Model and the Weighted Average Model, on the overall performance of the ACOPlan framework and to compare the two pheromone models in order to understand their limits and their strengths.

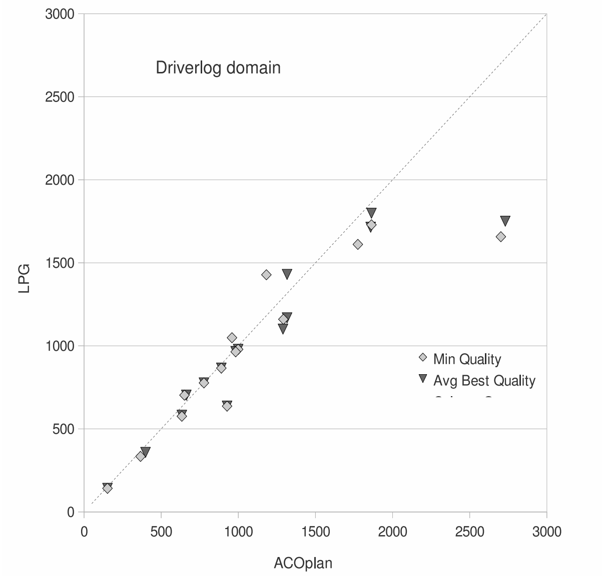

Figure 1: Scatter plot comparing ACOPlan and LPG results for *Driverlog* domain. On the x-axis and y-axis the minimum quality and the average best quality found by ACOPlan and LPG are respectively represented.

6 Conclusions and Future Works

In this paper a first version of the ACOPlan framework for numerical planning has been described. Increasing the plan-

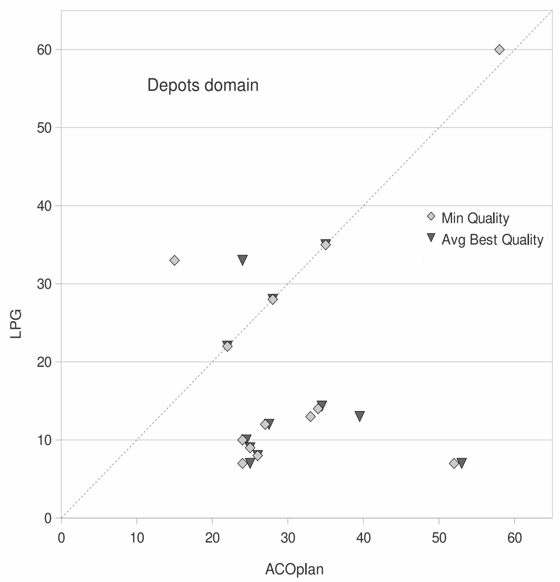

Figure 2: Scatter plot comparing ACOPlan and LPG on best quality for *Depots* domain. On the x-axis and y-axis the minimum quality and the average best quality found by ACOPlan and LPG are respectively represented.

ning model expressivity requires some deep modifications of the framework ACOPlan. To reach this aim, two pheromone models for numerical planning, the Bucket model and the Weighted Average model, have been introduced in this work, while an ongoing research focuses on the definition of a specific heuristic taking into account numerical aspects. In this way, the main idea is to compute a sort of "implicit" action costs (defined in terms of numerical effects and objective function) to be used on the FFAC *cost–sensible* heuristic function already presented in (Baioletti et al. 2010). The combined use of an effective pheromone model and a more informative heuristic function for numerical planning is expected to greatly enhance the performances of numerical ACOPlan.

Preliminary results of the comparison between the current implementation of ACOPlan with the Bucket pheromone model and the well known LPG planning system have been presented and discussed. Although experiments and the numerical ACOPlan system are still under development, it can be already stated that the approach is feasible and very promising, since the experimental results are comparable, at least in some domains, with the results obtained with the LPG system. The comparability, at least in some domain, with LPG can be anyway considered a good and encouraging result considering that the actual version of ACOPlan does not use a specific heuristic for numerical planning.

Moreover, other systematic experimental tests have been designed in order to have an estimate of the impact in the numerical ACOPlan performances of the proposed numerical pheromone models and to have comparisons with other state of art planners well performing in numerical domains (like for example SAPA (Do and Kambhampati 2003) and SGPlan (Chen and Hsu 2006)).

References

Baioletti, M.; Milani, A.; Poggioni, V.; and Rossi, F. 2009a. An ACO approach to planning. In *Proc of the 9th European Conference on Evolutionary Computation in Combinatorial Optimisation, EVOCOP 2009*.

Baioletti, M.; Milani, A.; Poggioni, V.; and Rossi, F. 2009b. Ant search strategies for planning optimization. In *Proc of the International Conference on Planning and Scheduling, ICAPS 2009*.

Baioletti, M.; Milani, A.; Poggioni, V.; and Rossi, F. 2009c. Optimal planning with ACO. In *Proc of AI*IA 2009, LNCS 5883*, 212–221.

Baioletti, M.; Milani, A.; Poggioni, V.; and Rossi, F. 2009d. PlACO: Planning with Ants. In *Proc of The 22nd International FLAIRS Conference. AAAI Press*.

Baioletti, M.; Milani, A.; Poggioni, V.; and Rossi, F. 2010. Experimental evaluation of pheromone models in ACOPlan. *Submitted to Annals of Mathematics and Artificial Intelligence*.

Chen, Y., and Hsu, C.and Wah, B. 2006. Temporal planning using subgoal partitioning and resolution in sgplan. *Journal of Artificial Intelligence Research (JAIR)* 26:323–369.

Do, M., and Kambhampati, S. 2003. Sapa: A multi–objective metric temporal planner. *Journal of Artificial Intelligence Research (JAIR)* 155–194.

Dorigo, M., and Stuetzle, T. 2004. *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press,.

Gerevini, A., and Serina, I. 2002. LPG: a Planner based on Local Search for Planning Graphs. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS'02), AAAI Press, Toulouse, France*.

Gerevini, A.; Saetti, A.; and Serina, I. 2008. An approach to efficient planning with numerical fluents and multi-criteria plan quality. *Artificial Intelligence* 172(8-9):899–944.

Hoffmann, J., and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research* 14:253 – 302.

Hoffmann, J. 2003. The metric-ff planning system: Translating "ignoring delete lists" to numerical state variables. *Journal of Artificial Intelligence Research* 20.

Kautz, H.; McAllester, D.; and Selman, B. 1996. Encoding plans in propositional logic. In *Proceedings of KR-96, Cambridge, Massachusetts, USA.*