Applying Ant Systems to two real-life assignment problems

Héctor Cancela
cancela@fing.edu.uy
Investigación Operativa - InCo - Pedeciba Informática
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

Reporte Técnico INCO 00.04
Mayo 2000

Abstract

Ant Systems (AS) is a recently proposed meta-heuristic inspired on biological behaviors, which has been applied to a variety of combinatorial optimization problems, including the QAP (Quadratic Assignment Problem).

In this work, we have studied the adaptation of the Ant Systems meta-heuristic to two different real-life assignment problems, which appear in educational institutions: the timetabling problem (assigning courses to classrooms and times), and the assignment of final projects to students. There is no standard definition for these problems, as in each institution the rules and objectives are different. We have been successful in adapting AS to tackle these problems as defined by our institution rules, showing the adaptability of this meta-heuristic to complex, real-life problems. In both cases the AS meta-heuristic obtained good quality solutions (in the timetabling case, at the cost of longer running times).
Key words — Combinatorial Optimization, Meta-heuristics, Assignment.

1 Introduction

The objective of this work is to study the application of the Ant Systems meta-heuristic to two real-life assignment problems. This meta-heuristic has been developed recently, and most of the literature concerns its application to benchmark combinatorial optimization problems such as the TSP, the VRP, the SOP, the QAP; but very little research seems to have been done on its application to real-life combinatorial problems, leaving it open the question of its applicability in more general contexts.

The paper is organized as follows. Section 2 presents the Ant Systems meta-heuristic. Section 3 presents the two problems to be studied, a timetabling problem and a project assignment problem occurring at the Facultad de Ingenieria of the UDELAR (Universidad de la República, Uruguay). Section 4 discusses some aspects of the adaptation of Ant Systems to the previously mentioned real-life problems, and presents some results of the experiments based on the implementation proposed. Finally, the conclusions of this work are presented in Section 5.

2 The Ant Systems meta-heuristic

The Ant Systems meta-heuristic is a recently developed approach to combinatorial optimization problems, which is based on the observation of the behavior of a very common and successful biological system, namely ant colonies. In particular, when individual ants search for food in the neighborhood of their nest, they deposit a chemical substance, called a *pheromone*. Other ants can detect this substance with their antennae. In the absence of pheromone, the ants explores the surrounding area in a random manner; but if pheromone trails exist, the ants will follow the "stronger" trails with large probability. This gives rise to a positive feedback effect, as these new ants will reinforce the existing paths with their own pheromone trails.

The indirect communication arising from this behavior allows to share the experience between all individual ants, and to find good paths between the nest and the food sources, even if each ant has an extremely limited view of the surrounding area.

These natural systems inspired Colorni, Dorigo and Maniezzo [CDM92] to design a new heuristic, based on the idea of considering artificial ant processes which build solutions to the optimization problem, and share their experience by means of an artificial pheromone trail, in effect a numerical common memory which can be read and modified by the ant processes to reflect the use of building blocks in the problem solution. Since then, a number of changes and extensions have been proposed to the original model, to improve its efficiency and to solve its most important shortcomings.

The Ant Systems meta-heuristic has been proposed [Tai99] as a model which generalizes the first heuristics based on ants behavior, and can take into account later extensions in a single framework. To do this, the original analogy is extended by considering the existence of a "queen" process; as the queen of the ant nest creates new ants, determines their type, and generally co-ordinates the whole colony, the artificial queen process will co-ordinate all artificial ants processes.

Following [Tai99], we consider that the model is based on the following three analogies:

- Real ants correspond to processes in charge of building solutions to the combinatorial problem considered; these processes are often referred as *Artificial ants*, or *Ant* processes.

- The pheromone trails correspond to a common memory, recording a value associated with each component of a solution, that is updated on the basis of the solutions built by the Ant processes.

- The queen corresponds to a central process in charge of activating and co-ordinating Ant processes, and of managing the common memory.

An Ant System is described by a set of Ant processes that receive problem data, memory state, and other parameters from the Queen process; build a new solution probabilistically, and send the new solution to the Queen process; and a Queen process wich initializes the memory, and then iterates choosing parameters to create new Ant processes, receiving the solutions built by the processes and updating the memory, until a stopping criterion is met; in which case returns the best solutions produced by the system.

To apply the Ant System meta-heuristic to any given combinatorial problem, it is necessary to define: 1) The type of information memorized. 2) The memory update rules applied by the Queen process. 3) The constructive procedure used by the Ant processes.

3 The assignment problems

The Ant Systems literature has been mainly concerned with the application of this meta-heuristic to a number of benchmark combinatorial problems, such as the TSP[DG97] and VRP [GTA99], the QAP [Stü99], the SOP [GD97], and the Job Shop Scheduling problem [CDMT94], with good results. Nevertheless, two common features among these problems is that they have a simple objective function, and that the domain constraints are (relatively) few. In real-life problems, the situation is generaly the inverse, as we may have more than one objective, and a big number of different domain constraints, which must be taken into account.

We have studied the application of Ant Systems to the solution of two real-life assignment problems which are present in our University (as in many other educational institutions). The first one is the timetabling problem, consisting in assigning classrooms and times to the different courses; the second one is the assignment of final projects to students.

Problems of the timetabling kind have received considerable attention in the literature (see for example [Her91, DS97, de 97]); there are many different formulations, which vary in the constraints and objectives. The first step in our work was to analyze how our University (implicitly) defined this problem; the main points are the following:

- There are a number of classrooms, with different capacity; the classrooms can be used from Monday to Saturday, from 08:00 to 24:00.
- There are a number of titulations (degrees); each degree is organized in semesters, there being a number of courses corresponding to the degrees in each semester. A course can be shared among different titulations; also, a course can be divided in groups, when the number of prospective students is too big.
- Each group should receive a given number of weekly lessons, of given duration (for example, two lessons of 90 minutes; or three lessons of 120 minutes, etc.).
- The students are enrolled in the titulations, and we know (approximately) the number of students in each semester for each titulation.

Based on these entities, there are a number of "hard" constraints, which must be satisfied: each group of each course must be assigned a classroom for each weekly lesson, with capacity at least equal to the expected number of students; a classroom can not be assigned to two groups at the same time; two courses corresponding to the same semester of the same titulation can not have any of their weekly lessons assigned at the same day and time of the day. There are also other "soft" constraints, which can be relaxed: the teachers of the courses can state their preferred time of the day; there should not be (if possible) empty time between lessons of different courses of the same semester of the same titulation; the weekly lessons of a course should be "uniformely distributed" in the week if possible (for example, two weekly lessons on Tuesday and Thursday are better distributed than on Monday and Tuesday, or on Monday and Friday). The objective is to find a feasible solution, or at least a solution which satisfies all hard constraints and "least violates" all soft ones.

A typical problem size for the Faculty of Engineering involves 8 titulations, 10 semesters per titulation, between 3 and 8 courses per titulation per semester, 30 classrooms and 5000 students. In our case we worked with a subset of the problem, involving only the most difficult task, namely the courses of the first semester of all titulations, which involves 4 courses (divided in 26 groups) and about 1200 students.

The second problem is less complex, both in the number and type of entities involved, and in the constraints to be taken into account. It consists in assigning (groups of) students to projects. The entities involved are:

- Projects: each project is proposed by a teacher, has a minimum and maximum group size, and a minimum and a maximum number of groups involved. The teachers may state their preferences for the students involved.
- Students: each student can state his preferences for each of the proposed projects; and also can state his affinity for working with other fellow students in each of the proposed projects. The affinities need not be coherent over the projects (student A may want to work with student B for project P1 but with student C in project P2), and need not reciprocal (student A may want to work with student B, who in his turn wants to work with C).

The constraints are that each student must be assigned to a project and that each project must either be assigned a number of groups between the minimum and maximum stated (each group composed by a number of students between the given minimum and maximum group size) or not be assigned at all. The objective is to maximize the satisfaction of both teachers and students.

An interesting point of this problem is the fact that the restrictions are stated in term of groups of students, but the assignment in itself is at the student level (not at the group one); this prevents it from being easily classified among standard assignment problems.

A typical problem size for the Computer Engineering degree involves about 40 projects and 100 students, most projects involving one or two groups of between 1 and 3 students.

4 Implementation and experimentation

The following step in this work consisted in adapting the AS meta-heuristic for the solution of the two problems stated in the previous section. As usual with optimization meta-heuristics, the implementations were different for each problem, in order to reflect its special characteristics.

For the timetabling problem, the memory was defined as a three-dimensional matrix, indexed by the lessons (belonging to a group, which in turn belongs to a course), the classrooms, and the times of the day (discretized in half-hour intervals). For the project assignment problem, the memory was defined as a two-dimensional matrix, indexed by the students and the groups of the projects. In both cases, a memory entry will be higher when that particular sub-assignment has appeared in good-quality solutions.

We experimented with different memory update rules. In particular, we compared the case where all ants were allowed to contribute to the memory update, with the case where at each iteration, only the ant which found the best solution was taken into account in order to update the memory (this is referred to in the literature as an elitist strategy); in both problems the elistist strategy had superior results. The memory update includes both a "forget" mechanism (at each iteration, old entries are exponentially smoothed), and a "reinforcement" mechanism (in the case of the timetable, if a lesson has been assigned to a classroom at a given time of the day in the best solution, then a positive number is added to the corresponding entry in the matrix; this number is inversely related to the cost, in order to reinforce low-cost solutions).

The usual constructive procedure employed by the Ant processes is based on the iterative building of partial solutions; at each step a new element is added to the solution. The choice is probabilistic, guided partially by the common memory (which reflects the frequency of use of partial assignments), and partially by some greedy evaluation function (usually called the visibility). In our case, the elements of the solution are partial assignments (of a lesson to a classroom at a given time, or of a student to a group of a given project). The evaluation function takes into account the suitability of this partial assignment, independently from the rest of the solution.

Two parameters control the relative weights of the memory and the greedy function. Tere is no general procedure to fix their values, which are often chosen empirically. We also experimented with the sensibility of the AS meta-heuristic with respect to these parameters, obtaining in both problems results similar to the ones published for benchmark problems in the existing literature.

The implementations were run both on “toy” problems, to debug the programming and adjust some parameters of the algorithm, and on real data corresponding to a particular semester. In the case of the project assignment problem, the AS meta-heuristic was very successful, obtaining in some minutes solutions which used to take many hours of a skilled person work. In the case of the timetabling problem, we also obtained feasible solutions in few iterations; but enforcing the “soft” constraints required running times considerably higher. It is important to note that the solutions which are applied in the institution, and which are obtained by manually applying informal heuristics, usually result in the violation of some constraints and/or the redefinition of the problem (i.e., groups can be split or joined, time restrictions changed, etc.).

5 Conclusions

In this paper we have presented the AS meta-heuristic and its application to two real-life assignment problems, showing that this method can be applied not only to benchmark problems, but is also useful in complex real-life combinatorial optimization problems.

Among many open questions, it would be of particular interest to further study the dependency of the AS meta-heuristic on the parameters which control the relative influence of the memory and the greedy function, and if possible to define an automatic procedure to fix their values.

Acknowledgment

The author wants to thank students Marcos Carlevaro, Santiago Costabel and Fernando Rey, which participated in the adaptation and implementation of the AS meta-heuristic under his direction.

References

[CDM92] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed optimization by ant colonies. In F.J. Varela and P. Bourgine, editors, *Toward a practice of autonomous systems: proceedings of the First European Conference on Artificial Life (ECAL 91)*, pages 134–142. MIT Press, Cambridge, 1992.

[CDMT94] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. Ant system for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1):39–53, 1994.

[de 97] D. de Werra. The combinatorics of timetabling. *EJOR*, 96:504–513, 1997.

[DG97] M. Dorigo and L.-M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1:53–66, 1997.

[DS97] Andreas Drexl and Frank Salewski. Distribution requirements and compactness constraints in school timetabling. *EJOR*, 102:193–214, 1997.

[GD97] L.M. Gambardella and M. Dorigo. HAS-SOP: Hybrid ant system for the sequential ordering problem. Research Report IDSIA-11-97, IDSIA, 1997.

[GTA99] L.M. Gambardella, É. Taillard, and G. Agazzi. MACS-VRPTW: a multiple Ant Colony System for Vehicle Routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.

[Her91] A. Hertz. Tabu search for large scale timetabling problems. *EJOR*, 54:39–47, 1991.

[Stü99] Thomas Stützle. ACO algorithms for the Qadratic Assignment Problem. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.

[Tai99] Éric D. Taillard. Ant systems. Research Report IDSIA-05-99, IDSIA, February 1999.