Hypercube FrameWork for ACO applied to timetabling

Franklin Johnson[1], Broderick Crawford[1], and Wenceslao Palma[1]

1   Pontificia Universidad Católica de Valparaíso, Escuela de
Ingenieria Informática, Valparaiso, Chile
franklin.johnson.p@mail.ucv.cl
{broderick.crawford,wenceslao.palma}@ucv.cl

Abstract. We present a resolution technique of the University course Timetabling problem (UCTP), this technique is based in the implementation of Hypercube framework using the Max-Min Ant System. We presented the structure of the problem and the design of resolution using this framework.
A simplification of the UCTP problem is used, involving three types of hard restrictions and three types of soft restrictions. We solve experimental instances and competition instances the results are presented of comparative form to other techniques. We presented an appropriate construction graph and pheromone matrix representation. A representative instance is solved in addition to the schedules of the school of Computer science engineering of the Catholic University of Valparaiso. The results obtained for this instance appear. Finally the conclusions are given.

1   Introduction

The Timetabling problems are faced periodically by each school, college and university in the world. In a basic problem, a set of events (particular classes, conferences, classes, etc) must be assigned to a set of hours of a way that all the students can attend all of their respective events. With the reservation of which restrictions of hard type which necessarily they must be satisfied and soft restrictions exist that deteriorate the quality of the generated schedule. Of course, the difficulty of any particular case of the UCTP [1] [2] depends on many factors and in addition the assignment of rooms perceivably makes the problem more difficult in general.

Many techniques have been used in the resolution of this problematic one, between these we can find evolutionary algorithms, simulated annealing, and tabu-search. Other technique has presented good results is the genetic algorithms [3]. But we looked for here specifically to represent the resolution through the ant colony optimization (ACO) and through the implementation of Hypercube framework for Max-Min Ant System (abbreviation in Spanish MTH-SHMM). We give a representation for the problem, generating an appropriate construction graph and the respective pheromone matrix associated.

In the following sections we present the UCTP problem, the problem design for Hypercube framework. The instances of the problem used and the results of the experimentation. Finally the conclusions of the work appear.

2   University Course Timetabling Problem (UCTP)

2.1   Problem description

The problem timetabling considered to make this study similar to one is presented initially by Paechter in [4]. Timetabling of university courses is a simplification of a typical problem [5]. It consists of a set of events E and must to be scheduled in a set of timeslots $T = \{t_1, ..., t_k\}$ ($k = 45$, they correspond to 5 days of 9 hours each), a set of rooms R in which the events will have effect, a set of students S who attend the events, and a set of features F required by the events and satisfied by the rooms. Each student attends a number of events and each room has a maximum capacity. A feasible timetable is one in which all the events have bee assigned a timeslot and a room so that the following hard constraints are satisfied:
   * No student attends more than one event at the same time;
   * The rooms must be sufficiently great for all students who attend a class and to satisfy all the features required by the event;
   * Only one event is in each room at any timeslot.

In addition, All possible timetable generated is penalized for each occurrence according to the number of violations that exists of the soft constraint of problem. Some of these restrictions appear next:
   * A student has a class in the last slot of the day;
   * A student has more than two classes in a row;
   * A student has exactly one class on a day.
Feasible solutions are always considered to be superior to infeasible solutions, independently of the numbers of soft constraint violations. In fact, in any comparison, all infeasible solutions are to be considered equally worthless. The objective is to minimize the number of soft constraint violations in a feasible solution.

3   Design of Hypercube Framework SHMM for Timetabling (MTH-SHMM)

3.1   Resolution Structure

Given restrictions presented in the previous section and the characteristics of problem, we can now consider the option to design an effective MTH-SHMM for the UCTP. We have to decide how to transform the assignment problem (to assign events to *timeslots*) into an optimal path problem which the ants can solve [12]. To

do this we must create an appropriate construction graph for the ants to follow. We must then decide on an appropriate pheromone matrix and heuristic information to influence the paths the ants will take through the graph.

We present the principal elements used to generate the UCTP solutions, presenting in a figure 1 these three elements.
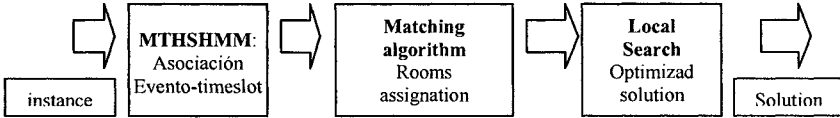
Figure 1. An instance of the problem is received like input, this it happens through an association process *event-timeslot*, assigns events to a *timeslot*, later a matching algorithm [8] is used for makes the assignation from rooms to each one of events associated to *timeslot*. In this point a solution is complete, but is low quality. Then a local search algorithm [16] is applied that improves the quality of the solution and gives like final result one optimal solution to the UCTP.

3.2   Construction graph

One of the main elements of the ACO metaheurístic is the power to model to the problem on construction graph [6] [7], that way a trajectory through the graph represents a problem solution. In this formulation of the UCTP it is required to assign each one of $|E|$ events to $|T|$ *timeslots*. Where direct representation of the construction graph this dice by $E \times T$; east dice graph we can then establish that the ants walk throughout a list of events, choosing *timeslot* for each event. The ants follow one list of events, and for each event *and,* the ants decide *timeslot t.* each event a this single time in *timeslot*, thus in each step an ant chooses any possible transition as it is in the figure 2.
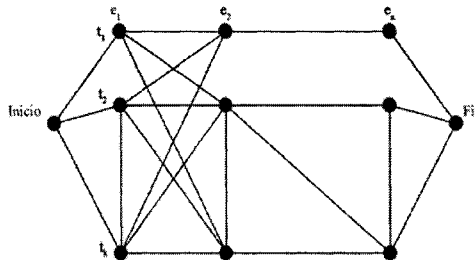
Figure 2. Each ant follows a list of events, and for each event $e \in E$, an ant chooses a *timeslot* $t \in T$.

The ants travel through the construction graph selecting ways of probabilistically way. Using the following function:

$$p_{(e,t_i)} = \frac{\left(\tau_{(e,t_i)}\right)^{\alpha}}{\sum_{\theta \in T} \left(\tau_{(e,t_\theta)}\right)^{\alpha}} \tag{1}$$

This probability function is come off the used in [6]. This function of probability directly depends on the pheromone information τ, they have the possible ways to follow. The parameter α is like appendix of the original function in [6], for this case its value is 1.

3.3   The Pheromone matrix

In search of a pheromone matrix we represented that pheromones indicates the absolute position where the events must be placed. With this representation the pheromone matrix is given by $\tau (A_i) = \tau$, $i=1,...,|E|$, the pheromone does not depend on the partial assignments A_i. It can to observe that in this case the pheromone will be associated with nodes in the construction graph rather than edges between the nodes.

A disadvantage of this directs pheromone representation is that the absolute position of events in the *timeslots* it does not matter very much in producing a good timetable. The relative placement of events is more important. For example, given a perfect timetable, it is usually possible to permute many groups of *timeslots* without affecting the quality of the timetable.

By another side we defined that for the use of the heuristic information η it must use a function that calculates a weighted sum of several or all of the soft and hard constrains in each assignation, which is to incur very high a computational cost stops this class of problem [8]. For this we will not use east type of information to orient the route of the ants.

3.4   Algorithm Description

We show the general structure of the algorithm, in which some modifications are made of presented in [9] [11]. A new assignation values to τ_{max}, τ_{min} ,a new pheromone update rules. We define the assignment A_i like the *timeslot* selected for the event i. The algorithm is the following:

1 Input: Problem instance I
2 $\tau_{max} \leftarrow 1$
3 $\tau_{(e,\,t)} \leftarrow \tau_{max} \ \forall \ (e,\,t) \in E \times T$
4 calculate $c(e,\,e') \ \forall \ (e,\,e') \in E^2$
5 calculate $d(e)$
6 sort E according $\ll$, resulting in $e_1 \ll e_2 \ll \ldots \ll e_n$
7 while time limit not reached do
8   for $a = 1$ to m do
9   {construction process of ant a}
10      $A_0 \leftarrow \varnothing$
11   for $i = 1$ to $|E|$ do
12   chooser timeslots t according to probabilities $p_{(e_i,\,t)}$ for event e_i
13          $A_i \leftarrow A_{i-1} \cup \{(e_i,\,t)\}$
14   end for
15      $C \leftarrow$ matching_algorithm (A_n)
16      $C_{best\_iteration} \leftarrow$ best of C and $C_{best\_iteration}$
17   end for
18   $C_{best\_iteration} \leftarrow$ applying local search to $C_{best\_iteration}$
19   $C_{global\_best} \leftarrow$ best of $C_{best\_iteration}$ y C_{global_best}
20 global pheromone update for τ using C_{global_best}, implicated to MTH
21 end while
22 Output: An optimized candidate solution C_{global_best} for I

Only the solution that causes the fewest number of hard constraint violations is selected for improvement by the Local Search. The pheromone matrix is updated only once by each iteration, and the global best solution is used for the update. Then A_{global_best} be the assignment of the best candidate solution C_{global_best} found since the beginning. The following update rule is used:

$$\tau(e,t) = \begin{cases} \rho \cdot \tau_{(e,t)} + (1-\rho) \cdot \Delta \tau_{(e,\,t)}^{upd} & \text{if } A_{major\_global}(e) = t, \\ \rho \cdot \tau_{(e,t)} & \text{otherwise} \end{cases}$$

$$(2)$$

In the right part of the equation is reduced to $\Delta \tau_{(e,t)}^{upd} = 1$ if $A_{global\_best}(e) = t$ and 0 in otherwise. Where A_{global_best} is the solution used for pheromone update. With this update rule to makes sure that the pheromone values of the graph, are going to be always between values $[0,1]$. The rate of evaporation $\rho \in [0.1]$.

4   Experimentation

4.1   Tests

The algorithm was implemented in C++ programming language, under Linux system using GNU G++ compiler GCC 2.96. The behavior of Hypercube framework Max-Min Ant System (MTH-MMAS) was observed in the resolution of the UCTP. The used instances appear to continuation.

Instances 1: Instances of the UCTP are structured using a generator described in [10]. This generator allows generating classes of instances *small, medium*, which reflect varied problems of timetabling of several sizes.

Instances 2: In addition it was used a series of 20 instances created for International Timetabling Competition, these instances is made with the same generator used in instances 1.

The parameters study is made initially, to evaluate the best values than they must to assume these parameters. The small (small1) instances was used for using the MTH-MMAS without local search making evaluations with different ants numbers *m* and with different evaporations factors ρ, the parameters of $\alpha = 1$, number on attempts = 10 and a maximum time by attempt = 90 seconds for all the tests. The results are in the following table.

Table 1. It presents the best results obtained when proving the instance small1.tim varying ants number *m* and evaporation factor ρ.

| Best solutions MTH-SHMM					
m	Evaluation	T° seg.	ρ	Evaluation	T° seg.
5	17	6,79	0,2	15	7,11
10	16	7,46	0,5	13	8,1
20	16	6,06	0,8	17	6,79

In the table to be observed the best results are obtained using the parameter *m*=20 obtaining a evaluation of 16 in 6.06 seconds. And for the case of evaporation factor the best value is =0,5 in 8.1 seconds.

The values shown in the tables previously presented they belong to a series of executions that allow of experimental form to determine as are more advisable parameters to use in the execution of the algorithm of MTH-MMAS. This way we compared the algorithm of the Max-Min Ant System with and without Hypercube framework, in addition the local search is included to increment the quality of the solutions in different instances.

4.2   Distribution results

We show a graphic in which they are a series of boxplot which they represent relative distribution of the number of constrained violations for hypercube

framework Max-Min Ant System (MTH-SHMM) and the Max-Min Ant System (MMAS) pure for all the instances of type small and medium with which they were proven.
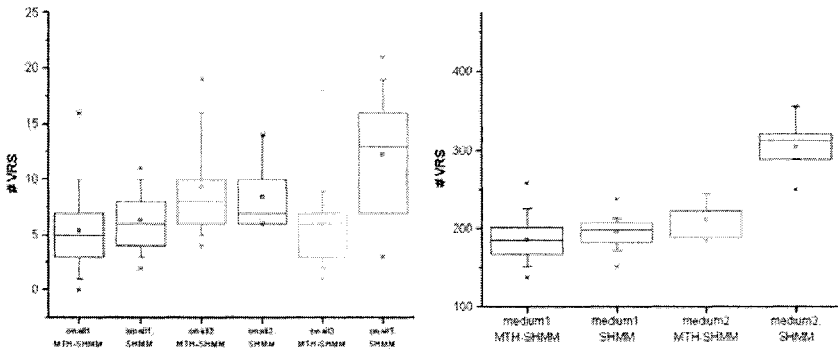
Figure 3. Like it is possible to be observed in most of the results for the different instance types, the obtained results using hypercube framework they are of better quality (smaller violation of soft constrain VRS) since 50% of the data represented by a horizontal line always are under the level of the same instance for the Max-Min Ant System (SHMM) pure.

4.3 Comparison with other techniques

Here it present a comparative picture between the solutions obtained for different instances for the UCTP doing use of different techniques like Simulated annealing, advanced search and simulated annealing with local search [13][14][16]. The results obtained for the competition instances appear.

Table 2. It present the best results obtained when proving the instances of the International Timetabling Competition compared with other techniques.

Technique	1	2	3	4	5	6	7	8	9	10
SA	45	25	65	115	102	13	44	29	17	61
AS	257	112	266	441	299	209	99	194	175	308
SA-LS	211	128	213	408	312	169	281	214	164	222
MTH-MMAS	270	193	294	586	406	221	305	244	201	358

Technique	11	12	13	14	15	16	17	18	19	20
SA	44	107	78	52	24	22	86	31	44	7
AS	273	242	364	156	95	171	148	117	414	113
SA-LS	196	282	315	345	185	185	409	153	281	106
MTH-MMAS	268	312	341	403	222	234	371	184	345	201

For these instances and compared with the other solutions the MTH-MMAS it present two characteristics a to evaluate; first it has the capacity to generate feasible solutions for these instances. These instances are of great you make difficult since they are for Timetabling competitions. Second the quality of the generated solutions is of very low category compared with the technique based on Simulated Annealing,

which has the best found historical results for these instances, but in comparison with the other instances do not present great difference. These evaluations are not feasible in order to decide if a technique is better than other, since the differences in variable results can be for different external variables.

To continuation it presents the comparison for the small and medium instances. We will compare the algorithm of MTH-MMAS and the MMAS pure with respect to Ant Colony System algorithm of Krzysztof Socha (ACS) and to algorithm based on random restart local search (RRLS).

Tabla 3. It present best results obtained when proving the test instances small and medium.

Technique	Small1	small2	small3	medium1	medium2
RRLS	11	8	11	199	202
ACS	1	3	1	195	184
MMAS	3	6	3	152	250
MTH-MMAS	0	4	1	138	186

As it is possible to be observed for these instances in the MTH-MMAS present a superiority in the quality of the generated solutions (smaller VRS). Always by on the quality the solutions generated with the MMAS. We can to say that the hypercube framework it improves the quality of the ant algorithm applied.

5   In the Practices

5.1   UCV Instance

As a form to approach investigation of the project to a practice plane we implemented a resolution for the UCTP using the Hypercube framework Max-Min Ant System. This problematic is common and it is present in all type of institution of study. It is by that one has been implemented resolution to this problematic creating an instance of the problem for the Catholic University of Valparaiso and specifically for the school of Informatics Engineering.

A tool in C language was implemented, to which him the courses enter indicating the semester, assistants, if it has assistantship, times to the week that are dictated and his characteristics. In addition the rooms are entered to him, certain their capacity and characteristics. The system generates an instance introducing a factor of correlation between the events, generating therefore an instance with the same format that those of competition, small, medium. Stored this information in file ucv.tim. Ready pair to use by the MTH-MMAS algorithm

Instance characteristic: Total number of rooms and laboratories: 9. total number of event: 194, total Attending: 600. Number of characteristics: 5, maximum of events by student: 8, maximum of students by event: variable according to the event.

Before using the instance it was necessary to correct some parameters of MTH-MMAS algorithm implemented, since for the instance of UCV the number of

timeslot that they are used are 40 and not 45 like for other problems of the UCTP. in addition to an adaptation for the evaluation of soft constraint.

The instance was executed using a number of ants = 20, evaporation factor = 0.5. Time local search 100 seconds, total time by reboots = 900 seconds, number of reboots = 10. The best solution was obtained approximately to the 600 seconds with an evaluation of (# VRS) = 0. Which implies that the algorithm generated a complete timetable feasible and with the best possible quality.

Had to the quality of the solution it can be inferred that the generated instance previously that simulated the hour load of a semester of the school of computer science engineering had a low degree of correlation between courses of different semesters, thus a high performance in the resolution was obtained of the problem.

Table 4. Show the first 20 assigned events a its respective timeslot and rooms as a form to represent the solutions.

Event	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Timeslot	2	36	36	43	41	13	31	32	18	16	19	30	27	20	0	19	19	18	7	29
Room	0	0	1	3	0	4	3	4	2	1	3	3	2	3	4	0	4	4	4	3

In addition a file is had which has associate the classes with its respective ones events, since a class can have to correspond to several events in one week. This is a form to make a more visible and usable timetable.

6 Conclusion

A formal model was given to apply Hypercube framework to solve the University course timetabling problem (UCTP) making use of Max-Min Ant System, was generated an efficient model that solves instances of this problem creating good construction graph of and expressing a good pheromone matrix.

We presented the test result made for the Max-Min Ant System doing use of Hypercube framework. We was observed traverse of the given results that this propose framework is good means of resolution of combinatorial problems and for the case of the UCTP it presented good results for instances of small and medium type. Although the results were of low quality for the instances of the Competition. it emphasizes the fact that always it generates solutions feasible and for instances of normal difficulty of good evaluation. not obtain the best results for this problem, but if it improves in contrast with the Max-Min Ant System without work frame. It was managed to present a applied instance to the school of Computer science of the UCV, for which created a solution feasible thus it clarifies the fact to a technique useful in real applications.

References

1. T. B. Cooper and J. H. Kingston. The complexity of timetable construction problems. In Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling (PATAT 1995). 1996.
2. H. M. M. ten Eikelder and R. J. Willemen. Some complexity aspects of secondary school timetabling problems. In Proceedings of the 3rd International Conference on Practice and Theory of Automated Timetabling (PATAT 2000), 2001.
3. E. K. Burke, J. P. Newall, and R. F. Weare. A memetic algorithm for university exam timetabling. In Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling (PATAT 1995), 1996.
4. B. Paechter. Course timetabling. Evonet Summer School, 2001.
5. B. Paechter, R. C. Rankin, A. Cumming, and T. C. Fogarty. Timetabling the classes of an entire university with an evolutionary algorithm. (1998).
6. M. Dorigo and G. Di Caro. The Ant Colony Optimization meta-heuristic. New Ideas in Optimization, 1999.
7. Christian Blum, Marco Dorigo, Andrea Roli, HC-ACO: The Hyper-Cube Framework for Ant Colony Optimization IEEE Transactions on Systems, Man and Cybernetics B, 34(2), 1161 - 1172, 2001.
8. Krzysztof Socha, Max-Min Ant System for International Timetabling Competition. (2003)
9. Krzysztof Socha, J. Knowles, y M. Sampels. Max-Min Ant System for the University Course Timetabling Problem.(2003)
10. http://www.dcs.napier.ac.uk/~benp
11. Thomas Stützle y H.H. Hoos, MAX-MIN Ant System. Future Genetic Computing System. 2000.
12. Christian Blum, Marco Dorigo, The Hyper-Cube Framework for Ant Colony Optimization IEEE Transactions on Systems, Man and Cybernetics B, 2004. HCF 2004.
13. http://www.or.ms.unimelb.edu.au/timetabling/ttframe.html?ttucp1.html
14. http://www.idsia.ch/Files/ttcomp2002/results.htm
15. http://www.idsia.ch/Files/ttcomp2002/IC_Problem/node3.html
16. Rossi-Doria, Blue, Knowles, Sampels, A local search for timetabling problem.(2002)