ASSIGNING CLASSES TO TEACHERS IN UNIVERSITIES VIA MATHEMATICAL MODELLING: USING BEAM SEARCH METHOD AND SIMULATION IN JAVA

Anibal Tavares de Azevedo[(a)], Andressa Fernanda S. M. Ohata[(b)], Joni A. Amorim[(c)], Per M. Gustavsson[(d)]

[(a)]University of Campinas – UNICAMP, Brazil
[(b)]State University of São Paulo – UNESP, Brazil
[(c)] Högskolan i Skövde – HiS, Sweden
[(d)]Saab Group, Sweden

[(a)]anibal.azevedo@fca.unicamp.br / atanibal@gmail.com, [(b)]andressa.matsubara@hotmail.com, [(c)]joni.amorim@his.se / joni.amorim@gmail.com, [(d)]per.m.gustavsson@saabgroup.com

ABSTRACT

In universities, before the beginning of each school year, it is held the distribution of classes among the available teachers. For such task, different constraints must be fulfilled like preventing a teacher to teach in two different places at the same time and avoid solutions in which some teachers have more class hours than others. This process, if performed manually, is time consuming and does not allow viewing other combinations of assignment of classes to teachers. In addition, it is subject to error. This study aims to develop a decision support tool for the problem of assigning teachers to classes in universities. The project includes the development of a computer program using the concepts of object orientation as a way to implement a search algorithm called Beam Search which explores the combinatorial nature of the problem. The programming language used is Java and the program has a graphical interface for insertion and manipulation of the relevant data.

Keywords: beam search, combinatorial optimization, teaching, timetable.

1. INTRODUCTION

Timetable is an event table that specifies who will participate, who will be held where and when such an event occurs. Thus a timetable should satisfy all constraints that are simultaneously involved and there should be no conflict in the schedule.

The Educational Timetabling problems can be classified in two categories: exam and course timetabling (Al-Yakoob, Sherali, and Al-Jazzaf, 2010; Carter and Laporte, 1998).

When constructing the course timetabling of a university, there is a great difficulty to relate the different variables such as students, teachers and classrooms. In special, it is necessary to consider prerequisites established by the university, individual preferences of teachers/students for certain disciplines to be taught/routed and, most often, the downtime between classes should be avoided. Added to this, there are risks of errors in the definition of the grids and these may be detected only when the classes have already begun (Al-Yakoob, Sherali, and Al-Jazzaf, 2010).

According to (Carter and Laporte, 1998) the course timetabling problem can be divided into five subproblems: teacher assignment, class-teacher timetabling, course scheduling, student scheduling and classroom assignment. The teacher assignment problem only allocates teachers to courses without using the information about the courses allocation to time periods. Course scheduling problem often uses a given allocation of teachers (Gunawan, Ng, and Poh, 2013).

This work will address the Problem of Assignment of Classes to Teachers (PACT) that combines teacher assignment and scheduling problem simultaneously within a university. A special feature had been considered in the model in order to consider teacher´s preference for classes with the same subject. That is, as general purpose, teachers must teach the classes with the least possible effort and different from the one considered in recent literature (Al-Yakoob and Sherali, 2013).

PACT is part of the set of combinatorial optimization problems (Schaerf, 1999; Willenmen, 2002) which justify the development of heuristics and meta-heuristics. Another contribution of this work is to develop and apply a Beam Search method for the PACT in a manner that the optimal solution or at least a very close solution to the optimal one is produced and constraints are all satisfied.

This paper is structured as follows. Section 2 presents the mathematical model of PACT, while section 3 presents the proposed solution method. Section 4 presents computational results and section 5 addresses conclusions and future work.

2. MATHEMATICAL MODEL

Some initial considerations are necessary for the development of the mathematical model for PACT:

- classes does not exceed their limit in terms of maximum number of students;

- the physical distance between classes is negligible, i.e., the time required for the teacher to go from one classroom to another can be neglected.

To better illustrate the developed approach, this work will employ a little number of teachers and classrooms, without loss in reliability of the results. The following data should be available:

- all classes should have the identification number of the group to which it belongs, the course initials and the name applied to the discipline, the workload and the time set in the grid;
- all teachers should have some kind of registration number, a full name and defined maximum workload.

The PACT can be formulated as follows. Let m classes to be assigned to n teachers. The cost of k teachers preferences for certain classes of subjects i $(i = 1, ..., m, k = 1, ..., n)$ is given by P_{ik}. The variable cost of similarity between the subjects of the classes, i.e., classes i and classes j $(i, j = 1, ..., m)$, is given by the variable S_{ij}. The demand of hours per week for each available class i $(i = 1, ..., m)$ is CT_i and the workload of each teacher per week k $(k = 1, ..., n)$ is CP_k.

Let:

- x_{ik} be such that the defined variable assumes the value 1 if i is assigned to the class teacher k and 0 otherwise.
- y_{ik} be the auxiliary variable defined such that it assumes the value 1 if the classes i and j are assigned to the same teacher k and 0 otherwise.

Thus, x_{ik} and y_{ik} are related by equation (1).

$$y_{ij} = \sum_{k=1}^{n} x_{ik} x_{jk} \text{ , } i, j = 1, ..., m \quad (1)$$

From these variables it is possible to derive the constraints of the problem as follows:

(a) Each class should be assigned to a single teacher.

$$\sum_{k=1}^{n} x_{ik} = 1 \text{, } i = 1, ..., m \quad (2)$$

(b) There is a maximum of hours for each teacher (workload) should be respected, i.e., the sum of the hours of classes assigned to the teacher must be less or equal to the weekly workload.

$$\sum_{i=1}^{m} CT_i x_{ik} \leq CP_k \text{ , } k = 1, ..., n \quad (3)$$

Besides these constraints related to teachers, it is necessary to check the compatibility of the allocation of classes to a given teacher k in terms of the time they occupy in its timetable. Thus, a timetable is divided in various *slot(r,c)´s* which corresponds to the interval r in the day c. Then a variable $h_i(r,c)$ is used represent if the *slot(r, c)* is allocated (it assumes the value 1) or not (value 0) to the class i. This new variable must obey the constraints given by equations (4) and (5).

(c) Respect the total number of hours in a week for each class i, as given by Equation (4).

$$\sum_{r=1}^{R} \sum_{c=1}^{C} h_i(r,c) = CT_i \text{ , } i = 1, ..., m \quad (4)$$

(d) Avoid the conflict of time between classes allocated to the same teacher. This means that a *slot(r, c)* occupied by a class i cannot be shared by another class assigned to the same teacher k. Otherwise, there will be a conflict of time between the classes. This constraint is represented by equation (5).

$$\sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{i=1}^{I} h_i(r,c) x_{ik} \leq 1 \text{, } k = 1, ..., n \quad (5)$$

The objective of this problem is to assign each class to a teacher in order to minimize the total cost of the assignments according to: individual discipline preferences and avoidance of allocation of many different disciplines to the same teacher. Then:

(a) The total cost according to individual discipline preference:

$$\sum_{i=1}^{m} \sum_{k=1}^{n} P_{ik} x_{ik}$$

(b) The total cost of similarity between different disciplines will be:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij} y_{ij}$$

The total cost is the sum **(a)** and **(b)**, and is given by Equation (6).

$$\sum_{i=1}^{m} \sum_{k=1}^{n} P_{ik} x_{ik} + \sum_{i=1}^{m} \sum_{j=1}^{m} S_{ij} y_{ij} \quad (6)$$

Thus, the problem to be solved is to minimize the objective function given by Equation (6) subject to the constraints corresponding to Equations (1)-(5). It is possible to modify the formulation of the problem in order to eliminate the variable y_{ij} from the objective function by replacing Equation (1) in Equation (6). Thus, we obtain a formulation with only variables x_{ik} and $h_i(r, c)$ as given by the mathematical model (7).

Min $$\sum_{i=1}^{m}\sum_{k=1}^{n} P_{ik}x_{ik} + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{n} S_{ij}x_{ik}x_{jk}$$

S.a.: $$\sum_{k=1}^{n} x_{ik} = 1,\ i = 1, ..., m$$

$$\sum_{i=1}^{m} CT_i x_{ik} \le CP_k,\ k = 1, ..., n \qquad (7)$$

$$\sum_{r=1}^{R}\sum_{c=1}^{C} h_i(r,c) = CT_i,\ i = 1, ...,m$$

$$\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{i=1}^{I} h_i(r,c)x_{ik} \le 1,\ k = 1, ..., n$$

$x_{ik} = 0$ or 1, $i = 1, ...,m,$, $k = 1, ...,n$
$h_i(r,c) = 0$ or 1, $i = 1, ...,m$

3. BEAM SEARCH METHOD

The problem of allocating m classes for n teachers at a university is a problem that can generate many different combinations as a result. There is no exact method able to find an optimal solution to the problem in reasonable time. The only way to guarantee an optimal solution is through an exhaustive search. In this case, it is necessary to examine the entire space of possible solutions, which is not feasible due to the large amount of solutions. For example, for a university with 50 classes and 10 teachers, the number of possible solutions is (10^{50}). If a computer can be used to examine a solution to every 1 nanosecond, it would take 3.17 x10^{33} years to examine all the results for the values mentioned. Therefore, for this problem it is convenient to use heuristics to find one feasible and good solution.

To solve the PACT, the Beam Search will be used. The Beam Search approach is a heuristic based on complete enumeration (Azevedo et al., 2012; Ribeiro and Azevedo, 2009; Sabuncuoglu and Bayiz, 1999; Ow and Morton, 1988; Valente and Alves, 2005).

Before presenting the proposed algorithm, it will be developed a simplified example of the problem in order to make it easier to see the decision tree (assignments), considering that all possible solutions. Thus, one can identify the best solution and compare it with the solution found by the developed Beam Search.

3.1. Numerical example

Let $M = 4$ be the number of classes that should be assigned to $n = 2$ teachers. Each class T_i requires a workload CT_i. Thus, the number of hours the class T_1 demand is CT_1 and CT_2 is the number of hours demanded by the class T_2, and so on. These hours (periods) will be occupied by a teacher P_k, if the lessons of the class T_i are taught by teacher P_k. Each teacher has a workload CP_k available to take classes. So, CP_1 is the workload of the teacher available P_1, CP_2 is the workload of the teacher available P_2, and so on. Each class can only be taught by a single teacher and there can be two or more classes in the same slot allocated the timetable of a teacher. Figure 1 shows the problem as a bipartite graph.

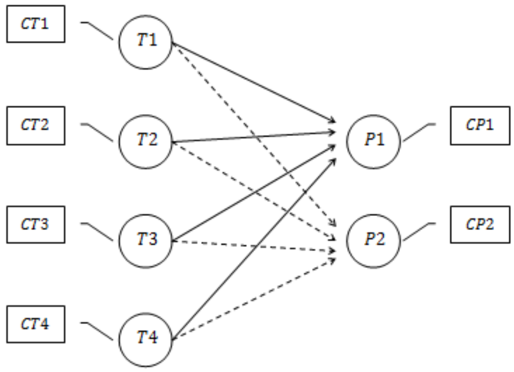

Figure 1: Representation of PACT as a graph.

As a way for the program to reach an optimal solution, it is necessary to seek information values of constraint satisfaction (maximum number of hours for each teacher and total number of hour for a class) and cost of each assignment (preferably by cost discipline and cost similarity between disciplines). Tables 1, 2, 3 and 4 show these values, respectively.

Table 1: Class demand of hours (T for "Total").

Class	Total in Week (hours)
T1	4
T2	2
T3	3
T4	3

Table 2: Teacher capacity in hours (P for "professor").

Teacher	Maximum (hours)
P1	8
P2	7

Table 3: Cost of preference for discipline.

Class\Teacher	P1	P2
T1	1	5
T2	2	3
T3	6	1
T4	6	1

Table 4: Cost of similarity between disciplines.

Class\Class	T1	T2	T3	T4
T1	0	5	12	12
T2	5	0	3	3
T3	12	3	0	0
T4	12	3	0	0

3.2. Tree solutions for the numerical example

The solution tree enumerates all possible solutions to the problem. To perform this enumeration it is necessary to establish some definitions:

- the solution tree has levels and the level *h* is assigned to the *h*-th class to a teacher;
- a complete solution is obtained only by setting all attribute classes to teachers to level *m*;
- each new attempted assignment is necessary to check if the workload of the teacher allows the assignment of the workload of the *h*-th class, considering all assignments *h-1* previous assignments;
- in each level *h*, when performing the assignment of *h*-th class to a teacher, it is essential that the costs are accounted preferably by discipline and by similarities between disciplines, considering all previous assignments *h-1*.

Figure 2 shows the tree with all possible solutions to the problem. In this figure, the numbers of teachers are represented within the nodes and the numbers related with classes are represented alongside the letter *T*, for example, T_1 represents the class 1. Each node of the tree represents an assignment, for example, the first level node N_1 symbolizes the assignment to the teacher of the class T_1 P_1. After the assignment of the root nodes in the case, P_1 and P_2, other duties are performed for the following nodes, and each node should be given the cost value considering the previous assignments.

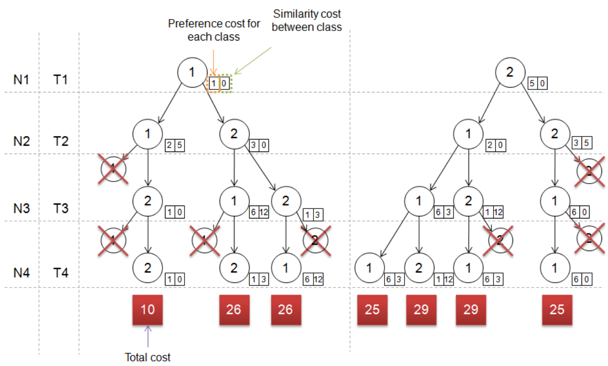

Figure 2: Possible solutions to the roots P_1 (left tree) and P_2 (right tree).

In Figure 2, at each node, the values of the cost of individual preference and similarity, in according to the allocation made to the node, are represented. Nodes marked with a cross (X) provide a solution infeasible and therefore are eliminated from the process. This is because after a few assignments, some teachers reach their maximum workload. Thus, it is understood that although the total number of solutions to be equal to $n^m = 2^4 = 16$, only 7 of these solutions are feasible.

It is also important to understand that assigning a class to a teacher held each level provides only a cost of partial similarity between disciplines. The total cost of similarity will only be achieved when all classes are assigned to a teacher. For example, at level 1 the assignment of class T_1 is done to each teacher (roots), but still cannot account for the cost of similarity between disciplines. The reason is that they do not know the assignment of the remaining classes.

At level 2, the allocation of T_2 class to each teacher is made, and it may already be counting the cost of similarity between subjects according to assignments made for each teacher. Thus, in the case of both classes T_1 and T_2 are assigned to the same teacher, and if the subjects are different, the cost will be some non-zero value. This represents the cost of preparing disciplines with different subject at the time. So, if the subject taught is the same or classes T_1 and T_2 are assigned to different teachers, the additional cost of similarity will be zero.

Similarly, at level 3, the cost of similarity will be given between classes T_1 and T_3 and between T_2 and T_3 groups. If the three groups of subjects are different, the cost of similarity between the classes should be added.

Thus, a solution will only be complete when you reach the last level of the tree (the last node) with all assignments completed and accounting costs computed.

3.3. Beam Search Method

The algorithm does not generate all the nodes of the tree as shown in the previous example. Therefore, the enumeration of all feasible solutions is not made while avoiding the exponential growth of the tree. Accordingly, we have created a few rules for the generation of nodes. These rules are also intended to prevent the creation of infeasible solutions.

First, it should be borne in mind that each node should be set up store the following information:

- identification of the teacher;
- identification of the class;
- sum of the costs of similarity between subjects from the root node;
- partial cost of the node, or adding cost, preferably in the discipline and the cost of similarity between subjects given assignments made to that node.

In addition, to create a node, it should be checked to load the remaining teacher. A node will only be created at level *h* if the workload CT_i class T_i is less than or equal to the workload $CP_k^{(h)}$ remaining teacher P_k on level *h*. So if $CP_k^{(h)} < CT_i$, all branches that originate from this node will not exist, because the class T_i cannot be attributed to the teacher P_k.

To select the nodes created, only the β nodes with smaller solutions, as pointed by a greedy algorithm, will remain as a part of the tree. From β we will create the bundles, and each development level of the tree, only still part of each beam node to generate the lowest cost solution and therefore will remain β nodes per level after the overall evaluation process. Thus, at the end of the process only β solutions will remain in the tree.

Briefly the algorithm should follow the steps.

i. Initially the algorithm is at level zero solution as no assignment has been made so far, then level = 0.

ii. Then it is level = level + 1 and nodes are created at this level, one for each teacher, taking into account their workload. For each node are calculated the cost preference for matter and similarity between subjects, and partial cost given by the sum of the other two most part calculated the cost to the previous level.

Observation: If the number of nodes created in step ii is less than or equal to the search width β back to step ii, and all nodes are expanded at the next level.

iii. For each node created in step ii, it is:

- Store the class and teacher identification;
- Store the partial node cost;
- Store the remaining capacity for each teacher, i.e., once a class is assign to a teacher his/her corresponding capacity is reduced in a number equal to total number of class hours.

iv. If the level set in step ii is less than the number of classes, applies the greedy algorithm, considering the costs and workloads partial remaining until then. The details of the greedy algorithm is as following:

(a) For the actual nodes level (parent nodes) makes it a tree structure, i.e., it creates child nodes whose tasks are feasible considering the remaining workloads of each teacher and calculate the costs of preference and similarity;
(b) For each parent node, order up the child nodes as the sum of costs and it selects the lowest cost node;
(c) Store up for the resulting nodes in (b), the partial cost and workload of the remaining teachers;
(d) If the level in question is less than the number of classes, it returns to the step (a), if not, whether the β-select solutions that generated the lowest cost. If the development of the tree is in bundles, you must select a node beam;
(e) Among the nodes created in step ii, pick up the β we generated the greedy lower-cost solutions obtained in step (c), and the others are discarded.

v. If the level set in step ii is equal to the number of classes the algorithm terminates, since the tree has reached the last level. If not, return to step ii.

In the proposed algorithm, the value of the total cost of a branch of the tree obtained from the greedy solution, acts as an upper bound (cutoff value) to generate or not the other nodes of the tree in step ii. This upper bound must be updated as a branch is found with value less cost. Thus, two criteria have been cut or not to generate a node of the tree, the remaining workload of teachers and the lowest total cost solution obtained by the greedy.

3.4. Applying the Beam Search Method to the numerical example

A detailed resolution to the example of Section 3.1 was developed, for a better understanding of the algorithm shown in Section 3.3

Recalling that m = 4 is the number of classes that should be assigned to n = 2 teachers. Furthermore, one must consider β = 2, and Tables 1, 2, 3 and 4 for query workload of classes, workload of teachers, teachers' costs preferred by subjects and costs similarity between disciplines, respectively.

In figure 4.3 we have set up the first level as the workloads of tables 1 and 2. Thus, for example, the level 1 node N_1 are: load the class and T_1 = 4 teacher load P_1 = 8 as $CP_1 > CT_1$ is possible to perform the assignment. Preferably costs (value of square left) and similarity (value of the square to the right) were obtained according query to nodes in Tables 3 and 4: the cost of the teacher's preference for the subject class P_1 is 1 and the value T_1 similarity is zero because it is the first assignment. As the number of nodes obtained is equal to the search width β = 2, it is not necessary to apply the greedy algorithm. Develop, then the nodes N_2 and feasible level are calculated costs. Up to this point were carried out only steps ii and iii.

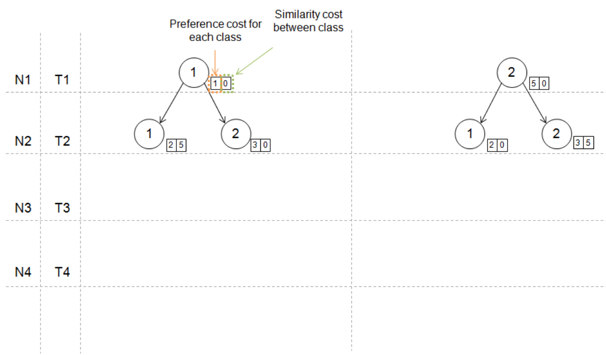

Figure 3: Development levels 1 and 2 corresponding to steps ii and iii of the Beam Search.

Figure 4 is made from the tree structure created in the standard parent nodes N_2. As can be seen in the figure, two nodes were eliminated due to the workload of the class that exceeded the remaining workload of teachers.

These nodes are marked with an X and drawn slightly above the rest. The cost preferred and similarity was calculated, obtaining thus the partial cost of each node.

As the greedy algorithm allows only one child node for each parent node, only one child node was maintained for each parent node and the others were eliminated.

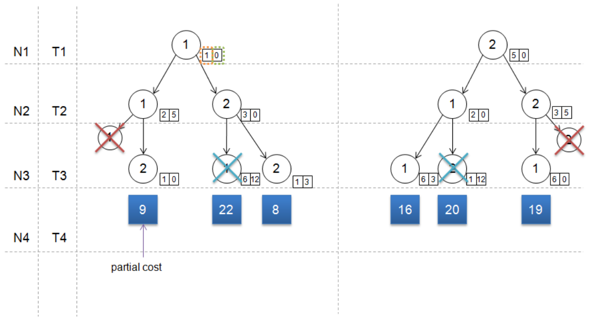

Figure 4: Tree structure of nodes of level 2 corresponding to the greedy algorithm of step iv.

Likewise, in Figure 5 was made from the tree structure of nodes that are left in the previous layer, calculate the costs and partial nodes are eliminated unnecessary.

As reached the top level of the tree, that is, level = the number of divisions = 4, β = 2 only the lowest total cost solutions remain part of the tree.

The two solutions are marked with an arrow below the total cost.

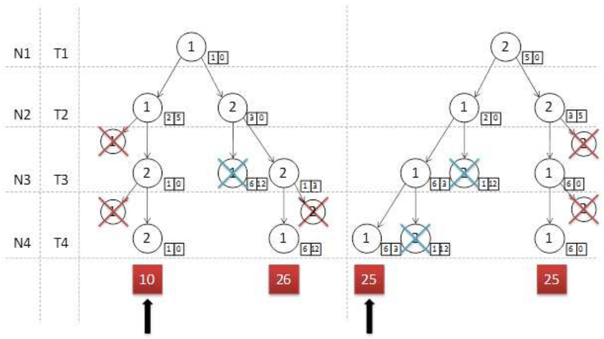

Figure 5: Tree structure of nodes of level 2 corresponding to the greedy algorithm of step iv to the sub step d.

Figures 4 and 5 correspond to step iv by the greedy algorithm of sub step d.

In Figure 6 are shown the level of the selected nodes N_2 for β = 2 greedy best solutions obtained in Figure 5, corresponding to step subsection and iv. Furthermore, we have been developed in the level N_3 and the costs calculated (steps ii and iii). In Figure 7, the greedy algorithm is applied again now to level nodes N_3 and selected the best solutions β = 2 (step iv to sub step d).

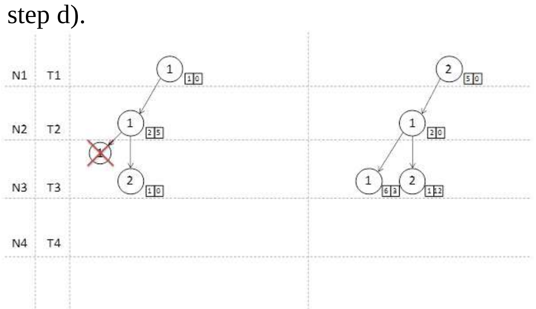

Figure 6: Selecting the nodes of level 2 (step iv sub step e) and development of nodes of level 3 (steps ii and iii).

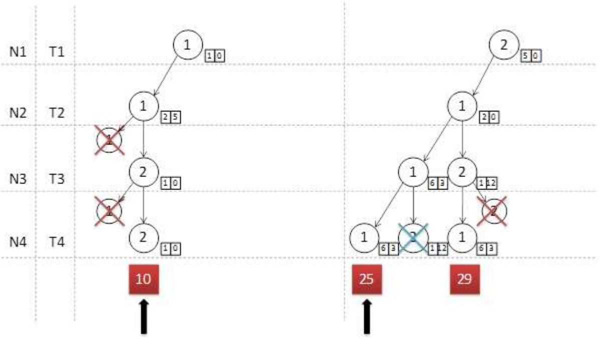

Figure 7: Arborescence level 3 (step iv to sub step d).

Figure 8 shows the selected nodes to continue the level N_3, chosen by the two best solutions greedy (step iv subsection e). Develop, too, the nodes N_4, and its cost level (steps ii and iii). Note that it is not needed to use the greedy algorithm, because the tree has reached its final level. Therefore, the calculated costs are the total costs of solutions, and among them will be selected only the best solutions β = 2, one for each beam.

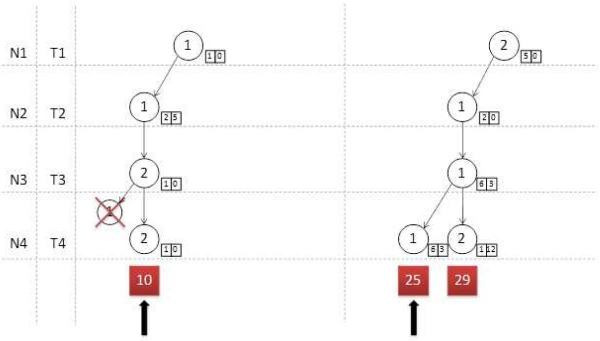

Figure 8: Selecting the nodes of level 3 (step iv to sub step e) and development of the nodes of level 4 (steps ii and iii).

Finally, Figure 9 shows the solution to the allocation problem completely, since the algorithm has reached the end N_4 which is the last level of the tree. Thus, among the best solutions β = 2, only one will be considered the final solution, which has a lower total cost. So, the final solution will be: classes T_1 and T_2 assigned to the teacher P_1 and classes T_3 and T_4 assigned to the teacher P_2.

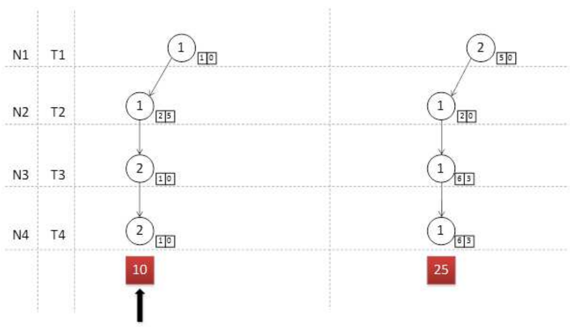

Figure 9: Final solution of the problem.

4. COMPUTATIONAL RESULTS

From the algorithm developed we present a computational tool for the PACT, using a set of classes implemented in Java. The Java language was chosen due to the ease in developing GUIs (JAVA, 2012). And for the development of the program it was used the IDE JCreator (http://www.jcreator.com/).

To archive data from teachers and class, the costs of preference for similarity between subjects and disciplines, and for the solution of the problem, it was used a set of “.txt” files. A machine-readable version of a MySQL database also began to be developed, but the time was spent to make changes to the database for use in small tests was too long. Moreover, as the PACT is only a part of a large project, how should be configured the database and its tables depend on previous steps of the project. Thus, it was decided to work with the files in “.txt”, eliminating the configuration of a database and user authentication, and making quick and convenient transportation of the program and its data.

The computer used to develop the program and its testing was a Notebook, processed with Intel Core2 Duo T6500 2.10GHz, with 3GB of RAM. The operating system used was Windows 7 32-bit.

Some screens of the graphical interface developed for the developed program are provided in Figures 9 and 10.

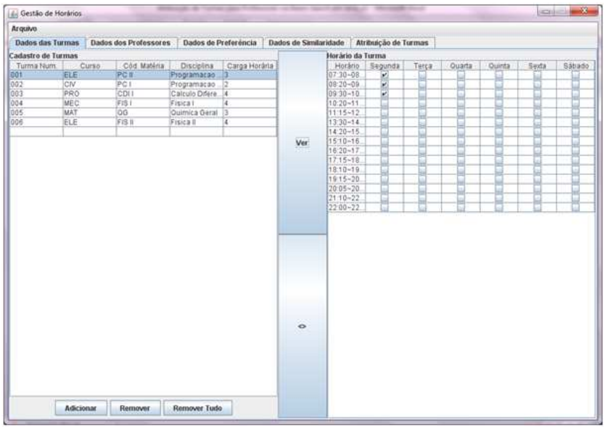

Figure 9: Graphical interface for data entry.

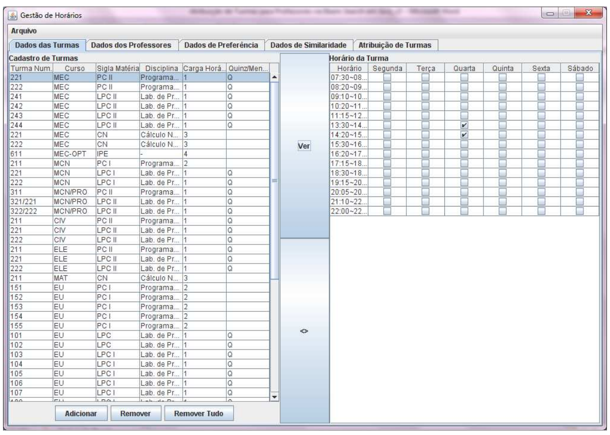

Figure 10: Results obtained for real data.

The program showed satisfactory performance for the test with real data where it is necessary to perform the allocation of 63 classes to 11 teachers. The detailed results of the program and its comparison with the manual allocation are provided in Appendix.

5. CONCLUSIONS AND FUTURE WORKS

The PACT is a problem of combinatorial nature that is part of a more complex problem of Course Timetabling is made where the efficient management of educational resources. PACT aims to make the distribution of teachers between classes at a university in order to respect the weekly class, restrictions workload of teachers and the preferences of the same subjects to be taught, assign subjects to a similar same teacher, do not assign a class to more than one teacher and not allow a teacher to be allocated to different classes in the same time. A special feature had been considered in the model in order to consider teacher´s preference for classes with the same subject in a manner that is not considered in recent literature.

To solve the problem we developed a totally new computational tool that is a heuristic for the automatic termination of the combinatorial problem. The tool contributes mainly by the speed and efficiency in decision making for the allocation of teachers, and prevents some teachers being overloaded. The developed algorithm is a heuristic based on complete enumeration technique through the search tree and the greedy algorithm.

A graphical interface was developed to facilitate data capture to problem resolution. It is possible, through the interface, create, open, save and edit files containing data of teachers, classes of data, preference values for subjects and similarity values between disciplines. Another facility that provides graphical interface is the possibility to simulate various assignments with various widths search soon. This facility brings a very big advantage is that the visualization and comparison of different configurations of the timetable, including performing rapid changes in the values of preference and similarity to determine the effect on assignments. At the end of the assignment, the program also lets you make manual adjustments in the result, if a teacher wants to make a simple change that does not result in shocks or extrapolation of the time course load.

This GUI can be expanded in the future to integrate data capture and resolution of the remaining issues in managing educational resources related to PACT. Furthermore, one can add other features such as printing files in formats ideal for viewing and distribution among teachers, and also improve the logical assignment to get results closer to the manual distribution. The development of the interface in additional languages is also a relevant future work to be accomplished.

ACKNOWLEDGMENTS
The authors would like to thank the following organizations for their support: Saab Group (www.saabgroup.com), University of Skövde (www.his.se), University of São Paulo (www.usp.br), University of Campinas (www.unicamp.br) and National Counsel of Technological and Scientific Development (www.cnpq.br).

APPENDIX
The following are the results obtained by manual and program.

Table 5: Manually obtained results.

Results without the use of computer simulations	
Discipline	**CHD**
Teacher 1	**15/9h**
PC II - MEC(222)	2*
LPC II - MEC(243)	2*
LPC II - MEC(244)	2*
LPC II - MCN(321)/PRO(221)	2*
LPC II - MCN(322)/PRO(222)	2*
LPC II - CIV(222)	2*
PC II - MCN(311)/PRO(222)	2*
PC II -CIV (211)	2*
LPC II - CIV (221)	2*
Teacher 2	**0/8h**
PC I - EU(155)	2
PC I - ESP(131)	2
LPC I - EU(109)	2*
LPC I - EU(110)	2*
LPC I - EU(161)	2*
LPC I - EU(162)	2*
Teacher 3	**14/10h**
CDI	2
CDI	2
PC I - MCN (211)	2
CDI	2
LPC I - MCN(221)	2*
LPC I -MCN(222)	2*
Teacher 4	**3/3h**
PC II - ELE(211)	2*
LPC II - ELE(221)	2*
LPC II - ELE(222)	2*
Teacher 5	**13/9h**
MAC - ESP(111)	2
PC I - ESP(132)	2
PC II - MEC(221)	2*
LPC II - MEC(241)	2*
LPC II - MEC(242)	2*
LPC I - ESP(163)	2*
LPC I - ESP(164)	2*
Teacher 6	**13/13h**
CN - MEC(222)	3
CN - MAT(211)	3
IPE - MEC-OPT(611)	4
CN - MEC (221)	3
Teacher 7	**0/8h**
PC I - EU(151)	2
PC I - EU(152)	2
LPC - EU(101)	2*
LPC I - EU(104)	2*
LPC - EU(102)	2*
LPC I - EU(103)	2*
Teacher 8	**12/12h**
PC I - EU(154)	2
LPC I - EU(108)	2*
CCN - ESP(211)	2
LPC I - EU(107)	2*
CCN - ESP(211)	2
PC I - ESP(133)	2
LPC I - ESP(165)	2*
LPC I - ESP(166)	2*
Teacher 9	**7,5/4h**
CN - LMN(211)	2
LCN - LMN(211)	2
Teacher 10	**3/3h**
PC - LMN(111)	2
LPC - LMN(111)	2*
Teacher 11	**6/4h**
PC I - EU(153)	2
LPC I - EU(105)	2*
LPC I - EU(106)	2*

Table 6: Results provided by the software in Java.

Computer based simulations	
Discipline	**CHD**
Teacher 1	**15/15h**
PC II - MEC(222)	2*
LPC II - MEC(241)	2*
LPC II - MEC(242)	2*
LPC II - MEC(244)	2*
PC II - MCN(311)/PRO(222)	2*
LPC II - MCN(321)/PRO(221)	2*
LPC II - MCN(322)/PRO(222)	2*
PC II -CIV (211)	2*
LPC II - CIV(222)	2*
PC II - ELE(211)	2*
LPC II - ELE(222)	2*
PC I - EU(151)	2
PC I - EU(152)	2
Teacher 2	**0/0h**
Teacher 3	**14/13h**
CN - MEC (221)	3

PC I - MCN (211)	2
PC - LMN(111)	2
CDI	2
CDI	2
CDI	2
Teacher 4	**3/3h**
LPC II - MEC(243)	2*
LPC II - CIV (221)	2*
LPC II - ELE(221)	2*
Teacher 5	**13/13h**
LPC I - MCN(221)	2*
LPC I -MCN(222)	2*
PC I - EU(153)	2
PC I - EU(154)	2
LPC I - EU(103)	2*
PC I - ESP(131)	2
PC I - ESP(133)	2
LPC I - ESP(163)	2*
LPC I - ESP(164)	2*
Teacher 6	**13/12h**
PC II - MEC(221)	2*
CN - MEC(222)	3
IPE - MEC-OPT(611)	4
CN - LMN(211)	2
LCN - LMN(211)	2
Teacher 7	**0/0h**
Teacher 8	**12/11h**
LPC I - EU(104)	2*
LPC I - EU(105)	2*
LPC I - EU(106)	2*
LPC I - EU(107)	2*
LPC I - EU(108)	2*
LPC I - EU(109)	2*
LPC I - EU(110)	2*
LPC I - EU(161)	2*
LPC I - EU(162)	2*
LPC I - ESP(165)	2*
LPC I - ESP(166)	2*
Teacher 9	**7,5/7h**
CN - MAT(211)	3
CCN - ESP(211)	2
CCN - ESP(211)	2
Teacher 10	**3/3h**
LPC - EU(101)	2*
LPC - EU(102)	2*
LPC - LMN(111)	2*
Teacher 11	**6/6h**
PC I - EU(155)	2
MAC - ESP(111)	2
PC I - ESP(132)	2

REFERENCES

Al-Yakoob, S.M., Sherali, H.D., A column generation mathematical programming approach for a class-faculty assignment problem with preferences, *to appear in Computational Management Science*, pp. 1-22, 2013.

Al-Yakoob, S. M. , Sherali, H.D., Al-Jazzaf, M., A mixed-integer mathematical modeling approach to exam timetabling, *Computational Management Science*, Vol. 7(1), pp. 19-46, 2010.

Azevedo, A.T., Ribeiro, C.M., Sena, G.J.D., Chaves, A.A., Neto, L.L.S., Moretti, A.C.: Solving the 3D Container Ship Loading Planning Problem by Representation by Rules and Beam Search. ;In ICORES, pp.132-141, 2012.

Deitel, H. M.; Deitel, P. J. Java: How to Programm. 6. ed. Bookman, 2006.

Della Croce, F.; T'kindt, V., A Recovering Beam Search Algorithm for the One-Machine Dynamic Total Completion Time Scheduling Problem, *Journal of the Operational Research Society*, vol 54, 2002, pp. 1275-1280.

Gunawan, A., Ng, K.M., Poh, K.L., Solving the Teacher Assignment-Course Scheduling Problem by a Hybrid Algorithm, CiteSeer. Available in: < http://130.203.133.150/viewdoc/download?doi=10.1.1.193.3646&rep=rep1&type=pdf>. Access: 21 jun. 2013.

Java™ , Sun Microsystems, Platform, Standard Edition 6 API Specification. Available in: <http://java.sun.com/javase/6/docs/api/>. Access: 20 jan. 2012.

Michael, W.C., Laporte, G., Recent Developments in Practical Course Timetabling, *Selected papers from the Second International Conference on Practice and Theory of Automated Timetabling II*, Springer-Verlag, London, UK, pp. 3-19, 1998.

Sabuncuoglu, I.; Bayiz, M., Job Shop Scheduling with Beam Search, *European Journal of Operational Research*, vol. 118, 1999, pp. 390-412.

Ow, P.S; Morton T.E., Filtered Beam Search in Scheduling, *International Journal of Production Research*, vol. 26, 1988, pp. 35-62.

Ribeiro, C.M., Azevedo, A.T., Teixeira, R.F.,Problem of assignment cells to switches in a cellular mobile network via Beam Search Method, WSEAS Transactions on Communications, Vol. 9(1): pp.11-21, 2009.

Schaerf, A. A Survey of Automated Timetabling. Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, 1999. Available in: <http://www. diegm.uniud.it/satt/papers/Scha99.pdf>. Access: 16 jun. 2011

Valente, J. M. S; Alves, R. A. F. S., Filtered and Recovering Beam Search Algorithm for the Early/Tardy Scheduling Problem with No Idle Time, *Computers & Industrial Engineering*, vol. 48, 2005, pp. 363-375.

Willenmen, R. J. School timetable construction:

algorithms and complexity. Technische Universiteit Eindhoven, 2002. Available in: < http://alexandria.tue.nl/extra2/200211248.pdf >. Access: 10 jul.2011.

AUTHORS BIOGRAPHY

Anibal Tavares de Azevedo. PhD in Engineering. Dr. Azevedo teaches at Universidade Estadual de Campinas – UNICAMP (http://www.unicamp.br/), in Brazil. Previously, he worked as a researcher and as a teacher at Universidade Estadual Paulista – UNESP (http://www.unesp.br/), Brazil. Dr. Azevedo graduated in Applied and Computational Mathematics from UNICAMP (1999), holds a Master's degree in Electrical Engineering from UNICAMP (2002) and received his Ph.D. degree in Electrical Engineering from UNICAMP (2006). He has experience in software development and in mathematical modeling for Production Engineering and Planning, for Scheduling of Power System Operation and for Education. His research has an emphasis on Linear Programming, Nonlinear Programming and Mixed Dynamics in the following topics: interior point methods, planning and production control of manufacturing, flows in networks, linear programming, graph generalized combinatorial optimization, allocation of cells to cellular centrals, loading and unloading of containers on ships 2D and 3D, genetic algorithms, beam search and simulated annealing. < http://lattes.cnpq.br/9760457138748737 >.

Andressa Fernanda Saemi Matsubara Ohata. Engineer. Previously worked as a researcher at Universidade Estadual Paulista – UNESP (http://www.unesp.br/), Brazil.

Joni A. Amorim. PhD in Engineering. Postdoctoral Fellow at the University of Skövde, or Högskolan i Skövde – HiS (http://www.his.se/english/), in Sweden, in collaboration with SAAB Training and Simulation (http://www.saabgroup.com/en/training-and-simulation/). The University of Skövde offers first-class programs and competitive research, which attracts research scientists and students from all over the world. The University of Skövde is one of the most specialized universities in Sweden and its research is focused on the development and use of advanced information technology systems and models. Dr. Amorim previously worked as a researcher and as a teacher at Universidade Estadual de Campinas – UNICAMP (http://www.unicamp.br/), in Brazil. Dr. Amorim collaborates with researchers at UNICAMP, a university with more than 3,600 original scientific publications published in 2009, 78% of which in journals indexed in the ISI/Web of Science. Dr. Amorim collaborates with researchers at Universidade de São Paulo – USP (http://www.usp.br/), the major institution of higher learning and research in Brazil. His research has an emphasis on multimedia production management, project portfolio management, distance education and training based on serious games and simulations. < http://lattes.cnpq.br/3278489088705449 >.

Per M. Gustavsson. PhD in Computer Science. Dr. Gustavsson works as a Research Scientist at Saab Group (http://www.saabgroup.com/). Saab serves the global market with products, services and solutions ranging from military defence to civil security. Dr. Gustavsson also works at the Swedish National Defence College – SNDC (http://www.fhs.se/en/), in Sweden. At SNDC, research is carried out in diverse, but inter-related subject areas and subsequently disseminated to other interested sectors of society both nationally and internationally; the College trains and educates military and civilian personnel in leading positions, both nationally and internationally as part of the contribution to the management of crisis situations and security issues. < http://se.linkedin.com/in/pergu >.