Expert Systems with Applications 36 (2009) 12349–12356

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem

Cagdas Hakan Aladag [a,*], Gulsum Hocaoglu [a], Murat Alper Basaran [b]

[a] Department of Statistics, Hacettepe University, Ankara 06800, Turkey
[b] Department of Mathematics, Nigde University, Nigde 51000, Turkey

ARTICLE INFO

Keywords:
ANOVA
Combinatorial optimization
Course timetabling problem
Neighborhood structure
Tabu search
Type of move

ABSTRACT

The course timetabling problem must be solved by the departments of Universities at the beginning of every semester. It is a though problem which requires department to use humans and computers in order to find a proper course timetable. One of the most mentioned difficult nature of the problem is context dependent which changes even from departments to departments. Different heuristic approaches have been proposed in order to solve this kind of problem in the literature. One of the efficient solution methods for this problem is tabu search. Different neighborhood structures based on different types of move have been defined in studies using tabu search. In this paper, the effects of moves called simple and swap on the operation of tabu search are examined based on defined neighborhood structures. Also, two new neighborhood structures are proposed by using the moves called simple and swap. The fall semester of course timetabling problem of the Department of Statistics at Hacettepe University is solved utilizing four neighborhood structures and the comparison of the results obtained from these structures is given.

1. Introduction

Assigning instructor-course-room combinations into specific time periods is a course timetabling problem which occurs in the Universities. The objective in a classical course timetabling problem is to reduce the number of conflicts, which occur when courses involve common students, common teachers or require the same classrooms (Aladag & Hocaoglu, 2007a). For large institution such as Universities, the more constraints are added to the problem, the more difficult the solution of the problem is obtained.

Solving a course timetabling problem is a very difficult task because of the size of the problem and the nature of the changing structure of the problem. The solution techniques range from graph coloring to complex metaheuristic algorithms, including linear programming formulations and heuristics tailored to the specific problem at hand. MirHassani (2006) has developed an integer programming approach, Al-Yakoob and Sherali (2006, 2007) have used mixed-integer programming model, Boland, Hughes, Merlot, and Stuckey (2008) has used integer linear programming. Gueret, Jussien, Boizumault, and Prins (1995) has developed a different approach, constraint logic programming. The more efficient procedures which have appeared in recent years are based on metaheuristics. Dowsland (1990) and Elmohamed et al. (1997) have used simulated annealing, Burke, Newall, and Weare (1996), Corne and Ross (1996) and Paechter et al. (1996) have developed procedures based on variants of genetic algorithms and Alvarez, Martin, and Tarmarit (1996) and Hertz (1991, 1992) and have used tabu search techniques. Chiarandini, Birattari, Socha, and Rossi-Doria (2006) has introduced a hybrid metaheuristic algorithm which combines various construction heuristics, tabu search, variable neighborhood descent and simulated annealing. Burke, Petrovic, and Qu (2006) has used a case-based heuristic selection approach. Head and Shaban (2007) have developed a method based on heuristic functions. Pongcharoen, Promtet, Yenradee, and Hicks (2008) has described the stochastic optimization timetabling tool which includes genetic algorithms, simulated annealing and random search methods. Beligiannis, Moschopoulos, Kaperonis, and Likothanassis (2008) has designed an adaptive algorithm based on evolutionary computation techniques. Causmaecker (2009) has developed a decomposed metaheuristic approach.

One of the most efficient algorithms for the solution of the problem is tabu search algorithm. Tabu search method has proved its efficiency in the solution of the combinatorial optimization problems (1997). A tabu search algorithm consists of the usage of advanced strategies and common components such as tabu list, various memories, neighborhood structures, and so on (Aladag, 2004). One of the most important factors which affect the efficiency of the algorithm is a defined neighborhood structure pertained to the nature of the problem (Aladag & Hocaoglu, 2007b). In this paper, we examine the four different neighborhood structures based on types of move such as simple, swap. Two of the four neighborhood structures used in this study were used by Aladag and Hocaoglu (2007a) and Alvarez et al. (2002). The two neighborhood

* Corresponding author. Tel.: +90 312 2977900.
E-mail address: chaladag@gmail.com (C.H. Aladag).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.04.051

structures differ in terms of the used moves which are simple and swap. Other used neighborhood structures proposed in this paper compose of combining the simple and swap moves. By doing so, we aimed at constituting a diversification effect in the used tabu search algorithm. Based on the usage of four neighborhood structures, the fall semester of course time tabling problem of the Department of Statistics at Hacettepe University is solved utilizing a tabu search algorithm introduced by Aladag and Hocaoglu (2007a). According to obtained results, multiple comparisons among all neighborhood structures are statistically conducted.

Section 2 includes the definition and the formulation of the solved problem. In Section 3, the flow chart of the used tabu search algorithm and the definitions of the used neighborhood structures based on simple and swap moves are given. The proposed neighborhood structures are introduced in Section 4. In the implementation section, the defined problem is solved based on four neighborhood structures and the results are given. The last section concludes the study.

2. The solved course timetabling problem

The definition of the solved problem given below was introduced by Aladag and Hocaoglu (2007a).

2.1. Objectives and constraints of the problem

First of all, it is necessary to explain some basic concepts in course timetabling problem. Let 'Probability' course has two section courses, two hours theoretical and 2 h practical per week. Its sections are denoted by 'Probability 01' and 'Probability 02' and its lessons are denoted by 'Probability 01 Theory', 'Probability 01 Practice', 'Probability 02 Theory' and 'Probability 02 Practice'. Class is a set of courses which is taken by a group of students. Lesson which has 1 h can be assigned to a single period.

In a course timetabling problem, generally, constraints are considered in two types. One of them is called hard constraints. Every acceptable timetable must satisfy these constraints. For our problem, these constraints are:

(H1) Every lesson has to be assigned to a period or a number of periods, depending on the lesson's length.
(H2) No teacher can give two simultaneous lessons.
(H3) All lessons of the same section cannot be assigned to the same day.
(H4) No room can be assigned to two simultaneous lessons.
(H5) The room assigned to a given lesson must be of the required type.
(H6) All the pre-assignments and forbidden periods for classes, teachers and rooms must be respected.
(H7) Lessons of sections of the same class cannot conflict.

There are some conditions that are considered helpful but not essential in a good timetable. The more these conditions are satisfied, the better the timetable will be. They are called soft constraints and therefore they will have weights in the objective function. For our problem these constraints are:

(S1) Class timetables should be as compact as possible, eliminating idle times for students.
(S2) In class timetables, students should not have lessons hours more than a specified quantity.
(S3) In class timetables, students should not have a day with a single lesson as possible.
(S4) Teachers' non-desired periods should be avoided.
(S5) For rooms, the objective is adjusting their capacity to the number of students assigned to them.

2.2. Problem formulation

First of all, following symbol should be denoted:

A	Set of courses
$Y=\{Y_{11}, Y_{12}, \ldots, Y_{jk}, \ldots\}$	Set of section lessons
Y_{jk}	Set of section k of course j
L	Set of lessons
C	Set of classes
$C_s = \{C_{11}, C_{12}, \ldots, C_{ij}, \ldots\}$	Set of classes with section
C_{ij}	Set of section j of class i
$\mathscr{T}$	Set of teachers
LT_k	Set of lessons of teacher $\mathscr{T}_k$
P	Set of periods
P_l	Set of periods of day l
D	Set of days
d_i	Duration of lesson i
R	Set of rooms
R_r	Set of rooms of type$r$
LR_r	Lessons requiring rooms of type r
TR	Different types of rooms
m_{rt}	Number of rooms of type r
F	Set of pre-assigned lessons
p_i	Pre-assigned period of lesson i
U_i	Set of forbidden periods of lesson i

The variables are defined as follows:

$$x_{ita} = \begin{cases} 1, & \textit{if lesson i starts at period t in room a} \\ 0, & \textit{otherwise} \end{cases}$$

In the objection function, for constraints (S1), (S2) and (S3) which are for the students, functions $f_{s1}(x)$,$f_{s2}(x)$ and $f_{s3}(x)$; for constraint (S4) which is for the teachers, function $f_t(x)$; for constraint (S5) which is for the rooms, function $f_r(x)$ are defined. Each objective appears with its corresponding weight w: w_{s1}, w_{s2} and w_{s3} correspond to students, w_t correspond to teachers, w_r correspond to rooms. In the used computer program, the weights can be determined by a user. In this way, a user can determine which constraint has how much importance.

The problem is:

$$\text{Min} \quad f(x) = w_{s1}f_{s1}(x) + w_{s2}f_{s2}(x) + w_{s3}f_{s3}(x) + w_t f_t(x) + w_r f_r(x) \tag{1}$$

$$\text{subject to} \quad \forall i \in L, \quad \sum_{a \in R} \sum_{t \in P} x_{ita} = 1 \tag{2}$$

$$\forall t \in P, \quad \forall k \in \mathscr{T}, \quad \sum_{a \in R} \sum_{i \in LT_k} \sum_{\tau = t - d_i + 1}^{t} x_{i\tau a} \leqslant 1 \tag{3}$$

$$\forall Y_{jk} \in Y, \quad \forall l \in D, \quad \sum_{a \in R} \sum_{t \in P_l} \sum_{i \in Y_{jk}} x_a \leqslant 1 \tag{4}$$

$$\forall a \in R, \quad \forall t \in P, \quad \sum_{i \in L} \sum_{\tau = t - d_i + 1}^{t} x_{i\tau a} \leqslant 1 \tag{5}$$

$$\forall t \in P, \quad \forall r \in TR, \quad \sum_{a \in R_r} \sum_{i \in LR_r} \sum_{\tau = t - d_i + 1}^{t} x_{i\tau a} \leqslant m_{rt} \tag{6}$$

$$\forall i \in F, \quad \sum_{a \in R} x_{ip_i a} = 1 \tag{7}$$

$$\forall i \in L, \quad \sum_{a \in R} \sum_{t \in U_i} \sum_{\tau = t - d_i + 1}^{t} x_{i\tau a} = 0 \tag{8}$$

$$\forall C_{ij} \in C_s, \quad \forall t \in P, \quad \sum_{a \in R} \sum_{i \in C_j} \sum_{\tau = t - d_i + 1}^{t} x_{i\tau a} \leqslant 1 \tag{9}$$