

FORMULATION SPACE SEARCH APPROACH FOR THE TEACHER/CLASS TIMETABLING PROBLEM

Yuri KOCHETOV¹, Polina KONONOVA², Mikhail PASCENKO¹

¹ *Sobolev Institute of Mathematics, Russia*

² *Novosibirsk State University, Russia*

Received: / Accepted:

Abstract: We consider the well known NP-hard teacher/class timetabling problem. Variable neighborhood search and tabu search heuristics are developed based on idea of the Formulation Space Search approach. Two types of solution representation are used in the heuristics. For each representation we consider two families of neighborhoods. The first family uses swapping of time periods for teacher (class) timetable. The second family bases on the idea of large Kernighan-Lin neighborhoods. Computation results for difficult random test instances show high efficiency of the proposed approach.

Keywords: Timetable design, metaheuristics, local search, FSS approach.

1. INTRODUCTION

In this paper we consider the teacher/class timetabling problem for the Specialized Physical-Mathematical School which is a part of the Novosibirsk State University. All students in this school are gathered into classes. Each class has its own list of subjects with prescribed teachers. The teachers in this school are, in fact, the scientific researchers. They are working at the Novosibirsk State University and at the Physical and Mathematical Institutes of the Novosibirsk Scientific Center. So, the teachers have a lot of restrictions for teaching time. Each teacher has a list of available time periods. Moreover, some time periods in this list actually are inconvenient for him (her). The managers of the school try to create the most suitable timetable for the teachers, i.e. a timetable without time gaps and with classes in the most appropriate time for the teachers.

In this paper we describe the mathematical model for this timetabling problem and develop local search heuristics based on idea of the Formulation Space Search approach (FSS) [4], [7]. In Section 2 we present an exact mathematical formulation for

this multi criteria optimization problem and reduce it to the classical timetable problem by a penalty function [9]. In Section 3 we introduce two types of solution representations and show how to transform one of them to the other. In Section 4 a polynomial time heuristic is developed in order to create a starting solution for local search. For each time period, the well known assignment problem is applied and used for finding a timetable with minimal violations of the teachers restrictions. In Section 5 we introduce four families of neighborhoods. The first two families use swapping of time periods for teacher and class timetables. The last two families are based on the idea of large Kernighan–Lin neighborhoods [5], [6]. In Section 6 we present the Probabilistic Tabu Search algorithm [2] where two types of the solution representations are systematically alternated to diversify the search. The randomization of the neighborhoods is used to speedup each iteration of the algorithm. In Section 7 the Variable Neighborhood Search heuristic [3] is presented. We modify the framework of this metaheuristic and change not only the neighborhoods but solution representations as well. In the final Section 8 we discuss the computational results for random test instances with real dimension for the School and illustrate the high efficiency of the proposed approach.

2. PROBLEM FORMULATION

In the teacher/class timetabling problem we are given the following finite sets: J is the set of subjects, K is the set of classes, L is the set of teachers, T is the set of time periods. All periods are distributed in 6 week days. By $T_l \subseteq T$ we denote the set of time periods which are available for teacher $l \in L$. We suppose that classes are disjoint sets of students. All students in each class have the same subjects, and correspondence between subjects and teachers for a chosen class is one-to-one. Without loss of generality we may assume that for each subject $j \in J$ it is known class $k(j)$ and teacher $l(j)$. By a_j we denote the number of lessons for subject $j \in J$. We assume that there is sufficient number of rooms in the School and it is possible to find an appropriated room for each class for any time period.

Let us introduce the decision variables:

$$x_{jt} = \begin{cases} 1, & \text{if subject } j \text{ is assigned in period } t \\ 0, & \text{otherwise.} \end{cases}$$

An arbitrary matrix $X = (x_{jt})$ is called a timetable.

Definition 1. We say that a timetable $X = (x_{jt})$ is feasible if the following restrictions hold:

a) a teacher l has at most one lesson at a time period t if $t \in T_l$ and no lessons otherwise:

$$\begin{aligned} \sum_{j|l(j)=l} x_{jt} &\leq 1, \quad t \in T_l, l \in L, \\ x_{jt} &= 0, \quad t \notin T_l, l \in L; \end{aligned}$$

b) a class k has at most one lesson at a time period t :

$$\sum_{j|k(j)=k} x_{jt} \leq 1, \quad k \in K, t \in T;$$

c) all subjects are distributed by time periods:

$$\sum_{t \in T} x_{jt} = a_j, \quad j \in J.$$

The goal is to find a feasible timetable with minimal number of violations of the following *soft* constraints:

1. each teacher has no time gaps;
2. each teacher has no lessons in inconvenient time periods;
3. each class has no double lessons.

This is a multi criteria discrete optimization problem. We reduce it to the well known timetable design problem (see [1], page 243) by a penalty function. More exactly, we wish to minimize the following objective function:

$$F(X) = \sum_{l \in L} \sum_{d=1}^6 \alpha_{ld} f_{ld}^1(X) + \sum_{l \in L} \sum_{t \in T} \beta_{lt} f_{lt}^2(X) + \sum_{k \in K} \sum_{d=1}^6 \gamma_{kd} f_{kd}^3(X),$$

where positive α , β , and γ are the penalties and $f^i(X)$ is the number of violations of the soft restriction i , $i = 1, 2, 3$.

The optimization problem is NP-hard. Moreover, the decision problem on existence of a feasible solution is NP-complete [1]. So, we introduce semi feasible solutions to enlarge the search space and apply metaheuristics to find near optimal solutions.

3. SOLUTION REPRESENTATIONS

We introduce two types of semi feasible solutions.

Definition 2. A timetable is a semi feasible solution of type *a* if it satisfies the restrictions (b) and (c).

Class\Period	1	2	3	4	5	...	T
1	1	2	3	1	2	...	4
2	J	0	J	5	6	...	5
3	11	8	7	9	8	...	11
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮
K	21	22	23	24	24	...	22

Figure 1: Semi feasible solution of type *a*

It is convenient to represent a such semi feasible solution as a $\mathbf{K} \times \mathbf{T}$ matrix (S_{kt}^a) , $\mathbf{K} = |K|$, $\mathbf{T} = |T|$, with values in $\{0, 1, \dots, \mathbf{J}\}$, $\mathbf{J} = |J|$, where the k -th row of the

matrix is a timetable for the k -th class. Nonzero entries of the row mean subjects for the class k at the time period t ; $S_{kt}^a = 0$ means free time. Figure 1 illustrates an example of the (S_{kt}^a) . For this representation, each class has at most one lesson per time period, but a teacher can get several lessons at the same time for different classes. Moreover, this time period may be unavailable for that teacher. Sure, we can easily compute solution X by the matrix S^a . So, we call S^a as a solution too.

Definition 3. A timetable is a semi feasible solution of type b if it satisfies the restrictions (a) and (c).

In a similar way we represent such a semi feasible solution as a $L \times T$ matrix (S_{lt}^b) , $L = |L|$, with values in $\{-1, 0, 1, \dots, J\}$ [10]. Entries of the matrix mean subjects for the teacher l at the time period t if $S_{lt}^b > 0$, and free time if $S_{lt}^b \leq 0$. The case $S_{lt}^b = -1$ means that the time period t is unavailable for the teacher l . In Figure 2 we present an example of (S_{lt}^b) .

Teacher\Period	1	2	3	4	5	...	T
1	1	0	J	2	2	...	J
2	3	-1	-1	22	0	...	3
3	-1	-1	8	14	4	...	-1
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮
L	7	9	5		21	...	0

Figure 2: Semifeasible solution of type b

The advantage of this representation is that it eliminates conflicts for teachers. The occurrence of conflicts in column happens when in a given period t more than one teacher is allocated to a class.

We say that a matrix S^a (a matrix S^b) contains m conflicts if m is a minimal number of its entries which vanishing lead that the restrictions (a) (restrictions (b)) hold. Sure, the restrictions (c) are not satisfied in this case.

Proposition 1. An arbitrary semi feasible solution of type a can be reduced to a semi feasible solution of type b in polynomial time without increasing of the number of conflicts.

Proof: Let m denote the number of conflicts for a solution S^a . We replace m entries in S^a by 0 in such a way that the restrictions (a) hold and denote the modified matrix by S' . The restrictions (a) and (b) hold for it and the total violation for the restrictions (c) is m . Using S' we can compute a matrix S^b with the same property. In order to satisfy the restriction (c) we identify the lost lessons for each teacher and spread these lessons through 0 entries of the S^b . As a result, the restrictions (c) hold, but the restrictions (b)

may be not satisfied. So, the S^b is semi feasible solution of type b . Obviously, the number of conflicts is at most m which complete the proof.

Proposition 2. *An arbitrary semi feasible solution of type b can be reduced to a semi feasible solution of type a in polynomial time without increasing of the number of conflicts.*

The proof is similar.

It is noticed that an arbitrary feasible solution can be presented as a solution S^a or S^b , but the inverse statement is not true.

For a semi feasible solution X let us introduce the following penalty function:

$$\bar{F}(X) = F(X) + \sum_{l \in L} \sum_{t \in T} \lambda_{lt} f_{lt}^a(X) + \sum_{k \in K} \sum_{t \in T} \mu_{kt} f_{kt}^b(X),$$

where $\min(\lambda, \mu) > \max(\alpha, \beta, \gamma)$ and the function $f^a(X)$, $f^b(X)$ are defined the number of violations of the restrictions (a) and (b) correspondingly. If $F(X) = F(X)$ then X is a feasible solution. If $F(X) = 0$ then X is optimal solution. But the optimal solution may have a positive value of the penalty function.

4. STARTING SOLUTION

In order to get a starting semi feasible solution for the local search methods we use the well known assignment problem [8]. We apply it for each time period t one by one with adaptation of the problem parameters. As a result we create a semi feasible solution of type b .

Let us introduce auxiliary variables

$$z_{kl} = \begin{cases} 1 & \text{if teacher } l \text{ has a lesson in class } k \text{ in time period } t, \\ 0, & \text{otherwise.} \end{cases}$$

Sure, if the set $K(l) = \{k \in K \mid \exists j \in J : k(j) = k, l(j) = l\}$ is empty or $t \notin T_l$ then we have to put $z_{kl} = 0$. But if $|K(l)| \geq 1$, we have an assignment problem for teachers and classes.

Let $b_l(t)$ denote the total number of lessons which are not included in the starting timetable for teacher l up to the time period t , $c_l(t)$ is the cardinality of the set $T_l \setminus \{1, \dots, t-1\}$ and $h_{kl}(t)$ denote the total number of lessons in class k with teacher l which are not included in the starting timetable up to the time period t . Put

$$H_{kl}^t = \begin{cases} b_l(t)h_{kl}(t)/c_l(t), & \text{if } b_l(t) < c_l(t), k \in K(l), t \in T_l, \\ H, & \text{if } b_l(t) = c_l(t), k \in K(l), t \in T_l, \\ -1, & \text{if } t \notin T_l, \\ 0, & \text{otherwise,} \end{cases}$$

where H is large number. Now, the assignment problem for the time period $t \in T$ is the following:

$$\begin{aligned}
& \max \sum_{k \in K} \sum_{l \in L} H_{kl}^t z_{kl} \\
\text{s.t.} \quad & \sum_{l \in L} z_{kl} \leq 1, \quad k \in K, \\
& \sum_{k \in K} z_{kl} \leq 1, \quad l \in L, \\
& z_{kl} \in \{0,1\}, \quad l \in L, k \in K.
\end{aligned}$$

Let z_{kl}^* be the optimal solution of the problem. Define

$$S_{lt} = \begin{cases} -1, & \text{if } t \notin T_l, \\ k, & \text{if } z_{kl}^* = 1, t \in T_l, k \in K(l), \\ k^0 = \min\{k \mid h_{kl}(t) > 0\}, & \text{if } b_l(t) = c_l(t), \sum_{k \in K} z_{kl}^* = 0 \text{ and } t \in T_l, \\ 0 & \text{for all other cases.} \end{cases}$$

In order to get a semi feasible solution of type b we should define a subject j for each positive entry of the matrix (S_{lt}) . By definition of the parameters H_{kl}^t , the set of appropriated subjects is not empty. We select one of them at random.

5. NEIGHBORHOODS

Now we introduce four families of neighborhoods:

- $N_i(S^a)$, $i \geq 1$, denote swap neighborhoods of a semi feasible solution S^a ,
- $N_i(S^b)$, $i \geq 1$, denote swap neighborhoods of a semi feasible solution S^b ,
- $KL_i(S^a)$, $i > 1$, denote Kernighan–Lin neighborhoods of S^a ,
- $KL_i(S^b)$, $i > 1$, denote Kernighan–Lin neighborhoods of S^b .

The neighborhood $N_1(S^a)$ consists of neighboring solutions which are obtained from S^a by swapping two different values of a given row in the matrix (S_{kt}^a) . Each element in this neighborhood is associated with a triplet $\langle k, t', t'' \rangle$, where t' and t'' are the time periods, k is the class, and $S_{kt'}^a$ and $S_{kt''}^a$ are the interchanged subjects. For $i > 1$, $N_i(S^a)$ are formed of solutions which are obtained by the sequence of interchanges with triplets $\{\langle k, t'_j, t''_j \rangle\}_{j \leq i}$, $k \in K$ is fixed. Families $N_i(S^b)$ are defined in a similar way. Moreover, only non-negative values of the matrix (S_{lt}^b) can be interchanged. We note that arbitrary feasible solution can be reached with the use of an appropriate sequence of neighboring solutions for the neighborhoods $N_1(S^a)$ or $N_1(S^b)$.

A Kernighan–Lin neighborhood $KL_i(S^a)$ consists of i elements and can be described by the following steps [5].

1. Choose a triplet $\langle k, t', t'' \rangle$ such that the corresponding neighboring solution $S' \in N_1(S^a)$ is the best even if it is worse than S^a .
2. Put $S^a := S'$.
3. Repeat steps 1, 2 i times; if a triplet was used at steps 1 or 2 of previous iterations, it can not be used any more.

The sequence of triplets $\{\langle k, t'_j, t''_j \rangle\}_{j \leq i}$ defines i neighbors S_j of the solution S^a . We say that S^a is a local minimum with respect to the KL_i –neighborhood if $\bar{F}(S^a) \leq \bar{F}(S_j)$ for all $j \leq i$. A local minimum with respect to the KL_i –neighborhood is a local minimum with respect to N_1 and is not necessary a local minimum with respect to N_i , $i > 1$. Family $KL_i(S^b)$ is defined similarly.

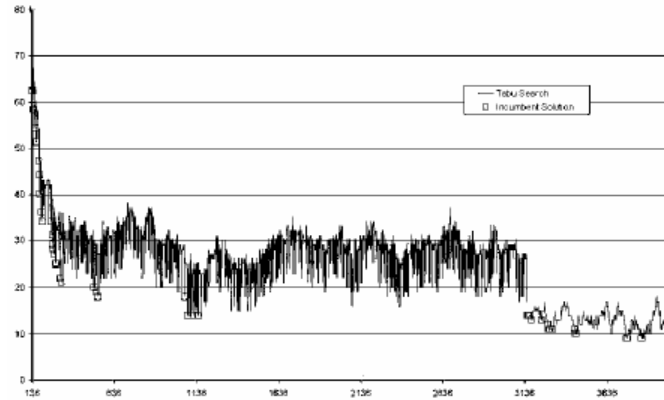


Figure 3: Typical PTSR behavior, $p = 0.5$

6. PROBABILISTIC TABU SEARCH

The Tabu Search method [2] is one of the powerful approaches for solving difficult combinatorial problems. We develop a probabilistic variant of this method where the solution representations are systematically changed during the search (PTSR for short). Let N^p denote a random part of the neighborhood N_1 . Each element of the N_1 is included into the N^p with probability $p > 0$ independently from other elements. We use this randomized neighborhood on each iteration of the PTSR. If $N^p = \emptyset$ for an iteration of the algorithm we omit it. The tabu list consists of the triplets $\langle k, t', t'' \rangle$ or $\langle l, t', t'' \rangle$ according to the solution representations. We change the solution

representation for diversification of the search if the best found solution S^* is not updated for a long time. The framework of the PTSR is the following:

PTSR Algorithm

1. Compute a starting solution S , put $S^* := S$ and create empty tabu list.
2. Repeat the following sequence until the stopping condition is met:
 - (a) Find the best nontabu neighbor $S' \in N^P(S)$.
 - (b) Put $S := S'$, and update the tabu list.
If $\bar{F}(S) < \bar{F}(S^*)$ then $S^* := S$.
If $\bar{F}(S) = 0$ then goto 3.
 - (c) Change representation of the current solution and put $S := S^*$ if S^* is not updated during prescribed number of iterations.
3. Return S^* .

Figure 3 shows the typical behavior of the PTSR. By small square we indicate the iterations when the incumbent solution S^* is updated. It is easy to identify the iteration 3000 when the solution representation is changed. We return to the best found solution (step 2(c) $S := S^*$) and get a new type of curve. As a result, new incumbent solutions founded.

7. VARIABLE NEIGHBORHOOD SEARCH

We adjust the framework of the Variable Neighborhood Search (VNS) meta-heuristic [3] for our problem and change not only the neighborhoods but solution representations as well.

VNSR Algorithm

1. Compute a starting solution S and sizes of neighborhood i_{\max}, j_{\max} .
2. Repeat the following sequence until the stopping condition is met:
 - (a) Set $i := 1$; if $F(S) = 0$ then goto 3.
 - (b) Repeat the following steps until $i = i_{\max}$:
 - i) Shaking. Generate a solution S' at random from the $N_i(S)$.
 - ii) Local search. Apply the local descent algorithm with respect to N_1 neighborhood.
 - iii) Large neighborhood search. Apply the local descent algorithm with respect to $KL_{j_{\max}}$ neighborhood. Denote the obtained local minimum as S'' .
 - iv) Move or not. If $\bar{F}(S'') < \bar{F}(S)$ then $S := S''$ and go to 2(a); otherwise, set $i \leftarrow i + 1$.
 - (c) Change the solution representation and go to 2(a).
3. Return S .

Figure 4 shows the typical behavior of the VNSR. The generation of random solution at the step Shaking is shown by dotted lines. One can observe large peaks which correspond to random jumps by $i \leq i_{\max}$ triplets. The continuous lines show the local descent under the N_1 neighborhood. The small dotted lines correspond to the local search under the KL-neighborhood. Note that the current solution S in this algorithm is actually the incumbent solution. So, additional intensification actions are not required.

8. COMPUTATIONAL RESULTS

The developed software are tested on random benchmarks with real dimensions for the School: $\mathbf{K} = 44$, $\mathbf{L} = 120$, $\mathbf{T} = 6 \times 3 = 18$ and $\mathbf{J} = 44 \times 18 = 792$.

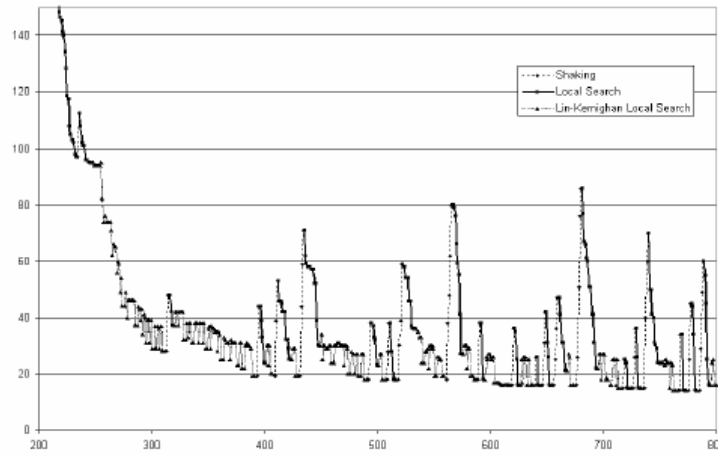


Figure 4: Typical VNSR behavior, $i_{\max} = 5$, $j_{\max} = 30$

Each teacher has $\mathbf{T}/2$ available time periods and 20% of them are inconvenient. Each class has \mathbf{T} lessons. The penalty coefficients are defined as follows: $\alpha = 1 + \alpha'$, $\beta = 3 + \beta'$, $\gamma = 5 + \gamma'$, $\lambda = 10 + \lambda'$, $\mu = 10 + \mu'$, where $\alpha', \beta', \gamma', \lambda', \mu'$ are random noise, $0 < \alpha', \beta', \gamma', \lambda', \mu' \leq 1$. This rule removes plateaus and improves the landscape for local search methods. The running time of the algorithms is limited by 5 minutes of IBM PC Pentium 1.8MgG.

Table 1: Computational results for the first class of benchmarks

N	PTSR			VNS			VNSR			Opt
	%Opt	Aver	Worst	%Opt	Aver	Worst	%Opt	Aver	Worst	
1	97	92.05	95.0	96	92.04	94.0	93	92.07	93.0	92
2	89	78.16	80.0	81	78.44	83.0	87	78.32	81.0	78
3	100	137.0	137.0	93	137.09	140.0	93	137.11	140.0	137

4	36	97.96	101.0	38	97.9	101.0	52	97.69	103.0	97
5	90	97.16	99.0	83	97.31	100.0	87	97.28	103.0	97

Table 1 shows the optimal values (Opt) for 5 instances and frequency (%Opt) of finding this values by metaheuristics. We solve each instance 100 times and present the average values of the best found solutions (Aver) and the worst values (Worst). Note that the difference between these columns is quite small. So, we have robust local search methods. For all instances we find the optimal solutions by CPLEX software in about 1.5 hours. The metaheuristics discover these solutions in 1.5 min in average. For comparing, we present the result for VNS algorithm without changing representations. For easy instances (all except 4) the both variants VNS and VNSR show the same results. But for the difficult instance 4, where the frequency of finding the optimal solution is small, we observe some advantages of VNSR against VNS. The same effect we can see for another class of test instances. We test the algorithms for benchmarks with the same dimensions but now each teacher has 30% of available time periods and 20% of them are inconvenient for teaching. In other words, the teachers are busier.

Table 2: Computational results for the second class of benchmarks

N	PTSR			VNS			VNSR			Opt
	%Opt	Aver	Worst	%Opt	Aver	Worst	%Opt	Aver	Worst	
1	18	82.3	86.0	62	80.48	83.0	58	80.54	83	80
2	24	112.38	118	58	110.9	113.0	76	110.56	113.0	110
3	16	139.56	145.0	50	137.7	141.0	40	137.8	140.0	137
4	0	137.16	155.0	0	137.84	153.0	6	132.5	144.0	124
5	28	121.02	125.0	42	121.04	130.0	48	120.58	130.0	119

Table 2 shows the computational results for this case. Sure, it is more difficult class. The frequency of finding the optimal solutions is not so optimistic as for previous case but the difference between Average values and Optimal values is small again. So, we may conclude that the local search methods are efficient and effective. Change of the solution representation is a useful idea for both methods. We hope it may be successfully applied for other approaches as well.

9. CONCLUSIONS

In this paper we consider the teacher/class timetabling problem and develop local search heuristics based on idea of FSS approach. Two types of solution representations are used in the heuristics and systematically alternate during the search process. We show by computational experiments that idea of two equivalent but not identical solution representations is useful for this timetable problem. We believe that the FSS approach can improve the local search heuristics for other NP-hard problems.

Acknowledgment: This work was partially supported by Russian Foundation of Basic Research, grant 06-01-00075.

REFERENCES

- [1] Garey, M.R., and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.
- [2] Glover, F., and Laguna, M., *Tabu Search*, Kluwer Academic Publishers, Boston–Dordrecht–London, 1997.
- [3] Hansen, P., Mladenović, N., "Developments of Variable Neighborhood Search", in: Ribeiro, C.C., Hansen, P. (eds.) *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, 2002, 415-440.
- [4] Hertz, A., Plumettaz, M., and Zufferey, N. "Variable space search for graph coloring", GERAD technical report g-2006-81, Montreal 2006 (to appear in *Discrete Applied Mathematics*).
- [5] Kernighan, B.W., and Lin, S., "An efficient heuristic procedure for partitioning graphs", *Bell System Technical Journal*, 49 (1970) 291-307.
- [6] Kochetov, Y., Alekseeva, E., Levanova, T., and Loresh, M., "Large neighborhood local search for the p-median problem", *Yugoslav Journal of Operations Research*, 15 (1) (2005) 53-63.
- [7] Mladenović, N., Plastria, F., and Urošević, D., "Reformulation descent applied to circle packing problems", *Computers and Operations Research*, 32 (9) (2005) 2419-2434.
- [8] Papadimitriou, C.H., and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [9] Petrovic, S., and Burke, E., "University timetabling", in: J.Y-T. Leung (ed.) *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, Chapman&Hall/CRC, 2004, 45.1-45.23.
- [10] Santos, H.G., Ochi, L.S., and Souza, M.J.F., "A Tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem", *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, Pittsburgh, USA, 2004, 343-358.