VNS BASED ALGORITHMS TO THE HIGH SCHOOL TIMETABLING PROBLEM

Landir Saviniec, Ademir Aparecido Constantino, Wesley Romão
Departamento de Informática – Universidade Estadual de Maringá (UEM)
Avenida Colombo, 5790 – 87020-900 – Bloco C56 – Maringá – PR – Brazil
{landir.saviniec, ademir.uem, wesley.uem}@gmail.com

ABSTRACT

The high school timetabling is a classical NP-Hard problem in combinatorial optimization. Since the use of exact methods for this problem is restricted, heuristics are usually employed. This paper presents an investigation on the development of a Variable Neighborhood Search (VNS) method which includes two powerful neighborhood operators to heuristically solve this problem. Two VNS based algorithms are presented and experimented on a benchmark data set from literature. The results have shown that the two algorithms are effective and efficient, as they have found proven optimal solutions (using pre-computed lower bounds) on a reasonable amount of time.

KEYWORDS. High School Timetabling Problem, Variable Neighborhood Search, Neighborhood Operators.

Main area. OR in Education, Metaheuristics, Combinatorial Optimization.

1. Introduction

The high school timetabling problem (HSTP) (SCHAERF, 1999; PILLAY, 2010) is a hard combinatorial optimization problem and taking into account the computational complexity theory, it is NP-Complete (EVEN *et al.*, 1975). Besides the original, already complicating constraints, real cases can include a multitude of different ones, as those collected in (POST *et al.*, 2012). As the best known algorithms to solve the problem to optimality are exponential time, their applicability to solve real instances of the problem, become impracticable due to the large amount of computational time required. For this reason, the problem is tackled by heuristic methods. Such methods do not guarantee to solve the problem to optimality, but are capable to find good solutions in a feasible computational time.

It is often the case that a timetabling problem is solved manually and in practice many constraints cannot be solved. In addition, a handmade timetable can take several days to be accomplished and due to frequent changes in the resources requirements, i.e. teachers who leave the school in medical absences, it has to be remade causing inconveniences for the school staff. For these reasons, this task is considered an onerous process.

Due to the above mentioned, more attention have been devoted to automatization of the timetabling process during the last years by researchers. In special, in the last two decades a large number of experimental papers tackling the problem by heuristics have been introduced at the literature.

The most common methods used at the literature to solve the problem are: Genetic Algorithms (SOUZA *et al.*, 2002); Simulated Annealing (BRITO *et al.*, 2012); Tabu Search (SANTOS *et al.*, 2005); Greedy Randomized Adaptive Search Procedures (SOUZA *et al.*, 2003); Variable Neighborhood Search (BRITO *et al.*, 2012) and Iterated Local Search (SAVINIEC; CONSTANTINO, 2012).

This paper is an extension of the research presented in (SAVINIEC *et al.*, 2013). There, we have proposed three iterated local search algorithms including two newly proposed neighborhood operators to solve the HSTP. These algorithms were successfully experimented on a well known benchmark data set of the problem (SOUZA *et al.*, 2003) and the results have encouraged us to test these same neighborhood operators with other metaheuristic. Taking this in mind, this paper proposes two algorithms based on the VNS metaheuristic to solve the problem. These two VNS algorithms are applied on the benchmark data set from (SOUZA *et al.*, 2003) and to validate the proposal, we contrast obtained results with lower bounds known for these instances (SANTOS *et al.*, 2012). The main findings are: the improved upper bounds obtained by our VNS algorithms have allowed us to prove optimality for instances where strong lower bounds were previously known and the statistical distribution of solutions of these two algorithms are very close to the optimal solutions.

The paper is organized as follows: Section 2 defines the problem. Section 3 explains the solution approach. Section 4 reports the experimental results and section 5 provides a summary and future works.

2. Problem definition

The HSTP considered in this paper (SOUZA *et al.*, 2003) is based on Brazilian high schools. There is a set $P = \{p|1 \leq p \leq np, p \in \mathbb{N}\}$ of teachers who teach a set $T = \{t|1 \leq t \leq nt, t \in \mathbb{N}\}$ of classes at school in a given shift, during a set $D = \{d|1 \leq d \leq nd, d \in \mathbb{N}\}$ of days, with each day composed by a set $H = \{h|1 \leq h \leq nh, h \in \mathbb{N}\}$ of periods. Classes are disjoint groups of students having the same subjects and no idle time periods during the week, and each subject of a class is taught by only one teacher. Lessons between teachers and classes are previous defined by the school. Classrooms are predefined and not considered in the scheduling. Most of the teachers are not full time at school, thus teachers' availability have to be considered and their workload have to be concentrated in a minimum number of days during the week. In this way, an instance of the problem is according to definition 2.1.

Definition 2.1 (HSTP instance) *An instance of the HSTP is the data entry to the timetable construction process in a given shift and it is represented by the following sets:*

- *A set* $L = \{\langle t, p, \theta, \lambda, \mu\rangle | t \in T,\ p \in P,\ \theta \in \mathbb{N},\ \lambda \in \mathbb{N}\ e\ \mu \in \mathbb{N}\}$ *of quintuples, named as lessons requirement set. Where* θ *is the number of lessons,* λ *is the maximum number of permitted lessons per day and* μ *is the minimum number of double lessons requested by teacher* p *with class* t.
- *A set* $U = \{\langle p, d, h\rangle | p \in P,\ d \in D,\ h \in H\}$ *of triples, named as set of teachers' unavailable periods. Where exists a triple* $\langle p, d, h\rangle$ *if teacher* p *is unavailable at period* h *of day* d.

Then, the problem consists in the scheduling of a weekly timetable Z, composed by five days with five periods each, for the lessons in L, satisfying the hard constraints (definition 2.2) and minimizing the soft constraints (definition 2.3).

Definition 2.2 (Hard constraints) *The hard constraints are represented by the set* $A = \{a_i | 1 \leq i \leq 5\}$ *of constraints:*

a_1 : *every* θ *lessons required for class* t *and teacher* p *must be scheduled;*
a_2 : *a class must attend a lesson with only one teacher by period;*
a_3 : *a teacher must teach only one class by period;*
a_4 : *teachers must not be scheduled in periods they are not available;*
a_5 : *a class* t *must not be scheduled to attend more than* λ *lessons with a same teacher* p *per day.*

Definition 2.3 (Soft constraints) *The soft constraints are represented by the set* $B = \{b_j | 1 \leq j \leq 3\}$ *of constraints:*

b_1 : *the number* μ *of double lessons requested by teacher* p *with class* t *has to be attended;*
b_2 : *idle times in the scheduling of teachers should be avoided;*
b_3 : *the scheduling for each teacher should encompass the least possible number of days.*

3. Heuristic approach

This section discusses some fundamental concepts for building heuristic approaches and defines the proposed approach to solve the HSTP. In the following, we present each component that composes our approach: solution representation structure (section 3.2), objective function (section 3.3), the heuristic used to build initial solutions (section 3.4), the local search technique applied (section 3.5), the neighborhood operators (section 3.6) and the ILS algorithms employed to solve the problem (section 3.7).

3.1. Concepts

On the context of combinatorial optimization (CO) problems, all possible solutions for a given instance of a problem, feasible or not, define the solution (or search) space S, and each solution in S can be seen as a candidate solution. Thus, solving a CO problem requires to formulate it as a maximization or minimization problem. In this type of formulation there is an objective function $f : S \rightarrow \mathbb{R}$ and the problem consists in finding solutions that maximize or minimize f.

On the context of the high school timetabling, the problem is generally formulated as minimization and f is measured by weighting the number of violations for each constraint of the problem and the aim is to satisfy the hard constraints and minimize the soft constraints. Then, to solve the problem, one has to find a solution $Z^* \in S$ with minimum objective function, that is,

$f(Z^*) \leq f(Z), \forall Z \in S$, where Z^* is called **global minimum** in S and the set $S^* \subseteq S$ of all solutions Z^* is the **set of global minimum**.

A powerful class of algorithms to solve CO problems, in which no polynomial time algorithm is known, are heuristic algorithms based on the concept of local search.

A local search heuristic starts from an initial solution Z_0 and iteratively replaces the current solution Z by a better solution Z' in an appropriately defined neighborhood $N(Z)$ of the current solution, until no more improvements are possible and the heuristic gets stuck in a **local minimum**.

Neighborhoods are generated by neighborhood operators (definition 3.1) and they enable to define the concept of **local minimum** (definition 3.2).

Definition 3.1 (Neighborhood operator) *A neighborhood operator is a function $N : S \rightarrow P(S)$ that assigns to every solution $Z \in S$ a set of neighbors $N(Z) \subseteq S$. $P(S)$ is the power set of S and $N(Z)$ is called neighborhood of Z.*

Definition 3.2 (Local minimum) *A local minimum solution with respect to a neighborhood operator N is a solution $Z^{*\prime}$, such that $\forall Z \in N(Z^{*\prime}) \Rightarrow f(Z^{*\prime}) \leq f(Z)$.*

3.2. Solution representation

A solution of the HSTP is represented according to definition 3.3.

Definition 3.3 (HSTP solution) *A HSTP solution is stored in a non-negative integer three-dimensional matrix $Z_{|T|\times|D|\times|H|}$, where $z_{t,d,h} \in \{1, 2, \ldots, np\}$ stores the teacher scheduled to teach for class t on period h of day d.*

Note that using this structure, constraints a_1 and a_2 are automatically satisfied and they are not included on the objective function.

3.3. Objective function

In order to solve the HSTP, it is treated as an optimization problem in which an objective function $f : S \rightarrow \mathbb{R}$ has to be minimized. The objective function f associates each solution Z in the solution space S to a real number and this is defined to measure the violation degree on the HSTP constraints. Thus, a timetable solution Z is evaluated according to the objective function in definition 3.4.

Definition 3.4 (Objective function) *A HSTP solution Z is evaluated by the following function:*

$$f(Z) = f_A(Z) + f_B(Z) \quad (1)$$

Such that:

$$f_A(Z) = \sum_{i=3}^{5} \alpha_{a_i} \times \beta_{a_i} \quad (2)$$

$$f_B(Z) = \sum_{j=1}^{3} \alpha_{b_j} \times \beta_{b_j} \quad (3)$$

Where equations 2 and 3, respectively, measure the feasibility and quality of a timetable solution and the weight α_{a_i} (resp. α_{b_j}) reflects the relative importance of minimizing the amount of violation β_{a_i} (resp. β_{b_j}) at constraint $a_i \in A$ (resp. $b_j \in B$).

From definition 3.4 a timetable is feasible if $f_A(Z) = 0$ and the variables β_{a_i} and β_{b_j} are computed as below.

$\beta_{a_3} = \sum_{p\in P} \sum_{d\in D} \sum_{h\in H} (\pi_{p,d,h} - 1), \forall(\pi_{p,d,h} > 1)$. Where $\pi_{p,d,h}$ is the total number of lessons allocated for teacher p on period h of day d;

$\beta_{a_4} = \sum_{p \in P} \sum_{d \in D} \sum_{h \in H} \rho_{p,d,h}$. Where $\rho_{p,d,h} = 1$ if teacher p has been scheduled to teach at an unavailable period h on day d, and $\rho_{p,d,h} = 0$ otherwise;

$\beta_{a_5} = \sum_{t \in T} \sum_{p \in P} \sum_{d \in D} (\sigma_{t,p,d} - \lambda_{t,p}), \forall(\sigma_{t,p,d} > \lambda_{t,p})$. Where $\sigma_{t,p,d}$ is the total number of lessons allocated for class t with teacher p on day d and $\lambda_{t,p}$ is the maximum of permitted lessons per day from definition 2.1;

$\beta_{b_1} = \sum_{t \in T} \sum_{p \in P} (\mu_{t,p} - \phi_{t,p}), \forall(\mu_{t,p} > \phi_{t,p})$. Where $\mu_{t,p}$ is the minimum number of double lessons requested by teacher p with class t (definition 2.1) and $\phi_{t,p}$ is the effective number of allocated double lessons;

$\beta_{b_2} = \sum_{p \in P} \sum_{d \in D} \eta_{p,d}$. Where $\eta_{p,d}$ is the number of idle times at the agenda of teacher p on day d. For example, if a teacher has been scheduled to teach at the first and fourth periods and is free at the second and third ones, then he has two idle times on this day;

$\beta_{b_3} = \sum_{p \in P} \chi_p$. Where χ_p is the total number of scheduled days for teacher p on the timetable.

3.4. Algorithm for building initial solutions

In this work, initial solutions of the HSTP are constructed by means of a randomized algorithm (see algorithm 3.1). This algorithm gets the lessons requirement set L from definition 2.1 as input and builds an initial solution by selecting and scheduling lessons randomly.

Algorithm 3.1 Algorithm for building initial solutions

GENERATE-RANDOM-SOLUTION(L)

$Initialize\ Z$
for **each** $e \in L$ **do**
  $t = e.t$
  $p = e.p$
  $NumberOfLessons = e.\theta$
  while $NumberOfLessons > 0$ **do**
    Put p in a randomly selected free cell $z_{t,d,h} \in Z$
    $NumberOfLessons = NumberOfLessons - 1$
return Z

3.5. Local search

In summary, for CO problems, given an initial solution Z_0 as input, a local search heuristic moves from Z_0 to a local minimum Z' by exploring neighborhoods. At the literature, the most used techniques to perform local search are: best improvement and first improvement (HANSEN *et al.*, 2010):

Best improvement: the heuristic start at an initial solution $Z' = Z_0$, and at each iteration, replaces Z' by $Z = min\{Z'' \in N(Z')\}$ while $f(Z) < f(Z')$. This technique explores the whole neighborhood and moves to the best solution.

First improvement: this technique is an alternative to the first one when the neighborhood is large to be entirely explored. This is similar to the first, but at each iteration it moves to the first solution $Z_i \in N(Z')$ found, if it improves the current solution Z'.

In this paper the first improvement technique is employed as local search (algorithm 3.2).

3.6. Neighborhood operators

Neighborhood operators are the key ingredient to develop powerful local search algorithms. Some researches (LAARHOVEN *et al.*, 1992; DELL'AMICO; TRUBIAN, 1993;

Algorithm 3.2 First improvement heuristic

FIH(Z_0, N)

$Z = Z_0$
repeat
  $Z' = Z$
  $i = 0$
  repeat
    $i = i + 1$
    $Z = min\{Z, Z_i\}$, $Z_i \in N(Z')$
  until $(f(Z) < f(Z')$ or $i = |N(Z')|)$
until $(f(Z) \geq f(Z'))$
return Z'

OSOGAMI; IMAI, 2000) have demonstrated, for some CO problems, that it is possible to define neighborhood operators that reduce the search space. Such operators exclude out of the search process, a large set of non-feasible solutions and the local search algorithm can efficiently search the restricted solution space.

In this paper two neighborhood operators, named as **Matching operator** (**MT**) and **Torque operator** (**TQ**), are employed. These operators exclude out of the search process a large set of undesirable solutions. These two operators are proposed and detailed described in (SAVINIEC *et al.*, 2013). By this reason, this paper will only comment them.

Matching operator: this is based on the resolution of Assignment Problems (AP). A neighbor of a solution Z is obtained by selecting a random class, and from this, a random set $\widehat{Y}$ of non-repeated teachers. This set of teachers are moved out of the timetable and we solve an assignment problem to re-schedule them. Figure 1 illustrates an operation of MT. To simplify, in this example only constraint a_3 with weight $\alpha_{a_3} = 1$ is taken into account. The MT operator is applied on lessons of class t_3 at the solution Z in figure 1(a), where $\widehat{Y} = \{\overbrace{2}^{t_3d_1h_1}, \overbrace{9}^{t_3d_1h_2}, \overbrace{1}^{t_3d_1h_3}, \overbrace{6}^{t_3d_1h_5}\}$ is the non-repeated set of teachers. Figures 1(b)-1(e) illustrate how to construct and solve an AP for the set $\widehat{Y} = \{2, 9, 1, 6\}$ and figure 1(f) shows the obtained neighbor Z'.

Torque operator: this is a generalization for the well known double move operator (DM) generally used to solve the HSTP. The DM operator consists in swapping two lessons of a class that are scheduled in two different periods h_i and h_j, for example. But when applying moves using DM operator, new clashes between lessons can occur and a_3 constraint is violated. Thus, the TQ operator is developed to prevent this disadvantage using the idea of *Kempe chain interchange* (LÜ *et al.*, 2011). In this operator, a graph G is built with lessons from two distinct periods h_i and h_j, where nodes are pairs of lessons and edges are added between nodes having conflicted lessons. A neighbor of a solution Z is obtained by swapping the lessons in each node of a connected component of G. Figure 2 illustrates this operator.

3.7. VNS based algorithms to the HSTP

The proposed approach is composed by two algorithms based on the VNS metaheuristic (HANSEN *et al.*, 2010).

VNS-MT-TQ: this VNS (algorithm 3.6) performs local search by exploring firstly, six neighborhoods $MT(Z)$ of different sizes and $TQ(Z)$ in the sequence.

VNS-TQ-MT: this algorithm is analogous to the first one, but in this case $TQ(Z)$ is explored before $MT(Z)$.

Classes	d_1				
	h_1	h_2	h_3	h_4	h_5
t_1	10	7	15	17	15
t_2	1	6	5	4	7
t_3	2	9	1	9	6
t_4	6	10	12	2	11
t_5	6	9	9	8	10

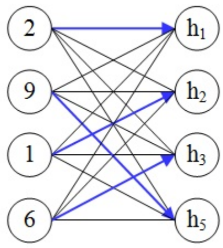
(a) Selection of a set $\widehat{Y}$ at a solution Z, where $f(Z) = 2$

Classes	d_1				
	h_1	h_2	h_3	h_4	h_5
t_1	10	7	15	17	15
t_2	1	6	5	4	7
t_3				9	
t_4	6	10	12	2	11
t_5	6	9	9	8	10

(b) Moving out of Z the set $\widehat{Y}$ and getting $Z - \widehat{Y}$ with $f(Z - \widehat{Y}) = 1$

	h_1	h_2	h_3	h_5
p=2	1	1	1	1
p=9	1	2	2	1
p=1	2	1	1	1
p=6	2	2	1	1

(c) Cost matrix $C_{|\widehat{Y}|\times|\widehat{Y}|}$



(d) Solving the AP

	h_1	h_2	h_3	h_5
p=2	1	1	1	1
p=9	1	2	2	1
p=1	2	1	1	1
p=6	2	2	1	1

(e) Resulting assignment x_{ij}

Classes	d_1				
	h_1	h_2	h_3	h_4	h_5
t_1	10	7	15	17	15
t_2	1	6	5	4	7
t_3	2	1	6	9	9
t_4	6	10	12	2	11
t_5	6	9	9	8	10

(f) The obtained neighbor Z' using x_{ij} information, with $f(Z') = 1$

Figure 1. The matching operator

These algorithms make use of the N-RANDOM-PERTURBATION procedure (algorithm 3.3), that applies a random move by using the TQ operator to perform perturbation and escape from local minimum.

Algorithm 3.3 Perturbation procedure

```
N-RANDOM-PERTURBATION(Z, N, n)
while (n > 0) do
    Z = Random Z' ∈ N(Z)
    n = n − 1
return Z
```

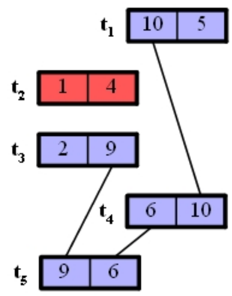
4. Experimental results

This section reports the experimental results of running the proposed algorithms on the high school timetabling benchmark from (SOUZA *et al.*, 2003). This benchmark has seven instances where timetables have five days ($nd = 5$) with five periods each ($nh = 5$). For the three largest instances in this benchmark, the optimal solutions are not known, however, lower bounds were computed using an extended Integer Linear Programming Formulation (SANTOS *et al.*, 2012).

The proposed approach was coded using MS Visual Basic 6. The experiment was performed on Windows Server 2008-R2 running on the KVM virtual machine set to work with 30GB of RAM and 50 cores of a server with 4 CPU Intel Xeon E7-4860 (24MB of Cache -

Classes	d_1				
	h_1	h_2	h_3	h_4	h_5
t_1	10	5	15	17	15
t_2	1	4	5	6	7
t_3	2	9	1	9	6
t_4	6	10	12	2	11
t_5	9	6	9	8	2

(a) Solution Z without clashes at constraint a_3



(b) Graph G for two distinct periods $h_i = h_1$ and $h_j = h_2$

Classes	d_1				
	h_1	h_2	h_3	h_4	h_5
t_1	5	10	15	17	15
t_2	1	4	5	6	7
t_3	9	2	1	9	6
t_4	10	6	12	2	11
t_5	6	9	9	8	2

(c) The neighbor Z' without clashes in a_3, after swapping the lessons at the blue color connected component

Figure 2. The torque operator

Algorithm 3.4 Randomized first improvement heuristic

RANDOMIZED-FIH(Z, n)

repeat
    $Z' = Z$
    $i = n \times |T|$
    while $(i > 0)$ **do**
        t = Random $t' \in T$ // select a random class t
        $\widehat{Y}$ = Random $\widehat{Y}' \in U_t$ // select a random set $\widehat{Y}$ in U_t
        Z = MT$(Z, \widehat{Y})$ // solve the AP to the set $\widehat{Y}$ and return the neighbor of Z
        $i = i - 1$
until $(f(Z) \geq f(Z'))$
return Z'

Algorithm 3.5 Acceptance criterion and operator change procedure

NEIGHBORHOOD-CHANGE$(Z^*, Z^{*\prime}, k)$

if $f(Z^{*\prime}) < f(Z^*)$ **then**
    $Z^* = Z^{*\prime}$
    $k = 1$
else
    $k = k + 1$

Algorithm 3.6 VNS-MT-TQ algorithm

```
VNS-MT-TQ(Z_0, t_max)
k_max = 7
Z* = Z_0
repeat
    k = 1
    repeat
        Z' = N-RANDOM-PERTURBATION(Z, TQ, 1)
        if k <= 6 then
            Z*' = RANDOMIZED-FIH(Z', k)      // local search using MT operator
        else
            Z*' = FIH(Z', TQ)                // local search using TQ operator
        NEIGHBORHOOD-CHANGE(Z*, Z*', k)      // acceptance criterion and operator change
    until (k > k_max)
    t = CPUTIME()
until (t > t_max or f(Z) = 0)
return Z*
```

2.26 GHz) with Linux CentOS 6 operating system. In this experiment 50 tests of 900 seconds were carried out for each instance. The whole experiment was performed in two phases, at each phase an algorithm was experimented by executing 50 simultaneous processes. The constraints were penalized with the follow weights on the objective function: $\beta_{a_3} = 100.000$, $\beta_{a_4} = 5.000$, $\beta_{a_5} = 100$, $\beta_{b_1} = 1$, $\beta_{b_2} = 3$, $\beta_{b_3} = 9$.

Table 1. Best results

Instance	$\|T\|$	$\|P\|$	Lessons	VNS-MT-TQ	VNS-TQ-MT	LB	TS	IP
1	3	8	75	$*$	$*$	202	$*$	$*$
2	6	14	150	$*$	$*$	333	$*$	$*$
3	8	16	200	$*$	426	423	$*$	$*$
4	12	23	300	$*$	$*$	652	653	$*$
5	13	31	325	$*$	$*$	762	766	764
6	14	30	350	$*$	$*$	756	760	765
7	20	33	500	$*$	$*$	1017	1029	1028

Figure 3 shows the statistical distribution of solutions and table 1 the best solutions found by the two algorithms. Column LB tabulates the lower bounds found by the cut and column generation algorithm from (SANTOS *et al.*, 2012). The distributions on the two boxplot graphics are based on the concept of **relative distance** in definition 4.1. By this concept we compare the results found by our algorithms with the lower bounds. For the open instances, 5 to 7, our algorithms have reached the lower bounds and it helps to prove the optimality for these instances. In addition, optimal solutions were found for all instances and according to the boxplot in figure 3(b) the statistical distribution of solutions, for these algorithms, are very close to the optimal solutions, less than 7% far from the optimum.

Definition 4.1 (Relative distance) *Given an instance of the HSTP. Lets Z be an arbitrary solution and Z_{best} the best known solution for this instance. The relative distance from Z to Z_{best} is denoted by:*

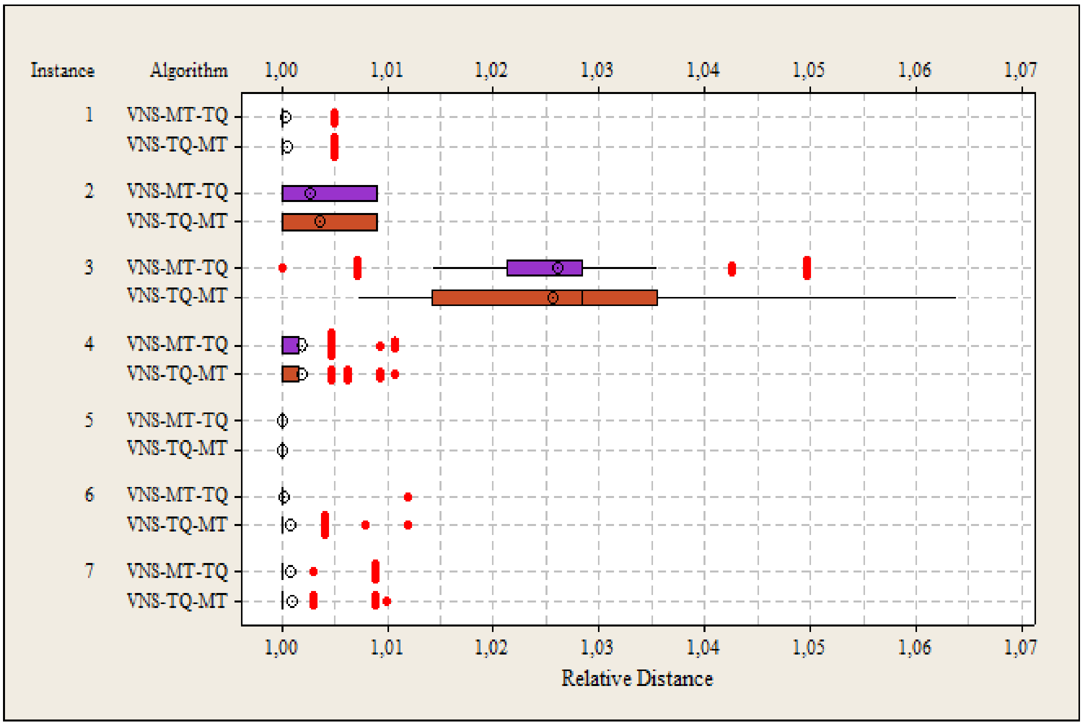
$$rd = \frac{f(Z)}{f(Z_{best})} \tag{4}$$

As additional information, columns TS[1] and IP[2] (table 1) show the best known results

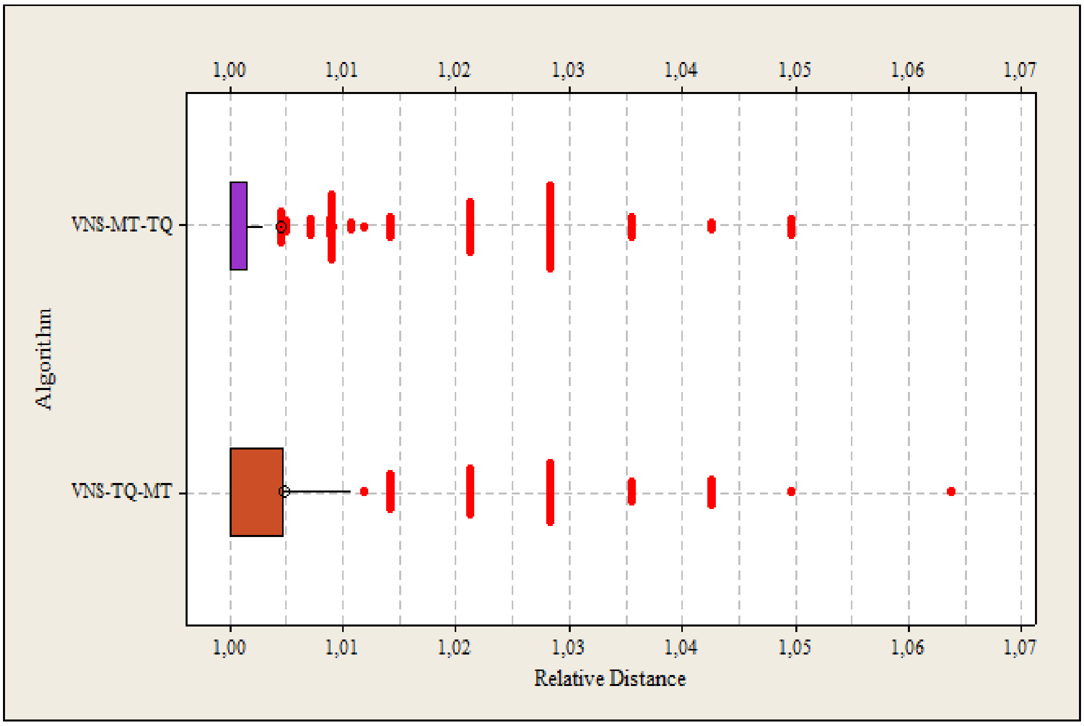
[1] Tabu Search (SANTOS *et al.*, 2005)

[2] Integer Programming (DORNELES *et al.*, 2012)

found in previous studies for these instances. The "∗" symbol in cells of table 1 means that the algorithm was able to reach the lower bound in column LB.



(a) By instance



(b) By algorithm

Figure 3. Statistical distribution of solutions

5. Conclusions and future works

In this paper we have proposed two VNS algorithms to solve the high school timetabling benchmark from (SOUZA *et al.*, 2003). These algorithms have shown to be effective and efficient to solve the problem, as their statistical distribution of solutions are very close to the optimal solutions and global optimum were found for all instances. Furthermore, we believe that the high performance of our algorithms are due to the use of the two neighborhood operators employed and the main contribution of this work was help to prove the optimality for the three open instances.

As future works we intend to test these algorithms with additional set of instances.

6. Acknowledgments

This work was supported by CAPES, Brazil. The authors are grateful for their support and would like to thank the anonymous referees for their helpful comments.

References

BRITO, S. *et al.* A sa-vns approach for the high school timetabling problem. *Electronic Notes in Discrete Mathematics*, Elsevier, v. 39, p. 169–176, 2012.

DELL'AMICO, M.; TRUBIAN, M. Applying tabu search to the job-shop scheduling problem. *Annals of Operations Research*, Springer, v. 41, n. 3, p. 231–252, 1993.

DORNELES, Á. P.; ARAÚJO, O. C. B. d.; BURIOL, L. S. The impact of compactness requirements on the resolution of high school timetabling problem. *Annals of XVI CLAIO - XLIV SBPO - LIA-SGT, Rio de Janeiro*, p. 3336–3347, 2012.

EVEN, S.; ITAI, A.; SHAMIR, A. On the complexity of time table and multi-commodity flow problems. In: IEEE. *Foundations of Computer Science, 1975., 16th Annual Symposium on*. [S.l.], 1975. p. 184–193.

HANSEN, P.; MLADENOVIĆ, N.; PÉREZ, J. M. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, Springer, v. 175, n. 1, p. 367–407, 2010.

LAARHOVEN, P. V.; AARTS, E.; LENSTRA, J. Job shop scheduling by simulated annealing. *Operations research*, JSTOR, p. 113–125, 1992.

LÜ, Z.; HAO, J.; GLOVER, F. Neighborhood analysis: a case study on curriculum-based course timetabling. *Journal of Heuristics*, Springer, v. 17, n. 2, p. 97–118, 2011.

OSOGAMI, T.; IMAI, H. Classification of various neighborhood operations for the nurse scheduling problem. *Algorithms and Computation*, Springer, p. 72–83, 2000.

PILLAY, N. An overview of school timetabling research. *Practice and Theory of Automated Timetabling (PATAT 2010), August 11-13, Queen's University, Belfast, UK*, 2010.

POST, G. *et al.* An xml format for benchmarks in high school timetabling. *Annals of Operations Research*, Springer, v. 194, n. 1, p. 385–397, 2012.

SANTOS, H.; OCHI, L.; SOUZA, M. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. *Journal of Experimental Algorithmics (JEA)*, ACM, v. 10, p. 2–9, 2005.

SANTOS, H. *et al.* Strong bounds with cut and column generation for class-teacher timetabling. *Annals of Operations Research*, Springer, v. 194, n. 1, p. 399, 2012.

SAVINIEC, L.; CONSTANTINO, A. Applying ils algorithm with a new neighborhood operator to solve a large benchmark of the high school timetabling problem. *Annals of XVI CLAIO - XLIV SBPO - LIA-SGT, Rio de Janeiro*, p. 3237–3248, 2012.

SAVINIEC, L. *et al.* Solving the high school timetabling problem to optimality by using ils algorithms. *Proceedings of XLV Brazilian Symposium of Operational Research, Natal, Brazil*, p. 1–12, 2013.

SCHAERF, A. A survey of automated timetabling. *Artificial Intelligence Review*, Springer, v. 13, n. 2, p. 87–127, 1999.

SOUZA, M.; COSTA, F.; GUIMARÃES, I. Um algoritmo evolutivo híbrido para o problema de programação de horários em escolas. *XXII Encontro Nacional de Engenharia de Produção, Curitiba*, 2002.

SOUZA, M.; OCHI, L.; MACULAN, N. A grasp-tabu search algorithm for solving school timetabling problems. *Metaheuristics: Computer Decision-Making. Kluwer Academic Publishers, Boston*, p. 659–672, 2003.