SOLVING THE HIGH SCHOOL TIMETABLING PROBLEM TO OPTIMALITY BY USING ILS ALGORITHMS

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

Haroldo Gambini Santos
Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)
Morro do Cruzeiro – 35400-000 – Ouro Preto – MG – Brazil
haroldo@iceb.ufop.br

ABSTRACT

The high school timetabling is a classical problem and has many combinatorial variations. It is NP-Complete and since the use of exact methods for this problem is restricted, heuristics are usually employed. This paper applies three Iterated Local Search (ILS) algorithms which includes two newly proposed neighborhood operators to heuristically solve a benchmark of the problem from literature. This benchmark has seven instances and the three largest ones are open. The results obtained by our algorithms have shown that these methods are effective and efficient to solve the problem, as they were capable to find optimal solutions for all instances and it helps to prove (using pre-computed lower bounds) the optimality for the open instances.

KEYWORDS. High School Timetabling Problem, Iterated Local Search, Neighborhood Operators.

Main area. Metaheuristics, OR in Education, 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.

The most common methods used at the literature to solve the problem are: Genetic Algorithms (RAGHAVJEE; PILLAY, 2010); 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).

But when a solution of the HSTP is found by a heuristic method, an important question arises about its quality. How much is the solution found far from the optimal solution?

This paper proposes three algorithms based on the ILS metaheuristic to solve the HSTP. The key component of these heuristic methods lies on the use of two powerful neighborhood operators. These algorithms are applied to solve a well known benchmark data set of the high school timetabling problem (SOUZA *et al.*, 2003) and the results are compared with the lower bounds known for these instances (SANTOS *et al.*, 2012).

The paper is organized as follows: Section 2 defines the problem. Section 3 explains the solution approach. Section 4 reports the obtained 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 $f : S \to \mathbb{R}$ 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:*

$$f(Z) = f_A(Z) + f_B(Z) \tag{1}$$

Such that:

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

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

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).

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'

3.6. Neighborhood operators

Neighborhood operators are the key ingredient to develop powerful local search algorithms. In special, some researches as (DELL'AMICO; TRUBIAN, 1993; 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 called MT and TQ are employed. These operators exclude out of the search process a large set of undesirable solutions.

3.6.1. Matching operator

The matching operator (MT), definition 3.5, is based on the assignment problem (AP). It is adapted from the technique that heuristically employs successive AP's to solve the nurse scheduling problem (CONSTANTINO *et al.*, 2009).

Definition 3.5 (Matching operator) *Lets:*

- Z *a timetable of the HSTP;*
- $\Delta P_t \subseteq P$ *the set of teachers who teach for a class* t*;*
- $\Delta Z_t = \{z_{tdh} \in \Delta P_t\}$ *the multiset defined by all lessons of the class* t *at the timetable* Z*;*
- $U_t = \{Y \in P(\Delta Z_t) | Y = \Delta P_t\}$*;*
- $\widehat{Y} = \{(z_{tdh})_i | 1 \le i \le |\Delta P_t|,\ i \in \mathbb{N}\}$ *an indexed family,* $\widehat{Y} \in U_t$*.*

The MT operator is a function $MT : S \to P(S)$ *that assigns for every solution* $Z \in S$*, a neighborhood* $MT(Z) \subseteq S$ *composed by solutions* $Z' \in S$ *obtained from* Z*, by solving an AP formulated on a set* $\widehat{Y} \in U_t$*, given by:*

$$\textbf{Minimize} \quad \sum_{i=1}^{|\widehat{Y}|} \sum_{j=1}^{|\widehat{Y}|} c_{ij} x_{ij}$$

$$\textbf{Subject to} \quad \sum_{i=1}^{|\widehat{Y}|} x_{ij} = 1 \quad (1 \le j \le |\widehat{Y}|) \tag{4}$$

$$\sum_{j=1}^{|\widehat{Y}|} x_{ij} = 1 \quad (1 \le i \le |\widehat{Y}|) \tag{5}$$

$$x_{ij} \in \{0,1\} \quad (1 \le i \le |\widehat{Y}|,\ 1 \le j \le |\widehat{Y}|) \tag{6}$$

Where the cost matrix $C_{|\widehat{Y}| \times |\widehat{Y}|}$ *is computed as follows:*

i) $Z' = Z - \widehat{Y}$*, subtract out of* Z *every cells in* $\widehat{Y}$*;*

ii) c_{ij} *is the objective function value* $f(Z')$ *if teacher at index* i *in* $\widehat{Y}$ *is rescheduled to the period where was teacher from index* j*;*

After solving the AP, Z' *is obtained by rescheduling the teachers from* $\widehat{Y}$*, based on the response variable* x_{ij}*.*

To solve AP's the polynomial time algorithm from (CARPANETO; TOTH, 1987) is applied. 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:

- $\Delta P_t = \{1, 2, 6, 9\}$;
- $\Delta Z_t = \{2, 9, 1, 9, 6\}$;
- $\widehat{Y} = \{\overbrace{2}^{t_3 d_1 h_1}, \overbrace{9}^{t_3 d_1 h_2}, \overbrace{1}^{t_3 d_1 h_3}, \overbrace{6}^{t_3 d_1 h_5}\}$.

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'.

The MT operator has the following property:

Property 3.1 *Given a solution* Z *of the HSTP,* $\forall Z' \in MT(Z)$*,* $f(Z') \le f(Z)$*.*

This property says that when MT is applied on a solution Z, in the worst case, the neighbor Z' will get $f(Z')$ equal to the current solution 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

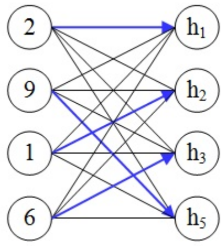
(a) Selection of a set $\widehat{Y}$ at a solution Z, where $f(Z) = 2$

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

3.6.2. Torque operator

The torque operator[1] (TQ), definition 3.6, 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. But when applying moves using DM operator, new clashes between lessons can occur and a_3 constraint is violated. Then, the TQ operator is developed to prevent this disadvantage using the idea of *Kempe chain interchanges* (LÜ *et al.*, 2011).

Definition 3.6 (Torque operator) *Lets: Z a solution of the HSTP; $t \in T$; $d_i, d_j \in D$, $h_i, h_j \in H$ with $d_i \neq d_j$ or $h_i \neq h_j$; and a graph $G = (V, A)$ where:*

- *$V = \{v_t | t \in T\}$ is the vertex set of G formed by ordered pairs of lessons $\langle z_{tdihi}, z_{tdjhj}\rangle$, such that:*

$$\langle z_{tdihi}, z_{tdjhj}\rangle \in V \iff z_{tdihi} \neq z_{tdjhj} \quad (1 \leq t \leq |T|) \tag{7}$$

- *$A = \{\langle u, v\rangle | u \neq v \text{ and } u, v \in V\}$ is the edge set of G and an edge exists between two nodes u and v with attributes $\langle i_u, j_u\rangle$ and $\langle i_v, j_v\rangle$ respectively, if they satisfy the following conditions:*

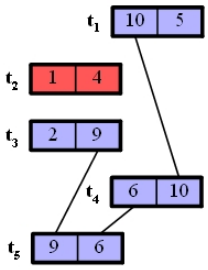
[1]This is an analogy with the system of two parallel forces that act over a body and tend to cause rotation.

$$i_u = j_v \text{ } or \text{ } j_u = i_v \tag{8}$$
$$\forall \langle u, v \rangle, \langle u, k \rangle \in A \text{ } if \text{ } i_u = j_v \text{ } and \text{ } i_u = j_k \text{ } then \text{ } v = k \tag{9}$$
$$\forall \langle u, v \rangle, \langle u, k \rangle \in A \text{ } if \text{ } j_u = i_v \text{ } and \text{ } j_u = i_k \text{ } then \text{ } v = k \tag{10}$$

The TQ operator is a function $TQ : S \rightarrow P(S)$ that assigns to every solution $Z \in S$, a neighborhood $TQ(Z) \subseteq S$, where $Z' \in TQ(Z)$ is obtained from Z by swapping each pair of lessons in a connected component of G.

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 a_3 constraint



(b) Graph G for $d_i = d_j = d_1$, $h_i = h_1$ 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

At the definition 3.6, condition 7 accepts in V, only nodes in which attributes i and j are different. Condition 8 accepts in A, only the edges that connect nodes in which their opposite attributes have equal values and conditions 9 and 10 impose that when swapping lessons in a connected component, no more than one lesson of a teacher will be moved from one period to another.

The TQ operator has the following property:

Property 3.2 *Given a solution Z of the HSTP, $\forall Z' \in TQ(Z)$, $f_{a_3}(Z') \leq f_{a_3}(Z)$.*

This property says that when TQ is applied on a solution Z, the number of clashes β_{a_3}, at the new neighbor Z', will not be augmented. Figure 2 illustrates this operator.

3.7. ILS based algorithms to the HSTP

The proposed approach is composed by three algorithms based on the ILS metaheuristic (LOURENÇO *et al.*, 2003).

ILS-TQ: this ILS (algorithm 3.4) incorporates the first improvement technique (algorithm 3.2) with TQ operator, as local search heuristic.

IMLS-MT-TQ: this algorithm (3.5) uses the idea of "heuristic composition", two heuristics run in sequence and the second starts from the local minimum found by the first one[2]. The

[2] We have named it as Iterated Multi Local Search (IMLS).

two local search heuristics are based on the first improvement technique (algorithm 3.2). The first heuristic explores the neighborhood $MT(Z)$ randomly and the second heuristic explores the neighborhood $TQ(Z)$ in a deterministic way.

IMLS-TQ-MT: this algorithm is the previous one with the heuristic sequence in reverse order.

These three 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
```

Algorithm 3.4 ILS-TQ algorithm

```
ILS-TQ(Z_0, t_max)
Z = FIH(Z_0, TQ)                        // local search
Z* = Z                                  // best solution found
NotImproved = 0
repeat
    Z = N-RANDOM-PERTURBATION(Z, TQ, 1)
    Z = FIH(Z, TQ)                      // local search
    if f(Z) < f(Z*) then
        NotImproved = 0
    else
        NotImproved = NotImproved + 1
    if f(Z) ≤ f(Z*) then                // acceptance criterion
        Z* = Z
    if NotImproved ≥ 3 then             // if no improvement after three iterations
        Z = Z*                          // return to Z*
        NotImproved = 0
    t = CPUTIME()
until (t > t_max or f(Z) = 0)
return Z*
```

Algorithm 3.5 IMLS-MT-TQ algorithm

```
IMLS-MT-TQ(Z_0, t_max)
Z = FIH-MT-TQ(Z_0)                      // composite local search
Z* = Z                                  // best solution found
NotImproved = 0
repeat
    Z = N-RANDOM-PERTURBATION(Z, TQ, 1)
    Z = FIH-MT-TQ(Z)                    // composite local search
    if f(Z) < f(Z*) then
        NotImproved = 0
    else
        NotImproved = NotImproved + 1
    if f(Z) ≤ f(Z*) then                // acceptance criterion
        Z* = Z
    if NotImproved ≥ 3 then             // if no improvement after three iterations
        Z = Z*                          // return to Z*
        NotImproved = 0
    t = CPUTIME()
until (t > t_max or f(Z) = 0)
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)[3]. Table 1 presents the main characteristics of the instances. Column sr shows the sparseness ratio (equation 11) for each instance.

$$sr = 1 - \frac{\sum_{i=1}^{|L|} \theta_i + |U|}{|P| \times |D| \times |H|} \tag{11}$$

This expression measures how much a problem instance is constrained regarding to the number of teachers' non-available periods. Higher values indicate flexible problems while lower values indicate more restrictive problems. This benchmark has seven instances and for the three largest ones, the optimal solutions are not known, however, lower bounds were computed using an extended Integer Linear Programming Formulation (SANTOS *et al.*, 2012).

Table 1. Characteristics of instances

Instance	$\|T\|$	$\|P\|$	$\|D\|$	$\|H\|$	Lessons	Double Lessons	sr
1	3	8	5	5	75	21	0,43
2	6	14	5	5	150	29	0,5
3	8	16	5	5	200	4	0,3
4	12	23	5	5	300	41*	0,18
5	13	31	5	5	325	71	0,58
6	14	30	5	5	350	63	0,52
7	20	33	5	5	500	84	0,39

* Researchers have printed 66 to this value in previous papers, but according to the instance files from http://labic.ic.uff.br/Instance/ this value is 41.

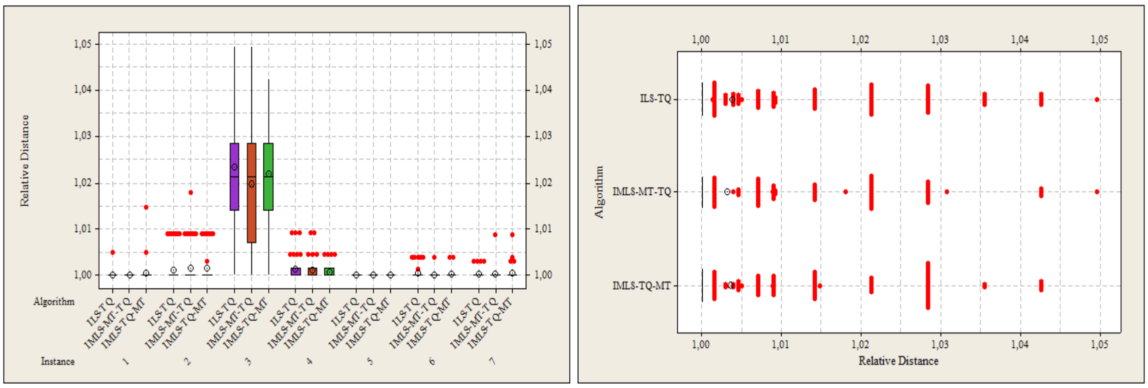
The proposed approach was coded using MS Visual Basic 6. The experiment was performed on Windows Server 2008-R2 running on the KVM virtual machine set to work with 30GB of RAM and 50 cores of a server with 4 CPU Intel Xeon E7-4860 (24MB of Cache - 2.26 GHz) with Linux CentOS 6 operating system. In this experiment 50 tests of 900 seconds were carried out for each instance. The whole experiment was performed in three 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 2. Best results

Instance	ILS-TQ	IMLS-MT-TQ	IMLS-TQ-MT	LB	TS	IP
1	$*$	$*$	$*$	202	$*$	$*$
2	$*$	$*$	$*$	333	$*$	$*$
3	$*$	$*$	$*$	423	$*$	$*$
4	$*$	$*$	$*$	652	653	$*$
5	$*$	$*$	$*$	762	766	764
6	$*$	$*$	$*$	756	760	765
7	$*$	$*$	$*$	1017	1029	1028

Figure 3 shows the statistical distribution of the solutions and table 2 the best solutions found by the three algorithms. Column LB shows 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, our algorithms have found optimal solutions for all instances and according to the boxplot in figure

[3] This benchmark can be downloaded from http://labic.ic.uff.br/Instance/.



(a) By instance (b) By algorithm

Figure 3. Statistical distribution of solutions

3(b), they have statistical distribution of solutions that are very close to the optimal solutions, less than 5% far from the optimum.

As additional information, columns TS[4] and IP[5] (table 2) show the best known results found in previous studies for these instances. The "$*$" symbol in cells of table 2 means that the algorithm was able to reach the lower bound in column LB.

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:*

$$rd = \frac{f(Z)}{f(Z_{best})} \tag{12}$$

5. Conclusions and future works

In this paper we have proposed three ILS 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 they were capable to find optimal solutions for all instances and the statistical distribution of solutions are very close to the optimal solutions. In addition, it has helped to prove the optimality for the three open instances.

The main contribution of this paper are twofold, it demonstrates the robustness of two distinct approaches for HSTP, our heuristic methods and the Integer Linear Programming Formulation from (SANTOS *et al.*, 2012).

As future works we intend to test our 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.

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

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

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.

CARPANETO, G.; TOTH, P. Primal-dual algorithms for the assignment problem. *Discrete applied mathematics*, Elsevier, v. 18, n. 2, p. 137–153, 1987.

CONSTANTINO, A. A. *et al.* Um algoritmo heurístico para otimização do problema de escalonamento de enfermeiros. *Anais: XLI SBPO, Porto Seguro*, p. 2783–2794, 2009.

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.

LOURENÇO, H.; MARTIN, O.; STÜTZLE, T. Iterated local search. *Handbook of metaheuristics*, Springer, p. 320–353, 2003.

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.

RAGHAVJEE, R.; PILLAY, N. Using genetic algorithms to solve the south african school timetabling problem. In: IEEE. *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*. [S.l.], 2010. p. 286–292.

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.

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

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.