UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ÁRTON PEREIRA DORNELES

A Matheuristic Approach for solving the High School Timetabling Problem

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dra. Luciana Salete Buriol
Coadvisor: Prof. Dr. Olinto C. B. de Araújo

Porto Alegre
2015

CIP — CATALOGING-IN-PUBLICATION

Dorneles, Árton Pereira

A Matheuristic Approach for solving the High School Timetabling Problem / Árton Pereira Dorneles. – Porto Alegre: PPGC da UFRGS, 2015.

149 f.: il.

This research was partially supported by CAPES and by the *Programa Petrobras de Formação de Recursos Humanos* (PRH PB-217).

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2015. Advisor: Luciana Salete Buriol; Coadvisor: Olinto C. B. de Araújo.

1. High school timetabling. 2. Mathematical programming. 3. Meta-heuristics. 4. Matheuristics. 5. Fix-and-optimize. I. Buriol, Luciana Salete. II. Araújo, Olinto C. B. de. III. Título.

"To attain knowledge, add things every day.
To attain wisdom, remove things every day."

— Lao-Tsé

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisors Luciana Buriol, Olinto Araújo and Dario Landa-Silva for their friendship, guidance and unconditional support to my research. Beside my advisors, I would like to thank my thesis committee for their feedback and helpful comments on my work.

Also, I would like to thank the Coordination for the Improvement of Higher Level Personnel (CAPES) and Petrobras S.A., Brazil for supporting my research. My sincere thanks also goes to all school managers that kindly shared their time and practical knowledge of timetabling. I specially thank my friend, Vinicius Dani, that helped me to collect the datasets used in this thesis.

Last but not the least, I would like to thank the many good friends I have made during last five years, especially my labmates from Porto Alegre and Nottingham, as well as my family, and my beloved partner Victória Simonetti.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CCG	Cut and Column Generation
CD	Class Decomposition
DD	Day Decomposition
GA	Genetic Algorithm
GHSTP	Generalized High School Timetabling Problem
HSTP^{+}	Extended High School Timetabling Problem
HSTP	High School Timetabling Problem
ILS	Iterated Local Search
IPCG	Integer Pricing Column Generation
ITC	International Timetabling Competition
LAHC	Late-Acceptance Hill Climbing
LS	Local Search
MIP	Mixed Integer Programming
MP	Master Problem
PATAT	International Conference on the Practice and Theory of Automated Timetabling.
RINS	Relaxation Induced Neighborhood Search
RMP	Restricted Master Problem
RPCG	Relaxed Pricing Column Generation
SA	Simulated Annealing
STL	Subproblem Time Limit
STP	School Timetabling Problem
TD	Teacher Decomposition
TS	Tabu Search
VND	Variable Neighborhood Descent

VNS	Variable Neighborhood Search
XHSTT	XML archive for High School TimeTabling
XML	eXtensible Markup Language

ABSTRACT

The school timetabling is a classic optimization problem that has been extensively studied due to its practical and theoretical importance. It consists in scheduling a set of class-teacher meetings in a prefixed period of time, satisfying requirements of different types. Given the combinatorial nature of this problem, solving medium and large instances of timetabling to optimality is a challenging task. When resources are tight, it is often difficult to find even a feasible solution. Several techniques have been developed in the scientific literature to tackle the high school timetabling problem, however, robust solvers do not exist yet. Since the use of exact methods, such as mathematical programming techniques, is considered impracticable to solve large real world instances, metaheuristics and hybrid metaheuristics are the most used solution approaches. In this research we develop techniques that combine mathematical programming and heuristics, so-called matheuristics, to solve efficiently and in a robust way some variants of the high school timetabling problem. Although we pay special attention to problems arising in Brazilian institutions, the proposed methods can also be applied to problems from different countries.

Keywords: High school timetabling. mathematical programming. meta-heuristics. matheuristics. fix-and-optimize.

Uma abordagem mateheurística para resolver o problema de geração de quadros de horários escolares do ensino médio.

RESUMO

A geração de quadros de horários escolares é um problema clássico de otimização que tem sido largamente estudado devido a sua importâncias prática e teórica. O problema consiste em alocar um conjunto de aulas entre professor-turma em períodos de tempo pré-determinados, satisfazendo diferentes tipos de requisitos. Devido a natureza combinatória do problema, a resolução de instâncias médias e grandes torna-se uma tarefa desafiadora. Quando recursos são escassos, mesmo uma solução factível pode ser difícil de ser encontrada. Várias técnicas tem sido propostas na literatura científica para resolver o problema de geração de quadros de horários escolares, no entanto, métodos robustos ainda não existem. Visto que o uso de métodos exatos, como por exemplo, técnicas de programação matemática, não podem ser utilizados na prática, para resolver instâncias grandes da realidade, meta-heurísticas e meta-heurísticas híbridas são usadas com frequência como abordagens de resolução. Nesta pequisa, são desenvolvidas técnicas que combinam programação matemática e heurísticas, denominadas mateheurísticas, para resolver de maneira eficiente e robusta algumas variações de problemas de geração de quadros de horários escolares. Embora neste trabalho sejam abordados problemas encontrados no contexto de instituições brasileiras, os métodos propostos também podem ser aplicados em problemas similares oriundo de outros países.

Palavras-chave: geração de quadros de horários escolares . programação matemática . meta-heurísticas . mateheurísticas . fixar-e-otimizar .

CONTENTS

1 INTRODUCTION

A common task to all educational institutions is to provide an assignment of classes that combines teachers, students, rooms and periods (or timeslots) to achieve a feasible timetable, satisfying personal, pedagogical and organizational requirements.

Usually, requirements are separated into *hard* and *soft* ones. By hard requirements we mean those that must be satisfied, while soft requirements are those that may be violated, but should be satisfied whenever possible. Soft requirements can have different levels of importance and are often conflicting with each other such that it may be impossible to satisfy all of them at the same time. Typically, the quality of a solution is associated directly to the satisfaction of soft requirements. The more soft requirements are satisfied, the better a solution is considered.

Quality is a critical solution attribute because, once the timetable is established, it will determine the use of physical resources and the daily routine of hundreds, possibly thousands, of people for a long period that usually is about one year. Due to repetition, even minor issues can turn into major problems in the course of time, affecting directly the quality of teachers' work, and the learning and health of students. Concerning the last issue, many studies agree that carrying overloaded school bags can lead to several health risks as back pain, fallings and, at long-term, irreversible postural changes (CHANSIRINUKOR et al., 2001; KISTNER; FIEBERT; ROACH, 2012; KISTNER et al., 2013). Although this is rarely considered in the construction of a schedule, when a timetabling solution allows two or more subjects with heavy books on the same day it will contribute to students to have these sort of injuries. Therefore, a high quality timetabling is essential for a proper operation of any educational institution.

In spite of its relevance, in many institutions this problem is solved manually in a process that can take weeks, even if carried out by an expert timetabler (MOURA; SCARAFICCI, 2010). Due to its difficulty, the automation of this task is becoming more common, and nowadays it is mandatory in medium and large institutions.

Educational timetabling problems have many variants proposed in the literature, and the set of objectives and requirements depends mostly on the context of the application, the institution and the place where it is located (POST et al., 2011; DREXL; SALEWSKI, 1997). Although the problem diversity, educational timetabling problems are commonly comprised in three classes: school timetabling, course timetabling, and examination timetabling (SCHAERF, 1999b). In both school and course timetabling the aim is to build a weekly schedule. However, in the school timetabling a set of classes must be assigned to timeslots, whereas in course timetabling a set of university courses must be scheduled avoiding overlaps of course lectures that have common students. Finally, in the examination problem a set of exams must be spread in a time horizon avoiding overlaps for the students.

In our present study we focus on the school timetabling problem. This problem first appeared in the scientific literature in the 60's (GOTLIEB, 1962) and since then it has gained increasing attention. The most basic variant of the problem is to schedule a set of class-teacher events (or meetings) in such a way that no teacher (nor class) is required in more than one lesson at a timepoint. This basic problem can be solved in polynomial time by a min-cost network flow algorithm (WERRA, 1971). However, in real-world applications, teachers can be unavailable in some periods. If this constraint is taken into account, the resulting timetabling problem is NP-complete (EVEN; ITAI; SHAMIR, 1975).

In fact, the most real-world timetabling problems come to light as combinatorial optimization problems that fall in the NP-Hard class. For this reason, many researchers around the world have investigated these problems and several different techniques have been developed. The most active research groups are located in the United Kingdom, Brazil, Italy, Canada, Denmark, Germany, Greece, Italy, Netherlands and Australia. Although these groups share similar interests, they have focused more on solving specific problem variants from their country. As a result, most of the works reported in the literature consider application-dependent (often unavailable) test cases, what makes it difficult to compare results among the different solution approaches (SCHAERF; GASPERO, 2001). In Brazil, for example, it is common that a teacher works in more than one school, having several jobs. In order to allow this possibility, it is important to schedule the lessons in each school in the minimal number of days. Furthermore, it is required to avoid idle periods between lessons in a teachers' schedule and satisfy pedagogical demands or personal preferences, like a teacher requesting double lessons. This set of requirements defines a problem arising in a typical Brazilian school and, not necessarily, reflects exactly the same problem found in other countries.

As an attempt to overcome these issues, along the last editions of international conferences on the Practice and Theory of Automated Timetabling (PATAT), a group of high school timetabling researchers has developed a XML based format, called XHSTT, to express problems from different countries in a unified way (POST et al., 2010; POST et al., 2011). Despite the verboseness of the XHSTT format, it has gained widespread acceptance by the research community and, recently, its use was promoted in the Third International Timetabling Competition (ITC2011) (POST et al., 2013). At the time of writing this chapter, there were around 50 instances available on the website dedicated to XHSTT format, although some of them are deprecated (Benchmarking Project, 2015). Problems that can be represented int the XHSTT format are normally refered as Generalized High School Timetabling Problem, hereafter denoted as GHSTP.

In this research we develop techniques that combine mathematical programming and (meta)heuristics, so called matheuristics, to solve efficiently and in a robust way two variants of the high school timetabling problem further defined as HSTP and HSTP$^+$. Although we pay special attention to problems arising in Brazilian institutions, the proposed methods can be generalized for similar problems originated from other countries.

1.1 Research Contributions

The research performed in this thesis have led to the following major contributions:

- An initial investigation evaluated the performance of state-of-the-art MIP solvers applied in instances of the HSTP. The experimental results demonstrated empirically that MIP solvers can be used for providing high quality solutions for small instances of the problem. The study also revealed that among the soft requirements, the idle times constraint is the one that most aggregates complexity into the resolution process. In addition, a novel MIP model that is better suitable for solving small instances was proposed and compared with the previously proposed model from the literature. During the experimental evaluations, one new optimal solution and two new best computed solutions were found for a well-known set of instances of the HSTP.

- A novel approach was proposed for solving the HSTP by exploring class, teacher and day decompositions through a fix-and-optimize heuristic combined with a variable neighborhood descent method. In addition, a simple construction procedure was proposed for quickly generating feasible initial solutions. Experimental results demonstrated that this novel approach is able to provide high quality feasible solutions in a smaller computational time when compared with results obtained by a state-of-the-art MIP solver. Furthermore, by applying the proposed approach, new best known solutions were found for several instances quoted in the literature. Among these new results, better solutions were found to four out of five HSTP instances from the first round of the Third International Timetabling Competition (held in 2011).

- A column generation approach was proposed for producing lower bounds to the HSTP by using a novel multicommodity flow representation. In comparison with the previous state-of-the-art approach, the experimental results show that the proposed approach is able to produce the same tight lower bounds, albeit with two significant advantages: i) the method is simpler; ii) and it is five times faster on average. During the experimental evaluations, best known lower bounds were found for all instances considered in the first round of the Third International Timetabling Competition.

- A new high School Timetabling Problem referred as HSTP$^+$ originated from 33 real-world Brazilian instances is introduced. The HSTP$^+$ is defined formally through a MIP formulation and a XHSTT model. In addition, the fix-and-optimize algorithm was adapted and evaluated in comparison with a state-of-the-art MIP solver, as well as two state-of-the-art local search based solvers designed for solving the GHSTP. The experimental evaluation, supported by statistical analysis, provided strong evidence that the fix-and-optimize approach is also suitable for solving the HSTP$^+$, outperforming the compared methods.

1.2 Publications

Along this research, a number of papers have been published in peer-reviewed conferences and journals. Additionally, we present in the list below, papers that either were submitted or are in preparation for submission:

1. DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. The impact of compactness requirements on the resolution of high school timetabling problem. In: **SIMPOSIO BRASILEIRO DE PESQUISA OPERACIONAL**. 44, 2012. **Anais...** Rio de Janeiro, Brazil: Sociedade Brasileira de Pesquisa Operacional, 2012. p. 3336–3347.
2. DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. A fix-and-optimize heuristic for the high school timetabling problem. **Computers & Operations Research**, Elsevier, Oxford, England, v. 52, p. 29–38, 2014.
3. DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. A column generation approach to the high school timetabling modeled as a multicommodity flow problem. **European Journal of Operational Research**, Elsevier, Berlin, Germany, 2015. (Submitted).
4. DORNELES, Á. P.; ARAÚJO, O. C. B.; LANDA-SILVA, D.; BURIOL, L. S. Solving large high school timetabling problems in Brazil by using fix-and-optimize and local branching. **European Journal of Operational Research**, Elsevier, Berlin, Germany, 2016. (Submitted).

Each one of these papers is presented as a chapter in this dissertation.

1.3 Outline of the Thesis

This dissertation is organized in seven chapters. Chapter 2 presents a literature review on timetabling resolution, state-of-the-art approaches and methods of combinatorial optimization. Chapter 3 presents an initial investigation to the first problem tackled in this study (HSTP), where we formally define the problem and compare mixed-integer programming formulations through empirical experiments on a well-known set of instances. Chapter 4 presents a fix-and-optimize heuristic and experimental results for it considering synthetic and real-world instances used in the Third International Timetabling Competition. Chapter 5 presents a column generation approach for producing tight lower bounds for HSTP. Chapter 6 introduces a new problem (HSTP$^+$) and describes a new benchmark intance set composed by several real-world instances. The models and methods presented in previous chapter are expanded in this chapter for tackling the new problem and several experiments are carried out in order to strenghten the conclusions we draw in previous chapters. Finally, Chapter 7 presents our major conclusions, the limitations of this research, and some perspectives for future work.

2 LITERATURE REVIEW

In this chapter we present a brief review on combinatorial optimization methods. Next, we describe main approaches on timetabling resolution and, finally, we discuss the literature.

2.1 Combinatorial Optimization Methods

Combinatorial optimization problems are applied to several real-world applications, e.g., assignment, networking, routing, scheduling, timetabling, cutting, packing, etc. However, these kind of problems are often NP-Hard and challenging because no efficient algorithm is known for solving them. The available methods for solving this class of problems can be split into two categories: *exact* and *approximate* methods. Exact methods are able to find a solution with optimality guarantee. A class of exact methods that had obtained significant success are the Integer Programming based methods. Some methods in this class are: Branch-And-Bound, Branch-And-Cut and Branch-And-Price (NEMHAUSER; WOLSEY, 1988). However, when applied on large or complex instances, it is well-known that exact methods might be very time-consuming. Usually, in this case, researchers sacrifice the optimality to achieve good/feasible solutions in polynomial time, resorting to approximate methods.

When an approximate method is required, heuristics and meta-heuristics are often used. According to Blum and Roli (2003) there are two main classes of meta-heuristic methods: single-solution and population-based methods. The first class comprises local-search based algorithms like Simulated Annealing (KIRKPATRICK; GELATT; VECCHI, 1983), Tabu Search (GLOVER, 1986), GRASP (FEO; RESENDE, 1989), Variable Neighborhood Search (MLADENOVIĆ; HANSEN, 1997), Iterated Local Search (LOURENÇO; MARTIN; STÜTZLE, 2003) and Late Acceptance Hill-Climbing (BURKE; BYKOV, 2012). The second class deals with multiple solutions during the search, often, combining them. Some representative examples are Genetic Algorithms (HOLLAND, 1975) and Scatter Search (MARTÍ; LAGUNA; GLOVER, 2006).

Despite the fact that it is very common to combine metaheuristics in a hybrid method, recent approaches combine exact and heuristic methods to exploit simultaneously the advantages of these methods (DUMITRESCU; STÜTZLE, 2003; PUCHINGER; RAIDL, 2005; RAIDL, 2006; JOURDAN; BASSEUR; TALBI, 2009). According to Puchinger and Raidl (2005) there are two ways to combine these methods: *collaborative* and *integrative* combinations.

In a collaborative combination the methods exchange information, but none is contained into another. They may be arranged to execute sequentially, in parallel or in intertwined mode. Whereas in an integrative combination, an exact method is embedded within a heuristic method or vice-versa.

Specially, in this study, we are interested in a class of methods resulting from the combination between mathematical programming and meta-heuristics called *matheuristics*. This class have been successfully applied to solve several real-world optimization problems (MANIEZZO; STÜTZLE; VOSS, 2009). Some of them are Local Branching, Relaxation Induced Neighborhood Search and Fix-And-Optimize.

2.1.1 Local Branching

Local Branching is a method proposed by Fischetti and Lodi (2003) to solve general MIP problems that are composed mainly by binary variables. The method works similarly as local search but the neighborhoods are generated by changing a MIP model through the introduction of invalid linear inequalities called *local branching cuts*. Each neighborhood defines a subproblem that is solved by using any general purpose MIP solver available. Although the method is designed to provide exact solutions, the main goal of local branching is to achieve good solutions in early stages of the search. Thus, it can be used as a heuristic method when short time limit is provided.

Let us consider the general MIP problem at following to explain the local branching framework:

$$
\begin{aligned}
\text{Minimize } & c^T x \\
\text{Subject to } & Ax \geq b \\
& x_j \geq 0 && \forall j \in G, x \text{ integer} \\
& x_j \geq 0 && \forall j \in C \\
& x_j \in \{0,1\} && \forall j \in B \neq \emptyset
\end{aligned}
$$

The index set of variables is split into three sets G, C and B defining, respectively, integer, continuous and binary variables. Given a feasible *reference solution* $\bar{x}$ and a parameter $k \in \mathbb{N}^*$ we can define a k_{OPT} neighborhood $\mathcal{N}(\bar{x}, k)$ of $\bar{x}$ comprising the set of feasible solutions of the MIP problem that satisfies the additional *local branching* constraint:

$$
\Delta(x,\bar{x}) := \sum_{j \in S}(1 - x_j) + \sum_{j \in B \setminus S} x_j \leq k \qquad \text{where } S = \{j \in B | \bar{x} = 1\} \tag{2.1}
$$

Terms in the left-side of (2.1) counts the number of binary variables flipping from 1 to 0 and from 0 to 1 regarding the reference solution $\bar{x}$. In other words, $\Delta(x,\bar{x})$ represents the *Hamming Distance* between x and $\bar{x}$.

The local branching constraint is used within an enumerative scheme for solving MIP subproblems. In fact, considering an incumbent solution $\bar{x}$, the solution space of the current branching node can be partitioned using the following disjunctive constraints:

$$
\Delta(x,\bar{x}) \leq k \text{ (left-branch)} \qquad \text{or} \qquad \Delta(x,\bar{x}) \geq k+1 \text{ (right-branch)} \tag{2.2}
$$

The value of k must be chosen in such way to make the left-branch neighborhood sufficiently small to be explored quickly and large enough to contain improved solutions. Typically, this value is strongly related to the problem instance size, the problem formulation, and the performance of MIP solver used.

The basic overall local branching algorithm is presented in the pseudo-code of Figure 2.1. Function `localBranching()` receives as input an initial feasible solution x' and the parameter k. In the inner loop several subproblems are generated by adding local branching constraints while the reference solution $\bar{x}$ is improved. The function `solveMIP()` solve the current MIP subproblem to optimality using an objective cutoff based on the objective value of $\bar{x}$. If it is able to find a better solution than $\bar{x}$ a new solution is returned, otherwise $\bar{x}$ is returned.

When the solution cannot be improved anymore in the main loop, then the remaining problem is solved at line 8. This last resolution is possibly very difficult since it will provide the optimal solution or state the optimality of the reference solution. In the last line, the best (optimal) solution found is finally returned.

Figure 2.1 – Pseudo-code of the basic local branching method

Algorithm `localBranching` (x', k)

1: **repeat**
2: $\quad \bar{x} \leftarrow x'$
3: $\quad$ add local branching constraint $\Delta(x, \bar{x}) \leq k$
4: $\quad x' \leftarrow$ `solveMIP(`$\bar{x}$`)`
5: $\quad$ remove previously added local branching constraint $\Delta(x, \bar{x}) \leq k$
6: $\quad$ add local branching constraint $\Delta(x, \bar{x}) \geq k + 1$
7: **until** x' is not better than $\bar{x}$
8: $\bar{x} \leftarrow$ `solveMIP(`$\bar{x}$`)`
9: **return** $\bar{x}$

Source: Figure created by author.

Fischetti and Lodi (2003) proposed several extensions in order to improve the performance of this basic local branching scheme: (i) imposing a node time limit to left-branchings; (ii) introducing diversification and intensification strategies that change the value of k systematically along the search to overcome often very time-demanding subproblems; (iii) proposing a method to manage initial infeasible solutions, and (iv) adapting the branching procedure to work with general integer variables. In Hansen, MladenoviĆ and UroŠeviĆ (2006) it was proposed a strategy that embed local branching as a local search procedure within a VNS procedure whose neighborhoods are arranged sequentially from small to large values of k.

2.1.2 Relaxation Induced Neighborhood Search

The Relaxation Induced Neighborhood Search (RINS) was introduced by Danna and Rothberg (2005) as a method to improve the incumbent solution $\bar{x}$ of a general MIP problem P within a branch-and-bound scheme. The basic idea is to build a subproblem smaller then P by exploiting information of the linear programming (LP) solutions of the branch-and-bound tree nodes. Specifically, the subproblem corresponds to the neighborhood of $\bar{x}$ which is created by fixing variables having the same values in the incumbent solution and in the relaxed solution.

The overall method is outlined in the Figure 2.2. The function `RINS` receives three input parameters: an incumbent feasible solution $\bar{x}$ of P, a relaxed solution $\hat{x}$ of P whose objective value is better than the objective value of $\bar{x}$, and a time limit value `TL`.

Figure 2.2 – Outline of the RINS algorithm

Algorithm `RINS` ($\bar{x}, \hat{x}$, `TL`)

1: Fix the variables with the same value in both $\bar{x}$ and $\hat{x}$
2: Add a cut-off based on the objective value of $\bar{x}$
3: Solve the resulting MIP subproblem within the time limit `TL`
4: Return an improved solution if found.

Source: Figure created by author.

Note that the RINS method can fail to obtain an improved solution if either the subproblem is infeasible or is not able to find a solution within the imposed time limit. This last issue usually happens when the resulting subproblem is large and/or difficult to solve.

Since each node of the branch-and-bound tree provides different relaxed LP solutions, the RINS method can be invoked several times in order to explore different neighborhoods.

2.1.3 Fix-And-Optimize

The fix-and-optimize heuristic was proposed independently by Gintner, Kliewer and Suhl (2005) and by Pochet and Wolsey (2006). In the latter, the method was called *exchange*, designed to improve the *relax-and-fix* heuristic (WOLSEY, 1998). However, the name fix-and-optimize used by the former was adopted in the literature. The fix-and-optimize heuristic iteratively decomposes a MIP problem into smaller subproblems. In each iteration of the algorithm, a decomposition process is applied with the aim of fixing most of the decision variables at their value in the current solution. Since the resulting subproblem is composed only by a small group of "free" variables to be optimized, each subproblem can be solved relatively fast by a MIP solver, when compared with the full model. The solution obtained in each iteration becomes the current solution in case it improves the objective value. In further iterations of the algorithm, a different group of variables is systematically selected to be optimized. This process is repeated until a termination condition is satisfied.

2.2 School Timetabling Decisions

There are two basic decisions to tackle when building a timetabling:

- **Resource Assignment**: consists in assigning human and physical resources as teachers, classes and rooms to events.
- **Timeslot Assignment**: consists in assigning a given amount of timeslots to each event.

While the timeslot assignment is a mandatory decision in every school, in many ones the resource assignment is previously done by the school board. Particularly in this study we focus in the development of techniques for solving timetabling problems in which the resource assignement is previously provided.

2.3 Timetabling Approaches

Due to its great practical importance, the timetabling problem has been intensively investigated since 1960 (GOTLIEB, 1962). The first computational attempts in solving the problem were inspired in the human way of solving it. This was usually done through constructive methods combined with backtracking procedures (PAPOULIAS, 1980; SCHMIDT; STRÖHLEIN, 1980; GANS, 1981;

JUNGINGER, 1986). In the beginning, the challenge was to find a feasible solution. Afterwards, variants of the problem were modeled by Integer Programming (TILLETT, 1975; TRIPATHY, 1984), but only small instances could be solved to optimality. Moreover, reduced instances to graph coloring and network flow problems were solved by techniques designed for these problems (NEUFELD; TARTAR, 1974; OSTERMANN; WERRA, 1982; WERRA, 1985). In the late 90's, Schaerf (1999b) classified educational timetabling problems into three groups: course timetabling, examination timetabling, and school timetabling. Each one of these groups has several variants of the problem proposed in the literature. We refer the reader to the survey of Schaerf (1999b) which presents in a comprehensive structure the main variants of the timetabling problem, its formulations and solution approaches. Since the early 90's, metaheuristics have been successfully applied to timetabling problems. Among them are Simulated Annealing (ABRAMSON, 1991; COLORNI; DORIGO, 1998; AVELLA et al., 2007; ZHANG et al., 2010), Tabu Search (COSTA, 1994; SCHAERF, 1999a; SANTOS; OCHI; SOUZA, 2005), and Genetic Algorithms (CALDEIRA; ROSA, 1997). Other advanced techniques used are Hyper-Heuristics (BURKE; KENDALL; SOUBEIGA, 2003), Column Generation (PAPOUTSIS C. VALOUXIS, 2003), and Constraint Programming (VALOUXIS; HOUSOS, 2003; MARTE, 2007).

We refer the reader to a recent survey presented by Pillay (2014) where an wide review is made and comprises several works on the high school timetabling problem. In the next sections, we focus on the literature regarding school timetabling in Brazil, as well as the literature related to the XHSTT format and matheuristic approaches applied on timetabling.

2.3.1 Brazilian High School Timetabling

The most noteworthy problem variant regarding school timetabling in Brazilian institutions was first defined by Souza and Maculan (2000). This problem comprises the most common requirements found in a typical Brazilian school. Here we denote this problem as HSTP. Souza and Maculan (2000) presented a MIP formulation for the HSTP, as well as an instance set that became a basic testbed used until nowadays. Their computational results for HSTP had shown that to solve the testbed instances with a general purpose MIP solver was impracticable.

Thus, Souza, Ochi and Maculan (2004) proposed an hybrid meta-heuristic (GTS-II) method to solve the testbed instances. The GTS-II uses a greedy randomized constructive heuristic to build an initial solution that later is refined by a Tabu Search. Since the Tabu Search also includes infeasible solutions in the search space, it is equipped with a procedure called Intraclasses-Interclasses that is invoked eventually in an attempt to retrieve the current solution feasibility. In this study, GTS-II is compared only with some of its variants.

In the work of Santos, Ochi and Souza (2005) a Tabu Search with diversification strategies (TSTR) is proposed to solve HSTP. Their experiments show that TSTR significantly outperforms GTS-II. In addition, the authors show empirically that the proposed diversification strategy can improve the robustness of TSTR. Another attempt to solve the HSTP, using graph coloring, is proposed by Bello, Rangel and Boeres (2008), however the results obtained by this approach were not compared against the state-of-the-art methods. More recently, Santos et al. (2012) proposed and applied a cut and column generation algorithm providing, for the first time, strong lower bounds for HSTP instances. That work is considered a landmark because it established a reliable base to evaluate heuristically generated solutions.

Apart from the HSTP, another few timetabling variants were reported but no computational comparison was performed with previous methods. Filho and Lorena (2001) used a Constructive Genetic Algorithm to solve four semi-artificial instances of two public-schools considering the following soft constraints for teachers: avoid undesirable and idle periods. Whereas in Moura and Scaraficci (2010) the authors use a classical GRASP procedure combined with a path-relinking improvement phase. They solve three instances of different schools for a more constrained Brazilian timetabling problem that, in addition to HSTP, we can highlight the following requirements: some teachers must teach lessons simultaneously in different classes and lessons in undesirable periods or days should be avoided. In Poulsen and Bandeira (2013), a MIP model and a heuristic approach are proposed for solving a problem more constrained than HSTP. Both approaches are compared using seven real-world instances originated from Brazilian schools. In that work, the proposed heuristic is a three-phase algorithm based on a divide-and-conquer strategy. The first phase consists in generating an initial feasible solution by solving a MIP model fulfilled only with hard requirements. By using the initial solution, phases two and three iteratively create several subproblems that are optimized using a MIP solver until a stop condition is met. Each subproblem comprises only of a given number of classes. Each subproblem is built by choosing randomly n classes from one single teacher, also chosen randomly. By using this heuristic procedure, the authors were able to provide better results than a MIP solver for 4 out of 7 instances evaluated.

2.3.2 XHSTT format

Along the last editions of international conferences on the Practice and Theory of Automated Timetabling (PATAT), a group of high school timetabling researchers has developed a XML based format, called XHSTT, to express problems from different countries in an unified way (POST et al., 2010; POST et al., 2011). Despite the verboseness of the XHSTT format, it has gained widespread acceptance by the research community and, recently, its use was promoted in the Third International Timetabling Competition (ITC2011) (POST et al., 2013). Problems that can represented in the XHSTT format are denoted here as Generalized High School Timetabling Problem (GHSTP). The objective of the GHSTP is to minimize the number of violations of hard and soft constraints.

The XHSTT format is composed by three basic entities: times, resources and events. Times represent discrete timeslots in a week where events can take place. Resources represent entities that participate in one or more events. Typically, resources are teachers, classes, students, and rooms. Events represent meetings between resources. Each event has a duration that indicates the number of times that are needed to be assigned to the event. Usually, a given set of resources or timeslots are pre-assigned to events. Entities can be organized in several groups.

Additionally, the format allows to represent a set of constraints that a solution should satisfy. Each constraint shares some common properties such as a weight, a cost function, a boolean flag indicating if the constraint is hard or soft, as well as the entities to which the constraint is applied. A constraint can be defined for an individual entity or for groups of entities. Table 2.1 shows a brief description of 15 different constraints that are available in the current version of the XHSTT format. A detailed description of each constraint is described in Kingston (2014a).

Table 2.1 – Overview of the constraints supported by the XHSTT format.

Constraint type	Description
AssignResource	Event resource should be assigned a resource
AssignTime	Event should be assigned a time
SplitEvents	Event should split into a constrained number of sub-events
DistributeSplitEvents	Event should split into sub-events of constrained durations
PreferResources	Event resource assignment should come from resource group
PreferTimes	Event time assignment should come from time group
AvoidSplitAssignments	Set of event resources should be assigned the same resource
SpreadEvents	Set of events should be spread evenly through the cycle
LinkEvents	Set of events should be assigned the same time
AvoidClashes	Resource's timetable should not have clashes
AvoidUnavailable Times	Resource should not be busy at unavailable times
LimitIdle Times	Resource's timetable should not have idle times
ClusterBusyTimes	Resource should be busy on a limited number of days
LimitBusyTimes	Resource should be busy a limited number of times each day
LimitWorkload	Resource's total workload should be limited

Source: (POST et al., 2013).

2.3.3 The Third International Timetabling Competition 2011

The main goal of the Third International Timetabling Competition 2011 (ITC-2011) was stimulating the research in real-world high school timetabling problems, as well as to encourage the use of the XHSTT format by the research community.

The ITC-2011 was split in three rounds. In the first round the competitors were invited to submit solutions for a benchmark set composed of 21 instances in which a subset is composed by instances of HSTP. Since the first round aimed to obtain all-time best solutions, the submitted solutions could be obtained with any technique, using any resources, without any time limit. In the second round a set of hidden instances was used and a time limit of 1000 seconds was imposed. During this round, only free third party tools were permitted, i.e. , commercial software such as CPLEX and Gurobi are excluded. Finally, in the third round the same rules of the first round were used but considering the hidden instances of the second round.

2.3.4 GOAL Team solvers

Among the finalists of the ITC-2011, the GOAL Team won the competition with the *GOAL solver*. This solver is designed as a hybrid approach that combines a Simulated Annealing followed by an Iterated Local Search procedure (FONSECA et al., 2012). In addition, this solver uses seven neighborhood structures that are interchanged along the search according to a given set of probabilities. Some variations of this solver are also presented by the same team in (FONSECA; BRITO; SANTOS, 2012; BRITO et al., 2012). More recently, (FONSECA; SANTOS, 2014) studied several approaches based on Variable Neighborhood Search (VNS). Among them, the Skewed VNS version, refered here as *SVNS solver*, provided the best performance on the whole. Both solvers, GOAL and SVNS, were kindly provided by their authors and are evaluated in this thesis in Chapter 6.

2.3.5 Matheuristic approaches

Regarding the educational timetabling problems, there are a few studies in the literature exploring matheuristics. To the best of our knowledge, there are a few publications related to course timetabling (BURKE et al., 2010; GUNAWAN; NG; POH, 2012) and, specifically, applied to school timetabling, apart from Poulsen and Bandeira (2013), only the work of Avella et al. (2007) used this approach. In that work, the authors proposed a two-phase algorithm applied to an Italian school whose problem is similar to HSTP. The first phase of the algorithm is a simulated annealing (SA) algorithm. While the second phase consists in a very large-scale neighborhood search that decomposes the problem into subproblems which are solved independently by a MIP solver. In each subproblem all teachers remain fixed with exception of a pair of randomly chosen teachers. This second phase can be classified as a fix-and-optimize approach.

2.3.6 Discussion

A brief literature review reveals that most of the works adopt metaheuristic and hybrid methods as solution approaches for solving the high school timetable problem. Despite some exceptions, works that use exact methods, as mathematical programming techniques, in general, are considered very time-consuming and impracticable for most real applications. However, since in the last years several improvements have been made in mixed integer programming solvers (LODI, 2010), we believe that this conclusion deserves to be revalidated. To the best of our knowledge, the literature related to matheuristics applied on timetabling is almost inexistent, what turns this approach a promising direction in the timetabling research.

Although the majority of the proposed techniques were able to successfully solve the problem for which they were developed, these results cannot be generalized because the studied problem is either too specific or evaluated by using a too small dataset. With the introduction of the XHSTT format, theses issues tend to be minimized along the time, however, current techniques proposed to solve this generalized problem are still in their childhood. Even the winner technique of the ITC-2011 was not able to produce final feasible solutions for several instances with known feasible solution in the competition. Although the timetabler may perform adjustments by hand to fix an infeasible solution, there are three main serious drawbacks: (i) the adjustment is made through a lazy negotiation process that, often, generates dissatisfaction among those involved; (ii) adjustments by hand usually degenerates significantly the solution quality. As result, decreasing the benefit provided by the software assistance (iii) the timetabler might not be able to fix the solution.

3 INTRODUCTION TO HSTP

The main goal of this chapter is to introduce the High School Timetabling Problem (HSTP) that was first defined by Souza and Maculan (2000). Also, we are particularly interested in evaluating the performance of MIP solvers when applied to instances of HSTP. In addition to a new MIP model, in this chapter we perform experiments to compare our model with the most recent compact formulation proposed by Santos et al. (2012).

3.1 Problem definition and modeling

The HSTP comes from the Brazilian High School System. The goal of the problem is to build a weekly timetabling. The week is organized as a set of days D, and each day is split into a set of periods P. Let C be a set of classes and T a set of teachers. A class $c \in C$ is a group of students that follow the same course and have full availability. A *timeslot* is a pair, composed of a day and a class period (d,p), with $d \in D$ and $p \in P$, wherein all periods have the same duration. Teachers $t \in T$ may be unavailable in some timeslots.

The main input for the problem is a set of events E that should be scheduled. Typically, an event is a meeting between class and teacher to address a particular subject in a given number of lessons (*workload*) in a given room. Particularly in the Brazilian context, a class, a teacher and a room are pre-assigned to each event $e \in E$. In addition, each event defines how lessons are distributed over a week by requesting an amount of double lessons, restricting the daily limit of lessons, and defining whether lessons taught on the same day are consecutive or not.

A *feasible* timetable has a timeslot assigned to each lesson of events satisfying the hard requirements H1-H6 below:

H1 The workload defined in each event must be satisfied.

H2 A teacher cannot be scheduled to more than one lesson in a given period.

H3 Lessons cannot be taught to the same class in the same period.

H4 A teacher cannot be scheduled to a period in which she/he is unavailable.

H5 The maximum number of daily lessons of each event must be respected.

H6 Lessons from the same event must be consecutive when scheduled for the same day, in case it is required by the event.

Besides feasibility regarding hard constraints, as many as possible of the soft requirements S1-S3 stated below should be satisfied:

S1 Avoid teachers' idle periods.

S2 Minimize the number of *working days* for teachers. In this context, working day means a day that the teacher has at least one lesson assigned to her/him.

S3 Provide the number of double lessons requested by each event.

3.1.1 Problem Formulation

In this subsection we present a MIP formulation for the HSTP adapted from the compact formulation proposed by Santos et al. (2012) considering all the hard and soft requirements mentioned above. The notation used in the problem formulation is presented in Table 3.1.

Our formulation, hereafter denoted as $\mathcal{M}_1$, is novel in three aspects. Firstly, we modified the previous formulation to simplify its presentation replacing each preassigned encounters between teacher/class by a set of events. Secondly, we included the hard requirement H6, which had not been considered in previous studies on HSTP and is further required in Chapter 4. Finally, we proposed a new formulation for the idle times requirement that is faster when solving small instances of this problem.

Table 3.1 – Notation used for the HSTP model.

Symbol	Definition
Sets	
$d \in D$	days of week. $D = \{1,2,...,\|D\|\}$.
$p \in P$	periods of a day. $P = \{1,...,\|P\|\}$.
P'	P without the last two periods of a day. $P' = \{1,...,\|P\|-2\}$.
$t \in T$	set of teachers.
$c \in C$	set of classes.
$e \in E$	set of events.
E_t	set of events assigned to teacher t.
E_c	set of events assigned to class c.
U	set of tuples (m,n) for $m \in P', n \in P : n \geq m+2$.
Q	set of tuples (m,n) for $m \in P', n \in P : n \geq m$.
SG_e	set of timeslots on which event e can start a double lesson ($SG_e = \{(d,p) : d \in D, p \in P$ and $p < \|P\|, V_{edp} + V_{ed,p+1} = 2\}$). The parameter V_{edp} is defined below.
Parameters	
ω_t	cost of each idle period of teacher t.
γ_t	cost of each working day of teacher t.
δ_e	cost of each double lesson of event e not taught sequentially.
R_e	workload of event e.
L_e	maximum daily number of lessons of event e.
V_{edp}	binary parameter that indicates whether the teacher assigned to event e is available in the timeslot (d,p).
MG_e	minimum amount of double lessons required by event e.
Variables	
x_{edp}	binary variable that indicates whether event e is scheduled to timeslot (d,p).
y_{td}	binary variable that indicates whether at least one lesson is assigned to teacher t on day d.
g_{edp}	binary variable that indicates whether event e has a double lesson starting at timeslot (d,p).
G_e	integer variable that indicates the number of double lessons remaining to reach MG_e.
b_{edp}	binary variable that indicates whether event e has a lesson at timeslot (d,p) and not at timeslot $(d,p-1)$.
z_{tdmn}	binary variable that indicates whether the teacher t has idle periods on day d between periods m and n.

Source: created by author.

$$\textbf{Minimize} \sum_{t\in T}\sum_{d\in D}\sum_{(m,n)\in U}\omega_t(n-m-1)z_{tdmn}+\sum_{t\in T}\sum_{d\in D}\gamma_t y_{td}+\sum_{e\in E}\delta_e G_e \tag{3.1}$$

Subject to

$$\sum_{d\in D}\sum_{p\in P}x_{edp}=R_e \qquad \forall e \tag{3.2}$$

$$\sum_{p\in P}x_{edp}\le L_e \qquad \forall e,d \tag{3.3}$$

$$x_{edp}\le V_{edp} \qquad \forall e,d,p \tag{3.4}$$

$$\sum_{e\in E_t}x_{edp}\le y_{td} \qquad \forall t,d,p \tag{3.5}$$

$$\sum_{e\in E_t}\sum_{p\in P}x_{edp}\ge y_{td} \qquad \forall t,d \tag{3.6}$$

$$\sum_{e\in E_c}x_{edp}\le 1 \qquad \forall c,d,p \tag{3.7}$$

$$b_{edp}\ge x_{edp}-x_{edp-1} \qquad \forall e,d,p:p>1 \tag{3.8}$$

$$\sum_{p\in P:p>1}b_{edp}+x_{ed1}\le 1 \qquad \forall e,d \tag{3.9}$$

$$g_{edp}\le x_{edp} \qquad \forall e,(d,p)\in SG_e \tag{3.10}$$

$$g_{edp}\le x_{edp+1} \qquad \forall e,(d,p)\in SG_e \tag{3.11}$$

$$G_e\ge MG_e-\sum_{(d,p)\in SG_e}g_{edp} \qquad \forall e \tag{3.12}$$

$$\sum_{d\in D}y_{td}\ge \max\left\{\left\lceil\frac{\sum_{e\in E_t}R_e}{|P|}\right\rceil,\max_{e\in E_t}\left\{\left\lceil\frac{R_e}{L_e}\right\rceil\right\}\right\} \qquad \forall t \tag{3.13}$$

$$\sum_{(m,n)\in Q}z_{tdmn}=y_{td} \qquad \forall t,d,m\in P' \tag{3.14}$$

$$\sum_{(m,n)\in U}z_{tdmn}\le y_{td} \qquad \forall t,d,n\in P:n\ge 3 \tag{3.15}$$

$$z_{tdpp}\le 1+\sum_{e\in E_t}\left(x_{edp+1}-x_{edp}\right) \qquad \forall t,d,p\in P' \tag{3.16}$$

$$z_{tdmm+1}\le 1-\sum_{e\in E_t}x_{edn} \qquad \forall t,d,(m,n)\in U \tag{3.17}$$

$$z_{tdmn}\le\sum_{e\in E_t}x_{edn} \qquad \forall t,d,(m,n)\in U \tag{3.18}$$

$$x_{edp},g_{edp},b_{edp}\in\{0,1\},G_e\ge 0 \qquad \forall e,d,p \tag{3.19}$$

$$y_{td},z_{tdmn}\in\{0,1\} \qquad \forall t,d,(m,n)\in Q \tag{3.20}$$

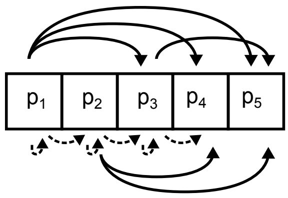
The objective function of the problem formulation consists of three weighted parts related to the soft requirements S1, S2 and S3, respectively. Regarding the soft requirement S1, the penalization is proportional to the number of idle periods.

Constraint set (3.2) ensures that the workload of each event is fully scheduled. Constraint set (3.3) provides a daily limit of lessons for each event. Constraint set (3.4) ensures that the lessons of an event are scheduled in available periods. Constraint sets (3.5) and (3.7) ensure that teacher and class are scheduled to only one lesson at a time. Constraint sets (3.5) and (3.6) identify the working days of teachers. Constraint sets (3.8) and (3.9) ensure that the lessons of an event are scheduled sequentially according to requirement H6. Constraint sets (3.10) and (3.11) enforce double lessons when the variable g_{edp} is equal to one. Constraint set (3.12) determines G_e, the number of double lessons remaining to reach MG_e. Since G_e accounts for the objective function, the sum in the right side of the inequality tends to increase, and thus the establishment of double lessons is promoted.

Constraint set (3.13) is a cut proposed by Souza (2000) that defines a minimum number of working days for each teacher and makes the formulation stronger.

Constraint sets (3.14)-(3.18) determine the number of idle periods in a solution. To explain these constraints, it is useful to consider an *idle periods graph* as shown in Figure 3.1. In this graph the vertices p_1, p_2, p_3, p_4 and p_5 are periods on a day d of a teacher t. There are two types of arcs: *idle period arcs*, with $(m,n) \in U$ that are penalized in the objective function, and *auxiliary arcs*, with $(m,n) \in Q \setminus U$. In Figure 3.1 the idle period arcs are drawn as solid lines and the auxiliary arcs are drawn as dashed lines. Note that each arc corresponds directly to a binary variable z_{tdmn} such that m is the tail and n is the head node of the arc. For example, variable z_{td13} corresponds to the arc (p_1, p_3). The underlying idea is to ensure that the idle period arcs are properly activated to compute the cost of idle periods. Constraint sets (3.14) and (3.15) ensure that there is exactly one arc leaving and reaching each period that can be the beginning or end of an idle period, respectively. Constraint set (3.16) states that an auxiliary arc (m,m) must be active only in two situations: when the teacher has no lesson in period m, or when the teacher has a lesson in periods m and $m+1$. Constraint set (3.17) states that an auxiliary arc $(m,m+1)$ must be active only when the last lesson on a working day of the teacher occurs at period m. Constraint set (3.18) states that an idle period arc (m,n) can be active only when the teacher has a lesson in period n.

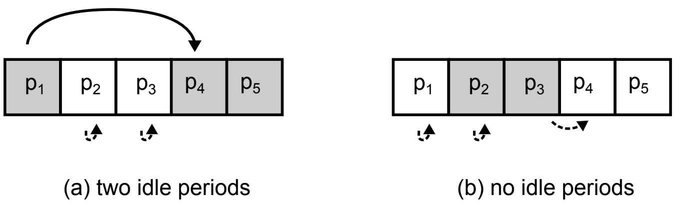
Figure 3.1 – Idle periods graph.



Source: Figure created by author.

Figure 3.2 presents only the activated arcs in the idle periods graph considering two allocation scenarios for a teacher. In scenario (a) there are two idle periods, p_2 and p_3, that are identified by the arc (p_1, p_4). In scenario (b) there are no idle periods. Thus, only auxiliary arcs are activated.

Figure 3.2 – Examples of allocation scenarios. The gray cells indicate periods in which a lesson occurs.



Source: Figure created by author.

3.2 Computational Experiments

In this section we present an experimental evaluation of the model $\mathcal{M}_1$ described previously in this chapter. The goal of our experiments is to answer the following questions:

i) How suitable is solving the HSTP by using a MIP solver?

ii) How the results obtained by solving the model $\mathcal{M}_1$ compare with the results obtained by solving the compact MIP model proposed by Santos et al. (2012)?

iii) Which soft requirement most impacts in the resolution of model $\mathcal{M}_1$?

In order to answer these questions we used the HSTP instances presented in Table 3.2. The first two columns present the instance identifier name. Since their names are long, we use the identifiers for referencing them along the text. Columns

$|D|$ and $|P|$ show the number of days and periods, respectively, while columns $|T|$, $|C|$ and $|E|$ present the number of teachers, classes and events, respectively. Finally, columns $\sum_{e \in E} MG_e$ and $\sum_{e \in E} R_e$ present the total of required double lessons and the total workload, respectively.

Table 3.2 – Main characteristics of HSTP instances.

Id	Name	$\|D\|$	$\|P\|$	$\|T\|$	$\|C\|$	$\|E\|$	$\sum_{e \in E} MG_e$	$\sum_{e \in E} R_e$
1	Inst1	5	5	8	3	21	21	75
2	Inst2	5	5	14	6	63	29	150
3	Inst3	5	5	16	8	69	4	200
4	Inst4	5	5	23	12	127	66	300
5	Inst5	5	5	31	13	119	71	325
6	Inst6	5	5	30	14	140	63	350
7	Inst7	5	5	33	20	205	84	500

Source: created by author.

For solving the models we used the mixed integer programming solver CPLEX 12.1 with default settings. The reported results were computed on a Desktop-PC equipped with an Intel Core i5-2300 processor clocked at 2.8 GHz, 4 GB of RAM, running a 64 bits Linux operating system. The parameters γ_t, ω_t and δ_e were set to 9, 3 and 1, respectively. These values are the same used by previous works on this problem.

Results presented in tables 3.3 and 3.4 are reported with a time limit of 60 minutes. Column *time* shows the running time of the solver in minutes. Column *obj* shows the value of the objective function. Column *gap* presents the percent deviation between the obtained solution and the lower bound computed by the solver. Columns *rows* and *cols* present, respectively, the number of constraints and variables after the pre-processing phase of the solver. Column *nodes* shows the number of explored nodes through the whole search. Finally, column *root* shows the time in seconds for solving the linear relaxation at the root node. Best results are shown in bold.

Table 3.3 presents a comparison between the model $\mathcal{M}_1$ and the compact model proposed by Santos et al. (2012) ($\mathcal{M}_2$). Although the model $\mathcal{M}_1$ produced the best results for most of the instances, results from model $\mathcal{M}_2$ were better on average. While the model $\mathcal{M}_1$ performed better on small size instances, the model $\mathcal{M}_2$ obtained more best results when solving large instances. Note that the number of rows and columns for $\mathcal{M}_1$ is strictly greater when compared to the model $\mathcal{M}_2$. This difference is mainly due to the distinct formulation of the idle periods requirement. Finally, the model $\mathcal{M}_1$ was faster concerning the resolution of the linear relaxation when compared with $\mathcal{M}_2$.

Table 3.3 – Comparison results between models $\mathcal{M}_1$ and $\mathcal{M}_2$

	$\mathcal{M}_1$							$\mathcal{M}_2$						
inst	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root
1	**60.0**	**6.40**	**202**	1491	1099	273744	0.1	**60.0**	6.44	**202**	1191	811	313263	0.1
2	**41.0**	**0.00**	**333**	3210	2889	36614	0.6	60.0	2.06	340	2700	2304	90890	0.8
3	**60.0**	**2.82**	**426**	2229	2296	125094	0.3	**60.0**	**2.82**	**426**	1665	1720	93516	0.5
4	**60.0**	**1.38**	**652**	3491	3906	20599	1.1	**60.0**	1.68	654	3075	3250	24697	1.6
5	**60.0**	5.62	801	7299	6277	8099	3.4	**60.0**	**4.67**	**793**	5979	4882	5227	4.1
6	**60.0**	**5.14**	**778**	7300	6704	4170	3.6	**60.0**	8.55	807	6196	5372	1407	5.5
7	**60.0**	20.65	1259	8962	9034	524	8.8	**60.0**	**13.51**	**1155**	7937	7549	514	12.6
Avg.	**57.3**	6.00	636	4854	4600	66977	2.6	60.0	**5.68**	**625**	4106	3698	75644	3.6

Source: created by author.

Table 3.4 presents the results obtained by model $\mathcal{M}_1$ disregarding, respectively, the requirements S1, S2 and S3. The results reported for $\mathcal{M}_1 \setminus$ S1 shows that disregarding the minimization of idle periods allows the solver to reach optimal solutions for all HSTP instances within 1 hour. The results reported for $\mathcal{M}_1 \setminus$ S2 shows that disregarding the minimization of working days for teachers allows the solver to reach optimal solutions for the majority of the instances. Finally, disregarding the satisfaction of double lessons not significantly affects the resolution.

Table 3.4 – Results for model $\mathcal{M}_1$ disregarding separately a soft requirement

	$\mathcal{M}_1 \setminus$ S1			$\mathcal{M}_1 \setminus$ S2			$\mathcal{M}_1 \setminus$ S3		
inst	time	obj	gap	time	obj	gap	time	obj	gap
1	0.0	189	0.00	0.1	0	0.00	60.0	201	5.97
2	0.1	333	0.00	0.1	0	0.00	6.7	333	0.00
3	0.1	414	0.00	0.5	0	0.00	60.0	426	2.82
4	1.9	643	0.00	4.4	4	0.00	60.0	648	1.39
5	13.4	756	0.00	10.9	0	0.00	60.0	762	0.79
6	13.9	738	0.00	2.1	0	0.00	60.0	765	3.53
7	54.6	999	0.00	60.0	33	100.00	60.0	1041	4.03
Avg.	12.0	582	0.00	11.1	5	14.29	52.4	597	2.65

Source: created by author.

3.3 Conclusions

In this chapter we presented a well-known variant to the High School Timetabling denoted as HTSP. The HSTP is formally defined through a novel MIP model obtained through reformulation of the idle times requirement. The experimenal results conducted in this chapter demonstrated that both model $\mathcal{M}_1$ and $\mathcal{M}_2$ performed similarly on solving HSTP. However, the former was best suitable for solving small instances. This property will be exploited for solving subproblems in the next chapter.

In addition, we performed computational experiments to evaluate the impact of soft requirements in the resolution process by a MIP solver. The obtained results show empirically that the idle periods requirement (S1) is the one that most significantly difficults the resolution of model $\mathcal{M}_1$ by CPLEX.

4 A FIX-AND-OPTIMIZE APPROACH

In the previous chapter we showed that a general purpose MIP solver is better suitable for solving small instances of the HSTP. In this chapter we propose a matheuristic approach that can be used for solving medium and large instances of the HSTP. Section 4.1 presents a fix-and-optimize heuristic combined with a variable neighborhood descent method and three different types of decompositions for the HSTP. Section 4.2 presents a set of experiments to evaluate the proposed method in comparison with previous results reported in the literature. Finally, Section 4.3 presents the main conclusions of this chapter.

4.1 Proposed fix-and-optimize heuristic combined with a variable neighborhood descent strategy

In the model $\mathcal{M}_1$ proposed in the previous chapter, the variable set x_{edp} is the most important one, since other decision variables depend on it. This means that if the values of x_{edp} variables are set, then the values of the remaining ones are easily inferred. This property indicates that the fix-and-optimize heuristic could succeed in solving the model, since fixing binary variables to integer values has only two possibilities. In the fix-and-optimize heuristic, the way we choose the fixed variables as well as the number of fixed variables impacts directly on the performance of the algorithm and on quality of the final solution. Thus, the decomposition operation must vary in *type* and *size*. In this work we propose three types of decomposition:

- *Class Decomposition* (CD): a certain number of classes are free to be optimized.
- *Teacher Decomposition* (TD): a certain number of teachers are free to be optimized.
- *Day Decomposition* (DD): a certain number of days are free to be optimized.

Note that when a class, teacher or day is free to be optimized it means that all variables for the events of this class, teacher or day are not fixed.

For each type of decomposition τ we can define a parameter k which defines the cardinality of the subset of variables that are free to be optimized. For instance, considering the CD, we can free 1,2,3,..., k classes, such that $k \leq |C|$.

The combination of a decomposition type $\tau \in$ {CD, TD, DD} and a size k defines different neighborhoods. The tuple (τ,k) can be used to represent a specific neighborhood. For instance, a neighborhood (DD, 2) of a solution x consists of all solutions that can be obtained by solving subproblems such that $|D| - 2$ days are fixed exactly as in x, but two days are free to be optimized.

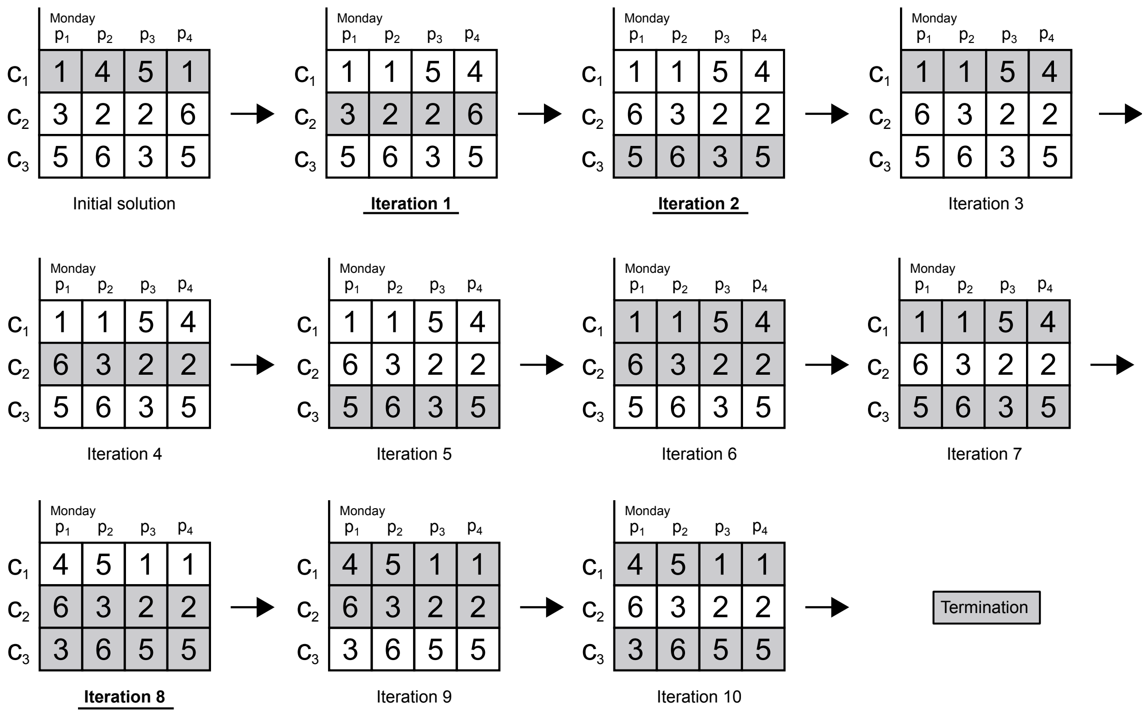
Since there are many possible neighborhoods, we explore them through a variable neighborhood descent (VND) approach (HANSEN; MLADENOVIĆ, 2001). The VND process implies an iteration over a sequence of neighborhoods $\mathcal{N}$ while better solutions are found using a *first improvement* selection strategy. Typically the neighborhoods, $(\tau, k) \in \mathcal{N}$, are explored by a general purpose MIP solver from the smaller to the larger ones.

Figure 4.1 presents the behavior of the fix-and-optimize heuristic applied on a toy instance of the problem. It is composed by three classes (c_1, c_2, c_3), six teachers, four periods (p_1, p_2, p_3, p_4) and only one day. The algorithm begins from a feasible solution and, at each step, solves a different subproblem. Note that the procedure begins with the neighborhood (CD, 1) and improves the current solution twice. At iteration 6, the neighborhood is changed to (CD, 2) and the algorithm is then able to improve the solution once again. In the following sections we present further details of this procedure.

4.1.1 Generating initial feasible solutions

Since the variable fixations described above are based on values of a previous solution, we need to provide an initial feasible solution to start the fix-and-optimize algorithm. In order to do so, we solve a feasibility version of the HSTP by disregarding the objective function of the MIP model presented in Section 3.1, i.e., all soft constraints of the problem. This approach allows the algorithm to quickly find an initial feasible solution.

Figure 4.1 – Example of the proposed fix-and-optimize heuristic using $\mathcal{N} = ((\mathrm{CD}, 1), (\mathrm{CD}, 2))$. Each table shows an iteration of the algorithm, and each cell shows a teacher assignment. The shaded cells denote the group of variables that are free to be optimized, while the remaining ones are fixed with values from the previous solution. Underlined iterations denote that the current solution was improved in the previous iteration. Note that teachers 1, 3, 5 and 6 have idle periods in the initial solution which are gradually removed along the iterations.



Source: Figure created by author.

4.1.2 The proposed algorithm

The overall algorithm is described in the pseudo-code of Figure 4.2. Function `fixAndOptimize()` receives as input a sequence of neighborhoods ($\mathcal{N}$), the overall time limit `(TL)`, and the time limit for each subproblem `(STL)`.

The algorithm begins by creating an initial feasible solution x^* (line 1) as described in Section 4.1.1. If the problem is infeasible it terminates returning no solution.

The outer loop (lines 5-23) iterates over a sequence of neighborhood structures $\mathcal{N}$ on the same fashion as a VND algorithm. Each neighborhood has a finite number of subproblems computed by function `subproblemCount()` (line 6) as described in the pseudo-code of Figure 4.3. The number of subproblems, indicated by `s`, depends on the type of the decomposition τ and on its size k.

In the inner loop (lines 9-22) the subproblems of the current neighborhood are explored until `noImprov=count`, i.e., the algorithm evaluates each subproblem (within the subproblem time limit STL) of the neighborhood (τ,k) and is not able to improve the quality of the current solution, i.e., the level of satisfaction of the soft constraints.

The function `decompose()` (line 10) is used to compute the set of variables to be optimized ($\mathcal{R}$) in the current subproblem according to the pseudo-code presented in Figure 4.4. The function `subsets`(S, k, s) returns in lexicographical order the s^{th} subset of all subsets of S containing exactly k elements.

After that, the subproblem is solved through function `solve()` which receives three parameters: the current solution (x^*), the set of variables to be optimized ($\mathcal{R}$), and the time limit of the subproblem `(STL)`. This function fixes all variables x_{edp} which do not belong to $\mathcal{R}$ to their values in x^*, and starts the solver. If it is able to find a better solution than x^*, then it is returned. Otherwise, if no better solution is found within the time limit, or if the subproblem is infeasible, it returns the previous current solution x^*. Note that subproblems can be infeasible since we add a cutoff constraint that forces the solver to search only for solutions whose objective value is less than the objective value of x^*. After the function `solve()` returns a result, all variables previously fixed are released.

Whenever a better solution is found it becomes the current solution x^* (line 13), and variable `noImprov` is reset. Otherwise, `noImprov` is incremented.

The algorithm terminates returning the best solution found when the time limit `TL` is reached (lines 18-20). In line 21 the variable `s` indexes the next subproblem. After all neighborhoods in the outer loop are explored, the algorithm terminates in line 23 returning the best visited solution x^*.

Figure 4.2 – Pseudo-code of the proposed fix-and-optimize heuristic.

Algorithm `fixAndOptimize` ($\mathcal{N}$, `TL`, `STL`)

1: $x^* \leftarrow$ `GenerateInitialSolution()`;
2: **if** $x^* = \emptyset$ **then**
3:     **return** $\emptyset$;
4: **end if**
5: **for all** $(\tau, k) \in \mathcal{N}$ **do**
6:     `count` $\leftarrow$ `subproblemCount`(τ, k);
7:     `s` $\leftarrow 1$;
8:     `noImprov` $\leftarrow 0$;
9:     **repeat**
10:         $\mathcal{R} \leftarrow$ `decompose`$(\tau, k,$ `s`$)$;
11:         $x \leftarrow$ `solve`$(x^*, \mathcal{R},$ `STL`$)$;
12:         **if** x is better than x^* **then**
13:             $x^* \leftarrow x$;
14:             `noImprov` $\leftarrow 0$;
15:         **else**
16:             `noImprov++`;
17:         **end if**
18:         **if** `TL` was reached **then**
19:             **return** x^*;
20:         **end if**
21:         `s` $\leftarrow$ `(s` **mod** `count)`$+1$;
22:     **until** `noImprov`$=$`count`;
23: **end for**
24: **return** x^*.

Source: Figure created by author.

Figure 4.3 – Function that computes the number of subproblems of a neighborhood.

Algorithm `subproblemCount` (τ, k)

switch(τ)
  case CD
    `count` $\leftarrow \binom{|C|}{k}$;
  case TD
    `count` $\leftarrow \binom{|T|}{k}$;
  case DD
    `count` $\leftarrow \binom{|D|}{k}$;
end switch
return `count`.

Source: Figure created by author.

Figure 4.4 – Decomposition function.

Algorithm `decompose` $(\tau, k, \mathtt{s})$

switch(τ)
  case CD
    $\mathcal{R} \leftarrow \{x_{edp} : c \in \mathtt{subsets}(C, k, s), e \in E_c, d \in D, p \in P\}$;
  case TD
    $\mathcal{R} \leftarrow \{x_{edp} : t \in \mathtt{subsets}(T, k, s), e \in E_t, d \in D, p \in P\}$;
  case DD
    $\mathcal{R} \leftarrow \{x_{edp} : e \in E, d \in \mathtt{subsets}(D, k, s), p \in P\}$;
end switch
return $\mathcal{R}$.

Source: Figure created by author.

4.2 Computational experiments

In this section we present an experimental evaluation for the fix-and-optimize heuristic proposed in this chapter. The goal of our experiments is to answer the following questions:

i) Does the proposed algorithm outperform a general purpose MIP solver?

ii) Which sequence of neighborhoods $\mathcal{N}$ provides the best results?

iii) How do our results compare with results of the state-of-the-art methods for solving the problem?

The subproblems are solved by CPLEX 12.1 (IBM, 2009) with default settings and the algorithms were implemented in C++ using the compiler g++ 4.6.1. The experimental results were computed in a Desktop-PC equipped with an Intel Core i5-2300 processor clocked at 2.8GHz, 4GB of RAM, over a 64 bits Linux operating system. Along this section, we report results of one run for each tested algorithm, since they are deterministic. The mathematical model parameters γ_t, ω_t and δ_e were set to 9, 3 and 1, respectively. These values are the same used by all previous works on HSTP of our knowledge.

4.2.1 Dataset

To evaluate the algorithm, we used the instances presented in Table 4.1. In the table, the first two columns present the instance identifier name. Since their names are long, we use the identifiers for shortening reference along the text. Columns $|D|$ and $|P|$ show the number of days and periods, respectively, while columns $|T|$, $|C|$ and $|E|$ present the number of teachers, classes and events, respectively. Finally, columns $\sum_{e \in E} MG_e$ and $\sum_{e \in E} R_e$ present the total number of required double lessons and the total amount of workload, respectively.

The instances are split into two sets. Instances 1-7 comprise set-1 and are available from the repository (LABIC, 2008) and to the best of our knowledge they were used in all previous works on HSTP. Requirement H6 is not considered in this group of instances. Instances A, D, E, F, G, from set-2, are different versions of instances 1, 4, 5, 6, 7, respectively. They differ mainly in two aspects: in set-2, teachers are available in all periods, and requirement H6 is considered. These modifications made the instances of set-2 more challenging to be included in the first round of the Third International Timetabling Competition 2011 (ITC-2011) (ITC, 2011). They are part of the XHSTT-2012 archive [1].

[1] <http://www.utwente.nl/ctit/hstt/archives/XHSTT-2012/>

Table 4.1 – Main characteristics of the tested instances.

Id	Name	$\|D\|$	$\|P\|$	$\|T\|$	$\|C\|$	$\|E\|$	$\sum_{e \in E} MG_e$	$\sum_{e \in E} R_e$
1	Inst1	5	5	8	3	21	21	75
2	Inst2	5	5	14	6	63	29	150
3	Inst3	5	5	16	8	69	4	200
4	Inst4	5	5	23	12	127	66	300
5	Inst5	5	5	31	13	119	71	325
6	Inst6	5	5	30	14	140	63	350
7	Inst7	5	5	33	20	205	84	500
A	BrazilInstance1	5	5	8	3	21	21	75
D	BrazilInstance4	5	5	23	12	127	66	300
E	BrazilInstance5	5	5	31	13	119	71	325
F	BrazilInstance6	5	5	30	14	140	63	350
G	BrazilInstance7	5	5	33	20	205	84	500

Source: created by author.

4.2.2 Initial solutions

Table 4.2 shows the initial feasible solution values obtained by CPLEX according to the procedure described in Section 4.1.1. Column *LB* presents the best known lower bounds computed for the instances. Lower bounds for instances 1-7 were provided by Santos et al. (2012), while the remaining lower bounds were obtained by solving the linear relaxation of the model presented in Section 3.1.1. Column *obj* shows the value of the objective function. Column gap_L presents the percentage deviation from the best known lower bound (*LB*). It is computed by $100*(obj-LB)/LB$. Column *time* shows the running times in seconds.

As one can notice, the proposed procedure provides feasible solutions in a short time, spending just 6 seconds on average. Despite their values begin far from the lower bound *LB*, it is important to note that the method provides feasible solutions quickly without burdening the overall total time. As expected, the generation of initial solutions for instances of set-2 are more time demanding, since the solution space is larger when all teachers have full availability.

Table 4.2 – Initial feasible solutions.

Id	LB	obj	gap_L (%)	time (s)
1	202	612	202.97	0.0
2	333	1089	227.03	0.2
3	423	1172	177.07	0.4
4	652	1598	145.09	0.9
5	762	2369	210.89	0.9
6	756	2299	204.10	1.1
7	1017	2758	171.19	2.9
A	189	567	200.00	0.2
D	621	1658	166.99	10.6
E	756	2109	178.97	10.8
F	738	2247	204.47	12.5
G	999	2776	177.88	31.6
Avg.	620.7	1771.2	188.89	6.0

Source: created by author.

4.2.3 Parameter setting

In this section we describe a set of experiments that supported us to define a standard parameter setting to be used by the proposed heuristic. Basically, we aimed to define the sequence of neighborhoods $\mathcal{N}$, the order in which the different neighborhoods are visited, and a suitable Subproblem Time Limit (STL).

Initially, we tested several neighborhoods composed by a single tuple (τ,k), i.e., $|\mathcal{N}| = 1$, with $\tau \in \{\text{CD, TD, DD}\}$ and with several values for the decomposition size k. For the neighborhoods involving day decompositions we tested $k \in \{1,...,5\}$ and for teacher and classes decompositions we used $k \in \{1,...,12\}$. Obviously, the maximum value for k is set to five for the decomposition DD since all tested instances have $|D| = 5$. Considering the decompositions CD and TD the maximum value for k was chosen in order to keep a good trade-off between performance and solution quality. These neighborhoods are combined with two different STL values: STL=30 and STL=∞, meaning that the subproblem time limit is 30 seconds in the first case, and when set to ∞ the subproblem runs to optimality, or the overall time limit (TL) is reached.

Figure 4.5 shows average results, considering all instances, for the gap (plot in the top), and running times (plot in the bottom) of the different combinations of neighborhoods and STL values. The gap value of each instance is calculated as the percentage deviation of the solution value found to the best known values. For each combination, the overall time limit (TL) of each run was set to 10 minutes.

Analyzing this figure, it can be observed that class and teacher decompositions provide better results than day decomposition. This occurs because the subproblems generated by day decompositions are too large and CPLEX spends a long time on each subproblem, leading to few solution improvements.

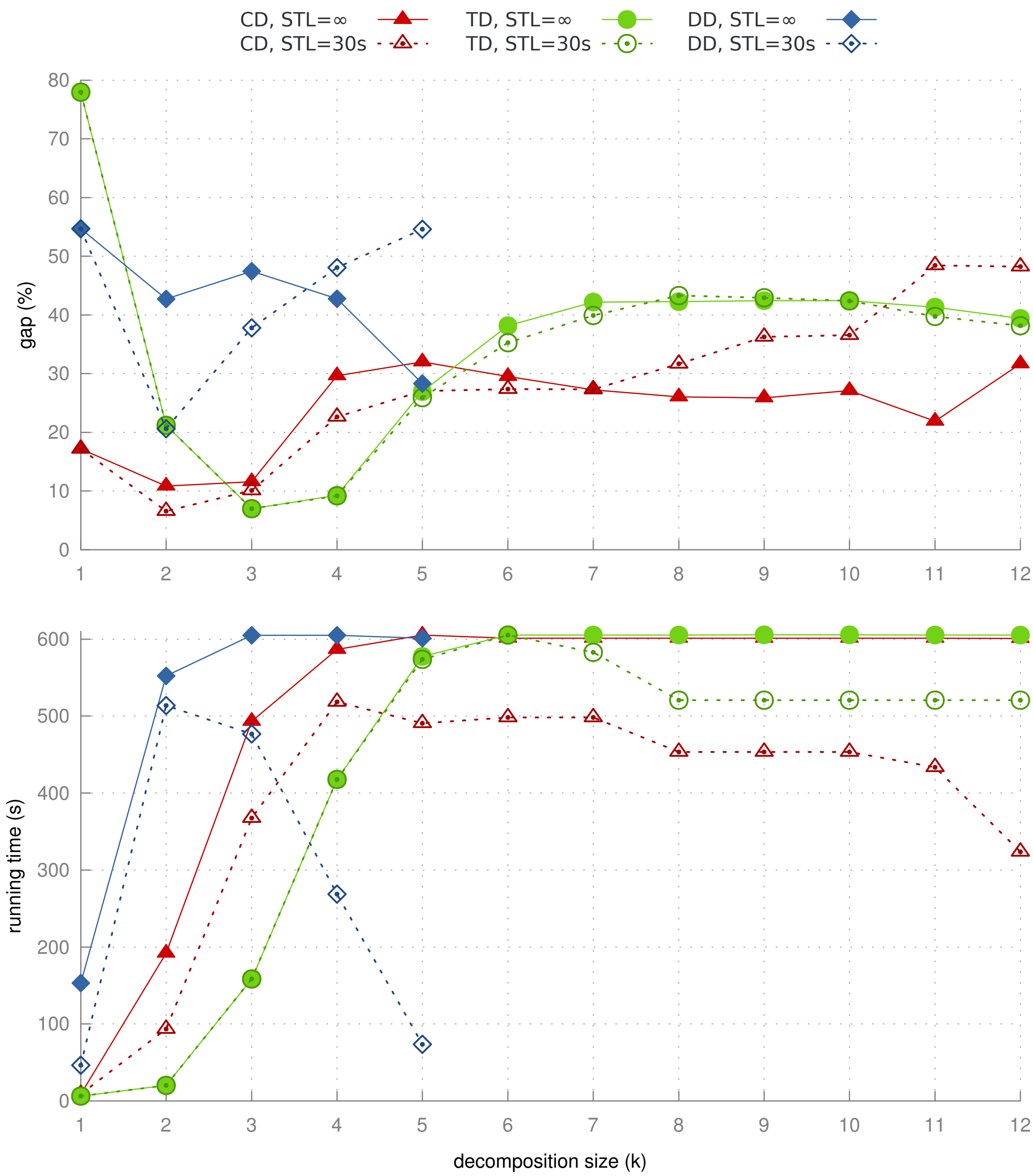
Another observation is related to the STL parameter. All runs with STL=30 spent less time than STL=∞. A suitable setting of this parameter is fundamental for the overall heuristic performance. If the time available is too short, the solver rarely solves the subproblem. This can be observed in Figure 4.5 in cases with STL=30 in class decomposition with $k \geq 8$, and for day decomposition with $k \geq 4$. This behavior is expected since the number of free variables increases with k. In these cases the algorithm often finishes prematurely. Finally, runs with teacher decomposition were almost not affected by STL since most of the subproblems were solved within the subproblem time limit.

Unexpectedly, in some situations in which the algorithm finished before the overall time limit, results using STL=30 were better than with STL=∞ (for instance, for class decomposition $k \leq 7$). In this case, when STL=∞ the algorithm performs large decreasing steps in the objective function value during the first iterations. As a side effect some parts of the solution are "frozen" in its local optima discouraging further interactions with other solution parts.

In summary, considering the overall results for single neighborhoods, we concluded that class and teacher decompositions with small size values (ranging from 1 to 4) might provide a good trade-off between running time and solution quality. Thus it is reasonable to consider them as strong candidates to take place the first positions in a sequence of neighborhoods for the VND method.

In order to answer which sequence of neighborhoods $\mathcal{N}$ provides the best results, we evaluated different sequences presented in Table 4.3, producing different variants (*Var*) of the algorithm (F1-F10). Each sequence is composed by different arrangements of teacher and class decompositions where neighborhoods with small values of k appear first since, as we concluded previously, they produce high quality solutions in a short time. All variants were tested considering three different values of parameter STL = $\{10, 30, 50\}$. For each test we report the average gap ($\overline{\text{gap}_L}$)

Figure 4.5 – Each line in the plots represents a combination of decomposition type and STL value. The solid lines show results from runs with STL=∞, while dashed lines show results from runs for STL=30. The x-axis represents the size (k) of each evaluated decomposition.



Source: Figure created by author.

and the number of optimal solutions found (*#opt*). The overall time limit (TL) of each run was set to 10 minutes.

Table 4.3 – Results for variants of the fix-and-optimize heuristic.

		STL = 10s		STL = 30s		STL = 50s	
Var	Sequence of neighborhoods ($\mathcal{N}$)	$\overline{\text{gap}_L}$(%)	#opt	$\overline{\text{gap}_L}$(%)	#opt	$\overline{\text{gap}_L}$(%)	#opt
F1	$((\text{TD},1),...,(\text{TD},\infty))$	3.91	1	3.82	2	3.85	2
F2	$((\text{CD},1),...,(\text{CD},\infty))$	3.36	2	3.32	2	3.22	2
F3	$((\text{TD},2),...,(\text{TD},\infty))$	3.95	1	3.86	2	3.86	2
F4	$((\text{CD},2),...,(\text{CD},\infty))$	3.36	2	3.24	2	3.42	2
F5	$((\text{TD},1),(\text{CD},1),...,(\text{TD},\infty),(\text{CD},\infty))$	3.04	3	3.19	3	3.19	3
F6	$((\text{CD},1),(\text{TD},1),...,(\text{CD},\infty),(\text{TD},\infty))$	3.04	3	3.13	3	3.23	3
F7	$((\text{TD},2),(\text{CD},2),...,(\text{TD},\infty),(\text{CD},\infty))$	2.90	3	2.84	3	3.11	3
F8	$((\text{CD},2),(\text{TD},2),...,(\text{CD},\infty),(\text{TD},\infty))$	3.02	4	2.92	4	3.17	3
F9	$((\text{TD},3),(\text{CD},3),...,(\text{TD},\infty),(\text{CD},\infty))$	3.42	2	3.68	2	3.61	2
F10	$((\text{CD},3),(\text{TD},3),...,(\text{CD},\infty),(\text{TD},\infty))$	6.02	3	7.34	3	8.29	3

Source: created by author.

According to the table, we observed that the algorithm was not significantly sensitive to changes in the parameter STL. The best results were obtained by variants F5-F8, which results are quite similar. While the variant F7 achieved the best average gap among them, the variant F8 found the largest number of optimal solutions. On the whole, the results indicated that variants mixing different types of decompositions (F5-F10) are better than variants with just one type of decomposition (F1-F4), except for variants F9 and F10. Particularly, these two performed poorly since the time limit imposed was too short to deal properly with sequences of neighborhoods starting with large decompositions ($k \geq 3$). In fact, according to our experience, when more time is available, in average, variants F5-F10 are strictly better than variants F1-F4. We chose the variant F8 using STL=30 as the standard setting given it obtained a high number of optimal solutions. This variant is used in the next experiments performed in this work.

4.2.4 Comparison with CPLEX

In Table 4.4 the results obtained by variant F8 of the proposed fix-and-optimize heuristic are compared with the results obtained by the general purpose solver CPLEX (CPX). The labels CPX and F8 are subscripted with the overall time limit used in the method. Column BKV shows the previous best known solution values. Whereas the values for instances 1, 2, 3, and 6 were obtained by Santos et al. (2012), the values for instances 4, 5, and 7 were obtained by the model $\mathcal{M}_1$ proposed in Chapter 3. Finally, results for set-2 were the best generated solutions reported in the first round of ITC-2011 By the competition rules, the best known solution could be obtained by any technique, using any resources, without any time limit. The teams had five months to produce these results. For each method we report the objective value (obj) and the percentage deviation (gap_B) from the best known value (BKV). Column gap_B is computed by $100 * (obj{-}BKV)/min(BKV{,}obj)$. Thus, a negative gap_B value represents an improvement over the best known solution value. Improved results are shown in boldface.

As the table shows, the proposed algorithm was able to find better solutions than CPLEX spending considerably less computational time. For example, $\text{F8}_{(10min)}$ found, on average, better or equal results than $\text{CPX}_{(10h)}$ for all instances, except for instance 3. In a separate experiment, the variant $\text{F8}_{(6h)}$ was also able to find the optimal result for this instance.

We can also observe that if more time is available, the proposed algorithm can improve the quality of the solutions even further. For set-1, for example, variant $\text{F8}_{(10min)}$ was able to obtain the best known value for three instances, and two new best known values (instances 5 and 7) were found. Variant $\text{F8}_{(30min)}$ found a new best known value for instance 6, and improved the previous best known value of instance 7. Variant $\text{F8}_{(1h)}$ improved the solution of instance 3. Regarding instances of set-2, considerable improvements are observed when comparing the proposed algorithm with the best known values and with CPLEX results. Both, our method and CPLEX, found the best known solution for instance A. For the remaining instances, $\text{CPX}_{(10h)}$ was unable to find the best known solution for any of them, and the gap ranges from 3.07% up to 80.64%. On the other hand, variant $\text{F8}_{(10min)}$ was able to

Table 4.4 – Comparison results between CPLEX and the proposed fix-and-optimize heuristic.

		$CPX_{(1h)}$		$CPX_{(10h)}$		$F8_{(10min)}$		$F8_{(30min)}$		$F8_{(1h)}$	
Id	BKV	obj	gap_B (%)	obj	gap_B (%)	obj	gap_B (%)	obj	gap_B (%)	obj	gap_B (%)
1	202	202	0.00	202	0.00	202	0.00	202	0.00	202	0.00
2	333	333	0.00	333	0.00	333	0.00	333	0.00	333	0.00
3	423	426	0.71	423	0.00	429	1.42	429	1.42	426	0.71
4	652	652	0.00	652	0.00	652	0.00	652	0.00	652	0.00
5	764	801	4.84	764	0.00	**762**	**−0.26**	**762**	**−0.26**	**762**	**−0.26**
6	760	778	2.37	765	0.66	761	0.13	**759**	**−0.13**	**759**	**−0.13**
7	1028	1259	22.47	1028	0.00	**1019**	**−0.88**	**1017**	**−1.08**	**1017**	**−1.08**
A	200	200	0.00	200	0.00	200	0.00	200	0.00	200	0.00
D	665	873	31.28	737	10.83	**653**	**−1.84**	**648**	**−2.62**	**648**	**−2.62**
E	799	1017	27.28	840	5.13	**795**	**−0.50**	**784**	**−1.91**	**778**	**−2.70**
F	815	1520	86.50	840	3.07	**796**	**−2.39**	**787**	**−3.56**	**787**	**−3.56**
G	1121	2025	80.64	2025	80.64	**1087**	**−3.13**	**1085**	**−3.32**	**1084**	**−3.41**
Avg.	646.8	840.5	21.34	734.1	8.36	**640.8**	**−0.62**	**638.2**	**−0.96**	**637.3**	**−1.09**

Source: created by author.

improve the best known solutions for all of them. New improvements are obtained for variants $F8_{(30min)}$ and $F8_{(1h)}$.

In fact, along the development of the fix-and-optimize algorithm, several new best results were found while testing different configurations of the algorithm. The next section presents the new best known results for the instances.

4.2.5 New best known results

Table 4.5 presents the best results produced in this study [2]. Results shown in boldface are new best known values. Columns *LB* and *BKV* were previously presented, respectively, in tables 4.2 and 4.4. Column *obj* presents the objective value obtained by the proposed fix-and-optimize heuristic. Columns BKV_{itc} and obj_{itc} present the solution evaluation provided by HSEVal validator [3]. HSEVal checks the solution feasibility and also computes the solution value. It was used to evaluate the solutions of ITC-2011. Note that $obj_{itc} = obj - LB$, as well as $BKV_{itc} = BKV - LB$. Some cells are filled with "-" since instances 1 to 7 were not tested in ITC-2011. Columns gap_L and gap_B are computed as mentioned, respectively, in Sections 4.2.2 and 4.2.4. Results whose gap_L value is zero represent an optimal solution. Finally, column *Variant* presents which variant of fix-and-optimize heuristic produced the best result reported for each instance in column *obj*.

The solutions achieved by our approach are equal or better in quality when compared to best known results reported in the literature. Our method was able to find seven new best values out of the 12 instances analysed. In addition to new optimal values achieved for instances 5, 6, and 7 we found the optimal results for all instances in set-1. Moreover, our algorithm was able to improve all solutions for HSTP instances of the first round of the Third International Timetabling Competition 2011, except for instance A, where the result matches the previous best known value. We would like to emphasize that the previous results were obtained by several techniques, and no time limit was imposed. This clearly illustrates the effectiveness of our method.

[2] The solution for each instance whose values are reported in Table 4.5 is available at <www.inf.ufrgs.br/~apdorneles/timetabling/2013HSFOPTVND>.

[3] <http://sydney.edu.au/engineering/it/~jeff/hseval.cgi>

Table 4.5 – New best known results.

Id	LB	BKV	obj	BKV_{itc}	obj_{itc}	gap_L (%)	gap_B (%)	Variant
1	202	202	202	-	-	0.00	0.00	$F8_{(10min)}$
2	333	333	333	-	-	0.00	0.00	$F8_{(10min)}$
3	423	423	423	-	-	0.00	0.00	$F10_{(1h)}$
4	652	652	652	-	-	0.00	0.00	$F8_{(10min)}$
5	762	764	**762**	-	-	**0.00**	-0.26	$F8_{(10min)}$
6	756	760	**756**	-	-	**0.00**	-0.53	$F10_{(1h)}$
7	1017	1028	**1017**	-	-	**0.00**	-1.08	$F8_{(30min)}$
A	189	200	200	11	11	5.82	0.00	$F8_{(10min)}$
D	621	665	**648**	44	**27**	**4.35**	-2.62	$F8_{(30min)}$
E	756	799	**776**	43	**20**	**2.65**	-2.96	$F10_{(1h)}$
F	738	815	**779**	77	**41**	**5.56**	-4.62	$F7_{(1h)}$
G	999	1121	**1066**	122	**67**	**6.71**	-5.16	$F10_{(2h)}$
avg.	620.7	646.8	634.5			2.09	-1.44	

Source: created by author.

Table 4.6 presents individually the level of satisfaction regarding soft requirements for the best solutions produced in this study. Columns *obj* and gap_L present, respectively, the objective value and the optimality gap for each instance. Column *S1* presents the number of idle periods. Column *S2* presents the number of working days exceeding the minimum number of days defined by constraint set (3.13) from the problem formulation. Column *S3* presents the number of unsatisfied double lessons. Column *H6* presents the number of non-consecutive lessons. We recall that requirement H6 is not taken into account for instances of set-1.

From Table 4.6, it can be appreciated that virtually all solutions present some violations of soft requirements, but these are expected and occur even in the optimal solutions. Among the soft requirements, the minimization of working days was the only one thoroughly satisfied in all solutions. Regarding the requirement S3, we can observe a discrepancy when comparing solutions of set-1 and set-2. In set-1 there is only a single violation for instance 1, while for set-2 several violations are identified. This difference suggests that considering H6 as a hard requirement impacts directly in the satisfaction of double lessons. In other words, it is hard to satisfy simultaneously both the soft requirement S3 and the hard requirement H6.

Table 4.6 – Soft requirement satisfaction for the best solutions found.

Id	obj	gap_L (%)	S1	S2	S3	H6
1	202	0.00	4	0	1	1
2	333	0.00	0	0	0	0
3	423	0.00	3	0	0	20
4	652	0.00	3	0	0	12
5	762	0.00	2	0	0	11
6	756	0.00	6	0	0	26
7	1017	0.00	6	0	0	17
A	200	5.82	3	0	2	0
D	648	4.35	3	0	18	0
E	776	2.65	2	0	14	0
F	779	5.56	7	0	20	0
G	1066	6.71	7	0	46	0

Source: created by author.

4.3 Conclusions

In this chapter, we presented a novel approach for solving a variant of the high school timetabling problem which explores class, teacher and day decompositions. We proposed a fix-and-optimize heuristic combined with a variable neighborhood descent method that produces solutions which satisfy all hard constraints, i.e., feasible solutions. In addition, we proposed a simple construction procedure that quickly generates feasible initial solutions. The experimental results show that our approach provides high quality feasible solutions in a smaller computational time when compared with results obtained with the general-purpose integer programming solver CPLEX. We have improved best known solutions in the case of seven out of 12 instances quoted in the literature. Among these new solutions, three are new optimal solutions for classical instances that have been available since 2000. Further, our method was able to obtain better solutions for four out of five HSTP instances from the first round of the Third International Timetabling Competition (held in 2011), outperforming the results obtained by state-of-the-art techniques. The results obtained in this chapter show that the proposed technique is very promising to solve the HSTP and motivates its use to other variants of this problem. A further investigation of the fix-and-optimize approach is presented in Chapter 6.

5 A COLUMN GENERATION APPROACH

In this chapter we propose a column generation algorithm for providing tight lower bounds for the HSTP. The remainder of this chapter is organized as follows. Section 5.1 reintroduces the HSTP as a multicommodity flow problem with additional constraints along with a novel compact MIP formulation. Section 5.2 presents a Dantzig-Wolf decomposition for the HSTP, a column generation algorithm, and two speedup strategies. Section 5.3 presents experimental results for the proposed column generation in comparison with an state-of-the-art approach for the problem. Finally, Section 5.4 presents our major conclusions.

5.1 Problem Definition and Modelling

The problem considers a set of classes C and a set of teachers T. A class $c \in C$ is a group of students that follow the same course and have full availability. The goal of the problem is to build a timetable for a week that is usually organized as a set of days D, and each day is split into a set of periods P. We call as *timeslot* a pair composed of a day and class period, (d,p), with $d \in D$ and $p \in P$, wherein all periods have the same duration. Teachers $t \in T$ may be unavailable in some timeslots.

The main input for the problem is a set of events that should be scheduled. Typically, an event is a meeting between a teacher t and a class c to address a particular subject in a given room. In this chapter, we denote an event by a pair (t,c). The parameter H_{tc} determines the *workload* of an event (t,c), i.e, the number of lessons that must be taught by the teacher t for the class c. In addition, each event defines how lessons are distributed in a week by requesting an amount of double lessons, restricting the daily limit of lessons, and defining whether lessons taught in a same day must be consecutive.

A *feasible* timetable has a timeslot assigned to each lesson of events satisfying the hard requirements H1-H6 below:

- H1 The workload of each event must be satisfied.
- H2 A teacher cannot be scheduled to more than one lesson in a given period.
- H3 Lessons cannot be taught to the same class in the same period.
- H4 A teacher cannot be scheduled to a period in which he/she is unavailable.
- H5 The maximum number of daily lessons of each event must be respected.
- H6 Two lessons from the same event must be consecutive when scheduled for the same day, in case it is required by the event.

Besides feasibility regarding hard constraints, as many as possible of the soft requirements S1-S3 stated below should be satisfied:

- S1 Avoid teachers' idle periods.
- S2 Minimize the number of *working days* for teachers. In this context, working day means a day that the teacher has at least one lesson assigned to him/her.
- S3 Provide the number of double lessons requested by each event.

The HSTP can be modeled as a multicommodity flow problem with additional constraints, where each teacher is represented by a commodity. It means that determining a teachers' schedule is the same as finding a path in an appropriate network graph. Formally, we represent this network as a directed acyclic graph $G = (V, A)$, where V is a set of nodes and A is a set of arcs. Although all commodities share the same set of nodes, including the same *source* and *sink* nodes, each commodity considers only a given subset of arcs $A_t \subset A$. Figure 5.1 presents an illustration of the graph G where all types of arcs are shown for a given commodity t. The figure is composed of days (vertical rounded rectangle) and periods of a day (horizontal straps). Each day block has vertical shaded rectangles related to the activities of each class (two classes are considered in the example). Next we describe the types of arcs of G:

- *Lesson arcs* are used to indicate which timeslots are assigned to a given teacher and class. Lessons arcs are usually shared between commodities and have an unitary capacity associated to ensure they are used only by a single commodity (teacher) at a time. In addition, for each lesson arc $a \in A_t$ is associated a label

S_{ta} that represents the duration (in periods) of the lesson represented by the arc. Lesson arcs are referred to as *single lesson arcs* when $S_{ta} = 1$, and as *double lesson arcs* when $S_{ta} = 2$. In the figure, lesson arcs are all curve arcs within a day block and within the shaded rectangle related to some class.

- W_t is the set of *idle period arcs.* These arcs are used to identify the idle periods for each teacher t. A cost ω is associated to each idle period arc. In the figure, idle period arcs are all straight arcs within a day block, outside the shaded rectangles related to classes.

- Sets Q_t^- and Q_t^+ are sets of auxiliary arcs called, respectively, as *pull-in* and *pull-out* arcs. While pull-in arcs are all arcs incoming a day block, pull-out arcs are the ones outgoing a day block.

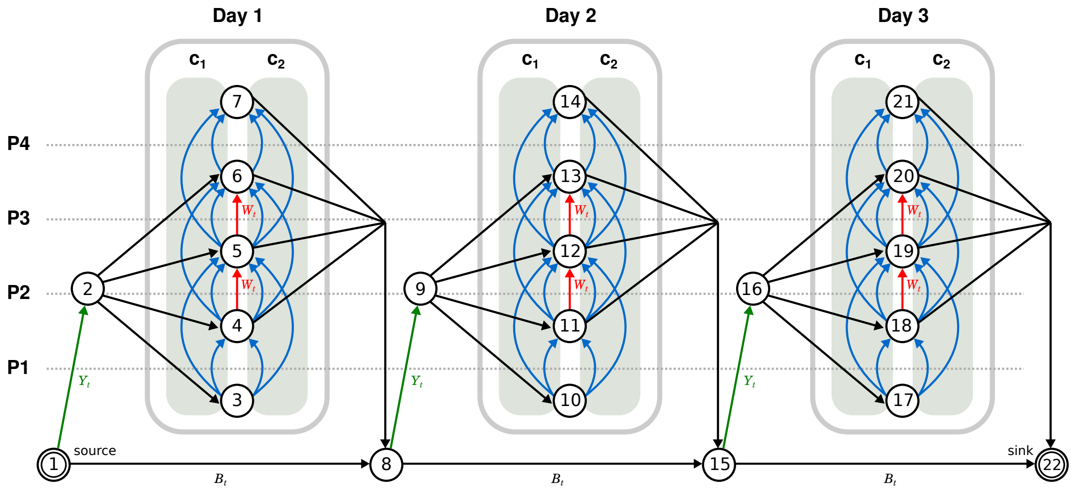
- Y_t is the set of *working day arcs.* These arcs are used to compute the number of working days for a teacher t. A cost γ is associated to each working day arc. In the figure, for each day, the head node of the working day arc corresponds to the tail node of each pull-in arc to that day.

- B_t is the set of *day-off arcs.* These arcs are used when a teacher t does not teaches any lesson in a given day. These are the arcs located in the lower base of the figure. Their tail nodes are the same of working day arcs.

Each path in the network is composed by a binary flow denoted by the variable x_{ta}, where $t \in T$ and $a \in A_t$. Each path starts from the source node, alternates through different types of arcs, ending at the sink node, as shown in Figure 5.2.

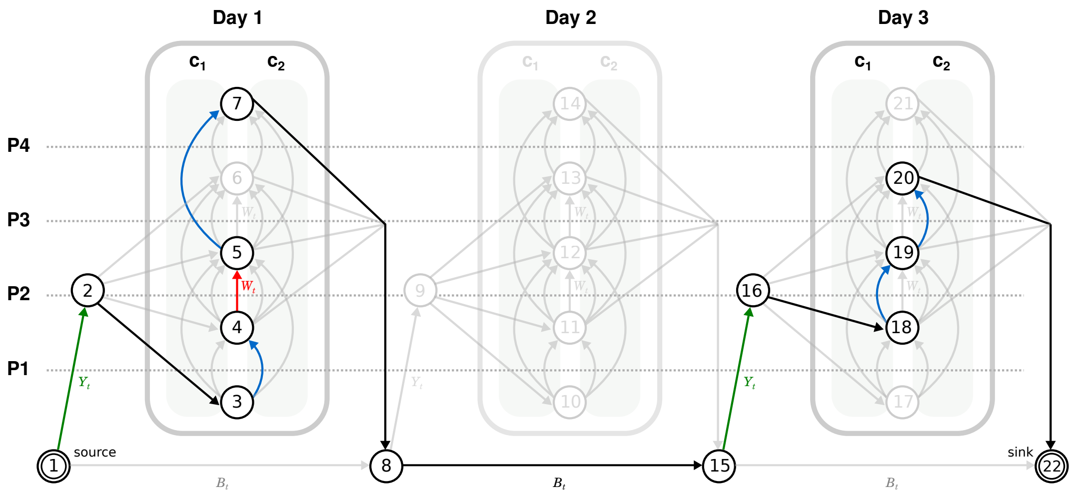
Next, we present a mixed integer linear programming formulation for the HSTP hereafter denoted as $\mathcal{F}_1$. For convenience, the complete notation used in the formulation is presented in Table 5.1.

Figure 5.1 – Example of a network graph in a toy instance composed by three days, four periods by day (P1, P2, P3, P4), and two classes (c₁, c₂). Each day of the week is represented by a rounded rectangle where lesson arcs and idle period arcs are located. Inside each, lesson arcs appear in two groups represented by a shaded rectangle, where each group represents the lesson arcs for classes c_1 and c_2.



Source: Figure created by author.

Figure 5.2 – Example of a feasible schedule for a teacher t represented by a path in the network. In this example, a teacher works only on days 1 and 3. On day 1, she/he teaches a single lesson for the class c_2 in the period P1, becomes idle in the period P2, and then gives a double lesson starting in the period P3 for the class c_1. On day 3, she/he teaches a single lesson for class c_1 in the period P2 and another one for class c_2 in the period P3.



Source: Figure created by author.

Table 5.1 – Notation used in the compact formulation $\mathcal{F}_1$.

Symbol	Definition
Sets	
$d \in D$	days of week. $D = \{1, 2, ..., \vert D\vert\}$.
$p \in P$	periods of a day. $P = \{1, ..., \vert P\vert\}$.
$t \in T$	set of teachers (commodities).
$c \in C$	set of classes.
$v \in V$	set of all nodes.
$a \in A_t$	set of all arcs of the commodity t ($A_t \subset A$).
$a \in A_{tcdp}$	set of lesson arcs of the commodity t on class c, day d, and period p.
$a \in A^-_{tv}$	set of all arcs incoming node v for the commodity t.
$a \in A^+_{tv}$	set of all arcs outgoing node v for the commodity t.
$a \in Q^-_t$	set of all pull-in arcs for the commodity t.
$a \in Q^+_t$	set of all pull-out arcs for the commodity t.
$a \in Y_t$	set of all working day arcs of teacher t.
$a \in W_t$	set of all idle periods arcs of teacher t.
$a \in G_{tc}$	set of all double lesson arcs of teacher t and class c.
Parameters	
b_v	assumes 1 when v is the source, -1 when v is the sink, otherwise 0.
$H_{tc} \in \mathbb{N}$	number of lessons that teacher t must teach to class c.
$L_{tc} \in \{1, 2\}$	maximum daily number of lessons that teacher t can taught to class c.
$S_{ta} \in \{1, 2\}$	duration of arc a for the commodity t.
$M_{tc} \in \mathbb{N}$	minimum amount of double lessons required by teacher t on class c.
$Y'_t \in \mathbb{N}$	minimum amount of working days for teacher t.
$h \in \{0, 1\}$	indicates whether requirement H6 is take into account.
$\delta = 1$	cost of each required double lesson not satisfied.
$\omega = 3$	cost for each idle period.
$\gamma = 9$	cost for each working day.
Variables	
$x_{ta} \in \{0, 1\}$	indicates whether commodity t uses arc a.
$g_{tc} \geq 0$	number of unsatisfied double lessons of class c taught by professor t.

Source: created by author.

$$\textbf{Minimize} \sum_{t\in T}(\sum_{c\in C}\delta g_{tc} + \sum_{a\in W_t}\omega x_{ta} + \sum_{a\in Y_t}\gamma x_{ta}) \tag{5.1}$$

Subject to

$$\sum_{a\in A^+_{tv}} x_{ta} - \sum_{a\in A^-_{tv}} x_{ta} = b_v \qquad \forall t\in T, v\in V \tag{5.2}$$

$$\sum_{t\in T}\sum_{a\in A_{tcdp}} x_{ta} \leq 1 \qquad \forall c\in C, d\in D, p\in P \tag{5.3}$$

$$\sum_{a\in \bigcup_{d\in D, p\in P} A_{tcdp}} S_{ta}x_{ta} = H_{tc} \qquad \forall t\in T, c\in C \tag{5.4}$$

$$\sum_{a\in \bigcup_{p\in P} A_{tcdp}} S_{ta}x_{ta} \leq L_{tc} \qquad \forall t\in T, c\in C, d\in D \tag{5.5}$$

$$\sum_{a\in \bigcup_{p\in P} A_{tcdp}} x_{ta} \leq 1 \qquad \forall t\in T, c\in C, d\in D, h=1 \tag{5.6}$$

$$g_{tc} \geq M_{tc} - \sum_{a\in G_{tc}} x_{ta} \qquad \forall t\in T, c\in C \tag{5.7}$$

$$\sum_{a\in Y_t} x_{ta} \geq Y'_t \qquad \forall t\in T \tag{5.8}$$

$$x_{ta}\in\{0,1\} \qquad \forall t\in T, a\in A_t \tag{5.9}$$

$$g_{tc}\geq 0 \qquad \forall t\in T, c\in C \tag{5.10}$$

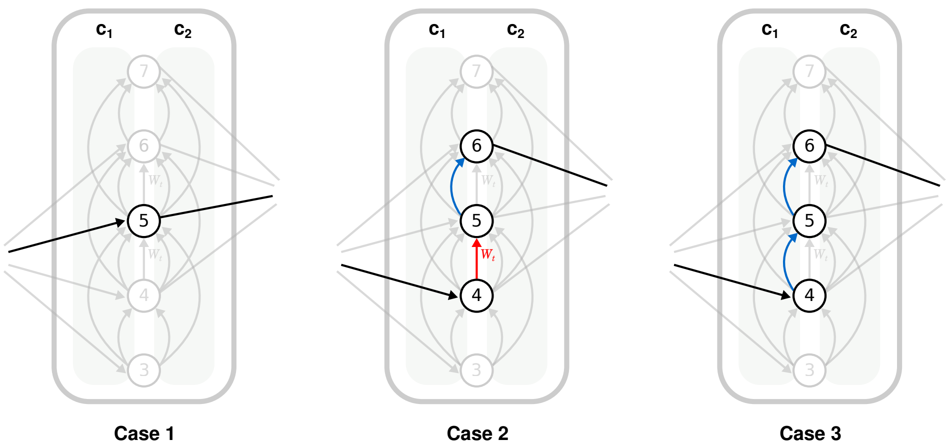
The objective function minimizes the violation of soft constraints. The flow conservation constraint set (5.2) ensures the total inflow equals the total outflow of each node (except source and sink), considering a given commodity t. Constraint set (5.3) ensures that the unitary capacity of the lesson arcs be respected. One can note that a structure of a multicommodity flow problem is represented by the constraint sets (5.2)-(5.3) and by the first two parts of the objective function (5.1). This structure is only able to address the requirements H2, H3, H4, S1, and S2. In order to model the remaining requirements, we included additional constraints and the last component of the objective function. Constraint set (5.4) ensures that the workload of each event is attended. Constraint set (5.5) ensures that the maximum number of daily lessons for each event is satisfied. Constraint set (5.6) ensures that lessons from the same event are scheduled in sequence by allowing the use of only one arc per day. This constraint is only activated when $h = 1$. Constraint set (5.7) computes the number of double lessons occurring in the solution. Constraint set (5.8) establishes a lower bound for the minimum number of working days for each teacher.

5.1.1 Additional cuts

Although the formulation $\mathcal{F}_1$ is suitable for representing the HSTP, due to the network structure, it eventually allows the construction of *unmeaningful paths* that overestimate the cost of sub-optimal solutions. Figure 5.3 illustrates three cases in which unmeaningful paths could occur.

In Case 1, the flow path crosses through the day component by using a single node. Since the path does not contain any lesson arc, the working day arc is used unnecessarily for accessing the day component. In Case 2, the solution cost is overestimated because an idle period arc is misused. Ideally, an idle arc should not appear in a path when succeeding a pull-in arc or preceding a pull-out arc. In Case 3, two single lessons are taught in sequence for the same class, while using a double lesson arc would be more appropriate. This situation can only occur when the requirement H6 is not being considered, i.e, $h = 0$. Otherwise, it is already avoided by the constraint set (5.6). In order to avoid these cases we can strengthen our formulation with the addition of some valid cut constraint sets (5.11)-(5.14). Constraint set (5.11) forbids the Case 1, constraint sets (5.12) and (5.13) forbid the Case 2, and constraint set (5.14) forbids the Case 3.

Figure 5.3 – Example of three different cases in which unmeaningful paths could be formed.



Source: Figure created by author.

$$x_{ti} + x_{tj} \leq 1 \quad \forall t \in T, v \in V, i \in Q_t^- \cap A_{tv}^-, j \in Q_t^+ \cap A_{tv}^+ \tag{5.11}$$

$$x_{ti} + x_{tj} \leq 1 \quad \forall t \in T, v \in V, i \in Q_t^- \cap A_{tv}^-, j \in W_t \cap A_{tv}^+ \tag{5.12}$$

$$x_{ti} + x_{tj} \leq 1 \quad \forall t \in T, v \in V, i \in Q_t^+ \cap A_{tv}^+, j \in W_t \cap A_{tv}^- \tag{5.13}$$

$$x_{ti} + x_{tj} \leq 1 \quad \forall t \in T, c \in C, d \in D, p \in P, i \in A_{tcdp}, j \in A_{tcdp+1}, \quad M_{tc} > 0, p < |P|, S_{ti} = S_{tj} = 1, h = 0 \tag{5.14}$$

5.2 Column Generation Applied to the HSTP

By applying the Dantzig-Wolfe decomposition principles (DANTZIG; WOLFE, 1960) on the compact formulation $\mathcal{F}_1$, we can obtain an alternative formulation for the HSTP, denoted as *Master Problem* (MP). In this formulation, stated by (5.15)-(5.18), let J_t be the set of all possible paths for a teacher t that satisfy all the hard requirements except H3. For each path $j \in J_t$ is associated a non-negative cost K_{tj} regarding the satisfaction of soft requirements. In addition, we define a binary variable λ_{tj} that indicates whether the path j is selected by teacher t.

$$\textbf{Minimize} \sum_{t \in T} \sum_{j \in J_t} K_{tj} \lambda_{tj} \tag{5.15}$$

subject to

$$\sum_{j \in J_t} \lambda_{tj} = 1 \qquad \forall t \in T \tag{5.16}$$

$$\sum_{t \in T} \sum_{j \in J_t} \sum_{a \in A_{tcdp}} \bar{x}_{taj} \lambda_{tj} \leq 1 \qquad \forall c \in C, d \in D, p \in P \tag{5.17}$$

$$\lambda_{tj} \in \{0,1\} \qquad \forall t \in T, j \in J_t \tag{5.18}$$

The objective of the MP, represented by Equation (5.15), is to minimize the cost of selected paths. Constraint set (5.16) ensures that exactly one path is selected for each teacher. Constraint set (5.17) ensures that the unitary capacity of the lesson arcs is respected, where $\bar{x}_{taj}$ indicates whether the arc a is used in path j of teacher t. By solving the MP, one can obtain an integer optimal solution to HSTP. However, this may be impracticable given the huge cardinality of J_t in problems faced in real applications. Instead, we propose to solve a linear relaxation of MP through a column generation approach, with the purpose to achieve tight lower bounds for the problem.

In a straightforward implementation, a column generation procedure starts with a master problem fulfilled with a restrict set of columns, hereafter called *Restricted Master Problem* (RMP). At each iteration, the RMP is solved and its dual variables are used to price out new columns by solving subproblems. During the resolution of each subproblem (pricing problem), columns with a negative reduced cost are added to the RMP. This procedure is repeated until no column with negative reduced cost is found. Precisely in our case, we consider the RMP stated by (5.19)-(5.23).

$$\textbf{Minimize} \sum_{t\in T}\Big(\sum_{j\in J_t} K_{tj}\lambda_{tj} + \varepsilon_t z_t\Big) \tag{5.19}$$

subject to

$$\sum_{j\in J_t} \lambda_{tj} + z_t = 1 \qquad \forall t \in T \tag{5.20}$$

$$\sum_{t\in T}\sum_{j\in J_t}\sum_{a\in A_{tcdp}} \bar{x}_{taj}\lambda_{tj} \le 1 \qquad \forall c \in C, d \in D, p \in P \tag{5.21}$$

$$\lambda_{tj} \ge 0 \qquad \forall t \in T, j \in J_t \tag{5.22}$$

$$z_t \ge 0 \qquad \forall t \in T \tag{5.23}$$

Note that variables are continuous and their upper bounds are implied by the constraint set (5.20). Besides, we chose to start the initial set of columns by introducing an artificial variable z_t for each teacher, that is penalized with a high cost in the objective function. As pointed out by LÜbbecke and Desrosiers (2005), assigning arbitrarily a too high cost to artificial variables may slowdown the convergence of the column generation. Thus, in order to keep the penalization as low as possible, we defined the cost of ε_t according to the Equation (5.24)

$$\varepsilon_t = \delta \sum_{c\in C} M_{tc} + \omega \max(0, |P|-2)|D| + \gamma|D|. \tag{5.24}$$

The cost ε_t is equal to the sum of three parts that correspond, respectively, to the upper bounds of the costs of the soft constraints S1, S2 and S3. In other words, ε_t is a trivial upper bound for the cost of a teacher path (K_{tj}).

After solving the RMP, the next step consists in a multiple pricing scheme, where the subproblem $\mathcal{P}_t$ is solved for each $t \in T$. Equations (5.25)-(5.33) present the formulation of $\mathcal{P}_t$:

$$\textbf{Minimize} \quad R_t = \sum_{c\in C} \delta g_{tc} + \sum_{a\in Y_t} \gamma x_{ta} + \sum_{a\in W_t} \omega x_{ta} - \sum_{c\in C}\sum_{d\in D}\sum_{p\in P}\sum_{a\in A_{tcdp}} \sigma_{cdp} x_{ta} - \pi_t \tag{5.25}$$

Subject to

$$\sum_{a\in A^+_{tv}} x_{ta} - \sum_{a\in A^-_{tv}} x_{ta} = b_v \qquad \forall v \in V \tag{5.26}$$

$$\sum_{a\in \bigcup_{d\in D, p\in P} A_{tcdp}} S_{ta} x_{ta} = H_{tc} \qquad \forall c \in C \tag{5.27}$$

$$\sum_{a\in \bigcup_{p\in P} A_{tcdp}} S_{ta} x_{ta} \leq L_{tc} \qquad \forall c \in C, d \in D \tag{5.28}$$

$$\sum_{a\in \bigcup_{p\in P} A_{tcdp}} x_{ta} \leq 1 \qquad \forall c \in C, d \in D, h = 1 \tag{5.29}$$

$$g_{tc} \geq M_{tc} - \sum_{a\in G_{tc}} x_{ta} \qquad \forall c \in C \tag{5.30}$$

$$\sum_{a\in Y_t} x_{ta} \geq Y'_t \tag{5.31}$$

$$x_{ta} \in \{0,1\} \qquad \forall a \in A_t \tag{5.32}$$

$$g_{tc} \geq 0 \qquad \forall c \in C \tag{5.33}$$

Assuming π_t and σ_{cdp} as dual variables associated, respectively, to the constraint sets (5.16) and (5.17), the reduced cost R_t is defined by Equation (5.25). Finally, observe that the remaining constraint sets, namely (5.26)-(5.33) are analogous to the ones presented in $\mathcal{F}_1$.

5.2.1 Speedup Strategies

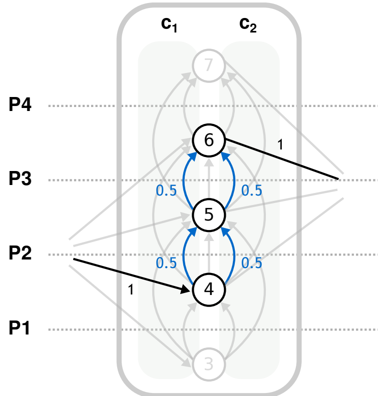
When comparing the expected computational effort required to solve the master and pricing problems, it is easy to predict that the bottleneck of the whole column generation process lies on the resolution of the pricing problem $\mathcal{P}_t$. Apart from being an integer problem, $\mathcal{P}_t$ also has to address the majority of the HSTP requirements. We resort to a MIP solver for solving it, however, one may note that even by using a state-of-the-art MIP solver, solving $\mathcal{P}_t$ to optimality might still be time consuming. This is particularly noteworthy with regard to the resolution of medium and large instances of the problem. Thus, in order to speedup the resolution of $\mathcal{P}_t$ with a MIP solver, we propose two trick strategies described next.

The first strategy is grounded in the principle that any column with negative reduced cost contributes to improve the objective value of the restricted master problem. Hence, $\mathcal{P}_t$ does not need to be solved exactly in every iteration since this is mandatory only in the last one. With this in mind, we design our column generation algorithm to operate in two sequential phases (I and II). In phase I, instead of solving a subproblem up to optimality, we stop the solver as soon as it finds a feasible solution proved to be within a given percentual α far from optimal. When no more solutions with $R_t < 0$ can be generated using the current value of α, the algorithm switches to phase II where α is set to zero and, consequently, the subproblems are solved to optimality.

The second speedup strategy consists in solving a relaxed version of $\mathcal{P}_t$, hereafter denoted by $\mathcal{P}'_t$, where the integrality constraints are enforced only for the variables associated to pull-in and pull-out arcs. In other words, it means that $\mathcal{P}'_t$ is able to precisely determine the first and last lesson periods for each working day of a teacher. However, as consequence of the relaxation, it may be not possible to identify exactly in which class a lesson should be given, as illustrated in Figure 5.4.

In spite of losing some information, the relaxation decreases considerably the number of integer variables of the subproblem. While $\mathcal{P}_t$ has a large number of integer variables that depends on the number of events, the $\mathcal{P}'_t$, uses only $|Q_t^- \cup Q_t^+| \times |D| = 2|P| \times |D|$ binary variables. In practical instances, $|D|$ is typically limited by 6 (Monday to Saturday) and $|P|$ hardly exceeds 20, even in schools holding full-day programs. In that sense, it is safe to claim that for practical purposes, the number

Figure 5.4 – Example of a relaxed subproblem solution in a working day. The number next to arcs represents the respective value flow. We can observe that the integral flows passing through the pull-in and pull-out arcs determine whether the respective teacher will teach from period P2 to P3. However, the flow is split into lesson arcs and we cannot determine for each period whether a lesson should be given either to class c_1 or c_2.



Source: Figure created by author.

of integer variables of $\mathcal{P}'_t$ is constrained between $12|P|$ and 240 for any real-world instance.

Finally, it is important to point out that even $\mathcal{P}'_t$ is less restrict than $\mathcal{P}_t$, as we show further in the computational results, both problems often find the same objective value, i.e., $\mathcal{R}'_t \lesssim \mathcal{R}_t$. As a result, the lower bound obtained by the column generation using $\mathcal{P}'_t$ is strongly close to the one obtained by using $\mathcal{P}_t$. We refer the column generation method that uses $\mathcal{P}_t$ in the pricing step as Integer Pricing Column Generation (`IPCG`), while the version that uses $\mathcal{P}'_t$ is identified as Relaxed Pricing Column Generation (`RPCG`).

5.3 Computational experiments

In this section we present an experimental evaluation for the proposed models and methods. The problems are solved by CPLEX 12.6.0 (IBM, 2015) with default settings but using a single core. The algorithms were implemented in C++ using the compiler g++ 4.6.1. The experimental results were computed in a Desktop-PC equipped with an Intel Core i5-2300 processor clocked at 2.8GHz, 4GB of RAM, over a 64 bits Linux operating system. Along this section, we report results of one run for each tested algorithm, since they are deterministic. The mathematical model parameters γ, ω, and δ were set to 9, 3, and 1, respectively. To the best of our knowledge these values are the same used by all previous works in HSTP.

5.3.1 Dataset

To evaluate the algorithm we used the instances presented in Table 5.2. In this table, the first two columns present the instance identifier and the name. Since their names are long, we use the identifiers for shortening reference along the text. Columns $|D|$ and $|P|$ show the number of days and periods, respectively, while columns $|T|$, $|C|$ and $|E|$ present the number of teachers, classes and events, respectively, where $E = \{(t, c) : t \in T, c \in C, H_{tc} > 0\}$. Columns $\sum_{(t,c)\in E} M_{tc}$ and $\sum_{(t,c)\in E} H_{tc}$ present the total double lessons required and the total workload, respectively. Finally, column BKV presents the best known values for these instances, all of them were obtained in Chapter 4.

The instances are split into two sets. Instances 1-7 comprise set-1 and are available from the repository (LABIC, 2008) and to our knowledge were used in all the previous works in HSTP. Requirement (H6) is not considered in this group of instances. Instances A, D, E, F, G, from set-2, are different versions of instances 1, 4, 5, 6, 7, respectively. They differ mainly in two aspects: in set-2, teachers are available in all periods, and requirement (H6) is considered. These modifications made the instances more challenging and were used in the first round of the Third International Timetabling Competition 2011 (ITC, 2011). They are part of the XHSTT-2012 archive that is available at <http://www.utwente.nl/ctit/hstt/archives/XHSTT-2012/>.

The next subsections have the aim of presenting results of the algorithms and models presented in this work, as well as comparing them with the previous state-of-the-art results for the problem.

5.3.2 Integer solutions obtained by MIP models

The first experiment aims at comparing results between the model $\mathcal{M}_1$ (presented in Section 3.1) and the flow model $\mathcal{F}_1$ proposed in this chapter. Each instance was run for at most 7200s (2h) for each model. Table 5.3 presents for each instance and MIP model, the objective value found (*obj*), the running times (*time*), the number of columns (*#col*) and rows (*#row*) generated by the respective model, the gap

Table 5.2 – Main characteristics of the tested instances.

Id	Name	$\|D\|$	$\|P\|$	$\|T\|$	$\|C\|$	$\|E\|$	$\sum_{(t,c)\in E} M_{tc}$	$\sum_{(t,c)\in E} H_{tc}$	BKV
1	Inst1	5	5	8	3	21	21	75	202
2	Inst2	5	5	14	6	63	29	150	333
3	Inst3	5	5	16	8	69	4	200	423
4	Inst4	5	5	23	12	127	66	300	652
5	Inst5	5	5	31	13	119	71	325	762
6	Inst6	5	5	30	14	140	63	350	756
7	Inst7	5	5	33	20	205	84	500	1017
A	BrazilInstance1	5	5	8	3	21	21	75	200
D	BrazilInstance4	5	5	23	12	127	66	300	648
E	BrazilInstance5	5	5	31	13	119	71	325	776
F	BrazilInstance6	5	5	30	14	140	63	350	779
G	BrazilInstance7	5	5	33	20	205	84	500	1066

Source: created by author.

provided by CPLEX (gap_C) and the gap (gap_B) from the best known value calculated as $100 * (obj - BKV)/min(BKV, obj)$. The lowest values for each column are shown in boldface.

From the 12 instances, the flow model was able to find better results in six instances, and similar results in other two instances. The model $\mathcal{M}_1$ found better results than $\mathcal{F}_1$ only in four instances. Although that, on average, $\mathcal{M}_1$ found better gap results. This is mainly due to the results obtained for instance 7 in which $\mathcal{F}_1$ found a solution with an objective cost considerable higher than one found by the model $\mathcal{M}_1$. In addition, the number of columns reported to $\mathcal{F}_1$ was higher than $\mathcal{M}_1$ in only three instances. Regarding the number of rows, the model $\mathcal{M}_1$ obtained highest reported values in all instances. On the whole, the results of both models appear to be comparable. However, it can be noted that $\mathcal{F}_1$ was able to proof optimality for two instances, as well as tigther values for gap_C in the most of instances. These results suggest that the relaxation of the flow model may provide a better lower bound than the relaxation of $\mathcal{M}_1$. This hypothesis is evaluated in the next experiment.

Table 5.3 – Comparison results between models $\mathcal{M}_1$ and $\mathcal{F}_1$ with a time limit of 2 hours.

	$\mathcal{M}_1$						$\mathcal{F}_1$					
Id	obj	*time*	#col	#row	gap$_C$	gap$_B$	obj	*time*	#col	#row	gap$_C$	gap$_B$
1	**202**	**2h**	1306	2914	6.44	**0.00**	**202**	**2h**	**1163**	**1187**	**5.82**	**0.00**
2	340	2h	3100	6855	2.06	2.10	**333**	**196s**	**3021**	**2480**	**0.00**	**0.00**
3	**426**	**2h**	2898	6532	**2.82**	**0.71**	429	**2h**	**2424**	**2305**	3.50	1.42
4	653	2h	5221	11642	2.14	0.15	**652**	**988s**	**4174**	**4081**	**0.00**	**0.00**
5	782	**2h**	**6349**	14057	3.32	2.62	**777**	**2h**	6602	**6613**	**2.70**	**1.97**
6	**780**	**2h**	**6850**	15153	**5.38**	**3.17**	804	**2h**	7000	**6653**	8.02	6.35
7	**1043**	**2h**	**9155**	20242	**4.22**	**2.56**	1645	**2h**	9364	**8805**	38.81	61.75
A	**200**	**2h**	2011	3910	5.50	**0.00**	**200**	**2h**	**1566**	**1513**	**4.00**	**0.00**
D	735	**2h**	10132	19008	15.51	13.43	**726**	**2h**	**7168**	**7571**	**12.12**	**12.04**
E	868	**2h**	10244	19513	12.90	11.86	**812**	**2h**	**7568**	**9594**	**5.36**	**4.64**
F	1174	**2h**	11530	21772	37.14	50.71	**952**	**2h**	**8312**	**10012**	**20.69**	**22.21**
G	**1248**	**2h**	16200	30328	19.95	**17.07**	1285	**2h**	**11380**	**14121**	**19.88**	20.54
Avg.	**704**	2h	7083	14327	**9.78**	**8.70**	735	**6099**	**5812**	**6245**	10.08	10.91

Source: created by author.

5.3.3 Lower bounds for the problem

This section aims at presenting and comparing lower bound results provided by the linear relaxation of $\mathcal{F}_1$ (denoted as $\mathcal{F}_1'$), the lower bound found by the linear relaxation of model $\mathcal{M}_1$ (denoted as $\mathcal{M}_1'$) , and the method `IPCG` proposed in this work with $\alpha = 0\%$, i.e., the column generation method without any speedup strategy. Table 5.4 presents for each instance and method, the lower bound found (LB), the running times (time) in seconds, and the percentage deviation from the best known value to the lower bound gap_L, which is computed as $100 * (BKV - LB)/LB$. Best results for each column are shown in boldface.

In summary, it can be observed from the results that $\mathcal{F}_1'$ is the fastest method whereas `IPCG` provides the best lower bounds considering all instances tested. Both approaches, $\mathcal{F}_1'$ and `IPCG`, present significant improvements in comparison with $\mathcal{M}_1'$. It can be seen that $\mathcal{F}_1'$ provides better or equal lower bounds than $\mathcal{M}_1'$, using about four times less time, on average. Although `IPCG` spent about twice the time of $\mathcal{M}_1'$, the gap_L achieved by the former is approximately ten times shorter. When comparing IPCG and $\mathcal{F}'_1$ the gap_L found by IPCG is considerable better, but it takes longer to run.

Table 5.4 – Comparison results of lower bounds provided by $\mathcal{F}'_1$, $\mathcal{M}'_1$ and IPCG.

	$\mathcal{F}'_1$			$\mathcal{M}'_1$			IPCG ($\alpha = 0\%$)		
Id	LB	time (s)	gap$_L$ (%)	LB	time (s)	gap$_L$ (%)	LB	time (s)	gap$_L$ (%)
1	189	**0.1**	6.88	189	**0.1**	6.88	**202**	11.3	**0.00**
2	**333**	**0.4**	**0.00**	**333**	1.1	**0.00**	**333**	8.7	**0.00**
3	414	**0.2**	2.17	414	0.9	2.17	**423**	12.0	**0.00**
4	643	**0.8**	1.40	639	2.2	2.03	**652**	10.0	**0.00**
5	756	**2.3**	0.79	756	6.8	0.79	**762**	13.5	**0.00**
6	738	**2.8**	2.44	738	9.0	2.44	**756**	40.5	**0.00**
7	999	**7.8**	1.80	999	24.1	1.80	**1017**	25.5	**0.00**
A	190	**0.1**	5.26	189	0.2	5.82	**200**	6.7	**0.00**
D	635.5	**3.8**	1.97	621	14.1	4.35	**646**	26.4	**0.31**
E	767.5	**3.9**	1.11	756	13.8	2.65	**775**	20.2	**0.13**
F	754	**5.5**	3.32	738	23.0	5.56	**773**	45.2	**0.78**
G	1023	**12.0**	4.20	999	77.2	6.71	**1039**	85.1	**2.60**
Avg.	620	**3.3**	2.61	614	14.5	3.43	**632**	25.4	**0.32**

Source: created by author.

5.3.4 Parameter testing for the proposed column generation algorithm

In this section we evaluate the performance of different settings and speedup strategies for the proposed column generation. Table 5.5 presents average results regarding to IPCG and RPCG with several values for α (presented in Section 5.2.1). For both methods are reported the total running times (*time*) in seconds, the total number of columns generated (*#col*), the total number of iterations (*#iter*), and the percentual of the total time spent in the pricing step (*pr*). The shortest running time is shown in boldface. Values shown inside parenthesis next to the columns *#col* and *#iter* present the number of columns/iterations that were inserted/performed in the phase II of the algorithm. We recall that when α is set to zero, the phase II is not required to be performed.

Table 5.5 – Average results for all instances comparing different settings for the proposed column generation.

	IPCG				RPCG			
α (%)	time (s)	#col	#iter	pr (%)	time (s)	#col	#iter	pr (%)
0	25.4	33	784	94.7	15.7	33	782	91.7
1	24.8	34 (1)	778	95.1	15.0	34 (1)	781	91.3
2	21.4	35 (1)	780	94.2	13.9	35 (1)	805	90.3
3	19.7	35 (1)	781	93.8	14.0	37 (1)	830	89.6
4	19.4	36 (1)	789	93.7	**13.8**	38 (1)	847	89.2
5	18.9	36 (1)	795	93.4	14.2	40 (1)	891	88.7
6	19.0	37 (1)	799	93.3	14.6	42 (1)	927	88.2
7	18.5	37 (1)	806	92.9	15.1	45 (1)	942	88.1
8	18.2	38 (1)	798	92.8	15.9	47 (1)	987 (3)	87.3
9	18.0	39 (1)	803	92.8	15.5	47 (1)	996	86.8
10	18.6	40 (1)	822 (1)	92.8	16.9	51 (1)	1039 (1)	86.3
20	24.2	53 (3)	893 (16)	92.6	17.0	58 (1)	1074 (6)	85.8
30	41.5	101 (5)	1027 (69)	93.2	32.7	105 (2)	1441 (27)	85.7
40	44.0	112 (8)	990 (157)	94.2	39.8	129 (5)	1333 (76)	87.4
50	35.0	79 (13)	873 (266)	94.5	31.6	124 (9)	1104 (166)	89.8
60	32.1	65 (17)	830 (350)	94.8	31.4	128 (11)	1082 (225)	90.2
70	35.3	77 (18)	823 (371)	95.2	34.1	144 (11)	1078 (217)	90.6
80	35.9	79 (20)	809 (413)	95.8	32.7	134 (12)	1061 (236)	91.2
90	35.9	77 (23)	827 (464)	95.6	29.6	124 (13)	1043 (259)	91.0

Source: created by author.

We would like to highlight that regardless the use of a relaxed pricing, all lower bounds generated by `RPCG` matched exactly the ones obtained by `IPCG` in Table 5.4 (we further discuss this topic in the next subsection). Thus, results shown in Table 5.5 are focussed in presenting their differences in terms of running times and number of iterations.

From the table, it is noteworthy that `RPCG` is faster than `IPCG` when the same value of α is considered. Both algorithms are affected similarly according to changes in the parameter α, taking longer when $\alpha = 0\%$ and $\alpha > 10\%$. In these cases, the slowdown is caused due to the quality of columns generated in the pricing step. In one hand, when $\alpha = 0\%$, extra computational time is spent to generate high quality columns by ensuring optimality for each pricing problem. In the other hand, when $\alpha > 10\%$, although the pricing step runs faster, it adds a higher number of low quality columns into the master, what increases the number of iterations required to reach optimality, specially in the phase II.

A suitable setting for α, which provides a good trade-off between quality and computational effort for generating a column, is comprised with $1\% \leq \alpha \leq 10\%$. In this range, we found the best overall results achieved by `RPCG` with $\alpha = 4\%$, that combines the two acceleration strategies proposed in Section 5.2.1. However, considering that the speedups for `IPCG` ($\alpha = 9\%$), `RPCG` ($\alpha = 0\%$) and `RPCG` ($\alpha = 4\%$) calculated over `IPCG` ($\alpha = 0\%$) are, respectively, 1.41, 1.61 and 1.84, it can be observed that the proposed acceleration strategies are able to improve significantly the convergence of the column generation even if used exclusively. In fact, the relaxed pricing strategy can provide a higher speedup than introducing an $\alpha > 0\%$. When both strategies are used together, the results are slightly better.

5.3.5 Objective values provided by $\mathcal{P}_t$ and $\mathcal{P}'_t$

In this section, we evaluate empirically the level of approximation provided by $\mathcal{P}'_t$ in comparison with $\mathcal{P}_t$. Since $\mathcal{P'}_t$ is a relaxation of $\mathcal{P}_t$, we can denote the difference between the optimal reduced cost of these problems by $\Delta = R_t - R'_t$. In order to measure the magnitude of this difference, we ran `IPCG` using $\alpha = 0\%$. During the pricing step, besides solving $\mathcal{P}_t$ we also solved $\mathcal{P}'_t$ for the sake of computing the value of Δ. We report in Table 5.6, for each instance, the total number of pricing problems solved (*#pricing*). Moreover, for the cases of $\Delta > 0$, $\Delta \leq 1$, and $\Delta > 1$, we report the number of occurrences of each of these cases, and the percentage of these occurrences considering the total number of pricings solved. Finally, the maximum and the average values of Δ are given in the last two columns.

Analyzing the table one may note that, on average, only 13.84% of the pricing problems solved revealed a difference between $\mathcal{P}_t$ and $\mathcal{P}'_t$. However, almost 100% of the differences are, on average, less or equal to one unit cost (see column $\Delta \leq 1$). In addition, among all runnings, only three pricing problems resulted in a difference higher than one unit cost (see column $\Delta > 1$). In these rare cases, the value of Δ still barely surpassed one unit cost.

As shown in the last column of the table, the average Δ is only 0.03, thus meaning that the difference between values computed by $\mathcal{P}'_t$ and $\mathcal{P}_t$ is tiny. In fact, 0.03 corresponds to a value which is about 33 times smaller than the least cost penalty associated to an unsatisfied double lesson ($\delta = 1$). We attribute to this tiny difference the fact that lower bounds obtained by `IPCG` and `RPCG` are the same, as mentioned in the previous section.

Table 5.6 – Results presenting the difference (Δ) between the reduced costs provided by $\mathcal{P}_t$ and $\mathcal{P}'_t$.

Id	#pricing	$\Delta > 0$	(%)	$\Delta \leq 1$	(%)	$\Delta > 1$	(%)	max(Δ)	avg(Δ)
1	232	51	(21.98)	231	(99.57)	1	(0.43)	1.08	0.06
2	420	37	(8.81)	420	(100.00)	0	(0.00)	0.89	0.02
3	416	5	(1.20)	416	(100.00)	0	(0.00)	0.56	0.00
4	782	66	(8.44)	782	(100.00)	0	(0.00)	1.00	0.02
5	806	74	(9.18)	806	(100.00)	0	(0.00)	0.50	0.01
6	1020	101	(9.90)	1020	(100.00)	0	(0.00)	0.38	0.01
7	1023	114	(11.14)	1023	(100.00)	0	(0.00)	0.61	0.02
A	248	5	(2.02)	248	(100.00)	0	(0.00)	0.83	0.00
D	759	161	(21.21)	758	(99.87)	1	(0.13)	1.18	0.04
E	1023	146	(14.27)	1023	(100.00)	0	(0.00)	0.85	0.03
F	1320	270	(20.45)	1320	(100.00)	0	(0.00)	1.00	0.05
G	1650	619	(37.52)	1649	(99.94)	1	(0.06)	1.12	0.07
Avg.	808	137	(13.84)	808	(99.95)	0.25	(0.05)	0.83	0.03

Source: created by author.

5.3.6 Comparison between the proposed method and a Cut and Column Generation approach

In this section we compare our column generation method with the approach proposed by Santos et al. (2012), hereafter referred to as Cut and Column Generation (`CCG`). We used their original `CCG` implementation, which was kindly provided by the authors. In order to compare both approaches in instances of set-2, we included the requirement H6 in their implementation. Table 5.7 presents results for each instance comparing the performance of `CGG` and `RPCG` using $\alpha = 4\%$. For both methods are reported the total running times (*time*) in seconds, the total number of columns generated (*#col*), and the total number of iterations (*#iter*) performed. Column *LB* presents the lower bound values computed by both methods, column $\mathcal{F}_1'$ presents the lower bound produced by the linear relaxation of the compact model $\mathcal{F}_1$, column *gap* reports the optimality gap for each instance and finally, column *speedup* presents the speedup of `RPCG` over `CCG`. Column *gap* is computed by $100*(BKV-LB)/LB$. Column *speedup* is computed by `CCG`/`RPCG`. Results with shortest running time are shown in boldface. Values marked with (*) are new best lower bounds in comparison with results presented in Table 4.5.

Table 5.7 – Comparison results between the lower bounds provided by the Cut and Column Generation (`CCG`) proposed by (SANTOS et al., 2012) and the proposed Relaxed Pricing Column Generation (`RPCG`).

				CCG			RPCG			
Id	LB	$\mathcal{F}_1'$	gap (%)	time (s)	#col	#iter	time (s)	#col	#iter	speedup
1	202	189	0.00	**0.17**	351	15	3.26	248	32	0.05
2	333	333	0.00	**5.12**	960	30	6.34	476	35	0.81
3	423	414	0.00	**2.22**	916	28	4.24	414	27	0.52
4	652	643	0.00	40.25	1474	44	**6.64**	735	37	6.06
5	762	756	0.00	34.43	1888	35	**11.29**	813	28	3.05
6	756	738	0.00	72.25	2102	56	**13.37**	835	29	5.41
7	1017	999	0.00	395.63	2284	67	**16.30**	1020	32	24.27
A	200*	190	0.00	**0.33**	443	19	3.25	232	30	0.10
D	646*	635.5	0.31	50.74	1524	44	**14.35**	847	40	3.54
E	775*	767.5	0.13	97.30	1918	73	**20.36**	1184	53	4.78
F	773*	754	0.78	34.49	1865	37	**23.53**	1338	49	1.47
G	1039*	1023	2.60	451.38	2740	78	**42.22**	2026	63	10.69
Avg.	631	620	0.32	98.69	1539	44	**13.76**	847	38	5.06

Source: created by author.

According to the results, besides `CCG` and `IPCG` are able to provide the same lower bounds that are tighter than the ones provided by the relaxation of model $\mathcal{F}_1$, they present a distinct performance according to the instance size. While `CCG` achieves short running times on four small instances (1,2,3 and A), `RPCG` escalates better, performing faster on the remaining eight medium and large instances. In addition, our method is approximately 5 times faster than `CCG`, on average, and particularly in the largest instances, the performance improvement becomes more preeminent, being about 24 times faster on instance 7, and about 10 times faster on instance G.

Besides finding optimal bounds for all instances of set-1, we were able to find new tighter lower bounds for all instances of set-2. These new results allow to reduce the average optimality gap for theses instances from 2.09%, as presented in Chapter 4, to 0.32%. Moreover, for the first time, optimality was proved for the instance A since the lower bound found matched the best known value.

5.4 Conclusions

In this chapter we tackle the HSTP, which is a well-known variant of the High School Timetabling Problem originated from Brazilian schools. This problem was considered in the Third International Timetabling Competition held in 2011. In addition to a novel mathematical programming formulation based on a multi commodity flow network for the HSTP, we proposed a column generation approach, using two speedup strategies, for proving strong lower bounds for this problem.

In comparison with the state-of-the-art column generation for HSTP, the experimental results show that our approach is able to produce the same lower bounds, albeit with two significant advantages: i) the method is simpler; ii) and it is five times faster on average. Moreover, we improved best known lower bounds of 5 out of 12 instances from the literature. Among these new results, one is proved to be optimal (namely Instance A). These results show that the proposed technique is efficient for producing lower bounds for the HSTP, motivating its use to other variants of this problem.

6 HSTP$^+$: EXTENDED HSTP AND A NOVEL SET OF BENCHMARK INSTANCES

In this chapter we introduce a new High School Timetabling Problem referred as HSTP$^+$, originated from 33 real-world Brazilian instances we have collected during this research. We modelled HSTP$^+$ in such a way it makes a bridge between HSTP and GHSTP, i.e, the problems obey the following relationship: HSTP $\subset$ HSTP$^+$ $\subset$ GHSTP. This relationship allows us not only expand and validate the methods proposed for HSTP by using a large instance set, but also gives us the opportunity to perform a comparison between the methods proposed in this thesis and the state-of-the-art approaches designed for solving GHSTP. The remainder of this chapter is organized as follows. Firstly, in Section 6.1 we describe the problem formally through a MIP formulation. Section 6.2 describes how we converted HSTP$^+$ into GHSTP by using the constraints available in the XHSTT format. Section 6.3 presents an extensive set of computational experiments in order to demonstrate empirically the effectiveness of the fix-and-optimize approach applied on HSTP$^+$. Finally, Section 6.4 presents a summary of the major conclusions we draw for this chapter.

6.1 Problem definition

In this section we formally define and introduce a compact formulation for the Extended High School Time Timetabling Problem (HSTP$^+$). The goal of the problem is to build a timetable for a week organized as a set of days D. Each day is split into a set of shifts K, and each shift is split into a set of periods P. In this problem, we call as *time slot* a tuple composed of a day, a shift, and class period,

(d, k, p), with $d \in D$, $k \in K$, and $p \in P$, wherein all periods have the same duration. For the sake of compactness, in the formulation we refer to timeslots using a single index s, where each $s \in S$ corresponds to a distinct timeslot tuple.

The problem considers three types of resources: a set of classes C, a set of teachers T and a set of shared rooms R. Both teachers $t \in T$ and classes $c \in C$ may be unavailable in a given set of timeslots. The main input for the problem is a set of events $e \in E$ that should be assigned to timeslots. Typically, an event is a meeting between resources, i.e, a meeting between a teacher and a class to address a particular subject (e.g. Biology) in a dedicated room. While the majority of events are composed by at most one resource of each type, events with multiple resources of the same type are often required.

The parameter W_e determines the *workload* of an event $e \in E$, i.e., the number of lessons that need to be distributed in a week. A distinct timeslot $s \in S$ must be assigned for each lesson of an event. In addition, mainly due to pedagogical demands, several requirements are defined to events in order to impose daily and shift limits for lessons, and also to ensure a given distribution of lessons along the week.

A feasible solution for an HSTP$^+$ instance must satisfy all the hard requirements H_1-H_7 presented below:

- H_1 The workload defined in each event must be satisfied.
- H_2 A teacher cannot be scheduled to more than one lesson in a given timeslot.
- H_3 Lessons cannot be taught to the same class in the same timeslot.
- H_4 A teacher cannot be scheduled to a timeslot in which he/she is unavailable.
- H_5 A class cannot be scheduled to a timeslot in which it is unavailable.
- H_6 Shared rooms cannot be used by distinct events in the same timeslot.
- H_7 Avoid idle times in class shifts.

In addition to feasibility, the violation of the soft requirements S_1-S_4 presented below should be minimized:

- S_1 The maximum number of lessons by shift of each event must be respected.
- S_2 Minimize the number of working shifts for teachers.
- S_3 Minimize the number of working days for teachers.
- S_4 Avoid teachers' idle periods on shifts.

Finally, HSTP$^+$ should take into account the medium requirements M_1-M_6 presented next:

M_1 The maximum number of daily lessons of each event should be respected.

M_2 Event lessons should be consecutive when scheduled on the same day.

M_3 Event lessons should not be consecutive when scheduled on the same day.

M_4 Events demand a specific distribution of blocks in the week.

M_5 Teachers cannot be scheduled to more than a given number of shifts per day.

M_6 Some teachers have mandatory working days.

Medium requirements are considered either hard or soft depending on the instance.

6.1.1 Formulation

In this section we formally define HSTP$^+$ through a MIP formulation. The full notation is presented in Table 6.1 along with a description of the parameters and variables defined for the problem. In order to provide a more compact notation we adopted a different convention for the exponentiation operator. Instead of the usual meaning, we use exponentiation for removing elements of a given set. Let S be a set where the order of the elements is important. An operation S^a, with a positive a, results in a subset of S in which the first a elements are removed. If a negative exponent is used, the last a elements of S are removed. For example, assuming $S = \{1,2,3,4,5\}$, then $S^{+2} = \{3,4,5\}$ and $S^{-2} = \{1,2,3\}$.

Table 6.1 – Notation used in the HSTP$_+$ model.

Symbol	Definition
Sets	
$d \in D$	days of week. $D = \{1, 2, ..., \vert D\vert\}$.
$k \in K$	shifts of day. $K = \{1, ..., \vert K\vert\}$.
$p \in P$	periods of shift. $P = \{1, ..., \vert P\vert\}$.
$s \in S$	timeslots of week. $S = \{1, ..., \vert S\vert\}$.
$s \in S_d$	timeslots of day d. $S_d \subseteq S$.
$s \in S_{dk}$	timeslots of shift k on day d. $S_{dk} \subseteq S_d$.
$t \in T$	set of teachers.
$c \in C$	set of classes.
$e \in E$	set of events.
$r \in R$	set of shared rooms.
E_t	set of events assigned to teacher t.
E_c	set of events assigned to class c.
E_r	set of events assigned to resource r.
U_{dk}	set of tuples (m,n) for $m \in S_{dk}^{-2}, n \in S_{dk} : n \geq m+2$.
Q_{dk}	set of tuples (m,n) for $m \in S_{dk}^{-2}, n \in S_{dk} : n \geq m$.
π_p	set of timeslots where an event can starts a lesson block of size p.
Parameters	
ω	cost of each idle period of teacher t.
γ	cost of each working day/shift of teacher t.
δ	cost of each lesson exceeding the limit L''_e of event e.
μ	cost of a violation of any medium requirement.
F_{td}	binary parameter that indicates whether teacher t must work on day d.
Y_t	maximum daily number of working shifts of teacher t.
W_e	workload of event e.
L'_e	maximum daily number of lessons of event e.
L''_e	maximum number of lessons of event e in a shift.
G_{ep}	minimum number of blocks of size p required by event e.
V_{ts}	binary parameter that indicates whether the teacher is available in the timeslot s.
V_{cs}	binary parameter that indicates whether the class is available in the timeslot s.
Variables	
$x_{es} \in \mathbb{B}$	active when event e is scheduled to timeslot s.
$a_{cs} \in \mathbb{B}$	active when class c has a lesson at timeslot s and not at timeslot $s-1$.
$b_{es} \in \mathbb{B}$	active when event e has a lesson at timeslot s and not at timeslot $s-1$.
$g_{esp} \in \mathbb{B}$	active when a block of size p starts at timeslot s for event e.
$y'_{td} \in \mathbb{B}$	active when at least one lesson is assigned to teacher t on day d.
$y''_{tdk} \in \mathbb{B}$	active when at least one lesson is assigned to teacher t on day d and shift k.
$l_{edk} \geq 0$	number of lessons of event e exceeding the limit L''_e on day d and shift k.
$z_{tdkmn} \in \mathbb{B}$	active when teacher t has idle periods on day d, on shift k, between timeslots m and n.
$v_{ed}^{\mathrm{M}_1} \geq 0$	number of lessons of event e exceeding L'_e on day d.
$v_{ed}^{\mathrm{M}_{2a}} \geq 0$	number of nonconsecutive lesson blocks of event e on day d.
$v_{edk}^{\mathrm{M}_{2b}} \geq 0$	assumes 1 when a block of event e is assigned between shifts k and $k+1$ on day d.
$v_{es}^{\mathrm{M}_3} \geq 0$	assumes 1 when a lesson of event e in the timeslot s precedes other lesson of event e.
$v_{ep}^{\mathrm{M}_4} \geq 0$	number of blocks of size p remaining to reach the amount G_{ep} required to event e.
$v_{td}^{\mathrm{M}_5} \geq 0$	number of working shifts exceeding the limit Y_t of teacher t.
$v_{td}^{\mathrm{M}_6} \geq 0$	assumes 1 when no lesson is assigned to teacher t on day d and $F_{td} = 1$.

Source: created by author.

Minimize (6.1)

$$\delta\sum_{e\in E}\sum_{d\in D}\sum_{k\in K} l_{edk}+\sum_{t\in T}\sum_{d\in D}\Big(\gamma y'_{td}+\sum_{k\in K}(\gamma y''_{tdk}+\sum_{(m,n)\in U_{dk}}\omega(n-m-1)z_{tdkmn})\Big)+$$
$$\mu\sum_{e\in E}\Big(\sum_{d\in D}(v^{\mathrm{M}_1}_{ed}+v^{\mathrm{M}_{2a}}_{ed}+\sum_{k\in K^{-1}}v^{\mathrm{M}_{2b}}_{edk}+\sum_{k\in K}\sum_{s\in S^{-1}_{dk}}v^{\mathrm{M}_3}_{es})+\sum_{p\in P}v^{\mathrm{M}_4}_{ep}\Big)+\mu\sum_{t\in T}\sum_{d\in D}(v^{\mathrm{M}_5}_{td}+v^{\mathrm{M}_6}_{td})$$

Subject to

$$\sum_{s\in S} x_{es}=W_e \qquad \forall e\in E \tag{6.2}$$
$$\sum_{e\in E_t} x_{es}\le 1 \qquad \forall t\in T, s\in S \tag{6.3}$$
$$\sum_{e\in E_c} x_{es}\le 1 \qquad \forall c\in C, s\in S \tag{6.4}$$
$$\sum_{e\in E_r} x_{es}\le 1 \qquad \forall r\in R, s\in S \tag{6.5}$$
$$x_{es}\le V_{ts} \qquad \forall t\in T, e\in E_t, s\in S \tag{6.6}$$
$$x_{es}\le V_{cs} \qquad \forall c\in C, e\in E_c, s\in S \tag{6.7}$$
$$\sum_{e\in E_c} a_{cs}\ge \sum_{e\in E_c}(x_{es}-x_{es-1}) \qquad \forall c\in C, d\in D, k\in K, s\in S^{+1}_{dk} \tag{6.8}$$
$$\sum_{s\in S^{+1}_{dk}} a_{cs}+\sum_{e\in E_c} x_{ei}\le 1 \qquad \forall c\in C, d\in D, k\in K, i=S_{dk1} \tag{6.9}$$
$$\sum_{s\in S_d} x_{es}\le L'_e+v^{\mathrm{M}_1}_{ed} \qquad \forall e\in E, d\in D \tag{6.10}$$
$$b_{es}\ge x_{es}-x_{es-1} \qquad \forall e\in E, d\in D, s\in S^{+1}_d \tag{6.11}$$
$$\sum_{s\in S^{+1}_d} b_{es}+x_{ei}\le 1+v^{\mathrm{M}_{2a}}_{ed} \qquad \forall e\in E, d\in D, i=S_{d11} \tag{6.12}$$
$$x_{ei}+x_{ei-1}\le 1+v^{\mathrm{M}_{2b}}_{edk} \qquad \forall e\in E, d\in D, k\in K^{-1}, i=S_{dk+1,1} \tag{6.13}$$
$$x_{es}+x_{es+1}\le 1+v^{\mathrm{M}_3}_{es} \qquad \forall e\in E, d\in D, k\in K, s\in S^{-1}_{dk} \tag{6.14}$$
$$g_{esp}\le x_{es+i-1} \qquad \forall e\in E, p\in P, s\in \pi_p, i\in[p] \tag{6.15}$$
$$g_{esp}\le 1-x_{es-1} \qquad \forall e\in E, d\in D, k\in K, p\in P, s\in \pi_p\cap S^{+1}_{dk} \tag{6.16}$$
$$g_{esp}\le 1-x_{es+p} \qquad \forall e\in E, p\in P^{-1}, s\in \pi_{p+1} \tag{6.17}$$
$$\sum_{s\in\pi_p} g_{esp}\ge G_{ep}-v^{\mathrm{M}_4}_{ep} \qquad \forall e\in E, p\in P \tag{6.18}$$
$$\sum_{p\in P}\sum_{s\in S_d\cap\pi_p} g_{esp}\le 1 \qquad \forall e\in E, d\in D \tag{6.19}$$

$$\sum_{e\in E_t} x_{es} \leq y''_{tdk} \qquad \forall t\in T, d\in D, k\in K, s\in S_{dk} \tag{6.20}$$

$$\sum_{k\in K} y''_{tdk} \leq Y_t + v^{\mathrm{M}_5}_{td} \qquad \forall t\in T, d\in D \tag{6.21}$$

$$\sum_{e\in E_t}\sum_{s\in S_d} x_{es} \geq F_{td} - v^{\mathrm{M}_6}_{td} \qquad \forall t\in T, d\in D \tag{6.22}$$

$$\sum_{s\in S_{dk}} x_{es} \leq L''_e + l_{edk} \qquad \forall e\in E, d\in D, k\in K \tag{6.23}$$

$$\sum_{e\in E_t} x_{es} \leq y'_{td} \qquad \forall t\in T, d\in D, s\in S_d \tag{6.24}$$

$$\sum_{e\in E_t}\sum_{s\in S_d} x_{es} \geq y'_{td} \qquad \forall t\in T, d\in D \tag{6.25}$$

$$\sum_{(m,n)\in Q_{dk}} z_{tdkmn} = 1 \qquad \forall t\in T, d\in D, k\in K, m\in S^{-2}_{dk} \tag{6.26}$$

$$\sum_{(m,n)\in U_{dk}} z_{tdkmn} \leq y'_{td} \qquad \forall t\in T, d\in D, k\in K, n\in S^{+2}_{dk} \tag{6.27}$$

$$z_{tdkss} \leq 1 + \sum_{e\in E_t} (x_{es+1} - x_{es}) \qquad \forall t\in T, d\in D, k\in K, s\in S^{-2}_{dk} \tag{6.28}$$

$$z_{tdkmm+1} \leq 1 - \sum_{e\in E_t} x_{en} \qquad \forall t\in T, d\in D, k\in K, (m,n)\in U_{dk} \tag{6.29}$$

$$z_{tdkmn} \leq \sum_{e\in E_t} x_{en} \qquad \forall t\in T, d\in D, k\in K, (m,n)\in U_{dk} \tag{6.30}$$

$$x_{es},\ b_{es} \in \{0,1\} \qquad \forall e\in E, s\in S \tag{6.31}$$

$$a_{cs} \in \{0,1\} \qquad \forall c\in C, d\in D, k\in K, s\in S^{+1}_{dk} \tag{6.32}$$

$$g_{esp} \in \{0,1\} \qquad \forall e\in E, s\in \pi_p, p\in P \tag{6.33}$$

$$y'_{td} \in \{0,1\} \qquad \forall t\in T, d\in D \tag{6.34}$$

$$y''_{tdk} \in \{0,1\} \qquad \forall t\in T, d\in D, k\in K \tag{6.35}$$

$$l_{edk} \geq 0 \qquad \forall e\in E, d\in D, k\in K \tag{6.36}$$

$$z_{tdkmn} \in \{0,1\} \qquad \forall t\in T, d\in D, k\in K, (m,n)\in Q_{dk} \tag{6.37}$$

$$v^{\mathrm{M}_1}_{ed}, v^{\mathrm{M}_{2a}}_{ed}, v^{\mathrm{M}_{2b}}_{edk} \geq 0 \qquad \forall e\in E, d\in D, k\in K^{-1} \tag{6.38}$$

$$v^{\mathrm{M}_3}_{es} \geq 0 \qquad \forall e\in E, d\in D, k\in K, s\in S^{-1}_{dk} \tag{6.39}$$

$$v^{\mathrm{M}_4}_{ep} \geq 0 \qquad \forall e\in E, p\in P \tag{6.40}$$

$$v^{\mathrm{M}_5}_{td}, v^{\mathrm{M}_6}_{td} \geq 0 \qquad \forall t\in T, d\in D \tag{6.41}$$

The objective function of the problem is composed by several weighted parts presented by equation (6.1). While the former minimizes the violations of the soft requirements S_1 to S_4, the latter minimizes the violation of the medium requirements M_1 to M_6 when they are considered as soft ones in specific instances of the problem. When medium requirements are considered hard, the corresponding slack variables v are simply removed from the model.

Constraint set (6.2) ensures that the workload of each event is fully scheduled. Constraint sets (6.3)-(6.5) ensure that teachers, classes, and shared rooms are scheduled to only one lesson at a time, respectively. Constraint sets (6.6) and (6.7) ensure, respectively, that teacher and classes are scheduled in available periods. Constraint sets (6.8) and (6.9) ensure that the lessons of a class are scheduled sequentially in each shift. Constraint set (6.10) ensures the number of daily lessons of each event is limited by $L'_e + v_{ed}^{M_1}$. Constraint sets (6.11)-(6.13) formulate the requirement M_2. Constraint set (6.11) identifies the block heads of each event. Constraint set (6.12) ensures the number of daily blocks of each event is limited by $1 + v_{ed}^{M_{2a}}$. Constraint set (6.13) is necessary to avoid blocks being assigned between shifts. Observe that i indicates the first timeslot in the shift $k+1$ on day d. Hence, $i-1$ indicates the last timeslot in the previous shift on the same day. Constraint set (6.14) formulates the requirement M_3 by avoiding more than one lesson been assigned to a pair of timeslots $(s, s+1)$ in a given shift.

Constraint sets (6.15)-(6.19) formulate the requirement M_4 in conjunction with constraint sets (6.11)-(6.13). Constraint set (6.15) enforces the formation of blocks of size p with a head beginning at the timeslot s when the variable g_{esp} is active for the event e. Constraint sets (6.16)-(6.17) ensure no lesson is assigned to the timeslots located immediately before and after the block associated to the variable g_{esp}. Constraint set (6.18) ensures at least $G_{ep} - v_{ep}^{M_4}$ blocks of size p are established to event e. Constraint set (6.19) ensures no more than one block variable g_{esp} is active on each day for a given event.

Constraint set (6.20) identifies the working shifts of teachers through the variable y''_{tdk}. Constraint set (6.21) ensures the number of working shifts for each teacher on a given day is limited by $Y''_{td} + v_{td}^{M_5}$. Constraint set (6.22) formulates the requirement M_6 by assigning at least $F_{td} - v_{td}^{M_6}$ lessons to a teacher t on each day d. Constraint set (6.23) ensures the number of lessons of each event in a given shift is limited by $L''_e + l_{edk}$. Constraint sets (6.24) and (6.25) identify the working days for each teacher through the variable y'_{td}. Constraint sets (6.26)-(6.30) determine the number of idle periods in a solution using the *idle periods graph* formulation as presented in Section 3.1.1.

In order to make the formulation stronger, we also included two cuts derived from the work of Souza (2000). Constraint sets (6.42) and (6.43) define, respectively, the minimum number of working days and the minimum number of working shifts for each teacher. In case the requirement M_1 is considered hard then $\lambda_e = L'_e$, otherwise, $\lambda_e = \infty$.

$$\sum_{d \in D} y'_{td} \geq \max\left\{ \left\lceil \frac{\sum_{e \in E_t} W_e}{|P||K|} \right\rceil, \max_{e \in E_t}\left\{ \left\lceil \frac{W_e}{\lambda_e} \right\rceil \right\} \right\} \qquad \forall t \in T \tag{6.42}$$

$$\sum_{d \in D} \sum_{k \in K} y''_{tdk} \geq \left\lceil \frac{\sum_{e \in E_t} W_e}{|P|} \right\rceil \qquad \forall t \in T \tag{6.43}$$

6.2 Modelling HSTP$^+$ as a XHSTT problem

In this section we describe how we mapped the HSTP$^+$ requirements to the constraints available in the XHSTT format presented in Section 2.3.2. Table 6.2 shows the problem requirement (*Req*), the constraint type in the XHSTT format used to represent the problem requirement, as well as the number of constraints of that type are required ($\forall$). The remaining columns shows the properties defined inside the XHSTT constraint. Columns *AppliesTo* and *TimeGroups* represents, respectively, the set of entities and the set of timeslots in which the constraint is applied. Finally, columns *Min*, *Max*, and *Du* define, respectively, the properties *Minimum*, *Maximum* and *Duration*. Depending on the type of the constraint, some properties are not required. We indicate these cases in the table using the symbol "$-$". Observe that most of problem requirements have a direct representation in the XHSTT. Only the requirements M_2 and M_4 required multiple constraints to be modelled properly. A major issue we faced when modelling in the XHSTT format is that some useful constraint types, as LimitBusyTimes, can be applied to resources but not to events. In order to overcome this limitation, for each event $e \in E$ we associated an artificial resource $e \in \widehat{E}$.

Table 6.2 – Mapping the requirements of HSTP$^+$ to the XHSTT format.

Req	Constraint type	$\forall$	AppliesTo	TimeGroups	Min	Max	Du
H_1	AssignTimes	1	$e \in E$	–	–	–	–
H_2	AvoidClashes	1	$t \in T$	–	–	–	–
H_3	AvoidClashes	1	$c \in C$	–	–	–	–
H_4	AvoidUnavailableTimes	$t \in T$	t	$\{s\} : s \in S, V_{ts} = 0$	–	–	–
H_5	AvoidUnavailableTimes	$c \in C$	c	$\{s\} : s \in S, V_{cs} = 0$	–	–	–
H_6	AvoidClashes	1	$r \in R$	–	–	–	–
H_7	LimitIdleTimes	1	$c \in C$	$S_{dk} : d \in D, k \in K$	0	0	–
M_1	LimitBusyTimes	$i \in [\lvert D \rvert \lvert P \rvert]^{-1}$	$e \in \widehat{E} : L'_e = i$	$S_d : d \in D$	0	i	–
M_2	SpreadEvents	1	$e \in E$	$S_d : d \in D$	0	1	–
	PreferTimes	$i \in P^{+1}$	$e \in E : W_e \geq i$	π_i	–	–	i
M_3	LimitBusyTimes	1	$e \in \widehat{E}$	$\{s, s+1\} : s \in \pi_2$	0	1	–
	DistributeEvents	$p \in P, f \in D$	$e \in E : G_{ep} = f$	–	f	∞	p
M_4	SpreadEvents	1	$e \in E$	$S_d : d \in D$	0	1	–
	PreferTimes	$i \in P^{+1}$	$e \in E : W_e \geq i$	π_i	–	–	i
M_5	ClusterBusyTimes	$d \in D, i \in K^{-1}$	$t \in T : Y_t = i$	$S_{dk} : d \in D, k \in K$	0	i	–
M_6	ClusterBusyTimes	$d \in D$	$t \in T : F_{td} = 1$	S_d	1	1	–
S_1	LimitBusyTimes	$p \in P^{-1}$	$e \in \widehat{E} : L''_e = p$	$S_{dk} : d \in D, k \in K$	0	p	–
S_2	ClusterBusyTimes	1	$t \in T$	$S_{dk} : d \in D, k \in K$	0	0	–
S_3	ClusterBusyTimes	1	$t \in T$	$S_d : d \in D$	0	0	–
S_4	LimitIdleTimes	1	$t \in T$	$S_{dk} : d \in D, k \in K$	0	0	–

Source: created by author.

6.3 Computational results

In this section we present an experimental evaluation on HSTP$^+$. The experiments are design to help us answering the following questions:

i) How important is to use a feasible solution as starting point?

ii) How does the fix-and-optimize heuristic compares to CPLEX and the state-of-the-art methods?

iii) Are the algorithms able to provide the same performance if we slight change the instances ?

6.3.1 Environment

The algorithms were coded in C++ using the compiler g++ 4.9.2. The mathematical models and subproblems are solved by CPLEX 12.6.2 (IBM, 2015) with default settings in single core mode. All runs were performed in a server machine equipped with an Intel® Xeon® E5-2697 processor clocked at 2.7GHz, 64GB of RAM, running a 64 bits Linux operating system. The mathematical model parameters ω, γ, δ and μ were set to 3, 9, 100 and 1000, respectively. All statistical tests were calculated by GNU R 2.15.2 and a significance level set to 5%. In the experiments that we apply a Student's t-test, a Shapiro–Wilk test is used to evaluate the normality of the data. In addition, the version of Students' t-test we used assumes that samples have different variances. Moreover, an implicit null hypothesis indicated by $\mathcal{H}_0$ is associated to each alternative hypothesis test proposed in the next sections. By default, $\mathcal{H}_0$ states that there is no significant difference between the compared results. The initial constructive solutions provided to GOAL and SVNS solvers were generated by the KHE library version 2014-05-07 (KINGSTON, 2014b).

6.3.2 Datasets

The dataset used in the experiments was provided by an industrial partner and its main features are presented in Table 6.3. It is composed by 33 instances originated from several schools, located in the south region of Brazil. In the table, the instances are sorted in increasing order according to the product $|S||E|$, i.e, the number of decision variables. We classified them in three categories according to their dimensions: *small* (01 to 11), *medium* (12 to 21) and *large* (22 to 33) instances. This classification matches with the size of the school. For each instance we report the number of timeslots ($|S|$), number of days ($|D|$), number of shifts ($|K|$), number of teachers ($|T|$), number of classes ($|C|$), number of shared rooms ($|R|$), number of events ($|E|$), and the total number of lessons that need to be scheduled ($\sum W_e$). The remaining columns describe the subset of requirements which each instance take into account. Cells marked with a bullet ($\bullet$) means that at least one requirement of that type is considered in the instance.

From the dataset presented in Table 6.3, we refer to three different groups of instances: A, B and C. Each one represents a version of the whole dataset. The group A represents the original version, while groups B and C are versions derived from A, differing in the amount and in the set of requirements are used. In group A, all medium requirements are set to hard, while in group B they are set to soft. Instances from group C are identical to the ones on group A, except we ignore the requirement H_4, i.e, all teachers have full availability. While group A allows us to evaluate the algorithms in a realistic scenario, groups B and C are useful to stress the algorithms in worst case conditions and reveal theirs limitations. In group B, the number of soft requirements is larger than in A and, as a result, the number of auxiliary variables the model needs to manage is also larger. Similarly, in group B, by ignoring requirement H_4, the number of "free" decision variables increases, what implies in exploring a larger search space. Together, groups A, B and C accounts a set of 99 novel test cases we made available in the XHSTT format.

Table 6.3 – Main characteristics of the dataset.

Id	$\|S\|$	$\|D\|$	$\|K\|$	$\|T\|$	$\|C\|$	$\|R\|$	$\|E\|$	$\sum W_e$	H_{123}	H_4	H_5	H_6	H_7	M_1	M_2	M_3	M_4	M_5	M_6	S_1	S_2	S_3	S_4
01	20	5	1	10	7	0	56	126	•				•	•	•							•	•
02	20	5	1	11	8	0	64	160	•					•	•							•	•
03	20	5	1	32	21	0	74	262	•	•			•	•	•		•					•	•
04	25	5	1	15	8	0	64	184	•	•	•			•	•							•	•
05	25	5	1	18	9	0	72	207	•	•	•			•	•							•	•
06	20	5	2	26	6	0	100	120	•	•				•	•		•					•	•
07	60	5	2	21	3	0	37	105	•	•	•			•	•		•					•	•
08	25	5	1	22	12	1	108	300	•	•		•		•	•		•					•	•
09	30	5	1	20	9	0	93	244	•	•	•			•			•					•	•
10	40	5	2	10	8	0	72	160	•		•			•					•			•	•
11	40	5	2	15	10	1	77	184	•	•	•	•		•	•							•	•
12	25	5	1	26	12	0	144	300	•	•				•	•							•	•
13	30	5	1	26	12	0	145	303	•	•	•			•	•							•	•
14	60	5	1	30	13	0	151	331	•	•	•			•	•							•	•
15	30	5	1	25	12	0	154	340	•	•	•			•	•							•	•
16	25	5	1	38	21	0	210	525	•					•								•	•
17	60	5	2	25	7	0	104	234	•	•	•		•	•	•						•	•	•
18	60	5	2	19	7	0	104	234	•	•	•		•	•	•						•	•	•
19	50	5	2	25	16	4	141	400	•	•	•	•		•	•		•					•	•
20	50	5	2	27	9	0	153	285	•	•	•		•	•	•		•					•	•
21	60	5	2	27	10	0	177	321	•	•	•		•	•	•		•					•	•
22	50	5	2	44	15	1	261	525	•	•	•	•		•	•		•					•	•
23	50	5	2	48	15	1	261	525	•	•	•	•		•	•		•					•	•
24	50	5	2	53	15	1	266	532	•	•	•	•		•	•		•					•	•
25	75	5	3	60	22	16	219	574	•	•	•	•	•	•	•		•	•				•	•
26	75	5	3	68	31	15	245	719	•		•	•	•	•	•		•	•				•	•
27	75	5	3	64	31	16	249	749	•	•	•	•	•	•	•		•	•				•	•
28	75	5	3	70	36	16	265	764	•	•	•	•	•	•	•		•	•				•	•
29	126	6	3	44	20	0	256	780	•	•	•		•	•		•	•			•		•	•
30	126	6	3	45	23	0	294	882	•	•	•		•	•		•	•			•		•	•
31	126	6	3	53	27	0	318	985	•	•	•		•	•		•	•			•		•	•
32	126	6	3	50	30	0	374	1131	•	•	•		•	•		•	•			•		•	•
33	126	6	3	53	30	1	380	1153	•	•	•	•	•	•		•	•			•		•	•

Source: created by author.

6.3.3 Experiments with a general purpose MIP solver

Table 6.4 reports the main results given by CPLEX for instances of the groups A, B, and C. Column Id displays the identifier of each instance. Columns *Obj* and *LB* show, respectively, values of the best solution and the best lower bound found by the solver within a time limit of 10 hours. Column *Gap* shows the percentage deviation between the best solution and the best lower bound, hereafter referred as *optimality gap*. It is computed as $100 \times (Obj - LB)/(Obj)$ and assumes the value zero when $(Obj - LB) = 0$, in this case, the solution is guaranteed optimal. Finally, column *Time* reports the running time in seconds. Cells marked with "*t.l.*" indicate the time limit was reached without proof of optimality. The last row (Avg_*) displays the average values for the whole instance group. Additionally, we also display average values corresponding to small (Avg_s), medium (Avg_m) and, large (Avg_l) instances. Further details are presented in Appendix A.

From the table we see that CPLEX found feasible solutions for all instances within the time limit. Analyzing the performance according to the dimensions of the instances, on the whole, it performed better when solving small and medium instances. Particularly on small instances the solver did very well, finding optimal solutions for the majority of instances and achieving an optimality gap less than 1%, on average, in all groups. Regarding medium size instances, CPLEX also obtained solutions very close to the optimal, with quality comparable to the small ones, except in instances of group C, where its performance decreased considerably. Finally, CPLEX provided the worst results when solving large instances. In this category, the smallest average gap achieved was 13% in group A, while reached up to 24% in group C, approximately. In contrast to small and medium size instances, on large instances CPLEX was terminated due to time limit in all runs.

If we focus our analyse on each group separately, a closer comparison shows that CPLEX performed similarly in groups A and B, while in group C, it performed twice times worse than both other groups. It means that the modifications we made in group A for creating the group C turned the instances harder for CPLEX. In order to support our conclusions, we performed a paired Student's t-test for evaluating two hypothesis presented in Table 6.5. From this table we can draw two main conclusions. In one hand, the p-value obtained to test $\mathcal{H}_1$ states that the

Table 6.4 – Results of CPLEX for all instances with a time limit of 10 hours.

	Group A				**Group B**				**Group C**			
Id	Obj	LB	Gap	Time	Obj	LB	Gap	Time	Obj	LB	Gap	Time
01	315	315.0	0.0	1	315	315.0	0.0	2	315	315.0	0.0	1
02	360	360.0	0.0	2	360	360.0	0.0	2	360	360.0	0.0	2
03	690	684.0	0.9	*t.l.*	690	684.0	0.9	*t.l.*	663	657.0	0.9	*t.l.*
04	417	414.0	0.7	*t.l.*	417	414.0	0.7	*t.l.*	408	405.0	0.7	*t.l.*
05	528	528.0	0.0	3915	528	528.0	0.0	1297	495	480.6	2.9	*t.l.*
06	333	333.0	0.0	2	333	333.0	0.0	1	324	324.0	0.0	1
07	387	387.0	0.0	9	387	387.0	0.0	31	351	351.0	0.0	709
08	600	597.0	0.5	*t.l.*	600	600.0	0.0	2438	576	567.0	1.6	*t.l.*
09	474	471.0	0.6	*t.l.*	474	474.0	0.0	20302	441	441.0	0.0	5376
10	282	276.0	2.1	*t.l.*	282	279.0	1.1	*t.l.*	282	276.0	2.1	*t.l.*
11	426	426.0	0.0	13287	426	423.0	0.7	*t.l.*	405	405.0	0.0	4515
Avg$_s$	437	435.5	0.4	17929	437	436.1	0.3	15279	420	416.5	0.7	17328
12	654	654.0	0.0	3027	654	654.0	0.0	1587	651	630.0	3.2	*t.l.*
13	648	630.0	2.8	*t.l.*	645	630.0	2.3	*t.l.*	618	603.0	2.4	*t.l.*
14	759	759.0	0.0	834	759	759.0	0.0	716	810	651.6	19.6	*t.l.*
15	690	613.8	11.0	*t.l.*	672	613.8	8.7	*t.l.*	630	612.0	2.9	*t.l.*
16	1077	1071.0	0.6	*t.l.*	1077	1071.0	0.6	*t.l.*	1077	1071.0	0.6	*t.l.*
17	903	886.4	1.8	*t.l.*	906	885.8	2.2	*t.l.*	1035	780.6	24.6	*t.l.*
18	1089	1089.0	0.0	237	1089	1089.0	0.0	392	1167	881.6	24.5	*t.l.*
19	783	783.0	0.0	22162	783	780.0	0.4	*t.l.*	756	736.2	2.6	*t.l.*
20	540	540.0	0.0	35124	540	540.0	0.0	5547	498	405.0	18.7	*t.l.*
21	576	541.5	6.0	*t.l.*	573	546.0	4.7	*t.l.*	528	454.5	13.9	*t.l.*
Avg$_m$	771	756.8	2.2	24138	769	756.9	1.9	22424	777	682.5	11.3	*t.l.*
22	1074	875.7	18.5	*t.l.*	1170	881.7	24.6	*t.l.*	1050	807.8	23.1	*t.l.*
23	1287	1080.8	16.0	*t.l.*	1323	1086.0	17.9	*t.l.*	1068	836.5	21.7	*t.l.*
24	1245	1074.0	13.7	*t.l.*	1239	1075.1	13.2	*t.l.*	1137	934.9	17.8	*t.l.*
25	1374	1069.9	22.1	*t.l.*	1296	1066.3	17.7	*t.l.*	1266	1062.6	16.1	*t.l.*
26	1557	1338.4	14.0	*t.l.*	1566	1334.2	14.8	*t.l.*	1557	1338.4	14.0	*t.l.*
27	1560	1391.0	10.8	*t.l.*	1539	1388.4	9.8	*t.l.*	1635	1343.5	17.8	*t.l.*
28	1509	1320.5	12.5	*t.l.*	1581	1300.6	17.7	*t.l.*	1494	1239.3	17.0	*t.l.*
29	1395	1288.5	7.6	*t.l.*	1344	1288.8	4.1	*t.l.*	1332	991.5	25.6	*t.l.*
30	1951	1713.6	12.2	*t.l.*	1942	1714.7	11.7	*t.l.*	1761	1072.8	39.1	*t.l.*
31	1636	1596.7	2.4	*t.l.*	1735	1593.9	8.1	*t.l.*	1476	1189.5	19.4	*t.l.*
32	1888	1620.2	14.2	*t.l.*	1951	1631.7	16.4	*t.l.*	2001	1191.0	40.5	*t.l.*
33	1921	1684.7	12.3	*t.l.*	1903	1689.2	11.2	*t.l.*	1980	1296.6	34.5	*t.l.*
Avg$_l$	1533	1337.8	13.0	*t.l.*	1549	1337.6	13.9	*t.l.*	1479	1108.7	23.9	*t.l.*
Avg$_*$	937	861.0	5.6	26382	942	861.1	5.7	24979	913	748.8	12.4	29776

Source: created by author.

performance of CPLEX is not significantly affected according to changes in medium requirements, i.e, its performance is not significantly affected whether requirements M_1-M_6 are modelled as hard or soft requirements. In some sense, these results could be expected since the only difference in modelling a requirement as hard or soft is the addition of a set of continuous slack variables in the latter case.

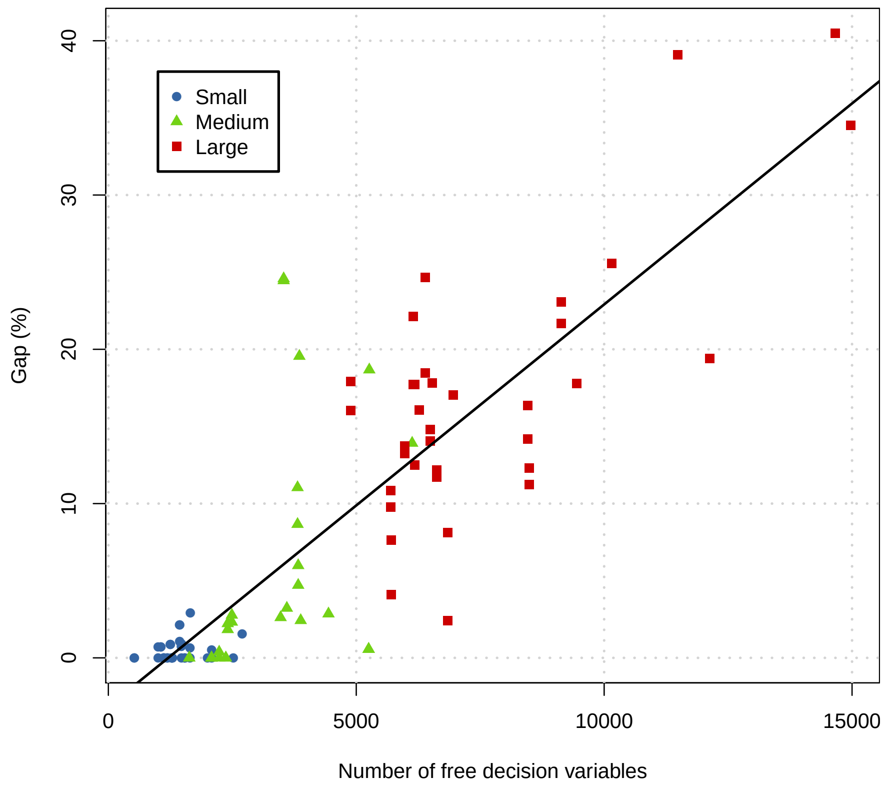
Table 6.5 – Paired Student's t-test performed on CPLEX results.

Hypothesis description	p-value	Result
$\mathcal{H}_1$: Gap of group B is higher than Gap of group A	0.313	Failed
$\mathcal{H}_2$: Gap of group C is higher than Gap of group A	0.000	Succeeded

Source: created by author.

In the other hand, the p-value computed to test $\mathcal{H}_2$ indicates that the performance of CPLEX significantly decreases when the number of free decision variables increases, i.e, the results suggest that there is a strong correlation of the performance with the dimensions of the instances. This behavior can be better observed in Figure 6.1 that shows a linear correlation of 0.85 between the optimality gap (Gap) and the number of free decisions variables on each instance. Among all instance parameters, this is the one that presents the highest linear correlation with the optimality gap.

Figure 6.1 – Linear regression for results of CPLEX on all instances.



Source: Figure created by author.

6.3.4 Experiments with methods for generating initial solutions.

In order to provide an initial feasible solution to start variants of the fix-and-optimize heuristic, for each instance we disregarded all soft constraints and picked the first feasible solution found by CPLEX. We hereafter refer to this approach by CPX0. Table 6.7 displays, for each instance group, the objective value of the solution (*Obj*), the optimality gap (*Gap*), and the running time in seconds (*Time*).

The results show this approach is quite effective in providing feasible solutions for the problem at hand. Although the majority of solutions are far from optimal, feasible solutions for all instances were found quickly, in less than 50 seconds, except for the instance A23 that took 157 seconds. As expected, the average time to find a solution increases consistently according to the size of the instances in all groups. If we compare the results obtained individually for each group, one can observe that the average running times for groups B and C are shorter than in group A. One may conclude the modifications performed on the dataset might turn the factibilization of the instances easier to CPX0. However, according to the hypothesis tests presented in Table 6.6, only the solutions of group B were obtained significantly faster than group A, by spending, on average, 2 seconds approximately. This short running time, clearly comes at the high cost of violating several medium requirements in instances of group B. Particularly in this group, the solutions are really poor. Finally, the results obtained by the two last hypothesis tests, which are presented in Table 6.6, support the observation that the quality of solutions obtained in groups B and C are significantly worse than solutions obtained in group A.

Table 6.6 – Paired Student's t-test performed on CPX0 results.

Hypothesis description	p-value	Result
$\mathcal{H}_1$: Time of group B is less than Time of group A	0.015	Succeeded
$\mathcal{H}_2$: Time of group C is less than Time of group A	0.064	Failed
$\mathcal{H}_3$: Gap of group B is greater than Gap of group A	0.000	Succeeded
$\mathcal{H}_4$: Gap of group C is greater than Gap of group A	0.000	Succeeded

Source: created by author.

Table 6.7 – Feasible solutions generated by CPX0.

	Group A			Group B			Group C		
Id	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
01	468	32.7	0	5486	94.3	0	468	32.7	0
02	516	30.2	0	7555	95.2	0	516	30.2	0
03	975	29.8	5	40032	98.3	0	1143	42.5	3
04	534	22.5	1	28549	98.5	0	753	46.2	0
05	615	14.1	1	25621	97.9	0	831	42.2	1
06	612	45.6	0	12684	97.4	0	702	53.8	0
07	477	18.9	0	20474	98.1	0	624	43.7	0
08	918	35.0	0	23915	97.5	0	1179	51.9	0
09	621	24.2	0	6639	92.9	0	954	53.8	0
10	522	47.1	0	28510	99.0	0	522	47.1	0
11	450	5.3	0	25453	98.3	0	672	39.7	0
Avg$_s$	609	27.8	1	20447	97.0	0	760	44.0	1
12	693	5.6	13	37729	98.3	0	1248	49.5	2
13	915	31.1	3	34924	98.2	0	1365	55.8	2
14	798	4.9	4	45807	98.3	0	1389	53.1	2
15	1149	46.6	5	33221	98.2	0	1464	58.2	3
16	2052	47.8	0	3052	64.9	0	2052	47.8	0
17	1392	36.3	2	35473	97.5	0	2013	61.2	1
18	1152	5.5	5	55224	98.0	0	2130	58.6	2
19	846	7.4	4	57861	98.7	0	1278	42.4	2
20	729	25.9	7	41762	98.7	1	1176	65.6	1
21	981	44.8	5	40293	98.6	0	1317	65.5	2
Avg$_m$	1070	25.6	5	38534	94.9	0	1543	55.8	2
22	1479	40.8	49	136788	99.4	7	1674	51.7	37
23	1443	25.1	157	131695	99.2	9	1746	52.1	29
24	1617	33.6	45	129896	99.2	6	1917	51.2	31
25	1728	38.1	17	165595	99.4	2	1827	41.8	24
26	2076	35.5	19	220066	99.4	2	2076	35.5	18
27	2055	32.3	12	213703	99.4	3	2118	36.6	21
28	2463	46.4	12	212201	99.4	2	2442	49.3	19
29	3893	66.9	13	76672	98.3	5	3444	71.2	2
30	4127	58.5	8	65312	97.4	19	3548	69.8	2
31	2644	39.6	13	95486	98.3	5	3315	64.1	2
32	4076	60.2	8	77483	97.9	3	4708	74.7	2
33	2899	41.9	8	86554	98.0	4	3762	65.5	3
Avg$_l$	2541	43.2	30	134287	98.8	6	2714	55.3	16
Avg$_*$	1451	32.7	13	67324	97.0	2	1708	51.7	7

Source: created by author.

In Table 6.8 we report the solutions generated by KHE library that are used as initial solution by GOAL and SVNS solvers. Columns and rows in this table have the same meaning than the previous one, but notice that column *Obj* has a slightly different format. Since KHE can produce infeasible solutions, the corresponding objective function value is represented by a pair (Inf / $Cost_s$), where Inf and $Cost_s$ display, respectively, the number of hard requirements violated and the cost associated to the violation of soft requirements. The Inf value can be interpreted as a *feasibility distance*. Hence, a solution is feasible when $Inf = 0$, and infeasible otherwise. In order to compute the optimality gap (*Gap*), we convert the pair into a single cost $Obj = \mathcal{M} * Inf + Cost_s$. This way, infeasible solutions are reasonably penalized according to the number of infeasibilities. In addition, we report in the last row (*#fea*), the number of feasible solutions found by KHE in each group of instances.

Analyzing the table, it can be seen that the majority of solutions produced by KHE are infeasible. The percentage of feasibility achieved for groups A, B, and C are, respectively, 21%, 44% and 45%. Another tendency, clearly observable, is that the running time increases according to the instance size. We noted that, while solutions for instances 01 to 28 were produced in less than 1 minute, the solutions for instances 29 to 33 required a significant amount of time, all these surpassing 5 minutes and reaching up to 17 minutes, approximately, in instance C33. Analyzing instances 29 to 33 in Table 6.3 it can be observed that besides the number of timeslots ($|S|$), these instances differ from the others only by requirements M_3 and S_1. Both these requirements are also modelled in the XHSTT format by using a LimitBusyTimes constraint. The slowdown observed in these particular instances, might be related to the application of the LimitBusyTimes over a set of dummy resources ($\hat{E}$) we created for surpassing modelling limitations of the XHSTT format. Possibly, KHE had some difficulty to tackle a high number of resources that does not behavior like usual resources.

Table 6.8 – Initial constructive solutions generated by KHE.

Id	Group A Obj	Gap	Time	Group B Obj	Gap	Time	Group C Obj	Gap	Time
01	0 / 351	10.3	1	0 / 342	7.9	1	0 / 351	10.3	1
02	0 / 414	13.0	3	0 / 408	11.8	3	0 / 414	13.0	3
03	23 / 957	97.1	6	5 / 5029	93.2	10	6 / 1137	90.8	8
04	12 / 489	96.7	4	0 / 5492	92.5	8	1 / 522	73.4	9
05	12 / 537	95.8	6	0 / 7585	93.0	9	1 / 645	70.8	7
06	0 / 360	7.5	1	0 / 369	9.8	1	0 / 369	12.2	1
07	29 / 459	98.7	9	3 / 16453	98.0	9	6 / 594	94.7	7
08	5 / 648	89.4	12	1 / 4645	89.4	29	3 / 624	84.4	10
09	0 / 504	6.5	8	1 / 513	68.7	9	0 / 483	8.7	8
10	0 / 324	14.8	3	0 / 351	20.5	4	0 / 324	14.8	3
11	7 / 435	94.3	3	0 / 4441	90.5	5	0 / 468	13.5	8
Avg_s	8 / 498	56.7	5	0 / 4148	61.4	8	1 / 539	44.2	6
12	16 / 672	96.1	5	1 / 8687	93.2	9	0 / 774	18.6	11
13	11 / 693	94.6	7	3 / 5687	92.7	10	1 / 735	65.2	9
14	28 / 741	97.4	5	1 / 21777	96.7	10	0 / 855	23.8	17
15	0 / 783	21.6	20	0 / 771	20.4	15	0 / 750	18.4	15
16	0 / 1083	1.1	20	0 / 1095	2.2	13	0 / 1083	1.1	21
17	11 / 1077	92.7	27	1 / 6116	87.6	30	0 / 1158	32.6	25
18	33 / 1050	96.8	25	1 / 21140	95.1	29	0 / 1350	34.7	36
19	29 / 801	97.4	27	6 / 17801	96.7	33	5 / 882	87.5	47
20	14 / 642	96.3	8	0 / 9636	94.4	12	0 / 729	44.4	9
21	3 / 693	85.3	26	0 / 2711	79.9	22	0 / 702	35.3	13
Avg_m	14 / 823	77.9	17	1 / 9542	75.9	18	0 / 901	36.2	20
22	34 / 1062	97.5	35	4 / 31044	97.5	49	30 / 1020	97.4	34
23	43 / 1161	97.6	28	14 / 31233	97.6	45	30 / 1044	97.3	38
24	33 / 1194	96.9	37	9 / 21194	96.4	46	20 / 1113	95.6	46
25	17 / 1386	94.2	23	0 / 20404	94.8	25	15 / 1344	93.5	23
26	20 / 1674	93.8	35	2 / 17689	93.2	30	20 / 1674	93.8	36
27	30 / 1614	95.6	26	7 / 21635	95.2	22	20 / 1605	93.8	25
28	37 / 2142	96.6	22	1 / 20214	93.9	21	31 / 2169	96.3	18
29	13 / 1407	91.1	322	1 / 1395	46.2	507	23 / 1296	95.9	377
30	18 / 1782	91.3	367	2 / 5785	78.0	408	28 / 1476	96.4	505
31	25 / 1765	94.0	379	7 / 5665	87.4	386	22 / 1557	95.0	612
32	19 / 1849	92.2	498	0 / 1873	12.9	483	24 / 1674	95.4	665
33	17 / 1879	91.1	408	0 / 1933	12.6	532	13 / 1746	91.2	1032
Avg_l	25 / 1576	94.3	182	3 / 15005	75.5	213	23 / 1476	95.1	284
Avg_*	16 / 988	76.8	73	2 / 9730	70.9	86	9 / 989	60.3	111
#fea	7			14			15		

Source: created by author.

According to the statistical tests presented in Table 6.9, in contrast with the performance observed in CPX0, the KHE required significantly more time for producing solutions to groups B and C than for group A. However, the extra time spent on groups B and C only resulted in significant improvements in solution quality in Group C. As a result, the performance of KHE decreased on Group B.

Table 6.9 – Paired Student's t-test performed on KHE results.

Hypothesis description	p-value	Result
$\mathcal{H}_1$: Time of group B is greater than Time of group A	0.034	Succeeded
$\mathcal{H}_2$: Time of group C is greater than Time of group A	0.035	Succeeded
$\mathcal{H}_3$: Gap of group B is less than Gap of group A	0.078	Failed
$\mathcal{H}_4$: Gap of group C is less than Gap of group A	0.001	Succeeded

Source: created by author.

In order to compare the results obtained by CPX0 and KHE we performed an additional set of statistical tests reported in Table 6.10. Concerning feasibility, CPX0 clearly outperformed KHE since the former produced feasible solutions for 100% of the instances. In addition, CPX0 spent significantly less computational time than KHE in all groups. Regarding solution quality, each method performed better in a distinct group of instances. While CPX0 was able to produce better solutions in Group A, KHE produced better solutions in Group B. Both approaches provided solutions with comparable quality in Group C.

Finally, the test result of $\mathcal{H}_5$ deserves a short discussion. We observed that even KHE had generated several *infeasible* solutions in Group B, the overall quality of these infeasible solutions is higher than the quality of *feasible* solutions provided by CPX0 in the same group. This result suggests a limitation of CPX0 when applied on instances with few hard requirements since, in a less constrained scenario, it is easier for CPX0 to find a solution that although being feasible has poor quality.

Table 6.10 – Paired Student's t-test comparing results obtained by KHE and CPX0.

Hypothesis description	p-value	Result
$\mathcal{H}_1$: CPX0 spent less time than KHE in group A	0.011	Succeeded
$\mathcal{H}_2$: CPX0 spent less time than KHE in group B	0.003	Succeeded
$\mathcal{H}_3$: CPX0 spent less time than KHE in group C	0.009	Succeeded
$\mathcal{H}_4$: Gap of CPX0 is less than Gap of KHE in group A	0.000	Succeeded
$\mathcal{H}_5$: Gap of CPX0 is higher than Gap of KHE in group B	0.000	Succeeded
$\mathcal{H}_6$: Gap of CPX0 is less than Gap of KHE in group C	0.092	Failed

Source: created by author.

6.3.5 Experiments with local-search based solvers

Here we evaluate two state-of-the-art solvers designed for solving GHSTP. These solvers, hereafter, referred as GOAL and SVNS are described in Section 2.3.4. While both solvers use originally KHE as a constructive method, we have observed in previous experiments, that KHE is outperformed by CPX0 regarding running time and feasibility. Thus, we are also interested in investigating how these solvers perform by receiving a feasible initial solution provided by CPX0. For a short representation, we indicate the results obtained by GOAL and SVNS solvers by using, respectively, the letters G and S. The method used for providing the initial solution to each solver is indicated by a subscript letter. The letter K refers to KHE and the letter C refers to CPX0. One may note that while G_K and S_K are hybrid meta-heuristics, when we replace KHE by CPX0, the resulting variants G_C and S_C became *matheuristic* approaches that can be classified as a collaborative combination arranged in sequential phases. Firstly CPX0 generates a solution and then it is further improved by SVNS or GOAL solvers.

Table 6.11 displays for each method the average gap computed from 5 runs in each instance by using different seeds. A time limit of 1 hour was imposed to each method besides the time required for generating the initial solution. Further details for GOAL and SVNS solvers are reported, respectively, in appendices B and C.

Table 6.11 – Comparative results between SVNS and GOAL solvers.

	Group A				**Group B**				**Group C**			
Id	G_K	S_K	G_C	S_C	G_K	S_K	G_C	S_C	G_K	S_K	G_C	S_C
01	**5.41**	**5.41**	**5.4**	**5.41**	**5.41**	**5.41**	76.21	76.21	**5.41**	**5.41**	**5.41**	**5.41**
02	**4.76**	**4.76**	7.1	6.83	**4.76**	**4.76**	11.50	11.11	**4.76**	**4.76**	6.69	5.21
03	96.37	96.41	**25.8**	25.97	**96.16**	96.71	97.94	98.00	79.16	79.14	**34.00**	34.59
04	96.68	96.68	13.0	**11.88**	92.45	92.45	92.71	**91.86**	73.48	73.39	18.97	**18.67**
05	95.79	95.79	6.5	**5.48**	**93.02**	**93.02**	95.52	95.01	70.31	70.29	**16.13**	16.74
06	**0.00**	**0.00**	**0.0**	**0.00**	0.54	**0.00**	96.79	96.79	0.55	**0.00**	1.10	**0.00**
07	98.59	98.59	8.9	**6.11**	98.01	98.01	97.12	**96.84**	94.68	94.68	**25.00**	**25.00**
08	89.34	89.34	8.0	**6.57**	89.29	89.29	91.52	**86.57**	84.15	84.14	6.34	**2.98**
09	3.09	**1.75**	6.1	3.68	3.19	**1.74**	91.38	91.36	9.03	**5.53**	10.04	7.31
10	2.54	**2.13**	3.4	**2.13**	2.31	**1.06**	98.39	98.39	2.95	**2.13**	3.36	**2.13**
11	93.39	93.39	2.5	**1.53**	**90.47**	**90.47**	97.22	97.15	11.76	10.60	9.76	**7.41**
Avg_s	53.27	53.11	7.9	**6.87**	52.33	**52.08**	86.03	85.39	39.66	39.10	12.43	**11.40**
12	95.83	95.83	2.7	**2.15**	94.10	**93.99**	96.17	95.98	15.05	14.98	13.15	**12.28**
13	94.21	93.64	13.5	**11.09**	92.56	**91.13**	94.72	94.13	**18.43**	34.80	19.86	18.62
14	97.17	97.17	4.0	**3.80**	96.67	**96.55**	96.78	97.69	22.37	22.32	20.96	**18.89**
15	16.90	**16.08**	19.1	18.09	16.63	16.69	17.43	**15.80**	14.64	14.79	16.94	**14.07**
16	0.34	**0.28**	0.5	0.34	0.39	**0.28**	0.83	0.34	0.56	**0.28**	0.56	**0.28**
17	92.00	91.99	22.7	**20.15**	87.53	87.51	88.85	**85.00**	31.02	30.17	31.17	**28.44**
18	96.72	96.71	5.0	**4.07**	96.02	**95.37**	97.00	97.91	35.19	33.36	33.36	**29.97**
19	96.97	96.97	3.3	**2.54**	**97.02**	97.07	97.67	98.41	87.41	87.38	16.47	**13.65**
20	95.86	95.73	12.5	**9.64**	94.39	**94.02**	96.79	96.54	40.00	38.86	31.61	**29.47**
21	85.23	85.15	19.4	**17.50**	79.50	**79.39**	93.32	92.96	29.86	28.33	29.47	**27.65**
Avg_m	77.12	76.95	10.3	**8.94**	75.48	**75.20**	77.96	77.48	29.45	30.53	21.35	**19.33**
22	97.49	97.44	24.2	**18.83**	**97.41**	**97.41**	99.33	99.33	97.40	97.39	21.95	**17.66**
23	97.44	97.47	22.4	**20.26**	**97.42**	97.46	99.14	99.14	97.31	97.30	20.11	**17.16**
24	96.50	96.84	20.1	**16.40**	**96.20**	96.44	99.16	99.16	95.57	95.57	17.12	**13.63**
25	94.16	94.16	15.1	**12.55**	94.77	**94.76**	99.30	99.30	93.50	93.49	16.27	**12.98**
26	93.52	93.51	13.1	**10.56**	93.21	**93.20**	99.32	99.32	93.52	93.52	12.94	**10.45**
27	95.11	94.96	11.1	**8.11**	**94.86**	**94.86**	99.32	99.32	93.78	93.77	12.74	**9.38**
28	96.44	96.36	26.9	**25.19**	93.84	**93.83**	99.25	99.25	95.74	96.25	29.19	**26.68**
29	41.44	46.11	**32.1**	51.93	34.37	**18.87**	97.96	98.07	15.91	**15.69**	29.51	27.91
30	74.37	73.23	**39.3**	48.89	**74.52**	75.29	96.60	96.76	79.83	79.78	18.87	**18.36**
31	66.48	68.70	**16.1**	21.20	69.11	**66.38**	97.73	97.85	65.57	65.63	20.29	**18.95**
32	38.61	51.57	35.9	**35.02**	**10.88**	10.94	96.92	97.18	21.04	**20.44**	31.30	29.84
33	26.39	**8.47**	16.5	21.23	10.04	**8.55**	96.91	97.24	**19.52**	19.78	20.75	20.38
Avg_l	76.50	76.57	**22.7**	24.18	72.22	**70.67**	98.41	98.49	72.39	72.39	20.92	**18.61**
Avg_*	68.94	68.87	14.0	**13.79**	66.58	**65.85**	88.08	87.76	48.47	48.60	18.22	**16.43**

Source: created by author.

In order to compare the variants of GOAL and SVNS solvers, we performed a set of statistical tests presented in Table 6.12. The tests regarding hypothesis $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ reveal that there is no significant differences in performance between variants G_K and S_K since both solvers provided solutions with similar quality when compared in the same group. This results also suggests that the performance gains of SVNS over GOAL reported by (FONSECA; SANTOS, 2014) are possibly related to resource assignments.

The variants G_C and S_C were compared individually in each group by testing the hypothesis $\mathcal{H}_4$, $\mathcal{H}_5$, and $\mathcal{H}_6$. While in groups A and B no significant differences can be stated, in Group C the variant S_C clearly provided better solutions than G_C. Particularly this last test is the only significant evidence in which a SVNS variant demonstrated superiority over a GOAL variant. Hence, since in all other tests no differences were stated, in further tests we only evaluate SVNS variants.

The aim of the last three hypothesis tests is to identify how results obtained by SVNS variants are impacted due to different initial solutions. We observed that the initial solution has a strong impact on the quality of the final solutions. In groups A and C the variant S_C clearly outperformed S_K by providing feasible solutions far better. In contrast, in group B the variant S_K outperformed S_C, however, in this case the difference observed in the quality of solutions is more moderate. Thus, we concluded based on the experiments reported here that, since GOAL and SVNS, on the whole demonstrated similar performance as a local search based heuristic, the variants that started with a better initial solution provided the best results.

Table 6.12 – Paired Student's t-test comparing results obtained by GOAL and SVNS.

Hypothesis description	p-value	Result
$\mathcal{H}_1$: Gap of $\mathrm{S}_K \neq$ Gap of G_K in Group A	0.914	Failed
$\mathcal{H}_2$: Gap of $\mathrm{S}_K \neq$ Gap of G_K in Group B	0.135	Failed
$\mathcal{H}_3$: Gap of $\mathrm{S}_K \neq$ Gap of G_K in Group C	0.797	Failed
$\mathcal{H}_4$: Gap of $\mathrm{S}_C \neq$ Gap of G_C in Group A	0.788	Failed
$\mathcal{H}_5$: Gap of $\mathrm{S}_C \neq$ Gap of G_C in Group B	0.113	Failed
$\mathcal{H}_6$: Gap of $\mathrm{S}_C <$ Gap of G_C in Group C	0.000	Succeeded
$\mathcal{H}_7$: Gap of $\mathrm{S}_C <$ Gap of S_K in Group A	0.000	Succeeded
$\mathcal{H}_8$: Gap of $\mathrm{S}_C >$ Gap of S_K in Group B	0.001	Succeeded
$\mathcal{H}_9$: Gap of $\mathrm{S}_C <$ Gap of S_K in Group C	0.000	Succeeded

Source: created by author.

6.3.6 Experiments with the fix-and-optimize approach

In this section we present an experimental evaluation for the fix-and-optimize heuristic proposed in Chapter 4 applied on the model proposed to HSTP$^+$ in Section 6.1.1. Our goal is to evaluate the variant F8 presented in Table 4.3 without any fine-tuning. Table 6.13 presents gap results for two versions of the fix-and-optimize heuristic: a deterministic version (F8) and a stochastic version ($\overline{\text{F8}}$). While in the deterministic version, the subproblems are explored in a lexicographical order, in the stochastic version the partitions are shuffled when a neighborhood is changed. The average results reported to $\overline{\text{F8}}$ were computed from 5 runs performed in each instance by using different seeds. A time limit of 1 hour was imposed for both methods. Further details for F8 and $\overline{\text{F8}}$ are reported, respectively, in appendices D and E.

The statistical tests presented in Table 6.14 reveals there is no significant difference between the deterministic and the stochastic versions of the fix-and-optimize heuristic. Thus, we proceed with the analysis only with the stochastic version. In addition, the results obtained to the last two hypothesis also indicates that the fix-and-optimize heuristic is impacted by the changes we made on group A once the achieved solutions for groups B and C are significantly worse than solutions of group A. The worst results of fix-and-optimize heuristic occurred in group B. Possibly it was due to the poor quality of the initial solution provided by CPX0, as observed in previous experiments.

Table 6.13 – Comparison results between fix-and-optimize variants F8 and $\overline{\text{F8}}$.

	Group A		Group B		Group C	
Id	F8	$\overline{\text{F8}}$	F8	$\overline{\text{F8}}$	F8	$\overline{\text{F8}}$
01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
03	**4.60**	8.58	**5.79**	6.86	13.44	**9.95**
04	**0.72**	**0.72**	**0.72**	0.86	**0.74**	1.03
05	**0.00**	**0.00**	**0.00**	0.11	2.32	**1.72**
06	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
07	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
08	**0.50**	**0.50**	**0.00**	0.40	**1.56**	2.07
09	**1.26**	1.63	1.86	**1.50**	1.34	**0.54**
10	**2.13**	**2.13**	**1.06**	**1.06**	**2.13**	**2.13**
11	**0.00**	**0.00**	**0.70**	**0.70**	**0.00**	**0.00**
Avg_s	**0.84**	1.23	**0.92**	1.05	1.96	**1.59**
12	0.46	**0.37**	0.46	**0.37**	**2.33**	**2.33**
13	4.55	**3.85**	76.32	**76.31**	**6.94**	7.88
14	1.56	**0.39**	0.78	**0.55**	8.35	**7.50**
15	**3.49**	4.21	4.39	**3.76**	2.86	**1.64**
16	0.56	**0.50**	0.83	**0.45**	0.56	**0.50**
17	**2.81**	3.45	4.44	**3.44**	**10.59**	11.44
18	1.63	**0.49**	**0.00**	0.98	14.33	**11.49**
19	**0.00**	**0.00**	56.18	**0.54**	3.76	**3.00**
20	4.76	**2.60**	3.74	**2.17**	11.76	**10.36**
21	6.48	**6.09**	6.19	**5.41**	14.41	**8.95**
Avg_m	2.63	**2.19**	15.33	**9.40**	7.59	**6.51**
22	**11.01**	12.61	**11.21**	12.48	**12.29**	12.46
23	**10.60**	11.39	**51.17**	54.69	**12.86**	13.41
24	11.39	**10.73**	50.46	**32.81**	10.71	**9.72**
25	10.84	**10.08**	12.88	**10.52**	11.67	**11.23**
26	**8.21**	9.77	**8.87**	8.94	**8.21**	9.77
27	**6.89**	7.30	74.73	**74.68**	**7.66**	8.31
28	**8.87**	10.06	11.34	**10.39**	**10.20**	11.73
29	4.56	**4.21**	5.99	**5.58**	13.26	**12.01**
30	7.62	**7.17**	8.01	**7.50**	**12.99**	14.04
31	**6.51**	6.68	**5.85**	7.72	**14.18**	15.82
32	11.37	**10.37**	**8.33**	9.03	**16.24**	19.01
33	9.33	**8.80**	8.64	**7.68**	**19.06**	19.33
Avg_l	**8.93**	9.10	21.46	**20.17**	**12.44**	13.07
Avg_*	**4.32**	4.38	12.76	**10.53**	7.48	**7.25**

Source: created by author.

Table 6.14 – Paired Student's t-test comparing results of fix-and-optimize variants.

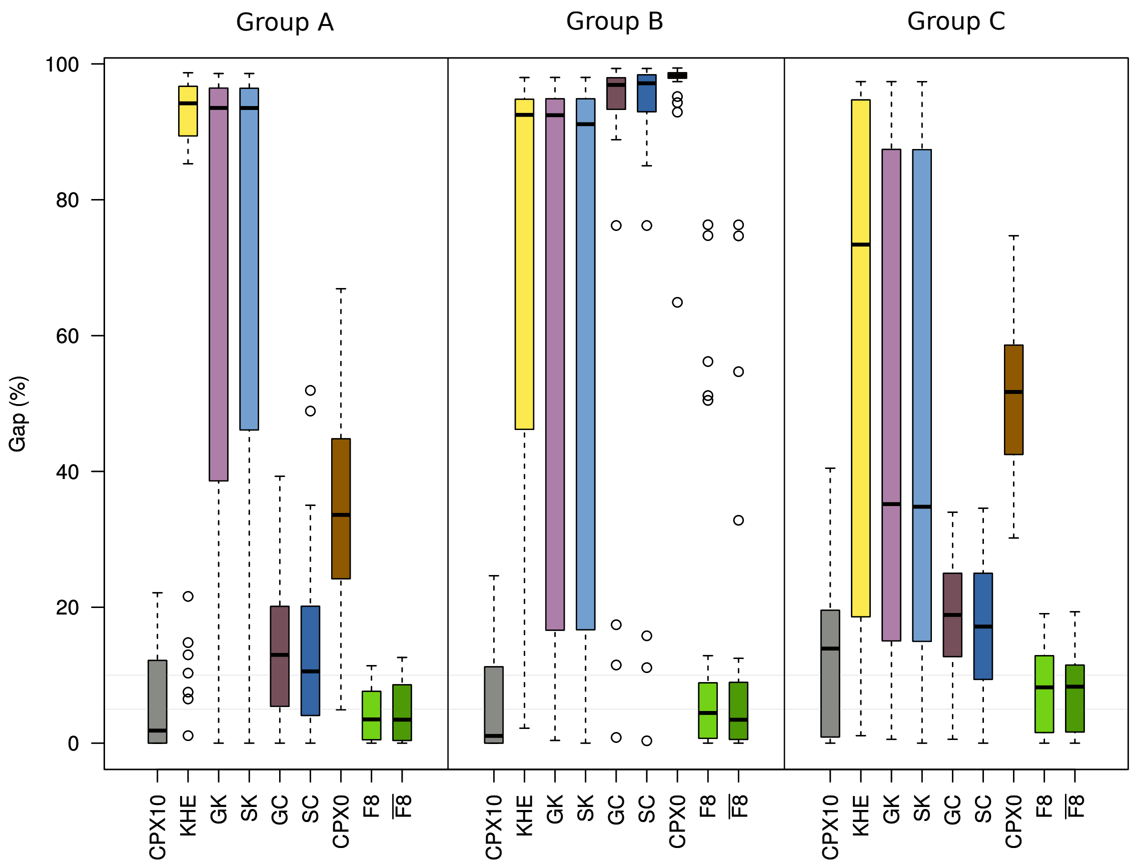
Hypothesis description	p-value	Result
$\mathcal{H}_1$: Gap of F8 $\neq$ Gap of $\overline{\text{F8}}$ in Group A	0.744	Failed
$\mathcal{H}_2$: Gap of F8 $\neq$ Gap of $\overline{\text{F8}}$ in Group B	0.216	Failed
$\mathcal{H}_3$: Gap of F8 $\neq$ Gap of $\overline{\text{F8}}$ in Group C	0.410	Failed
$\mathcal{H}_4$: Gap of $\overline{\text{F8}}$ in Group B $>$ Gap of $\overline{\text{F8}}$ in Group A	0.032	Succeeded
$\mathcal{H}_5$: Gap of $\overline{\text{F8}}$ in Group C $>$ Gap of $\overline{\text{F8}}$ in Group A	0.000	Succeeded

Source: created by author.

6.3.7 Comparative results

In this section we compare the methods tested in previous sections. Figure 6.2 presents a boxplot graphic comparing the average optimality gaps for all evaluated methods obtained within each group.

Figure 6.2 – Graphical comparison of optimality gaps between all approaches evaluated.



Source: Figure created by author.

Table 6.15 presents gap results for each method, and the best results are presented in bold. Results of CPLEX using a time limit of 10 hours are presented in column CPX_{10}. It can be observed that, on average, the fix-and-optimize heuristic outperformed all methods evaluated in groups A and C. In group B, the results are worse than CPLEX mainly due to the poor quality of the initial solution provided by CPX0 in this particular group. Finally, one may still note that SVNS solvers are significantly outperformed by other approaches in all group of instances. Although the variant S_C has gained some boost on performance in groups A and C due to CPX0, even taking into account this improvement it is not able to cope with other methods.

Table 6.15 – Comparison of optimality gaps between the main approaches evaluated.

	Group A				**Group B**				**Group C**			
Id	CPX_{10}	S_K	S_C	$\overline{F8}$	CPX_{10}	S_K	S_C	$\overline{F8}$	CPX_{10}	S_K	S_C	$\overline{F8}$
01	**0.00**	5.41	5.41	**0.00**	**0.00**	5.41	76.21	**0.00**	**0.00**	5.41	5.41	**0.00**
02	**0.00**	4.76	6.83	**0.00**	**0.00**	4.76	11.11	**0.00**	**0.00**	4.76	5.21	**0.00**
03	**0.87**	96.41	25.97	8.58	**0.87**	96.71	98.00	6.86	**0.90**	79.14	34.59	9.95
04	**0.72**	96.68	11.88	**0.72**	**0.72**	92.45	91.86	0.86	**0.74**	73.39	18.67	1.03
05	**0.00**	95.79	5.48	**0.00**	**0.00**	93.02	95.01	0.11	2.91	70.29	16.74	**1.72**
06	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	96.79	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
07	**0.00**	98.59	6.11	**0.00**	**0.00**	98.01	96.84	**0.00**	**0.00**	94.68	25.00	**0.00**
08	**0.50**	89.34	6.57	**0.50**	**0.00**	89.29	86.57	0.40	**1.56**	84.14	2.98	2.07
09	**0.63**	1.75	3.68	1.63	**0.00**	1.74	91.36	1.50	**0.00**	5.53	7.31	0.54
10	**2.13**	**2.13**	**2.13**	**2.13**	**1.06**	**1.06**	98.39	**1.06**	**2.13**	**2.13**	**2.13**	**2.13**
11	**0.00**	93.39	1.53	**0.00**	**0.70**	90.47	97.15	**0.70**	**0.00**	10.60	7.41	**0.00**
Avg_s	**0.44**	53.11	6.87	1.23	**0.31**	52.08	85.39	1.05	**0.75**	39.10	11.40	1.59
12	**0.00**	95.83	2.15	0.37	**0.00**	93.99	95.98	0.37	3.23	14.98	12.28	**2.33**
13	**2.78**	93.64	11.09	3.85	**2.33**	91.13	94.13	76.31	**2.43**	34.80	18.62	7.88
14	**0.00**	97.17	3.80	0.39	**0.00**	96.55	97.69	0.55	19.56	22.32	18.89	**7.50**
15	11.04	16.08	18.09	**4.21**	8.66	16.69	15.80	**3.76**	2.86	14.79	14.07	**1.64**
16	0.56	**0.28**	0.34	0.50	0.56	**0.28**	0.34	0.45	0.56	**0.28**	**0.28**	0.50
17	**1.84**	91.99	20.15	3.45	**2.23**	87.51	85.00	3.44	24.58	30.17	28.44	**11.44**
18	**0.00**	96.71	4.07	0.49	**0.00**	95.37	97.91	0.98	24.46	33.36	29.97	**11.49**
19	**0.00**	96.97	2.54	**0.00**	**0.38**	97.07	98.41	0.54	**2.62**	87.38	13.65	3.00
20	**0.00**	95.73	9.64	2.60	**0.00**	94.02	96.54	2.17	18.67	38.86	29.47	**10.36**
21	**5.99**	85.15	17.50	6.09	**4.71**	79.39	92.96	5.41	13.92	28.33	27.65	**8.95**
Avg_m	2.22	76.95	8.94	**2.19**	**1.89**	75.20	77.48	9.40	11.29	30.53	19.33	**6.51**
22	18.47	97.44	18.83	**12.61**	24.64	97.41	99.33	**12.48**	23.07	97.39	17.66	**12.46**
23	16.02	97.47	20.26	**11.39**	**17.92**	97.46	99.14	54.69	21.68	97.30	17.16	**13.41**
24	13.74	96.84	16.40	**10.73**	**13.23**	96.44	99.16	32.81	17.78	95.57	13.63	**9.72**
25	22.13	94.16	12.55	**10.08**	17.72	94.76	99.30	**10.52**	16.07	93.49	12.98	**11.23**
26	14.04	93.51	10.56	**9.77**	14.80	93.20	99.32	**8.94**	14.04	93.52	10.45	**9.77**
27	10.83	94.96	8.11	**7.30**	**9.78**	94.86	99.32	74.68	17.83	93.77	9.38	**8.31**
28	12.49	96.36	25.19	**10.06**	17.73	93.83	99.25	**10.39**	17.05	96.25	26.68	**11.73**
29	7.63	46.11	51.93	**4.21**	**4.10**	18.87	98.07	5.58	25.57	15.69	27.91	**12.01**
30	12.17	73.23	48.89	**7.17**	11.70	75.29	96.76	**7.50**	39.08	79.78	18.36	**14.04**
31	**2.40**	68.70	21.20	6.68	8.13	66.38	97.85	**7.72**	19.41	65.63	18.95	**15.82**
32	14.18	51.57	35.02	**10.37**	16.36	10.94	97.18	**9.03**	40.48	20.44	29.84	**19.01**
33	12.30	**8.47**	21.23	8.80	11.23	8.55	97.24	**7.68**	34.51	19.78	20.38	**19.33**
Avg_l	13.03	76.57	24.18	**9.10**	**13.95**	70.67	98.49	20.17	23.88	72.39	18.61	**13.07**
Avg_*	5.56	68.87	13.79	**4.38**	**5.75**	65.85	87.76	10.53	12.35	48.60	16.43	**7.25**

Source: created by author.

6.3.8 Best solutions found

Table 6.16 presents the best known solutions found for each instance considering all methods evaluated in this chapter. The column *Time* displays the time in seconds when the solution was found. In case two or more methods reached a solution with the same value, we reported only the quickest one. In addition, all these solutions and instances were submitted to the High School Benchmarking Project website and might be available online soon. From this table, it can be seen that still there is space for improvements on several instances, specially in the larger ones, as well as medium and large instances of group C.

Table 6.16 – Best solutions found in this study.

	Group A				**Group B**				**Group C**			
Id	Method	Obj	Gap	Time	Method	Obj	Gap	Time	Method	Obj	Gap	Time
01	$\overline{\text{F8}}$	315	0.00	0.4	$\overline{\text{F8}}$	315	0.00	0.7	$\overline{\text{F8}}$	315	0.00	0.4
02	CPX$_{10}$	360	0.00	2.2	CPX$_{10}$	360	0.00	1.6	CPX$_{10}$	360	0.00	2.3
03	CPX$_{10}$	690	0.87	1911.1	CPX$_{10}$	690	0.87	13827.0	CPX$_{10}$	663	0.90	22662.5
04	CPX$_{10}$	417	0.72	2822.1	$\overline{\text{F8}}$	417	0.72	257.1	F8	408	0.74	484.4
05	$\overline{\text{F8}}$	528	0.00	199.9	F8	528	0.00	11.4	$\overline{\text{F8}}$	489	1.72	352.3
06	$\overline{\text{F8}}$	333	0.00	1.2	CPX$_{10}$	333	0.00	0.7	CPX$_{10}$	324	0.00	0.9
07	CPX$_{10}$	387	0.00	2.6	CPX$_{10}$	387	0.00	7.0	$\overline{\text{F8}}$	351	0.00	35.0
08	$\overline{\text{F8}}$	600	0.50	167.1	$\overline{\text{F8}}$	600	0.00	87.2	F8	576	1.56	852.9
09	CPX$_{10}$	474	0.63	4376.2	G$_K$	474	0.00	2032.0	$\overline{\text{F8}}$	441	0.00	1373.5
10	$\overline{\text{F8}}$	282	2.13	8.0	F8	282	1.06	7.6	$\overline{\text{F8}}$	282	2.13	8.1
11	$\overline{\text{F8}}$	426	0.00	6.5	F8	426	0.70	4.9	$\overline{\text{F8}}$	405	0.00	5.9
12	CPX$_{10}$	654	0.00	1637.2	$\overline{\text{F8}}$	654	0.00	105.1	F8	645	2.33	286.5
13	CPX$_{10}$	648	2.78	11083.9	CPX$_{10}$	645	2.33	35434.6	CPX$_{10}$	618	2.43	35676.8
14	CPX$_{10}$	759	0.00	817.0	CPX$_{10}$	759	0.00	716.2	$\overline{\text{F8}}$	699	6.78	1095.3
15	$\overline{\text{F8}}$	636	3.49	769.3	$\overline{\text{F8}}$	636	3.49	2117.2	$\overline{\text{F8}}$	618	0.97	154.3
16	S$_K$	1074	0.28	82.0	S$_K$	1074	0.28	156.0	S$_K$	1074	0.28	73.0
17	$\overline{\text{F8}}$	903	1.84	3024.8	CPX$_{10}$	906	2.23	35953.6	$\overline{\text{F8}}$	864	9.66	2363.8
18	CPX$_{10}$	1089	0.00	196.7	CPX$_{10}$	1089	0.00	361.6	$\overline{\text{F8}}$	963	8.45	3506.4
19	$\overline{\text{F8}}$	783	0.00	174.9	$\overline{\text{F8}}$	783	0.38	1582.8	$\overline{\text{F8}}$	747	1.45	1897.9
20	CPX$_{10}$	540	0.00	35124.0	$\overline{\text{F8}}$	540	0.00	2311.5	$\overline{\text{F8}}$	450	10.00	2566.9
21	$\overline{\text{F8}}$	567	4.50	1042.3	$\overline{\text{F8}}$	573	4.71	1652.9	$\overline{\text{F8}}$	495	8.18	2644.0
22	F8	984	11.01	2453.2	$\overline{\text{F8}}$	990	10.94	1348.1	$\overline{\text{F8}}$	900	10.25	2146.8
23	$\overline{\text{F8}}$	1206	10.38	3554.9	CPX$_{10}$	1323	17.92	34549.4	$\overline{\text{F8}}$	948	11.76	3559.6
24	$\overline{\text{F8}}$	1170	8.21	3550.6	$\overline{\text{F8}}$	1176	8.58	2295.6	$\overline{\text{F8}}$	1020	8.34	1858.7
25	$\overline{\text{F8}}$	1179	9.25	1539.4	$\overline{\text{F8}}$	1179	9.56	2484.6	$\overline{\text{F8}}$	1188	10.56	1495.0
26	F8	1458	8.21	2756.4	$\overline{\text{F8}}$	1449	7.92	2798.9	F8	1458	8.21	2740.1
27	$\overline{\text{F8}}$	1491	6.71	1947.1	CPX$_{10}$	1539	9.78	31857.7	$\overline{\text{F8}}$	1449	7.28	2806.7
28	$\overline{\text{F8}}$	1443	8.49	2573.7	$\overline{\text{F8}}$	1416	8.15	3463.4	$\overline{\text{F8}}$	1377	10.00	2161.0
29	$\overline{\text{F8}}$	1335	3.48	3421.9	CPX$_{10}$	1344	4.10	31889.8	$\overline{\text{F8}}$	1107	10.44	3496.1
30	$\overline{\text{F8}}$	1837	6.72	3539.8	$\overline{\text{F8}}$	1843	6.96	1483.5	F8	1233	12.99	3396.3
31	CPX$_{10}$	1636	2.40	35927.1	F8	1693	5.85	2181.4	$\overline{\text{F8}}$	1377	13.62	3460.7
32	$\overline{\text{F8}}$	1771	8.51	3319.7	F8	1780	8.33	2817.5	F8	1422	16.24	3398.2
33	$\overline{\text{F8}}$	1801	6.46	2680.7	$\overline{\text{F8}}$	1810	6.67	2047.5	$\overline{\text{F8}}$	1575	17.68	3561.0

Source: created by author.

6.4 Conclusions

In this chapter is presented a novel and high-constrained variant of the High School Timetabling Problem referred as the Extended High School Timetabling Problem (HTSP^{+}). Here, we defined HSTP^{+} in terms of hard, soft and medium requirements through two models: a Mixed Integer Programming model that generalizes HSTP, and an XHSTT model that formalizes the HSTP$^+$ as a subproblem of the Generalized High School Timetabling Problem (GHSTP).

The computational experiments were carried out on a novel set composed by 33 real-world instances originated from Brazilian schools. In this experiments we evaluated the performance of the fix-and-optimize approach proposed in Chapter 4 in comparison with a state-of-the-art MIP solver, as well as two state-of-the-art local search based solvers designed for solving the GHSTP. When analyzing results, conclusions we had drawn were supported by statistical analysis.

The obtained results show strong evidence that the fix-and-optimize approach is suitable for solving the HSTP$^+$. In addition to provide quick feasible solutions, it was able to produce high quality solutions for the majority of the instances evaluated. The comparative results also demonstrate that the fix-and-optimize approach significantly outperforms the other tested methods when solving real-world instances of the HSTP$^+$.

7 FINAL CONSIDERATIONS

7.1 Conclusions

The research carried out in this thesis presents as major contribution a novel matheuristic approach that is able to produce high quality feasible solutions for the High School Timetabling Problem. Different from the majority of the works proposed in the related literature, in this research, the claims about the performance of the proposed approach is stated in a large set of real-world instances and comparisons to previously proposed methods are conducted, as well as to state-of-the-art MIP solvers.

This investigation focused in two variants of the High School Timetabling Problem originated from Brazilian institutions. The first variant, denoted as HSTP, is a well-known problem previously proposed in the literature that has gained increasing attention due to be part of the set of instances considered in the Third International Timetabling Competition. The second variant, denoted as HSTP$^+$, is a new problem that was introduced in this research in order to validate the proposed methods in a practical scenario using a broader set of instances. The HSTP$^+$ is formally defined in this thesis through both a MIP formulation and a XHSTT model.

In addition to defining the studied problems, along this research, several MIP models were proposed in order to evaluate the performance of MIP solvers. The experimental results performed on HSTP and HSTP$^+$ instances revealed that MIP solvers are best suitable for providing solutions to small instances and, particularly in these cases, they often produce better solutions that other compared methods by using a realistic time limit. This study also investigated how certain requirements may affect the resolution process of a MIP solver. It was found that among the soft requirements, the idle times constraint is the one that most degraded the performance of the solver. It was also observed that if all soft requirements are disregarded, the MIP solver is able to find a feasible solution in few seconds, even for the largest instances evaluated. Particularly this feature demonstrated to be an effective approach for producing initial feasible solutions in real-world conditions.

A novel approach was proposed for solving the HSTP by exploring class, teacher and day decompositions through a fix-and-optimize heuristic combined with a variable neighborhood descent method. A initial experimental investigation on HSTP demonstrated that this novel approach is able to provide high quality feasible solutions in a smaller computational time when compared with results obtained by a MIP solver. The effectiveness of this approach was also demonstrated by producing new best known solutions for several instances quoted in the literature. Among these new results, better solutions were found to four out of five HSTP instances from the first round of the Third International Timetabling Competition. A further set of experiments carried out on HSTP$^+$ and supported by statistical analysis, provided stronger evidence about the potential of the fix-and-optimize heuristic in providing high quality results for real-world instances. Furthermore, a set of experiments were conducted with two state-of-the-art local search based solvers designed for solving the GHSTP, referred as GOAL and SVNS. The obtained results demonstrated these solvers are not efficient for solving instances of the HSTP$^+$. A comparative experiment also revealed that they are outperformed by the fix-and-optimize approach, as well as by a MIP solver. While the results of these comparisons should be interpreted with reservations, since both GOAL and SVNS solvers are designed for a more general problem, the results obtained in the HSTP$^+$ instances demonstrated clearly that there is space for improvements in solvers designed to GHSTP.

In order to better state the quality of heuristic solutions provided by the proposed fix-and-optimize algorithm, a column generation approach was proposed for producing lower bounds to the HSTP by using a novel multicommodity flow representation. In comparison with the previous state-of-the-art approach, the experimental results show that the proposed approach is able to produce the same tight lower bounds, albeit with two significant advantages: i) the method is simpler; ii) and it is five times faster on average. During the experimental evaluations, best known lower bounds were found for all instances considered in the first round of the

Third International Timetabling Competition.

Finally, this work helped to reduce one of the major issues in the high school timetabling that is the lack of available test cases. All instances and results collected to the HSTP^{+} were submitted to the High School Benchmark Project website. To the best of our knowledge, this is the largest set of real-world instances that have been made publicly available for the High School Timetabling problem in an standardized format like the XHSTT. Hopefully, this contribution will help in stablishing a strong foundation for future investigation in the High School Timetabling Problem.

7.2 Perspectives

As future work, there are several directions in which the research conducted in this work can be extended. The main ones are discussed next.

7.2.1 Selection of fruitful partitions in the fix-and-optimize heuristic

In the fix-and-optimize heuristic proposed in this research we explored the subproblems generated by different neighborhoods and decompositions through two simple orders: lexicographical and random. While these approaches are problem-independent, they may limit the performance of the algorithm by wasting time when unfruitful partitions are chosen to be optimized, i.e., the most of the selected partitions lead to either infeasible state or terminate due to time limit. An interesting direction for future research would be to use a metric for predicting fruitful partitions for optimization such as the Hamming-Oriented Partition Search (HOPS) proposed by (CAMARGO; TOLEDO; ALMADA-LOBO, 2014). In the HOPS approach, a partition is chosen in a deterministic way by using information obtained from an *Partion Attractivness Array* that is updated along the search whenever a new incumbent solution is found. The array is updated in such way to give highest priority to partitions with variables that often change their values.

7.2.2 A Branch-and-price approach

To the best of our knowledge, there is no Branch-and-price approach proposed in the literature for solving the High School Timetabling Problem. Next, we describe three potential improvements in which the column generation approach we proposed in Chapter 5 may be extended in order to provide a practicable Branch-and-price method. Firstly, one may propose a tailored method for solving $\mathcal{P}_t$ or $\mathcal{P}'_t$ in a more efficient way than using a generic MIP solver. Secondly, since about 90% of the computational time is spent in the pricing step, the gains with parallelization might be promising. In fact, given that one pricing is solved for each teacher, they could be trivially solved in parallel. Finally, in order to harness the full potential of the column generation, one may propose branching strategies inside a branch-and-price framework for providing, ultimately, optimal integer solutions for the problem.

7.2.3 Methods for resource assignment

The development of diferent techniques for resource assignment is another promising research area in High School Timetabling. In the problems HSTP and HSTP$^+$ presented in this thesis, the resource assignment is previously done by the school board. A natural extension for the fix-and-optimize heuristic proposed in Chapter 4 is the inclusion of mechanisms for handling resource assignments programatically. This addition, may allow the resolution of GHSTP in a two-stage approach: first, assigning resources and then, proceed to a improvement stage carried out by a fix-and-optimize heuristic.

REFERENCES

ABRAMSON, D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. **Management Science**, INFORMS, Catonsville, USA, v. 37, n. 1, p. 98–113, 1991.

AVELLA, P.; D'AURIA, B.; SALERNO, S.; VASIL'EV, I. A computational study of local search algorithms for Italian high-school timetabling. **Journal of Heuristics**, Springer US, New York, USA, v. 13, p. 543–556, 2007.

BELLO, G.; RANGEL, M.; BOERES, M. An approach for the class/teacher timetabling problem using graph coloring. In: **INTERNATIONAL CONFERENCE ON PRACTICE AND THEORY OF AUTOMATED TIMETABLING**. 7, 2008. **Proceedings...** Montreal, Canada: [s.n.], 2008.

Benchmarking Project. **Benchmarking Project for (High) School Timetabling**. 2015. <http://www.utwente.nl/ctit/hstt>. Accessed: 2015-07-23.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, ACM, New York, NY, USA, v. 35, n. 3, p. 268–308, 2003.

BRITO, S. S.; FONSECA, G. H.; TOFFOLO, T. a.M.; SANTOS, H. G.; SOUZA, M. J. A SA-VNS approach for the High School Timetabling Problem. **Electronic Notes in Discrete Mathematics**, [s.l.], v. 39, p. 169–176, 2012.

BURKE, E.; KENDALL, G.; SOUBEIGA, E. A tabu-search hyperheuristic for timetabling and rostering. **Journal of Heuristics**, Springer US, New York, USA, v. 9, n. 6, p. 451–470, 2003.

BURKE, E. K.; BYKOV, Y. **The Late Acceptance Hill-Climbing Heuristic**. 2012. <http://www.cs.stir.ac.uk/research/publications/techreps/pdf/TR192.pdf>. Accessed: 2015-07-23.

BURKE, E. K.; MARECEK, J.; PARKES, A. J.; RUDOVá, H. Decomposition, reformulation, and diving in university course timetabling. **Computers & Operations Research**, Elsevier, Oxford, England, v. 37, n. 3, p. 582–597, 2010.

CALDEIRA, J.; ROSA, A. School timetabling using genetic search. In: **INTERNATIONAL CONFERENCE OF PRACTICE AND THEORY OF AUTOMATED TIMETABLING**. 2, 1997. **Proceedings...** Toronto, Canada: [s.n.], 1997. p. 115–122.

CAMARGO, V. C.; TOLEDO, F. M.; ALMADA-LOBO, B. {HOPS} – Hamming-Oriented Partition Search for production planning in the spinning industry. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 234, n. 1, p. 266 – 277, 2014.

CHANSIRINUKOR, W.; WILSON, D.; GRIMMER, K.; DANSIE, B. Effects of backpacks on students: Measurement of cervical and shoulder posture. **Australian Journal of Physiotherapy**, Elsevier, [s.l], v. 47, n. 2, p. 110–116, 2001.

COLORNI, A.; DORIGO, M. Metaheuristics for High School Timetabling. **Computational Optimization and Applications**, Springer, New York, USA, v. 9, n. 3, p. 275–298, 1998.

COSTA, D. A tabu search algorithm for computing an operational timetable. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 76, n. 1, p. 98–110, 1994.

DANNA, E.; ROTHBERG, E. Exploring relaxation induced neighborhoods to improve MIP solutions. **Mathematical Programming**, Springer, Berlin, Germany, v. 90, p. 71–90, 2005.

DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations research**, INFORMS, Catonsville, USA, v. 8, n. 1, p. 101–111, 1960.

DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. The impact of compactness requirements on the resolution of high school timetabling problem. In: **SIMPOSIO BRASILEIRO DE PESQUISA OPERACIONAL**. 44, 2012. **Anais...** Rio de Janeiro, Brazil: Sociedade Brasileira de Pesquisa Operacional, 2012. p. 3336–3347.

DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. A fix-and-optimize heuristic for the high school timetabling problem. **Computers & Operations Research**, Elsevier, Oxford, England, v. 52, p. 29–38, 2014.

DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. A column generation approach to the high school timetabling modeled as a multicommodity flow problem. **European Journal of Operational Research**, Elsevier, Berlin, Germany, 2015. (Submitted).

DORNELES, Á. P.; ARAÚJO, O. C. B.; LANDA-SILVA, D.; BURIOL, L. S. Solving large high school timetabling problems in Brazil by using fix-and-optimize and local branching. **European Journal of Operational Research**, Elsevier, Berlin, Germany, 2016. (Submitted).

DREXL, A.; SALEWSKI, F. Distribution requirements and compactness constraints in school timetabling. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 102, n. 1, p. 193–214, 1997.

DUMITRESCU, I.; STÜTZLE, T. Combinations of local search and exact algorithms. In: CAGNONI, S. et al. (Ed.). **Applications of Evolutionary Computing**. Berlin, Germany: Springer Berlin Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2611). p. 211–223.

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

FEO, T.; RESENDE, M. A probabilistic heuristic for a computationally difficult set covering problem. **Operations research letters**, Elsevier, [s.l.], v. 8, n. 2, p. 67–71, 1989.

FILHO, G.; LORENA, L. A constructive evolutionary approach to school timetabling. In: BOERS, E. (Ed.). **Applications of Evolutionary Computing**. Berlin, Germany: Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2037). p. 130–139.

FISCHETTI, M.; LODI, A. Local branching. **Mathematical Programming**, Springer, Berlin, Germany, v. 98, n. 1-3, p. 23–47, 2003.

FONSECA, G.; BRITO, S.; SANTOS, H. A simulated annealing based approach to the high school timetabling problem. In: YIN, H.; COSTA, J.; BARRETO, G. (Ed.). **Intelligent Data Engineering and Automated Learning - IDEAL 2012**. Berlin, Germany: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7435). p. 540–549.

FONSECA, G. H.; SANTOS, H. G. Variable neighborhood search based algorithms for high school timetabling. **Computers & Operations Research**, Elsevier, Oxford, England, v. 52, Part B, p. 203 – 208, 2014.

FONSECA, G. H.; SANTOS, H. G.; TOFFOLO, T. a.M.; BRITO, S. S.; SOUZA, M. J. A SA-ILS approach for the high school timetabling problem. In: **INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING**. 9, 2012. **Proceedings...** Son, Norway: [s.n.], 2012.

GANS, O. de. A computer timetabling system for secondary schools in the netherlands. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 7, n. 2, p. 175–182, 1981.

GINTNER, V.; KLIEWER, N.; SUHL, L. Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. **OR Spectrum**, Springer, Berlin, Germany, v. 27, n. 4, p. 507 – 523, 2005.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, Elsevier, Oxford, England, v. 13, n. 5, p. 533–549, 1986.

GOTLIEB, C. The construction of class-teacher timetables. In: **INTERNATIONAL FEDERATION OF INFORMATION PROCESSING**. 1, 1962. **Proceedings...** Amsterdam, Germany: North-Holland, 1962. p. 73–77.

GUNAWAN, A.; NG, K. M.; POH, K. L. A hybridized lagrangian relaxation and simulated annealing method for the course timetabling problem. **Computers & Operations Research**, Elsevier, Oxford, England, v. 39, n. 12, p. 3074 – 3088, 2012.

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 130, n. 3, p. 449–467, 2001.

HANSEN, P.; MLADENOVIĆ, N.; UROŠEVIĆ, D. Variable neighborhood search and local branching. **Computers & Operations Research**, Elsevier, Oxford, England, v. 33, n. 10, p. 3034–3045, 2006.

HOLLAND, J. **Adaptation in natural and artificial systems**. Oxford, England: University of Michigan Press, 1975. 183 p.

IBM. **ILOG CPLEX 12.1 User's Manual**. Mountain View, CA, 2009. Accessed: 2015-07-23. Available from Internet: <http://www.ilog.com/products/cplex>.

IBM. **ILOG CPLEX 12.6 User's Manual**. Mountain View, CA, 2015. Accessed: 2015-07-23. Available from Internet: <http://www.ilog.com/products/cplex>.

ITC. **Third International Timetabling Competition**. 2011. <http://www.utwente.nl/ctit/hstt/itc2011/>. Accessed: 2015-07-23.

JOURDAN, L.; BASSEUR, M.; TALBI, E.-G. Hybridizing exact methods and metaheuristics: A taxonomy. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 199, n. 3, p. 620–629, 2009.

JUNGINGER, W. Timetabling in Germany - A survey. **Interfaces**, INFORMS, Catonsville, USA, v. 16, n. 4, p. 66–74, 1986.

KINGSTON, J. H. **High School Timetable Data Format Specification**. 2014. <http://sydney.edu.au/engineering/it/~jeff/hseval.cgi?op=spec>. Accessed: 2015-08-03.

KINGSTON, J. H. **KHE web site**. 2014. <http://www.it.usyd.edu.au/~jeff/khe>. Accessed: 2014-09-12.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, American Association for the Advancement of Science, [S.l.], v. 220, n. 4598, p. 671–680, 1983.

KISTNER, F.; FIEBERT, I.; ROACH, K. Effect of backpack load carriage on cervical posture in primary schoolchildren. **Work**, IOS Press, Clifton, USA, v. 41, n. 1, p. 99–108, 2012.

KISTNER, F.; FIEBERT, I.; ROACH, K.; MOORE, J. Postural compensations and subjective complaints due to backpack loads and wear time in schoolchildren. **Pediatric Physical Therapy**, Wolters Kluwer, S.l., v. 25, n. 1, p. 15–24, 2013.

LABIC. **Instances of School Timetabling**. 2008. <http://labic.ic.uff.br/Instance/index.php?dir=SchoolTimetabling/>. Accessed: 2015-07-23.

LODI, A. Mixed integer programming computation. In: JÜNGER, M.; LIEBLING, T. M.; NADDEF, D.; NEMHAUSER, G. L.; PULLEYBLANK, W. R.; REINELT, G.; RINALDI, G.; WOLSEY, L. A. (Ed.). **50 Years of Integer Programming 1958-2008**. Berlin, Germany: Springer, 2010. p. 619–645.

LOURENÇO, H.; MARTIN, O.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). **Handbook of Metaheuristics**. New York, USA: Springer US, 2003, (International Series in Operations Research & Management Science, v. 57). p. 320–353.

LÜBBECKE, M. E.; DESROSIERS, J. Selected topics in column generation. **Operations Research**, INFORMS, Catonsville, USA, v. 53, n. 6, p. 1007–1023, 2005.

MANIEZZO, V.; STÜTZLE, T.; VOSS, S. **Matheuristics: Hybridizing metaheuristics and mathematical programming**. New York, USA: Springer, 2009.

MARTE, M. Towards constraint-based school timetabling. **Annals of Operations Research**, Springer, New York, USA, v. 155, n. 1, p. 207–225, 2007.

MARTÍ, R.; LAGUNA, M.; GLOVER, F. Principles of scatter search. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 169, n. 2, p. 359–372, 2006.

MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, Elsevier, Oxford, England, v. 24, n. 11, p. 1097–1100, 1997.

MOURA, A.; SCARAFICCI, R. A GRASP strategy for a more constrained School Timetabling Problem. **International Journal of Operational Research**, Inderscience, [S.l.], v. 7, n. 2, p. 152–170, 2010.

NEMHAUSER, G. L.; WOLSEY, L. A. **Integer and combinatorial optimization**. New York, USA: Wiley, 1988.

NEUFELD, G.; TARTAR, J. Graph coloring conditions for the existence of solutions to the timetable problem. **Communications of the ACM**, ACM, [S.l.], v. 17, n. 8, p. 450–453, 1974.

OSTERMANN, R.; WERRA, D. de. Some experiments with a timetabling system. **OR Spectrum**, Springer, Berlin, Germany, v. 3, n. 4, p. 199–204, 1982.

PAPOULIAS, D. The assignment-to-days problem in a school time-table, a heuristic approach. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 4, n. 1, p. 31–41, 1980.

PAPOUTSIS C. VALOUXIS, E. H. K. A Column Generation Approach for the Timetabling Problem of Greek High Schools. **The Journal of the Operational Research Society**, Palgrave Macmillan Journals, [S.l.], v. 54, n. 3, p. 230–238, 2003.

PILLAY, N. A survey of school timetabling research. **Annals of Operations Research**, Springer, New York, USA, v. 218, n. 1, p. 261–293, 2014.

POCHET, Y.; WOLSEY, L. Mixed integer programming algorithms. In: **Production planning by mixed integer programming**. New York, USA: Springer, 2006. p. 77–113.

POST, G.; AHMADI, S.; DASKALAKI, S.; KINGSTON, J. H.; KYNGAS, J.; NURMI, C.; RANSON, D. An XML format for benchmarks in High School Timetabling. **Annals of Operations Research**, Springer US, New York, USA, v. 194, n. 1, p. 385–397, 2010.

POST, G.; GASPERO, L.; KINGSTON, J. H.; MCCOLLUM, B.; SCHAERF, A. The Third International Timetabling Competition. **Annals of Operations Research**, Springer US, New York, USA, v. 239, n. 1, p. 69–75, 2013.

POST, G. et al. XHSTT: an XML archive for high school timetabling problems in different countries. **Annals of Operations Research**, Springer US, New York, USA, v. 218, n. 1, p. 295–301, 2011.

POULSEN, C. J. B. P.; BANDEIRA, D. L. Uma eficiente heurística baseada na estrategia de divisao-e-conquista para o School Timetabling Problem. In: **SIMPOSIO BRASILEIRO DE PESQUISA OPERACIONAL**. 45, 2013. **Anais...** Natal, Brazil: Sociedade Brasileira de Pesquisa Operacional, 2013. p. 812–823.

PUCHINGER, J.; RAIDL, G. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: MIRA, J.; ÁLVAREZ, J. (Ed.). **Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach**. Berlin, Germany: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3562). p. 41–53.

RAIDL, G. A unified view on hybrid metaheuristics. In: ALMEIDA, F.; AGUILERA, M. B.; BLUM, C.; VEGA, J. M.; PéREZ, M. P.; ROLI, A.; SAMPELS, M. (Ed.). **Hybrid Metaheuristics**. Berlin, Germany: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 4030). p. 1–12.

SANTOS, H. G.; OCHI, L. S.; SOUZA, M. J. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. **Journal of Experimental Algorithmics**, ACM, New York, NY, USA, v. 10, 2005.

SANTOS, H. G.; UCHOA, E.; OCHI, L. S.; MACULAN, N. Strong bounds with cut and column generation for class-teacher timetabling. **Annals of Operations Research**, Springer, New York, USA, v. 194, p. 399–412, 2012.

SCHAERF, A. Local search techniques for large high school timetabling problems. **IEEE Transactions on Systems, Man and Cybernetics, Part A**, IEEE, [S.l.], v. 29, n. 4, p. 368–377, 1999.

SCHAERF, A. A survey of automated timetabling. **Artificial Intelligence Review**, Kluwer Academic Publishers, [S.l.], v. 13, n. 2, p. 87–127, 1999.

SCHAERF, A.; GASPERO, L. D. Local search techniques for educational timetabling problems. In: **International Symposium on Operations Research**. 6, 2001. **Proceedings...** Preddvor, Slovenia: [s.n.], 2001. p. 13–23.

SCHMIDT, G.; STRÖHLEIN, T. Timetable construction–An annotated bibliography. **The Computer Journal**, Oxford University Press, Oxford, England, v. 23, n. 4, p. 307–316, 1980.

SOUZA, M. **Programação de horários em escolas: Uma aproximação por metaheuristicas**. 160 p. Thesis (PhD) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, 2000.

SOUZA, M.; MACULAN, N. Melhorando quadros de horário de escolas através de caminhos mínimos. **Tendências em Matemática Aplicada e Computacional**, Sociedade Brasileira de Matemática Aplicada e Computacional, São Carlos, Brazil, v. 1, n. 2, p. 515–524, 2000.

SOUZA, M.; OCHI, L.; MACULAN, N. Metaheuristics: Computer decision-making. In: ______. Boston, USA: Springer US, 2004. chp. A GRASP-Tabu Search Algorithm for Solving School Timetabling Problems, p. 659–672.

TILLETT, P. An operations research approach to the assignment of teachers to courses. **Socio-Economic Planning Sciences**, Elsevier, [S.l.], v. 9, n. 3, p. 101–104, 1975.

TRIPATHY, A. School timetabling – A case in large binary integer linear programming. **Management Science**, INFORMS, Catonsville, USA, v. 30, n. 12, p. 1473–1489, 1984.

VALOUXIS, C.; HOUSOS, E. Constraint programming approach for school timetabling. **Computers & Operations Research**, Elsevier, Oxford, England, v. 30, n. 10, p. 1555–1572, 2003.

WERRA, D. de. Construction of school timetables by flow methods. **INFOR**, [s.n.], [S.l.], n. 9, p. 12–22, 1971.

WERRA, D. de. An introduction to timetabling. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 19, n. 2, p. 151–162, 1985.

WOLSEY, L. A. **Integer programming**. London: Wiley, 1998. 288 p.

ZHANG, D.; LIU, Y.; M'HALLAH, R.; LEUNG, S. C. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. **European Journal of Operational Research**, Elsevier, Berlin, Germany, v. 203, n. 3, p. 550–558, 2010.

GLOSSARY

Class is a set of students that share the same teaching program.

Curriculum is a set of informations that defines a teaching program.

Cycle is the period of time which a timetable is enforced. One-year and one-semester are the most common cycles. After the cycle a new timetable must be built.

Double lessons are lessons given in two consecutive periods.

Event is a meeting between class and teacher to address a particular subject in a given number of lessons allocated in a given room.

Idle period is a free period of time between two lessons.

Lesson is a particular event associated with a timeslot.

Period is the standard duration of lessons.

Requirement is an attribute or characteristic that needs to be provided or handled in a timetable.

Resource is the generic name given to entities which are needed by an event, e.g., teachers, rooms, etc.

Room is any place where the events occur, e.g., a classroom, a sports court or a chemistry lab.

Subject is a topic taught, e.g., Mathematics.

Teacher is a person who provides education for the students. We refer as teacher anyone who is responsible to administer events.

Timeslots are the periods along the week which a lesson can be scheduled.

Timetable is a table that presents the time in which all events of an institution occur.

Timetabler is the professional responsible to construct the timetable.

APPENDIX A — RESULTS OF CPLEX SOLVER ON HSTP$^+$

Table A.1 – CPLEX results using a time limit of 10 hours on instances of group A.

Id	Obj	LB	Gap	Gap_L	t^*	Time	Columns	Rows	x_0	t_0
01	315	315.0	0.00	0.0	1	1	3.7×10^3	1.7×10^3	624	0
02	360	360.0	0.00	0.0	2	2	4.6×10^3	2.3×10^3	360	2
03	690	684.0	0.87	0.9	1911	36000	8.5×10^3	4.3×10^3	831	89
04	417	414.0	0.72	0.7	2822	36000	6.7×10^3	3.6×10^3	498	7
05	528	528.0	0.00	0.0	282	3915	7.7×10^3	4.2×10^3	576	12
06	333	333.0	0.00	0.0	2	2	5.3×10^3	2.4×10^3	1026	0
07	387	387.0	0.00	0.0	3	9	1.5×10^4	7.9×10^3	423	2
08	600	597.0	0.50	0.5	284	36000	9.2×10^3	4.3×10^3	690	19
09	474	471.0	0.63	0.6	4376	36000	1.1×10^4	5.2×10^3	513	7
10	282	276.0	2.13	2.2	71	36000	7.9×10^3	3.6×10^3	510	4
11	426	426.0	0.00	0.0	35	13287	1.1×10^4	5.9×10^3	453	2
12	654	654.0	0.00	0.0	1637	3027	1.3×10^4	7.1×10^3	675	68
13	648	630.0	2.78	2.9	11084	36000	1.7×10^4	8.9×10^3	720	99
14	759	759.0	0.00	0.0	817	834	1.6×10^4	8.8×10^3	795	55
15	690	613.8	11.04	12.4	34764	36000	1.8×10^4	10×10^3	735	317
16	1077	1071.0	0.56	0.6	2546	36000	1.6×10^4	7.7×10^3	2025	45
17	903	886.4	1.84	1.9	13661	36000	3.1×10^4	1.4×10^4	1173	48
18	1089	1089.0	0.00	0.0	197	237	3.4×10^4	1.6×10^4	1173	38
19	783	783.0	0.00	0.0	1488	22162	2.5×10^4	1.3×10^4	795	37
20	540	540.0	0.00	0.0	35124	35124	2.8×10^4	1.5×10^4	627	84
21	576	541.5	5.99	6.4	14078	36000	4×10^4	2.2×10^4	702	243
22	1074	875.7	18.47	22.6	16502	36000	7.9×10^4	3.6×10^4	1137	3012
23	1287	1080.8	16.02	19.1	34040	36000	8×10^4	3.6×10^4	1293	2186
24	1245	1074.0	13.74	15.9	35850	36000	8×10^4	3.6×10^4	1254	1940
25	1374	1069.9	22.13	28.4	10069	36000	1.4×10^5	5.4×10^4	1374	10069
26	1557	1338.4	14.04	16.3	31984	36000	1.5×10^5	6.2×10^4	1614	8928
27	1560	1391.0	10.83	12.1	16790	36000	1.6×10^5	6.2×10^4	1668	6666
28	1509	1320.5	12.49	14.3	35203	36000	1.6×10^5	6.7×10^4	1566	3589
29	1395	1288.5	7.63	8.3	35459	36000	1.2×10^5	5.5×10^4	1482	1352
30	1951	1713.6	12.17	13.9	35074	36000	1.4×10^5	6.3×10^4	2195	3405
31	1636	1596.7	2.40	2.5	35927	36000	1.5×10^5	6.9×10^4	2659	4639
32	1888	1620.2	14.18	16.5	36000	36000	1.6×10^5	7.6×10^4	1936	3857
33	1921	1684.7	12.30	14.0	35250	36000	1.7×10^5	7.7×10^4	2767	4625
Avg_s	437	435.5	0.44	0.4	890	17929	8.2×10^3	4.1×10^3	591	13
Avg_m	771	756.8	2.22	2.4	11540	24138	2.4×10^4	1.2×10^4	942	104
Avg_l	1533	1337.8	13.03	15.3	29846	36000	1.3×10^5	5.8×10^4	1745	4522
Avg_*	937	861.0	5.56	6.5	14646	26382	5.8×10^4	2.6×10^4	1117	1680

Source: created by author.

Table A.2 – CPLEX results using a time limit of 10 hours on instances of group B.

Id	Obj	LB	Gap	Gap$_L$	t^*	Time	Columns	Rows	x_0	t_0
01	315	315.0	0.00	0.0	2	2	3.7×10^3	2×10^3	92645	0
02	360	360.0	0.00	0.0	2	2	4.6×10^3	2.7×10^3	360	2
03	690	684.0	0.87	0.9	13827	36000	8.5×10^3	4.6×10^3	10711	15
04	417	414.0	0.72	0.7	2910	36000	6.7×10^3	4.1×10^3	4450	7
05	528	528.0	0.00	0.0	681	1297	7.7×10^3	4.7×10^3	2564	13
06	333	333.0	0.00	0.0	1	1	5.3×10^3	2.4×10^3	16026	0
07	387	387.0	0.00	0.0	7	31	1.5×10^4	8.3×10^3	7414	3
08	600	600.0	0.00	0.0	546	2438	9.2×10^3	4.7×10^3	5672	22
09	474	474.0	0.00	0.0	2575	20302	1.1×10^4	5.3×10^3	5525	8
10	282	279.0	1.06	1.1	10	36000	7.9×10^3	3.9×10^3	22777	0
11	426	423.0	0.70	0.7	36	36000	1.1×10^4	6.6×10^3	3438	2
12	654	654.0	0.00	0.0	1587	1587	1.3×10^4	7.8×10^3	672	66
13	645	630.0	2.33	2.4	35435	36000	1.7×10^4	9.6×10^3	12801	52
14	759	759.0	0.00	0.0	716	716	1.6×10^4	9.5×10^3	13777	48
15	672	613.8	8.66	9.5	35749	36000	1.8×10^4	1.1×10^4	7777	99
16	1077	1071.0	0.56	0.6	4088	36000	1.6×10^4	8.2×10^3	1116	254
17	906	885.8	2.23	2.3	35954	36000	3.1×10^4	1.5×10^4	1083	98
18	1089	1089.0	0.00	0.0	362	392	3.4×10^4	1.6×10^4	13155	16
19	783	780.0	0.38	0.4	36000	36000	2.5×10^4	1.4×10^4	14798	16
20	540	540.0	0.00	0.0	5547	5547	2.8×10^4	1.6×10^4	12636	59
21	573	546.0	4.71	4.9	5772	36000	4×10^4	2.3×10^4	13005	106
22	1170	881.7	24.64	32.7	3851	36000	7.9×10^4	3.6×10^4	1170	3851
23	1323	1086.0	17.92	21.8	34549	36000	8×10^4	3.6×10^4	4347	1251
24	1239	1075.1	13.23	15.2	35742	36000	8×10^4	3.6×10^4	2299	1584
25	1296	1066.3	17.72	21.5	35849	36000	1.4×10^5	5.4×10^4	190574	3113
26	1566	1334.2	14.80	17.4	22169	36000	1.5×10^5	6.2×10^4	123085	1598
27	1539	1388.4	9.78	10.8	31858	36000	1.6×10^5	6.2×10^4	1593	3847
28	1581	1300.6	17.73	21.6	34544	36000	1.6×10^5	6.8×10^4	2590	6717
29	1344	1288.8	4.10	4.3	31890	36000	1.2×10^5	6.9×10^4	8340	216
30	1942	1714.7	11.70	13.3	35898	36000	1.4×10^5	7.8×10^4	69906	4559
31	1735	1593.9	8.13	8.9	35804	36000	1.5×10^5	8.4×10^4	79647	4348
32	1951	1631.7	16.36	19.6	31046	36000	1.6×10^5	9.1×10^4	1972	4521
33	1903	1689.2	11.23	12.7	35779	36000	1.7×10^5	9.3×10^4	2020	4085
Avg$_s$	437	436.1	0.31	0.3	1872	15279	8.2×10^3	4.5×10^3	15598	7
Avg$_m$	769	756.9	1.89	2.0	16121	22424	2.4×10^4	1.3×10^4	9082	81
Avg$_l$	1549	1337.6	13.95	16.6	30748	36000	1.3×10^5	6.4×10^4	40628	3307
Avg$_*$	942	861.1	5.75	6.8	16690	24979	5.8×10^4	2.9×10^4	22725	1230

Source: created by author.

Table A.3 – CPLEX results using a time limit of 10 hours on instances of group C.

Id	Obj	LB	Gap	Gap_L	t^*	Time	Columns	Rows	x_0	t_0
01	315	315.0	0.00	0.0	1	1	3.7×10^3	1.7×10^3	624	0
02	360	360.0	0.00	0.0	2	2	4.6×10^3	2.3×10^3	360	2
03	663	657.0	0.90	0.9	22662	36000	8.6×10^3	4.3×10^3	894	118
04	408	405.0	0.74	0.7	20505	36000	6.8×10^3	3.7×10^3	756	12
05	495	480.6	2.91	3.0	15691	36000	7.8×10^3	4.2×10^3	1284	0
06	324	324.0	0.00	0.0	1	1	5.3×10^3	2.4×10^3	1170	0
07	351	351.0	0.00	0.0	709	709	1.5×10^4	8.1×10^3	1434	0
08	576	567.0	1.56	1.6	2759	36000	9.2×10^3	4.3×10^3	681	23
09	441	441.0	0.00	0.0	5376	5376	1.1×10^4	5.2×10^3	507	15
10	282	276.0	2.13	2.2	71	36000	7.9×10^3	3.6×10^3	510	4
11	405	405.0	0.00	0.0	249	4515	1.2×10^4	6.2×10^3	909	0
12	651	630.0	3.23	3.3	34708	36000	1.3×10^4	7.4×10^3	2022	0
13	618	603.0	2.43	2.5	35677	36000	1.7×10^4	8.9×10^3	1371	72
14	810	651.6	19.56	24.3	33290	36000	1.8×10^4	9.9×10^3	1008	286
15	630	612.0	2.86	2.9	6545	36000	1.8×10^4	1×10^4	2388	0
16	1077	1071.0	0.56	0.6	2427	36000	1.6×10^4	7.7×10^3	2025	45
17	1035	780.6	24.58	32.6	34875	36000	3.1×10^4	1.4×10^4	3117	0
18	1167	881.6	24.46	32.4	33834	36000	3.4×10^4	1.6×10^4	3921	0
19	756	736.2	2.62	2.7	33303	36000	2.6×10^4	1.4×10^4	927	63
20	498	405.0	18.67	23.0	34370	36000	2.8×10^4	1.5×10^4	1239	124
21	528	454.5	13.92	16.2	35501	36000	4×10^4	2.2×10^4	657	809
22	1050	807.8	23.07	30.0	14513	36000	7.9×10^4	3.6×10^4	1050	14513
23	1068	836.5	21.68	27.7	17675	36000	8×10^4	3.6×10^4	1092	17667
24	1137	934.9	17.78	21.6	15062	36000	8×10^4	3.6×10^4	1152	15059
25	1266	1062.6	16.07	19.1	35029	36000	1.4×10^5	5.4×10^4	1356	10206
26	1557	1338.4	14.04	16.3	30432	36000	1.5×10^5	6.2×10^4	1614	9059
27	1635	1343.5	17.83	21.7	29924	36000	1.6×10^5	6.2×10^4	1653	15722
28	1494	1239.3	17.05	20.6	4558	36000	1.6×10^5	6.7×10^4	1494	4558
29	1332	991.5	25.57	34.3	36000	36000	1.2×10^5	5.6×10^4	2233	6652
30	1761	1072.8	39.08	64.1	27993	36000	1.4×10^5	6.4×10^4	3706	12198
31	1476	1189.5	19.41	24.1	36000	36000	1.5×10^5	7×10^4	1560	7955
32	2001	1191.0	40.48	68.0	18299	36000	1.6×10^5	7.6×10^4	4419	15864
33	1980	1296.6	34.51	52.7	19083	36000	1.7×10^5	7.7×10^4	3366	16939
Avg_s	420	416.5	0.75	0.8	6184	17328	8.3×10^3	4.2×10^3	829	16
Avg_m	777	682.5	11.29	14.0	28453	36000	2.4×10^4	1.3×10^4	1867	140
Avg_l	1479	1108.7	23.88	33.4	23714	36000	1.3×10^5	5.8×10^4	2057	12199
Avg_*	913	748.8	12.35	16.6	19307	29776	5.8×10^4	2.6×10^4	1590	4484

Source: created by author.

APPENDIX B — RESULTS OF GOAL SOLVER ON HSTP^{+}

Table B.1 – Results of KHE+GOAL using a time limit of 1 hour on instances of group A.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	378.0	5.0	4.76	5.0	0.0 / 378.0	0.00 / 0.00	0 / 378
03	18860.8	2633.4	96.37	2657.4	17.8 / 1060.8	0.84 / 20.74	17 / 1041
04	12475.2	2891.7	96.68	2913.3	12.0 / 475.2	0.00 / 3.42	12 / 471
05	12531.0	2273.3	95.79	2273.3	12.0 / 531.0	0.00 / 0.00	12 / 531
06	333.0	0.0	0.00	0.0	0.0 / 333.0	0.00 / 0.00	0 / 333
07	27482.4	7001.4	98.59	7001.4	27.0 / 482.4	0.00 / 8.05	27 / 468
08	5601.2	833.5	89.34	838.2	5.0 / 601.2	0.00 / 6.57	5 / 597
09	486.0	2.5	3.09	3.2	0.0 / 486.0	0.00 / 4.74	0 / 480
10	283.2	0.4	2.54	2.6	0.0 / 283.2	0.00 / 2.68	0 / 282
11	6444.0	1412.7	93.39	1412.7	6.0 / 444.0	0.00 / 0.00	6 / 444
12	15690.0	2299.1	95.83	2299.1	15.0 / 690.0	0.00 / 4.24	15 / 687
13	10883.4	1579.5	94.21	1627.5	10.2 / 683.4	0.45 / 13.81	10 / 684
14	26808.2	3432.0	97.17	3432.0	26.0 / 808.2	0.00 / 8.11	26 / 798
15	738.6	7.0	16.90	20.3	0.0 / 738.6	0.00 / 3.91	0 / 732
16	1074.6	-0.2	0.34	0.3	0.0 / 1074.6	0.00 / 1.34	0 / 1074
17	11080.0	1127.0	92.00	1150.0	10.0 / 1080.0	0.00 / 5.61	10 / 1071
18	33168.8	2945.8	96.72	2945.8	32.0 / 1168.8	0.00 / 38.75	32 / 1128
19	25828.0	3198.6	96.97	3198.6	25.0 / 828.0	0.00 / 0.00	25 / 828
20	13053.4	2317.3	95.86	2317.3	12.4 / 653.4	0.55 / 4.93	12 / 657
21	3666.0	536.5	85.23	577.0	3.0 / 666.0	0.00 / 10.39	3 / 651
22	34854.8	3145.3	97.49	3880.3	33.8 / 1054.8	0.45 / 29.44	33 / 1107
23	42224.6	3180.9	97.44	3806.6	41.0 / 1224.6	1.22 / 35.71	39 / 1269
24	30663.6	2362.9	96.50	2755.2	29.4 / 1263.6	0.89 / 14.29	28 / 1245
25	18334.4	1234.4	94.16	1613.7	17.0 / 1334.4	0.00 / 13.81	17 / 1320
26	20662.0	1227.0	93.52	1443.8	19.0 / 1662.0	0.00 / 9.95	19 / 1647
27	28429.6	1722.4	95.11	1943.8	26.8 / 1629.6	2.05 / 14.60	25 / 1623
28	37130.0	2360.6	96.44	2711.8	35.0 / 2130.0	0.00 / 15.73	35 / 2106
29	2200.4	57.7	41.44	70.8	0.8 / 1400.4	0.45 / 9.34	0 / 1407
30	6687.2	242.8	74.37	290.2	4.8 / 1887.2	0.84 / 71.96	4 / 1921
31	4763.2	191.1	66.48	198.3	3.0 / 1763.2	0.00 / 6.57	3 / 1753
32	2639.4	39.8	38.61	62.9	0.8 / 1839.4	0.84 / 62.14	0 / 1882
33	2288.6	19.1	26.39	35.8	0.4 / 1888.6	0.55 / 19.83	0 / 1879
Avg_s	7746.2	1550.9	53.27	1555.7	7.3 / 491.6	0.08 / 4.20	7 / 487
Avg_m	14199.1	1744.3	77.12	1756.8	13.4 / 839.1	0.10 / 9.11	13 / 831
Avg_l	19239.8	1315.3	76.50	1567.8	17.6 / 1589.8	0.61 / 25.28	16 / 1596
Avg_*	13881.1	1523.8	68.94	1621.0	12.9 / 996.3	0.28 / 13.35	12 / 994

Source: created by author.

Table B.2 – Results of KHE+GOAL using a time limit of 1 hour on instances of group B.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	378.0	5.0	4.76	5.0	0.0 / 378.0	0.00 / 0.00	0 / 378
03	17830.2	2484.1	96.16	2506.8	0.0 / 17830.2	0.00 / 1090.26	0 / 17029
04	5480.0	1214.1	92.45	1223.7	0.0 / 5480.0	0.00 / 0.00	0 / 5480
05	7564.0	1332.6	93.02	1332.6	0.0 / 7564.0	0.00 / 0.00	0 / 7564
06	334.8	0.5	0.54	0.5	0.0 / 334.8	0.00 / 4.02	0 / 333
07	19451.8	4926.3	98.01	4926.3	3.0 / 16451.8	0.00 / 1.64	3 / 16450
08	5602.4	833.7	89.29	833.7	1.0 / 4602.4	0.00 / 4.93	1 / 4597
09	489.6	3.3	3.19	3.3	0.0 / 489.6	0.00 / 10.69	0 / 474
10	285.6	1.3	2.31	2.4	0.0 / 285.6	0.00 / 3.29	0 / 282
11	4438.0	941.8	90.47	949.2	0.0 / 4438.0	0.00 / 0.00	0 / 4438
12	11081.0	1594.3	94.10	1594.3	0.0 / 11081.0	0.00 / 543.62	0 / 10681
13	8465.4	1212.5	92.56	1243.7	2.8 / 5665.4	0.45 / 5.37	2 / 5675
14	22778.2	2901.1	96.67	2901.1	0.2 / 22578.2	0.45 / 1924.48	0 / 21780
15	736.2	9.6	16.63	19.9	0.0 / 736.2	0.00 / 4.55	0 / 729
16	1075.2	-0.2	0.39	0.4	0.0 / 1075.2	0.00 / 2.68	0 / 1074
17	7101.6	683.8	87.53	701.7	0.0 / 7101.6	0.00 / 12.62	0 / 7083
18	27342.4	2410.8	96.02	2410.8	0.2 / 27142.4	0.45 / 4001.89	0 / 28143
19	26205.8	3246.8	97.02	3259.7	2.0 / 24205.8	0.00 / 550.23	2 / 23798
20	9630.0	1683.3	94.39	1683.3	0.0 / 9630.0	0.00 / 0.00	0 / 9630
21	2663.6	364.9	79.50	387.8	0.0 / 2663.6	0.00 / 4.93	0 / 2660
22	34096.8	2814.3	97.41	3767.3	3.0 / 31096.8	0.00 / 13.01	3 / 31077
23	42070.2	3079.9	97.42	3774.0	10.0 / 32070.2	0.71 / 837.15	9 / 31287
24	28257.0	2180.6	96.20	2528.3	6.8 / 21457.0	0.45 / 456.14	6 / 22272
25	20387.2	1473.1	94.77	1812.0	0.0 / 20387.2	0.00 / 16.65	0 / 20359
26	19644.6	1154.4	93.21	1372.4	2.0 / 17644.6	0.00 / 7.47	2 / 17638
27	27028.4	1656.2	94.86	1846.7	5.4 / 21628.4	0.89 / 1.34	5 / 21629
28	21115.6	1235.6	93.84	1523.5	1.0 / 20115.6	0.00 / 3.91	1 / 20112
29	1963.8	46.1	34.37	52.4	0.6 / 1363.8	0.55 / 7.53	0 / 1353
30	6730.4	246.6	74.52	292.5	0.8 / 5930.4	1.10 / 163.31	0 / 5894
31	5159.6	197.4	69.11	223.7	1.4 / 3759.6	0.55 / 15.06	1 / 3753
32	1831.0	-6.6	10.88	12.2	0.0 / 1831.0	0.00 / 24.83	0 / 1789
33	1877.8	-1.3	10.04	11.2	0.0 / 1877.8	0.00 / 25.95	0 / 1852
Avg_s	5653.4	1068.0	52.33	1071.7	0.4 / 5289.8	0.00 / 101.35	0 / 5214
Avg_m	11707.9	1410.7	75.48	1420.3	0.5 / 11187.9	0.13 / 705.04	0 / 11125
Avg_l	17513.5	1173.0	72.22	1434.7	2.6 / 14930.2	0.35 / 131.03	2 / 14917
Avg_*	11800.9	1210.1	66.58	1309.3	1.2 / 10582.7	0.17 / 295.08	1 / 10534

Source: created by author.

Table B.3 – Results of KHE+GOAL using a time limit of 1 hour on instances of group C.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	378.0	5.0	4.76	5.0	0.0 / 378.0	0.00 / 0.00	0 / 378
03	3152.6	375.5	79.16	379.8	2.0 / 1152.6	0.00 / 5.77	2 / 1146
04	1527.4	274.4	73.48	277.1	1.0 / 527.4	0.00 / 1.34	1 / 525
05	1618.6	227.0	70.31	236.8	1.0 / 618.6	0.00 / 2.51	1 / 615
06	325.8	0.6	0.55	0.6	0.0 / 325.8	0.00 / 4.02	0 / 324
07	6594.0	1778.6	94.68	1778.6	6.0 / 594.0	0.00 / 0.00	6 / 594
08	3578.4	521.2	84.15	531.1	3.0 / 578.4	0.00 / 3.29	3 / 576
09	484.8	9.9	9.03	9.9	0.0 / 484.8	0.00 / 1.64	0 / 483
10	284.4	0.9	2.95	3.0	0.0 / 284.4	0.00 / 3.29	0 / 282
11	459.0	13.3	11.76	13.3	0.0 / 459.0	0.00 / 6.00	0 / 453
12	741.6	13.9	15.05	17.7	0.0 / 741.6	0.00 / 1.34	0 / 741
13	739.2	19.6	18.43	22.6	0.0 / 739.2	0.00 / 4.02	0 / 735
14	839.4	3.6	22.37	28.8	0.0 / 839.4	0.00 / 3.91	0 / 837
15	717.0	13.8	14.64	17.2	0.0 / 717.0	0.00 / 5.61	0 / 711
16	1077.0	0.0	0.56	0.6	0.0 / 1077.0	0.00 / 3.67	0 / 1074
17	1131.6	9.3	31.02	45.0	0.0 / 1131.6	0.00 / 5.77	0 / 1122
18	1360.2	16.6	35.19	54.3	0.0 / 1360.2	0.00 / 6.91	0 / 1353
19	5849.6	673.8	87.41	694.6	5.0 / 849.6	0.00 / 10.26	5 / 840
20	675.0	35.5	40.00	66.7	0.0 / 675.0	0.00 / 6.36	0 / 666
21	648.0	22.7	29.86	42.6	0.0 / 648.0	0.00 / 0.00	0 / 648
22	31033.2	2855.5	97.40	3741.8	30.0 / 1033.2	0.00 / 6.22	30 / 1026
23	31079.4	2810.1	97.31	3615.4	30.0 / 1079.4	0.00 / 15.65	30 / 1068
24	21090.8	1755.0	95.57	2156.0	20.0 / 1090.8	0.00 / 7.82	20 / 1083
25	16338.0	1190.5	93.50	1437.6	15.0 / 1338.0	0.00 / 7.04	15 / 1326
26	20651.2	1226.3	93.52	1443.0	19.0 / 1651.2	0.00 / 5.45	19 / 1644
27	21594.2	1220.7	93.78	1507.3	20.0 / 1594.2	0.00 / 6.91	20 / 1587
28	29088.6	1847.0	95.74	2247.2	27.0 / 2088.6	0.00 / 6.50	27 / 2079
29	1179.0	-13.0	15.91	18.9	0.0 / 1179.0	0.00 / 18.37	0 / 1152
30	5317.6	202.0	79.83	395.7	4.0 / 1317.6	0.00 / 18.66	4 / 1305
31	3454.4	134.0	65.57	190.4	2.0 / 1454.4	0.00 / 10.26	2 / 1440
32	1508.4	-32.7	21.04	26.6	0.0 / 1508.4	0.00 / 20.72	0 / 1476
33	1611.0	-22.9	19.52	24.2	0.0 / 1611.0	0.00 / 19.09	0 / 1584
Avg_s	1703.3	292.0	39.66	294.6	1.2 / 521.5	0.00 / 2.53	1 / 519
Avg_m	1377.9	80.9	29.45	99.0	0.5 / 877.9	0.00 / 4.79	0 / 872
Avg_l	15328.8	1097.7	72.39	1400.3	13.9 / 1412.2	0.00 / 11.89	13 / 1397
Avg_*	6559.4	521.0	48.47	637.4	5.6 / 953.3	0.00 / 6.62	5 / 945

Source: created by author.

Table B.4 – Results of CPX0+GOAL using a time limit of 1 hour on instances of group A.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	387.6	7.7	7.12	7.7	0.0 / 387.6	0.00 / 10.48	0 / 378
03	921.6	33.6	25.78	34.7	0.0 / 921.6	0.00 / 2.51	0 / 918
04	475.8	14.1	12.99	14.9	0.0 / 475.8	0.00 / 3.42	0 / 471
05	564.6	6.9	6.48	6.9	0.0 / 564.6	0.00 / 7.47	0 / 558
06	333.0	0.0	0.00	0.0	0.0 / 333.0	0.00 / 0.00	0 / 333
07	424.8	9.8	8.90	9.8	0.0 / 424.8	0.00 / 6.57	0 / 414
08	649.2	8.2	8.04	8.7	0.0 / 649.2	0.00 / 14.17	0 / 633
09	501.6	5.8	6.10	6.5	0.0 / 501.6	0.00 / 8.85	0 / 495
10	285.6	1.3	3.36	3.5	0.0 / 285.6	0.00 / 3.29	0 / 282
11	436.8	2.5	2.47	2.5	0.0 / 436.8	0.00 / 5.02	0 / 429
12	672.0	2.8	2.68	2.8	0.0 / 672.0	0.00 / 0.00	0 / 672
13	728.4	12.4	13.51	15.6	0.0 / 728.4	0.00 / 8.32	0 / 714
14	790.8	4.2	4.02	4.2	0.0 / 790.8	0.00 / 1.64	0 / 789
15	759.0	10.0	19.13	23.7	0.0 / 759.0	0.00 / 9.25	0 / 744
16	1076.4	-0.1	0.50	0.5	0.0 / 1076.4	0.00 / 5.37	0 / 1074
17	1147.2	27.0	22.74	29.4	0.0 / 1147.2	0.00 / 14.94	0 / 1125
18	1146.0	5.2	4.97	5.2	0.0 / 1146.0	0.00 / 0.00	0 / 1146
19	810.0	3.4	3.33	3.4	0.0 / 810.0	0.00 / 7.65	0 / 801
20	617.4	14.3	12.54	14.3	0.0 / 617.4	0.00 / 6.50	0 / 612
21	671.4	16.6	19.35	24.0	0.0 / 671.4	0.00 / 12.97	0 / 651
22	1155.0	7.5	24.18	31.9	0.0 / 1155.0	0.00 / 12.90	0 / 1143
23	1392.6	8.2	22.39	28.8	0.0 / 1392.6	0.00 / 22.19	0 / 1362
24	1344.6	8.0	20.13	25.2	0.0 / 1344.6	0.00 / 17.67	0 / 1320
25	1259.4	-9.1	15.05	17.7	0.0 / 1259.4	0.00 / 16.76	0 / 1242
26	1540.2	-1.1	13.10	15.1	0.0 / 1540.2	0.00 / 13.35	0 / 1518
27	1564.8	0.3	11.11	12.5	0.0 / 1564.8	0.00 / 9.86	0 / 1554
28	1806.0	19.7	26.88	36.8	0.0 / 1806.0	0.00 / 21.94	0 / 1785
29	1896.8	36.0	32.07	47.2	0.0 / 1896.8	0.00 / 66.48	0 / 1789
30	2822.2	44.7	39.28	64.7	0.0 / 2822.2	0.00 / 20.19	0 / 2800
31	1904.0	16.4	16.14	19.2	0.0 / 1904.0	0.00 / 18.85	0 / 1886
32	2527.6	33.9	35.90	56.0	0.0 / 2527.6	0.00 / 36.27	0 / 2479
33	2018.2	5.1	16.52	19.8	0.0 / 2018.2	0.00 / 29.59	0 / 1978
Avg_s	483.1	8.7	7.88	9.2	0.0 / 483.1	0.00 / 5.62	0 / 476
Avg_m	841.9	9.6	10.28	12.3	0.0 / 841.9	0.00 / 6.66	0 / 832
Avg_l	1769.3	14.1	22.73	31.2	0.0 / 1769.3	0.00 / 23.84	0 / 1738
Avg_*	1059.5	10.9	14.01	18.2	0.0 / 1059.5	0.00 / 12.56	0 / 1043

Source: created by author.

Table B.5 – Results of CPX0+GOAL using a time limit of 1 hour on instances of group B.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	1324.0	320.3	76.21	320.3	0.0 / 1324.0	0.00 / 0.00	0 / 1324
02	406.8	13.0	11.50	13.0	0.0 / 406.8	0.00 / 4.02	0 / 405
03	33162.4	4706.1	97.94	4748.3	0.0 / 33162.4	0.00 / 449.23	0 / 32960
04	5678.8	1261.8	92.71	1271.7	0.0 / 5678.8	0.00 / 446.23	0 / 5471
05	11777.2	2130.5	95.52	2130.5	0.0 / 11777.2	0.00 / 835.43	0 / 10573
06	10360.0	3011.1	96.79	3011.1	0.0 / 10360.0	0.00 / 0.00	0 / 10360
07	13445.8	3374.4	97.12	3374.4	0.0 / 13445.8	0.00 / 4.02	0 / 13444
08	7074.4	1079.1	91.52	1079.1	0.0 / 7074.4	0.00 / 1339.73	0 / 5672
09	5496.2	1059.5	91.38	1059.5	0.0 / 5496.2	0.00 / 5.45	0 / 5489
10	17308.4	6037.7	98.39	6103.7	0.0 / 17308.4	0.00 / 3.91	0 / 17306
11	15239.2	3477.3	97.22	3502.6	0.0 / 15239.2	0.00 / 446.22	0 / 14441
12	17079.2	2511.5	96.17	2511.5	0.0 / 17079.2	0.00 / 1342.88	0 / 15675
13	11929.6	1749.6	94.72	1793.6	0.0 / 11929.6	0.00 / 445.23	0 / 11723
14	23584.8	3007.4	96.78	3007.4	0.0 / 23584.8	0.00 / 1791.63	0 / 20783
15	743.4	10.6	17.43	21.1	0.0 / 743.4	0.00 / 9.10	0 / 732
16	1080.0	0.3	0.83	0.8	0.0 / 1080.0	0.00 / 7.65	0 / 1074
17	7946.0	777.0	88.85	797.0	0.0 / 7946.0	0.00 / 842.14	0 / 7134
18	36356.8	3238.5	97.00	3238.5	0.0 / 36356.8	0.00 / 4606.22	0 / 30155
19	33416.0	4167.7	97.67	4184.1	0.0 / 33416.0	0.00 / 1348.39	0 / 32804
20	16821.6	3015.1	96.79	3015.1	0.0 / 16821.6	0.00 / 1304.30	0 / 15621
21	8179.2	1327.4	93.32	1398.0	0.0 / 8179.2	0.00 / 549.43	0 / 7769
22	131696.8	11156.1	99.33	14837.1	0.0 / 131696.8	0.00 / 556.29	0 / 131083
23	126302.6	9446.7	99.14	11530.4	0.0 / 126302.6	0.00 / 17.67	0 / 126287
24	127271.4	10172.1	99.16	11738.3	0.0 / 127271.4	0.00 / 13.97	0 / 127251
25	151812.0	11613.9	99.30	14137.4	0.0 / 151812.0	0.00 / 9.25	0 / 151800
26	196089.2	12421.7	99.32	14597.2	0.0 / 196089.2	0.00 / 20.41	0 / 196064
27	204656.8	13198.0	99.32	14640.2	0.0 / 204656.8	0.00 / 538.48	0 / 204058
28	173938.2	10901.8	99.25	13273.4	0.0 / 173938.2	0.00 / 894.44	0 / 173337
29	63265.2	4607.2	97.96	4808.7	0.0 / 63265.2	0.00 / 955.76	0 / 62570
30	50363.4	2493.4	96.60	2837.2	0.0 / 50363.4	0.00 / 429.60	0 / 49599
31	70161.4	3943.9	97.73	4301.9	0.0 / 70161.4	0.00 / 1052.69	0 / 69116
32	52915.2	2612.2	96.92	3142.9	0.0 / 52915.2	0.00 / 1221.08	0 / 51251
33	54604.6	2769.4	96.91	3132.5	0.0 / 54604.6	0.00 / 866.64	0 / 53781
Avg_s	11024.8	2406.4	86.03	2419.5	0.0 / 11024.8	0.00 / 321.30	0 / 10676
Avg_m	15713.7	1980.5	77.96	1996.7	0.0 / 15713.7	0.00 / 1224.70	0 / 14347
Avg_l	116923.1	7944.7	98.41	9414.7	0.0 / 116923.1	0.00 / 548.02	0 / 116349
Avg_*	50954.1	4291.3	88.08	4835.1	0.0 / 50954.1	0.00 / 677.50	0 / 50215

Source: created by author.

Table B.6 – Results of CPX0+GOAL using a time limit of 1 hour on instances of group C.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	385.8	7.2	6.69	7.2	0.0 / 385.8	0.00 / 8.64	0 / 378
03	995.4	50.1	34.00	51.5	0.0 / 995.4	0.00 / 7.16	0 / 987
04	499.8	22.5	18.97	23.4	0.0 / 499.8	0.00 / 8.64	0 / 489
05	573.0	15.8	16.13	19.2	0.0 / 573.0	0.00 / 2.12	0 / 570
06	327.6	1.1	1.10	1.1	0.0 / 327.6	0.00 / 4.93	0 / 324
07	468.0	33.3	25.00	33.3	0.0 / 468.0	0.00 / 0.00	0 / 468
08	605.4	5.1	6.34	6.8	0.0 / 605.4	0.00 / 9.34	0 / 594
09	490.2	11.2	10.04	11.2	0.0 / 490.2	0.00 / 8.38	0 / 480
10	285.6	1.3	3.36	3.5	0.0 / 285.6	0.00 / 3.29	0 / 282
11	448.8	10.8	9.76	10.8	0.0 / 448.8	0.00 / 7.22	0 / 441
12	725.4	11.4	13.15	15.1	0.0 / 725.4	0.00 / 15.65	0 / 714
13	752.4	21.7	19.86	24.8	0.0 / 752.4	0.00 / 12.97	0 / 738
14	824.4	1.8	20.96	26.5	0.0 / 824.4	0.00 / 8.85	0 / 816
15	736.8	17.0	16.94	20.4	0.0 / 736.8	0.00 / 11.54	0 / 723
16	1077.0	0.0	0.56	0.6	0.0 / 1077.0	0.00 / 3.00	0 / 1074
17	1134.0	9.6	31.17	45.3	0.0 / 1134.0	0.00 / 4.74	0 / 1128
18	1323.0	13.4	33.36	50.1	0.0 / 1323.0	0.00 / 10.17	0 / 1311
19	881.4	16.6	16.47	19.7	0.0 / 881.4	0.00 / 14.76	0 / 867
20	592.2	18.9	31.61	46.2	0.0 / 592.2	0.00 / 7.53	0 / 585
21	644.4	22.0	29.47	41.8	0.0 / 644.4	0.00 / 3.29	0 / 642
22	1035.0	-1.4	21.95	28.1	0.0 / 1035.0	0.00 / 10.61	0 / 1023
23	1047.0	-2.0	20.11	25.2	0.0 / 1047.0	0.00 / 12.19	0 / 1032
24	1128.0	-0.8	17.12	20.7	0.0 / 1128.0	0.00 / 17.36	0 / 1110
25	1269.0	0.2	16.27	19.4	0.0 / 1269.0	0.00 / 12.55	0 / 1254
26	1537.2	-1.3	12.94	14.9	0.0 / 1537.2	0.00 / 10.31	0 / 1521
27	1539.6	-6.2	12.74	14.6	0.0 / 1539.6	0.00 / 20.39	0 / 1518
28	1750.2	17.1	29.19	41.2	0.0 / 1750.2	0.00 / 30.04	0 / 1713
29	1406.6	5.6	29.51	41.9	0.0 / 1406.6	0.00 / 21.47	0 / 1379
30	1322.4	-33.2	18.87	23.3	0.0 / 1322.4	0.00 / 31.56	0 / 1287
31	1492.2	1.1	20.29	25.5	0.0 / 1492.2	0.00 / 16.10	0 / 1467
32	1733.6	-15.4	31.30	45.6	0.0 / 1733.6	0.00 / 16.35	0 / 1712
33	1636.2	-21.0	20.75	26.2	0.0 / 1636.2	0.00 / 11.73	0 / 1620
Avg_s	492.1	14.9	12.43	15.8	0.0 / 492.1	0.00 / 5.43	0 / 486
Avg_m	869.1	13.2	21.35	29.0	0.0 / 869.1	0.00 / 9.25	0 / 859
Avg_l	1408.1	-4.8	20.92	27.2	0.0 / 1408.1	0.00 / 17.56	0 / 1386
Avg_*	939.4	7.2	18.22	24.0	0.0 / 939.4	0.00 / 11.00	0 / 926

Source: created by author.

APPENDIX C — RESULTS OF SVNS SOLVER ON HSTP$^+$

Table C.1 – Results of KHE+SVNS using a time limit of 1 hour on instances of group A.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	378.0	5.0	4.76	5.0	0.0 / 378.0	0.00 / 0.00	0 / 378
03	19045.8	2660.3	96.41	2684.5	18.0 / 1045.8	0.00 / 21.49	18 / 1014
04	12475.2	2891.7	96.68	2913.3	12.0 / 475.2	0.00 / 1.64	12 / 474
05	12534.6	2274.0	95.79	2274.0	12.0 / 534.6	0.00 / 3.29	12 / 531
06	333.0	0.0	0.00	0.0	0.0 / 333.0	0.00 / 0.00	0 / 333
07	27468.0	6997.7	98.59	6997.7	27.0 / 468.0	0.00 / 0.00	27 / 468
08	5598.8	833.1	89.34	837.8	5.0 / 598.8	0.00 / 2.68	5 / 597
09	479.4	1.1	1.75	1.8	0.0 / 479.4	0.00 / 2.51	0 / 477
10	282.0	0.0	2.13	2.2	0.0 / 282.0	0.00 / 0.00	0 / 282
11	6440.4	1411.8	93.39	1411.8	6.0 / 440.4	0.00 / 3.91	6 / 435
12	15670.8	2296.1	95.83	2296.1	15.0 / 670.8	0.00 / 4.02	15 / 669
13	9903.2	1428.3	93.64	1471.9	9.2 / 703.2	0.45 / 14.79	9 / 705
14	26784.8	3429.0	97.17	3429.0	26.0 / 784.8	0.00 / 14.01	26 / 771
15	731.4	6.0	16.08	19.2	0.0 / 731.4	0.00 / 2.51	0 / 729
16	1074.0	-0.3	0.28	0.3	0.0 / 1074.0	0.00 / 0.00	0 / 1074
17	11064.4	1125.3	91.99	1148.3	10.0 / 1064.4	0.00 / 8.59	10 / 1053
18	33089.0	2938.5	96.71	2938.5	32.0 / 1089.0	0.00 / 21.94	32 / 1065
19	25819.0	3197.4	96.97	3197.4	25.0 / 819.0	0.00 / 6.36	25 / 810
20	12651.6	2242.9	95.73	2242.9	12.0 / 651.6	0.00 / 8.05	12 / 639
21	3647.4	533.2	85.15	573.6	3.0 / 647.4	0.00 / 5.37	3 / 639
22	34263.8	3090.3	97.44	3812.8	33.2 / 1063.8	0.84 / 21.70	32 / 1074
23	42771.8	3223.4	97.47	3857.3	41.6 / 1171.8	1.34 / 23.39	40 / 1185
24	33979.6	2629.3	96.84	3063.9	32.8 / 1179.6	0.45 / 13.48	32 / 1191
25	18327.2	1233.9	94.16	1613.0	17.0 / 1327.2	0.00 / 11.34	17 / 1320
26	20633.2	1225.2	93.51	1441.7	19.0 / 1633.2	0.00 / 12.30	19 / 1614
27	27596.6	1669.0	94.96	1883.9	26.0 / 1596.6	1.73 / 18.30	25 / 1581
28	36310.2	2306.2	96.36	2649.7	34.2 / 2110.2	0.45 / 14.94	34 / 2097
29	2391.0	71.4	46.11	85.6	1.0 / 1391.0	0.00 / 40.77	1 / 1365
30	6401.6	228.1	73.23	273.6	4.6 / 1801.6	0.55 / 53.70	4 / 1846
31	5101.2	211.8	68.70	219.5	3.4 / 1701.2	0.55 / 42.09	3 / 1717
32	3345.6	77.2	51.57	106.5	1.6 / 1745.6	0.55 / 78.20	1 / 1722
33	1840.6	-4.4	8.47	9.3	0.0 / 1840.6	0.00 / 5.77	0 / 1831
Avg_s	7760.7	1552.8	53.11	1557.6	7.3 / 488.0	0.00 / 3.23	7 / 483
Avg_m	14043.6	1719.6	76.95	1731.7	13.2 / 823.6	0.04 / 8.56	13 / 815
Avg_l	19413.5	1330.1	76.57	1584.7	17.9 / 1546.9	0.54 / 28.00	17 / 1545
Avg_*	13902.0	1522.4	68.87	1620.2	12.9 / 974.7	0.21 / 13.85	12 / 970

Source: created by author.

Table C.2 – Results of KHE+SVNS using a time limit of 1 hour on instances of group B.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	378.0	5.0	4.76	5.0	0.0 / 378.0	0.00 / 0.00	0 / 378
03	20821.8	2917.7	96.71	2944.1	1.4 / 19421.8	0.89 / 4968.50	0 / 27008
04	5485.4	1215.4	92.45	1225.0	0.0 / 5485.4	0.00 / 4.93	0 / 5480
05	7565.2	1332.8	93.02	1332.8	0.0 / 7565.2	0.00 / 1.64	0 / 7564
06	333.0	0.0	0.00	0.0	0.0 / 333.0	0.00 / 0.00	0 / 333
07	19450.6	4926.0	98.01	4926.0	3.0 / 16450.6	0.00 / 1.34	3 / 16450
08	5601.2	833.5	89.29	833.5	1.0 / 4601.2	0.00 / 4.02	1 / 4597
09	482.4	1.8	1.74	1.8	0.0 / 482.4	0.00 / 2.51	0 / 480
10	282.0	0.0	1.06	1.1	0.0 / 282.0	0.00 / 0.00	0 / 282
11	4438.0	941.8	90.47	949.2	0.0 / 4438.0	0.00 / 0.00	0 / 4438
12	10883.4	1564.1	93.99	1564.1	0.0 / 10883.4	0.00 / 442.53	0 / 10681
13	7099.0	1000.6	91.13	1026.8	1.2 / 5899.0	0.45 / 441.18	1 / 5699
14	21982.4	2796.2	96.55	2796.2	0.4 / 21582.4	0.55 / 2046.39	0 / 21777
15	736.8	9.6	16.69	20.0	0.0 / 736.8	0.00 / 4.55	0 / 732
16	1074.0	-0.3	0.28	0.3	0.0 / 1074.0	0.00 / 0.00	0 / 1074
17	7091.4	682.7	87.51	700.5	0.0 / 7091.4	0.00 / 11.10	0 / 7083
18	23537.0	2061.3	95.37	2061.3	0.6 / 22937.0	0.55 / 2486.37	0 / 25128
19	26599.8	3297.2	97.07	3310.2	2.2 / 24399.8	0.45 / 1340.08	2 / 23795
20	9024.6	1571.2	94.02	1571.2	0.0 / 9024.6	0.00 / 551.31	0 / 8621
21	2649.8	362.4	79.39	385.3	0.0 / 2649.8	0.00 / 5.45	0 / 2642
22	34071.0	2812.1	97.41	3764.3	2.4 / 31671.0	0.55 / 542.29	2 / 31077
23	42811.4	3135.9	97.46	3842.2	11.6 / 31211.4	0.55 / 34.75	11 / 31236
24	30192.2	2336.8	96.44	2708.4	9.0 / 21192.2	0.00 / 2.68	9 / 21188
25	20345.2	1469.8	94.76	1808.0	0.0 / 20345.2	0.00 / 7.53	0 / 20338
26	19629.0	1153.4	93.20	1371.2	2.0 / 17629.0	0.00 / 3.67	2 / 17626
27	27012.2	1655.2	94.86	1845.5	5.0 / 22012.2	0.00 / 897.12	5 / 21605
28	21096.4	1234.4	93.83	1522.0	1.0 / 20096.4	0.00 / 11.10	1 / 20079
29	1588.6	18.2	18.87	23.3	0.2 / 1388.6	0.45 / 37.60	0 / 1359
30	6938.0	257.3	75.29	304.6	1.2 / 5738.0	0.45 / 49.72	1 / 5737
31	4741.0	173.3	66.38	197.4	1.0 / 3741.0	1.00 / 16.57	0 / 3723
32	1832.2	-6.5	10.94	12.3	0.0 / 1832.2	0.00 / 25.95	0 / 1801
33	1847.2	-3.0	8.55	9.4	0.0 / 1847.2	0.00 / 20.74	0 / 1819
Avg_s	5924.6	1107.2	52.08	1111.3	0.5 / 5433.7	0.08 / 453.00	0 / 6122
Avg_m	11067.8	1334.5	75.20	1343.6	0.4 / 10627.8	0.20 / 732.90	0 / 10723
Avg_l	17675.4	1186.4	70.67	1450.7	2.8 / 14892.0	0.25 / 137.48	2 / 14799
Avg_*	11756.2	1204.9	65.85	1305.1	1.3 / 10447.1	0.18 / 423.08	1 / 10671

Source: created by author.

Table C.3 – Results of KHE+SVNS using a time limit of 1 hour on instances of group C.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	378.0	5.0	4.76	5.0	0.0 / 378.0	0.00 / 0.00	0 / 378
03	3150.2	375.1	79.14	379.5	2.0 / 1150.2	0.00 / 5.85	2 / 1143
04	1522.0	273.0	73.39	275.8	1.0 / 522.0	0.00 / 0.00	1 / 522
05	1617.4	226.7	70.29	236.5	1.0 / 617.4	0.00 / 2.51	1 / 615
06	324.0	0.0	0.00	0.0	0.0 / 324.0	0.00 / 0.00	0 / 324
07	6594.0	1778.6	94.68	1778.6	6.0 / 594.0	0.00 / 0.00	6 / 594
08	3576.0	520.8	84.14	530.7	3.0 / 576.0	0.00 / 0.00	3 / 576
09	466.8	5.9	5.53	5.9	0.0 / 466.8	0.00 / 7.53	0 / 456
10	282.0	0.0	2.13	2.2	0.0 / 282.0	0.00 / 0.00	0 / 282
11	453.0	11.9	10.60	11.9	0.0 / 453.0	0.00 / 0.00	0 / 453
12	741.0	13.8	14.98	17.6	0.0 / 741.0	0.00 / 0.00	0 / 741
13	924.8	49.6	34.80	53.4	0.2 / 724.8	0.45 / 11.34	0 / 720
14	838.8	3.6	22.32	28.7	0.0 / 838.8	0.00 / 1.64	0 / 837
15	718.2	14.0	14.79	17.4	0.0 / 718.2	0.00 / 2.68	0 / 717
16	1074.0	-0.3	0.28	0.3	0.0 / 1074.0	0.00 / 0.00	0 / 1074
17	1117.8	8.0	30.17	43.2	0.0 / 1117.8	0.00 / 7.53	0 / 1107
18	1323.0	13.4	33.36	50.1	0.0 / 1323.0	0.00 / 8.49	0 / 1311
19	5835.8	671.9	87.38	692.7	5.0 / 835.8	0.00 / 8.64	5 / 828
20	662.4	33.0	38.86	63.6	0.0 / 662.4	0.00 / 4.93	0 / 657
21	634.2	20.1	28.33	39.5	0.0 / 634.2	0.00 / 4.55	0 / 630
22	31006.2	2853.0	97.39	3738.4	30.0 / 1006.2	0.00 / 10.31	30 / 996
23	31023.0	2804.8	97.30	3608.7	30.0 / 1023.0	0.00 / 8.22	30 / 1014
24	21095.6	1755.4	95.57	2156.5	20.0 / 1095.6	0.00 / 7.77	20 / 1086
25	16317.6	1188.9	93.49	1435.7	15.0 / 1317.6	0.00 / 8.05	15 / 1305
26	20638.6	1225.5	93.52	1442.1	19.0 / 1638.6	0.00 / 8.85	19 / 1623
27	21571.4	1219.4	93.77	1505.6	20.0 / 1571.4	0.00 / 10.90	20 / 1557
28	33083.2	2114.4	96.25	2569.5	31.0 / 2083.2	0.00 / 8.64	31 / 2070
29	1176.0	-13.3	15.69	18.6	0.0 / 1176.0	0.00 / 7.65	0 / 1170
30	5305.0	201.2	79.78	394.5	4.0 / 1305.0	0.00 / 15.73	4 / 1287
31	3461.0	134.5	65.63	191.0	2.0 / 1461.0	0.00 / 15.15	2 / 1443
32	1497.0	-33.7	20.44	25.7	0.0 / 1497.0	0.00 / 12.19	0 / 1479
33	1616.4	-22.5	19.78	24.7	0.0 / 1616.4	0.00 / 8.05	0 / 1611
Avg_s	1699.7	291.2	39.10	293.8	1.2 / 517.9	0.00 / 1.44	1 / 516
Avg_m	1387.0	82.7	30.53	100.6	0.5 / 867.0	0.04 / 4.98	0 / 862
Avg_l	15649.2	1119.0	72.39	1425.9	14.2 / 1399.2	0.00 / 10.12	14 / 1386
Avg_*	6677.5	529.0	48.60	646.9	5.7 / 944.2	0.01 / 5.67	5 / 937

Source: created by author.

Table C.4 – Results of CPX0+SVNS using a time limit of 1 hour on instances of group A.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	386.4	7.3	6.83	7.3	0.0 / 386.4	0.00 / 4.93	0 / 378
03	924.0	33.9	25.97	35.1	0.0 / 924.0	0.00 / 5.61	0 / 918
04	469.8	12.7	11.88	13.5	0.0 / 469.8	0.00 / 2.68	0 / 468
05	558.6	5.8	5.48	5.8	0.0 / 558.6	0.00 / 6.50	0 / 549
06	333.0	0.0	0.00	0.0	0.0 / 333.0	0.00 / 0.00	0 / 333
07	412.2	6.5	6.11	6.5	0.0 / 412.2	0.00 / 4.02	0 / 405
08	639.0	6.5	6.57	7.0	0.0 / 639.0	0.00 / 7.65	0 / 630
09	489.0	3.2	3.68	3.8	0.0 / 489.0	0.00 / 3.67	0 / 486
10	282.0	0.0	2.13	2.2	0.0 / 282.0	0.00 / 0.00	0 / 282
11	432.6	1.5	1.53	1.5	0.0 / 432.6	0.00 / 2.51	0 / 429
12	668.4	2.2	2.15	2.2	0.0 / 668.4	0.00 / 2.51	0 / 666
13	708.6	9.4	11.09	12.5	0.0 / 708.6	0.00 / 9.10	0 / 693
14	789.0	4.0	3.80	4.0	0.0 / 789.0	0.00 / 0.00	0 / 789
15	749.4	8.6	18.09	22.1	0.0 / 749.4	0.00 / 6.84	0 / 738
16	1074.6	-0.2	0.34	0.3	0.0 / 1074.6	0.00 / 1.34	0 / 1074
17	1110.0	22.9	20.15	25.2	0.0 / 1110.0	0.00 / 3.00	0 / 1107
18	1135.2	4.2	4.07	4.2	0.0 / 1135.2	0.00 / 10.73	0 / 1122
19	803.4	2.6	2.54	2.6	0.0 / 803.4	0.00 / 3.29	0 / 801
20	597.6	10.7	9.64	10.7	0.0 / 597.6	0.00 / 10.04	0 / 585
21	656.4	14.0	17.50	21.2	0.0 / 656.4	0.00 / 3.29	0 / 651
22	1078.8	0.4	18.83	23.2	0.0 / 1078.8	0.00 / 15.39	0 / 1062
23	1355.4	5.3	20.26	25.4	0.0 / 1355.4	0.00 / 4.93	0 / 1350
24	1284.6	3.2	16.40	19.6	0.0 / 1284.6	0.00 / 14.60	0 / 1269
25	1223.4	-12.3	12.55	14.3	0.0 / 1223.4	0.00 / 14.76	0 / 1200
26	1496.4	-4.0	10.56	11.8	0.0 / 1496.4	0.00 / 9.81	0 / 1482
27	1513.8	-3.1	8.11	8.8	0.0 / 1513.8	0.00 / 10.08	0 / 1503
28	1765.2	17.0	25.19	33.7	0.0 / 1765.2	0.00 / 7.22	0 / 1758
29	2680.4	92.1	51.93	108.0	0.0 / 2680.4	0.00 / 1106.95	0 / 1868
30	3352.8	71.9	48.89	95.7	0.0 / 3352.8	0.00 / 713.61	0 / 2688
31	2026.2	23.9	21.20	26.9	0.0 / 2026.2	0.00 / 345.44	0 / 1865
32	2493.4	32.1	35.02	53.9	0.0 / 2493.4	0.00 / 36.02	0 / 2443
33	2138.8	11.3	21.23	27.0	0.0 / 2138.8	0.00 / 425.63	0 / 1918
Avg_s	478.1	7.6	6.87	8.0	0.0 / 478.1	0.00 / 3.42	0 / 473
Avg_m	829.3	7.8	8.94	10.5	0.0 / 829.3	0.00 / 5.01	0 / 822
Avg_l	1867.4	19.8	24.18	37.4	0.0 / 1867.4	0.00 / 225.37	0 / 1700
Avg_*	1089.7	12.1	13.79	19.4	0.0 / 1089.7	0.00 / 84.61	0 / 1025

Source: created by author.

Table C.5 – Results of CPX0+SVNS using a time limit of 1 hour on instances of group B.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	1324.0	320.3	76.21	320.3	0.0 / 1324.0	0.00 / 0.00	0 / 1324
02	405.0	12.5	11.11	12.5	0.0 / 405.0	0.00 / 0.00	0 / 405
03	34181.6	4853.9	98.00	4897.3	0.0 / 34181.6	0.00 / 441.86	0 / 33978
04	5084.8	1119.4	91.86	1128.2	0.0 / 5084.8	0.00 / 546.63	0 / 4483
05	10573.0	1902.5	95.01	1902.5	0.0 / 10573.0	0.00 / 711.36	0 / 9564
06	10360.0	3011.1	96.79	3011.1	0.0 / 10360.0	0.00 / 0.00	0 / 10360
07	12241.0	3063.0	96.84	3063.0	0.0 / 12241.0	0.00 / 1095.44	0 / 11441
08	4469.0	644.8	86.57	644.8	0.0 / 4469.0	0.00 / 2276.41	0 / 1675
09	5488.4	1057.9	91.36	1057.9	0.0 / 5488.4	0.00 / 6.50	0 / 5480
10	17306.6	6037.1	98.39	6103.1	0.0 / 17306.6	0.00 / 3.29	0 / 17303
11	14841.0	3383.8	97.15	3408.5	0.0 / 14841.0	0.00 / 546.38	0 / 14438
12	16279.2	2389.2	95.98	2389.2	0.0 / 16279.2	0.00 / 2075.38	0 / 13678
13	10733.8	1564.2	94.13	1603.8	0.0 / 10733.8	0.00 / 992.59	0 / 9723
14	32792.0	4220.4	97.69	4220.4	0.0 / 32792.0	0.00 / 12342.33	0 / 18780
15	729.0	8.5	15.80	18.8	0.0 / 729.0	0.00 / 7.04	0 / 723
16	1074.6	-0.2	0.34	0.3	0.0 / 1074.6	0.00 / 1.34	0 / 1074
17	5905.8	551.9	85.00	566.7	0.0 / 5905.8	0.00 / 1099.29	0 / 5098
18	51998.2	4674.9	97.91	4674.9	0.0 / 51998.2	0.00 / 4417.11	0 / 47155
19	49042.4	6163.4	98.41	6187.5	0.0 / 49042.4	0.00 / 8091.30	0 / 41834
20	15628.2	2794.1	96.54	2794.1	0.0 / 15628.2	0.00 / 4.02	0 / 15624
21	7758.8	1254.1	92.96	1321.0	0.0 / 7758.8	0.00 / 4.55	0 / 7751
22	132259.6	11204.2	99.33	14900.9	0.0 / 132259.6	0.00 / 1098.83	0 / 131074
23	126541.6	9464.7	99.14	11552.4	0.0 / 126541.6	0.00 / 472.20	0 / 126260
24	127262.4	10171.4	99.16	11737.4	0.0 / 127262.4	0.00 / 7.77	0 / 127251
25	151764.0	11610.2	99.30	14132.9	0.0 / 151764.0	0.00 / 11.42	0 / 151749
26	196056.8	12419.6	99.32	14594.8	0.0 / 196056.8	0.00 / 9.86	0 / 196043
27	204221.4	13169.7	99.32	14608.8	0.0 / 204221.4	0.00 / 480.06	0 / 203974
28	172510.6	10811.5	99.25	13163.6	0.0 / 172510.6	0.00 / 822.25	0 / 171337
29	66692.8	4862.3	98.07	5074.6	0.0 / 66692.8	0.00 / 5829.44	0 / 62476
30	52873.2	2622.6	96.76	2983.5	0.0 / 52873.2	0.00 / 6969.45	0 / 49451
31	74153.6	4174.0	97.85	4552.3	0.0 / 74153.6	0.00 / 12098.00	0 / 65892
32	57797.4	2862.5	97.18	3442.1	0.0 / 57797.4	0.00 / 11079.75	0 / 51121
33	61172.4	3114.5	97.24	3521.3	0.0 / 61172.4	0.00 / 14208.34	0 / 53754
Avg_s	10570.4	2309.7	85.39	2322.7	0.0 / 10570.4	0.00 / 511.62	0 / 10041
Avg_m	19194.2	2362.0	77.48	2377.7	0.0 / 19194.2	0.00 / 2903.50	0 / 16144
Avg_l	118608.8	8040.6	98.49	9522.1	0.0 / 118608.8	0.00 / 4423.95	0 / 115865
Avg_*	52470.4	4409.5	87.76	4957.3	0.0 / 52470.4	0.00 / 2659.10	0 / 50371

Source: created by author.

Table C.6 – Results of CPX0+SVNS using a time limit of 1 hour on instances of group C.

Id	Obj	Gap_B	Gap	Gap_L	Average	Std. Deviation	Best
01	333.0	5.7	5.41	5.7	0.0 / 333.0	0.00 / 0.00	0 / 333
02	379.8	5.5	5.21	5.5	0.0 / 379.8	0.00 / 4.02	0 / 378
03	1004.4	51.5	34.59	52.9	0.0 / 1004.4	0.00 / 10.69	0 / 996
04	498.0	22.1	18.67	23.0	0.0 / 498.0	0.00 / 6.00	0 / 492
05	577.2	16.6	16.74	20.1	0.0 / 577.2	0.00 / 6.22	0 / 570
06	324.0	0.0	0.00	0.0	0.0 / 324.0	0.00 / 0.00	0 / 324
07	468.0	33.3	25.00	33.3	0.0 / 468.0	0.00 / 0.00	0 / 468
08	584.4	1.5	2.98	3.1	0.0 / 584.4	0.00 / 5.77	0 / 576
09	475.8	7.9	7.31	7.9	0.0 / 475.8	0.00 / 6.22	0 / 468
10	282.0	0.0	2.13	2.2	0.0 / 282.0	0.00 / 0.00	0 / 282
11	437.4	8.0	7.41	8.0	0.0 / 437.4	0.00 / 3.91	0 / 435
12	718.2	10.3	12.28	14.0	0.0 / 718.2	0.00 / 4.55	0 / 714
13	741.0	19.9	18.62	22.9	0.0 / 741.0	0.00 / 4.74	0 / 735
14	803.4	-0.8	18.89	23.3	0.0 / 803.4	0.00 / 14.14	0 / 780
15	712.2	13.0	14.07	16.4	0.0 / 712.2	0.00 / 8.64	0 / 702
16	1074.0	-0.3	0.28	0.3	0.0 / 1074.0	0.00 / 0.00	0 / 1074
17	1090.8	5.4	28.44	39.7	0.0 / 1090.8	0.00 / 15.96	0 / 1071
18	1258.8	7.9	29.97	42.8	0.0 / 1258.8	0.00 / 15.09	0 / 1236
19	852.6	12.8	13.65	15.8	0.0 / 852.6	0.00 / 7.47	0 / 843
20	574.2	15.3	29.47	41.8	0.0 / 574.2	0.00 / 7.53	0 / 567
21	628.2	19.0	27.65	38.2	0.0 / 628.2	0.00 / 6.91	0 / 621
22	981.0	-7.0	17.66	21.4	0.0 / 981.0	0.00 / 11.22	0 / 969
23	1009.8	-5.8	17.16	20.7	0.0 / 1009.8	0.00 / 7.82	0 / 1002
24	1082.4	-5.0	13.63	15.8	0.0 / 1082.4	0.00 / 6.84	0 / 1074
25	1221.0	-3.7	12.98	14.9	0.0 / 1221.0	0.00 / 8.22	0 / 1212
26	1494.6	-4.2	10.45	11.7	0.0 / 1494.6	0.00 / 15.06	0 / 1482
27	1482.6	-10.3	9.38	10.4	0.0 / 1482.6	0.00 / 11.30	0 / 1470
28	1690.2	13.1	26.68	36.4	0.0 / 1690.2	0.00 / 7.82	0 / 1683
29	1375.4	3.3	27.91	38.7	0.0 / 1375.4	0.00 / 22.69	0 / 1352
30	1314.0	-34.0	18.36	22.5	0.0 / 1314.0	0.00 / 18.00	0 / 1296
31	1467.6	-0.6	18.95	23.4	0.0 / 1467.6	0.00 / 15.50	0 / 1446
32	1697.6	-17.9	29.84	42.5	0.0 / 1697.6	0.00 / 28.73	0 / 1667
33	1628.4	-21.6	20.38	25.6	0.0 / 1628.4	0.00 / 12.97	0 / 1614
Avg_s	487.6	13.8	11.40	14.7	0.0 / 487.6	0.00 / 3.89	0 / 483
Avg_m	845.3	10.2	19.33	25.5	0.0 / 845.3	0.00 / 8.50	0 / 834
Avg_l	1370.4	-7.8	18.61	23.7	0.0 / 1370.4	0.00 / 13.85	0 / 1355
Avg_*	917.0	4.9	16.43	21.2	0.0 / 917.0	0.00 / 8.91	0 / 907

Source: created by author.

APPENDIX D — RESULTS OF F8 VARIANT ON HSTP$^+$

Table D.1 – Results of F8 variant on instances of group A using a time limit of 1 hour.

Id	x^*	t^* (s)	time (s)	Gap$_B$	Gap	Gap$_L$	x_0	t_0 (s)	imp (%)	n.f. (%)	inf (%)
01	315	0	51	0.00	0.00	0.00	468	0	5	0	94
02	360	5	101	0.00	0.00	0.00	516	1	7	0	92
03	717	1170	3600	3.91	4.60	4.82	990	4	12	0	88
04	417	3531	3600	0.00	0.72	0.72	534	1	1	40	59
05	528	540	3600	0.00	0.00	0.00	615	1	2	25	72
06	333	2	3600	0.00	0.00	0.00	630	0	19	0	81
07	387	3	3600	0.00	0.00	0.00	471	0	10	0	90
08	600	1174	3600	0.00	0.50	0.50	903	0	7	0	93
09	477	1928	3600	0.63	1.26	1.27	600	0	5	15	80
10	282	19	3600	0.00	2.13	2.17	522	0	0	61	39
11	426	11	3600	0.00	0.00	0.00	450	0	1	18	80
12	657	95	3600	0.46	0.46	0.46	693	13	8	16	76
13	660	1425	3600	1.85	4.55	4.76	915	3	8	13	79
14	771	705	3600	1.58	1.56	1.58	798	4	16	1	83
15	636	3384	3600	-8.49	3.49	3.62	1149	5	3	64	33
16	1077	249	3600	0.00	0.56	0.56	2052	0	15	7	78
17	912	2737	3600	1.00	2.81	2.89	1392	2	7	36	57
18	1107	35	3600	1.65	1.63	1.65	1152	5	9	20	71
19	783	360	3600	0.00	0.00	0.00	849	5	9	0	91
20	567	341	3600	5.00	4.76	5.00	741	4	6	38	56
21	579	2728	3600	0.52	6.48	6.93	936	4	6	34	59
22	984	2453	3600	-9.15	11.01	12.37	1485	61	8	9	81
23	1209	3172	3600	-6.45	10.60	11.86	1473	177	10	1	84
24	1212	3405	3600	-2.72	11.39	12.85	1614	33	10	0	89
25	1200	1138	3600	-14.50	10.84	12.16	1854	18	12	1	87
26	1458	2756	3600	-6.79	8.21	8.94	2109	18	7	1	92
27	1494	1913	3600	-4.42	6.89	7.40	2103	15	8	0	91
28	1449	2658	3600	-4.14	8.87	9.73	2397	13	9	0	91
29	1350	3328	3600	-3.33	4.56	4.77	3893	12	11	3	86
30	1855	889	3600	-5.18	7.62	8.25	3766	7	12	0	88
31	1708	1964	3600	4.40	6.51	6.97	2710	11	12	0	88
32	1828	2794	3600	-3.28	11.37	12.82	4076	8	11	0	89
33	1858	1877	3600	-3.39	9.33	10.29	2899	8	12	0	87
Avg$_s$	440	762	2959	0.41	0.84	0.86	609	1	6	15	79
Avg$_m$	775	1206	3600	0.36	2.63	2.74	1067	4	9	23	68
Avg$_l$	1467	2362	3600	-4.91	8.93	9.87	2531	32	10	1	88
Avg$_*$	915	1478	3386	-1.54	4.32	4.71	1447	13	9	12	79

Source: created by author.

Table D.2 – Results of F8 variant on instances of group B using a time limit of 1 hour.

Id	x^*	t^* (s)	time (s)	Gap$_B$	Gap	Gap$_L$	x_0	t_0 (s)	imp (%)	n.f. (%)	inf (%)
01	315	1	50	0.00	0.00	0.00	5486	0	6	0	94
02	360	4	106	0.00	0.00	0.00	7555	0	8	0	92
03	726	1663	3600	5.22	5.79	6.14	40032	0	11	0	89
04	417	788	3600	0.00	0.72	0.72	28549	0	2	48	50
05	528	11	3600	0.00	0.00	0.00	25621	0	2	34	64
06	333	12	3600	0.00	0.00	0.00	12684	0	20	0	80
07	387	63	3600	0.00	0.00	0.00	20474	0	11	1	88
08	600	2704	3600	0.00	0.00	0.00	23915	0	5	18	77
09	483	2759	3600	1.90	1.86	1.90	6639	0	3	18	80
10	282	8	3600	0.00	1.06	1.08	28510	0	0	71	29
11	426	5	3600	0.00	0.70	0.71	25453	0	1	10	89
12	657	196	3600	0.46	0.46	0.46	37729	0	7	14	79
13	2660	1312	3600	312.40	76.32	322.22	34924	0	4	29	67
14	765	38	3600	0.79	0.78	0.79	45807	0	16	0	84
15	642	375	3600	-4.67	4.39	4.59	33221	0	4	48	48
16	1080	519	3600	0.28	0.83	0.84	3052	0	13	5	82
17	927	2197	3600	2.32	4.44	4.65	35473	0	5	50	46
18	1089	604	3600	0.00	0.00	0.00	55224	0	11	13	77
19	1780	359	3600	127.33	56.18	128.21	57861	0	10	2	88
20	561	2108	3600	3.89	3.74	3.89	41762	1	11	45	44
21	582	551	3600	1.57	6.19	6.59	40293	0	9	37	54
22	993	2206	3600	-17.82	11.21	12.63	136788	7	8	4	88
23	2224	3566	3600	68.10	51.17	104.79	131695	9	9	4	87
24	2170	1544	3600	75.14	50.46	101.84	129896	6	13	0	87
25	1224	1729	3600	-5.88	12.88	14.79	165595	2	8	26	66
26	1464	1533	3600	-6.97	8.87	9.73	220066	2	7	3	91
27	5494	2921	3600	256.99	74.73	295.70	213703	3	9	3	88
28	1467	2885	3600	-7.77	11.34	12.79	212201	2	11	3	86
29	1371	2195	3600	2.01	5.99	6.37	76672	5	12	0	88
30	1864	3551	3600	-4.18	8.01	8.71	65312	20	13	0	86
31	1693	2181	3600	-2.48	5.85	6.22	95486	5	10	1	89
32	1780	2818	3600	-9.61	8.33	9.09	77483	3	12	0	88
33	1849	2215	3600	-2.92	8.64	9.46	86554	4	14	0	86
Avg$_s$	442	729	2960	0.65	0.92	0.96	20447	0	6	18	76
Avg$_m$	1074	826	3600	44.44	15.33	47.22	38534	0	9	24	67
Avg$_l$	1966	2445	3600	28.72	21.46	49.34	134287	6	10	4	86
Avg$_*$	1188	1382	3387	24.12	12.76	32.57	67324	2	9	15	77

Source: created by author.

Table D.3 – Results of F8 variant on instances of group C using a time limit of 1 hour.

Id	x^*	t^* (s)	time (s)	Gap_B	Gap	Gap_L	x_0	t_0 (s)	imp (%)	n.f. (%)	inf (%)
01	315	0	51	0.00	0.00	0.00	468	0	5	0	95
02	360	5	100	0.00	0.00	0.00	516	0	7	0	93
03	759	3311	3600	14.48	13.44	15.53	1128	2	5	0	95
04	408	484	3600	0.00	0.74	0.74	753	0	3	46	51
05	492	1989	3600	-0.61	2.32	2.37	831	1	4	70	26
06	324	1	3600	0.00	0.00	0.00	711	0	19	0	81
07	351	57	3600	0.00	0.00	0.00	594	0	8	1	91
08	576	853	3600	0.00	1.56	1.59	1212	0	5	29	66
09	447	2332	3600	1.36	1.34	1.36	945	0	4	57	39
10	282	19	3600	0.00	2.13	2.17	522	0	0	61	39
11	405	101	3600	0.00	0.00	0.00	672	0	1	17	82
12	645	286	3600	-0.93	2.33	2.38	1248	2	3	56	41
13	648	1728	3600	4.85	6.94	7.46	1365	2	5	43	52
14	711	712	3600	-13.92	8.35	9.12	1389	2	5	43	53
15	630	1754	3600	0.00	2.86	2.94	1464	2	3	58	39
16	1077	241	3600	0.00	0.56	0.56	2052	0	13	8	79
17	873	3555	3600	-18.56	10.59	11.84	2013	1	7	60	33
18	1029	3391	3600	-13.41	14.33	16.72	2130	2	9	71	21
19	765	406	3600	1.19	3.76	3.91	1248	2	7	3	90
20	459	2269	3600	-8.50	11.76	13.33	1209	1	9	27	63
21	531	309	3600	0.57	14.41	16.83	1293	2	9	24	68
22	921	1868	3600	-14.01	12.29	14.02	1659	34	10	8	82
23	960	1835	3600	-11.25	12.86	14.76	1779	38	9	5	85
24	1047	1642	3600	-8.60	10.71	11.99	1842	27	9	9	81
25	1203	1916	3600	-5.24	11.67	13.22	1761	18	7	5	87
26	1458	2740	3600	-6.79	8.21	8.94	2109	18	7	1	92
27	1455	3430	3600	-12.37	7.66	8.30	2082	18	6	0	93
28	1380	990	3600	-8.26	10.20	11.35	2412	15	8	0	91
29	1143	3592	3600	-16.54	13.26	15.28	3444	2	16	4	80
30	1233	3396	3600	-42.82	12.99	14.93	4106	2	15	2	84
31	1386	2544	3600	-6.49	14.18	16.52	3411	4	17	1	82
32	1422	3398	3600	-40.72	16.24	19.40	4708	2	18	1	81
33	1602	2560	3600	-23.60	19.06	23.55	3762	3	13	2	85
Avg_s	429	832	2959	1.38	1.96	2.16	759	0	6	25	69
Avg_m	737	1465	3600	-4.87	7.59	8.51	1541	2	7	39	54
Avg_l	1268	2493	3600	-16.39	12.44	14.36	2756	15	11	3	85
Avg_*	827	1628	3386	-6.97	7.48	8.52	1722	6	8	22	70

Source: created by author.

APPENDIX E — RESULTS OF $\overline{\text{F8}}$ VARIANT ON HSTP^{+}

Table E.1 – Average results of $\overline{\text{F8}}$ variant on instances of group A using a time limit of 1 hour.

Id	x^*	sd	best	worst	t^* (s)	time (s)	Gap$_B$	Gap	Gap$_L$	x_0	t_0 (s)	imp (%)	n.f. (%)	inf (%)
01	315.0	0.0	315	315	1	52	0.00	0.00	0.00	468	0	6	0	94
02	360.0	0.0	360	360	4	103	0.00	0.00	0.00	516	0	7	0	93
03	748.2	11.9	732	765	1637	3600	8.43	8.58	9.39	990	4	10	0	90
04	417.0	0.0	417	417	984	3600	0.00	0.72	0.72	536	1	2	37	60
05	528.0	0.0	528	528	702	3600	0.00	0.00	0.00	615	1	2	21	77
06	333.0	0.0	333	333	1	3600	0.00	0.00	0.00	630	0	19	0	81
07	387.0	0.0	387	387	2	3600	0.00	0.00	0.00	471	0	10	0	90
08	600.0	0.0	600	600	1187	3600	0.00	0.50	0.50	903	0	7	4	90
09	478.8	1.6	477	480	1564	3600	1.01	1.63	1.66	600	0	3	17	80
10	282.0	0.0	282	282	17	3600	0.00	2.13	2.17	522	0	0	71	29
11	426.0	0.0	426	426	7	3600	0.00	0.00	0.00	450	0	1	25	74
12	656.4	1.3	654	657	687	3600	0.37	0.37	0.37	693	13	7	9	84
13	655.2	3.4	651	660	994	3600	1.11	3.85	4.00	915	3	7	16	77
14	762.0	5.2	759	771	1214	3600	0.40	0.39	0.40	799	6	16	0	84
15	640.8	5.0	636	648	993	3600	-7.68	4.21	4.40	1149	5	4	47	48
16	1076.4	2.5	1074	1080	1540	3600	-0.06	0.50	0.50	2052	0	13	7	80
17	918.0	10.6	903	930	2041	3600	1.66	3.45	3.57	1418	2	7	50	43
18	1094.4	8.0	1089	1107	884	3600	0.50	0.49	0.50	1152	5	10	21	70
19	783.0	0.0	783	783	865	3600	0.00	0.00	0.00	849	6	10	1	89
20	554.4	4.9	549	558	1635	3600	2.67	2.60	2.67	741	4	10	16	74
21	576.6	8.0	567	585	1330	3600	0.10	6.09	6.48	936	4	8	35	57
22	1002.0	5.2	999	1011	2198	3600	-7.19	12.61	14.42	1485	62	7	10	81
23	1219.8	12.7	1206	1230	3165	3600	-5.51	11.39	12.86	1473	181	8	2	85
24	1203.0	24.6	1170	1239	2867	3600	-3.49	10.73	12.01	1614	39	11	2	87
25	1189.8	13.0	1179	1212	2005	3600	-15.48	10.08	11.21	1854	18	8	10	82
26	1483.2	21.7	1458	1518	2548	3600	-4.98	9.77	10.82	2109	19	7	2	91
27	1500.6	6.5	1491	1506	2229	3600	-3.96	7.30	7.88	2103	15	7	0	92
28	1468.2	15.7	1443	1485	2288	3600	-2.78	10.06	11.18	2382	14	8	0	91
29	1345.2	7.8	1335	1356	3119	3600	-3.70	4.21	4.40	3614	6	11	0	89
30	1846.0	9.2	1837	1858	1695	3600	-5.69	7.17	7.72	3766	7	13	0	87
31	1711.0	17.4	1693	1738	1433	3600	4.58	6.68	7.16	2699	12	12	0	87
32	1807.6	27.1	1771	1846	2985	3600	-4.45	10.37	11.57	4076	8	10	0	90
33	1847.2	36.7	1801	1900	1870	3600	-4.00	8.80	9.65	2899	9	14	0	85
Avg$_s$	443.2	1.2	442	445	555	2960	0.86	1.23	1.31	609	1	6	16	78
Avg$_m$	771.7	4.9	766	778	1218	3600	-0.09	2.19	2.29	1070	5	9	20	71
Avg$_l$	1468.6	16.5	1449	1492	2367	3600	-4.72	9.10	10.07	2506	32	10	2	87
Avg$_*$	915.6	7.9	906	926	1415	3387	-1.46	4.38	4.79	1438	13	8	12	79

Source: created by author.

Table E.2 – Average results of $\overline{\text{F8}}$ variant on instances of group B using a time limit of 1 hour.

Id	x^*	sd	best	worst	t^* (s)	time (s)	Gap$_B$	Gap	Gap$_L$	x_0	t_0 (s)	imp (%)	n.f. (%)	inf (%)
01	315.0	0.0	315	315	1	49	0.00	0.00	0.00	6683	0	6	0	94
02	360.0	0.0	360	360	5	105	0.00	0.00	0.00	7555	0	7	0	93
03	734.4	6.5	726	741	1943	3600	6.43	6.86	7.37	40032	0	12	0	88
04	417.6	1.3	417	420	517	3600	0.14	0.86	0.87	28549	0	2	45	54
05	528.6	1.3	528	531	821	3600	0.11	0.11	0.11	25621	0	3	28	69
06	333.0	0.0	333	333	10	3600	0.00	0.00	0.00	12684	0	20	0	80
07	387.0	0.0	387	387	44	3600	0.00	0.00	0.00	20474	0	11	1	89
08	602.4	2.5	600	606	1274	3600	0.40	0.40	0.40	23915	0	5	12	83
09	481.2	4.0	477	486	1952	3600	1.52	1.50	1.52	6639	0	3	28	69
10	282.0	0.0	282	282	25	3600	0.00	1.06	1.08	26518	0	0	72	28
11	426.0	0.0	426	426	44	3600	0.00	0.70	0.71	25453	0	1	20	79
12	656.4	1.3	654	657	56	3600	0.37	0.37	0.37	37729	0	7	10	83
13	2658.8	6.2	2651	2666	1887	3600	312.22	76.31	322.03	34924	0	4	31	65
14	763.2	3.4	759	768	1134	3600	0.55	0.55	0.55	45807	0	16	1	82
15	637.8	2.7	636	642	2679	3600	-5.36	3.76	3.91	32417	0	5	58	37
16	1075.8	1.6	1074	1077	1019	3600	-0.11	0.45	0.45	3052	0	12	18	70
17	917.4	4.9	912	924	2247	3600	1.26	3.44	3.57	31268	0	7	48	45
18	1099.8	7.5	1089	1110	302	3600	0.99	0.98	0.99	55224	0	10	20	70
19	784.2	2.7	783	789	1635	3600	0.15	0.54	0.54	57861	0	10	1	90
20	552.0	7.6	540	558	1635	3600	2.22	2.17	2.22	41762	1	10	26	64
21	577.2	3.4	573	582	2407	3600	0.73	5.41	5.71	40293	1	7	51	42
22	1007.4	10.7	990	1017	2032	3600	-16.14	12.48	14.26	136171	7	9	13	77
23	2397.0	445.6	2185	3194	3148	3600	81.18	54.69	120.72	131695	10	10	3	86
24	1600.0	539.8	1176	2194	2204	3600	29.14	32.81	48.83	129896	6	13	6	81
25	1191.6	11.1	1179	1206	2701	3600	-8.76	10.52	11.75	165595	2	10	16	73
26	1465.2	14.3	1449	1485	2786	3600	-6.88	8.94	9.82	220066	2	8	2	90
27	5483.8	17.6	5461	5503	2238	3600	256.32	74.68	294.96	213703	3	8	4	88
28	1451.4	25.5	1416	1476	2398	3600	-8.93	10.39	11.59	200191	2	9	3	88
29	1365.0	8.7	1356	1377	2335	3600	1.56	5.58	5.91	69484	5	12	0	87
30	1853.8	8.6	1843	1864	1157	3600	-4.76	7.50	8.11	65312	20	13	0	87
31	1727.2	16.2	1708	1747	1601	3600	-0.45	7.72	8.36	95486	5	13	0	87
32	1793.8	12.1	1783	1813	2098	3600	-8.76	9.03	9.93	77483	3	12	0	88
33	1829.8	17.7	1810	1855	2474	3600	-4.00	7.68	8.32	86554	4	13	0	87
Avg$_s$	442.5	1.4	441	444	603	2959	0.78	1.05	1.10	20374	0	6	19	75
Avg$_m$	972.3	4.2	967	977	1500	3600	31.30	9.40	34.03	38033	0	9	26	65
Avg$_l$	1930.5	94.0	1863	2061	2264	3600	25.79	20.17	46.05	132636	6	11	4	85
Avg$_*$	1144.1	35.9	1118	1194	1479	3387	19.13	10.53	27.42	66548	2	9	16	76

Source: created by author.

Table E.3 – Average results of $\overline{\text{F8}}$ variant on instances of group C using a time limit of 1 hour.

Id	x^*	sd	best	worst	t^* (s)	time (s)	Gap_B	Gap	Gap_L	x_0	t_0 (s)	imp (%)	n.f. (%)	inf (%)
01	315.0	0.0	315	315	1	52	0.00	0.00	0.00	468	0	6	0	94
02	360.0	0.0	360	360	4	101	0.00	0.00	0.00	516	0	7	0	93
03	729.6	24.5	702	759	1988	3600	10.05	9.95	11.05	1128	2	5	0	95
04	409.2	1.6	408	411	702	3600	0.29	1.03	1.04	753	0	2	61	37
05	489.0	0.0	489	489	1455	3600	-1.23	1.72	1.75	826	1	3	60	37
06	324.0	0.0	324	324	5	3600	0.00	0.00	0.00	711	0	20	0	80
07	351.0	0.0	351	351	47	3600	0.00	0.00	0.00	594	0	8	0	92
08	579.0	4.2	576	585	848	3600	0.52	2.07	2.12	1212	0	4	27	69
09	443.4	1.3	441	444	1977	3600	0.54	0.54	0.54	945	0	6	33	62
10	282.0	0.0	282	282	18	3600	0.00	2.13	2.17	523	0	0	70	29
11	405.0	0.0	405	405	242	3600	0.00	0.00	0.00	672	0	1	19	80
12	645.0	0.0	645	645	953	3600	-0.93	2.33	2.38	1248	2	3	47	50
13	654.6	5.4	648	660	1852	3600	5.92	7.88	8.56	1365	2	4	44	52
14	704.4	4.4	699	711	1767	3600	-14.99	7.50	8.10	1389	2	4	51	45
15	622.2	3.4	618	627	1920	3600	-1.25	1.64	1.67	1464	3	4	55	41
16	1076.4	2.5	1074	1080	1504	3600	-0.06	0.50	0.50	2052	0	13	7	80
17	881.4	12.6	864	894	2881	3600	-17.43	11.44	12.92	1981	1	7	63	29
18	996.0	25.1	963	1017	2929	3600	-17.17	11.49	12.98	2130	2	11	57	32
19	759.0	10.6	747	774	1820	3600	0.40	3.00	3.10	1248	2	6	11	83
20	451.8	4.0	450	459	1538	3600	-10.23	10.36	11.56	1209	1	10	28	62
21	499.2	4.5	495	504	2119	3600	-5.77	8.95	9.84	1293	2	11	30	59
22	922.8	14.5	900	936	2025	3600	-13.78	12.46	14.24	1659	36	9	10	80
23	966.0	15.7	948	990	2971	3600	-10.56	13.41	15.48	1776	38	9	9	81
24	1035.6	22.2	1020	1074	2676	3600	-9.79	9.72	10.77	1842	30	7	6	86
25	1197.0	7.0	1188	1203	1860	3600	-5.76	11.23	12.65	1761	19	9	9	82
26	1483.2	21.7	1458	1518	2535	3600	-4.98	9.77	10.82	2109	19	7	2	91
27	1465.2	11.5	1449	1479	2315	3600	-11.59	8.31	9.06	2082	18	8	0	92
28	1404.0	20.1	1377	1431	2485	3600	-6.41	11.73	13.29	2354	16	9	0	90
29	1126.8	11.9	1107	1137	2682	3600	-18.21	12.01	13.65	3353	2	14	4	81
30	1248.0	4.2	1242	1251	2707	3600	-41.11	14.04	16.33	4106	2	13	3	84
31	1413.0	30.4	1377	1443	2934	3600	-4.46	15.82	18.79	3411	3	15	0	84
32	1470.6	17.5	1449	1494	3420	3600	-36.07	19.01	23.48	4708	2	16	1	83
33	1607.4	27.4	1575	1647	3337	3600	-23.18	19.33	23.97	3762	3	13	1	86
Avg_s	426.1	2.9	423	430	663	2959	0.93	1.59	1.70	758	0	6	25	70
Avg_m	729.0	7.3	720	737	1928	3600	-6.15	6.51	7.16	1537	2	7	39	53
Avg_l	1278.3	17.0	1258	1300	2662	3600	-15.49	13.07	15.21	2743	16	11	4	85
Avg_*	827.8	9.4	817	839	1773	3386	-7.19	7.25	8.27	1716	6	8	21	70

Source: created by author.