

UNIVERSIDADE FEDERAL DE OURO PRETO

Métodos de Busca Heurística para Problemas de Programação de Horários Modelados em XHSTT

George Henrique Godim da Fonseca
Universidade Federal de Ouro Preto

Orientador: Haroldo Gambini Santos

Coorientador: Túlio Ângelo Machado Toffolo

Dissertação submetida ao Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto para obtenção do título de Mestre em Ciência da Computação

Ouro Preto, Fevereiro de 2013

F676m Fonseca, George Henrique Godim da.
Métodos de busca heurística para problemas de programação de horários modelados em XHSTT [manuscrito] / George Henrique Godim da Fonseca. – 2013.
xiv, 69 f.: il.; tabs.

Orientador: Prof. Dr. Haroldo Gambini Santos.
Coorientador: Prof. MSc. Túlio Ângelo Machado Toffolo.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-graduação em Ciência da Computação.
Área de concentração: Otimização e Inteligência Computacional.

1. Programação heurística - Teses. 2. Otimização combinatória - Teses.
3. Problema de geração de horários escolares (Timetabling problem) - Teses.
I. Universidade Federal de Ouro Preto. II. Título.

CDU: 004.023

Catálogo: sisbin@sisbin.ufop.br

Métodos de Busca Heurística para Problemas de Programação de Horários Modelados em XHSTT

George Henrique Godim da Fonseca
Universidade Federal de Ouro Preto

Orientador: Haroldo Gambini Santos

Coorientador: Túlio Ângelo Machado Toffolo





MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas – ICEB
Programa de Pós-Graduação em Ciência da Computação



Ata da Defesa Pública de Dissertação de Mestrado

Aos 04 dias do mês de fevereiro de 2013, às 14 horas na Sala de Seminários do DECOM no Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Haroldo Gambini Santos (presidente e orientador)**, **Prof. Ms. Túlio Ângelo Machado Toffolo**, **Prof. Dr. Luiz Satoru Ochi** e **Prof. Dr. Sebastián Alberto Urrutia**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de argüirem o mestrando **George Henrique Godim da Fonseca**, com o título “**Métodos de Busca Heurística para Problemas de Programação de Horários Modelados em XHSTT**”. Aberta a sessão pelo presidente, coube ao candidato, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

☒ Aprovada sem recomendações

() Reprovada

() Aprovada com recomendações: _____

Banca Examinadora:

Prof. Dr. Haroldo Gambini Santos

Prof. Ms. Túlio Ângelo Machado Toffolo

Prof. Dr. Luiz Satoru Ochi

Prof. Dr. Sebastián Alberto Urrutia

Prof. Dr. Haroldo Gambini Santos

Coordenador do Programa de Pós-Graduação em Ciência da Computação
DECOM/ICEB/UFOP

Ouro Preto, 04 de fevereiro de 2013.

Dedico este trabalho aos meus pais Rogério e Petrina, ao meu irmão Luis e à minha namorada Kathlin, pessoas de suma importância na minha vida.

Métodos de Busca Heurística para Problemas de Programação de Horários Modelados em XHSTT

Resumo

O Problema da Programação de Horários Escolares é alvo de diversas pesquisas em Pesquisa Operacional e Inteligência Artificial devido a sua dificuldade de resolução e importância prática. Uma solução para esse problema consiste basicamente na alocação de aulas a horários e na alocação dos recursos para essas aulas. Essa alocação deve atender a várias restrições especificadas *a priori*. O presente trabalho considera a solução do problema proposto pela *Third International Timetabling Competition* (ITC2011), a qual inclui um amplo conjunto de instâncias originadas de diversas instituições educacionais ao redor do mundo. O presente trabalho propõe diversas técnicas de busca local para solucionar o problema. O formato de especificação de instâncias considerado foi o XHSTT, permitindo que qualquer instância especificada no referido formato possa ser manipulada pelos algoritmos propostos. Uma característica estrutural importante da nossa abordagem é o uso da plataforma KHE para gerar soluções iniciais, combinada com uma abordagem de busca multi-vizinhança. Os resultados obtidos incluem o desenvolvimento do algoritmo vencedor da competição. Fomos ainda capazes de encontrar soluções factíveis para treze de dezoito instâncias e melhorar quinze de dezesseis melhores soluções conhecidas.

Heuristic Search Methods to Timetabling Problems Modeled in XHSTT

Abstract

The High School Timetabling Problem remains subject of many research in Operational Research and Artificial Intelligence fields because of its hardness to solve and practical importance. A solution for this problem basically consists in the schedule of lessons to timeslots and the assignment of resources for these lessons. This allocation should satisfy many a priori specified constraints. This work considers the solution of the problem of the Third International Timetabling Competition (ITC2011), which includes a diverse set of instances from many educational institutions around the world. This work presents many local search methods to solve the problem. The format for specifying instances considered was XHSTT, allowing any instance specified in that format to be manipulated by the proposed algorithms. One important structural feature of our approach is the use of the KHE engine to generate initial solutions combined with a multi-neighborhood search approach. The achieved results include the development of the algorithm winner of competition. Moreover, we found feasible solutions to thirteen out of eighteen instances and we improved the best known solution to fifteen out of sixteen instances.

Declaração

Esta dissertação é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para outra qualificação nesta nem em outra universidade.

George Henrique Godim da Fonseca

Agradecimentos

Primeiramente, gostaria de agradecer a *Deus*. Muito obrigado Senhor por estar comigo nessa caminhada, pela força concedida, pelo ânimo nos momentos difíceis e pelas vitórias que obtive. Nada seria possível ou faria sentido sem a Tua glória.

Agradeço ao meu orientador *Haroldo Gambini Santos* pela oportunidade concedida, pela impecável orientação e pelo apoio no desenvolvimento do trabalho. Agradeço ainda aos colegas do DECOM *Túlio Toffolo* e *Samuel Brito* pelas valiosas contribuições para o presente trabalho e ao *Dr. Jeffrey Kingston*, da universidade de Sydney, por produzir a engine KHE e colaborar com as correções necessárias a esse trabalho.

Sou grato também aos membros da banca examinadora *Luiz Satoru* e *Sebastián Urutia* pela cuidadosa revisão do presente trabalho e aos professores *Fabrizio Benevenuto*, *Joubert Lima*, *Gustavo Peixoto* e, novamente, *Haroldo Santos* pelo conhecimento transmitido através das disciplinas ministradas.

Por fim, agradeço à minha família, em especial minha namorada, *Kathlin Dias*, e minha mãe, *Petrina Godim*. Duas pessoas que sempre acreditaram em mim, tiveram paciência comigo durante essa caminhada e me deram forças para seguir em frente.

Sumário

Lista de Figuras	xix
Lista de Tabelas	xxi
Lista de Algoritmos	xxiii
Nomenclatura	1
1 Introdução	3
1.1 Objetivo	4
1.2 Organização do Trabalho	4
2 Problema da Programação de Horários Escolares	7
2.1 Tempos	7
2.2 Recursos	7
2.3 Eventos	8
2.4 Restrições	9
2.4.1 Restrições Básicas de Agendamento	9
2.4.2 Restrições para os Eventos	9
2.4.3 Restrições para os Recursos	10
2.4.4 Função Objetivo	11

2.5	Espaço de Busca	11
2.6	Formato XHSTT	13
2.7	ITC2011	14
2.7.1	Participantes	16
3	Técnicas Heurísticas Abordadas	19
3.1	Método Construtivo	19
3.2	Estruturas de Vizinhaça	20
3.2.1	Event Swap (ES)	20
3.2.2	Event Move (EM)	20
3.2.3	Event Block Move (EBM)	21
3.2.4	Resource Swap (RS)	22
3.2.5	Resource Move (RM)	22
3.2.6	Kempe Chain (KC)	23
3.2.7	Permut Resources (PR)	24
3.2.8	Geração de Vizinhos	25
3.3	Busca Local	26
3.3.1	Método Não Ascendente Randômico	26
3.3.2	<i>Iterated Local Search</i>	26
3.3.3	<i>Simulated Annealing</i>	27
3.3.4	<i>Variable Neighborhood Search</i>	29
3.3.5	<i>Late Acceptance Hill Climbing</i>	31
3.4	Resolvedor Submetido à ITC2011	33
4	Experimentos Computacionais	35
4.1	Caracterização das Instâncias	35

4.1.1	Instâncias Públicas	35
4.1.2	Instâncias Ocultas	38
4.2	Resultados Obtidos	40
4.3	Resultados Oficiais da ITC2011	49
4.4	Discussão dos Resultados	57
5	Considerações Finais	61
5.1	Trabalhos Futuros	62
A	Apêndices	63
A.1	Publicações	63
	Referências Bibliográficas	65

Lista de Figuras

2.1	Exemplo de alto nível de instância especificada no formato XHSTT . . .	15
3.1	Exemplo do movimento Event Swap	20
3.2	Exemplo do movimento Event Move	21
3.3	Exemplo do movimento Event Block Move	21
3.4	Exemplo do movimento Resource Swap	22
3.5	Exemplo do movimento Resource Move	23
3.6	Exemplo do movimento Kempe Chain	24
3.7	Exemplo do movimento Permut Resources	25
3.8	Composição do resolvidor da equipe GOAL	34
4.1	Adaptação do resolvidor para a primeira fase da ITC2011	50

Lista de Tabelas

2.1	Participantes da ITC2011	17
4.1	Características das instâncias públicas da ITC2011	36
4.2	Restrições das instâncias públicas da ITC2011	37
4.3	Características das instâncias ocultas da ITC2011	38
4.4	Restrições das instâncias ocultas da ITC2011	39
4.5	Resultados obtidos pelo algoritmo <i>Simulated Annealing</i>	40
4.6	Resultados obtidos pelo algoritmo <i>Iterated Local Search</i>	41
4.7	Resultados obtidos pelo algoritmo <i>Late Acceptance Hill Climbing</i>	42
4.8	Resultados obtidos pelo algoritmo <i>Variable Neighborhood Search</i>	43
4.9	Resultados obtidos pelo algoritmo <i>Reduced Variable Neighborhood Search</i>	44
4.10	Resultados obtidos pelo alg. <i>Variable Decomposition Neighborhood Search</i>	45
4.11	Resultados obtidos pelo algoritmo <i>Skewed Variable Neighborhood Search</i>	46
4.12	Comparativo entre os algoritmos propostos	48
4.13	Resultados obtidos pela equipe GOAL na primeira fase da ITC2011	51
4.14	Resultado da primeira fase da ITC2011	52
4.15	Resultados obtidos pela equipe GOAL na segunda fase da ITC2011	54
4.16	Resultados da segunda fase da ITC2011	55
4.17	Resultados obtidos pela equipe GOAL na terceira fase da ITC2011	56

4.18 Resultados da terceira fase da ITC2011	57
---	----

Lista de Algoritmos

3.1	Implementação desenvolvida do Método Não Ascendente Randômico . . .	26
3.2	Implementação desenvolvida da metaheurística ILS	28
3.3	Implementação desenvolvida da metaheurística SA	29
3.4	Implementação desenvolvida da metaheurística VNS	30
3.5	Implementação desenvolvida do LAHC	32

“Computer science is no more about computers than astronomy is about telescopes.”

— Edsger Dijkstra

Nomenclatura

BALV	Busca Adaptativa em Larga Vizinhança
GOAL	Grupo de Otimização e Algoritmos
ILS	<i>Iterated Local Search</i>
ITC2011	<i>Third International Timetabling Competition</i>
KHE	<i>Kingston High School Timetabling Engine</i>
LAHC	<i>Late Acceptance Hill Climbing</i>
RVNS	<i>Reduced Variable Neighborhood Search</i>
SA	<i>Simulated Annealing</i>
SVNS	<i>Skewed Variable Neighborhood Search</i>
VDNS	<i>Variable Decomposition Neighborhood Search</i>
VNS	<i>Variable Neighborhood Search</i>
XHSTT	<i>eXtensible High School TimeTabling</i>
XML	<i>eXtensible Markup Language</i>

Capítulo 1

Introdução

O Problema da Programação de Horários Escolares, cujo estudo computacional iniciou-se em Gotlieb (Gotlieb 1963), consiste em programar um horário escolar de modo que nenhuma classe ou professor participe de mais de uma aula no mesmo instante de tempo. Além disso, o horário deve atender a outras restrições especificadas *a priori*.

A programação automática de horários escolares tem sido alvo de diversas pesquisas nas áreas de Pesquisa Operacional e Inteligência Artificial. Em Santos e Souza (Santos and Souza 2007) são apresentadas algumas das razões para este interesse:

- dificuldade de resolução: encontrar um quadro de horários que satisfaça todos os interesses envolvidos é uma tarefa difícil, ademais, frequentemente, a simples construção de um quadro de horários válido já é uma tarefa complicada;
- importância prática: a confecção de um bom quadro de horários pode melhorar a satisfação do corpo docente e permitir que a instituição de ensino seja mais eficiente na gestão de seus recursos, além do mais, a programação adequada das atividades letivas permite um melhor desempenho dos alunos;
- importância teórica: o problema apresentado neste trabalho é classificado como \mathcal{NP} -Difícil (Garey and Johnson 1979) e progressos na solução de problemas desse tipo são um dos grandes objetivos das pesquisas correntes em computação, matemática e pesquisa operacional.

Três competições internacionais foram promovidas com o intuito de estimular cientistas a melhorar as técnicas de otimização existentes para a solução do problema. A

primeira ocorreu em 2003 (IDSIA 2012) e foi vencida por Kostuch (Kostuch 2005) com uma abordagem de 3 fases baseada no algoritmo *Simulated Annealing* (Kirkpatrick, Gellat and Vecchi 1983). A segunda (McCollum 2012), que ocorreu em 2007, foi composta por três subproblemas separados e vencida por Muller (Muller 2009) também com uma abordagem baseada em *Simulated Annealing*. A última ocorreu em 2011-2012 (of Twente 2012) e foi vencida pelo autor deste trabalho juntamente com a equipe do GOAL (Grupo de Otimização e Algoritmos) da Universidade Federal de Ouro Preto (Fonseca, Santos, Toffolo, Brito and Souza 2012). Essa dissertação apresenta o resolvidor ganhador da competição e bem como métodos e melhorias desenvolvidos subsequentemente à competição.

1.1 Objetivo

Métodos baseados em Programação Inteira já foram propostos para o problema (Santos, Uchoa, Ochi and Maculan 2012), porém são capazes de resolver apenas um subconjunto de instâncias do problema em tempo de processamento viável. Atualmente, as abordagens metaheurísticas são mais comumente aplicadas ao problema (Barbosa and Souza 2011, Muller 2009, Lú and Hao 2010).

Nesse sentido, o principal objetivo do presente trabalho é propor técnicas heurísticas para o modelo do Problema da Programação de Horários Escolares proposto pela *Third International Timetabling Competition*. O referido modelo do problema é genérico e trata um conjunto amplo de restrições, tornando-se desafiador para a comunidade acadêmica.

Os algoritmos implementados consideram instâncias especificadas no formato XHSTT como entrada. Tal fato possibilita que qualquer instância especificada no referido formato possa ser trabalhada pelos algoritmos propostos. O formato XHSTT foi proposto por Post *et al.* (Post, Ahmadi, Daskalaki, Kingston, Kyngas, Nurmi and Ranson 2010, Post, Kingston, Ahmadi, Daskalaki, Gogos, Kyngas, Nurmi, Musliu, Pillay, Santos and Schaerf 2011) e vem sendo utilizado para especificar as instâncias da ITC2011.

1.2 Organização do Trabalho

O restante do trabalho está organizado da seguinte maneira:

- Capítulo 2: apresenta em detalhes o modelo do Problema da Programação de Horários Escolares abordado bem como o formato XHSTT e a *Third International Timetabling Competition*;
- Capítulo 3: descreve os métodos heurísticos implementados bem como a estrutura de vizinhança adotada;
- Capítulo 4: apresenta os experimentos computacionais realizados. Considerou-se as instâncias e regras fornecidas pela *Third International Timetabling Competition*;
- Capítulo 5: apresenta as considerações finais sobre o trabalho bem como sugestões de trabalhos futuros.

Capítulo 2

Problema da Programação de Horários Escolares

Sugerido pela *Third International Timetabling Competition* 2011, o modelo de Programação de Horários Escolares (*High School Timetabling*) surgiu com o objetivo de fornecer um modelo genérico capaz de atender às diversas particularidades do Problema de Programação de Horários Escolares ao redor do mundo (Kingston 2005, Wright 1996, Nurmi and Kyngas 2007, Valourix and Housos 2003, de Haan, Landman, Post and Ruizenaar 2007, Santos and Souza 2007). O modelo é dividido em quatro entidades principais:

2.1 Tempos

A entidade tempo consiste de um espaço de tempo (*timeslot*) ou de um conjunto de espaços (grupo de espaços de tempo) que estão disponibilizados para a realização das alocações.

2.2 Recursos

Os recursos são divididos em três categorias (Post, Ahmadi, Daskalaki, Kingston, Kyngas, Nurmi and Ranson 2010):

classes: grupo de estudantes que atende a um determinado conjunto de eventos em comum. Restrições importantes para as classes são controlar o tempo ocioso e o número de aulas por dia;

professores: um professor pode ser predefinido para atender um evento. Em alguns casos ele não é predefinido e deve ser alocado de acordo com sua qualificação e limites de carga de trabalho;

salas: a maioria dos eventos ocupa uma sala. Uma sala possui uma determinada capacidade e um conjunto de características, como por exemplo, a disponibilidade de recursos multimídia ou a possibilidade de realização de aulas de educação física.

2.3 Eventos

Um evento é um conjunto de encontros (aulas) sobre um determinado assunto. Eventos podem também ser agrupados, gerando um grupo de eventos. O termo classe é usado para designar um conjunto de estudantes que assistem ao mesmo conjunto de eventos (Post, Ahmadi, Daskalaki, Kingston, Kyngas, Nurmi and Ranson 2010).

Uma alocação de um evento a um espaço de tempo é chamada encontro e uma alocação de um recurso a um evento é denominada tarefa. Um evento possui os seguintes atributos:

- A *duração*, que representa o número de espaços de tempo que têm que ser alocados ao evento;
- A *classe* relacionada ao evento;
- Os recursos que são pré-alocados ao evento (opcional);
- A *carga de trabalho*, que irá ser adicionada à carga de trabalho total dos recursos alocados ao evento (opcional);
- O *timeslot* em que deverá ser alocado (opcional).

2.4 Restrições

Post *et al.* (Post, Ahmadi, Daskalaki, Kingston, Kyngas, Nurmi and Ranson 2010) agrupa as restrições em três grupos: restrições básicas do problema de agendamento, restrições para os eventos e restrições para os recursos. As subseções a seguir apresentam cada um dos conjuntos de restrições bem como a avaliação de função objetivo.

2.4.1 Restrições Básicas de Agendamento

1. `ASSIGNTIMECONSTRAINT`. Alocar espaços de tempo a cada evento;
2. `ASSIGNRESOURCECONSTRAINT`. Alocar os recursos a cada evento;
3. `PREFERTIMESCONSTRAINT`. Indica que determinado evento tem preferência por determinado(s) timeslot(s);
4. `PREFERRESOURCESCONSTRAINT`. Indica que determinado evento tem preferência por determinado(s) recurso(s).

2.4.2 Restrições para os Eventos

1. `LINKEVENTSCONSTRAINT`. Agendar os grupos de eventos no mesmo tempo de início;
2. `SPREADEVENTSCONSTRAINT`. Agendar os eventos de cada grupo de eventos para cada grupo de tempo entre um número mínimo e um máximo de vezes. Essa restrição pode ser usada, por exemplo, para definir um limite diário de aulas;
3. `AVOIDSPLITASSIGNMENTSCONSTRAINT`. Para cada grupo de eventos, alocar um determinado recurso a todos os eventos do grupo. Com essa restrição, pode-se forçar que, por exemplo, todas as alocações de um evento ocorram na mesma sala;
4. `DISTRIBUTESPLITEVENTSCONSTRAINT`. Para cada evento, alocar entre um número mínimo e máximo de aulas consecutivas de uma duração dada. Essa restrição se justifica pois um número grande de aulas consecutivas do mesmo assunto pode prejudicar o aprendizado;
5. `SPLITEVENTSCONSTRAINT`. Limita o número de aulas não consecutivos em que um evento será agendado e sua duração. Um exemplo de aplicação dessa restrição

é forçar que um dado evento com quatro aulas seja alocado em dois blocos de aulas pareadas, uma vez que uma aula em separado não seria suficiente para se dar andamento ao conteúdo.

2.4.3 Restrições para os Recursos

1. **AVOIDCLASHESCONSTRAINT.** Agendar os recursos sem conflitos, ou seja, sem alocar o mesmo recurso a mais de um evento ao mesmo tempo;
2. **AVOIDUNAVAILABLETIMESCONSTRAINT.** Evitar alocar os recursos nos horários em que estão indisponíveis. Com essa restrição é possível, por exemplo, evitar que uma sala seja alocada num horário em que está reservada para limpeza ou evitar que um professor seja alocado num horário em que participa de algum curso de capacitação;
3. **LIMITWORKLOADCONSTRAINT.** A cada evento está associada uma carga de trabalho. Essa carga é somada à carga de trabalho do recurso que estiver a ele designado. Essa restrição objetiva a manter a carga de trabalho dos recursos entre um limite mínimo e máximo. Essa restrição pode ser utilizada, por exemplo, para limitar o número de disciplinas que um professor irá lecionar;
4. **LIMITIDLETIMES.** O número de horários ociosos em cada grupo de espaços de tempo deve estar entre um limite mínimo e máximo para cada recurso selecionado. Tipicamente, um grupo de espaços de tempo consiste dos *timeslots* de um dia da semana. Essa restrição é utilizada para evitar horários sem atividade entre horários ativos na agenda de cada recurso;
5. **LIMITBUSYTIMESCONSTRAINT.** O número de horários ocupados em cada grupo de espaços de tempo deve estar entre um limite mínimo e máximo para cada recurso selecionado. Um número grande de alocações num mesmo dia pode prejudicar o desempenho dos estudantes e dos professores, o que justifica o uso dessa restrição;
6. **CLUSTERBUSYTIMESCONSTRAINT.** O número de grupos de tempo com um *time-slot* alocado a um recurso deve figurar entre um limite mínimo e máximo. Tipicamente, o grupo de tempo são dias e, por exemplo, um professor requer no máximo três dias com aulas.

2.4.4 Função Objetivo

A função objetivo $f(\cdot)$ é calculada para uma solução s em termos de violações a cada restrição, penalizadas de acordo com um peso que pode ser especificado por restrição. Divide-se ainda as restrições entre fortes, cujo atendimento é mandatório e fracas, cujo atendimento é desejável mas não obrigatório. Cada instância pode definir se um dada restrição é forte ou fraca. Para mais detalhes, veja Post *et al.* (Post, Ahmadi, Daskalaki, Kingston, Kyngas, Nurmi and Ranson 2010, Post, di Gaspero, Kingston, McCollum and Schaerf 2012).

A avaliação final da função objetivo é dada por um único valor onde a parte inteira indica o custo com relação às restrições fortes e a parte decimal indica o custo com relação às restrições fracas. Como exemplo, o valor 2,00053 indica que a solução possui custo 2 de factibilidade e custo 53 de qualidade.

2.5 Espaço de Busca

O espaço de busca para instâncias mais simples Problema da Programação de Horários (o formato XHSTT tem grande flexibilidade), pode ser modelado da seguinte maneira. Dados:

- Um conjunto de horários $T = \{1, \dots, t\}$;
- Um conjunto de aulas $A = \{1, \dots, a\}$;
- Um conjunto de recursos $R = \{1, \dots, r\}$;
- Um conjunto de categorias de recursos $R^K = \{R_1^K, \dots, R_k^K\}$, onde $R_1^K \cup R_2^K \cup \dots \cup R_k^K = R$ e k é o número de categorias de recursos;
- Um conjunto de pares de aulas em conflito $C = \{(a_1, a_2), \dots, (a_n, a_m)\}$.

Assumindo-se que as variáveis:

- $x_{a,t} \in \{0, 1\}$ indicam se uma aula a possui uma alocação no horário t ; e
- $y_{a,t,r} \in \{0, 1\}$ indicam se um recurso r é usado pela aula a no horário t .

Deve-se encontrar um assinalamento para $x_{a,t}$ e $y_{a,t,r}$ que satisfaça as restrições fortes definidas em cada instância. Usualmente, temos ao menos as seguintes restrições definidas como fortes:

Assinalar um *timeslot* t para cada uma das aulas:

$$\sum_{t \in T} x_{a,t} = 1, \forall a \in A \quad (2.1)$$

Evitar que pares de aulas a_1 e a_2 que compartilham recursos pré-alocados $(a_1, a_2) \in C$ sejam alocados ao mesmo horário t :

$$x_{a_1,t} + x_{a_2,t} \leq 1, \forall t \in T, \forall (a_1, a_2) \in C \quad (2.2)$$

Deve-se ainda evitar que um recurso r seja alocado a mais de uma aula num mesmo horário t :

$$\sum_{a \in A} y_{a,t,r} \leq 1, \forall t \in T, \forall r \in R \quad (2.3)$$

Alocar recursos de cada categoria $K \in R^K$ para uma aula a apenas quando pertencerem ao conjunto $G(a, K)$ de recursos possíveis da categoria K para a aula a :

$$y_{a,t,r} \leq x_{a,t}, \forall a \in A, \forall t \in T, \forall K \in R^K, \forall r \in G(a, K) \quad (2.4)$$

$$\sum_{r \in G(a,K)} y_{a,t,r} \leq x_{a,t}, \forall a \in A, \forall t \in T, \forall K \in R^K \quad (2.5)$$

2.6 Formato XHSTT

Há vários modelos do Problema de Programação de Horários Escolares difundidos na literatura. Alguns modelos do problema ao redor do mundo são discutidos a seguir:

Austrália: escolas australianas têm dias escolares curtos e alta utilização dos professores nesses dias. Consequentemente não há grande demanda por agendas compactas para professores. Porém, há grande utilização de salas especializadas como laboratórios de ciências e computação dessa forma `AVOIDSPLITASSIGNMENTSCONSTRAINT` é uma restrição chave. Restrições ligadas à carga de trabalho dos professores também são críticas (Abramson 1991, Kingston 2005);

Brasil: escolas brasileiras distribuem seus conteúdos semanalmente. Professores usualmente trabalham em mais de uma instituição, assim, restrições ligadas à indisponibilidade de horário dos professores (`AVOIDUNAVAILABLETIMES`) são críticas. Restrições ligadas à duração das aulas também são importantes (Santos 2006, Santos and Souza 2007);

Inglaterra: escolas inglesas possuem grande variação entre a duração de disciplinas curriculares, podendo ser cicladas em uma ou duas semanas. Adicionalmente, estudantes podem se matricular em disciplinas eletivas a sua escolha. Restrições ligadas à duração dos eventos, como `DISTRIBUTESPLITEVENTSCONSTRAINT` e `SPLITEVENTSCONSTRAINT` são importantes nesse modelo (Wright 1996).

Finlândia: em escolas finlandesas estudantes assistem a disciplinas do grupo base ao qual pertencem e podem atender ainda a disciplinas eletivas. Geralmente professores e salas são pré-alocados de acordo com a compatibilidade com as disciplinas. Agendas compactas para estudantes (`LIMITIDLETIMES`) são mandatórias e horários ociosos para os estudantes são altamente inapropriados (Nurmi and Kyngas 2007).

Grécia: escolas gregas são divididas em duas porções: lyceum (séries 7 a 9) e gymnasium (séries 10 a 12). Estudantes devem escolher entre três especializações com grades curriculares distintas. As aulas são distribuídas em cinco dias com seis a sete horários cada. Essa alocação é repetida semanalmente. Restrições ligadas à duração das aulas (`DISTRIBUTESPLITEVENTSCONSTRAINT` e `SPLITEVENTSCONSTRAINT`) são críticas. Manter poucos horários ociosos para os profes-

sores também é importante (Birbas, Daskalaki and Housos 1997, Valourix and Housos 2003);

Holanda: em escolas holandesas os horários são repetidos semanalmente e sua validade pode ser de seis semanas, um trimestre, semestre ou o ano inteiro. Geralmente professores são pré-allocados às disciplinas e acordos trabalhistas permitem um ou dois dias de folga, assim a restrição `AVOIDUNAVAILABLETIMES` é mandatória. Estudantes com talento para esportes ou dança são liberados da primeira ou última aula de alguns dias (Willemen 2002, de Haan, Landman, Post and Ruizenaar 2007).

A variedade de restrições e de necessidades entre as instituições de ensino fez com que, tradicionalmente, cada pesquisador abordasse um modelo do problema e usasse procedimentos de entrada e saída diferentes. Isso dificulta o compartilhamento de instâncias e resolvedores, bem como análises experimentais mais abrangentes.

O problema da compatibilidade de dados em problemas de programação de horários foi discutido na *First International Conference on the Theory and Practice of Automated Timetabling*, o que posteriormente culminou na especificação do formato XHSTT para expressar instâncias e soluções do problema.

O formato XHSTT (*eXtensible High School TimeTabling*) é uma extensão do conhecido formato XML (*eXtensible Markup Language*) (W3C Acessado em junho de 2012). O código da Figura 2.1 apresenta um exemplo em alto nível de arquivo no formato XHSTT. Para mais detalhes sobre a especificação de instâncias e soluções no formato XHSTT, veja Post *et al.* (Post, Ahmadi, Daskalaki, Kingston, Kyngas, Nurmi and Ranson 2010) e Kingston (Kingston 2012a).

2.7 ITC2011

Seguindo a primeira e segunda competições, a *Third International Timetabling Competition* (Post, di Gaspero, Kingston, McCollum and Schaerf 2012) objetiva estimular a pesquisa em programação de horários em geral. Outro objetivo geral da competição é alinhar as pesquisas teóricas na área com a aplicação prática ao oferecer instâncias reais do problema especificadas no formato XHSTT. Os objetivos específicos da competição são (of Twente 2012):

```

<HighSchoolTimetableArchive>
  <Instances>
    <Instance Id="Sudoku4x4">
      <Times>
        ...
      </Times>
      <Resources>
        ...
      </Resources>
      <Events>
        ...
      </Events>
      <Constraints>
        ...
      </Constraints>
    </Instance>
  </Instances>
  <SolutionGroups>
    ...
  </SolutionGroups>
</HighSchoolTimetableArchive>

```

Figura 2.1: Exemplo de alto nível de instância especificada no formato XHSTT

- permitir que pesquisadores avaliem suas técnicas em um ambiente competitivo de problemas práticos reais;
- estimular a pesquisa na área de problemas reais complexos pertencentes à classe \mathcal{NP} -Difícil;
- atrair pesquisadores de variadas áreas para competir;
- aumentar o desenvolvimento de algoritmos para a programação de horários;
- gerar melhores soluções de todos os tempos para as instâncias problema.

A competição teve início em Outubro de 2011, atraiu dezessete grupos de pesquisa para competir e se encerrou em Julho de 2012. A mesma foi constituída de três etapas (of Twente 2012):

- A primeira utiliza instâncias públicas e objetiva a gerar a melhor solução de todos os tempos para as mesmas. Competidores podem usar qualquer recurso e tecnologia para gerar as soluções. Os competidores devem enviar seus resultados sobre as

instâncias públicas seguindo as regras da segunda fase para que sejam selecionados cinco finalistas.

- A segunda fase utiliza instâncias adicionais ocultas. Os resolvidores dos cinco finalistas são executados sobre as referidas instâncias com um limite de 1000 segundos normalizado de acordo com uma máquina de referência, usando um programa de *benchmark* fornecido. Os resolvidores não podem conhecer previamente as instâncias nem soluções, devem ser monoprocessados. Os mesmos podem utilizar bibliotecas adicionais de software, desde que as mesmas tenham seu código fonte disponível sobre uma licença permissiva.
- A terceira fase é semelhante à primeira: objetiva a gerar a melhor solução de todos os tempos para as instâncias em execuções sem restrições de tempo ou tecnologia; porém, são consideradas as instâncias ocultas.

Todas as regras e procedimentos da competição foram divulgados publicamente em <http://www.utwente.nl/ctit/itc2011/> desde o início da competição de modo a promover a lisura da mesma.

2.7.1 Participantes

Dezessete equipes participaram da ITC2011. A Tabela 2.1 apresenta seus codinomes, país de origem e data de registro. A equipe do Departamento de Computação da Universidade Federal de Ouro Preto, composta por George Fonseca, Haroldo Santos, Túlio Toffolo e Samuel Brito participou da competição sob o codinome GOAL (Grupo de Otimização e Algoritmos).

Tabela 2.1: Participantes da ITC2011

Número	Nome do projeto	País	Registro
1	Ying	China	Out 11, 2011
2	HSU	Alemanha	Out 13, 2011
3	Borburata	Venezuela	Out 14, 2011
4	Bond	Canadá	Out 17, 2011
5	Optable	Brasil	Nov 1, 2011
6	Agata	Brasil	Nov 25, 2011
7	Aziz	Malásia	Jan 7, 2012
8	HST	Polônia	Jan 7, 2012
9	VAGOS	Grécia	Jan 23, 2012
10	Join	Índia	Fev 2, 2012
11	Lectio	Dinamarca	Fev 12, 2012
12	Saman	Sri Lanka	Fev 19, 2012
13	DarwinEd	Chile	Fev 23, 2012
14	GOAL	Brasil	Fev 24, 2012
15	HySST	Reino Unido	Mar 6, 2012
16	HFT	Alemanha	Abr 18, 2012
17	Aspen	Estados Unidos	Abr 20, 2012

Capítulo 3

Técnicas Heurísticas Abordadas

3.1 Método Construtivo

O *Kingston High School Timetabling Engine* (KHE) (Kingston 2012b) é uma plataforma de manipulação de instâncias do problema abordado que conta ainda com um resolvidor, utilizado para a geração de soluções iniciais na presente abordagem. O resolvidor do KHE foi escolhido para gerar soluções iniciais uma vez que é capaz de encontrar soluções iniciais razoáveis de forma eficiente (veja a Tabela 4.12).

O resolvidor do KHE é baseado no conceito Programação de Horários Hierárquica (Kingston 2006), onde alocações menores são unidas de modo a gerar blocos maiores de agendamento até que uma representação completa da solução seja desenvolvida.

A Programação de Horários Hierárquica é suportada pela estrutura de dados *Layer Tree* (Kingston 2006), formada por nós que representam um número qualquer de alocações de espaços de tempo. Uma alocação pode figurar em no máximo um nó. Um *Layer* é um subconjunto de nós com a propriedade de que nenhum deles pode ser sobreposto no tempo. Comumente são agrupados em um *Layer* nós que compartilham recursos.

As restrições fortes do problema são modeladas para essa estrutura de dados e posteriormente resolve-se um problema de *Matching* para encontrar a alocação de horários/recursos. O *Matching* é feito ligando cada nó a um espaço de tempo ou recurso respeitando a propriedade da *Layer Tree*. Para mais detalhes, veja Kingston (Kingston 2006, Kingston 2012b).

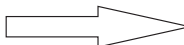
3.2 Estruturas de Vizinhança

A estrutura de vizinhança $N(s)$ é composta de sete movimentos. A vizinhança a respeito de apenas um movimento k será denotada por $N_k(s)$. As sessões a seguir apresentam cada um dos sete movimentos implementados, bem como o procedimento para a geração de vizinhos.

3.2.1 Event Swap (ES)

Nesse movimento, duas aulas a_1 e a_2 são selecionadas e têm seus *timeslots* t_1 e t_2 trocados. A Figura 3.1 apresenta um exemplo desse movimento, onde trocou-se o horário da aula $Geog_3$ com o horário de uma aula de Eng_5 .

Seg	Ter	Qua	Qui	Sex
Math ₁	Eng ₁	Math ₄	Phis ₁	Eng ₃
Math ₂	Eng ₂	Math ₅	Phis ₂	Eng ₄
Math ₃	Chem ₁	Geog ₃	Span ₁	Eng ₅
Geog ₁	Chem ₂	His ₁	Span ₂	Phis ₃
Geog ₂	Chem ₃	His ₂	His ₃	

$ES(Eng_3, Eng_5)$


Seg	Ter	Qua	Qui	Sex
Math ₁	Eng ₁	Math ₄	Phis ₁	Eng ₃
Math ₂	Eng ₂	Math ₅	Phis ₂	Eng ₄
Math ₃	Chem ₁	Eng ₅	Span ₁	Geog ₃
Geog ₁	Chem ₂	His ₁	Span ₂	Phis ₃
Geog ₂	Chem ₃	His ₂	His ₃	

Figura 3.1: Exemplo do movimento Event Swap

3.2.2 Event Move (EM)

Nesse movimento, uma aula a_1 é selecionada e movida de seu horário original t_1 para um novo horário t_2 . A Figura 3.2 apresenta um exemplo desse movimento, onde moveu-se a aula $Chem_3$ do último horário da terça-feira para o último horário da sexta-feira.

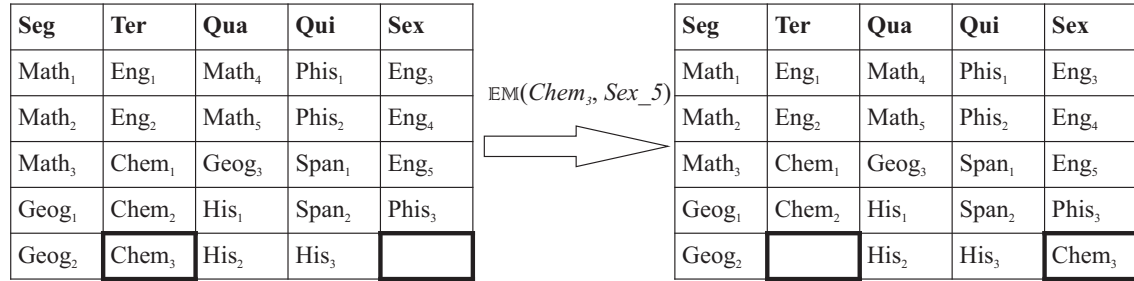


Figura 3.2: Exemplo do movimento Event Move

3.2.3 Event Block Move (EBM)

Assim como o movimento ES, o movimento EBM troca o horário de duas aulas a_1 e a_2 ; porém, caso as aulas tenham duração diferente, a_1 é movida para o *timeslot* subsequente ao último *timeslot* ocupado por a_2 . Esse movimento possibilita trocas de horário sem se perder a contiguidade das alocações. A Figura 3.3 apresenta um exemplo desse movimento, onde a aula $Span_1$ é trocada com a aula $Math_1$. Porém, $Math_1$ possui uma alocação pareada, assim o par de aulas $Math_1$ e $Math_2$ é trocado com a aula $Span_1$.

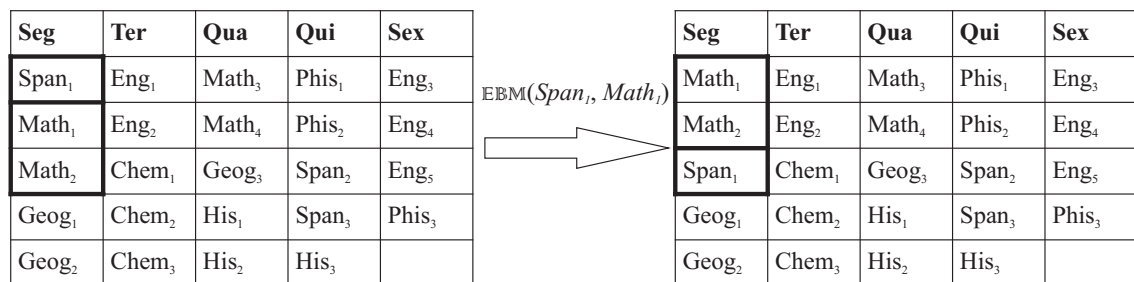


Figura 3.3: Exemplo do movimento Event Block Move

3.2.4 Resource Swap (RS)

Nesse movimento, duas aulas a_1 e a_2 têm seus recursos r_1 e r_2 de um tipo específico K trocados. A Figura 3.4 apresenta um exemplo desse movimento, onde as aulas $Geog_3$ e His_4 têm seus professores $Kate$ e $Arnald$ trocados.

Seg	Ter	Qua	Qui	Sex
Math ₁ Smith	Eng ₁ Anne	Math ₄ Smith	Phis ₁ Laura	Eng ₃ Anne
Math ₂ Smith	Eng ₂ Anne	Math ₅ Smith	Phis ₂ Laura	Eng ₄ Anne
Math ₃ Smith	Chem ₁ John	Geog ₃ Kate	Span ₁ Mark	Eng ₅ Anne
Geog ₁ Kate	Chem ₂ John	His ₁ Arnald	His ₃ Arnald	Phis ₃ Laura
Geog ₂ Kate	Chem ₃ John	His ₂ Arnald	His ₄ Arnald	

$ES(Geog_3, His_4)$

Seg	Ter	Qua	Qui	Sex
Math ₁ Smith	Eng ₁ Anne	Math ₄ Smith	Phis ₁ Laura	Eng ₃ Anne
Math ₂ Smith	Eng ₂ Anne	Math ₅ Smith	Phis ₂ Laura	Eng ₄ Anne
Math ₃ Smith	Chem ₁ John	Geog ₃ Arnald	Span ₁ Mark	Eng ₅ Anne
Geog ₁ Kate	Chem ₂ John	His ₁ Arnald	His ₃ Arnald	Phis ₃ Laura
Geog ₂ Kate	Chem ₃ John	His ₂ Arnald	His ₄ Kate	

Figura 3.4: Exemplo do movimento Resource Swap

3.2.5 Resource Move (RM)

Uma aula a_1 passa a ter um recurso r_2 dentre os recursos disponíveis a ele alocado em detrimento do recurso r_1 anterior. A Figura 3.5 apresenta um exemplo desse movimento em que a aula $Span_1$ teve o recurso professor $Mark$ trocado pelo recurso professor $Jane$.

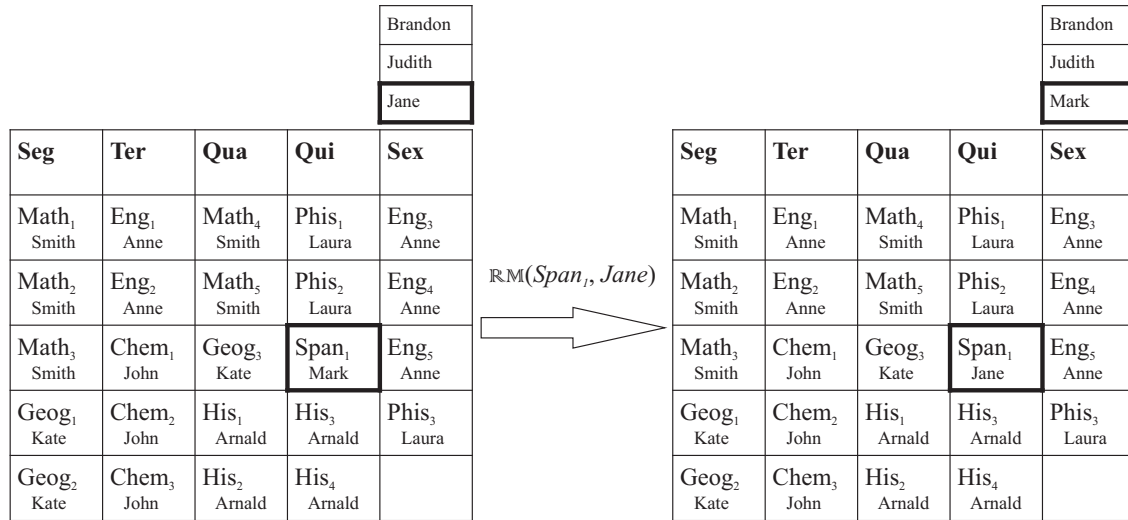


Figura 3.5: Exemplo do movimento Resource Move

3.2.6 Kempe Chain (KC)

Para esse movimento são selecionados dois *timeslots* t_1 e t_2 . Todas as aulas de t_1 e t_2 são listadas, sendo que cada uma representa um nó de um grafo. Caso dois nós n_1 e n_2 (aulas) nesse grafo compartilhem recursos, os mesmos são ligados com uma aresta. Cria-se arestas apenas para nós alocados em *timeslots* distintos; assim, o grafo gerado é um grafo bipartido, conhecido como grafo de conflitos. Cada aresta do grafo de conflitos possui ainda um peso, formado pela diferença de custo na função objetivo supondo a troca de horário entre os eventos representados no par de nós (n_1, n_2) . Posteriormente, busca-se o caminho com o menor custo no grafo de conflitos e efetua-se a troca dos horários em cadeia.

Note que esse movimento possui um custo computacional elevado, por isso não é realizado frequentemente. Em contrapartida, o mesmo é capaz de realizar uma grande modificação numa solução sem depreciá-la significativamente. Assim, o movimento prevê capacidade de diversificação no processo de busca. Relatos de trabalhos que utilizam esse movimento para problemas de *timetabling* podem ser encontrados em Abdullah *et al.* (Abdullah, Shaker, McCollum and McMullan 2010), Tuga *et al.* (Tuga, Berretta and Mendes 2007) e em Lú e Hao (Lú and Hao 2010).

A Figura 3.6 apresenta um exemplo do movimento KC onde selecionou-se os *timeslots*

Seg_2 e Qua_4 para compor o grafo de conflitos. As aulas $Math^1$, $Geog^2$, $Phis^3$ e $Geog^4$ estavam alocadas ao *timeslot* Seg_2 e as aulas $Geog^1$, His^2 , $Span^5$ e Eng^3 estavam alocadas a Qua_4 . Note que para cada aula, temos a classe correspondente em sobrescrito. Assim, ligou-se com aresta as aulas que compartilham ou a classe ou o professor. Como a cadeia $\{Math^1, Geog^1, Geog^2, Hist^2\}$ era a de menor custo (-3), as aulas pertencentes à cadeia tiveram seus *timeslots* invertidos.

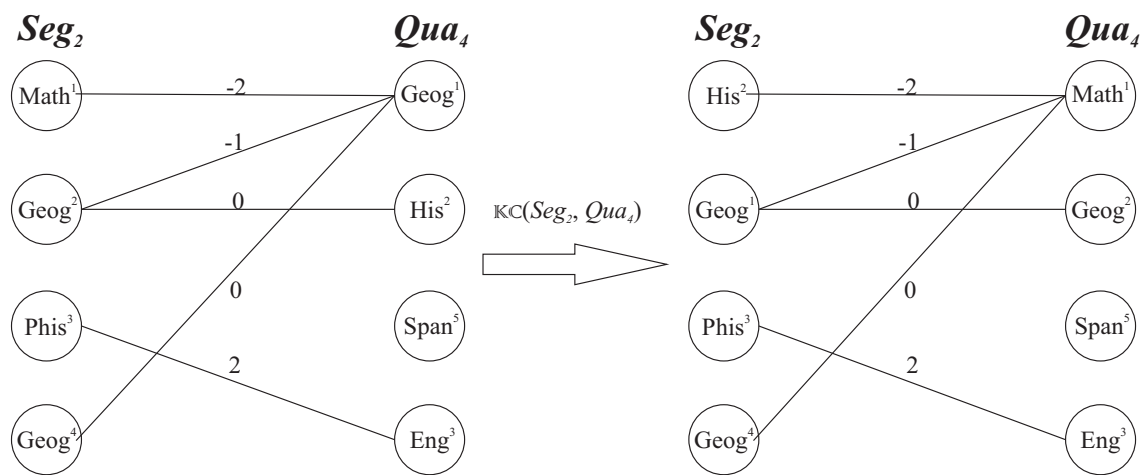


Figura 3.6: Exemplo do movimento Kempe Chain

3.2.7 Permut Resources (PR)

Nesse movimento, seleciona-se um recurso r_1 e realiza-se uma permutação das aulas a ele alocadas. São escolhidas até sete aulas aleatoriamente a serem permutadas ¹. Cada permutação gera uma nova solução. Posteriormente, seleciona-se a melhor solução (de menor custo) dentre as permutações obtidas. A Figura 3.7 apresenta um exemplo desse movimento, onde o recurso professor *Adam* foi selecionado. Cinco novas soluções foram geradas a partir da permutação das aulas a ele alocadas. A melhor delas, em negrito, foi selecionada.

¹Estabeleceu-se como sete o limite de aulas a serem permutadas para que as permutações sejam computadas em um tempo viável.

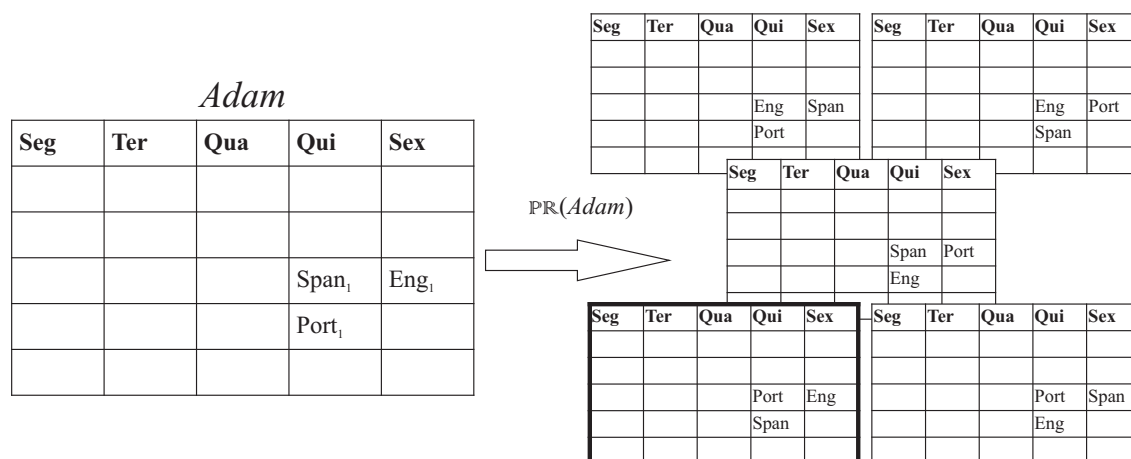


Figura 3.7: Exemplo do movimento Permut Resources

3.2.8 Geração de Vizinhos

O movimento k em $N(s)$ a ser executado é selecionado de forma aleatória na geração de um vizinho. Caso a instância requiera a alocação de recursos (i.e. possua a restrição ASSIGNRESOURCECONSTRAINT), o movimento é selecionado seguindo as seguintes probabilidades: ES = 0.20, EM = 0.38, EBM = 0.10, RS = 0.20, RM = 0.10 e KC = 0.02; caso contrário, os movimentos RS e RM são descartados e as probabilidades são: ES = 0.40, EM = 0.38, EBS = 0.20 e KC = 0.02. O movimento PR é utilizado apenas na fase de perturbação do algoritmo *Iterated Local Search*. Esses valores foram ajustados empiricamente.

Selecionado o movimento, procede-se à escolha dos elementos a serem manipulados. Para cada movimento ², armazenou-se um vetor contendo todas as suas trocas possíveis. Um elemento é selecionado nesse vetor e procede-se a troca para a geração de um vizinho. Esse elemento é então trocado com o último elemento do vetor que ainda não foi escolhido, e será desconsiderado até que a solução seja atualizada. Nesse caso, torna-se a permitir todas as trocas de todas as vizinhanças.

Como exemplo, para o movimento ES armazena-se um vetor contendo todos os pares de aulas distintas. Um par de aulas (a_1, a_2) é selecionado e tem seus horários trocados. Esse par de aulas é desconsiderado para que não se repita durante a busca. Caso a

²Exceto os movimentos KC e PR, para os quais não é viável listar todas as trocas possíveis.

solução seja atualizada, esse movimento se torna válido novamente.

3.3 Busca Local

3.3.1 Método Não Ascendente Randômico

Este método parte de uma solução inicial qualquer e a cada passo analisa um vizinho qualquer da solução corrente e o aceita caso o mesmo seja melhor ou igual à solução. O método para após um número fixo $Iter_{max}$ de iterações sem melhora no valor da solução.

A possibilidade de realizar movimentos laterais (movimentos que não alteram o valor da solução) permite percorrer caminhos de descida que passam por regiões planas. Caso a busca chegue a uma região dessas, o método tem condições de mover-se nela e sair através de uma solução diferente daquela que a ela chegou. O Algoritmo 3.1 apresenta a implementação desenvolvida do Método Não Ascendente Randômico, onde $f(.)$ representa a função objetivo e $N(.)$ a estrutura de vizinhança. Assumiu-se $Iter_{Max} = 1000000$ para a realização dos experimentos.

Algoritmo 3.1: Implementação desenvolvida do Método Não Ascendente Randômico

Entrada: $f(.)$, $N(.)$, s , $Iter_{Max}$, $timeout$

Saída: Melhor solução s encontrada.

```

1  $Iter \leftarrow 0$ ;
2 enquanto  $Iter < Iter_{max}$  e  $elapsedTime < timeout$  faça
3    $Iter \leftarrow Iter + 1$ ;
4   Gere um vizinho aleatório  $s' \in N(s)$ ;
5   se  $f(s') \leq f(s)$  então
6      $Iter \leftarrow 0$ ;
7      $s \leftarrow s'$ ;
8 retorna  $s$ ;
```

3.3.2 Iterated Local Search

O método *Iterated Local Search* (ILS) foi proposto por Lourenço *et al.* (Lourenco, Martin and Stutzle 2001) é baseado na ideia de que um procedimento de busca local pode ser melhorado gerando-se novas soluções de partida, as quais são obtidas por meio

de perturbações na solução ótima local. O tamanho da perturbação é um parâmetro chave do método. Caso esteja muito elevado, pode eliminar as características desejadas da solução ótimo local; caso esteja muito baixo pode não ser capaz de escapar do ótimo local corrente.

O algoritmo ILS parte de uma busca local realizada na solução inicial s_0 e realiza perturbações de tamanho p_{size} sobre s sucedidas de busca local no intuito de escapar de ótimos locais. Uma perturbação é definida como o aceite incondicional de um vizinho qualquer de s . Assumiu-se o Método de Não Ascendente Randômico (*naoDescRandomico*) como o procedimento de busca local.

A busca local origina uma solução s' que será aceita caso seja melhor que a melhor solução encontrada s^* . Caso aceita, o tamanho da perturbação p_{size} volta ao inicial p_0 . Caso a iteração $Iter$ atinja um limite $Iter_{max}$ o tamanho da perturbação é incrementado. Caso ainda o tamanho da perturbação atinja um limite p_{max} , o mesmo volta ao tamanho inicial p_0 . O Algoritmo 3.2 apresenta a implementação desenvolvida do ILS. Os parâmetros considerados nos experimentos computacionais foram $ILS_{max} = 10$, $p_0 = 2$ e $MaxIter_p = 5$.

3.3.3 Simulated Annealing

Proposta por Kirkpatrick *et al.* (Kirkpatrick, Gellat and Vecchi 1983), a metaheurística *Simulated Annealing* trata de um método probabilístico fundamentado em uma analogia à termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos. Esta técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um laço que gera aleatoriamente, em cada iteração, um único vizinho s_0 da solução corrente s .

Seja Δ a variação de valor da função objetivo ao mover-se para uma solução vizinha candidata, isto é, $\Delta = f(s_0) - f(s)$, o método aceita o movimento e a solução vizinha passa a ser a nova solução corrente se $\Delta < 0$. Caso $\Delta \geq 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de temperatura e que regula a probabilidade de se aceitar soluções de pior custo. A temperatura T pode assumir, inicialmente, um valor elevado T_0 . Após um número fixo de iterações (o qual representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento α , tal que

Algoritmo 3.2: Implementação desenvolvida da metaheurística ILS**Entrada:** $f(\cdot), N(\cdot), ILS_{max}, p_0, MaxIter_p, s, timeout$ **Saída:** Melhor solução s encontrada.

```

1  $s \leftarrow naoAscRandomico(s); s^* \leftarrow s;$ 
2  $p_{size} \leftarrow p_0; Iter \leftarrow 0;$ 
3 para  $i \leftarrow 0$  até  $ILS_{max}$  faça
4   para  $j \leftarrow 0$  até  $p_{size}$  faça
5      $s \leftarrow s_p \in N(s);$ 
6     se  $elapsedTime \leq timeout$  então
7       vá para 18;
8    $s' \leftarrow naoAscRandomico(s);$ 
9   se  $f(s') < f(s^*)$  então
10     $s \leftarrow s'; s^* \leftarrow s';$ 
11     $Iter \leftarrow 0; p_{size} \leftarrow p_0;$ 
12  senão
13     $s \leftarrow s^*;$ 
14     $Iter \leftarrow Iter + 1;$ 
15  se  $Iter = MaxIter$  então
16     $p_{size} \leftarrow p_{size} + p_0;$ 
17    se  $p_{size} \geq p_{max}$  então  $p_{size} \leftarrow p_0;$ 
18  $s \leftarrow s^*;$ 
19 retorna  $s;$ 

```

$T_k \leftarrow \alpha \times T_{k-1}$, sendo $0 < \alpha < 1$. Com esse procedimento, dá-se, no início uma chance maior para escapar de mínimos locais e, à medida que T aproxima-se de zero, o algoritmo comporta-se como o método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora (Glover and Kochenberger 2003). Kirkpatrick *et al.* (Kirkpatrick, Gellat and Vecchi 1983) sugere ainda um procedimento de reaquecimento do sistema até $maxReaq$ vezes, fazendo $T \leftarrow T_0$ caso $T < \epsilon$ para prosseguir com a busca, onde ϵ é um valor suficientemente pequeno (0.0001).

A implementação desenvolvida do *Simulated Annealing* será descrita no Algoritmo 3.3. Os parâmetros considerados nos experimentos computacionais foram $\alpha = 0,97$, $T_0 = 1$, $S_{Amax} = 10.000$ e $maxReaq = \infty$.

Algoritmo 3.3: Implementação desenvolvida da metaheurística SA**Entrada:** $f(\cdot)$, $N(\cdot)$, α , SA_{max} , T_0 , $maxReaq$, s , $timeout$ **Saída:** Melhor solução s encontrada.

```

1  $s^* \leftarrow s$ ;
2  $IterT \leftarrow 0$ ;
3  $T \leftarrow T_0$ ;
4  $req \leftarrow 0$ ;
5 enquanto  $req < maxReaq$  e  $elapsedTime < timeout$  faça
6   enquanto  $IterT < SA_{max}$  e  $elapsedTime < timeout$  faça
7      $IterT \leftarrow IterT + 1$ ;
8     Gere um vizinho aleatório  $s' \in N(s)$ ;
9      $\Delta = f(s') - f(s)$ ;
10    se  $\Delta < 0$  então
11       $s \leftarrow s'$ ;
12      se  $f(s') < f(s^*)$  então  $s^* \leftarrow s'$ ;
13    senão
14      Tome  $x \in [0, 1]$ ;
15      se  $x < e^{-\Delta/T}$  então  $s \leftarrow s'$ ;
16     $T \leftarrow \alpha \times T$ ;
17     $IterT \leftarrow 0$ ;
18    se  $T < \epsilon$  então
19       $T \leftarrow T_0$ ;
20       $req \leftarrow req + 1$ ;
21  $s \leftarrow s^*$ ;
22 retorna  $s$ ;

```

3.3.4 Variable Neighborhood Search

Proposta por Hansen (Hansen and Mladenović 1999), a Busca em Vizinhança Variável, ou *Variable Neighborhood Search* é um método de busca local que consiste em explorar o espaço de busca através de trocas sistemáticas na estrutura de vizinhança.

Na presente implementação, inicia-se o algoritmo considerando a estrutura de vizinhança ES (1) e essa estrutura é alterada seguindo a ordem definida na Seção 3.2 até atingir a estrutura de vizinhança KC. Definiu-se um limite de iterações $N_{max} = 5.000$ em que cada estrutura de vizinhança será testada. Quando esse limite é atingido, a busca segue para a próxima estrutura de vizinhança. Quando a última estrutura de vizinhança (KC) atinge seu limite, a busca volta à primeira estrutura de vizinhança definida (ES).

A busca se encerra quando o limite de tempo (*timeout*) é atingido, retornando a

melhor solução encontrada. O Algoritmo 3.4 apresenta a implementação desenvolvido da metaheurística VNS, onde $f(\cdot)$ denota a função objetivo e $N(\cdot)$, a estrutura de vizinhança.

Algoritmo 3.4: Implementação desenvolvida da metaheurística VNS

Entrada: Solução inicial s .

Saída: Melhor solução s encontrada.

```

1 enquanto  $elapsedTime < timeout$  faça
2    $k \leftarrow 1$ ;
3   enquanto  $k \leq k_{max}$  faça
4     Gere um vizinho aleatório  $s' \in N_k(s)$ ;
5      $s'' \leftarrow naoAscRandomico(s')$ ;
6     se  $f(s'') \leq f(s)$  então
7        $s \leftarrow s''$ ;
8        $k \leftarrow 1$ ;
9     senão
10       $k \leftarrow k + 1$ ;
11 retorna  $s$ ;
```

Reduced Variable Neighborhood Search

Uma redução da busca em vizinhança variável original foi também proposta por Mladenovic (Mladenovic and Hansen 1997), na qual não se tem um método de busca local para melhorar a solução gerada s' a cada iteração. Tal fato pode melhorar a performance do algoritmo VNS em casos em que um método de busca local tem um custo computacional muito elevado. Essa redução é chamada *Reduced Variable Neighborhood Search* (RVNS).

Variable Decomposition Neighborhood Search

Outra variação ao método VNS original é o algoritmo *Variable Decomposition Neighborhood Search* (VDNS) (Hansen, Mladenović and Perez-Britos 2001). A única diferença entre o VNS básico e o método VDNS é que ao invés de permitir que todas as estruturas de vizinhança sejam exploradas na busca local, permite-se que apenas um subconjunto delas seja explorado a cada iteração.

Na presente implementação, foi feita a busca local em cada iteração considerando apenas uma estrutura de vizinhança k ($s'' \leftarrow naoAscRandomico_k(s')$).

Skewed Variable Neighborhood Search

Ao assumir vizinhanças cada vez maiores, a informação relacionada ao melhor ótimo local pode se dissolver e o VNS degenerar em um algoritmo de múltiplos reinícios (Lin and Kernighan 1973) (Hansen and Mladenovic 2000). Para lidar com esses caso, uma nova variante do VNS, o algoritmo *Skewed Variable Neighborhood Search* (SVNS), foi proposto. Nessa variante, tem-se uma regra relaxada para aceitar a solução candidata s'' . A regra relaxada usa uma função de avaliação linear com relação à distância da solução incumbente: i.e. $f(x'')$ é substituído por $f(x'') - \alpha \times \rho(x, x'')$, onde $\rho(x, x'')$ é a diferença do valor da função objetivo entre x e x'' e α é um parâmetro. Definiu-se $\alpha = 0.9$ após um conjunto de testes para ajuste desse parâmetro.

3.3.5 Late Acceptance Hill Climbing

A metaheurística *Late Acceptance Hill Climbing* foi proposta recentemente por Burke e Bykov (Burke and Bykov 2008). Se trata de uma adaptação do método *Hill Climbing* (Russell and Norvig 2002) clássico. Ela consiste em comparar uma nova solução candidata com a solução que era a corrente a algumas iterações atrás para aceitá-la ou rejeitá-la. Note que a solução candidata pode ser aceita mesmo sendo pior que a solução corrente, uma vez que ela é comparada com a solução corrente na i -ésima iteração.

Segundo Burke e Bykov (Burke and Bykov 2012), essa metaheurística foi criada com três objetivos em mente: ser um procedimento de busca local que não exige um procedimento de resfriamento artificial; usar eficientemente a informação coletada durante iterações anteriores da busca; e aplicar um novo mecanismo de aceite de soluções candidatas que não seja complicado(i.e. quase tão simples quanto o do *Hill-Climbing*).

Nesse método, armazena-se uma lista $f'_k = \{f_0, f_1, \dots, f_{l-1}\}$ contendo o valor de função objetivo (*fitness*) das soluções. Inicialmente essa lista é preenchida com o custo da solução inicial s : $f'_k \leftarrow f(s)$, $\forall k \in \{0, \dots, l-1\}$. A cada iteração, uma solução candidata s' é gerada. A solução candidata é aceita se seu *fitness* é menor ou igual ao *fitness* armazenado na posição $i \bmod l$ ³ da lista f' . Ainda, se essa solução for melhor que a melhor solução encontrada, a última é atualizada: $s^* \leftarrow s'$. Posteriormente, a posição $i \bmod l$ da lista f' é atualizada: $f'_v \leftarrow f(x)$. Esse procedimento continua até uma condição de parada ser alcançada. O Algoritmo 3.5 apresenta a implementação desenvolvida

³*mod*: módulo da divisão entre dois valores inteiros.

do LAHC. Observe que a condição de parada adotada foi um tempo limite, que será discutido no Capítulo 4. Após alguns experimentos definiu-se o tamanho da lista f'_k como 500. Alguns exemplos bem sucedidos da aplicação do algoritmo LAHC podem ser encontrados em Verstichel *et al.* (Verstichel and Vanden Berghe 2009), Abuhamdah *et al.* (Abuhamdah 2010) e Ozcan *et al.* (Özcan, Bykov, Birben and Burke 2009).

Algoritmo 3.5: Implementação desenvolvida do LAHC

Entrada: Solução inicial s e parâmetro l .

Saída: Melhor solução s^* encontrada.

```

1   $f'_k \leftarrow f(s) \ \forall k \in \{0, \dots, l - 1\};$ 
2   $s^* \leftarrow s;$ 
3   $i \leftarrow 0;$ 
4  enquanto  $elapsedTime < timeout$  faça
5      Gere um vizinho aleatório  $s' \in N(s);$ 
6       $v \leftarrow i \bmod l;$ 
7      se  $f(s') \leq f'_v$  então
8           $s \leftarrow s';$ 
9          se  $f(s) < f(s^*)$  então
10              $s^* \leftarrow s;$ 
11          $f'_v \leftarrow f(s);$ 
12          $i \leftarrow i + 1;$ 
13 retorna  $s^*;$ 
```

3.4 Resolvedor Submetido à ITC2011

O resolvedor submetido para a ITC2011 utiliza a plataforma KHE para gerar soluções iniciais e é composto por uma abordagem mesclada dos algoritmos *Simulated Annealing* e *Iterated Local Search*. A composição do resolvedor foi definida considerando as técnicas estavam implementadas e devidamente testadas até a data limite da submissão. Outras possibilidades de mistura de algoritmos também são viáveis, porém não foram exploradas na ITC2011 nem no presente trabalho.

No referido resolvedor, o algoritmo *Simulated Annealing* recebe como entrada a solução s gerada pelo KHE e retorna uma solução s' . Posteriormente, o algoritmo ILS recebe a solução s^* como entrada e retorna a solução final s^* . Para que haja tempo hábil de se executar o algoritmo ILS, limitou-se em 5 o número de reaquecimentos do algoritmo *Simulated Annealing*, ou seja, definiu-se $maxReaq = 5$. A Figura 3.8 apresenta a composição do resolvedor submetido pela equipe GOAL à *Third International Timetabling Competition*.

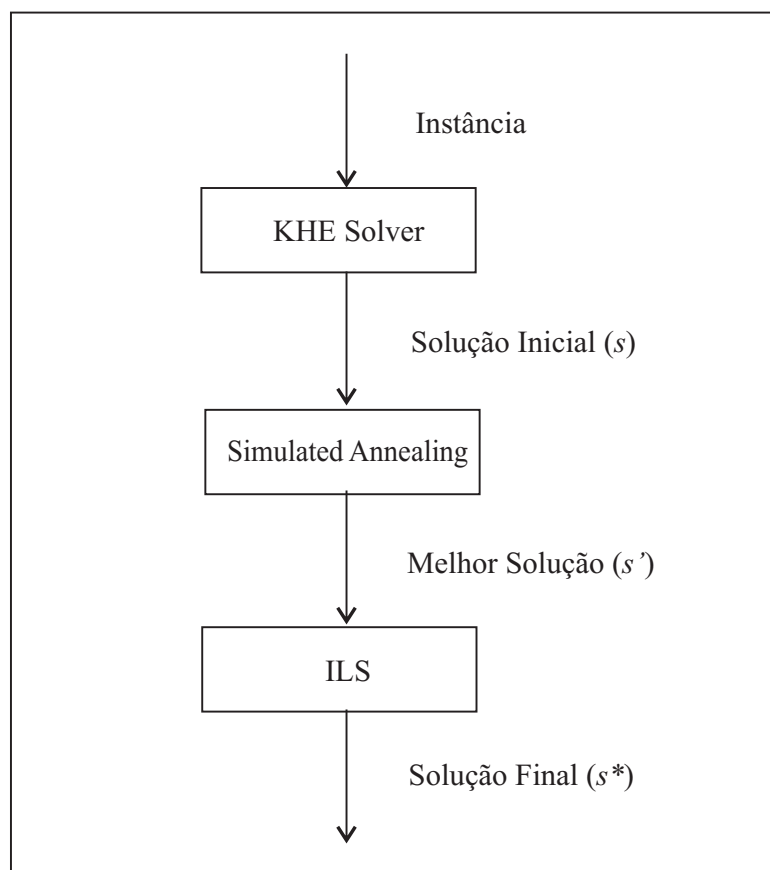


Figura 3.8: Composição do resolvidor da equipe GOAL

Capítulo 4

Experimentos Computacionais

Todos os experimentos foram realizados em um notebook Intel[®] i5 2.4 Ghz com 4GB de RAM sobre o sistema operacional Linux 11.10. A linguagem de programação usada no desenvolvimento do software foi C++ compilada pelo GCC 4.6.1. Os tempos de processamento foram ajustados de acordo com o *benchmark* disponibilizado pela *Third International Timetabling Competition*. O *timeout* foi definido em 1000 segundos, conforme requerido pela segunda etapa da ITC2011. Todos os resultados foram validados pelo validador HSEval <http://sydney.edu.au/engineering/it/~jeff/hseval.cgi>.

4.1 Caracterização das Instâncias

O conjunto de instâncias disponibilizado pela ITC2011 <http://www.utwente.nl/ctit/hstt/archives/XHSTT-2012> foi originado de diversos países e varia de instâncias com somente 75 alocações de aulas necessárias até instâncias cuja solução envolve a alocação de milhares de aulas e centenas de recursos. As instâncias estão divididas em dois conjuntos.

4.1.1 Instâncias Públicas

As instâncias públicas foram disponibilizadas desde o início da ITC2011 junto com a melhor solução até então conhecida para cada. Esse conjunto foi utilizado na primeira fase da ITC2011, onde o objetivo consistia apenas em encontrar a melhor solução possível para as instâncias, sem restrições quanto a tempo nem tecnologias. A Tabela 4.1 apre-

senta as principais características desse conjunto de instâncias, contendo o número de *timeslots* disponíveis para alocação, o número de professores, de salas e de classes de cada instância, bem como o número de aulas a alocar.

Tabela 4.1: Características das instâncias públicas da ITC2011

Instância	<i>Timeslots</i>	Professores	Salas	Classes	Aulas
AustraliaBGHS98	40	56	45	30	1564
AustraliaSAHS96	60	43	36	20	1876
AustraliaTES99	30	37	26	13	806
BrazilInstance1	25	8		3	75
BrazilInstance4	25	23		12	300
BrazilInstance5	25	31		13	325
BrazilInstance6	25	30		14	350
BrazilInstance7	25	33		20	500
EnglandStPaul	27	68	67	67	1227
FinlandArtificialSchool	20	22	12	13	200
FinlandCollege	40	46	34	31	854
FinlandHighSchool	35	18	13	10	297
SecondarySchool	35	25	25	14	306
GreeceHighSchool1	35	29		66	372
GreecePatras3rdHS2010	35	29		84	340
GreecePreveza3rdHS2008	35	29		68	340
ItalyInstance1	36	13		3	133
NetherlandsGEPRO	44	132	80	44	2675
NetherlandsKottenpark2003	38	75	41	18	1203
NetherlandsKottenpark2005	37	78	42	26	1272
SouthAfricaLewitt2009	148	19	2	16	838

A Tabela 4.2 apresenta a distribuição de restrições entre as instâncias consideradas, onde H representa que a restrição é forte e S representa uma restrição fraca. Cada coluna informa a presença (H ou S) de uma restrição ou sua ausência. O índice de cada coluna segue a numeração da Seção 2.4, sendo que A equivale às restrições de agendamento, E equivale às restrições de eventos e R às de recursos.

Tabela 4.2: Restrições das instâncias públicas da ITC2011

Instância	A.1	A.2	A.3	A.4	E.1	E.2	E.3	E.4	E.5	R.1	R.2	R.3	R.4	R.5	R.6
AustraliaBGHS98	H	H	H	H	H	H	H	S	H	H	H			S	H
AustraliaSAHS96	H	H	H	H	H	H	S	S	H	H	H			S	H
AustraliaTES99	H	H	H	H	H		S	S	H	H	H			S	H
BrazilInstance1		H	H	S		S		H		H		S	S		
BrazilInstance4		H	H	S		S		H		H		S	S		
BrazilInstance5		H	H	S		S		H		H		S	S		
BrazilInstance6		H	H	S		S		H		H		S	S		
BrazilInstance7		H	H	S		S		H		H		S	S		
EnglandStPaul	H	H						H	H	H	S	S			
FinlandArtificialSchool	H	H	H		H	H		H		H	H	S		S	
FinlandCollege	H	H	H			H		H		H	H	S		S	
FinlandHighSchool	H	H	H					H		H		S		S	
FinlandSecondarySchool		H	H			H		H		H	H	S		S	
GreeceHighSchool1		H						H	H	H	H				
GreecePatras3rdHS2010		H	H			S		H	H	H	H	S		S	
GreecePreveza3rdHS2008		H	H			S		H	H	H	H	S		S	
ItalyInstance1		H	H			H		H	H	H	H	S		S	
NetherlandsGEPRO		H	H			H		H	H	H	S	S	S	S	
NetherlandsKottenpark2003	H	H	H		H	H		H	H	H	H	S	S	S	
NetherlandsKottenpark2005	H	H	H		H	H		H	H	H	H	S	S	S	
SouthAfricaLewitt2009		H	S			H			S	H					

4.1.2 Instâncias Ocultas

As instâncias ocultas foram disponibilizadas apenas ao final da segunda fase da ITC2011. Esse conjunto foi utilizado na segunda fase da competição com o objetivo de avaliar e comparar os resolvedores em instâncias ocultas e na terceira fase da competição, novamente com o objetivo de encontrar a melhor solução de todos os tempos sem restrições quanto a sua obtenção. De forma análoga ao apresentado na Tabela 4.1, a Tabela 4.3 apresenta as principais características do conjunto de instâncias ocultas da ITC2011.

Tabela 4.3: Características das instâncias ocultas da ITC2011

Instância	Times	Professores	Salas	Classes	Aulas
BrazilInstance2	25	14		6	150
BrazilInstance3	25	16		8	200
BrazilInstance4	25	23		12	300
BrazilInstance6	25	30		14	350
FinlandElementarySchool	35	22	21	291	445
FinlandSecondarySchool2	40	22	21	469	566
Aigio1stHighSchool10-11	35	37		208	532
Italy_Instance4	36	61		38	1101
KosovaInstance1	62	101		63	1912
Kottenpark2003	38	75	41	18	1203
Kottenpark2005A	37	78	42	26	1272
Kottenpark2008	40	81	11	34	1118
Kottenpark2009	38	93	53	48	1301
Woodlands2009	42	40			1353
Spanishschool	35	66	4	21	439
WesternGreeceUniversity3	35	19		6	210
WesternGreeceUniversity4	35	19		12	262
WesternGreeceUniversity5	35	18		6	184

A Tabela 4.4 apresenta a distribuição de restrições entre as instâncias ocultas seguindo o mesmo padrão da Tabela 4.2.

Tabela 4.4: Restrições das instâncias ocultas da ITC2011

Instância	A.1	A.2	A.3	A.4	E.1	E.2	E.3	E.4	E.5	R.1	R.2	R.3	R.4	R.5	R.6
BrazilInstance2	H		S			H		S	H	H	H		S		S
BrazilInstance3	H		S			H		S	H	H	H		S		S
BrazilInstance4	H		S			H		S	H	H	H		S		S
BrazilInstance6	H		S			H		S	H	H	H		S		S
FinlandElementarySchool	H		H			S			H	H	H			S	
FinlandSecondarySchool2	H					H			H	H			S	S	
Aigio1stHighSchool10-11	H		S		H	H			H	H	H		S	S	
Italy_Instance4	H		H			H			H	H	H		S	S	
KosovaInstance1	H		H			H		S	H	H	H		H	H	
Kottenpark2003	H	H	H	H	H	H			H	H	H		S	S	S
Kottenpark2005A	H	H	H	H	H	H			H	H	H		S	S	S
Kottenpark2008	H	H	H	H	H	H			H	H	H		H		S
Kottenpark2009	H	H	H	H	H	H			H	H	H		S		S
Woodlands2009	H		S		S					H	S				
Spanish school	H	H		H		H			H	H	H			S	S
WesternGreeceUniversity3	H					S				H	H		H	H	
WesternGreeceUniversity4	H				H	S				H	H		H	H	
WesternGreeceUniversity5	H					S				H	H		H	H	

4.2 Resultados Obtidos

Todos os resultados apresentados são expressos pelo par x/y , onde x contém a medida de infactibilidade da solução e y a medida de qualidade. Foram consideradas cinco execuções de cada metaheurística abordada para registrar os resultados. Assim, as Tabelas 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 e 4.11 apresentam os resultados obtidos por cada metaheurística, onde $f(s^*)$ apresenta a melhor solução obtida, $f(\bar{s})$ apresenta a média dos resultados e σ o desvio padrão.

Tabela 4.5: Resultados obtidos pelo algoritmo *Simulated Annealing*

Instância	Simulated Annealing		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	0 / 96	0,0 / 107,6	$\pm 0,0$ / $\pm 8,6$
<i>BrazilInstance3</i>	0 / 166	0,0 / 170,6	$\pm 0,0$ / $\pm 5,5$
<i>BrazilInstance4</i>	1 / 147	2,0 / 167,4	$\pm 0,7$ / $\pm 16,2$
<i>BrazilInstance6</i>	0 / 312	0,0 / 332,2	$\pm 0,0$ / $\pm 20,0$
<i>FinlandElementarySchool</i>	0 / 7	0,0 / 10,0	$\pm 0,0$ / $\pm 1,7$
<i>FinlandSecondarySchool2</i>	0 / 925	0,0 / 1036,4	$\pm 0,0$ / $\pm 95,1$
<i>Aigio1stHighSchool10-11</i>	0 / 361	0,0 / 397,4	$\pm 0,0$ / $\pm 34,1$
<i>Italy_Instance4</i>	0 / 12679	0,0 / 13979,0	$\pm 0,0$ / $\pm 821,6$
<i>KosovaInstance1</i>	0 / 4671	3,0 / 5837,8	$\pm 5,6$ / $\pm 1381,4$
<i>Kottenpark2003</i>	0 / 86863	0,4 / 90052,4	$\pm 0,5$ / $\pm 4197,9$
<i>Kottenpark2005A</i>	29 / 29816	30,0 / 33967,0	$\pm 1,0$ / $\pm 3480,7$
<i>Kottenpark2008</i>	8 / 165569	10,0 / 138993,8	$\pm 1,6$ / $\pm 20364,8$
<i>Kottenpark2009</i>	21 / 468315	24,6 / 432784,0	$\pm 2,5$ / $\pm 46523,5$
<i>Woodlands2009</i>	2 / 215	2,0 / 223,8	$\pm 0,0$ / $\pm 6,6$
<i>Spanish school</i>	0 / 5544	0,0 / 7077,6	$\pm 0,0$ / $\pm 1592,8$
<i>WesternGreeceUniversity3</i>	0 / 27	0,0 / 28,4	$\pm 0,0$ / $\pm 0,9$
<i>WesternGreeceUniversity4</i>	0 / 38	0,0 / 39,4	$\pm 0,0$ / $\pm 0,9$
<i>WesternGreeceUniversity5</i>	0 / 50	0,0 / 56,0	$\pm 0,0$ / $\pm 4,0$

Tabela 4.6: Resultados obtidos pelo algoritmo *Iterated Local Search*

Instância	Iterated Local Search		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	0 / 28	0,0 / 42,2	$\pm 0,0$ / $\pm 9,4$
<i>BrazilInstance3</i>	0 / 95	0,0 / 105,8	$\pm 0,0$ / $\pm 14,5$
<i>BrazilInstance4</i>	3 / 112	5,2 / 122,6	$\pm 1,5$ / $\pm 21,3$
<i>BrazilInstance6</i>	0 / 147	1,0 / 158,4	$\pm 1,7$ / $\pm 20,8$
<i>FinlandElementarySchool</i>	0 / 3	0,0 / 3,6	$\pm 0,0$ / $\pm 0,5$
<i>FinlandSecondarySchool2</i>	0 / 0	0,0 / 0,8	$\pm 0,0$ / $\pm 0,8$
<i>Aigio1stHighSchool10-11</i>	0 / 6	0,0 / 12,2	$\pm 0,0$ / $\pm 9,0$
<i>Italy_Instance4</i>	0 / 280	0,0 / 405,2	$\pm 0,0$ / $\pm 163,4$
<i>KosovaInstance1</i>	0 / 16	0,2 / 19,0	$\pm 0,4$ / $\pm 4,1$
<i>Kottenpark2003</i>	1 / 8241	1,8 / 9370,2	$\pm 1,1$ / $\pm 659,1$
<i>Kottenpark2005A</i>	33 / 15509	33,4 / 17679,8	$\pm 0,9$ / $\pm 1955,6$
<i>Kottenpark2008</i>	14 / 20375	14,4 / 23683,8	$\pm 0,9$ / $\pm 2124,3$
<i>Kottenpark2009</i>	28 / 8375	31,8 / 7976,0	$\pm 2,3$ / $\pm 851,5$
<i>Woodlands2009</i>	2 / 7	2,0 / 9,2	$\pm 0,0$ / $\pm 1,9$
<i>Spanish school</i>	0 / 667	0,0 / 773,8	$\pm 0,0$ / $\pm 145,6$
<i>WesternGreeceUniversity3</i>	0 / 5	0,0 / 5,0	$\pm 0,0$ / $\pm 0,0$
<i>WesternGreeceUniversity4</i>	0 / 6	0,0 / 6,6	$\pm 0,0$ / $\pm 0,9$
<i>WesternGreeceUniversity5</i>	0 / 0	0,0 / 0,0	$\pm 0,0$ / $\pm 0,0$

Tabela 4.7: Resultados obtidos pelo algoritmo *Late Acceptance Hill Climbing*

Instância	Late Acceptance Hill Climbing		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	0 / 57	0,0 / 78,6	$\pm 0,0$ / $\pm 24,1$
<i>BrazilInstance3</i>	0 / 142	0,0 / 145,6	$\pm 0,0$ / $\pm 4,8$
<i>BrazilInstance4</i>	5 / 100	8,2 / 121,4	$\pm 2,2$ / $\pm 17,9$
<i>BrazilInstance6</i>	1 / 171	1,4 / 204,8	$\pm 0,5$ / $\pm 39,0$
<i>FinlandElementarySchool</i>	0 / 3	0,0 / 3,8	$\pm 0,0$ / $\pm 0,4$
<i>FinlandSecondarySchool2</i>	0 / 0	0,0 / 0,4	$\pm 0,0$ / $\pm 0,5$
<i>Aigio1stHighSchool10-11</i>	0 / 0	0,8 / 11,4	$\pm 1,8$ / $\pm 8,4$
<i>Italy_Instance4</i>	0 / 82	0,0 / 224,8	$\pm 0,0$ / $\pm 117,8$
<i>KosovaInstance1</i>	0 / 144	14,2 / 1504,8	$\pm 27,9$ / $\pm 2561,7$
<i>Kottenpark2003</i>	0 / 9125	1,6 / 9711,0	$\pm 1,3$ / $\pm 1116,2$
<i>Kottenpark2005A</i>	32 / 17740	33,0 / 18671,0	$\pm 1,0$ / $\pm 1287,0$
<i>Kottenpark2008</i>	14 / 24928	15,2 / 23855,0	$\pm 0,8$ / $\pm 3182,6$
<i>Kottenpark2009</i>	24 / 11280	28,0 / 9192,0	$\pm 2,3$ / $\pm 3319,7$
<i>Woodlands2009</i>	2 / 11	2,0 / 13,2	$\pm 0,0$ / $\pm 2,3$
<i>Spanish school</i>	0 / 638	0,0 / 846,0	$\pm 0,0$ / $\pm 171,4$
<i>WesternGreeceUniversity3</i>	0 / 5	0,0 / 5,0	$\pm 0,0$ / $\pm 0,0$
<i>WesternGreeceUniversity4</i>	0 / 5	0,0 / 9,2	$\pm 0,0$ / $\pm 4,0$
<i>WesternGreeceUniversity5</i>	0 / 0	0,0 / 0,0	$\pm 0,0$ / $\pm 0,0$

Tabela 4.8: Resultados obtidos pelo algoritmo *Variable Neighborhood Search*

Instância	Variable Neighborhood Search		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	0 / 32	0,0 / 40,6	$\pm 0,0$ / $\pm 8,4$
<i>BrazilInstance3</i>	0 / 101	0,0 / 113,0	$\pm 0,0$ / $\pm 12,2$
<i>BrazilInstance4</i>	3 / 113	4,8 / 108,2	$\pm 1,5$ / $\pm 9,0$
<i>BrazilInstance6</i>	0 / 134	0,0 / 157,4	$\pm 0,0$ / $\pm 19,3$
<i>FinlandElementarySchool</i>	0 / 3	0,0 / 3,4	$\pm 0,0$ / $\pm 0,5$
<i>FinlandSecondarySchool2</i>	0 / 0	0,0 / 0,4	$\pm 0,0$ / $\pm 0,5$
<i>Aigio1stHighSchool10-11</i>	0 / 2	0,4 / 10,2	$\pm 0,5$ / $\pm 7,4$
<i>Italy_Instance4</i>	0 / 180	0,0 / 409,0	$\pm 0,0$ / $\pm 195,4$
<i>KosovaInstance1</i>	0 / 18	1,2 / 20,4	$\pm 2,7$ / $\pm 2,3$
<i>Kottenpark2003</i>	1 / 8772	2,0 / 10217,2	$\pm 1,0$ / $\pm 942,0$
<i>Kottenpark2005A</i>	33 / 15762	33,8 / 19059,2	$\pm 1,1$ / $\pm 2898,9$
<i>Kottenpark2008</i>	13 / 24236	15,6 / 23962,0	$\pm 1,7$ / $\pm 3318,5$
<i>Kottenpark2009</i>	29 / 6540	35,0 / 8543,0	$\pm 4,5$ / $\pm 1635,8$
<i>Woodlands2009</i>	2 / 5	2,0 / 8,2	$\pm 0,0$ / $\pm 2,2$
<i>Spanish school</i>	0 / 609	0,0 / 907,8	$\pm 0,0$ / $\pm 185,5$
<i>WesternGreeceUniversity3</i>	0 / 5	0,0 / 5,4	$\pm 0,0$ / $\pm 0,5$
<i>WesternGreeceUniversity4</i>	0 / 5	0,0 / 6,4	$\pm 0,0$ / $\pm 1,1$
<i>WesternGreeceUniversity5</i>	0 / 0	0,0 / 0,0	$\pm 0,0$ / $\pm 0,0$

Tabela 4.9: Resultados obtidos pelo algoritmo *Reduced Variable Neighborhood Search*

Instância	Reduced Variable Neighborhood Search		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	2 / 64	2,2 / 71,4	$\pm 0,4$ / $\pm 5,0$
<i>BrazilInstance3</i>	2 / 143	2,4 / 151,4	$\pm 0,5$ / $\pm 8,7$
<i>BrazilInstance4</i>	19 / 130	21,0 / 112,8	$\pm 1,6$ / $\pm 16,0$
<i>BrazilInstance6</i>	5 / 281	6,0 / 271,0	$\pm 0,7$ / $\pm 21,8$
<i>FinlandElementarySchool</i>	2 / 6	2,6 / 7,4	$\pm 0,5$ / $\pm 1,3$
<i>FinlandSecondarySchool2</i>	0 / 64	0,6 / 86,8	$\pm 0,5$ / $\pm 15,1$
<i>Aigio1stHighSchool10-11</i>	9 / 191	11,2 / 200,0	$\pm 2,6$ / $\pm 18,4$
<i>Italy_Instance4</i>	0 / 2225	0,4 / 2666,6	$\pm 0,5$ / $\pm 290,1$
<i>KosovaInstance1</i>	13 / 208	31,6 / 278,8	$\pm 23,5$ / $\pm 68,3$
<i>Kottenpark2003</i>	1 / 33217	2,4 / 34766,0	$\pm 0,9$ / $\pm 1535,8$
<i>Kottenpark2005A</i>	35 / 22695	35,0 / 22914,0	$\pm 0,0$ / $\pm 169,0$
<i>Kottenpark2008</i>	35 / 31871	36,8 / 38936,6	$\pm 1,5$ / $\pm 4206,6$
<i>Kottenpark2009</i>	43 / 147940	45,4 / 148601,0	$\pm 1,5$ / $\pm 11550,4$
<i>Woodlands2009</i>	10 / 18	10,8 / 16,4	$\pm 0,4$ / $\pm 0,9$
<i>Spanish school</i>	0 / 3008	0,0 / 3068,0	$\pm 0,0$ / $\pm 49,4$
<i>WesternGreeceUniversity3</i>	0 / 20	0,0 / 20,4	$\pm 0,0$ / $\pm 0,5$
<i>WesternGreeceUniversity4</i>	0 / 30	0,0 / 30,0	$\pm 0,0$ / $\pm 0,0$
<i>WesternGreeceUniversity5</i>	2 / 12	2,8 / 16,2	$\pm 0,8$ / $\pm 5,3$

Tabela 4.10: Resultados obtidos pelo alg. *Variable Decomposition Neighborhood Search*

Instância	Variable Decomposition Neighborhood Search		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	0 / 73	0,6 / 63,8	$\pm 0,5$ / $\pm 22,8$
<i>BrazilInstance3</i>	1 / 128	1,6 / 136,8	$\pm 0,5$ / $\pm 10,4$
<i>BrazilInstance4</i>	12 / 102	13,6 / 103,4	$\pm 1,8$ / $\pm 15,6$
<i>BrazilInstance6</i>	1 / 222	2,2 / 231,2	$\pm 0,8$ / $\pm 22,0$
<i>FinlandElementarySchool</i>	0 / 4	0,0 / 4,0	$\pm 0,0$ / $\pm 0,0$
<i>FinlandSecondarySchool2</i>	0 / 0	0,0 / 1,0	$\pm 0,0$ / $\pm 2,0$
<i>Aigio1stHighSchool10-11</i>	3 / 212	4,8 / 259,0	$\pm 2,7$ / $\pm 68,5$
<i>Italy_Instance4</i>	0 / 1107	0,0 / 1271,0	$\pm 0,0$ / $\pm 165,1$
<i>KosovaInstance1</i>	0 / 63	2,0 / 75,6	$\pm 2,7$ / $\pm 9,6$
<i>Kottenpark2003</i>	2 / 8136	2,8 / 7937,8	$\pm 0,4$ / $\pm 845,2$
<i>Kottenpark2005A</i>	24 / 9675	27,0 / 10118,0	$\pm 2,3$ / $\pm 461,1$
<i>Kottenpark2008</i>	13 / 31954	16,8 / 33443,6	$\pm 2,5$ / $\pm 3311,2$
<i>Kottenpark2009</i>	25 / 7730	31,2 / 8563,0	$\pm 5,1$ / $\pm 1112,8$
<i>Woodlands2009</i>	2 / 13	2,0 / 14,4	$\pm 0,0$ / $\pm 1,1$
<i>Spanish school</i>	0 / 989	0,0 / 1126,0	$\pm 0,0$ / $\pm 108,9$
<i>WesternGreeceUniversity3</i>	0 / 13	0,0 / 15,2	$\pm 0,0$ / $\pm 2,2$
<i>WesternGreeceUniversity4</i>	0 / 20	0,0 / 23,6	$\pm 0,0$ / $\pm 3,0$
<i>WesternGreeceUniversity5</i>	0 / 1	1,2 / 3,0	$\pm 1,3$ / $\pm 2,1$

Tabela 4.11: Resultados obtidos pelo algoritmo *Skewed Variable Neighborhood Search*

Instância	Skewed Variable Neighborhood Search		
	$f(s^*)$	$f(\bar{s})$	σ
<i>BrazilInstance2</i>	0 / 22	0,0 / 35,8	$\pm 0,0$ / $\pm 10,3$
<i>BrazilInstance3</i>	0 / 98	0,0 / 105,8	$\pm 0,0$ / $\pm 7,0$
<i>BrazilInstance4</i>	2 / 117	3,8 / 132,4	$\pm 1,3$ / $\pm 17,6$
<i>BrazilInstance6</i>	0 / 112	0,0 / 145,8	$\pm 0,0$ / $\pm 25,7$
<i>FinlandElementarySchool</i>	0 / 3	0,0 / 3,8	$\pm 0,0$ / $\pm 0,4$
<i>FinlandSecondarySchool2</i>	0 / 0	0,0 / 0,8	$\pm 0,0$ / $\pm 0,8$
<i>Aigio1stHighSchool10-11</i>	0 / 2	0,4 / 8,8	$\pm 0,5$ / $\pm 7,8$
<i>Italy_Instance4</i>	0 / 167	0,0 / 321,0	$\pm 0,0$ / $\pm 144,3$
<i>KosovaInstance1</i>	0 / 16	1,2 / 19,6	$\pm 2,7$ / $\pm 4,7$
<i>Kottenpark2003</i>	1 / 9538	2,0 / 10427,0	$\pm 1,0$ / $\pm 903,6$
<i>Kottenpark2005A</i>	33 / 15862	33,8 / 18834,6	$\pm 1,1$ / $\pm 2897,8$
<i>Kottenpark2008</i>	12 / 24363	15,2 / 23756,6	$\pm 1,9$ / $\pm 3374,8$
<i>Kottenpark2009</i>	29 / 7555	34,0 / 8649,0	$\pm 4,6$ / $\pm 1447,5$
<i>Woodlands2009</i>	2 / 5	2,0 / 6,8	$\pm 0,0$ / $\pm 1,3$
<i>Spanish school</i>	0 / 526	0,0 / 783,8	$\pm 0,0$ / $\pm 201,7$
<i>WesternGreeceUniversity3</i>	0 / 5	0,0 / 5,2	$\pm 0,0$ / $\pm 0,4$
<i>WesternGreeceUniversity4</i>	0 / 5	0,0 / 5,2	$\pm 0,0$ / $\pm 0,4$
<i>WesternGreeceUniversity5</i>	0 / 0	0,0 / 0,0	$\pm 0,0$ / $\pm 0,0$

A Tabela 4.12 apresenta um comparativo entre a média das soluções apresentadas por cada metaheurística. O valor de função objetivo da solução inicial dada pelo KHE também é apresentado. No intuito de comparar os resultados, utilizou-se um procedimento similar ao adotado na segunda fase da ITC2011. Para cada instância as metaheurísticas foram avaliadas de 1 a 7, assinalando-se com 1 a metaheurística que obteve, em média, os melhores resultados e com 7 a metaheurística com os piores resultados. A linha Ranking apresenta a classificação média de cada metaheurística em todo o conjunto de instâncias. Os melhores resultados são destacados em negrito.

Tabela 4.12: Comparativo entre os algoritmos propostos

Instância	KHE	SA	ILS	LAHC	VNS	RVNS	VDNS	SVNS
<i>BrazilInstance2</i>	4 / 90	0,0 / 107,6	0,0 / 42,2	0,0 / 78,6	0,0 / 40,6	2,2 / 71,4	0,6 / 63,8	0,0 / 35,8
<i>BrazilInstance3</i>	3 / 240	0,0 / 170,6	0,0 / 105,8	0,0 / 145,6	0,0 / 113,0	2,4 / 151,4	1,6 / 136,8	0,0 / 105,8
<i>BrazilInstance4</i>	39 / 144	2,0 / 167,4	5,2 / 122,6	8,2 / 121,4	4,8 / 108,2	21,0 / 112,8	13,6 / 103,4	3,8 / 132,4
<i>BrazilInstance6</i>	11 / 291	0,0 / 332,2	1,0 / 158,4	1,4 / 204,8	0,0 / 157,4	6,0 / 271,0	2,2 / 231,2	0,0 / 145,8
<i>FinlandElementarySchool</i>	9 / 30	0,0 / 10,0	0,0 / 3,6	0,0 / 3,8	0,0 / 3,4	2,6 / 7,4	0,0 / 4,0	0,0 / 3,8
<i>FinlandSecondarySchool2</i>	2 / 1821	0,0 / 1036,4	0,0 / 0,8	0,0 / 0,4	0,0 / 0,4	0,6 / 86,8	0,0 / 1,0	0,0 / 0,8
<i>Aigio1stHighSchool10-11</i>	14 / 757	0,0 / 397,4	0,0 / 12,2	0,8 / 11,4	0,4 / 10,2	11,2 / 200,0	4,8 / 259,0	0,4 / 8,8
<i>Italy_Instance4</i>	39 / 21238	0,0 / 13979,0	0,0 / 405,2	0,0 / 224,8	0,0 / 409,0	0,4 / 2666,6	0,0 / 1271,0	0,0 / 321,0
<i>KosovaInstance1</i>	1333 / 566	3,0 / 5837,8	0,2 / 19,0	14,2 / 1504,8	1,2 / 20,4	31,6 / 278,8	2,0 / 75,6	1,2 / 19,6
<i>Kottenpark2003</i>	3 / 78440	0,4 / 90052,4	1,8 / 9370,2	1,6 / 9711,0	2,0 / 10217,2	2,4 / 34766,0	2,8 / 7937,8	2,0 / 10427,0
<i>Kottenpark2005A</i>	35 / 23677	30,0 / 33967,0	33,4 / 17679,8	33,0 / 18671,0	33,8 / 19059,2	35,0 / 22914,0	27,0 / 10118,0	33,8 / 18834,6
<i>Kottenpark2008</i>	63 / 140083	10,0 / 1386993,8	14,4 / 23683,8	15,2 / 23855,0	15,6 / 23962,0	36,8 / 38936,6	16,8 / 33443,6	15,2 / 23756,6
<i>Kottenpark2009</i>	55 / 211095	24,6 / 432784,0	31,8 / 7976,0	28,0 / 9192,0	35,0 / 8543,0	45,4 / 148601,0	31,2 / 8563,0	34,0 / 8649,0
<i>Woodlands2009</i>	19 / 0	2,0 / 223,8	2,0 / 9,2	2,0 / 13,2	2,0 / 8,2	10,8 / 16,4	2,0 / 14,4	2,0 / 6,8
<i>Spanish school</i>	1 / 4103	0,0 / 7077,6	0,0 / 773,8	0,0 / 846,0	0,0 / 907,8	0,0 / 3068,0	0,0 / 1126,0	0,0 / 783,8
<i>Western Greece University3</i>	0 / 30	0,0 / 28,4	0,0 / 5,0	0,0 / 5,0	0,0 / 5,4	0,0 / 20,4	0,0 / 15,2	0,0 / 5,2
<i>Western Greece University4</i>	0 / 41	0,0 / 39,4	0,0 / 6,6	0,0 / 9,2	0,0 / 6,4	0,0 / 30,0	0,0 / 23,6	0,0 / 5,2
<i>Western Greece University5</i>	17 / 44	0,0 / 56,0	0,0 / 0,0	0,0 / 0,0	0,0 / 0,0	2,8 / 16,2	1,2 / 3,0	0,0 / 0,0
Raking	-	4,22	2,61	3,39	3,27	6,78	5,11	2,61

4.3 Resultados Oficiais da ITC2011

Primeira fase

Na primeira fase da ITC2011 não havia restrição de tempo de processamento nem de tecnologias utilizadas. Assim, para essa fase, executou-se o nosso resolvedor tomando a melhor solução conhecida como solução inicial ¹. Um limite de 1000 segundos de execução (ajustado de acordo com o *benchmark* provido) foi definido. Caso alguma melhoria fosse alcançada, o resolvedor era executado novamente, tomando a solução melhorada como entrada. A Figura 4.1 apresenta o procedimento de execução do resolvedor nessa fase.

A Tabela 4.13 apresenta os resultados obtidos pelo resolvedor da equipe GOAL na primeira fase da ITC2011, onde $ITC\ f(s^*)$ denota a melhor solução previamente conhecida e $GOAL\ f(s^*)$ denota a melhor solução encontrada pela equipe GOAL. A Tabela 4.14 apresenta o vencedor de cada instância nessa fase da competição.

¹Para as instâncias *AustraliSAHS96* e *AustraliaTES99* a solução inicial gerada pelo KHE foi melhor que a melhor solução conhecida, assim, o KHE foi utilizado nesses casos

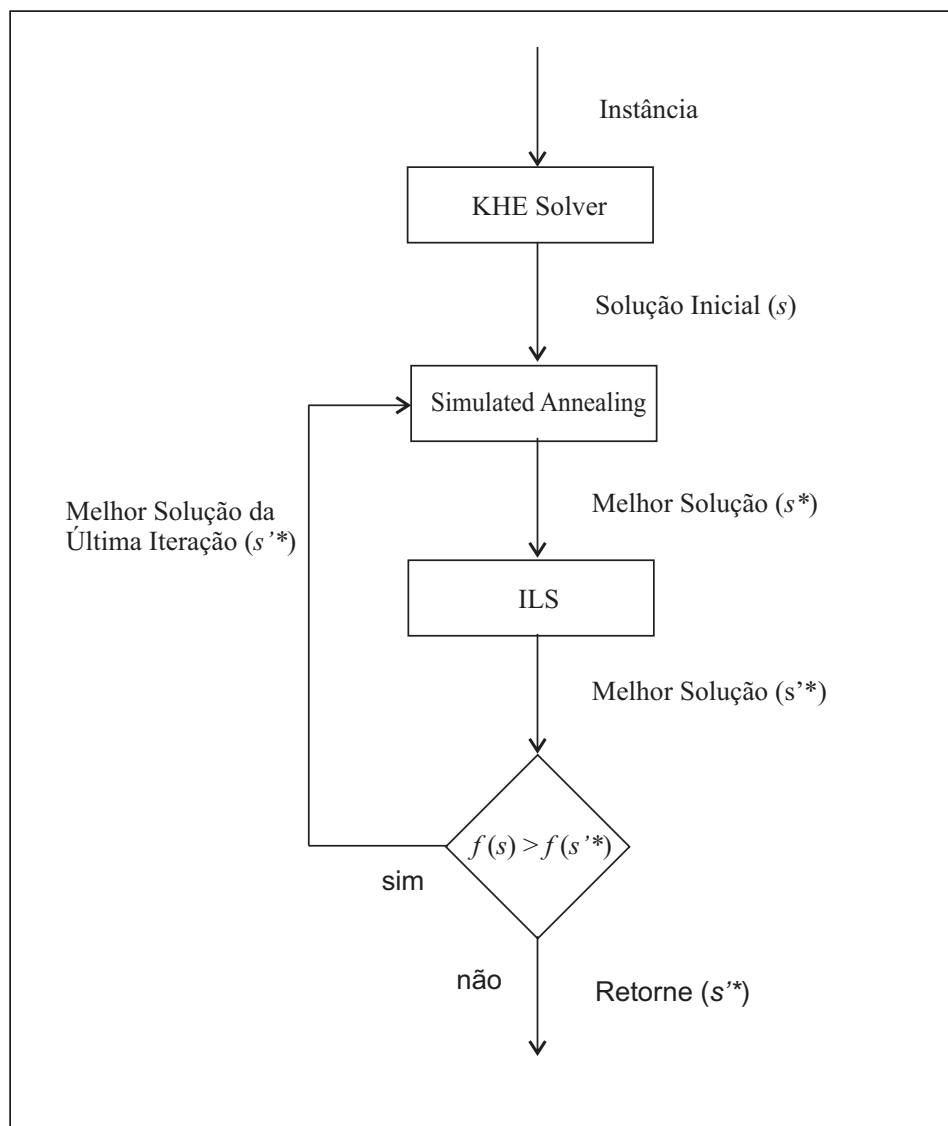


Figura 4.1: Adaptação do resolvidor para a primeira fase da ITC2011

Tabela 4.13: Resultados obtidos pela equipe GOAL na primeira fase da ITC2011

Instância	ITC $f(s_*)$	GOAL $f(s_*)$
<i>AustraliaBGHS98</i>	7 / 433	4 / 367
<i>AustraliaSAHS96</i>	23 / 44	10 / 12
<i>AustraliaTES99</i>	26 / 134	5 / 148
<i>BrazilInstance1</i>	0 / 24	0 / 15
<i>BrazilInstance4</i>	0 / 112	0 / 103
<i>BrazilInstance5</i>	0 / 225	0 / 198
<i>BrazilInstance6</i>	0 / 209	0 / 156
<i>BrazilInstance7</i>	0 / 330	0 / 294
<i>EnglandStPaul</i>	0 / 18444	0 / 11732
<i>FinlandArtificialSchool</i>	0 / 0	0 / 0
<i>FinlandCollege</i>	0 / 0	0 / 0
<i>FinlandHighSchool</i>	0 / 1	0 / 1
<i>FinlandSecondarySchool</i>	0 / 106	0 / 102
<i>GreeceHighSchool1</i>	0 / 0	0 / 0
<i>GreecePatras3rdHS2010</i>	0 / 0	0 / 0
<i>GreecePreveza3rdHS2008</i>	0 / 0	0 / 0
<i>ItalyInstance1</i>	0 / 28	0 / 23
<i>NetherlandsGEPRO</i>	1 / 566	1 / 382
<i>NetherlandsKottenpark2003</i>	0 / 1410	0 / 1189
<i>NetherlandsKottenpark2005</i>	0 / 1078	0 / 963
<i>SouthAfricaLewitt2009</i>	0 / 58	0 / 0

Tabela 4.14: Resultado da primeira fase da ITC2011

Instância	ITC $f(s^*)$	Melhor $f(s^*)$	Equipe vencedora
<i>AustraliaBGHS98</i>	7 / 433	3 / 494	HySTT
<i>AustraliaSAHS96</i>	23 / 44	8 / 52	HySTT
<i>AustraliaTES99</i>	26 / 134	1 / 140	HySTT
<i>BrazilInstance1</i>	0 / 24	0 / 11	VAGOS
<i>BrazilInstance4</i>	0 / 112	0 / 44	VAGOS
<i>BrazilInstance5</i>	0 / 225	0 / 43	VAGOS
<i>BrazilInstance6</i>	0 / 209	0 / 77	VAGOS
<i>BrazilInstance7</i>	0 / 330	0 / 122	VAGOS
<i>EnglandStPaul</i>	0 / 18444	0 / 136	Lectio
<i>FinlandArtificialSchool</i>	0 / 0	-	-
<i>FinlandCollege</i>	0 / 0	-	-
<i>FinlandHighSchool</i>	0 / 1	-	-
<i>FinlandSecondarySchool</i>	0 / 106	0 / 88	Lectio
<i>GreeceHighSchool1</i>	0 / 0	-	-
<i>GreecePatras3rdHS2010</i>	0 / 0	-	-
<i>GreecePreveza3rdHS2008</i>	0 / 0	-	-
<i>ItalyInstance1</i>	0 / 28	0 / 12	VAGOS
<i>NetherlandsGEPRO</i>	1 / 566	1 / 382	GOAL
<i>NetherlandsKottenpark2003</i>	0 / 1410	0 / 532	Lectio
<i>NetherlandsKottenpark2005</i>	0 / 1078	0 / 533	Lectio
<i>SouthAfricaLewitt2009</i>	0 / 0	0 / 0	VAGOS

Segunda fase

Na segunda fase da competição, os organizadores executaram cada resolvidor dez vezes para cada instância com um tempo limite de 1000 segundos. Reproduziu-se os experimentos dos organizadores considerando o *benchmark* disponibilizado e os resultados obtidos são apresentados na Tabela 4.15. A segunda coluna da Tabela 4.15 apresenta os resultados obtidos pelo resolvidor do KHE e a terceira coluna apresenta os resultados do resolvidor da equipe GOAL, sendo $f(s^*)$ a melhor solução encontrada, $f(\bar{s})$ a média e σ o desvio padrão dentre dez execuções.

A Tabela 4.16 apresenta a classificação final dos competidores na segunda fase da ITC2011. Apenas quatro equipes se classificaram para essa etapa, assim a classificação varia de 1 (melhor) a 4 (pior) para cada instância. Vale ressaltar que a equipe GOAL não pode competir nas instâncias *BrazilInstance2*, *BrazilInstance3*, *BrazilInstance4* e *BrazilInstance6* uma vez que elas foram providas pela própria equipe. Assim, elas foram excluídas da classificação, que varia de 1 a 3 nesses casos.

Tabela 4.15: Resultados obtidos pela equipe GOAL na segunda fase da ITC2011

Instância	KHE		GOAL		
	$f(s^*)$	t_s	$f(s^*)$	$f(\bar{s})$	σ
BrazilInstance2	4 / 90	0	1 / 54	1,0 / 63,9	$\pm 0,0 / \pm 6,5$
BrazilInstance3	3 / 240	1	0 / 117	0,0 / 127,8	$\pm 0,0 / \pm 7,9$
BrazilInstance4	39 / 144	1	17 / 92	17,2 / 99,6	$\pm 0,4 / \pm 5,9$
BrazilInstance6	11 / 291	0	4 / 207	4,0 / 223,5	$\pm 0,0 / 10,4$
FinlandElementarySchool	9 / 30	6	0 / 3	0,0 / 4,0	$\pm 0,0 / \pm 0,5$
FinlandSecondarySchool2	2 / 1821	109	0 / 0	0,0 / 0,4	$\pm 0,0 / \pm 0,7$
Aigio1stHighSchool10-11	14 / 757	20	0 / 4	0,0 / 15,3	$\pm 0,0 / \pm 7,9$
Italy-Instance4	39 / 21238	28	0 / 305	0,0 / 658,4	$\pm 0,0 / \pm 280,2$
KosovoInstance1	1333 / 566	152	0 / 4238	14,0 / 6934,4	$\pm 10,7 / \pm 1862,4$
Kottenpark2003	3 / 78440	402	0 / 41479	0,6 / 90195,8	$\pm 0,7 / \pm 17996,0$
Kottenpark2005A	35 / 23677	489	33 / 27929	33,9 / 27480,4	$\pm 0,9 / \pm 2759,2$
Kottenpark2008	63 / 140083	333	25 / 27410	25,7 / 31403,7	$\pm 0,7 / \pm 3969,3$
Kottenpark2009	55 / 211095	229	33 / 159895	36,6 / 154998,5	$\pm 2,1 / \pm 24265,8$
Woodlands2009	19 / 0	22	2 / 10	2,0 / 15,8	$\pm 0,0 / \pm 2,8$
Spanishschool	1 / 4103	17	0 / 642	0,0 / 865,2	$\pm 0,0 / \pm 177,4$
WesternGreeceUniversity3	0 / 30	126	0 / 5	0,0 / 5,6	$\pm 0,0 / \pm 0,5$
WesternGreeceUniversity4	0 / 41	92	0 / 6	0,0 / 7,4	$\pm 0,0 / \pm 1,0$
WesternGreeceUniversity5	17 / 44	50	0 / 0	0,0 / 0,0	$\pm 0,0 / \pm 0,0$

Tabela 4.16: Resultados da segunda fase da ITC2011

Instância	GOAL	HFT	HySST	Lectio
<i>BrazilInstance2</i>	-	3	2	1
<i>BrazilInstance3</i>	-	3	1	2
<i>BrazilInstance4</i>	-	3	1,9	1,1
<i>BrazilInstance6</i>	-	3	2	1
<i>FinlandElementarySchool</i>	1,9	4	2,95	1,15
<i>FinlandSecondarySchool2</i>	1	4	2,4	2,6
<i>Aigio1stHighSchool10-11</i>	1	4	2,8	2,2
<i>Italy_Instance4</i>	1,1	4	3	1,9
<i>KosovaInstance1</i>	1	3	4	2
<i>Kottenpark2003</i>	1,4	4	1,6	3
<i>Kottenpark2005A</i>	1,4	3,6	1,6	3,4
<i>Kottenpark2008</i>	1	4	2,1	2,9
<i>Kottenpark2009</i>	1	4	2	3
<i>Woodlands2009</i>	1,7	4	2,8	1,5
<i>Spanishschool</i>	1	4	2	3
<i>WesternGreeceUniversity3</i>	1	3	2	4
<i>WesternGreeceUniversity4</i>	1	4	2	3
<i>WesternGreeceUniversity5</i>	1	4	2	3
Média	1,18	3,64	2,23	2,32

Terceira fase

A terceira fase da ITC2011 foi similar à primeira; porém, o conjunto de instâncias ocultas foi considerado em detrimento das instâncias públicas e teve-se apenas um mês para gerar as soluções. As melhores soluções de cada instância não foram providas nessa fase. Assim, utilizou-se o mesmo procedimento da Figura 4.1 nessa etapa, porém, ao invés de utilizar a solução inicial do arquivo da instância, a mesma foi gerada pelo resolvidor do KHE. A Tabela 4.17 apresenta os resultados obtidos pela equipe GOAL nessa etapa.

Tabela 4.17: Resultados obtidos pela equipe GOAL na terceira fase da ITC2011

Instância	GOAL $f(s^*)$
<i>BrazilInstance2</i>	0 / 32
<i>BrazilInstance3</i>	0 / 101
<i>BrazilInstance4</i>	1 / 136
<i>BrazilInstance6</i>	0 / 160
<i>FinlandElementarySchool</i>	0 / 3
<i>FinlandSecondarySchool2</i>	0 / 0
<i>Aigio1stHighSchool10-11</i>	0 / 0
<i>Italy_Instance4</i>	0 / 61
<i>KosovaInstance1</i>	0 / 3
<i>Kottenpark2003</i>	0 / 5355
<i>Kottenpark2005A</i>	24 / 13930
<i>Kottenpark2008</i>	8 / 27909
<i>Kottenpark2009</i>	19 / 5565
<i>Woodlands2009</i>	0 / 441
<i>Spanishschool</i>	0 / 12
<i>WesternGreeceUniversity3</i>	0 / 5
<i>WesternGreeceUniversity4</i>	0 / 8
<i>WesternGreeceUniversity5</i>	0 / 0

A Tabela 4.18 apresenta a classificação dos competidores nessa fase. Novamente, a equipe GOAL não pôde competir nas instâncias *BrazilInstance2*, *BrazilInstance3*, *BrazilInstance4* e *BrazilInstance6*. A equipe VAGOS enviou soluções apenas para al-

gumas instâncias, dessa forma, eles receberam a pior classificação nas instâncias em que não submeteram soluções.

Tabela 4.18: Resultados da terceira fase da ITC2011

Instância	GOAL	HFT	HySST	Lectio	VAGOS
<i>BrazilInstance2</i>	-	4	3	2	1
<i>BrazilInstance3</i>	-	4	3	2	1
<i>BrazilInstance4</i>	-	4	3	2	1
<i>BrazilInstance6</i>	-	4	3	1	2
<i>FinlandElementarySchool</i>	2,5	2,5	2,5	2,5	-
<i>FinlandSecondarySchool2</i>	2	4	2	2	-
<i>Aigio1stHighSchool10-11</i>	1	4	3	2	-
<i>Italy_Instance4</i>	2	4	1	3	-
<i>KosovaInstance1</i>	1	3	2	4	-
<i>Kottenpark2003</i>	3	4	2	1	-
<i>Kottenpark2005A</i>	2	3	1	4	-
<i>Kottenpark2008</i>	1	4	2	3	-
<i>Kottenpark2009</i>	2	4	1	3	-
<i>Woodlands2009</i>	1	4	2	3	-
<i>Spanishschool</i>	1	4	3	2	-
<i>WesternGreeceUniversity3</i>	1,5	4	3	5	1,5
<i>WesternGreeceUniversity4</i>	1	3	2	4	-
<i>WesternGreeceUniversity5</i>	2	4	2	5	2
Média	1,64	3,75	2,25	2,75	3,86

4.4 Discussão dos Resultados

Essa abordagem utilizou o resolvidor do KHE para gerar soluções iniciais combinado com uma busca local multi-vizinhança. O resolvidor do KHE foi capaz de encontrar rapidamente soluções boas e a estrutura de vizinhança proposta foi capaz de explorar consistentemente o espaço de soluções e alcançar melhoras significantes na solução inicial conforme mostra a Tabela 4.12.

Pelas Tabelas 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 e 4.11 pode-se observar que todos os métodos propostos foram capazes de resolver satisfatoriamente até as instâncias grandes do problema no pequeno tempo limite fixado. O desvio padrão também foi pequeno na maioria dos casos, evidenciando a robustez dos métodos propostos para o problema.

Pela Tabela 4.12, ao observar o ranking médio final das metaheurísticas, pode-se concluir que, aplicados individualmente, os métodos ILS e SVNS são os mais adequados para o problema abordado. Ambos algoritmos obtiveram um ranking médio de 2,61 unidades e dominaram as demais metaheurísticas, especialmente para as instâncias menores do problema, com menos de 1.000 aulas a alocar.

Ao observar o conjunto de instâncias grandes do problema, com mais de 1.000 aulas a alocar, a metaheurística *Simulated Annealing* obteve um desempenho superior. O algoritmo dominou os demais, especialmente nas instâncias do conjunto *Kottenpark* ao obter 3 de 5 melhores custos médios. Em contrapartida, o método obteve um desempenho abaixo da média para as instâncias menores, como as dos grupos *FinlandSchool* e *WesternGreeceUniversity*. Tal comportamento pode ser explicado pelo fato desse algoritmo possuir muitos parâmetros para se configurar. Assim, uma configuração de parâmetros pode obter bons resultados para instâncias grandes, porém obter uma performance ruim para instâncias pequenas.

O algoritmo RVNS obteve resultados ruins, obtendo o pior ranking médio, 6,78. Tal resultado se justifica pelo fato de que o referido algoritmo não apresenta método de descida em sua composição, o mesmo apenas gera um vizinho de cada movimento k por iteração. Assim, os movimentos são selecionados com igual probabilidade. Consequentemente vizinhanças mais custosas e menos promissoras impedem outras vizinhanças performem movimentos mais promissores.

O algoritmo LAHC apresentou um desempenho competitivo, ao apresentar um ranking médio de 3,38 e superar os resultados de metaheurísticas amplamente conhecidas, como *Simulated Annealing* e *Variable Decomposition Neighborhood Search*. O fato de ter apenas um parâmetro a ajustar, juntamente com o desempenho competitivo e a facilidade de implementação tornam o LAHC uma metaheurística promissora a ser explorada futuramente em outros problemas de otimização.

De modo geral pode-se dizer que o algoritmos *Simulated Annealing* e ILS, utilizados no resolvidor submetido à ITC2011, formam uma boa combinação. Como afirmado anteriormente, o algoritmo *Simulated Annealing* apresentou bons resultados para instâncias com mais de 1.000 alocações; porém, teve um desempenho ruim para instâncias menores.

Essa fraqueza do algoritmo é compensada pela aplicação do algoritmo ILS sobre a melhor solução encontrada pelo *Simulated Annealing*, capaz de melhorar os resultados do último, especialmente nas instâncias pequenas do problema.

Até mesmo encontrar soluções factíveis para as instâncias da ITC2011 é uma tarefa difícil. As mesmas comumente definem a muitas das restrições como fortes (veja as tabelas 4.2 e 4.4). Dessa forma, a ITC2011 não espera que um resolvidor seja capaz de encontrar todas as soluções factíveis e o uso do par *infactibilidade/qualidade* para avaliar soluções é recomendado. Nossa abordagem foi capaz de alcançar treze de dezoito ² soluções factíveis no limite de tempo estipulado.

Na primeira fase da ITC2011, fomos capazes de melhorar quinze de dezesseis melhores soluções conhecidas ³. Infelizmente, apenas uma delas foi a melhor solução encontrada ao fim dessa fase. Para a instância *SouthAfricaLewitt2009* encontramos uma solução com o mesmo custo da melhor solução encontrada nesta fase, porém, não fomos a primeira equipe a encontrá-la. Assim, a equipe VAGOS recebeu o prêmio nessa instância. Os resultados abaixo da média do nosso resolvidor nessa etapa podem ser explicados por dois fatores: (1) nosso resolvidor ainda estava em desenvolvimento durante essa fase e algumas melhorias no resolvidor não chegaram a tempo de melhorar as soluções enviadas; e (2) não tivemos muito tempo para executar os experimentos.

A segunda fase da ITC2011 foi a mais importante da competição, uma vez que todos os resolvidores tiveram sua performance comparada sob as mesmas condições. Como mostra a Tabela 4.16, nessa fase, nosso resolvidor foi o melhor classificado em doze de catorze instâncias ⁴. Fomos ainda competitivos nas duas instâncias em que não vencemos. Assim, a equipe GOAL venceu essa etapa com larga vantagem sobre o segundo colocado. O sucesso da nossa equipe nessa etapa pode ser explicado pelo fato de que procedimentos de busca local são capazes de encontrar boas soluções, até mesmo para instâncias muito grandes, em um curto período de tempo.

Na terceira etapa da ITC2011, conforme a Tabela 4.18, o resolvidor da equipe GOAL foi o melhor classificado em dez de catorze instâncias. Novamente, o nosso resolvidor foi competitivo ainda nas demais instâncias. Assim, vencemos essa etapa também. Nessa fase a versão final do nosso resolvidor estava completa; dessa forma, conseguimos resultados bem melhores que os da primeira etapa.

²Não há sequer garantia de que há uma solução factível para as cinco instâncias restantes.

³As cinco instâncias restantes já possuem uma solução de custo zero conhecida.

⁴Excluindo as instâncias brasileiras, nas quais não pudemos competir.

Vale ressaltar ainda a contribuição dos demais finalistas da ITC2011 ao propor outras técnicas para o problema. Ocupando o quarto lugar, a equipe HTF (Domros and Homberger 2012) desenvolveu uma abordagem evolucionária para o problema. Em terceiro lugar, a equipe Lectio (Sorensen, Kristiansen and Stidsen 2012) desenvolveu uma abordagem baseada em Busca Adaptativa em Larga Vizinhança (BALV), utilizando-se de um framework de BALV previamente desenvolvido para solução de problemas combinatoriais. O segundo lugar foi ocupado pela equipe HySTT (Kheiri, Ozcan and Parkes 2012) que aplicou estratégias de busca hiper-heurística sobre soluções iniciais geradas pelo KHE. A equipe VAGOS contribuiu com algumas boas soluções nas primeira e terceira etapas da competição, porém não participou da principal fase da mesma.

Capítulo 5

Considerações Finais

O principal objetivo do presente trabalho foi alcançado, uma vez que foram apresentadas abordagens eficientes baseadas em busca local para o Problema da Programação de Horários Escolares proposto pela *Third International Timetabling Competition*. O resolvidor desenvolvido é capaz de processar qualquer instância modelada no formato XHSTT. Assim, devido ao poder de modelagem do formato, o resolvidor desenvolvido pode inclusive solucionar instâncias de outros problemas no contexto do agendamento de horários modeladas em XHSTT (Post, Kingston, Ahmadi, Daskalaki, Gogos, Kyngas, Nurmi, Musliu, Pillay, Santos and Schaerf 2011). Os algoritmos implementados foram validados sobre uma vasta variedade de instâncias reais do problema, sendo avaliado pela ITC2011 como o melhor resolvidor da competição.

As implementações propostas das metaheurísticas foram capazes encontrar soluções factíveis para treze de dezoito instâncias e melhorar quinze de dezesseis melhores soluções conhecidas. Na segunda fase da competição fomos a equipe melhor classificada em doze de catorze instâncias e na terceira fase, vencemos dez de catorze instâncias. Tais resultados reafirmam a supremacia do resolvidor da equipe GOAL na *Third International Timetabling Competition*.

O sucesso da equipe GOAL na ITC2011 se deve principalmente a três fatores: (1) a variedade de movimentos de vizinhança, capazes de explorar consistentemente o espaço de busca e realizar melhorias significativas nas soluções iniciais, (2) a aplicação efetiva da busca local através de uma implementação customizada dos algoritmos *Simulated Annealing* e ILS e (3) a eficiência da plataforma KHE para gerar soluções iniciais (veja a Tabela 4.15).

Os resultados apresentados no presente trabalho, juntamente com os resultados das últimas competições internacionais de programação de horários, apontam que métodos de busca local são a abordagem mais consistente para a referida classe de problemas. Nesse contexto, destaca-se o algoritmo *Simulated Annealing*, que compôs o resolvidor de todos os vencedores das competições. Dentre as metaheurísticas implementadas, ao serem executadas separadamente, os algoritmos ILS e SVND apresentaram os melhores resultados.

5.1 Trabalhos Futuros

Ainda há espaço para melhorias a serem feitas na abordagem proposta. Algumas sugestões de trabalhos futuros são:

1. desenvolver novas estruturas de vizinhança capazes de realizar alterações maiores em uma solução, como *Ejection Chains* (Lú and Hao 2010);
2. desenvolver e realizar um estudo computacional de outras metaheurísticas como Algoritmos Genéticos e Busca Tabu para o problema;
3. realizar um estudo mais profundo acerca da configuração de parâmetros para o resolvidor.

Outro possível trabalho futuro é o desenvolvimento de uma interface gráfica com o usuário para o software, de modo a permitir que escolas e universidades de todo o mundo possam produzir suas instâncias do problema e solucioná-las com o presente resolvidor.

Apêndice A

Apêndices

A.1 Publicações

Neste apêndice são listados os trabalhos publicados em periódicos ou apresentados em eventos científicos desenvolvidos durante o período de realização da presente pesquisa.

1. FONSECA, G. H. G. ; SANTOS, H. G. ; BRITO, S. S. . **A Simulated Annealing based Approach to the High School Timetabling Problem**. In: *The 13th International Conference on Intelligent Data Engineering and Automated Learning*, 2012, Natal, Brazil. Intelligent Data Engineering and Automated Learning - IDEAL 2012. Berlin: Springer Berlin Heidelberg, 2012. v. 7435. p. 540-549.
2. FONSECA, G. H. G. ; SANTOS, H. G. ; TOFFOLO, T. A. M. ; BRITO, S. S. . **A SA-ILS approach for the High School Timetabling Problem**. In: *The 9th International Conference on the Practice and Theory of Automated Timetabling*, 2012, Son, Norway. The 9th International Conference on the Practice and Theory of Automated Timetabling, 2012.
3. SANTOS, H. G. ; TOFFOLO, T. A. M. ; BRITO, S. S. ; FONSECA, G. H. G. . **Towards Generic MIP solvers for Rostering and Timetabling**. In: *Fourth International Workshop on Model-based Metaheuristics*, 2012, Angra dos Reis, Brazil.
4. FONSECA, G. H. G. ; BRITO, S. S. ; TOFFOLO, T. A. M. ; SANTOS, H. G. . **Técnicas de Busca Local para o Problema da Programação de Horários**

Escolares. In: *XLIV Simpósio Brasileiro de Pesquisa Operacional*, 2012, Rio de Janeiro, Brasil.

5. BRITO, S. S. ; FONSECA, G. H. G. ; TOFFOLO, T. A. M. ; SANTOS, H. G. . **A SA-VNS approach for the High School Timetabling Problem.** In: *EURO Mini Conference XXVIII on Variable Neighbourhood Search*, 2012, Belgrade, Montenegro.
6. BRITO, S. S. ; FONSECA, G. H. G. ; TOFFOLO, T. A. M. ; SANTOS, H. G. ; SOUZA, M. J. F. . **A SA-VNS approach for the High School Timetabling Problem.** *Electronic Notes in Discrete Mathematics*, v. 39, p. 169, 2012.

Os trabalhos a seguir foram submetidos para periódicos ou eventos científicos e aguardam resposta dos editores.

1. FONSECA, G. H. G. ; SANTOS, H. G. . **A Computational Study about VNS based methods applied to the High School Timetabling Problem.** Submetido para *Computers & Operations Research*, 2013.
2. SANTOS, H. G. ; TOFFOLO, T. A. M. ; BRITO, S. S. ; FONSECA, G. H. G. . **Towards Generic MIP solvers for Rostering and Timetabling.** Submetido para *International Transactions in Operations Research*, 2013.
3. FONSECA, G. H. G. ; SANTOS, H. G. ; TOFFOLO, T. A. M. ; BRITO, S. S. . **International Timetabling Competition 2011: Goal Team Solver Description.** Submetido para *Annals of Operations Research*, 2013.
4. FONSECA, G. H. G. ; SANTOS, H. G. . **Memetic Algorithms to the High School Timetabling Problem.** Submetido para *IEEE Congress on Evolutionary Computation*, 2013.
5. FONSECA, G. H. G. ; SANTOS, H. G. ; TOFFOLO, T. A. M. . **Late Acceptance Hill-Climbing Applied to the High School Timetabling Problem.** Submetido para *Annals of Operations Research*, 2013.
6. FONSECA, G. H. G. ; SANTOS, H. G. . **A New Formulation to the Examination Timetabling Problem.** Submetido para *Multidisciplinary International Scheduling Conference: Theory & Applications*, 2013.

Referências Bibliográficas

- Abdullah, S., Shaker, K., McCollum, B. and McMullan, P.: 2010, Incorporating great deluge with kempe chain neighbourhood structure for the enrolment-based course timetabling problem, *in* J. Yu, S. Greco, P. Lingras, G. Wang and A. Skowron (eds), *Rough Set and Knowledge Technology*, Vol. 6401 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 70–77.
URL: http://dx.doi.org/10.1007/978-3-642-16248-0_5
- Abramson, D.: 1991, Constructing school timetables using simulated annealing: Sequential and parallel algorithms, *Management Science*, Vol. 37, pp. 98–113.
URL: <http://mansci.journal.informs.org/content/37/1/98.abstract>
- Abuhamdah, A.: 2010, Experimental result of late acceptance randomized descent algorithm for solving course timetabling problems, *IJCSNS- International Journal of Computer Science and Network Security*, Vol. 10 No. 1, January.
- Barbosa, S. H. and Souza, S. R.: 2011, Resolução do problema de programação de cursos universitários baseada em currículos via uma meta-heurística híbrida grasp-ils-relaxado, *XLIII Simpósio Brasileiro de Pesquisa Operacional, Ubatuba, Proceedings of XLIII SBPO, Ubatuba : SOBRAPO*, pp. 1 : 2827–2882.
- Birbas, T., Daskalaki, S. and Housos, E.: 1997, Timetabling for greek high schools., *Journal of the Operational Research Society*, Vol. 48, Palgrave Macmillan, Basingstoke, pp. 1191–1200.
- Burke, E. K. and Bykov, Y.: 2008, A Late Acceptance Strategy in Hill-Climbing for Exam Timetabling Problems, *PATAT '08 Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling*.
URL: http://w1.cirrelt.ca/~patat2008/PATAT_7-PROCEEDINGS/Papers/Bykov-HC2a.pdf

- Burke, E. K. and Bykov, Y.: 2012, The late acceptance hill-climbing heuristic, *Department of Computing Science and Mathematics, University of Stirling*, number CSM-192, p. 19.
- de Haan, P., Landman, R., Post, G. and Ruizenaar, H.: 2007, A case study for timetabling in a Dutch secondary school, *Lecture notes in computer science: VI Practice and theory of automated timetabling. Berlin : Springer*, pp. 3867 : 267–279.
- Domros, J. and Homberger, J.: 2012, *An Evolutionary Algorithm for High School Timetabling*, PATAT '12 Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling.
- Fonseca, G., Santos, H., Toffolo, T., Brito, S. and Souza, M.: 2012, *A SA-ILS approach for the High School Timetabling Problem*, PATAT '12 Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling.
- Garey, M. R. and Johnson, D. S.: 1979, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, USA.
- Glover, F. and Kochenberger, G.: 2003, *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Kluwer Academic Publishers.
URL: http://books.google.com.br/books?id=O_10T_KeqOgC
- Gotlieb, C. C.: 1963, The construction of class-teacher time-tables, *Proc. IFIP Congress, Munich, North Holland Pub. Co.*, pp. 73–77.
- Hansen, P. and Mladenović, N.: 1999, Variable neighborhood search: Methods and recent applications, *In: Proceedings of the Third Metaheuristics International Conference, Angra dos Reis, Brazil*, pp. 275–280.
- Hansen, P. and Mladenovic, N.: 2000, *Variable Neighborhood Search: A Chapter of Handbook of Applied Optimization.*, Les Cahiers du GERAD G-2000-3. Montreal, Canada, chapter 8.
- Hansen, P., Mladenović, N. and Perez-Britos, D.: 2001, *Variable Neighborhood Decomposition Search*, Vol. 7 of *Journal of Heuristics*, Kluwer Academic Publishers, Hingham, MA, USA, pp. 335–350.
URL: <http://dx.doi.org/10.1023/A:1011336210885>

- IDSIA: 2012, International Timetabling Competition 2002. Available at <http://www.idsia.ch/Files/ttcomp2002/>, Acessado em dezembro / 2012.
- Kheiri, A., Ozcan, E. and Parkes, A. J.: 2012, *HySST: Hyper-heuristic Search Strategies and Timetabling*, PATAT '12 Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling.
- Kingston, J. H.: 2005, A tiling algorithm for high school timetabling, *Lecture notes in computer science: V Practice and theory of automated timetabling*. Berlin: Springer, pp. 3616 : 208–225.
- Kingston, J. H.: 2006, Hierarchical timetable construction, *Problems, Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling*.
- Kingston, J. H.: 2012a, High School Timetable Data Format Specification, <http://sydney.edu.au/engineering/it/~jeff/hseval.cgi?op=spec>, Acessado em junho de 2012.
- Kingston, J. H.: 2012b, A Software Library for School Timetabling, Disponível em <http://sydney.edu.au/engineering/it/~jeff/khe/>, Acessado em abril de 2012.
- Kirkpatrick, S., Gellat, D. C. and Vecchi, M. P.: 1983, Otimization by Simulated Annealing, *Science*, pp. 202, 671–680.
- Kostuch, P.: 2005, The university course timetabling problem with a three-phase approach, *Proceedings of the 5th international conference on Practice and Theory of Automated Timetabling*, PATAT'04, Springer-Verlag, Berlin, Heidelberg, pp. 109–125.
URL: <http://dx.doi.org/10.1007/115935777>
- Lin, S. and Kernighan, B. W.: 1973, An effective heuristic algorithm for the travelling-salesman problem, *Operations Research*, Vol. 21, pp. 498–516.
- Lourenco, H. R., Martin, O. C. and Stutzle, T.: 2001, Iterated local search, in F. Glover and G. Kochenberger (eds), *Social Science Research Network*, Vol. 57, Kluwer Academic Publishers, p. 49.
URL: <http://arxiv.org/abs/math/0102188>
- Lú, Z. and Hao, J.-K.: 2010, Adaptive tabu search for course timetabling., *European Journal of Operational Research*, Vol. 200, pp. 235–244.
URL: <http://dblp.uni-trier.de/db/journals/eor/eor200.html#LuH10>

- McCollum, B.: 2012, International Timetabling Competition 2007. Available at <http://www.cs.qub.ac.uk/itc2007/>, Acessado em dezembro / 2012.
- Mladenovic, N. and Hansen, P.: 1997, Variable neighborhood search, *Computers and Operations Research*, pp. 24, 1097–1100.
- Muller, T.: 2009, ITC2007 solver description: a hybrid approach., *Annals OR*, Vol. 172, pp. 429–446.
URL: <http://dblp.uni-trier.de/db/journals/anor/anor172.htmlMuller09>
- Nurmi, K. and Kyngas, J.: 2007, A framework for school timetabling problem, *Proceedings of the 3rd multidisciplinary international scheduling conference: theory and applications, Paris*, pp. 386–393.
- of Twente, U.: 2012, International Timetabling Competition 2012. Available at <http://www.utwente.nl/ctit/hstt/itc2011/welcome/>, Acessado em dezembro / 2012.
- Özcan, E., Bykov, Y., Birben, M. and Burke, E. K.: 2009, Examination timetabling using late acceptance hyper-heuristics, *Proceedings of the Eleventh conference on Congress on Evolutionary Computation, CEC'09*, IEEE Press, Piscataway, NJ, USA, pp. 997–1004.
URL: <http://dl.acm.org/citation.cfm?id=1689599.1689731>
- Post, G., Ahmadi, S., Daskalaki, S., Kingston, J. H., Kyngas, J., Nurmi, C. and Ranson, D.: 2010, An XML format for benchmarks in High School Timetabling, *Annals of Operations Research DOI 10.1007/s10479-010-0699-9*, pp. 3867 : 267–279.
- Post, G., di Gaspero, L., Kingston, J. H., McCollum, B. and Schaerf, A.: 2012, The third international timetabling competition, *Proceedings of the 9th International Conference on Practice and Theory of Automated Timetabling*, Son, Norway.
- Post, G., Kingston, J., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H. and Schaerf, A.: 2011, XHSTT: an XML archive for high school timetabling problems in different countries, *Annals of Operations Research* p. 17. 10.1007/s10479-011-1012-2.
URL: <http://dx.doi.org/10.1007/s10479-011-1012-2>
- Russell, S. J. and Norvig, P.: 2002, *Artificial Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall, pp. 111–114.
URL: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0137903952>

- Santos, H. G.: 2006, *Formulações e Algoritmos para o Problema da Programação de Horários em Escolas*, Tese (Doutorado em Computação), Universidade Federal Fluminense, Niteroi.
- Santos, H. G. and Souza, M. J. F.: 2007, Programação de horários em instituições educacionais: formulações e algoritmos, *XXXIX Simpósio Brasileiro de Pesquisa Operacional, Fortaleza. Anais do XXXIX SBPO. Rio de Janeiro : SOBRAPO*, pp. 1 : 2827–2882.
- Santos, H. G., Uchoa, E., Ochi, L. S. and Maculan, N.: 2012, Strong bounds with cut and column generation for class-teacher timetabling, *Annals OR*, Vol. 194, pp. 399–412.
- Sorensen, M., Kristiansen, S. and Stidsen, T. R.: 2012, *International Timetabling Competition 2011: An Adaptive Large Neighborhood Search algorithm*, PATAT '12 Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling.
- Tuga, M., Berretta, R. and Mendes, A.: 2007, A hybrid simulated annealing with kempe chain neighborhood for the university timetabling problem, *6th Annual IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), 11-13 July 2007, Melbourne, Australia*, IEEE Computer Society, pp. 400–405.
- Valourix, C. and Housos, E.: 2003, Constraint programming approach for school timetabling, *Computers & Operations Research*, pp. 30 : 1555–1572.
- Verstichel, J. and Vanden Berghe, G.: 2009, A late acceptance algorithm for the lock scheduling problem, in S. Voss, J. Pahl and S. Schwarze (eds), *Logistik Management, Hamburg, 2-4 September 2009*, Springer.
URL: <https://lirias.kuleuven.be/handle/123456789/249443>
- W3C: Acessado em junho de 2012, Extensible markup language (xml), <http://www.w3.org/XML/>.
- Willemen, R.: 2002, *School Timetable Construction: Algorithms and Complexity*, Technische Universiteit Eindhoven.
URL: <http://books.google.com.br/books?id=y7VxuAAACAAJ>
- Wright, M.: 1996, School timetabling using heuristic search, *Journal of Operational Research Society*, pp. 47 : 347–357.