UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

RENAN COSTALONGA MONTEIRO

ALGORITMO HÍBRIDO ITERATED LOCAL SEARCH E SIMULATED ANNEALING PARA O PROBLEMA DE TABELA-HORÁRIO DE UNIVERSIDADES

ALEGRE
2016

RENAN COSTALONGA MONTEIRO

ALGORITMO HÍBRIDO ITERATED LOCAL SEARCH E SIMULATED ANNEALING PARA O PROBLEMA DE TABELA-HORÁRIO DE UNIVERSIDADES

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. MSc. Edmar Hell Kampke.

ALEGRE

2016

RENAN COSTALONGA MONTEIRO

ALGORITMO HÍBRIDO ITERATED LOCAL SEARCH E SIMULATED ANNEALING PARA O PROBLEMA DE TABELA-HORÁRIO DE UNIVERSIDADES

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em ____ de novembro de 2016.

COMISSÃO EXAMINADORA

Prof. MSc. Edmar Hell Kampke.
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

Prof. MSc. Dayan de Castro Bissoli
Universidade Federal do Espírito Santo

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida, e por nunca me desamparar durante todo o caminho percorrido até aqui.

Aos meus pais e meu irmão João Pedro, por todo amor, dedicação e incentivo na realização dos meus sonhos. Nada seria possível sem o apoio de vocês.

A toda a minha família que sempre esteve ao meu lado durante toda a vida, em especial, aos meus avós Arlindo, Neli e Luís, que mesmo não estando presente em vida, estarão sempre em meu coração.

A todos os meus professores da graduação, por todo o empenho e inspiração durante as aulas, tudo que sei hoje, é por causa de vocês.

Ao meu amigo e orientador Edmar, por toda paciência, dedicação e suporte necessário na realização desse trabalho.

A todos os amigos que fiz em Alegre, em especial, a turma de 2012/2. O caminho se tornou mais fácil com a alegria e companheirismo de vocês.

Aos meus companheiros de república Otto e Romerson, por todos os bons momentos e dificuldades que enfrentamos juntos durante essa jornada.

Por fim, a toda a equipe do Labotim, por permitir a utilização da infraestrutura necessária para a conclusão deste trabalho.

RESUMO

O problema de tabela-horário educacional consiste em alocar um conjunto de aulas em salas disponíveis e períodos de tempo pré-determinados, considerando alunos e professores, e satisfazendo algumas restrições. Encontram-se na literatura diversos modelos matemáticos propostos para esse problema. Nesse trabalho, é adotado o modelo voltado para universidades, baseado em currículo de cursos, proposto no segundo campeonato internacional de tabela-horário, *Internacional Timetabling Competition*, realizado em 2007 (ITC-2007). Para a solução do problema, é utilizada a meta-heurística *Iterated Local Search* - ILS com a meta-heurística *Simulated Annealing* como método de busca local, e Cadeia de Kempe como movimento de perturbação. Testes computacionais com instâncias disponibilizadas pelo ITC-2007 foram realizados, e os resultados obtidos comparados com as melhores soluções obtidas no ITC-2007 e com as soluções apresentadas em outro trabalho da literatura.

Palavras-Chave: Tabela-horário para Universidades, *Iterated Local Search*, Cadeia de Kempe, *Simulated Annealing*, ITC-2007.

ABSTRACT

The educational timetabling problem consists of allocating a set of classes to available classrooms and predetermined times, taking into consideration students and teachers, and complying with the existing constraints. Several mathematical models proposed for this problem can be found in the literature. In this work, the university-based model, which was proposed on the second international timetabling competition (ITC-2007), was chosen. To solve the problem, in this work, the used approach is the Iterated Local Search - ILS metaheuristic with Simulated Annealing as local search and Kempe Chain as perturbation phase. Computational tests with instances available by the ITC-2007 were performed using the ILS, and then the results were compared with the best solutions obtained on ITC-2007 and with the solutions available in other work in the literature.

Keywords: Univesity Timetaling, Iterated Local Search, Kempe Chain, Simulated Annealing, ITC-2007.

LISTA DE FIGURAS

Figura 1: Algoritmo genérico do ILS. 16
Figura 2: Algoritmo da Solução Inicial proposto por Lu e Hao (2010). 18
Figura 3: Exemplo de movimento do tipo *Move*. 19
Figura 4: Exemplo de movimento do tipo *Swap*. 19
Figura 5: Exemplo de movimento do tipo *Time-Move*. 19
Figura 6: Exemplo de movimento do tipo *Room-Move*. 20
Figura 7: Algoritmo do SA apresentado por Rocha (2013). 21
Figura 8: Execução do SA para fase de busca local. 22
Figura 9: Exemplo de Cadeias de Kempe nos horários t_1 e t_2. 23
Figura 10: Exemplo de um movimento de Cadeia de Kempe nos horários t_1 e t_2. 24

LISTA DE TABELAS

Tabela 1: Parâmetros de valores do ILS. 25
Tabela 2: Melhores soluções obtidas pelo algoritmo. 26
Tabela 3: Resultados da primeira fase do ITC-2007. 28
Tabela 4: Resultados obtidos na segunda fase do ITC-2007. 29

SUMÁRIO

1. Introdução 10
1.1. O problema e sua importância 10
1.2. Objetivos deste trabalho 13
1.2.1. Objetivos Gerais 13
1.2.2. Objetivos Específicos 13
2. Revisão de Literatura 14
3. Metodologia 15
3.1. Modelagem do Problema 15
3.2. Iterated Local Search 16
3.3. Solução Inicial 16
3.4. Busca Local 18
3.5. Perturbação 22
3.6. Critério de Aceitação 24
4. Resultados Computacionais 25
4.1. Escolha dos Parâmetros 25
4.2. Detalhes de Implementação 26
4.3. Análise dos Resultados 26
5. Conclusões 30
6. Referências Bibliográficas 31

1. INTRODUÇÃO

Os problemas de programação (*scheduling*) são problemas clássicos da área de otimização combinatória, pois lidam com a alocação de recursos, respeitando algumas restrições e maximizando (ou minimizando) uma função matemática. Um importante subconjunto de problemas desta área é composto pelos Problemas de Tabela-Horário (PTH), já que dispõem de diversas aplicações como, por exemplo, criação de escala de trabalho de funcionários em uma indústria e agendamento de partidas em campeonatos esportivos.

Existem muitos estudos relacionados ao PTH de escolas e universidades, visto que a elaboração manual destas tabelas-horário não é uma tarefa trivial. Além disso, as instituições de ensino precisam fazê-la, frequentemente. É importante realçar que nem sempre a alocação manual é completamente aceitável, ou mesmo válida, de modo que é difícil atender todos os desejos das partes envolvidas e respeitar todas as restrições impostas.

Por isso, um tratamento especial tem sido dado a resolução automática de tabela-horário. Com Gotlieb (1962) esse problema ganhou grande destaque na área de otimização combinatória, sendo que a partir disso, diversos trabalhos foram realizados (SCHAERF, 1995; LEWIS 2008).

Como descrito por Schaerf (1995) e Souza (2000), os PTH da área educacional podem ser especificados em três categorias: PTH de Escolas, PTH de Universidades e PTH de Exames. No Problema de Tabela de Horários de Universidades (PTHU), o problema consiste em alocar uma sequência de encontros entre professores e alunos em um período prefixado de tempo, satisfazendo a um conjunto de restrições (SOUZA, 2000). O objetivo é alocar um conjunto de aulas em um número pré-determinado de horários, satisfazendo diversas restrições envolvendo professores, alunos e o espaço físico disponível (SANTOS, 2007).

1.1. O problema e sua importância

O PTHU possui uma das mais altas complexidades dá área de otimização combinatória e esta aumenta à medida em que são adicionadas novas restrições. Segundo Schaerf (1995) o problema é classificado como NP-Completo para a maioria das formulações, e assim, uma solução exata só pode ser garantida para instâncias pequenas, o que não corresponde à realidade da maioria das instituições de ensino.

Com objetivo de incentivar o estudo de diferentes abordagens sobre o PTH, o PATAT - *The International Series of Conferences on the Practice and Theory of Automated Timetabling* promoveu três competições (2002, 2007 e 2011) de tabela-horário educacional, chamadas de *International Timetabling Competition* - ITC. Cada edição do ITC abordou um tema diferente: O ITC-2002 abordou o PTH de Exames Finais, em que exames que possuem alunos em comum não podem ser alocados no mesmo horário, e o objetivo é utilizar o menor número de horários possíveis, já o ITC-2007 abordou o PTHU e o ITC-2011 abordou o PTH de Escolas de Ensino Médio, que se diferencia do PTH, pois cada turma permanece em uma sala fixa.

Portanto, o ITC-2007 foi o último campeonato a abordar o PTHU. Nesse caso, houve três diferentes formulações, sendo que os competidores podiam concorrer independentemente em cada uma delas. Cada formulação tem seu próprio conjunto de instâncias. A primeira formulação é específica para a aplicação de exames finais das disciplinas de uma universidade. A segunda e terceira formulação tratam da alocação semanal das aulas de uma universidade. A diferença básica entre elas é que a segunda formulação (chamada de *post enrolment*) trata o problema considerando dados de matrícula dos alunos, enquanto a terceira formulação (chamada de *curriculum*) leva em conta os currículos que compõe os cursos da universidade. Currículos são definidos como os grupos de disciplinas que têm alunos em comum e não podem ser alocadas no mesmo horário.

O ITC-2007 incentivou um debate mais abrangente entre a comunidade que pesquisa problemas de tabela-horário. Muitas técnicas de resolução têm sido implementadas e/ou aprimoradas após o campeonato (ROCHA, 2013). Outra vantagem do ITC é a disponibilização de instâncias próximas à realidade das universidades, permitindo assim a comparação de resultados entre as diferentes técnicas de resolução.

Para o desenvolvimento desse trabalho, foi escolhida a terceira formulação estabelecida pelo ITC-2007, conforme descrito em PATAT (2008). A principal razão dessa escolha é facilitar a comparação com os resultados obtidos por outros algoritmos propostos na literatura.

Nessas instâncias do ITC-2007, e no PTHU de forma geral, para encontrar a melhor solução possível, é desejável alcançar a satisfação de todas as restrições. Entretanto, nem sempre é possível atendê-las totalmente. Assim, as restrições estabelecidas são classificadas de acordo com sua importância. Conforme Santos e Souza (2007), frequentemente são utilizados dois grupos para classificar as restrições:

- **Restrições Fortes**: Devem ser obedecidas a qualquer custo. O não atendimento de alguma dessas restrições inviabiliza a solução.

- **Restrições Fracas**: A satisfação dessas restrições é desejável, porém, o não atendimento não inviabiliza a solução.

Conforme descrito em PATAT (2008), na terceira formulação do ITC-2007, as restrições são descritas como segue:

Restrições Fortes (RFt):

- **Aulas**: Todas as aulas das disciplinas devem ser alocadas em períodos diferentes. Uma violação ocorre se uma aula não é alocada. (RFt1)
- **Conflitos**: Aulas de disciplinas do mesmo currículo ou lecionadas pelo mesmo professor devem ser alocadas em períodos diferentes. (RFt2)
- **Ocupação de Sala**: Duas aulas não podem ocupar uma sala no mesmo horário. (RFt3)
- **Disponibilidade**: Uma aula não pode ser alocada num horário em que a disciplina é indisponível. (RFt4)

Restrições Fracas (RFc):

- **Dias Mínimos de Trabalho**: As aulas de cada disciplina devem ser espalhadas por uma quantidade mínima de dias. Cada dia abaixo do mínimo é contado como uma violação. (RFc1)
- **Aulas Isoladas**: Aulas do mesmo currículo devem ser alocadas em períodos adjacentes. Cada aula isolada é contada como uma violação. (RFc2)
- **Capacidade da Sala**: O número de alunos da disciplina deve ser menor ou igual ao número de assentos da sala em que a aula for alocada. Cada aluno excedente contabiliza uma violação. (RFc3)
- **Estabilidade de Sala**: Todas as aulas de uma disciplina devem ser alocadas na mesma sala. Cada sala distinta é contada como uma violação. (RFc4)

Conforme dito anteriormente, devido à dificuldade de resolver o problema e por causa de sua importância, existe a necessidade de propor algoritmos com diferentes abordagens, que produzam soluções satisfatórias para o problema em tempo viável, independente do tamanho da instância. A partir disso, é desejável verificar o

desempenho da meta-heurística *Iterated Local Search* (ILS) com o objetivo de demonstrar a eficiência dos resultados apresentados pelo método, quando aplicado ao PTHU.

1.2. Objetivos deste trabalho

1.2.1. Objetivos Gerais

Aplicar a meta-heurística ILS utilizando *Simulated Annealing* (SA) como método de busca local para refinamento da solução e Cadeia de Kempe como movimento de perturbação, na terceira formulação do Problema de Tabela-Horário de Universidades proposto pelo ITC-2007.

1.2.2. Objetivos Específicos

1. Elaborar as estruturas computacionais que serão utilizadas para a resolução do problema, de acordo com os dados contidos nas instâncias disponíveis na terceira formulação do ITC-2007.
2. Implementar um método de construção de uma solução inicial.
3. Desenvolver os movimentos utilizados na geração de soluções vizinhas.
4. Aplicar a meta-heurística *Simulated Annealing* como método de busca local.
5. Estudar o método Cadeia de Kempe como procedimento de perturbação.
6. Implementar o método *Iterated Local Search*.
7. Aplicar o método proposto (ILS+SA) para solucionar o PTHU com base nas instâncias do ITC-2007.
8. Avaliar o desempenho do método por meio de métricas computacionais comparando as soluções encontradas com as melhores soluções obtidas no ITC-2007 e com as soluções disponíveis em outro trabalho da literatura.

2. REVISÃO DE LITERATURA

Segundo Rocha (2013), o problema de tabela-horário de instituições de ensino foi estudado inicialmente na década de 60, quando Gotlieb (1962) iniciou estudos nesta área com o objetivo de apresentar resoluções por meio de métodos automáticos de geração da Tabela-Horário. Desde então, o tema ganhou relevância (SCHAERF 1995; LEWIS 2008).

O problema não possui formulação única, como pode ser observado em Schaerf (1995). Ao longo do tempo surgiram diversas modelagens. Essa variedade surgiu pelo fato das restrições do problema serem específicas a realidade de cada instituição de ensino.

Muller (2009) resolve o PTHU do ITC-2007 utilizando *Conflict-based Statistics* para gerar a solução inicial e *Hill Climbing* combinado com *Great Deluge* e *Simulated Annealing* para refinamento da solução.

Rocha (2013) trata especificamente de tabela-horário de universidades adotando a formulação do ITC-2007. O problema é resolvido com a meta-heurística *Greedy Randomized Adaptive Search Procedure* - GRASP, sendo testados os métodos *Hill Climbing* e *Simulated Annealing* como métodos de busca local, o método *Path-Relinking* também é utilizado, mas para intensificar a busca por soluções de boa qualidade.

Segatto et al. (2015) deu continuidade ao trabalho de Rocha (2013), aplicando Cadeia de Kempe como um novo movimento para a geração de vizinhos na busca local.

Kiefer et al. (2016) utiliza *Adaptive Large Neighborhood Search* para a resolução do PTHU formulado pelo ITC-2007. O método tem como base a destruição de uma solução, seguida da reconstrução de uma parte da solução afim de gerar novos vizinhos. Com isso, foram obtidos os melhores resultados até então em 5 das 21 instâncias da competição.

Os trabalhos aqui citados são apenas uma parte dos diversos trabalhos que abordam o tema, outros trabalhos propuseram diferentes metodologias na resolução do problema.

3. METODOLOGIA

Esse capítulo apresenta a metodologia abordada para o desenvolvimento deste trabalho.

3.1. Modelagem do Problema

A modelagem do problema foi baseada nas instâncias utilizadas pelo ITC-2007. Assim, cada instância possui as seguintes informações:

- **Dias, Horários e Períodos:** É dado o número de dias na semana em que existirão aulas e é estabelecido o número fixo de períodos de aulas em cada dia. Um horário é composto por um dia e um período. O total de horários disponíveis em uma semana é calculado multiplicando o número de dias pela quantidade de períodos no dia.
- **Disciplinas e Professores:** Cada disciplina possui uma quantidade de aulas semanais que devem ser alocadas em períodos diferentes, dados do professor que a leciona e o total de alunos que a assiste. Um número mínimo de dias é determinado para a distribuição das aulas em uma semana, e é permitido que um professor lecione mais de uma disciplina.
- **Salas:** Cada sala possui sua capacidade de assentos.
- **Currículo:** Um currículo é composto por um grupo de disciplinas que possuem alunos em comum.
- **Indisponibilidades:** Períodos que são indisponíveis para determinadas disciplinas.

Cada solução possui um valor associado que representa sua qualidade, esse valor é chamado de função objetivo (*f*). Cada restrição infringida aumenta o valor da função objetivo de acordo com seu peso. A melhor tabela-horário para o problema é aquela que minimiza o valor da função objetivo, dada pela formula:

$$f = \mathrm{RFt} + \mathrm{RFc} \tag{1}$$

Em que RFt = |RFt1| + |RFt2| + |RFt3| + |RFt4| e RFc = 5 * |RFc1| + 2 * |RFc2| + |RFc3| + |RFc4|. O símbolo |.| representa a quantidade de violações de cada restrição. Uma solução válida é aquela que não possui nenhuma violação em restrições fortes, ou seja, RFt = 0. Vale

ressaltar que a função objetivo *f* e os pesos das restrições foram definidos na formulação utilizada no ITC-2007.

3.2. Iterated Local Search

A meta-heurística *Iterated Local Search* - ILS foi introduzida por Lourenço et al. (2003) partindo da ideia que um procedimento de busca local pode ser melhorado gerando-se novas soluções iniciais, as quais são obtidas por meio de perturbações na solução ótima local.

Na Figura 1, é apresentado o pseudocódigo genérico da meta-heurística ILS.

Entrada: ILS_{max}
Saída: Solução *S*
1. $S \leftarrow$ *Solucao Inicial Qualquer*;
2. $S \leftarrow$ *Busca-Local(S);*
3. **para** $i \leftarrow 1$ **até** ILS_{max} **faça**
4. $S_2 \leftarrow$ *Pertubacao(S);*
5. $S_2 \leftarrow$ *Busca-Local(S_2);*
6. $S \leftarrow$ *Aceitacao(S, S_2);*
7. **fim para**
8. **retorne** *S*

Figura 1: Algoritmo genérico do ILS.

O ILS é um procedimento que inicia com a construção de uma solução inicial qualquer, que em seguida é submetida a uma busca local, retornando possivelmente uma solução de melhor qualidade, que corresponde a uma solução ótima local. Em seguida, o método inicia a fase iterativa, sendo executados três componentes em cada iteração: perturbação, busca local e aceitação. A seguir, cada elemento do método ILS é descrito com maiores detalhes.

3.3. Solução Inicial

A construção inicial de uma solução é realizada de forma gulosa, isto é, iterativamente sempre é alocada a aula que possui maior dificuldade em ser alocada, até que todas as aulas sejam alocadas. Essa estratégia foi definida por Lu e Hao (2010) para produzir soluções de melhor qualidade. A construção tem como objetivo produzir uma tabela-horário válida e que minimize as restrições fracas. Essa estratégia foi adotada para que as fases seguintes do

algoritmo se concentrem em encontrar soluções que minimizem ainda mais as violações das restrições fracas.

A etapa de construção parte de uma tabela-horário (X) inicialmente vazia. Em seguida, são alocadas as aulas uma a uma até que ao final, todas estejam alocadas em X. Assim, cada iteração é composta por duas operações distintas: primeiro seleciona-se uma aula, entre as aulas ainda não alocadas, e logo após é escolhido um horário e sala disponíveis em X para alocar a aula selecionada anteriormente. As escolhas são feitas considerando as seguintes definições sobre a tabela-horário X:

- $apd_i(X)$: Total de horários disponíveis para uma disciplina i em X.
- $aps_i(X)$: Total de combinações de horário e sala disponíveis para uma disciplina i em X.
- $nl_i(X)$: Total de aulas não alocadas da disciplina i em X.
- $uac_{i,j}(X)$: Total de aulas das disciplinas ainda não alocadas completamente que se tornam indisponíveis para alocação no horário j após a inserção de uma aula da disciplina i no horário j.

Sobre qualquer tabela-horário parcial X, a escolha da aula de uma disciplina, entre as disciplinas ainda não totalmente alocadas, é realizada de acordo com a heurística que possui os seguintes critérios (*HR1*):

1. Escolhe-se a disciplina com o menor valor de $apd_i(X)/\sqrt{nl_i(X)}$.
2. Se existir múltiplas disciplinas com o mesmo valor, é escolhida a disciplina com o menor valor de $aps_i(X)\sqrt{nl_i(X)}$.
3. Se ainda houver a ocorrência de mais de uma disciplina com o menor valor, é escolhida a disciplina com maior valor de $conf_i$, sendo $conf_i$ o número de disciplinas que compartilham estudantes ou professores com a disciplina i. Se ainda existir empates isso é resolvido seguindo a ordem alfabética do nome da disciplina.

Na heurística descrita acima, disciplinas com poucos horários disponíveis para alocação e um grande número de aulas não alocadas tem prioridade na escolha das aulas. Essa prioridade existe pois essas disciplinas possuem poucas opções de horários disponíveis para suas aulas serem alocadas. Por outro lado, também tem prioridade as disciplinas que possuem um grande número de aulas ainda não alocadas. No caso, se ocorrer o empate entre duas ou mais

disciplinas, é levado em consideração o número de horários e salas disponíveis, e o número de disciplinas em conflito.

Na segunda fase da heurística, é realizada a escolha do horário e a sala que a aula selecionada a_i^* será alocada. É escolhido o horário *j* e sala *k* com o menor valor da função *g* (*j*, *k*), dada pela fórmula (*HR2*):

$$g\ (j, k) = k_1\ uac_{i^*, j}\ (X) + k_2\ \Delta f_s\ (i^*, j, k) \qquad (2)$$

Sendo $\Delta f_s\ (i^*, j, k)$ o valor da penalidade das restrições fracas sobre a possível alocação da aula a_i^* no horário *j* e sala *k* . Os pesos k_1 e k_2 são relativos as restrições fortes e fracas, respectivamente, e $uac_{i^*, j}\ (X)$ representa o número de aulas das disciplinas ainda não alocadas completamente, que se tornam indisponíveis para alocação no horário *j* após a possível inserção da aula a_i^* no horário *j* e sala *k* .

A Figura 2 apresenta o algoritmo da heurística de construção de uma solução inicial.

Entrada: Instância *I*
Saída: *Solução X*
1. *LC* ← *ListaAulasNãoAlocadas()*
2. **enquanto** |*LC*| > 0 **faça**
3. escolha uma aula a_i disponível de acordo com *HR1* em *LC*.
4. escolha um horário *j* e sala *k* para a aula baseada em *HR2*.
5. faça a alocação da aula a_i no horário *j* e sala *k* em *X*.
6. retire a_i de LC.
7. **fim enquanto**
8. **retorne** X.

Figura 2: Algoritmo da Solução Inicial proposto por Lu e Hao (2010).

Conforme demonstrado por Lu e Hao (2010), este método guloso garante a construção de uma tabela-horário válida para as instâncias do ITC-2007.

3.4. Busca Local

A busca local tem como objetivo melhorar a solução construída inicialmente. Para atingir esse objetivo, é necessária a definição de estratégias para explorar a vizinhança de uma solução. Neste trabalho, foram implementados os seguintes movimentos para geração de soluções vizinhas:

- *Move*: Uma aula é movida para uma posição desocupada na tabela-horário. A Figura 3 apresenta um exemplo da execução de um movimento do tipo *Move*.

Figura 3: Exemplo de movimento do tipo *Move*.

- *Swap*: Duas aulas trocam de posição na tabela-horário. A Figura 4 apresenta um exemplo da execução de um movimento do tipo *Swap*.

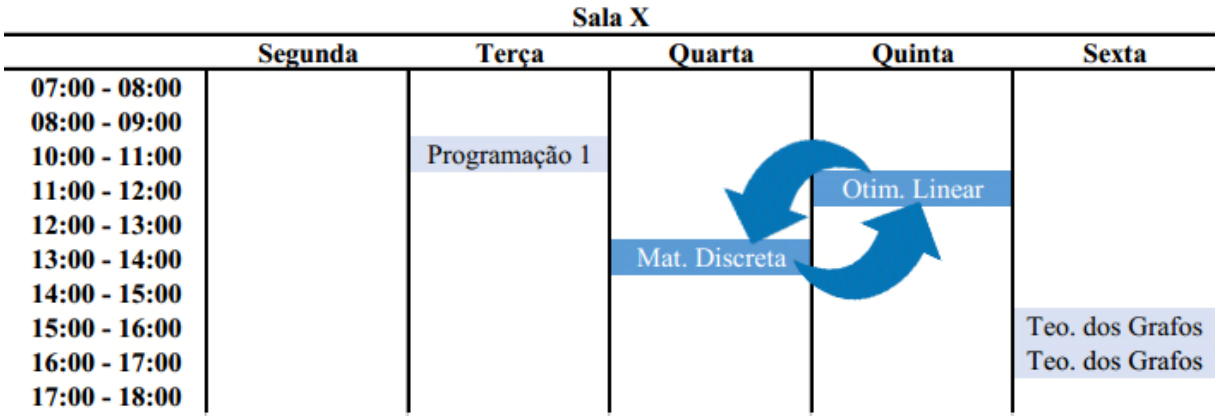

Figura 4: Exemplo de movimento do tipo *Swap*.

- *Time-Move*: Uma aula é movida para outro dia na tabela-horário, sem alterar o horário e sala onde está alocada, desde que o horário esteja disponível. A Figura 5 apresenta um exemplo da execução de um movimento do tipo *Time-Move*.

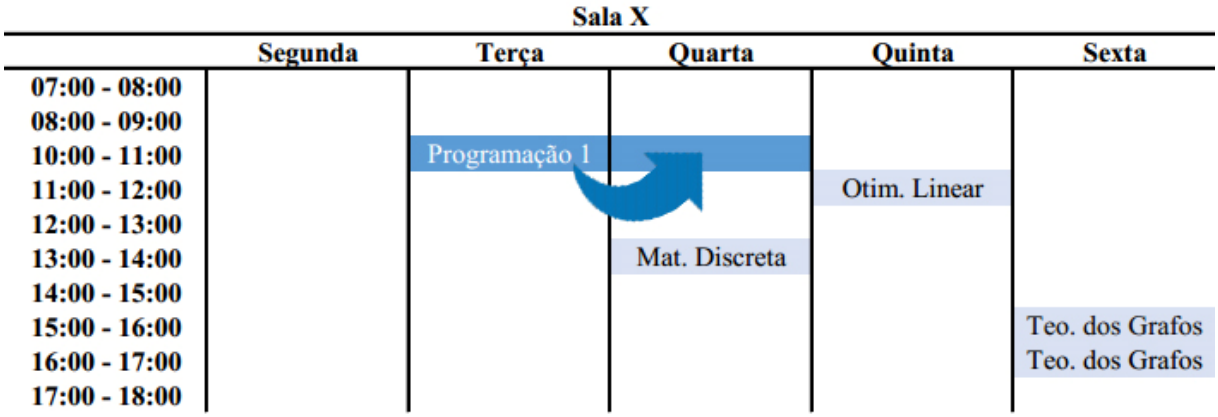

Figura 5: Exemplo de movimento do tipo *Time-Move*.

- *Room-Move*: Uma aula é movida para uma sala disponível na tabela-horário, no mesmo dia e horário em que está alocada. A Figura 6 apresenta um exemplo da execução de um movimento do tipo *Room-Move*.

Figura 6: Exemplo de movimento do tipo *Room-Move*.

Como algoritmo de busca local é utilizada a meta-heurística denominada *Simulated Annealing* (SA), pois a mesma se mostrou satisfatória nos trabalhos de Rocha (2013) e Segatto et al. (2015).

A meta-heurística possui cinco parâmetros principais: a solução inicial S para o problema, a temperatura inicial T_i, a temperatura final T_f, a taxa de resfriamento β e o número de soluções vizinhas Nv que deverão ser geradas a cada temperatura. A estratégia faz analogia ao processo de metalurgia, que aquece um metal sólido e realiza o resfriamento lentamente até que o material se solidifique, sendo acompanhado e controlado até que a estrutura se mantenha uniforme (KIRKPATRICK et al., 1983).

O algoritmo parte de uma temperatura inicial que vai sendo resfriada até chegar a temperatura final. Em cada temperatura, são gerados N_v vizinhos. Se o vizinho gerado é melhor que a solução atual, esta é atualizada. Se o vizinho possuir valor de f maior do que a solução atual, ele pode ser aceito com uma probabilidade igual a $e^{-\Delta f/T}$, sendo Δf a diferença entre o valor de f do vizinho e da solução atual, e T a temperatura atual. Quanto maior for o valor de Δf e menor a temperatura, menores serão as chances de aceitar a solução vizinha. O comportamento inicial do algoritmo é de aceitar grande amplitude de soluções quando a temperatura está alta. À medida que a temperatura cai, poucas pioras vão sendo aceitas e uma determinada região da busca é intensificada. A Figura 7 apresenta o pseudocódigo do algoritmo *Simulated Annealing* para a fase de busca local.

Entrada: Solução S, T_i, T_f, β, N_v
Saída: Solução S_{Melhor}

```
1.  T ← Ti;     SAtual ← S;     SMelhor ← S;
2.  enquanto T > Tf faça
3.      para i ← 1 até Nv faça
4.          SVizinho ← GeraVizinho(SAtual);
5.          Δf ← f(SVizinho) - f(SAtual);
6.          se Δf < 0 então
7.              SAtual ← SVizinho;
8.              se f(SVizinho) < f(SMelhor) então
9.                  SMelhor ← SVizinho;
10.             fim se
11.         senão
12.             Gere um número aleatório p ∈ ]0, 1]
13.             se p < e^(−Δf/T) então
14.                 SAtual ← SVizinho;
15.             fim se
16.         fim se
17.     fim para
18.     T ← T * β;
19. fim enquanto
20. retorne SMelhor
```

Figura 7: Algoritmo do SA apresentado por Rocha (2013).

Na fase de busca local, o algoritmo *Simulated Annealing* (SA) é executado utilizando os movimentos *Move* e *Swap*. Se houver melhora na solução, o algoritmo SA é executado utilizando agora apenas os movimentos *Time-Move* e *Room-Move*. Em ambas execuções a probabilidade de escolha ocorre na mesma proporção: 50% de chance de escolha para cada um dos movimentos. O ciclo se repete até que não ocorra nenhuma melhora na solução, conforme apresentado na Figura 8. Essa estratégia foi adotada, pois foi verificado através de testes prévios, que os movimentos *Move* e *Swap* apresentam melhores resultados, do que os movimentos *Time-Move* e *Room-Move*, quando aplicados a uma solução não refinada, isto é, uma solução que ainda não passou por uma busca local. Os movimentos *Time-Move* e *Room-Move* apresentam melhores resultados quando aplicados a uma solução melhorada por um procedimento de busca local.

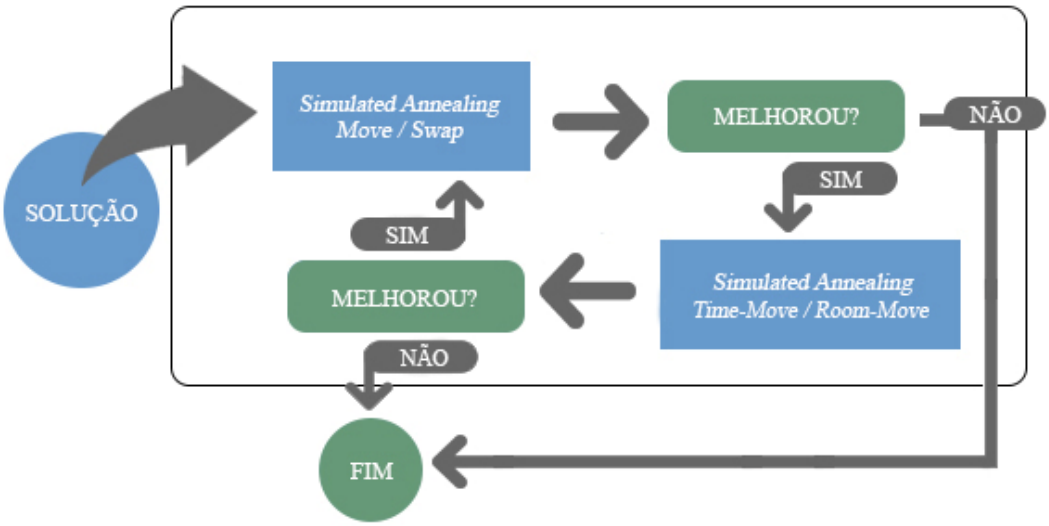

Figura 8: Execução do SA para fase de busca local.

3.5. Perturbação

Como estratégia de perturbação, é utilizado o movimento Cadeia de Kempe conforme definido por Segatto et. al. (2015). No PTHU, a Cadeia de Kempe é executada da seguinte forma:

A partir de uma tabela-horário válida, são escolhidos randomicamente dois horários, denominados t_1 e t_2. A aula a_{11} está alocada no horário t_1 na sala 1, a aula a_{12} esta alocada no mesmo horário na sala 2, e assim por diante, conforme pode ser visualizado na Figura 9. Em seguida, é gerado um grafo[1] de conflitos, sendo que cada aula representa um vértice. Se duas aulas conflitam por causa de uma restrição forte, são ligadas por uma aresta. Cada subgrafo[2] conexo formado representa uma Cadeia de Kempe. A seguir, são escolhidas uma ou mais Cadeias de Kempe para que seja realizada a troca entre as aulas pertencentes a essa cadeia.

[1]Dado um conjunto V, e um conjunto E de pares de elementos de V, dizemos que o par ordenado G = (V, E) é um grafo. Os elementos do conjunto E, são chamados de arestas (JURKIEZICZ, 2009).

[2]G' é dito um subgrafo de G se V(G') ⊆ V(G) e A(G') ⊆ A(G) (JURKIEZICZ, 2009).

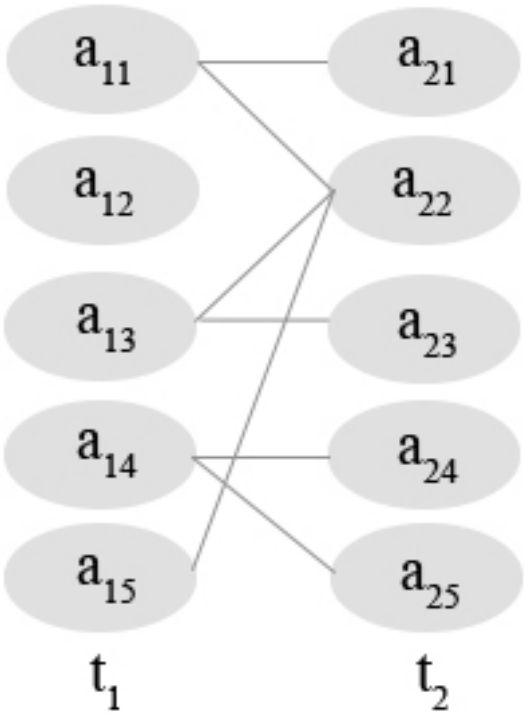

Figura 9: Exemplo de Cadeias de Kempe nos horários t_1 e t_2.

No exemplo da Figura 9, existem três Cadeias de Kempe: a primeira é a cadeia que contém as aulas: a_{11}, a_{21}, a_{22}, a_{13}, a_{23} e a_{15}, a segunda é a cadeia formada pelas aulas a_{14}, a_{24} e a_{25}, e a terceira é a cadeia formada apenas pela aula a_{12}. A tabela horário obtida após a realização das trocas continuará obrigatoriamente sendo válida. Essa garantia ocorre pois todas as aulas dos horários escolhidos que conflitam através de uma restrição forte, serão trocadas de horário. Um movimento da Cadeia de Kempe válido seria trocar as aulas da primeira Cadeia de Kempe que pertencem ao horário t_1 com as aulas que pertencem a mesma cadeia no horário t_2, obtendo uma nova alocação das aulas nesses dois horários t_1 e t_2, como demonstrado na Figura 10.

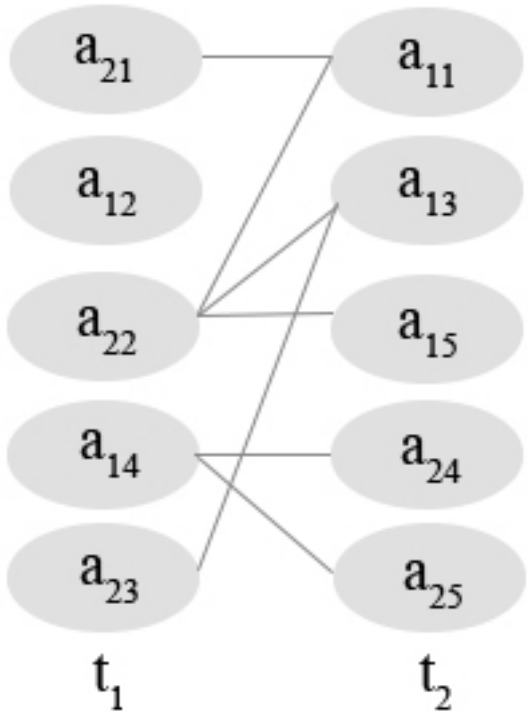

Figura 10: Exemplo de um movimento de Cadeia de Kempe nos horários t_1 e t_2.

Na fase de perturbação, é executado um movimento de Cadeia de Kempe qualquer, pois foi verificado através de testes que tal movimento é forte o bastante na modificação de uma solução para que ocorra a exploração de novas regiões de busca. Ao mesmo tempo, a Cadeia de Kempe se mostrou fraca o bastante para que a solução não seja desconstruída totalmente.

3.6. Critério de Aceitação

Dado uma a solução atual *S*, uma solução intermediária *S'* é aceita se $f(S') < f(S)$. Nesse caso, *S* é descartada e as próximas fases do algoritmo ILS são executadas com *S'*. Caso contrário, *S'* é descartada a exploração continua com *S*.

4. RESULTADOS COMPUTACIONAIS

Neste capítulo, são apresentados os resultados dos experimentos computacionais. Além disso, é apresentado alguns detalhes de implementação, a escolha dos parâmetros do ILS e os parâmetros específicos da busca local *Simulated Annealing*. Os resultados computacionais obtidos neste trabalho são comparados com o trabalho de Rocha (2013) e os resultados oficiais do ITC-2007.

4.1. Escolha dos Parâmetros

A meta-heurística ILS possui apenas um parâmetro: o número máximo de iterações ILS_{max}. Neste trabalho, o número máximo de iterações foi substituído por unidade de tempo. Para obter o tempo equivalente a execução do algoritmo em uma máquina utilizada no ITC-2007, os organizadores do campeonato ofereceram uma ferramenta executável para realizar o *benchmark* na máquina de testes dos competidores, disponível para *download* em PATAT (2008). Utilizando essa ferramenta na máquina em que foram realizados os testes, foi estipulado o tempo de execução de 192 segundos.

Os parâmetros de execução do algoritmo *Simulated Annealing* foram baseados no trabalho de Rocha (2013). A Tabela 1 apresenta todos os parâmetros utilizados neste trabalho.

Tabela 1: Parâmetros de valores do ILS.

Parâmetro	Descrição	Valor
ILS_{max}	Tempo máximo de execução do ILS	192 segundos
T_i	Temperatura inicial para o SA	1,5
T_f	Temperatura final para o SA	0,005
β	Taxa de resfriamento para o SA	0,999
N_v	Número máximo de iterações para o SA	500

4.2. Detalhes de Implementação

Todos os algoritmos descritos nesse trabalho foram implementados utilizando linguagem de programação C++, compilados com GCC versão 4.8.4 64 bits e executados em um computador com processador Intel I5-3570 3.4 GHz, 8 Gb de memória RAM, e sistema operacional Ubuntu 14.04 LTS 64bits. Os arquivos contendo o código fonte do algoritmo descrito neste trabalho estão disponíveis em: https://github.com/renancmonteiro/ils_simulated_annealing_timetabling.

4.3. Análise dos Resultados

Assim como feito no ITC-2007, o algoritmo foi executado 10 vezes com diferentes *seeds* para a geração de números aleatórios. A Tabela 2 apresenta as melhores respostas encontradas para cada instância do ITC-2007. Os valores representam apenas as violações das restrições fracas, dado que ao final da execução todas as soluções obtidas se apresentaram viáveis.

Tabela 2: Melhores soluções obtidas pelo algoritmo proposto.

Instância	ILS + SA		
	Melhor	Média	Desvio Padrão
comp01	5	5	0
comp02	68	89,35	10,03
comp03	97	117	11,56
comp04	44	52,75	3,54
comp05	430	536,75	46,78
comp06	68	89,70	8,60
comp07	43	52,60	6,04
comp08	55	59,45	3,09
comp09	114	125,55	6,66
comp10	44	57,20	8,17
comp11	0	0	0
comp12	428	480	28,36
comp13	83	89,20	4,46
comp14	66	75,40	3,69
comp15	90	116,55	11,83
comp16	59	70,05	7,47
comp17	102	113,45	7,08
comp18	93	98,1	3,28
comp19	94	111,45	10,88
comp20	73	92,85	11,64

comp21	136	115,3	9,14
Média	104,38		

Nas Tabelas Tabela 3 e Tabela 4, podem ser vistos os resultados obtidos pelos competidores do ITC-2007. As colunas estão ordenadas de acordo com o resultado geral da competição. Foram adicionadas duas colunas ao final das tabelas: uma com os resultados obtidos por Rocha (2013) e outra com os resultados encontrados pelo algoritmo implementado neste trabalho. A melhor resposta de cada instância está destacada em negrito.

O ITC-2007 foi realizado em duas fases detalhadas a seguir. Na primeira fase, todos os competidores registrados receberam um conjunto inicial contendo sete instâncias. Duas semanas antes do término da primeira fase, os competidores receberam um novo conjunto de sete instâncias. Na primeira fase, os competidores submeteram os algoritmos e as melhores respostas encontradas para cada instância. A Tabela 3 lista os resultados obtidos na primeira fase. O resultado final da primeira fase foi divulgado apenas com os nomes dos 5 primeiros colocados na competição. Os demais (6º ao 17º) foram identificados apenas pela posição final da primeira fase.

Os cinco primeiros colocados na primeira fase foram selecionados para a etapa seguinte. Neste momento, os organizadores executaram os algoritmos com sete novas instâncias, chamadas de ocultas por não serem divulgadas previamente aos competidores. A Tabela 4 lista os resultados obtidos na segunda fase.

Tabela 3: Resultados da primeira fase do ITC-2007.

Instância	Muller	Lu	Atzuna	Clark	Geiger	6°	7°	8°	9°	10°	11°	12°	13°	14°	15°	16°	17°	Rocha (2013)	ILS+SA
comp01	**5**	**5**	**5**	10	**5**	9	23	6	31	18	30	114	97	112	**5**	61	943	**5**	**5**
comp02	43	**34**	55	83	108	154	86	185	218	206	252	295	393	485	127	1976	128034	73	68
comp03	72	**70**	91	106	115	120	121	184	189	235	249	229	314	433	141	739	55403	98	97
comp04	**35**	38	38	59	67	66	63	158	145	156	226	199	283	405	72	713	25333	48	44
comp05	**298**	**298**	325	362	408	750	851	421	573	627	522	723	672	1096	10497	28249	79234	409	430
comp06	**41**	47	69	113	94	126	115	298	247	236	302	278	464	520	96	3831	346845	75	68
comp07	**14**	19	45	95	56	113	92	398	327	229	353	291	577	643	103	7470	396343	36	43
comp08	**39**	43	42	73	75	87	71	211	163	163	224	204	373	412	75	833	64435	58	55
comp09	103	**102**	109	130	153	162	177	232	220	260	275	273	412	494	159	776	44943	119	114
comp10	**9**	16	32	67	66	97	60	292	262	215	311	250	464	498	81	1731	365453	41	44
comp11	**0**	**0**	**0**	1	**0**	**0**	5	**0**	8	6	13	26	99	104	**0**	56	470	**0**	**0**
comp12	331	**320**	344	383	430	510	828	458	594	676	577	818	770	1276	629	1902	204365	375	428
comp13	66	**65**	75	105	101	89	112	228	206	213	257	214	408	460	112	779	56547	97	83
comp14	53	**52**	61	82	88	114	96	175	183	206	221	239	487	393	88	756	84386	72	66
Média	**79,21**	**79,21**	92,21	119,21	126,14	171,21	192,86	231,86	240,43	246,14	272,29	296,64	415,21	523,64	870,36	3562,29	132338,14	107,57	110,35

Tabela 4: Resultados obtidos na segunda fase do ITC-2007.

Instância	Muller	Lu	Atsuta	Geiger	Clark	Rocha (2013)	ILS+SA
comp15	84	**71**	82	128	119	101	90
comp16	**34**	39	40	81	84	69	59
comp17	**83**	91	102	124	152	105	102
comp18	83	69	**68**	116	110	102	93
comp19	**62**	65	75	107	111	87	94
comp20	**27**	47	61	88	144	88	73
comp21	**103**	106	123	174	169	136	136
Média	**68**	69,71	78,71	116,86	127	98,29	92,42

Considerando a média de soluções alcançadas na primeira fase do ITC-2007, o ILS+SA obteve a média de resultados 28% inferior aos dois primeiros colocados da competição e 8% superior ao quarto colocado. Quando comparado à média dos resultados obtidos por Rocha (2013), o ILS+SA obteve um resultado próximo, sendo inferior em apenas 2%.

Na segunda fase, o ILS+SA foi inferior ao resultado do primeiro colocado em 36% e superior as respostas obtidas pelo último colocado em 37%. Em média, caso o algoritmo proposto participasse da competição, ocuparia a posição de quarto lugar. Ao ser comparado com o resultado de Rocha (2013), o ILS+SA se mostrou superior em 6%.

5. CONCLUSÕES

Este trabalho tratou o Problema de Tabela Horário de Universidades - PTHU utilizando a terceira formulação do *International Timetabling Competition* realizado no ano de 2007.

Para uma comparação fiel, ou seja, mais próxima ao ambiente computacional no qual os algoritmos submetidos no campeonato foram testados, utilizou-se as mesmas instâncias e tempo equivalente a execução em uma máquina utilizada no ITC-2007.

Com os resultados obtidos após a execução do método proposto, pode-se observar que a utilização da meta-heurística ILS combinada com *Simulated Annealing* e Cadeia de Kempe para a resolução do PTHU se mostrou eficiente, apresentando soluções competitivas quando comparadas às soluções obtidas pelos competidores do ITC-2007. No pior caso, com a instância *comp05,* foi obtido o sétimo lugar entre os 17 competidores. Comparado com os resultados obtidos por Rocha (2013), o ILS obteve resultados superiores em 13 das 21 instâncias executadas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

GOTLIEB, C. C. **The construction of class-teacher time-tables**. In: *IFIP Congress*. [S.l.: s.n.], 1962. p. 73–77.

JURKIEWICZ, S. **Grafos – Uma introdução**. OBMEP. 2009.

KIEFER, A; HARTL, R. F; SCHNELL, A. **Adaptive large neighborhood search for the curriculum-based course timetabling problem**. *Annals of Operations Research*, 2016.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. **Optimization by Simulated Annealing**. *Science*, v. 220, n. 4598, p. 671–680, 1983.

LEWIS, R. **A survey of metaheuristic-based techniques for University Timetabling problems**. OR Spectrum, v. 30, n. 1, p. 167–190, 2008. Springer-Verlag.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. **Iterated local search**. In. Glover, T.; Kochenberger, G. A. *Handbook of Metaheuristics*, cap. 11, p. 321–353, 2003.

LU, Z.; HAO, J. **Adaptive Tabu Search for Course Timetabling**. European Journal of Operational Research 200(1), p. 235-244, 2010. Disponível em: <http://www.info.univ-angers.fr/pub/hao/papers/EJOR08.pdf>

MÜLLER, T. **ITC2007 solver description: A hybrid approach**. *Annals of Operations Research*, v. 172, n. 1, p. 429–446, 2009.

PATAT. *International Timetabling Competition*. 2008. URL: .

ROCHA, W. DE S. **Algoritmo GRASP para o Problema de Tabela-horário de Universidades**. Dissertação (Mestrado em Informática) Universidade Federal do Espírito Santo (UFES), Vitória, 2013.

SANTOS, H. G. **Formulações e algoritmos para o problema da programação de horários em escolas**. Tese (Doutorado em Computação) Universidade Federal Fluminense (UFF), Niterói, 2007.

SANTOS, H. G.; SOUZA, M. J. F. **Programação de Horários em Instituições Educacionais: Formulações e Algoritmos**. XXXIX SBPO - A Pesquisa Operacional e o Desenvolvimento Sustentável, p. 2827–2882, 2007.

SCHAERF, A. **A survey of automated timetabling**. *Artificial Intelligence Review*, v. 13, p. 87–127, 1995.

SEGATTO, E. DE A.; BOERES, M. C. S.; RANGEL, M. C.; KAMPKE, E. H. **Um Algoritmo GRASP com Cadeia de Kempe Aplicado ao Problema de Tabela-horário para Universidades.** XLVII SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2015.

SOUZA, M. J. F. **Programação de horários em escolas: uma aproximação por metaheurísticas**, 2000. Rio de Janeiro: Universidade Federal do Rio de Janeiro.