META-HEURÍSTICA GRASP PARA O PROBLEMA DE TABELA-HORÁRIO DE DISCIPLINAS DO DEPARTAMENTO DE COMPUTAÇÃO DO CCA-UFES

Lucas Vital Moreira
Universidade Federal do Espírito Santo - UFES
Alto Universitário, s/nº - Cx. Postal 16, Guararema – CEP: 29500-000 – Alegre-ES
lucasvitalmoreira@gmail.com

Renan Costalonga Monteiro
Universidade Federal do Espírito Santo - UFES
Alto Universitário, s/nº - Cx. Postal 16, Guararema – CEP: 29500-000 – Alegre-ES
renanmonteiro@msn.com

Edmar Hell Kampke, Geraldo Regis Mauri
Universidade Federal do Espírito Santo - UFES
Alto Universitário, s/nº - Cx. Postal 16, Guararema – CEP: 29500-000 – Alegre-ES
{edmar.kampke, geraldo.mauri}@ufes.br

RESUMO

O Problema de Tabela-Horário em Universidades (PTHU) possui grande relevância no âmbito acadêmico, por ser um processo complexo e desgastante que deve ser realizado todo semestre. O problema consiste em alocar um conjunto de aulas em determinados horários e salas, de tal forma que a solução encontrada atenda os anseios dos envolvidos da melhor forma possível. Sendo assim, este trabalho propõe a resolução do PTHU do Departamento de Computação (DCOMP) do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo (CCA-UFES), que atualmente é elaborado manualmente, por meio da utilização da meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) com a meta-heurística *Simulated Annealing* sendo usada na fase de busca local do algoritmo. Os resultados são comparados com um trabalho da literatura, que utiliza as mesmas instâncias, e também com a solução manual elaborada pelos coordenadores de curso do DCOMP do CCA-UFES.

PALAVRAS CHAVE. Problema de Tabela-Horário em Universidades, GRASP, Simulated Annealing.

Tópicos: MH-Meta-heurísticas, OC-Otimização Combinatória.

ABSTRACT

The University Timetabling Problem has great relevance in the academic area, and is a complex and exhausting process that must be done every semester. The problem is allocate a set of classes at certain times and rooms, so that the solution meets the desires of the people involved in the best possible way. This work proposes the resolution of problem in Computational Departament (DCOMP) of Agricultural Sciences Center of the Federal University of Espírito Santo (CCA-UFES), which is currently prepared manually, through the use of metaheuristic Greedy Randomized Adaptive Search Procedure (GRASP) with the metaheuristic Simulated Annealing being used in local search phase of the algorithm. The results are compared with a work in the literature, which uses the same instances, and also with the manual solution prepared by the course coordinators of DCOMP-CCA-UFES.

KEYWORDS. University Timetabling Problem. GRASP. Simulated Annealing.

Paper topics: MH-Metaheuristics, OC-Combinatorial Optimization.

1. Introdução

Os problemas de programação (*scheduling*) são problemas clássicos de otimização combinatória, pois tratam da alocação de recursos em determinados horários, satisfazendo algumas restrições, e minimizando (ou maximizando) uma função matemática. Além disso, os Problemas de Tabela-Horário (PTH) formam um subconjunto importante de problemas, pois possuem variadas aplicações em situações do cotidiano como, por exemplo, criação de escala de trabalho de funcionários de uma indústria e agendamento de reuniões em uma empresa.

Nesse sentido, existem muitos estudos relacionados ao PTH de escolas ou universidades, pois a solução manual desse problema não é uma tarefa fácil, e as instituições de ensino precisam resolvê-lo frequentemente. Vale ressaltar que nem sempre a alocação de horários manual é satisfatória, ou mesmo viável, visto que é difícil atender todos os desejos das partes envolvidas e até satisfazer algumas restrições.

Segundo [Schaerf 1999] e [Souza 2000], os PTHs da área educacional podem ser classificados em três categorias: PTH de Escolas, PTH de Universidades e PTH de Exames. No caso do Problema de Programação 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]. Em outras palavras, o objetivo é alocar um conjunto de aulas em um número pré-determinado de horários, satisfazendo diversas restrições que envolvem professores, alunos e o espaço físico disponível [Santos 2007].

Para a solução do problema, a satisfação de todos os requisitos é desejável, porém, nem sempre é possível atendê-los completamente. Portanto, as restrições estabelecidas devem ser classificadas de acordo com sua importância [Eiselt e Laporte 1987]. De acordo com [Santos e Souza 2007], usualmente são utilizadas duas categorias:

- Restrições Fortes: devem ser satisfeitas a qualquer custo. O não atendimento desse tipo de restrição inviabiliza a solução;
- Restrições Fracas: são aquelas cuja satisfação é desejável, mas caso não seja possível atendê-las, a solução não é inviabilizada.

Para o PTHU considerado neste trabalho, ou seja, considerando o caso real do Departamento de Computação (DCOMP) do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo (CCA-UFES), foram consideradas as 15 restrições definidas por [Mariano 2014].

A complexidade do PTHU está entre as mais altas da área de otimização combinatória e aumenta à medida que são adicionadas restrições. Segundo [Schaerf 1999], o PTHU é classificado como NP-Completo para a maioria das formulações. Assim, uma solução exata só pode ser garantida para instâncias bem pequenas, que não correspondem à realidade da maioria das instituições de ensino.

O PTHU é de difícil generalização, como dito por [Souza 2000], pois cada sistema é desenvolvido para atender características específicas da instituição de ensino e do regime educacional da região onde será aplicado. Assim, a análise comparativa entre as metodologias empregadas fica prejudicada.

No entanto, existe a necessidade de propor algoritmos cada vez mais eficientes e que produzam soluções satisfatórias para o PTHU em um tempo razoável, independente do tamanho da instância. Além disso, os algoritmos devem ser facilmente adaptáveis para a realidade da instituição de ensino. Por serem relativamente simples de implementar e produzirem bons resultados, diferentes meta-heurísticas têm sido aplicadas ao PTHU, com destaque para Algoritmos Genéticos [Erben e Keppler 1995] e Busca Tabu [Elloumi et al. 2008].

Em geral, percebe-se que o PTHU é um problema de fácil entendimento, porém de difícil solução. Portanto, o objetivo deste trabalho é aplicar a meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP), com a meta-heurística *Simulated Annealing* como procedimento de busca local, para resolução do PTHU, considerando o caso real do DCOMP do

CCA-UFES.

O restante deste trabalho está organizado como segue: Na seção seguinte é apresentada uma revisão da literatura para o problema. Já na seção 3 o problema é descrito de forma mais detalhada. Na seção 4 é apresentada a metodologia para o desenvolvimento deste trabalho. Os resultados obtidos são apresentados na seção 5 e, por fim, na seção 6 são apresentadas as conclusões.

2. Revisão da Literatura

Segundo [Rocha 2013], o problema de tabela-horário de instituições de ensino foi analisado inicialmente na década de 60, quando [Gotlieb 1962] iniciou estudos nesta área com o intuito de apresentar soluções através de otimização combinatória. A partir de então, este tema ganhou notoriedade com os trabalhos de [Schaerf 1995] e [Lewis 2008].

[Müller 2009] resolve o problema de tabela-horário de universidades do *International Timetabling Competition* (ITC), realizado no ano de 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] também trata o PTHU apresentado na formulação do ITC-2007, apresentando bons resultados ao aplicar a meta-heurística GRASP às 21 instâncias da competição. Além disso, [Rocha 2013] testou os métodos *Hill Climbing* e *Simulated Annealing* como métodos de busca local do GRASP, sendo que o método *Path-Relinking* também é utilizado, mas para intensificar a busca por soluções de boa qualidade. Assim, o trabalho de [Rocha 2013] foi a motivação para aplicar a meta-heurística GRASP com *Simulated Annealing* ao PTHU do DCOMP do CCA-UFES.

[Mariano 2014] utiliza a meta-heurística *Adaptive Large Neighborhood Search* (ALNS), aplicando-a ao problema de tabela-horário em universidades, sendo que as instâncias utilizadas são específicas do DCOMP do CCA-UFES. O ALNS é considerado um método recente e que tem apresentado bons resultados com problemas desta natureza.

[Kostuch 2005] desenvolve um algoritmo para o PTH de escolas, usando a formulação proposta no ITC-2002, que constrói a tabela-horário em três etapas. Na primeira é usada um algoritmo de coloração de grafos para se obter uma solução inicial viável. Na segunda e terceira etapas aplica-se o *Simulated Annealing*, mas em cada etapa é usada uma estrutura de vizinhança diferente. [Fonseca et al. 2011] resolvem o mesmo problema reduzindo-o ao problema da Satisfatibilidade Proposicional (SAT) para gerar uma solução inicial e aplica uma Busca Tabu para otimização dessa solução.

Muitos outros trabalhos propuseram diferentes abordagens para resolver o problema. Os trabalhos citados nesta seção representam apenas uma parte dos diversos trabalhos sobre o PTH.

3. Descrição do Problema

O PTHU considerado neste trabalho foi modelado para atender às necessidades do DCOMP do CCA-UFES. Dessa forma, a seguir é apresentada a modelagem deste problema.

3.1. Modelagem

A modelagem usada para testar o método proposto é baseada no trabalho de [Mariano 2014]. Cada estrutura utilizada é definida por um identificador e as demais características.

As estruturas são armazenadas em listas que possibilitam estabelecer as relações entre os conjuntos de elementos que são trabalhados. Desse modo, as estruturas mais simples seriam: aulas, salas, professores, disciplinas, cursos e horários. Além dessas, é estabelecida uma estrutura denominada oferta, que promove a relação entre todas as outras.

A entidade “professores” representa o conjunto de professores da instituição. Já a entidade “disciplinas” representa o conjunto de disciplinas a serem ministradas, sendo que cada disciplina, além do código e nome, possui um valor representando o seu grau de dificuldade (0 – disciplinas normais, 1 – disciplinas difíceis).

O conjunto de salas disponíveis também foi uma entidade modelada, sendo que cada sala deste conjunto possui uma capacidade e pode ser de um tipo (0 – Salas Normais, 1 – Laboratório). A entidade turma representa a combinação de um curso com um período como, por exemplo, Ciência da Computação - 4º período, que engloba todos os alunos do curso de Ciência da Computação que estão no 4º período. Assim, um conjunto de turmas pode ser definido, sendo que cada turma possui um turno preferencial (0 – Matutino, 1 – Vespertino, 2 – Noturno) que representa o melhor turno para se alocar as aulas daquela turma.

Por fim, a partir da definição dessas entidades, é criada a principal entidade do problema, denominada oferta, que além de representar a combinação de uma disciplina, com seu respectivo professor e a respectiva turma, possui informações como, por exemplo, em que tipo de sala deve ser alocada aquela aula (0 – Salas Normais, 1 – Laboratório), o número de vagas, o turno (1 – Diurno: Matutino ou Vespertino, 2 – Noturno) e a carga horária.

A carga horária de cada oferta é a duração da aula na semana, podendo ser de uma, duas, três ou quatro horas. Com isso, se houver duas aulas de uma disciplina na semana, sendo uma de duas horas e a outra de uma hora, então haverá duas ofertas, cada uma com a respectiva carga horária. Uma particularidade do DCOMP do CCA-UFES é com relação às ofertas com carga horária igual a três horas, que preferencialmente devem ser alocadas nos três primeiros horários do turno diurno, ou nos três primeiros horários do turno noturno, ou ainda nos três últimos horários do turno noturno.

Por fim pode-se dizer então que a alocação de todas as ofertas representa uma solução do PTHU abordado neste trabalho.

3.2. Representação da Solução

Uma solução para o PTHU é representada com uma matriz tridimensional, com dimensões *L*, *M* e *N*, que representam respectivamente a quantidade de salas de aula, a quantidade de horários disponíveis e os dias letivos da semana.

Se a matriz possui todas as células com valor (-1), isso representa que nenhuma aula foi alocada ainda, ou seja, todos os horários e salas estão disponíveis em todos os dias letivos da semana. Caso contrário, as células que estiverem com valor (-1) representarão os horários vagos e as demais terão o código da entidade oferta que foi alocada para aquele horário. Na Figura 1 é apresentado um exemplo da estrutura de uma solução.

Observe que existem diversas soluções possíveis para uma mesma instância do problema, já que o PTHU é um problema de otimização combinatória. Dessa forma, é necessário definir as restrições que incidem sobre o problema, para que a partir disso as soluções sejam classificadas de acordo com sua qualidade, ou seja, quanto menor o número de violações das restrições, melhor será a qualidade da solução.

Sala X						Sala Y					
	Seg	Ter	Quar	Quin	Sex		Seg	Ter	Quar	Quin	Sex
07:00h-08:00h	-1	-1	-1	2	-1	07:00h-08:00h	-1	-1	-1	-1	-1
08:00h-09:00h	0	-1	6	2	6	08:00h-09:00h	-1	3	-1	-1	-1
09:00h-10:00h	0	-1	6	2	6	09:00h-10:00h	-1	3	-1	-1	-1
10:00h-11:00h	-1	0	-1	-1	-1	10:00h-11:00h	3	-1	4	5	5
11:00h-12:00h	-1	0	1	-1	-1	11:00h-12:00h	3	-1	-1	5	5
13:30h-14:30h	-1	9	-1	-1	10	13:30h-14:30h	7	-1	9	10	-1
14:30h-15:30h	-1	9	-1	-1	10	14:30h-15:30h	7	-1	9	10	-1
15:30h-16:30h	-1	-1	8	-1	11	15:30h-16:30h	8	7	-1	11	-1
16:30h-17:30h	-1	-1	8	-1	11	16:30h-17:30h	8	7	-1	11	-1
18:20h-19:10h	12	-1	-1	-1	-1	18:20h-19:10h	-1	16	17	16	17
19:10h-20:00h	12	-1	-1	-1	-1	19:10h-20:00h	-1	16	17	16	17
20:00h-20:50h	12	13	14	13	14	20:00h-20:50h	-1	-1	-1	-1	-1
21:00h-21:50h	-1	13	14	13	14	21:00h-21:50h	15	-1	-1	-1	-1
21:50h-22:40h	-1	-1	-1	-1	-1	21:50h-22:40h	15	-1	-1	-1	-1

Figura 1: Exemplo de estrutura de uma solução.

3.3. Restrições

Para o PTHU considerado neste trabalho, ou seja, considerando o caso real do DCOMP do CCA-UFES, algumas restrições particulares da instituição de ensino foram levadas em consideração. A partir de então, cada restrição foi classificada em forte ou fraca, da mesma forma como proposto por [Mariano 2014].

3.3.1. Restrições Fortes

1. Conflitos de professor: um professor não poderá ministrar mais de uma disciplina no mesmo dia e horário;
2. Conflitos de turmas: uma turma não poderá assistir a mais de uma aula no mesmo dia e horário;
3. Conflitos de salas: uma sala de aula não poderá estar reservada para mais de uma disciplina no mesmo dia e horário;
4. Capacidade da sala: uma turma não poderá ser alocada em uma sala cuja capacidade seja inferior ao número de alunos da turma;
5. Tipo incompatível de sala: as aulas não poderão ser alocadas em uma determinada sala que não é compatível ao tipo solicitado, por exemplo, aulas que deveriam ser realizadas em laboratórios e foram alocadas em salas normais de aula;
6. Disciplinas "especiais": disciplinas com 3 horas aulas semanais deverão ser alocadas nos três primeiros horários do turno diurno e nos três primeiros ou três últimos horários do turno noturno, permitindo assim que outras disciplinas possam ser alocadas entre esses horários;
7. Aulas fora do turno: uma aula não poderá ser alocada fora do turno da oferta (diurno ou noturno).

3.3.2. Restrições Fracas

8. Intervalo de trabalho do professor: o intervalo entre o primeiro e o último dia da semana em que um professor ministrará as aulas deverá ser minimizado;
9. Janelas de horário: intervalos na grade de horários de cada turma, entre duas aulas, deverão ser reduzidos;
10. Período preferencial: as turmas diurnas deverão ter suas disciplinas concentradas no período da manhã ou da tarde. Assim, a quantidade de disciplinas ofertadas fora do turno "preferencial" de cada turma deverá ser minimizada;
11. Aulas seguidas: aulas repetidas de uma disciplina ministradas para uma turma no mesmo dia devem ser evitadas;
12. Intervalo entre períodos: a ocorrência de professores que ministram aula em um dia à noite e no dia seguinte pela manhã deverá ser minimizada;
13. Aulas seguidas de nível "difícil": as aulas de complexidade "difícil" ministradas em horários sequenciais devem ser evitadas;
14. Aulas de nível "difícil" no último horário: aulas de complexidade "difícil" ministradas no último horário de cada dia deverão ser evitadas;
15. Aulas de carga horária par: aulas com 2 ou 4 horas do turno diurno deverão ser alocadas fora do primeiro horário do dia.

Assim as restrições de 1 a 7 são do tipo forte, e o não atendimento destas inviabiliza a solução. Já as demais, de 8 a 15, são do tipo fraca, pois elas não inviabilizam a solução, mas é necessário que sejam minimizadas.

3.4. Função Objetivo

Para medir a qualidade das soluções foi utilizada uma função objetivo com penalizações para cada restrição. Dessa forma, as restrições fortes serão mais penalizadas que as restrições fracas. Portanto, para avaliação de uma solução s do problema, considerando P, T e D a quantidade de professores, turmas e disciplinas, respectivamente, a função objetivo f é definida como:

$$MINIMIZAR\ f(s) = \delta_1 \sum_{p=1}^{P} CP_p + \delta_2 \sum_{t=1}^{T} CT_t + \delta_3 \sum_{l=1}^{L} CS_l + \delta_4 FTO + \delta_5 \sum_{l=1}^{L} VS_l + \delta_6 TSI + \delta_7 D3H$$

$$+\delta_8\sum_{p=1}^{P} IT_p + \delta_9\sum_{t=1}^{T} JH_t + \delta_{10}\sum_{t=1}^{T} PP_t + \delta_{11}\sum_{d=1}^{D} AS_d + \delta_{12}\sum_{p=1}^{P} NP_p$$

$$+\delta_{13}ASD + \delta_{14}ADU + \delta_{15}DHP$$

Sendo que:

- CP_p é o número de conflitos do professor p, ou seja, o número de vezes que o professor p ministra aula no mesmo dia e horário;
- CT_t é o número de conflitos da turma t, ou seja, o número de vezes que os alunos da turma t assistem mais de uma aula no mesmo dia e mesmo horário;
- CS_l é o número de conflitos da sala l, ou seja, o número de vezes que a sala l está atribuída a mais de uma turma no mesmo dia e mesmo horário;
- VS_l é o número de violações na capacidade da sala l, ou seja, o número de turmas alocadas na sala l cujo número de alunos é maior que a capacidade da sala;
- *TSI* é o número de aulas alocadas em salas de tipo "incompatível", ou seja, se 10 aulas devem ser em laboratório e foram alocadas em salas normais, $TSI = 10$;
- *D3H* é o número de disciplinas de 3 horas aulas semanais alocadas fora do primeiro horário do turno diurno ou noturno, e fora do quinto horário do turno noturno;
- *FTO* é o número de aulas alocadas fora do turno especificado;
- IT_p é a diferença entre o primeiro e o último dia em que o professor p ministra aulas em relação a um intervalo padrão;
- JH_t é o número de janelas de horário da turma t, ou seja, o número de horários vagos entre aulas ao longo da semana para a turma t;
- PP_t é o número de aulas da turma t fora do seu período preferencial;
- AS_d é o número de aulas seguidas da disciplina d, ou seja, o número de vezes que a disciplina d é repetida num mesmo dia;
- NP_p é o número de vezes que o professor p ministra aula à noite a partir das 20h em um dia e pela manhã (qualquer horário) no dia seguinte;
- *ASD* é o número de aulas seguidas de disciplinas de nível difícil, ou seja, o número de vezes ao longo da semana em que duas disciplinas "difíceis" são consecutivas para a mesma turma;
- *ADU* é o número de aulas de disciplinas de nível difícil ministradas no último horário, ou seja, o número de vezes ao longo da semana em que disciplinas "difíceis" são ministradas no último horário da tarde e da noite;
- *DHP* é o número de aulas de disciplinas, do turno diurno, com carga horária par, alocadas no primeiro horário do dia.

A função objetivo (FO) apresentada contempla todas as restrições descritas anteriormente. Os pesos ($\delta = [\delta_1, \delta_2, \delta_3,..., \delta_{15}]$) devem ser ajustados para penalizar as restrições fortes e fracas, a fim de viabilizar uma solução. Assim, para obter uma solução final com qualidade, é necessário eliminar os sete primeiros termos da FO e minimizar os demais.

4. Algoritmo Proposto para Solução do PTHU do DCOMP do CCA-UFES

Devido o elevado número de restrições, o PTHU abordado neste trabalho é um problema difícil de resolver. Sendo que atualmente a tabela de horários é feita semestralmente de forma manual pelos coordenadores dos cursos do DCOMP do CCA-UFES, em um processo que tem duração mínima de dois dias.

A partir disso, este trabalho propõe a utilização da meta-heurística *Greedy Randomized Adaptative Search Procedure* (GRASP), com o procedimento de busca local baseado na meta-heurística *Simulated Annealing* (SA), para o PTHU do DCOMP do CCA-UFES. A meta-heurística GRASP já apresentou bons resultados em outros problemas complexos, inclusive PTHUs, e por isso foi escolhida para ser aplicada ao PTHU do DCOMP do CCA-UFES.

4.1. Algoritmo GRASP

A meta-heurística GRASP foi introduzida por [Feo e Resende 1989] para tratar o problema de cobertura de conjuntos. Na Figura 2 é apresentado o pseudocódigo genérico da meta-heurística GRASP.

Entrada: MaxIter, α
Saída: Solução S_{Melhor}
$f^* \leftarrow \infty$;
para $i \leftarrow 1$ **até** *MaxIter* **faça**
    $S_{Inicial} \leftarrow GeraSolucaoInicial(\alpha)$;
    $S_{Atual} \leftarrow BuscaLocal(S_{Inicial})$;
    se $f(S_{Atual}) < f^*$ **então**
        $S_{Melhor} \leftarrow S_{Atual}$;
        $f^* \leftarrow f(S_{Melhor})$;
    fim se
fim para

Figura 2: Algoritmo GRASP apresentado por [Feo e Resende 1989].

O algoritmo GRASP é um procedimento iterativo composto de duas fases, que são detalhadas nas próximas duas subseções, sendo que a primeira fase constrói uma solução inicial, e a segunda fase aplica um procedimento de busca local para melhorá-la. A resposta final é a melhor solução obtida entre todas as iterações [Feo e Resende 1995]. Essa meta-heurística possui dois parâmetros principais: o número máximo de iterações e α.

4.2. Construção da Solução Inicial

A solução inicial para o problema de tabela-horário em universidade é construída a partir de uma lista inicial com todas as ofertas e uma tabela-horário completamente vazia. A partir disso, a lista de ofertas é ordenada pelo número de horários da tabela-horário no qual a oferta pode ser alocada. Assim são feitas simulações de alocação, em todas as posições da matriz de salas, com a oferta que ocupa a primeira posição da lista ordenada, ou seja, com a oferta que possui naquele instante menos horários disponíveis. Cada simulação tem sua FO calculada e assim essas simulações formarão uma lista de candidatos (LC).

A LC possui a posição na qual a simulação de alocação foi efetuada e seu custo de alocação. A LC é ordenada pelo valor do custo e servirá de base para formar uma segunda lista, denominada lista restrita de candidatos (LRC), que é composta por um intervalo da LC ordenada. Esse intervalo é definido de acordo com o parâmetro α, da seguinte forma: $[c_{min}, c_{min} + \alpha(c_{max}\ c_{min})]$, em que c_{min} representa o menor custo de alocação dentre as simulações realizadas, c_{max} representa, por sua vez, o maior custo, e $\alpha \in [0,1]$.

Sendo que, se $\alpha = 1$, LRC terá todos os elementos de LC, e se $\alpha = 0$, LRC terá apenas o primeiro elemento de LC. Um horário disponível é escolhido aleatoriamente da LRC e a oferta é alocada na respectiva tabela-horário que representa a solução, ou seja, quando $\alpha = 1$ a escolha do horário é feita de forma totalmente aleatória, caso $\alpha = 0$ sempre será escolhido o horário de LC que possui menor custo, o que representa uma escolha totalmente gulosa.

Após a alocação da oferta na tabela-horário, ela é retirada da lista inicial e o procedimento é repetido até que todas as ofertas sejam inseridas na tabela-horário.

4.3. Busca Local

A busca local tem como objetivo melhorar a solução construída inicialmente. Como estratégia de busca local, foi utilizada a meta-heurística SA, pois se mostrou satisfatória em outros trabalhos, como o de [Rocha 2013]. Essa meta-heurística possui cinco parâmetros principais: a solução inicial para o problema, a temperatura inicial, a temperatura final, a taxa de resfriamento e o número de soluções vizinhas que deverão ser geradas a cada iteração. 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 1984].

O algoritmo parte de uma temperatura inicial que vai sendo resfriada até chegar à 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 for pior, ele pode ser aceito com uma probabilidade P igual $e^{-\Delta f/T}$, sendo Δf a diferença entre o valor da função objetivo do vizinho e da solução atual, e T é a temperatura atual. Quanto maior for 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 3 apresenta o algoritmo SA para a fase de busca local.

Entrada: Solução S, T_i, T_f, β, N_v
Saída: Solução S_{Melhor}
$T \leftarrow T_i$; $S_{Atual} \leftarrow S$; $S_{Melhor} \leftarrow S$;
enquanto $T > T_f$ **faça**
  para $i \leftarrow 1$ **até** N_v **faça**
    $S_{Vizinho} \leftarrow$ GeraVizinho(S_{Atual});
    $\Delta f \leftarrow f(S_{Vizinho})$ - $f(S_{Atual})$;
    se $\Delta f < 0$ **então**
      $S_{Atual} \leftarrow S_{Vizinho}$;
      se $f(S_{Vizinho}) < f(S_{Melhor})$ **então**
        $S_{Melhor} \leftarrow S_{Vizinho}$;
      fim se
    senão
      Gere um número aleatório $P \in]0, 1]$
      se $P < e^{-\Delta f/T}$ **então**
        $S_{Atual} \leftarrow S_{Vizinho}$;
      fim se
    fim se
  fim para
  $T \leftarrow T * \beta$;
fim enquanto

Figura 3: Algoritmo do SA apresentado por [Rocha 2013].

Para explorar novas combinações nas vizinhanças da solução foram utilizados dois movimentos, cuja escolha é feita de forma aleatória:

- *Move*: Uma oferta é movida para uma posição desocupada na tabela-horário.
- *Swap*: Duas ofertas trocam de posição na tabela-horário.

5. Resultados Computacionais

Essa seção apresenta os resultados computacionais alcançados neste trabalho, constando as análises comparativas com um trabalho existente na literatura, bem como as soluções manuais elaboradas pelos coordenadores de curso do DCOMP do CCA-UFES. Além disso, as escolhas dos parâmetros do GRASP, os parâmetros específicos da busca local SA e os pesos de penalização das restrições na FO também são apresentados.

5.1. Escolha de Parâmetros

O GRASP possui dois parâmetros: número máximo de iterações *MaxIter* e o valor α, que define a forma como a construção da solução inicial será conduzida. Segundo [Rocha 2013], dentre alguns valores do intervalo [0.01,0.5] usados para teste, o valor que melhor se ajusta as condições encontradas é de $\alpha = 0.15$, pois produziu soluções iniciais diversificadas e de qualidade razoável. O *MaxIter*, número máximo de iterações do GRASP, foi substituído por unidade de tempo, afim de uma análise comparativa com o trabalho de [Mariano 2014], o qual utiliza 500 segundos. O algoritmo SA possui mais parâmetros, tornando a tarefa de calibração mais complexa. Foram escolhidos parâmetros que permitem certo grau de diversificação no início da busca e maior intensificação no final do processo. Com base nestas observações e em outros trabalhos da literatura, foram escolhidos os parâmetros apresentados na Tabela 1, e todos se mostraram satisfatórios para este trabalho.

Tabela 1: Parâmetros e valores do método proposto.

Parâmetro	Descrição	Valor
MaxIter	Número máximo de iterações do GRASP	500 seg
α	Valor limiar da LRC	0,15
T_i	Temperatura inicial para o SA	1000
T_f	Temperatura de congelamento para o SA	0,01
β	Taxa de resfriamento para o SA	0,975
N_v	Número máximo de iterações para o SA	500

Como abordado na Seção 4.3, as soluções vizinhas na busca local são geradas somente com *Move*, ou somente com *Swap*. O movimento a ser utilizado é escolhido aleatoriamente e ambos possuem igual probabilidade de ocorrer. Para a escolha das penalizações de 1 a 15 foram utilizados os valores definidos pelo trabalho de [Mariano 2014], afim de comparação.

Tabela 2: Penalizações e valores do PHTU.

Penalização	Valor
δ_1	5000
δ_2	5000
δ_3	5000
δ_4	5000
δ_5	5000
δ_6	300
δ_7	5000
δ_8	10
δ_9	20
δ_{10}	4
δ_{11}	600
δ_{12}	10
δ_{13}	10
δ_{14}	10
δ_{15}	500

5.2. Detalhes de Implementação

Todos os algoritmos descritos neste trabalho foram implementados na linguagem C++, e foram testados em máquina *Windows* com a distribuição *Windows 7 Professional*, com processador *Intel dual-core* 2.0 GHz e 2 Gb de memória RAM. As ofertas das disciplinas de Trabalho de Conclusão de Curso I, Trabalho de Conclusão de Curso II e Estágio em Informática foram desconsideradas do problema, uma vez que são disciplinas que não necessitam ser consideradas no horário e não requerem espaço físico.

5.3. Análise dos Resultados

O algoritmo foi executado 5 vezes para cada um dos semestres letivos: 2013/2 e 2016/1, e ao final das execuções, todas as soluções obtidas se apresentaram viáveis, ou seja, não feriram nenhuma restrição forte para o problema.

Como apresentado nas Tabelas 3 e 4, a média de tempo observada para se encontrar a melhor solução para 2013/2 e 2016/1 foi de 343,82 e 255,67 segundos, respectivamente. A melhor solução encontrada para 2013/2 teve o valor de $f(s) = 466$, enquanto que para 2016/1 o menor valor encontrado foi $f(s) = 322$. Em média, o valor da função objetivo para 2013/2 e 2016/1 foi de $f(s) = 482{,}4$ e $f(s) = 335{,}6$, respectivamente.

Com base no trabalho de [Mariano 2014], é possível realizar uma análise comparativa da solução de 2013/2 e da melhor solução obtida pelo algoritmo *Adaptive Large Neighborhood Search* (ALNS), que foi $f(s)$ = 480. A melhor solução encontrada neste trabalho, pelo algoritmo GRASP, obteve $f(s)$ = 466. Assim, é possível verificar que a melhor solução obtida pelo algoritmo proposto neste trabalho alcançou uma melhora de 2,91% quando comparada à melhor solução de Mariano (2004).

Tabela 3: Resultados obtidos pelo algoritmo GRASP para o semestre 2013/2

Penalização	Melhor Solução GRASP	Média Soluções GRASP	Solução Manual	Melhor solução Mariano (2014)
CPp	0	0	0	0
CTt	0	0	0	0
CSs	0	0	0	0
OFT	0	0	0	0
VSs	0	0	0	0
TSI	0	0	0	0
DH3	0	0	0	0
Itp	9	9,8	10	9
JHt	0	0,2	6	0
PPt	24	24,6	0	0
ASd	0	0	0	0
NDp	0	0	1	0
ASD	15	15,4	34	26
ADU	13	12,8	19	13
DHP	0	0	0	0
Função Objetivo	**466**	**482,4**	**760**	**480**
Tempo (segundos)	**389,63**	**343,816**	**-**	**450,51**

Tabela 4: Resultados obtidos pelo algoritmo GRASP para o semestre 2016/1

Penalização	Melhor Solução GRASP	Média Soluções GRASP	Solução Manual
CPp	0	0	0
CTt	0	0	0
CSs	0	0	0
OFT	0	0	0
VSs	0	0	0
TSI	0	0	0
DH3	0	0	0
Itp	5	5,8	2
JHt	0	0	8
PPt	18	17,4	16
ASd	0	0	3
NDp	0	0	1
ASD	12	12,4	14
ADU	8	8,4	18
DHP	0	0	0
Função Objetivo	**322**	**335,7**	**2374**
Tempo (segundos)	**84,55**	**255,666**	**-**

Em relação à solução manual, elaborada pelos coordenadores de curso do DCOMP do CCA-UFES, em um período mínimo de dois dias, nota-se pelas Tabelas 3 e 4 que a melhor solução obtida pelo algoritmo GRASP alcançou uma melhora de 38,68% e 86,44%, respectivamente, para os semestres 2013/2 e 2016/1.

Assim, pode-se afirmar que o algoritmo proposto neste trabalho alcançou resultados satisfatórios, ocasionando uma boa distribuição das aulas e proporcionando um melhor aproveitamento dos horários, tanto para alunos quanto para professores.

6. Conclusões

Pode-se observar neste trabalho que a aplicação da meta-heurística GRASP combinada com *Simulated Annealing* para resolução do Problema de Tabela-Horário em Universidades (PTHU) foi eficiente, apresentando boas soluções quando comparadas às obtidas por [Mariano 2014] e comparadas à solução manual do DCOMP do CCA-UFES.

Também foi possível concluir com este trabalho que a elaboração do problema encontra desafios com relação à inconstância das instâncias disponíveis do PTHU, como por exemplo, a frequente troca de professores e a falta de espaços físicos definidos e totalmente livres para a alocação de aulas do DCOMP do CCA-UFES.

Com a aplicação do GRASP aliado ao *Simulated Annealing*, foi possível encontrar uma melhora significativa nas soluções. Isto é, com a aplicação desses métodos, foi possível obter tabelas de horários de sala, professor e turma com uma menor ocorrência de restrições quando comparadas às soluções atuais que são construídas manualmente.

Agradecimentos: Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (processo 303052/2013-9 e 454569/2014-9) e à Fundação de Amparo à Pesquisa e Inovação do Espírito Santo - FAPES (processos 67656021/2014, 67627153/2014 e 73290475/2015) pelo apoio financeiro.

Referências

Eiselt, H. A. e Laporte, G. (1987), Combinatorial optimization problems with soft and hard requirements. *Journal of Operations Research*, v. 38, n. 9, p. 785-795.

Elloumi, A., Kamoun, H., Ferland, J. e Dammak, A. (2008), A tabu search procedure for course timetabling at a tunisian university. In *Proceedings of the 7th PATAT Conference*.

Erben, W. e Keppler, J. (1995), A genetic algorithm solving a weekly course-timetabling problem. In *Proceedings of ICPTAT'95*, *1st International Conference on the Practice and Theory of Automated Timetabling*, Napier University, Edinburgh, UK, pp. 21-32.

Feo, T. A. e Resende, M. G. C. (1989). A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem. *Oper. Res. Lett.*, v. 8, n. 2, p. 67–71.

Feo, T. A. e Resende, M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, v. 6, n. 2, p. 109–133.

Fonseca, G. H. G, Ribeiro, R. G. e Martins, F. V. C. (2011), Uma abordagem híbrida de SAT e busca tabu para o problema da programação de horários escolares. In: *Simpósio Brasileiro de Pesquisa Operacional*, 43, Ubatuba. Anais... SBPO.

Gotlieb, C. C. (1962), The construction of class-teacher timetables. In: *Proceedings of IFIP Congress*, Anais... North-Holland Pub. Co., Amsterdam.

Kirkpatrick, S. (1984), Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, Kluwer Academic Publishers-Plenum Publishers.

Kostuch, P. (2005), The University Course Timetabling Problem with a Three-Phase Approach. Lecture Notes in Computer Science. v. 3616, p.109–125.

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

Mariano, G. P. (2014), *Resolução do problema de programação de horários de disciplinas do CCA-UFES utilizando a meta-heurística ALNS*. Monografia de Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal do Espírito Santo (UFES), Alegre-ES.

Müller, T. (2007), ITC2007 solver description: A hybrid approach. *Annals of Operations Research*, v. 172, n. 1, p. 429–446, 2009.

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

Schaerf, A. (1995), Survey of automated timetabling. *Artificial Intelligence Review*, v. 13, n. 2, p. 87–127.

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

Santos, H. G. e Souza, M. J. F. (2007), Programação de horários em instituições educacionais: formulações e algoritmos. In: *Simpósio Brasileiro de Pesquisa Operacional*, 39, Fortaleza. Anais... SBPO.

Souza, M. J. F. (2000), *Programação de horários em escolas: uma aproximação por metaheurísticas*. Tese (Doutorado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro-RJ.