

SIMULATED ANNEALING APLICADO AO PROBLEMA DE PROGRAMAÇÃO DE HORÁRIOS DO CCA-UFES

André Soares Carvalho

Universidade Federal do Espírito Santo - UFES
andresoaescarvalho@gmail.com

Gelinton Pablo Mariano

Universidade Federal do Espírito Santo - UFES
pablo06@gmail.com

Edmar Hell Kampke

Universidade Federal do Espírito Santo - UFES
edmar.kampke@ufes.br

Geraldo Regis Mauri

Universidade Federal do Espírito Santo - UFES
geraldo.mauri@ufes.br

Resumo

Atualmente, as instituições de ensino superior têm grande dificuldade em montar de maneira adequada os horários das disciplinas dos cursos, devido aos inúmeros requisitos pedagógicos, pessoais e institucionais envolvidos. Este trabalho tem como objetivo desenvolver e aplicar uma nova alternativa para resolver o Problema de Programação de Horários em Universidades (PPHU). O PPHU é um problema clássico na área de otimização combinatória e tem grande relevância prática e teórica. Para solucionar o problema, foi utilizada a meta-heurística *Simulated Annealing* (SA) que, além de ser de fácil implementação, tem apresentado bons resultados para outros problemas relacionados. Com o intuito de avaliar o desempenho da SA para resolução do PPHU, foram utilizados os dados referentes à oferta de disciplinas do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo (CCA-UFES) realizada no período letivo 2013/2. Os resultados apresentam o bom desempenho do algoritmo proposto.

Palavras-Chaves: Otimização Combinatória; Meta-heurísticas; Problema de Programação de Horários em Universidades; *Simulated Annealing*.

Abstract

Universities have great difficulty in creating the timetables of each course because of the many restrictions involved. The objective of this work is to develop and apply a new alternative to solve the timetabling problem in universities. The problem is a classical problem in combinatorial optimization area and has great relevance. To solve the problem, we used the metaheuristic *Simulated Annealing*, because it is easy to implement and has shown good results for other related problems. To evaluate the performance of SA we used courses data concerning of the Federal University of Espírito Santo, in the school year 2013/2. The results present the good performance of the proposed algorithm.

Keywords: Combinatorial Optimization; Metaheuristics; University Timetabling Problem; *Simulated Annealing*.

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 Programação de Horários (PPH) formam um subconjunto importante de problemas, pois possuem variadas aplicações no nosso cotidiano como, por exemplo, elaboração de escala de funcionários de uma indústria e agendamento de partidas para campeonatos esportivos.

Nesse sentido, o PPH de escolas ou universidades sempre recebeu muita atenção, pois a solução manual desse problema não é uma tarefa trivial, e as instituições de ensino precisam resolvê-lo anualmente ou semestralmente. Vale ressaltar que nem sempre a alocação de horários manual é satisfatória, ou mesmo viável, visto que é difícil contemplar todos os anseios das partes envolvidas e até mesmo atender algumas restrições.

Segundo Schaerf (1999) e Souza (2000), os PPH da área educacional podem ser classificados em três categorias: PPH de Escolas, PPH de Universidades e PPH de Exames. No caso do Problema de Programação de Horários de Universidades (PPHU), o problema consiste em fixar uma sequência de encontros entre professores e alunos em um período prefixado de tempo satisfazendo a um conjunto de restrições de vários tipos (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.

Segundo Santos (2007), as restrições do PPHU podem ser classificadas como:

- Organizacionais: relativos à instituição de ensino;
- Pedagógicos: pedidos importantes para o bom aproveitamento das aulas;
- Pessoais: requisitos de acordo com as preferências pessoais dos membros do corpo docente.

Para a solução do problema, a satisfação de todas as restrições é desejável, porém, nem sempre é possível de se realizar. 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 divisões:

- 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.

Neste trabalho, é abordado o PPHU aplicado ao Centro de Ciências Agrárias (CCA) da Universidade Federal do Espírito Santo (UFES), que possui um conjunto de 17 cursos de graduação (Ciência da Computação, Engenharia Química, etc.) que por sua vez são compostos por um conjunto de disciplinas, e cada disciplina possui uma quantidade de aulas. Os estudantes podem estar matriculados nas turmas das disciplinas do seu curso e uma turma pode ter alunos de vários cursos. Outro conjunto é formado pelos horários para realização das aulas, sendo que cada horário possui uma quantidade limitada de salas com sua respectiva capacidade. O problema consiste em alocar as aulas de cada turma em uma sala em um determinado horário sem que alguma turma esteja tendo mais de uma aula ao mesmo tempo.

Dessa forma, o PPHU é considerado um dos mais difíceis da área de otimização combinatória e sua dificuldade aumenta na medida em que são adicionadas restrições. Schaerf (1999) mostrou que o PPHU 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 às instâncias reais da maioria das instituições de ensino.

Na década de 60, Csima e Gotlieb (1964) e Gotlieb (1962) deram início ao processo de automatização do PPHU. Desde então, o problema tem sido amplamente estudado, e diferentes abordagens e métodos de solução vêm sendo apresentados.

O PPHU 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 PPHU em uma quantidade de tempo razoável, independente do tamanho da instância. Além disso, 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 PPHU, com destaque para Algoritmos Genéticos (Erben e Keppler, 1995) e Busca Tabu (Elloumi *et. al.*, 2008).

Em geral, percebe-se que o PPHU é um problema de fácil entendimento, porém de difícil solução. Portanto, o objetivo deste trabalho é aplicar a meta-heurística *Simulated Annealing* (SA) ao PPHU, considerando instâncias reais do CCA-UFES, e comparar a qualidade das soluções obtidas pela meta-heurística com soluções obtidas manualmente, além de comparar as soluções obtidas pela meta-heurística com soluções obtidas pelo *software* comercial *aSc TimeTables*.

O restante deste trabalho está organizado da seguinte forma: 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

Csima e Gotlieb (1964) foram os precursores na automação do processo de gerar quadros de horários. Nesse trabalho, os autores propuseram um método envolvendo matrizes binárias.

Segundo Souza (2000) e Schmidt e Ströhlein (1980), a maioria dos primeiros trabalhos utilizavam heurísticas construtivas. A partir de então, os pesquisadores começaram a utilizar técnicas gerais para resolver o problema, como Programação Inteira Mista (Ferland e Roy, 1985; Tripathy, 1984), Fluxo em Grafo (Ostermann e Werra, 1983) e Coloração em Grafos (Cangalović e Schreuder, 1991; Neufeld e Tartar, 1974; Wood, 1969).

Novas propostas de solução para o problema têm sido apresentadas utilizando meta-heurísticas como Busca Tabu, Algoritmos Genéticos, Algoritmos Meméticos, além da combinação de diferentes métodos. Várias publicações com essas técnicas podem ser encontradas em Souza (2000), Santos (2007), Barbosa e Souza (2011), Carvalho (2011) e Fonseca *et. al.* (2011).

Em Souza (2000), é utilizado um procedimento heurístico de caminhos mínimos com capacidade para recuperar a viabilidade de um quadro de horários de professores. O autor utiliza a meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) com uma busca local realizada por meio de um algoritmo de Busca Tabu.

Santos (2007) trata o Problema de Programação de Horários em três partes. Na primeira, é apresentada uma heurística híbrida baseada em Busca Tabu. Na segunda parte, foram apresentadas propostas de solução utilizando Programação Linear Inteira Mista e finalmente, a terceira parte explora a sinergia entre heurísticas e métodos exatos. Em todas as partes, bons resultados foram obtidos.

Carvalho (2011) utiliza o *Variable Neighborhood Descent* (VND) como procedimento de busca local para um *Iterated Local Search* (ILS). Barbosa e Souza (2011) utilizam uma meta-heurística híbrida que utiliza o GRASP para geração de uma solução inicial e o ILS com relaxamento para o refinamento da solução.

Fonseca *et. al.* (2011) resolve o problema de programação de horários escolares

reduzindo-o ao problema da Satisfazibilidade Proposicional (SAT) para gerar uma solução inicial e aplica uma Busca Tabu para otimização dessa solução.

Muitos outros trabalhos propuseram abordagens diferentes para resolver o problema. Os trabalhos citados neste capítulo representam apenas uma parte dos principais trabalhos sobre o PPHU.

3. DESCRIÇÃO DO PROBLEMA

O PPHU considerado neste trabalho foi modelado para atender às necessidades do CCA-UFES. Dessa forma, a seguir são apresentadas as particularidades e a modelagem deste PPHU.

3.1. MODELAGEM

Para facilitar a implementação e a resolução do PPHU proposto neste trabalho, foi necessário modelar o problema e assim identificar as entidades a serem utilizadas. Além disso, foi necessário definir a estrutura de dados que melhor representa cada entidade.

Dentre as entidades identificadas, existe o conjunto de professores da instituição e 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 identificada, 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 dessa entidade, 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 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.

Dessa forma, pode-se dizer que alocação de todas as ofertas representa uma solução do PPHU abordado neste trabalho.

3.2. REPRESENTAÇÃO DA SOLUÇÃO

Uma solução para o PPHU 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 oferta 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 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 PPHU 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 PPHU considerado neste trabalho, ou seja, considerando o caso real 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.

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 será 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 o número de professores, T o número de turmas, L o número de salas e D o número de disciplinas, a função objetivo f é definida como:

$$\begin{aligned} MINIMIZAR f(s) = & \delta_1 \sum_{p=1}^P CP_p + \delta_2 \sum_{t=1}^T CT_t + \delta_3 \sum_{l=1}^L CS_l + \delta_4 \sum_{l=1}^L VS_l + \delta_5 TSI + \delta_6 D3H + \delta_7 FTO \\ & + \delta_8 \sum_{p=1}^P IT_p + \delta_9 \sum_{t=1}^T JH_t + \delta_{10} \sum_{t=1}^T PP_t + \delta_{11} \sum_{d=1}^D AS_d + \delta_{12} \sum_{p=1}^P NP_p \\ & + \delta_{13} ASD + \delta_{14} ADU + \delta_{15} DHP \end{aligned}$$

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 na oferta;
- 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 ($\delta = [\delta_1, \delta_2, \delta_3, \dots, \delta_{15}]$) 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. SIMULATED ANNEALING PARA O PPHU DO CCA-UFES

O PPHU abordado neste trabalho é um problema bastante difícil de resolver, pois possui muitas restrições. A utilização de um *software* comercial que automatiza a construção da tabela de horários é inviável, pois o problema possui muitas restrições específicas do CCA-UFES. Já o desenvolvimento de um *software* customizado para este fim teria um alto custo para a instituição.

Atualmente, a tabela de horários é feita semestralmente de forma manual pelos coordenadores dos cursos do CCA-UFES, ou seja, este processo envolve o trabalho de pelo menos 17 pessoas num período de no mínimo 2 dias.

A partir disso, este trabalho propõe a utilização da meta-heurística *Simulated Annealing* para o PPHU do CCA-UFES, por ser de simples implementação e por obter bons resultados em outros trabalhos relacionados como, por exemplo, Ceschia *et. al.* (2011).

A *Simulated Annealing* (SA) foi originalmente proposta por Kirkpatrick *et. al.* (1983) e se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. Na analogia, um metal inicialmente superaquecido (alta temperatura) vai sendo resfriado controladamente até atingir o ponto de solidificação.

A SA procura explorar novas áreas no espaço de soluções do problema em um curto período de tempo e com pouco esforço. Para isso, o algoritmo recebe como entrada uma solução s do problema e os valores da Temperatura Inicial (T_0), Temperatura Final (T_c), Taxa de Resfriamento (β) e o número de soluções (S_{max}) a serem exploradas em cada iteração.

4.1. SOLUÇÃO INICIAL

Neste trabalho, a solução inicial, que é um dos parâmetros do algoritmo SA, é construída usando uma heurística gulosa. As ofertas são ordenadas decrescentemente pelo número de vagas. A partir dessa ordenação cada oferta é inserida na solução, ou seja, ofertas que possuem maior número de vagas serão alocadas primeiro na solução, sendo que uma oferta será alocada somente em salas do mesmo tipo especificado para aquela oferta e que tenham capacidade maior ou igual ao número de vagas da oferta. Além de serem alocadas também no mesmo turno que foi especificado para aquela oferta.

4.2. SIMULATED ANNEALING PROPOSTO

A Figura 2 apresenta um pseudocódigo da SA implementada neste trabalho. Observe que o algoritmo trabalha em um laço de repetições, onde se considera uma temperatura T . O valor de T inicialmente é elevado e, em cada iteração, geram-se soluções em temperaturas

decrecentes até que o equilíbrio de T seja alcançado. Ou seja, a temperatura é rebaixada até que entre no estágio de congelamento, no qual não se consegue mais melhorias naquela solução.

Para cada temperatura T , iterativamente, soluções vizinhas (s') da solução atual s serão escolhidas aleatoriamente (linha 7). A qualidade da solução s' é então comparada com a qualidade da solução s por meio da função objetivo f do problema (linha 8). Caso a qualidade de s' seja melhor que s , a solução s' será considerada para obter outras soluções nas próximas iterações (linha 9). No entanto, a exploração exaustiva da vizinhança de s favorece a rápida estagnação do espaço de busca e, por isso, a SA possui um mecanismo que permite que soluções piores que s sejam aceitas, e assim uma nova vizinhança seja explorada.

Na linha 13 a solução vizinha s' , que é pior que s , é aceita com uma probabilidade igual a $e^{-\Delta f/T}$, onde Δf é a diferença de valor da função objetivo de s' e s . Dessa forma, quanto maior o valor de Δf e menor a temperatura, menores serão as chances de aceitar s' . O comportamento típico do algoritmo é aceitar grande diversificação no início, quando a temperatura está alta. À medida que ela decresce, poucas soluções piores são aceitas e assim a busca em uma determinada vizinhança é intensificada.

```

SA ( $\beta, S_{max}, T_0, T_c, s$ )
1.  $s^* \leftarrow s$ ;
2.  $T \leftarrow T_0$ ;
3.  $iterT \leftarrow 0$ ;
4. ENQUANTO ( $T > T_c$ ) FACA
5.   ENQUANTO ( $iterT < S_{max}$ ) FACA
6.      $iterT \leftarrow iterT + 1$ ;
7.      $s' \leftarrow$  uma solução vizinha qualquer de  $s$ ;
8.     SE ( $f(s') < f(s)$ )
9.        $s \leftarrow s'$ ;
10.    SE ( $(f(s') < f(s^*))$ )  $s^* \leftarrow s'$ ; FIM-SE;
11.    SENÃO
12.      TOMAR ( $x \in [0,1]$ )
13.      SE ( $x < e^{-\Delta f/T}$ )  $s \leftarrow s'$ ; FIM-SE;
14.    FIM-SE;
15.    FIM-ENQUANTO;
16.     $T \leftarrow \beta * T$ ;  $iterT \leftarrow 0$ ;
17.  FIM-ENQUANTO;
18. RETORNAR ( $s^*$ );

```

Figura 2: Algoritmo SA. Fonte: Mauri (2005).

Observe que o desempenho do algoritmo está diretamente relacionado à qualidade da solução inicial e do método para se obter uma solução vizinha.

4.3. SOLUÇÕES VIZINHAS

O algoritmo SA, descrito anteriormente, explora um espaço de busca, também denominado vizinhança, de uma solução s . Essa exploração consiste em obter uma nova solução s' vizinha de s . Neste trabalho, para obter uma solução vizinha de s será feito apenas uma mudança na estrutura da solução s , que consiste basicamente na troca de alocação entre

duas disciplinas que possuem a mesma carga horária.

5. RESULTADOS COMPUTACIONAIS

Neste artigo, testa-se o desempenho da meta-heurística *Simulated Annealing* (SA), na resolução do PPHU aplicado a uma instância real do CCA-UFES, referente ao segundo semestre letivo de 2013. Vale ressaltar que a modelagem deste problema levou em consideração muitas restrições específicas desta instituição de ensino.

Para a execução da meta-heurística, os parâmetros e as penalizações da função objetivo foram definidas de forma empírica, por meio da realização de vários testes. Observou-se que a SA se comportou melhor quando o valor das penalizações δ_2 e δ_3 foram igualados. Por haver poucas ofertas com carga horária de 3 horas, optou-se por colocar o valor da penalização δ_6 abaixo das demais. Os valores dos parâmetros e penalizações são apresentados nas Tabelas 1 e 2, respectivamente.

Tabela 1: Parâmetros utilizados pela SA.

Parâmetro	Valor
S_{Amax}	1000
β	0.995
T_0	100
T_c	0.001

Tabela 2: Penalizações da Função Objetivo.

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

O algoritmo SA foi executado 10 vezes, sendo que o limite máximo de tempo estabelecido para cada execução do algoritmo foi de 500 segundos. A SA foi implementada na Linguagem C++ e os experimentos foram realizados em um computador com processador Core i7-3610 QM de 2.30 Ghz com 8GB de memória RAM utilizando o sistema operacional Windows 7 de 64 bits.

5.1. ANÁLISE DOS RESULTADOS OBTIDOS

Para comprovar a eficiência dos resultados obtidos pelo algoritmo proposto, os resultados obtidos foram comparados com a solução manual construída pelos coordenadores de curso do CCA-UFES.

É importante destacar que foram feitas tentativas de utilizar o *software* comercial denominado *aSc TimeTables*, através do *download* da versão de teste, disponível em <http://www.asctimetables.com>. Após o *download*, foram feitos pequenos ajustes nos dados de entrada da instância real do CCA-UFES para que o mesmo fosse usado pelo *software*. Essa etapa consumiu cerca de 2 horas e o tempo de execução do *software* foi 15 minutos. No entanto, o *software* emitiu uma mensagem informando que não foi possível construir a tabela de horários com tantas restrições.

A Tabela 3 apresenta os resultados da quantidade de violações para cada restrição em cada uma das 10 execuções. Note que em todas as execuções do algoritmo as sete primeiras

restrições (restrições fortes) não foram violadas, o que indica que todas as soluções encontradas pelo método proposto são viáveis. Observe também que apenas as restrições fracas 8, 9, 12 (em 2 execuções), 13 e 14 foram violadas.

Tabela 3: Comparativo entre os resultados obtidos pela SA.

Conflitos	Soluções – Algoritmo SA										Solução Manual
	1	2	3	4	5	6	7	8	9	10	
CP_p	0	0	0	0	0	0	0	0	0	0	0
CT_t	0	0	0	0	0	0	0	0	0	0	0
CS_l	0	0	0	0	0	0	0	0	0	0	0
VS_l	0	0	0	0	0	0	0	0	0	0	0
TS_l	0	0	0	0	0	0	0	0	0	0	0
$D3H$	0	0	0	0	0	0	0	0	0	0	0
FTO	0	0	0	0	0	0	0	0	0	0	0
IT_p	8	11	10	12	11	10	12	10	11	11	10
JH_t	6	5	4	4	3	2	6	7	4	4	6
PP_t	0	0	0	0	0	0	0	0	0	0	0
AS_d	0	0	0	0	0	0	0	0	0	0	0
NP_p	0	0	0	0	1	0	0	0	0	1	1
ASD	18	16	19	17	18	21	17	14	17	16	34
ADU	13	12	12	12	11	12	12	11	12	12	19
DHP	0	0	0	0	0	0	0	0	0	0	0
f	510	490	490	490	470	470	530	490	480	480	760
$Tempo$	186	197	445	459	457	410	214	219	442	197	-

Pela análise da Tabela 3 é possível perceber claramente que o desempenho da meta-heurística *Simulated Annealing* foi satisfatório em todas as execuções, uma vez que foi possível encontrar soluções melhores, ou seja, com menor valor da função objetivo, quando comparadas com a solução construída manualmente. Em média, a SA encontrou soluções 35,5% melhores que a solução manual.

Destaca-se também a última linha da Tabela 3, que mostra o tempo necessário para o algoritmo alcançar o melhor valor da função objetivo. Assim, em média a SA encontrou as melhores soluções em 322,6 segundos, o que é um tempo muito curto quando comparado com o tempo utilizado na construção da solução manual do problema.

6. CONCLUSÕES

Neste trabalho foi proposta a utilização da meta-heurística *Simulated Annealing* para resolver o Problema de Programação de Horários em Universidades, denominado PPHU, considerando restrições específicas e um caso real do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo (CCA-UFES).

O PPHU do CCA-UFES é resolvido semestralmente de forma manual pelos coordenadores de curso, mas o processo além de ser demorado e burocrático, é também muito desgastante para todos os envolvidos. Uma alternativa seria a utilização de um *software* comercial, mas, no entanto, não foi encontrado nenhum *software* que atenda todas as restrições do CCA-UFES. Por isso, foram feitas pequenas adaptações nos dados de entrada para que o *software aSc TimeTables*, um conhecido *software* de geração de tabelas de horários para escolas, fosse utilizado. O *software* não conseguiu fornecer uma resposta para o problema, mesmo após longo período em execução.

Dessa forma, foi aplicada a meta-heurística *Simulated Annealing*, que se mostrou muito eficiente na resolução deste problema, uma vez que, ao comparar as soluções obtidas pela meta-heurística com a solução construída manualmente, foi possível notar que a meta-

heurística obteve solução 32,9% melhor que a solução manual, em apenas 186 segundos. A meta-heurística obteve soluções até 38,2% melhores que a solução manual, considerando um tempo máximo de 500 segundos.

7. AGRADECIMENTOS

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

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Barbosa, S. H. D e Souza, S. R. (2011), Resolução do problema de programação de Cursos universitários baseada em currículos via uma meta-heurística híbrida GRASP-ILS-relaxado. In: *Simpósio Brasileiro de Pesquisa Operacional*, 43, Ubatuba. Anais... SBPO.
- [2] Cangalovié, M. e Schreuder, J. A. M. (1991), Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths. *European Journal of Operational Research Society*, v. 51, p. 248-258.
- [3] Carvalho, R. (2011), *Abordagem heurística para o problema de programação de horários de cursos*. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais (UFMG), Belo Horizonte-MG.
- [4] Ceschia, S., Gaspero, L. D. e Schaerf, A. (2011), Design, engineering, and experimental analysis of a simulated annealing approach to the postenrolment course timetabling problem. *Computers & Operations Research*, v. 39, n. 7, p. 1615–1624.
- [5] Csimá, J. e Gotlieb, C. C. (1964), Tests on a computer method for construction of school timetables. *Communications of the ACM*, v. 7, p.160-163.
- [6] 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.
- [7] 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*.
- [8] 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.
- [9] Ferland, J. A. e Roy, S. (1985), Timetabling problem for university as assignment of activity to resources. *Computers & Operations Research*, v. 12, p. 207-218.
- [10] 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.
- [11] Gotlieb, C. C. (1962), The construction of class-teacher timetables. In: *Proceedings of IFIP Congress*, Anais... North-Holland Pub. Co., Amsterdam.
- [12] Kirkpatrick, S. (1984), Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, Kluwer Academic Publishers-Plenum Publishers.

- [13] Mauri, G. R. (2005), *Novas heurísticas para o problema de escalonamento de tripulações*. Dissertação de mestrado - Instituto Nacional de Pesquisa Espacial (INPE), São José dos Campos-SP.
- [14] Neufeld, G. A. e Tartar, J. (1974), Generalized graph colorations. *SIAM Journal on Applied Mathematics*, v. 29, p. 91-98.
- [15] Ostermann, R. e Werra, D. (1983), Some experiments with a timetabling system. *OR Spektrum*, v. 3, p.199-204.
- [16] 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.
- [17] 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.
- [18] Schaerf, A. (1999), A survey of automated timetabling. *Artificial Intelligence Review*, v.13, p. 87-127.
- [19] Schmidt, G. e Ströhlein, T. (1980), Timetable construction – an annotated bibliography. *The Computer Journal*, v. 23, n. 4, p. 307-316.
- [20] 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.
- [21] Tripathy, A. (1984), School timetabling – a case in large binary integer linear programming. *Management Science*, v. 30, p. 1473-1489.
- [22] Wood, D. C. (1969), A technique for colouring a graph applicable to large scale timetabling problems. *The Computer Journal*, v. 12, p. 317-319.