

## **UMA HEURÍSTICA PARA O PROBLEMA DE PROGRAMAÇÃO DE HORÁRIOS: UM ESTUDO DE CASO**

**Arydiane Magalhães Jardim**

Instituto do Noroeste Fluminense de Educação Superior – Universidade Federal Fluminense  
Santo Antônio de Pádua - RJ  
arydianejardim@gmail.com

**Gustavo Silva Semaan**

Instituto do Noroeste Fluminense de Educação Superior – Universidade Federal Fluminense  
Santo Antônio de Pádua - RJ  
gustavosemaan@id.uff.br

**Puca Huachi Vaz Penna**

Departamento de Computação - Universidade Federal de Ouro Preto  
Ouro Preto - MG  
puca@iceb.ufop.br

### **RESUMO**

Este trabalho propõe algoritmos para resolver um Problema de Programação de Horários de Cursos em Universidades. O problema consiste em definir os quadros de horários dos cursos de um departamento, alocando professores às disciplinas, horários e espaços físicos, de acordo com um conjunto de restrições. Tais restrições são definidas pelos currículos dos cursos, na alocação de professores e nos tipos e números de salas disponíveis no instituto. Devido à complexidade do problema foi proposto o uso da meta-heurística ILS. O estudo de caso foi aplicado a dados reais de um departamento da Universidade Federal Fluminense. Nos experimentos computacionais realizados foram consideradas instâncias reais que possui informações referentes ao primeiro semestre de 2014. Os resultados obtidos indicam que o algoritmo é uma boa alternativa para a resolução do problema abordado, uma vez que os resultados obtidos foram superiores aos da ferramenta, de código-aberto, utilizada pelo departamento.

**PALAVRAS CHAVE.** Problema de Programação de Horários, Busca Local Iterada, Meta-heurísticas.

**Áreas:** EDU – PO na Educação; MH – Metaheurísticas; OC – Otimização Combinatória.

### **ABSTRACT**

This work proposes an algorithm to solve a University Timetabling Problem (UTP). The problem is to define the timetabling of the department's courses, allocating professors to courses, schedules and physical spaces, according to a set of constraints. These constraints are defined by the curricula of the courses, i.e., the allocation of professors and the types and numbers of rooms available at the institute. Due to the problem complexity, the proposed algorithm is based on the ILS metaheuristic. In computational experiments real instances were considered, based on information from the first semester of 2014. The obtained results indicate that the algorithm is a good alternative for solving the problem addressed, since the results were better than the one obtained by an open source software used by Department.

**KEYWORDS.** Timetabling Problem, Iterated Local Search, Metaheuristics.

**Main Areas:** OR in Education; MH – Metaheuristics; OC – Combinatorial Optimization.

## 1. Introdução

O Problema de Programação de Horários (PPH) pode ser aplicado a todas as instituições que requerem a criação de quadros de horários. Dentre estas instituições pode-se citar: indústrias, hospitais, companhias aéreas, escolas, universidades e até em eventos esportivos.

Para criação de um quadro de horário é necessário considerar os recursos e requisitos da instituição. Uma vez que cada instituição possui características próprias, torna-se difícil propor uma solução geral para criação de quadros de horários automatizados. Alguns métodos de resolução para o PPH podem ser encontrados em [Santos 2007], [Carvalho 2011], [Oliveira *et al.* 2012], [Barbosa 2012], [Andrade 2014], [Neukirchen *et al.* 2014] e [Carvalho *et al.* 2015].

Em [Carvalho 2011], o autor considera que o PPH tem como objetivo a alocação de entidades (eventos, tarefas, pessoas, etc.) a um número restrito de recursos ao longo do tempo, respeitando as restrições existentes. Nas instituições de ensino antes do início de cada semestre letivo é necessário gerar quadros de horários. Muitas vezes a criação destes quadros demoram dias e requerem muito esforço. Conforme a quantidade de entidades e recursos existentes, a construção manual pode-se tornar inviável devido ao tempo necessário e a dificuldade em atender todas as restrições.

Devido a complexidade e ao elevado esforço necessário para a criação de quadros de horários de maneira manual, este trabalho propõe algoritmos para resolver o Problema de Programação de Horários de Cursos em Universidades (PPHU) baseados em currículos. Nesse sentido, será realizado um estudo de caso aplicado a dados reais de um departamento da Universidade Federal Fluminense (UFF).

Os PPHs pertencem à classe de problemas *NP-Difíceis* [Barbosa 2012]. Para resolução do problema é proposto a utilização da meta-heurística Busca Local Iterada (do inglês, *Iterated Local Search* – ILS).

O restante do texto está organizado como segue. Na Seção 2 é realizada uma revisão da literatura e descrição do problema abordado. Os algoritmos propostos para resolução do problema são apresentados na Seção 3. Os experimentos computacionais com dados reais do departamento estudado estão contidos na Seção 4. Por fim na Seção 5 são realizadas as considerações finais.

## 2. Revisão da Literatura e Descrição do Problema

O Problema da Programação de Horários (PPH) começou a ser estudado em 1960 [Carvalho 2011] e ainda existem pesquisas em andamento. Nesse sentido, são realizadas conferências internacionais para discutir sobre o tema, como a *Conferência Internacional de Automação de Problemas de Horário* (PATAT, do inglês *Practice and Theory on Automated Timetabling* – PATAT, 2015) e a *Competição Internacional de Programação de Horários* (ITC, do inglês *International Timetabling Competition* – ITC, 2011).

O PPH tem como objetivo a alocação de entidades (eventos, tarefas, pessoas, etc.) a um número restrito de recursos ao longo do tempo respeitando as restrições existentes [Carvalho 2011]. Neste problema encontram-se algumas dificuldades de resolução como a satisfação de todos os interesses envolvidos na produção de um quadro de horário válido e o uso adequado dos recursos disponíveis. A construção de um quadro de horário satisfatório de maneira manual pode ser inviável devido ao tempo e esforço necessários, conforme as entidades e recursos existentes.

O Problema de Programação de Horário Escolar (PPHE) é uma das variantes do PPH. O PPHE tem como objetivo programar os encontros (geralmente semanais) entre professores e estudantes, satisfazendo um conjunto de restrições: (i) organizacionais relativos à instituição de ensino e aos recursos necessários; (ii) pedagógicas que são solicitações importantes para bom aproveitamento das aulas; (iii) pessoais como satisfação/disponibilidade dos professores. Os objetivos e as restrições podem variar de acordo com cada instituição. De modo geral adota-se a divisão do PPHE baseado no tipo de entidade (Escola ou Universidade) e nos tipos de restrições associadas. Segundo [Santos e Souza 2007] o problema pode ser dividido nos seguintes tipos:

- **Problema de Programação de Horários em Escolas (PPHE)** – É a programação de horários típica, normalmente, encontrada nas escolas de ensino fundamental e médio.
- **Problema de Programação de Horários de Cursos em Universidades (PPHU)** – consiste em determinar o horário e local onde os cursos (disciplinas como: Algoritmo, Cálculo, etc.) serão realizados. Esta programação pode ser feita antes ou depois da matrícula dos alunos nos cursos.
- **Problema de Alocação de Aulas a Salas (PAAS)** – trata-se da alocação de aulas com horários já definidos as salas pode ser tratado junto com a programação de horários ou posteriormente.
- **Problema de Programação de Horários de Exames em Universidades (PPHEU)** – sua finalidade é realizar a programação dos exames.

Por pertencerem a classe de problemas *NP-Difíceis* [Barbosa 2012], os PPHs ainda necessitam de pesquisas para serem resolvidos satisfatoriamente. Os problemas de otimização combinatória podem ser resolvidos de várias formas e, entre elas, encontram-se a utilização de métodos exatos e métodos heurísticos. Os métodos exatos, por exemplo, podem consistir em enumerar todas as possíveis soluções e escolher a solução ótima. Porém a utilização deste método torna-se inviável mesmo para alguns problemas de pequeno porte, uma vez que é necessário muito tempo computacional para enumeração de todas as soluções possíveis. Por outro lado, os métodos heurísticos não garantem encontrar a solução ótima, mas geralmente encontram soluções de boa qualidade em tempos computacionais aceitáveis.

Nos últimos anos tem-se utilizado técnicas meta-heurísticas para resolução de problemas de otimização. Entre as meta-heurísticas que podem ser encontradas com destaque na literatura estão: os Algoritmos Genéticos [Corrêa 2010], GRASP [Oliveira *et al.* 2012], o ILS [Lourenço *et al.* 2010] [Andrade 2014] e a Busca Tabu [Sousa *et al.* 2008]. Na literatura existem vários trabalhos que apontam possíveis soluções para Problema de Programação de Horários. Para resolver o Problema de Programação de Horários Professor x Turma [Santos 2007] utiliza métodos híbridos, uma nova heurística híbrida baseada na Busca Tabu e Programação Linear Inteira Mista. Em [Carvalho 2011] é proposta uma resolução para problema de programação de cursos universitários baseados em currículos, e utiliza instâncias propostas pela Competição Internacional de Programação de Horários [ITC 2007]. Para resolver o problema foram desenvolvidos métodos heurísticos baseados no ILS e *Variable Neighborhood Descent* (VND). Em [Oliveira *et al.* 2012], os autores propuseram a resolução do problema de programação de horários para escolas do ensino médio utilizando a meta-heurística GRASP. Em [Barbosa 2012] é tratado o problema de programação de cursos universitários baseados em currículos, considerando as características e instâncias propostas pela ITC 2007. Em [Andrade 2014], o autor propõe a resolução do PPH aplicada a escolas de ensino fundamental utilizando duas abordagens para resolução; uma por meio de um modelo matemático de programação linear e a outra baseada na meta-heurística ILS. Estudos do PPH aplicado a instituições de ensino superior podem ser encontrados em [Neukirchen *et al.* 2014] e [Carvalho *et al.* 2015]. No primeiro trabalho foi proposto a resolução do PPH utilizando programação inteira mista. No segundo, os autores resolveram o PPHU aplicado ao Centro de Ciências Agrárias da Universidade Federal do Espírito Santo (CCA-UFES). Nesse caso foi utilizada a meta-heurística *Simulated Annealing* (SA).

## 2.1. Problema Abordado

Este trabalho propõe uma resolução de uma variante PPHE mais especificamente o Problema de Programação de Horários de Cursos em Universidades (PPHU) baseados em currículos. O PPHU estudado foi aplicado a um departamento da Universidade Federal Fluminense (UFF). O problema consiste em definir os horários para os cursos ofertados pelo departamento, tal como alocar professores às disciplinas, respeitando suas preferências quanto a horário e disciplinas. Devem ser obedecidas as restrições de conflitos definidas nos currículos dos cursos e a quantidade de salas disponíveis na instituição estudada. Deve-se considerar os

requerimentos da instituição para produção de um quadro de horário. Estes requerimentos são apresentados em forma de restrições.

De acordo com [Santos e Souza 2007] as restrições são divididas em dois tipos: (i) **Restrições fortes:** são aquelas que devem ser sempre atendidas para que uma solução seja considerada válida. Normalmente são restrições de conflitos e físicas; (ii) **Restrições fracas:** são aquelas em que o não-atendimento não inviabiliza a solução, mas servem como medida de qualidade. Nesse sentido, quanto mais restrições fracas forem atendidas, melhor será a solução encontrada.

Os dados de entrada do problema abordado estão divididos em três classes: organizacionais, pessoais e pedagógicos.

- **Organizacionais:** Cada aula (ou atividade) tem duração de 2 horas e os horários disponíveis para a sua alocação são Manhã (7h, 9h e 11h), Tarde (14h e 16h) e Noite (18h e 20h); Cada disciplina é vinculada a um período de um curso e é associada a um tipo de espaço físico (sala de aula ou laboratório específico); O professor é vinculado às disciplinas que irá lecionar; O professor está alocado no Departamento e, dessa maneira, pode lecionar para qualquer um dos cursos pertencentes ao mesmo.
- **Pessoais:** cada professor apresenta sua disponibilidade máxima diária conforme horários de aula, e informa se existe interesse em ministrar 6 horas de aulas consecutivas (três horários); O professor também expressa seu interesse em ministrar aulas em dias de semana consecutivos.
- **Pedagógica:** para disciplinas de 60 horas (possuem 4 horas semanais) deve-se forçar a divisão em dois dias da semana.

De acordo com os requisitos da instituição de ensino acima citados, foram definidas restrições fortes e fracas para avaliar a validade e a qualidade dos quadros de horários produzidos. O não atendimento de uma restrição sempre resulta na geração de uma penalidade.

Segue as descrições das restrições fortes criadas, que se violadas tornam uma solução inviável.

- I. **Período:** não podem existir duas ou mais disciplinas de um mesmo período no mesmo horário.
- II. **Espaço Físico** (número máximo de salas): o total de atividades de um horário e seus respectivos espaços físicos necessários (sala de aula ou laboratório, por exemplo) não podem ultrapassar a quantidade especificada como parâmetro (quantidade de espaços disponíveis por tipo no instituto).
- III. **Professor Disciplina:** não podem existir duas ou mais alocações de um determinado professor em um mesmo horário.

Segue as descrições das restrições fracas criadas, que são aquelas que quando atendidas tornam uma solução melhor.

- I. **Professor Disponibilidade:** cada professor apresenta uma disponibilidade preferencial, com o objetivo de também organizar as outras atividades docentes (pesquisa, extensão e a administração) da melhor maneira.
- II. **Horas Consecutivas:** o professor informa se concorda em ser alocado para atuar em três atividades consecutivas (6 horas em sala de aula).
- III. **Janela:** horários alocados de maneira não consecutiva podem gerar janelas. Por exemplo, se um professor lecionar nos horários de 14h às 16h e de 18h às 20h gera uma janela no horário de 16h às 18h. O professor estaria sem alocação para docência (aula) e poderia estar com tempo insuficiente para realizar outras atividades.
- IV. **Janela em dias da Semana:** horários alocados em dias de maneira não consecutiva podem gerar janelas. Por exemplo, se um professor lecionar na terça-feira e na quinta-feira estaria sem alocação para docência (aula) na quarta-feira. Trata-se de uma opção do professor e é informado como parâmetro de entrada.

- V. **Atividade Isolada (AULA):** ocorre quando em um dia da semana no horário de um determinado período de um curso foi alocada apenas uma atividade (aula).

### 3. Algoritmos Propostos

A presente seção apresenta a meta-heurística ILS, considerada no presente trabalho, e os procedimentos propostos, sejam eles: de construção de soluções iniciais, buscas locais (refinamento) e perturbações. Destaca-se ainda a divisão dos procedimentos de busca local em *MoveAtividade*, *IntraPeriodo* e *InterPeriodos*, conforme apresenta a subseção 3.1.2.

#### 3.1. ILS

A meta-heurística ILS é considerada por [Lourenço *et al.* 2010] de fácil implementação devido a sua simplicidade, porém é robusta e muito eficaz. A Busca Local Iterada (ILS), segundo [Temponi 2007], “*é baseada na simples ideia de que uma solução ótima local pode ser melhorada, aplicando-se um procedimento de busca local em uma nova solução, que é obtida perturbando-se a solução ótima local*”.

O método ILS, apresentado no Algoritmo 1, primeiramente realiza uma busca local na solução inicial  $s_0$ , gerando assim uma solução ótima local  $s$ . Nas próximas etapas, a cada iteração, a solução atual  $s$  é perturbada, o que gera uma solução  $s'$  e, em seguida, é realizada uma busca local em  $s'$ , obtendo-se uma nova solução ótima local  $s''$ . Por último é verificado se a solução  $s''$  é melhor do que a solução atual  $s$ . Caso ocorra melhoria na qualidade da solução, a nova solução  $s''$  é adotada com solução atual  $s$  [Temponi 2007].

#### Algoritmo 1: Pseudocódigo do ILS.

```

procedimento ILS ()
     $s_0 \leftarrow$  GeraSoluçãoInicial();
     $s \leftarrow$  BuscaLocal ( $s_0$ );
    Enquanto (os critérios de parada não estiverem satisfeitos) faça
         $s' \leftarrow$  Perturbação ( $s$ , histórico);
         $s'' \leftarrow$  BuscaLocal ( $s'$ );
         $s \leftarrow$  CritérioAceitação( $s$ ,  $s''$ );
    Fim-Enquanto
    Retorne  $s$ ;
fim ILS();
    
```

O pseudocódigo do algoritmo ILS é formado por 4 componentes: (i) um procedimento de construção que gera uma solução inicial; (ii) um procedimento de busca local, que retorna uma solução ótima local; (iii) um procedimento de perturbação que modifica a solução atual gerando assim uma solução intermediária; (iv) um critério de aceitação que decide em qual solução será aplicada a próxima perturbação.

##### 3.1.1. Construção

Na construção uma solução inicial para o problema abordado é gerada através de um procedimento aleatório. Inicialmente é selecionando aleatoriamente uma disciplina em uma lista que contém todas as disciplinas não alocadas. Em seguida, um dia da semana e um horário são escolhidos também de maneira aleatória. Assim, uma solução inicial (um quadro de horário) é formada sem considerar nenhuma restrição do problema.

##### 3.1.2. Busca Local

O método de Busca Local visa melhorar uma solução atual com base na análise de vizinhança. Nesse sentido foram implementados três movimentos para a fase de busca local: (i) *MoveAtividade*; (ii) *IntraPeriodo* e o (iii) *InterPeriodos*. O pseudocódigo da Busca Local está

representado no Algoritmo 2. Inicialmente movimentos do tipo *MoveAtividade* são realizados na solução atual  $s$  obtendo-se uma solução  $s'$ . Caso a solução  $s'$  possua uma qualidade melhor que a solução atual, isto é, um custo menor que o custo da solução atual, a solução  $s'$  passa a ser a nova solução adotada. Em seguida, o movimento *IntraPeriodo* é realizado na solução  $s'$ , obtendo-se uma nova solução  $s''$ . Finalmente, é realizado o movimento *InterPeriodos* na solução  $s''$  e obtendo-se uma nova solução  $s'''$ . Caso a solução  $s'''$  possua uma melhor qualidade que a solução atual  $s$ , então  $s'''$  será adotada com a nova solução corrente. Estes dois últimos movimentos são realizados até que um critério de parada seja satisfeito, que é encontrar a melhor solução ou realizar um número fixo de iterações sem melhora.

### Algoritmo 2: Pseudocódigo da Busca Local

```

s = Armazena a solução atual
contador = 0
Para i = 0 até número fixo de iterações faça
    s' = MoveAtividade(s);
Fim-Para
Se (custo da solução s' < custo da solução s) então
    A nova solução s' é adotada
Fim-Se
Enquanto (custo da solução s > 0) e (contador < número fixo de iterações) faça
    s'' = IntraPeriodo(s');
    s''' = InterPeriodos(s'');
    Se (custo da solução s''' < custo da solução s) então
        A nova solução s''' é adotada
        contador = 0
    Senão
        Incrementa o contador
    Fim-Se
Fim-Enquanto

```

O movimento do tipo *MoveAtividade* que têm como objetivo realizar trocas de atividades em todos os períodos de uma solução. Destaca-se que todas as atividades trocadas são escolhidas aleatoriamente. Como é mostrado no pseudocódigo apresentado no Algoritmo 3. Inicialmente são escolhidos dois dias da semana e horários. Caso as trocas das atividades nestes dias e horários sejam realizadas com sucesso obtém-se uma nova solução  $s'$ . Se o custo da nova solução  $s'$  for menor que o custo da solução atual  $s$ , então a nova solução  $s'$  é adotada como solução corrente e o *contador* recebe o número máximo de iterações. Caso contrário o *contador* é incrementado e estes passos se repetem por um número fixo de iterações para cada período da solução.

### Algoritmo 3: MoveAtividade

```

s = Armazena a solução atual
Para i = 0 até número total de períodos faça
    contador = 0
    Enquanto (contador < número fixo iterações) faça
        escolhe aleatoriamente dois dias da semana
        escolhe aleatoriamente dois horários
        Se (não foram gerados dias e horários iguais) faça
            s' = troca as atividades nos dias e horários escolhidos no período i (s);
            Se (foi possível realizar a troca) faça
                Se (custo da solução s' < custo da solução s) então
                    A nova solução s' é adotada
                Senão
                    Atualiza solução s' com a solução atual
                Fim-Se
                Contador recebe o número máximo de iterações
            Senão
                Incrementa o contador
        Fim-Enquanto
    Fim-Para

```



<b>Fim-Se</b>
<b>Fim-Se</b>
<b>Fim-Enquanto</b>
<b>Fim-Para</b>

O movimento do tipo *IntraPeriodo* tem como objetivo solucionar os conflitos existentes dentro de um mesmo período, em uma determinada solução. Estes conflitos são relativos ao não atendimento da restrição “*Período*” e restrição “*Professor Disponibilidade*” (veja Seção 2.1). Primeiramente a solução atual é armazenada. Um período e o tipo de conflito que será verificado são selecionados aleatoriamente. Logo em seguida é realizado o movimento descrito no Algoritmo 4. É realizada a troca da atividade conflitante, em que um novo dia da semana e horário são selecionados, aleatoriamente, para realocar a atividade. Estes passos são repetidos por um número fixo de iterações. A nova solução é adotada apenas se possuir um custo menor que a solução atual.

O movimento do tipo *InterPeriodos* possui funcionamento semelhante ao anterior e tem como objetivo solucionar os conflitos existentes entre diferentes períodos da solução. Estes conflitos são relativos ao não-atendimento das seguintes restrições: “*Número máximo de salas*”, “*Professor Disciplina*”, “*Janela em Dias da Semana*”, “*Janela*” e “*Horas Consecutivas*” (descritos na seção 2.1).

**Algoritmo 4:** Pseudocódigo do funcionamento dos movimentos *IntraPeriodo* e *InterPeriodos*.

```

s=Armazena a solução atual
Enquanto (contador < número fixo) faça
    Seleciona aleatoriamente o tipo de conflito verificado
    Se (houver conflito) então
        Realiza a troca da atividade conflitante
    Fim-Se
    Senão
        Incrementa o contador
    Fim-Senão
Fim-Enquanto
Se (custo da nova solução < custo atual s) então
    A nova solução é adotada
Fim-Se

```

### 3.1.3. Perturbação

O objetivo do procedimento de perturbação é a geração de uma nova solução, que é obtida modificando-se a solução atual. De acordo com [Temponi 2007] “*essa perturbação precisa ser suficientemente forte para permitir que a busca local explore diferentes espaço de soluções, mas, também, fraca o suficiente para evitar um reinício aleatório*”. Os procedimentos de perturbação são semelhantes aos métodos *MoveAtividade*, *IntraPeriodo* e *InterPeriodos*. Porém a principal diferença que estes procedimentos aceitam qualquer solução gerada, mesmo que não possuam qualidade melhor que a solução atual.

### 3.1.4. Função de Avaliação

Para avaliar as soluções obtidas para o problema abordado foi considerada a minimização das penalidades relacionadas a cada restrição descrita na Seção 2.1. Para isto, foi proposta a função de avaliação descrita na Equação (1), em que é possível observar que o somatório considera  $n$  restrições (fortes e fracas) e, para cada restrição  $k$  existe um peso  $P_k$  associado. Além disso,  $x_k$  é o total de violações de cada restrição  $k$ . Os valores dos pesos associados foram obtidos (calibrados) de maneira empírica, em experimentos preliminares, e são apresentados na Seção 4.

Ressalta-se que diferentes soluções podem possuir o mesmo valor para  $f(x)$ . Portanto, podem existir diversas soluções ótimas globais. É fácil observar que  $f(x) = 0$  é um limite inferior para a solução do problema e indicam que todas as restrições consideradas foram atendidas. No entanto, podem ocorrer situações onde não exista uma solução que atenda a todas as restrições consideradas.

$$\min f(x) = \sum_{k=1}^n P_k * x_k \quad (1)$$

## 4. Experimentos Computacionais

Com o objetivo de verificar a qualidade das soluções obtidas, nesta seção são apresentados os experimentos realizados com o algoritmo ILS proposto. A Seção 4.1 descreve a instância utilizada nos experimentos e as suas características. E na Seção 4.2 são apresentados os resultados obtidos.

### 4.1. Instância Utilizada

A instância utilizada é composta por dados reais do departamento do primeiro semestre de 2014 que foram fornecidas pelo chefe de departamento. Estes dados contêm: tipos de salas e suas respectivas quantidades; dados das disciplinas que serão oferecidas naquele semestre como carga horária, turno, tipo de sala, períodos e cursos que estão relacionadas; dados dos professores como preferência de horário e quais disciplinas irão lecionar.

Algumas características da instância real utilizada são apresentadas a seguir. Ao todo são 91 disciplinas relativas aos cinco cursos do departamento e algumas disciplinas que são oferecidas pelo outro departamento. Foram considerados 41 professores, três tipos de espaços físicos, em 16 são salas de aula, um Laboratório de Informática e um Laboratório de Física.

Estes dados foram organizados em 4 arquivos de entrada, em formato .csv, que possuem os dados necessários para construção de um quadro de horário utilizado pelo algoritmo, a saber. **Disciplina.csv** que possui a descrição da disciplina, períodos ao qual a disciplina está associada, tipo de sala requerido e carga horária; **DiscProf.csv** apresenta as disciplinas que serão lecionadas por cada professor; **DispProf.csv** contém dados sobre a disponibilidade e preferência dos professores; **MaxTipoSala.csv** referente aos tipos de salas existentes no instituto e suas respectivas quantidades.

### 4.2 Experimento

Os resultados dos experimentos foram obtidos em um notebook Intel Core i7 CPU 2.30GHz e 8GB de RAM, utilizando o sistema operacional Windows 8.1. O ILS foi implementado considerando o paradigma orientado a objetos e linguagem C++.

Para calcular o valor de cada solução obtida foram estabelecidas penalidades para as restrições descritas na Seção 2.1. Essas penalidades foram definidas em forma de pesos na função objetivo e seus valores são apresentados na Tabela 1. Os pesos para as restrições fortes e fracas possuem diferentes valores, sendo maiores nas restrições fortes. Os valores dos pesos utilizados foram obtidos através de testes empíricos. Sendo que, os pesos para o não atendimento das restrições fortes possuem os mesmos valores, pois todas devem ser atendidas com mesmo nível de prioridade. Já os valores dos pesos das restrições fracas variam de acordo com a importância (nível de prioridade) de cada restrição e foram definidas pelos professores do departamento. Portanto, para a obtenção dos valores dos pesos das restrições fracas levou-se em consideração a opinião dos professores do departamento estudado.



**Tabela 1 - Penalidades e pesos associados.**

Restrições / Penalidades		Pesos
<b>P1</b>	<b>Período</b>	15 por horário com conflito
<b>P2</b>	<b>Número máximo Salas</b>	15 por sala excedente
<b>P3</b>	<b>Professor Disciplina</b>	15 por horário com conflito
<b>P4</b>	<b>Professor Disponibilidade</b>	5 por horário alocado fora da disponibilidade
<b>P5</b>	<b>6h de Trabalho</b>	7 por dia que exceder
<b>P6</b>	<b>Janela</b>	2 por janela
<b>P7</b>	<b>Dias Consecutivos</b>	5 por dia ocioso
<b>P8</b>	<b>Atividades Isoladas</b>	2 por atividade

Neste experimento, para avaliar a qualidade de uma solução, foi considerada a Equação (1) com  $n = 8$ . Na realização do experimento foram considerados três grupos, cada um com um conjunto de restrições (penalidades), conforme ilustra a Tabela 2. Cada grupo possui um nível de dificuldade, conforme as restrições consideradas. Seguem os grupos:

**Tabela 2 - Penalidades por grupo.**

Grupo	P1	P2	P3	P4	P5	P6	P7
G1	X	X	X	X	X	X	X
G2	X	X	X	X	X		
G3	X	X	X				

O **Grupo 1 (G1)** apresenta nível de dificuldade Difícil. Neste grupo são verificadas todas as restrições (fortes e fracas) consideradas no presente trabalho. Vale destacar que as restrições fortes inviabilizam a solução enquanto as restrições fracas são relativas ao atendimento de pedidos que melhoram o quadro de horário dos professores e a qualidade de serviço. O **Grupo 2 (G2)** possui nível de dificuldade Médio. Neste grupo são verificadas as penalidades fortes, que inviabilizariam uma solução, e também as penalidades fracas relativas a preferência dos professores. O **Grupo 3 (G3)** pertence ao nível de dificuldade Fácil. Neste grupo são verificadas apenas as penalidades fortes, que inviabilizariam uma solução.

Neste experimento foi utilizada a instância descrita na Seção 4.1. Com o objetivo de calibrar os parâmetros de entrada, nos experimentos preliminares com a instância real foram realizadas 10 execuções utilizando diferentes quantidades de iterações do ILS (30, 40, 50, 60, 70 e 80). Os parâmetros utilizados na Busca Local para o movimento *MoveAtividade* foram: executar o dobro do número de professores; movimentos *IntraPeriodo* e *InterPeriodos* até 2 vezes sem melhoria na qualidade da solução.

**Tabela 3 - Resultados do experimento**

Instância	Penalidades								$f(x)$ da Melhor Solução			Tempo (s)
	P1	P2	P3	P4	P5	P6	P7	P8	G1	G2	G3	
PH2014_30	0	0	0	38	0	11	3	12	251	190	0	1773,0
PH2014_40	0	0	0	42	0	8	2	22	280	210	0	1972,0
PH2014_50	0	0	0	16	0	19	3	17	167	80	0	3848,0
PH2014_60	0	0	0	35	0	14	5	16	260	175	0	4515,0
PH2014_70	0	0	0	44	0	2	2	16	266	220	0	4829,0
PH2014_80	0	0	0	36	0	7	4	19	252	180	0	4907,0

Conforme pode ser observado na Tabela 3, a coluna instância é formada pelo nome da instância concatenado a quantidade de iterações realizadas pelo ILS. Todas as soluções apresentadas nessa tabela são soluções válidas, visto que todas as restrições fortes foram atendidas. Em outras palavras, considerando as restrições existentes para o grupo G3 foram encontradas soluções ótimas globais. Para os grupos G2 e G1 não foram encontradas soluções sem penalidades.

Pode-se observar que o algoritmo demandou no mínimo 1773 segundos (aproximadamente 29 minutos) para obter a solução. Nesse sentido, mesmo com o aumento na quantidade de iterações do ILS o algoritmo não obteve soluções de melhor qualidade. A execução com 50 iterações destacou-se ao obter a melhor solução (menor  $f(x)$ ).

Os quadros de horários gerados pelo algoritmo proposto foram comparados com o quadro de horário do departamento adotado no primeiro semestre de 2014. O departamento utilizou a ferramenta FET (*Free Timetabling Software*) [Liviu Lalescu 2016] para auxiliar na geração dos quadros de horários.

Para efeito de comparação, foram calculadas as penalidades e o custo de um quadro de horário gerado pelo FET para o departamento, utilizando-se as mesmas restrições para geração de penalidades e com os mesmos pesos associados proposto no trabalho. As penalidades e o custo do quadro de horário são apresentados na Tabela 4.

**Tabela 4 - Penalidades e custo do quadro de horário do departamento.**

Arquivo	Penalidades								$f(x)$ da Melhor Solução		
	P1	P2	P3	P4	P5	P6	P7	P8	G1	G2	G3
PH2014_FET	0	0	0	7	5	33	0	24	184	70	0

Uma solução de melhor qualidade com custo 167 foi encontrada pelo algoritmo proposto, uma vez que a solução gerada pelo FET (utilizada pelo departamento em 2014) teve um custo de 184. Apesar do tempo elevado de processamento, 64 minutos, os algoritmos propostos conseguiram obter resultados de melhor qualidade comparados com a ferramenta FET que é muito utilizada.

Embora experimentos computacionais indiquem resultados de alta qualidade, acredita-se que algoritmo pode-se tornar mais eficiente com a realização de ajustes nas estruturas de dados e otimizações nos métodos propostos. Devido à importância da geração de quadros de horários, que determina o dia-a-dia de professores e alunos, a distribuição de tarefas e o uso dos espaços físicos, é importante destacar que o tempo de processamento necessário para alcançar os resultados de melhor qualidade (cerca de uma hora) é plausível do ponto de vista operacional. Sendo assim, algoritmos apresentados neste trabalho são considerados eficazes para a resolução do PPHU.

## 5. Considerações Finais

A proposta deste trabalho foi desenvolver algoritmos para resolver o Problema de Programação de Horários de Cursos em Universidades (PPHU) baseados em currículos, aplicado a um departamento da Universidade Federal Fluminense (UFF). O algoritmo desenvolvido foi avaliado utilizando uma instância formada por dados reais de um departamento relativos ao primeiro semestre de 2014.

A fim de verificar a eficiência dos algoritmos propostos foram realizados experimentos com os dados reais de um departamento. Nos experimentos foram realizadas 10 execuções do algoritmo, combinando diferentes valores para o número de iterações do ILS. O algoritmo proposto encontrou soluções válidas nas diferentes combinações testadas, ou seja, todas as restrições fortes foram atendidas. E conseguiu encontrar uma solução de menor custo do que a adotada pelo departamento em 2014.

O tempo de processamento necessário para a obtenção do melhor resultado é plausível, dada à importância que um quadro de horário exerce sobre todos envolvidos no ambiente

educacional. A partir dos resultados computacionais apresentados considera-se que os algoritmos propostos são uma boa alternativa para resolução do PPHU, uma vez que apresentou resultados superiores aos obtidos como uso da ferramenta FET.

Como trabalhos futuros ainda podem ser realizadas algumas melhorias como a otimização das estruturas de dados e algoritmos visando à redução do tempo de processamento.

## Referências

Andrade, P. R. L. Otimização na Geração de Grade Horária Escolar através de um Modelo Matemático e das Meta-Heurísticas Busca Local e Iterated Local Search. 2014. 222 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Programa de Pós-graduação em Engenharia de Produção, Universidade Federal do Paraná, Curitiba, 2014.

Barbosa, S. H. D. Contribuições para a resolução do problema de programação de cursos universitários baseada em currículos via meta-heurísticas. 2012. 88 f. Dissertação (Mestrado) - Curso de Modelagem Matemática e Computacional, Programa de Mestrado em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2012.

Carvalho, A. S.; Mariano, G. P.; Kampke, E. H.; Mauri, G. R. Simulated Annealing Aplicado ao Problema de Programação de Horários do CCA-UFES. In: XVIII Simpósio de Pesquisa Operacional e Logística da Marinha, 2015, Rio de Janeiro. XVIII SPOLM, 2015. v. 1. p. 1-12.

Carvalho, R. Abordagem Heurística para o Problema de Programação de Horários de Cursos. 2011. 97f. Dissertação (Mestrado) – Curso de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, 2011.

Corrêa, M. V. Implementação de um sistema de alocação de professores e disciplinas em grades horárias utilizando algoritmos genéticos. 2010. 65 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Anhembi Morumbi, São Paulo, 2010.

ITC - International Timetabling Competition 2011. Disponível em <<http://www.utwente.nl/ctit/hstt/itc2011>>. Acesso em: 11 jul. 2015.

Liviu Lalescu. FET Free Timetabling Software. Disponível em: <<http://lalescu.ro/liviu/fet/>>. Acesso em: 22 fev. 2016.

Lourenço, H.R., Martin, O. and Stützle, T. (2010), Iterated Local Search: Framework and Applications. In Handbook of Metaheuristics, 2nd. Edition. Vol.146. M. Gendreau and J.Y. Potvin (eds.), Springer New York, International Series in Operations Research & Management Science, pp. 363-397.

Neukirchen, F. V. P.; Dorneles, A. P.; Weber, R. F.; Buriol, L. S. Um estudo de caso sobre a geração de quadros de horários no departamento de Ciência da Computação da UFRGS. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, XLVI, 2014, Salvador. Anais do XLVI SBPO. Salvador: Sobrapo, 2014. p. 3272 - 3279.

Oliveira, J. G.; Vianna, D. S.; Vianna, M. F. D. Uma Heurística GRASP+VND para o Problema de Programação de Horário Escolar. Sistemas & Gestão Revista Eletronica, N, v. 7, n. 3, p.326-335, nov. 2012. Disponível em: <<http://www.revistasg.uff.br/index.php/sg/article/viewFile/V7N3A3/V7N3A3>>. Acesso em: 20 mar. 2015.

PATAT - The International Series of Conferences on the Practice and Theory of Automated Timetabling. Disponível em <<http://www.patatconference.org/>>. Acesso em: 11 jul. 2015.

Santos, H. G. Formulações e Algoritmos para o Problema de Programação de Horários em Escolas. 2007. 74 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Computação, Universidade Federal Fluminense, Niterói, 2007.

Santos, H. G.; Souza, M. J. F. Programação de horários em instituições educacionais: Formulações e algoritmos. Relatório técnico, Universidade Federal do Rio de Janeiro, 2007.

Sousa, V. N.; Moretti, A. C.; Podestá, V. A. Programação da Grade de Horário em Escolas de Ensino Fundamental e Médio. Pesquisa Operacional, Rio de Janeiro, v. 28, n. 3, p.399-421, nov. 2008.

Temponi, E. C.C. Uma Proposta de Resolução do Problema de Corte Bidimensional via Abordagem Metaheurística. 70f. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte. 2007.