Problema de Alocação de Horários: um Estudo de Caso Utilizando o *Software* Livre FET

Bruno Marcelo Pena Barata
Ciência da Computação, Centro Universitário Serra dos Órgãos (UNIFESO), Teresópolis-RJ, Brasil
brbarata@gmail.com

Raphael Carlos de Medeiros
Ciência da Computação, Centro Universitário Serra dos Órgãos (UNIFESO), Teresópolis-RJ, Brasil
raphaelcmedeiros@hotmail.com

Carlos Eduardo Costa Vieira
Sistemas de Informação, Centro Universitário de Volta Redonda (UniFOA), Volta Redonda-RJ, Brasil
cadu.vieira@gmail.com

Júlio César da Silva
Ciência da Computação, Centro Universitário Serra dos Órgãos (UNIFESO), Teresópolis-RJ, Brasil e Mestrado Profissional em Educação Matemática, Universidade Severino Sombra (USS), Vassouras-RJ, Brasil
jcesarop@gmail.com

Resumo: *Um dos grandes problemas das Instituições de Ensino Superior brasileiras, ao iniciar o seu período letivo, é a programação dos horários das aulas. Este problema, abordado por diversos pesquisadores, se torna de difícil resolução devido ao grande número de possibilidades a serem analisadas e à necessidade de verificação de uma série de requisitos, muitos dos quais, conflitantes entre si, o que torna o espaço de busca vasto e altamente restrito. É uma tarefa difícil que demanda tempo e nem sempre consegue resolver o problema sem conflitos de disponibilidades dos docentes envolvidos. O objetivo deste artigo é propor a automatização da geração das grades horárias para o corpo docente do curso de Ciência da Computação do Centro Universitário Serra dos Órgãos (UNIFESO), utilizando o software livre Free Timetabling Program (FET), tornando o processo mais rápido, seguro e eficaz.*

Palavras-chave: *Alocação. Horários. Metaheurísticas. FET.*

Timetabling Allocation Problem:

a Case Study Using FET Free Software

Abstract: *The planning of class timetabling is a problem of Brazilian higher education institutions to start your school year. This problem, discussed by several researchers, it becomes difficult to resolve because of the large number of possibilities to be analyzed and the need to check a series of requirements, many of which conflict with each other, making the search space vast and highly restricted. It is a difficult task that takes time and does not always solve the problem without conflict availability of teachers involved. The object of this paper is to automate the generation of operating hours for the Computer Science course of University Center Serra dos Órgãos (UNIFESO), using some heuristics that solve this problem and the timetabling software Free Timetabling Program (FET), making the process fast, safe and effective.*

Keywords: *Allocation. Timetabling. Metaheuristics. FET.*

1. Introdução

O problema da construção de grades horárias em qualquer instituição de ensino se repete a cada semestre, ocasionado principalmente, pela indisponibilidade de horário dos professores e pelas aposentadorias, contratação de novos professores e aditivos de contratos.

A situação do curso de Ciência da Computação não é muito diferente deste cenário, pois a maior parte dos professores vem de outras cidades e não há um *software* que execute a montagem das grades horárias. O curso não possui qualquer sistema informatizado para resolver o problema de alocação de professores e gerar seu quadro de horários com facilidade e agilidade, de acordo com a disponibilidade diária e horária dos professores. As restrições são tantas, que o resultado implica dificuldades no cotidiano da Instituição durante todo o período, pois as soluções encontradas não atendem às expectativas, e todo esse trabalho é realizado manualmente.

O propósito do artigo é apontar uma solução para o problema de geração da grade horária do curso de Ciência da Computação, tendo em vista a situação em que o sistema atual se encontra. O resultado final visa a relação dos professores com suas respectivas matérias e períodos, de modo que não haja conflitos entre as partes envolvidas (matérias, disponibilidade de tempo, professores), (Souza, Maculan and Ochi 2000) (Souza, Maculan e Ochi 2001), que obedeçam a restrições que garantirão um melhor arranjo das aulas, e gerem um quadro de horários viável para o professor, e aloquem o máximo de matérias possíveis.

Para buscar este objetivo e automatizar a geração das grades horárias foi utilizado o *software* livre *Free Timetabling Program* (FET), que atendeu prontamente a todas as

iterações do processo, desde a fase de cadastro, até a geração definitiva das grades do corpo docente.

Este artigo está dividido como se segue: a Seção 2 descreverá o problema de alocação de horários de uma forma geral e o problema específico do curso de Ciência da Computação; a Seção 3 abordará algumas técnicas utilizadas para resolver problemas de otimização combinatória, entre os quais se enquadra o problema de alocação de horários; a Seção 4 apresentará o *software* FET utilizado para gerar as grades horárias; por último, a Seção 5 apresentará as considerações finais e propostas de trabalhos futuros.

2. Problema de Alocação de Horários

O problema de construção de grades horárias relaciona professores, alunos, matérias e horários semanais para a realização das aulas e podem ser de vários tipos, como horários de exames escolares, programação de horários de cursos, entre outros. Devido à tanta diversidade, o problema da alocação de horários em instituições de ensino superior é um dos mais completos, levando-se em consideração o número de variáveis e soluções possíveis.

A relação deste conjunto é exemplificada por $x = d*h$, donde, x são os horários semanais, d o número de dias da semana e h o número de horas de aula por dia (Souza 2000). De acordo com Even, Etail e Shamir (1976), é um problema classificado como NP-difícil, de extrema dificuldade de solução, pois a possibilidade de existirem algoritmos exatos que os resolvam em tempo de execução polinomial é muito pequena. O passo inicial a ser tomado é consultar a disponibilidade horária de professores e depois alocá-los nas turmas disponíveis, evitando, ao máximo, que as turmas fiquem desmoduladas, com tempos vagos e muitas aulas diferentes no mesmo dia, precavendo também que os professores ministrem poucas aulas no dia.

No curso de Ciência da Computação existem algumas restrições específicas que devem ser consideradas para a confecção da grade horária, como:

- Não permitir a alocação de mais de um professor em uma mesma turma e mesmo horário;
- Um professor não pode ser alocado a um horário no qual não esteja disponível;
- Evitar quebras de aulas, isto é, aulas não consecutivas de uma disciplina para uma turma em um mesmo dia;
- Respeitar o limite diário de aulas de uma mesma disciplina e turma;
- Eliminar, sempre que possível, os “buracos” nos horários dos professores;
- Atender ao maior número de aulas geminadas, ou seja, aulas realizadas em até dois horários consecutivos / dia, ou até quatro na mesma semana;
- Minimizar a quantidade de dias que cada professor necessita ir à faculdade.

3. Técnicas Utilizadas para Resolver o Problema de Alocação de Horários

A tarefa de conciliar todas as restrições citadas na Seção 2 não é nada simples. Alguns pesquisadores obtêm sucesso graças a técnicas de busca por soluções, cujo intuito é o de sempre obter àquelas satisfatórias num tempo razoável.

Independentemente da técnica utilizada é importante obedecer a algumas regras básicas, para que no momento da geração das grades horárias, não apareçam surpresas desagradáveis, como geralmente ocorre ao se utilizar o processo manual, em que muitas vezes é preciso desfazer todo o processo. Dentre as técnicas mais usuais, existem as de Algoritmos Exatos (AE) e os Algoritmos Heurísticos e Metaheurísticos.

3.1. Algoritmos Exatos ou de Programação Inteira

Algoritmos Exatos (AE) são modelos computacionais cuja característica é o tempo de execução/resposta, geralmente não satisfatório, mas em contrapartida, com a garantia de se encontrar uma solução ótima para o problema.

Quando se utiliza AEs para resolver problemas de alocação de horários, devem-se criar regras e restrições com bastante cautela, pois quaisquer mudanças nas regras são perigosas, o que ocasiona consequências em cascata. Por exemplo, ao final do processo de geração de grades, descobre-se alguma inconsistência nas regras, o que resulta em revisão de todo o processo, com demanda de tempo. Podem-se citar como exemplos de AE algoritmos do tipo *Branch-and-Bound*, *Branch-and-Cut*, *Branch-and-Price* etc. (Goldbarg e Luna 2005).

3.2. Algoritmos Heurísticos e Metaheurísticos

As técnicas heurísticas e metaheurísticas não garantem encontrar a solução ótima, porém apresentam tempos computacionais razoáveis. Dentre as metaheurísticas mais usuais existem o *GRASP*, Busca Tabu e Algoritmos Genéticos.

3.2.1. Greedy Randomized Adaptive Search Procedures (GRASP)

O método GRASP, Procedimento de Busca Adaptativa Gulosa e Randomizada, é um processo iterativo, em que cada um consiste de duas fases: de construção e a de busca local (Resende e Ribeiro 2010). Na fase de construção, executa-se a chamada busca gulosa, em que soluções são construídas elemento a elemento, seguindo-se algum critério heurístico de otimização, até que se tenha uma solução viável.

Uma solução viável é gerada iterativamente, elemento a elemento, até que a solução esteja completa, e gere uma lista, que, ordenada por uma função gulosa, mede o benefício que o mais recente elemento escolhido concede à parte da solução já construída.

Já a fase de busca local consiste da busca de um ótimo local na vizinhança da solução construída. A melhor solução encontrada ao longo das iterações GRASP é retornada como resultado (Ferreira e Glazar 2005). Portanto, este procedimento permite que diferentes soluções de boa qualidade sejam geradas.

3.2.2. Busca Tabu

A Busca Tabu é um procedimento heurístico proposto por Glover, para resolver problemas de otimização combinatória. Sua finalidade é evitar que a busca por soluções ótimas termine ao encontrar um mínimo local (Glover e Laguna 1997).

A partir da solução inicial gerada pelo procedimento construtivo, a metaheurística Busca Tabu faz uma busca agressiva no espaço de soluções, e procura àquelas melhores que a corrente com uma estrutura de memória que armazena os últimos movimentos realizados. A essa estrutura é dado o nome de lista *tabu*, porque os movimentos que se neles encontram são proibidos por um certo número de iterações. A lista *tabu* será útil quando, em algum momento, não for possível melhorar a solução corrente, o que caracteriza um ótimo local.

Nesse momento, o algoritmo terá que escolher a melhor solução, o que poderia ocasionar um retorno à solução anteriormente visitada, e gerada um ciclo. Como os últimos movimentos são proibidos pela lista *tabu*, o algoritmo deve prosseguir por outro caminho que ainda não foi visitado, aumentando o seu universo de busca (Ferreira e Glazar 2005) e (Oliveira 2003).

3.2.3. Algoritmos Genéticos

Algoritmos Genéticos (AGs) são métodos computacionais de busca baseados nos mecanismos de evolução natural e na genética. Em AGs, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua (Linden 2008).

Os Algoritmos Genéticos têm as seguintes características:

- Operam numa população (conjunto) de pontos, e não a partir de um ponto isolado;
- Operam num espaço de soluções codificadas, e não no espaço de busca direta;
- Necessitam somente de informação sobre o valor de uma função-objetivo para cada membro da população, e não requerem derivadas ou qualquer outro tipo de conhecimento;
- Utilizam transições probabilísticas e não regras determinísticas.

AGs são algoritmos iterativos e a cada iteração a população é modificada. A iteração de um AG é denominada uma geração, embora nem todos os indivíduos de uma população sejam necessariamente “filhos” de indivíduos da população na iteração anterior.

Na maioria das aplicações, a população inicial de N indivíduos é gerada aleatoriamente ou mediante algum processo heurístico. Como no caso biológico, não há evolução sem variedade, ou seja, a Teoria da Seleção Natural ou “lei do mais forte”, necessita que os indivíduos tenham diferentes graus de adaptação ao ambiente em que vivem. Assim, é importante que a população inicial cubra a maior área possível do espaço de busca. O fluxo básico de um algoritmo genético tem três operadores: Seleção, Cruzamento e Mutação (Linden 2008).

- Seleção: o mecanismo de seleção em AGs emula os processos de reprodução assexuada e seleção natural. Gera-se uma população temporária de N indivíduos extraídos com probabilidade proporcional à adequabilidade relativa de cada indivíduo na população. Neste processo, indivíduos com baixa adequabilidade terão alta probabilidade de desaparecerem da população, ou seja, serem extintos, ao passo que indivíduos adequados terão grandes chances de sobreviverem;
- Cruzamento: o processo de cruzamento é sexuado, ou seja, envolve mais de um indivíduo - que emula o fenômeno de "*crossover*", a troca de fragmentos entre pares de "cromossomos". Na forma mais simples, trata-se um processo aleatório que ocorre com probabilidade fixa que deve ser especificada pelo usuário;
- Mutação: este processo é uma forma de busca aleatória no "cromossomo", pois nele é capturada uma posição qualquer, e trocada por um possível valor para esse "gene". Logicamente, quando são tratados "cromossomos" com valores binários, se a posição capturada tiver o valor "1", muda para "0", e vice-versa. Como na natureza, ocorre com baixa probabilidade. Este operador é de extrema relevância, pois previne que todas as soluções do problema sejam combinações das soluções encontradas a partir da população inicial, o que aumenta ainda mais a diversidade de soluções.

O Algoritmo Genético precisa de uma condição de finalização de seu processo, (condição de parada), ou seja, uma condição, um valor, ou uma variável que faça com que as iterações cheguem ao fim, podendo ser o tempo, estagnação da população, o número de gerações, entre outras.

4. *Free Timetabling Program* (FET)

É importante destacar que a elaboração da grade horária de todos os oito períodos do curso de Ciência da Computação do UNIFESO é realizada de forma manual, com utilização do *software* MS Word, o que demanda dias ou semanas principalmente, quando imprevistos acontecem (contratação, desligamento ou não disponibilidade de professores, entre outros).

Pensando nestas dificuldades e mediante muitos estudos e testes, foi escolhido o *software* livre *Free Evolutionary Timetabling* (FET) para realizar a geração das grades horárias do corpo docente. O FET é um programa que foi desenvolvido por *Liviu Lalescu*, para resolver problemas de compromissos, tarefas, horários de escolas etc. O FET é um *software* bastante completo, traduzido para diversos idiomas, inclusive o português, e tem como método de busca a metaheurística Algoritmos Genéticos, que combinam resultados e executam mutações, a fim de retornar a melhor solução (FET 2010).

A versão utilizada neste artigo é a 5.13.4, de 13 de junho de 2010, que possibilita inúmeras configurações, desde o cadastro das variáveis (professores, matérias, períodos etc.), até manipulações como horário de início/fim de cada aula, número de aulas que cada professor leciona, e inclusive sua disponibilidade na instituição. É possível ainda, dividir o número de aulas por dia e por semana, manejar matérias pré-requisitos de outras, e tantos recursos adicionais de cadastro, como por exemplo, os de salas e laboratórios.

Qualquer disciplina alocada num horário em que o professor não esteja disponível ou se forem alocados mais de um professor para lecionar disciplina comum, o *software* automaticamente acusa o erro, e exibe mensagem na tela, o que para imediatamente as iterações da geração da grade horária. Assim, o usuário pode corrigir o erro imediatamente, ao contrário da elaboração manual, cuja inconsistência seria "possivelmente" notada no final do processo. Com o FET é possível descrever o número de alunos em cada turma e horário de início e término das aulas. A Figura 1 mostra a tela inicial do FET.

A Figura 2 ilustra a tela de alocação de disciplinas, em que foi implementado um atalho, possibilitando encontrar as principais ferramentas do FET. A Figura 3 apresenta uma das mais importantes telas do FET, se realizado todo o processo de relacionamento de professores, disciplinas, períodos e tempos de aula. Uma vez feito todo o cadastro de disciplinas, professores, horários e dias da semana corretamente, o sistema se encarrega de gerar as grades horárias automaticamente. O FET possibilita a exportação das grades horárias para relatórios XML (*Extensible Markup Language*) ou HTML (*HyperText Markup Language*). A Tabela 1 apresenta um demonstrativo sobre a comparação do processo manual, feito hoje em dia, com o processo automatizado, realizado no FET. Pode-se observar pela tabela como os resultados são bem contrastantes.

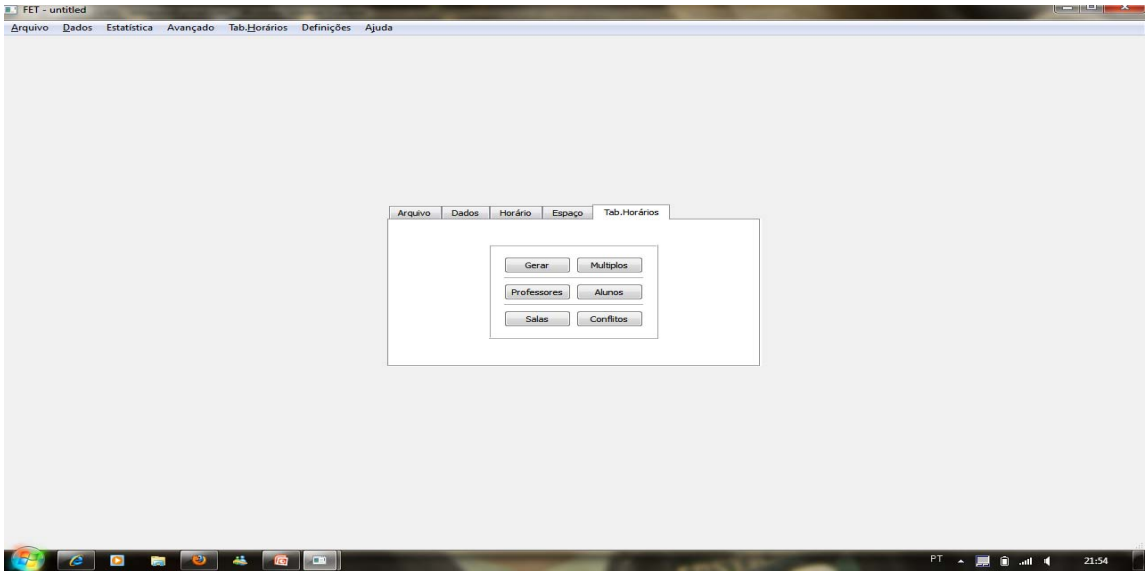

Figura 1. Tela inicial do FET 5.13.4.

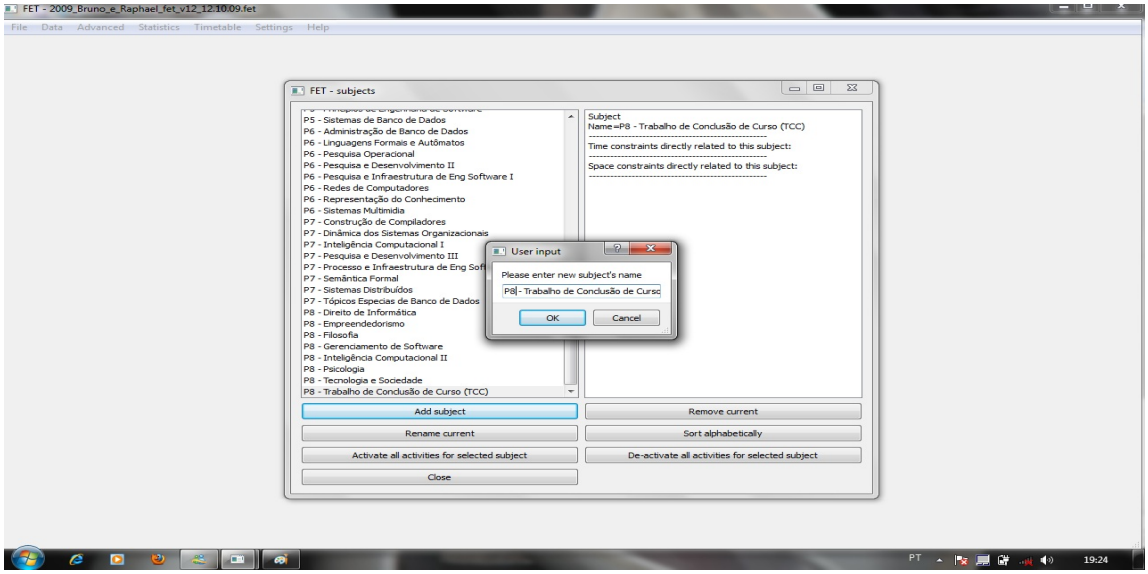

Figura 2. Alocação de matérias no FET 5.13.4.

Tabela 1. Comparação entre o processo manual e informatizado.

Quesito	Processo Manual (Atual)	Processo Informatizado (*FET*)
Tempo gasto	Alguns dias	Poucos segundos
Qualidade da solução	Satisfatória	Boa
Facilidade de alteração	Difícil ou Impossível	Fácil
Dependência do usuário	Total	Somente para cadastro
Praticidade	Pouca ou Nenhuma	Muita

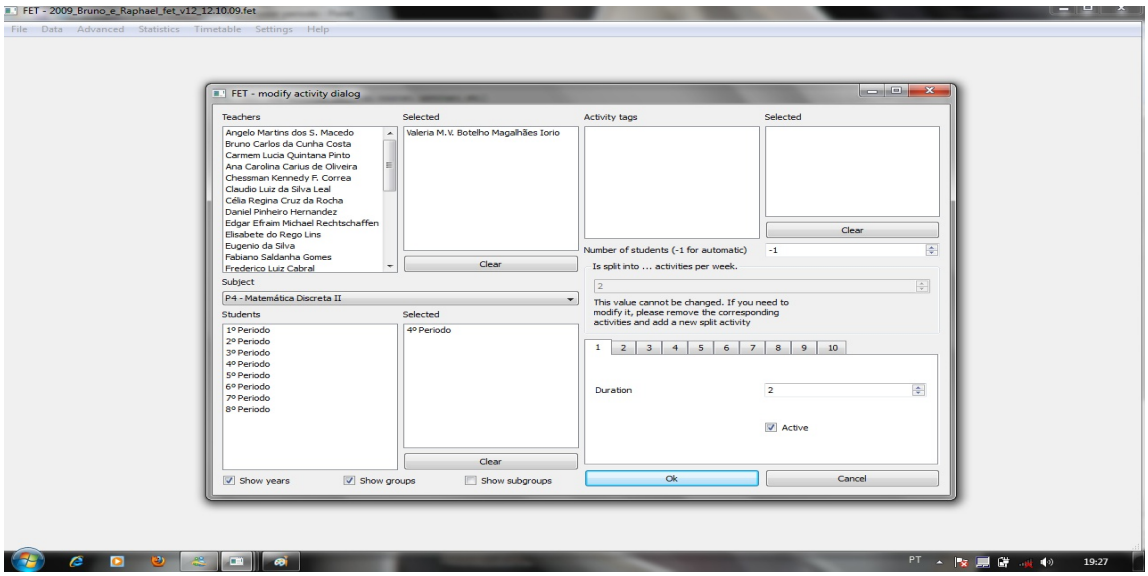

Figura 3. Relação de matérias com respectivos professores e horários.

5. Considerações Finais

Apesar de ser um tema bastante complexo, o problema de automatização de grades horárias é bastante viável de ser entendido e implementado, por meio de metaheurística como as que foram apresentadas, de modo a gerar boas soluções para o problema de alocação de horários.

Durante todo o período de testes e simulações reais, o *software* FET impressionou, por se mostrar rápido, coeso e eficiente e provar que pode ser utilizado na geração das grades horárias do corpo docente do curso de Ciência da Computação. Além disso, é um sistema bastante flexível, que pode ser utilizado em computadores com configurações de *hardware* e *software* desde as mais inferiores, até as mais atuais.

A redução do tempo gasto foi a principal vantagem do FET, pois foram necessários somente alguns dias para deixar o banco de dados totalmente acertado. Após a finalização do cadastro, a geração das grades horárias para o corpo docente foi executada automaticamente pelo *software*. Por se tratar de um sistema multiplataforma, o FET pode ser executado sem problemas numa máquina *Pentium®/AMD®* 266Mhz, com 256Mb de memória.

Concluímos que podem ser apontadas as seguintes vantagens do FET, em relação ao processo executado de forma manual:

- Economia de tempo, custo e pessoal, para geração das grades horárias;
- Possibilidade de modificar os resultados obtidos em qualquer momento;
- O fornecimento de informações mais detalhadas e interação com os usuários do seu programa.

Como propostas de trabalhos futuros sugerem-se a utilização dos demais recursos do *software* FET, como as restrições de disponibilidade de locais (salas e laboratórios), e a divulgação do sistema aos alunos, para que possam gerar suas grades no FET.

Referências

Even, S., Itai, A. and Shamir, A. (1976). On the complexity of timetabling and multicommodity flow problems. *SIAM Journal of Computation*, 5(4), 691-703.

Ferreira, J. C. dos S e Glazar, J. E. **(**2005). *Definição de parâmetros na utilização de meta-heurísticas para a programação de horários escolares. Revista Educação e Tecnologia.* 1(1), 1-11.

FET. (2010). "Free Evolutionary Timetabling", endereço: http://lalescu.ro/liviu/fet/, Acesso em 03/10/2010.

Glover, F. and Laguna, M. (1997). *Tabu Search, Kluwer Academic Publisher*s.

Godbarg, M. C. e Luna, H. P. L. (2005). Otimização Combinatória e Programação Linear: Modelos e Algoritmos, *Elsevier,* 2. ed. rev. e atual.

Linden, R. (2008). *Algoritmos Genéticos,* Brasport, 2. ed..

Oliveira, J. A. Construção de Tabela de Horário Escolar na Web. 2003. 84 f. *Dissertação (Mestrado em Informática Aplicada) – Curso de Ciência da Computação,* Universidade de Fortaleza, Fortaleza, 2003.

Resende, M. G. C. and Ribeiro, C. C. (2010) "GRASP", In: *Search Methodologies, Edited by E.K. Burke e G. Kendall, Springer,* 2nd edition, 1-25.

Souza, M. J. F. Programação de Horários em Escolas: Uma Aproximação por Metaheurísticas. (2000). 160 f. *Tese (Doutorado em Engenharia de Sistemas e Computação)* – Programa de Pós-Graduação em Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2000.

Souza, M. J. F., Maculan, N. e Ochi, L. S., (2000). Melhorando Quadros de Horários através de Caminhos Mínimos. *Tendências em Matemática Aplicada e Computacional.* 1(2), 515-524.

Souza, M. J. F., Maculan, N. e Ochi, L. S., (2001). Uma Heurística para o Problema de Programação de Horários em Escolas. *Tendências em Matemática Aplicada e Computacional*. (2), 213-222.