

Uma Heurística para o Problema de Programação de Horários em Escolas ¹

M.J.F. Souza², Departamento de Computação, Universidade Federal de Ouro Preto, 35400-000 Ouro Preto, MG, Brasil.

N. Maculan³, Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Cx.P. 68511, 21945-970 Rio de Janeiro, RJ, Brasil.

L.S. Ochi⁴, Departamento de Ciência da Computação, Universidade Federal Fluminense, Rua Passo da Pátria, 156, 24210-240 Niterói, RJ, Brasil.

Resumo. Apresentamos GBT-II, uma heurística para resolver problemas de programação de horários em escolas. A partir de uma solução inicial gerada por um procedimento construtivo parcialmente guloso, um procedimento de Busca Tabu inicia sua pesquisa. Quando uma solução sem sobreposições, mas com, possivelmente, algum outro tipo de inviabilidade, é gerada, acionamos um procedimento chamado Intraturmas-Interturmas para tentar recuperar a viabilidade. Sendo bem sucedido, ele é novamente acionado, agora para tentar melhorar a compactidade da agenda dos professores, bem como outras medidas de qualidade. Após um certo número de iterações sem melhora, todo o processo, desde a geração de uma solução inicial, é repetido até que uma condição de parada seja satisfeita. Resultados computacionais mostram que a utilização do procedimento Intraturmas-Interturmas faz produzir soluções melhores mais rapidamente.

Abstract. We present GBT-II: an heuristic approach for solving school timetabling problems. From an initial solution produced by a partially greedy constructive procedure, a tabu search procedure starts its search. When a solution without overlapping but possibly with other types of infeasibilities is generated, we activate a procedure called Intraturmas-Interturmas (II) in order to try to retrieve the feasibility. If it is successful it is activated again, now in order to try to improve the compactness of the timetable, as well other quality requirements. After a number of iterations without improve, the whole process, including the construction of a new initial solution, is repeated until a given condition is satisfied. Computational results show that (II) speeds the process of obtaining good solutions.

¹Trabalho parcialmente financiado pelo PICDT/CAPES e desenvolvido no Programa de Engenharia de Sistemas e Computação/COPPE/UFRJ.

²marcone@iceb.ufop.br, marcone@cos.ufrj.br

³maculan@cos.ufrj.br

⁴satoru@dcc.ic.uff.br

1. Introdução

O problema de programação de horários em escolas (PPHE) trata da alocação de professores a turmas satisfazendo certas condições. É um problema de difícil generalização, em virtude da diversidade de regimes educacionais e das características de cada instituição de ensino, o que torna cada caso praticamente ímpar. É também um problema NP-difícil [6], o que justifica sua abordagem por técnicas heurísticas.

Entre as técnicas recentes, usadas com sucesso para resolver o problema, destacamos as metaheurísticas: Busca Tabu [10, 5, 2], *Simulated Annealing* [1], Algoritmos Genéticos [4], etc. Uma deficiência, no entanto, dessas aplicações é que elas, em geral, utilizam um mesmo mecanismo de busca durante todo o processo. Por exemplo, em [1] e [5] utiliza-se uma vizinhança *1-optimal*, enquanto em [10] a vizinhança é *2-optimal*.

Desenvolvemos uma técnica de busca local, denominada Intraturmas-Interturmas (II), a qual é inserida na fase de refinamento de um algoritmo GRASP [7]. Esse procedimento tenta melhorar uma solução em 2 etapas: inicialmente, procurando recuperar a viabilidade e depois procurando melhorar requisitos de qualidade do quadro de horário. O procedimento II baseia-se em movimentos dos horários das aulas das turmas, sendo esses movimentos do tipo *k-optimal*, $k \geq 2$, com relação à vizinhança de um quadro de horário de uma dada turma. Ele gera, a partir de uma solução sem sobreposições, soluções melhores através da detecção de ciclos de custo negativo nos grafos associados aos quadros de horário de cada uma das turmas.

2. Descrição do Problema

O problema de programação de horários em escolas (PPHE) abordado consiste em um conjunto de m professores, n turmas, s matérias e p horários semanais reservados para a realização das aulas. Os p horários semanais são distribuídos em d dias da semana de h horários diários realizados em um único turno, isto é, $p = d * h$. As turmas, as quais estão sempre disponíveis, são conjuntos disjuntos de estudantes que cursam as mesmas matérias. A cada matéria de uma turma está associado um único professor, previamente fixado. Além disso, a carga horária de cada turma é exatamente p . Os seguintes requisitos devem ser satisfeitos:

- (a) um professor não pode ministrar aula para mais de uma turma ao mesmo tempo;
- (b) uma turma não pode ter aula com mais de um professor em um mesmo horário;
- (c) cada professor tem que cumprir sua carga horária semanal;
- (d) um professor não pode ser alocado a um horário no qual não esteja disponível;
- (e) uma turma não pode ter mais do que 2 horários diários de aula de uma mesma matéria;
- (f) atender, tanto quanto possível, ao pedido de aulas duplas (aulas realizadas em dois horários consecutivos) dos professores que as requererem;
- (g) a agenda dos professores deve ser tão compacta quanto possível.

Definição 2.1: Um quadro Q de horário de professores que não satisfaz a pelo menos um dos requisitos (a), ..., (e) é considerado **inviável** e não pode ser praticado pela escola.

Definição 2.2: Um quadro Q se diz inviável do tipo 1 se, em pelo menos um horário, uma das condições a seguir for satisfeita: (i) Existe turma tendo aula com mais de um professor (restrição (b) não é verificada); (ii) Existe turma sem aula;

Definição 2.3: Um quadro Q se diz inviável do tipo 2 se a restrição (e) não for satisfeita para algum professor;

3. Representação do Problema

Um quadro de horário de professores é representado como uma matriz $Q_{m \times p}$ de valores inteiros, onde em cada linha i de Q representa-se a alocação semanal do professor i . Cada elemento $q_{ik} \in \{-1, 0, 1, 2, \dots, n\}$ indica a atividade do professor i no horário k . Valores negativos indicam que o professor está indisponível, enquanto valores nulos indicam inatividade no horário.

Um movimento consiste na simples troca de dois valores distintos e não negativos de uma dada linha de Q . Tal movimento é identificado pela tripla $\langle i, k, \bar{k} \rangle$, onde k e \bar{k} representam os horários nos quais as atividades q_{ik} e $q_{i\bar{k}}$ do professor i serão permutadas.

Observamos que esse tipo de movimento pode produzir inviabilidade do tipo 1 ou 2. Entretanto, a possibilidade de que um professor ensine simultaneamente para mais de uma turma (violação à restrição (a)) é automaticamente rejeitada pela representação escolhida.

4. A Função Objetivo

Um quadro Q de horário é avaliado com base na seguinte função objetivo, baseada em penalidade, a qual deve ser minimizada:

$$f(Q) = \omega * f_1(Q) + \delta * f_2(Q) + \rho * f_3(Q) \quad (4.1)$$

onde as duas primeiras componentes mensuram o nível de inviabilidade dos tipos 1 e 2, respectivamente, e a terceira, o nível de satisfação dos professores com relação ao atendimento de seus requisitos pessoais.

Os pesos ω , δ e ρ são escolhidos satisfazendo a condição: $\omega > \delta \gg \rho$. Evidentemente, de acordo com a definição 2.1, Q é viável se $f_1(Q) = f_2(Q) = 0$.

O nível de inviabilidade do tipo 1 de Q , $f_1(Q)$, é mensurado somando-se, para cada horário k : (a) o número de vezes l_k que uma turma está sem atividade em k ; (b) o número de vezes s_k que mais de um professor dá aula para uma mesma turma no horário k .

Com relação ao nível de inviabilidade do tipo 2, o quadro Q é avaliado somando-se o número de vezes e_i que a restrição (e) não é atendida para cada professor i .

A satisfação ao atendimento dos requisitos pessoais dos professores é medida com relação à compacidade do quadro de horário (restrição (g)), bem como ao atendimento do número de aulas duplas requeridas (restrição (f)). Mais precisamente,

$$f_3(Q) = \sum_{i=1}^m (\alpha_i * b_i + \beta_i * v_i + \gamma_i * c_i) \quad (4.2)$$

onde α_i , β_i e γ_i são pesos e refletem, respectivamente, a importância relativa do número de buracos b_i (horários ociosos entre dois horários de aula de um mesmo turno), do número de dias v_i na semana que cada professor está envolvido em alguma atividade de ensino em um mesmo turno e da diferença não-negativa c_i entre o número mínimo requerido de aulas duplas e o efetivamente existente na agenda corrente de cada professor i . ($c_i = \max \{0, \text{duplas}(Q_i^{\text{requerido}}) - \text{duplas}(Q_i^{\text{corrente}})\}$)

5. O Algoritmo GBT-II

O Algoritmo GBT-II é um procedimento GRASP [7], onde uma solução inicial é gerada por um procedimento construtivo parcialmente guloso (conforme seção 6.) e o refinamento é feito através de um método de Busca Tabu (seção 7.). Quando uma solução sem inviabilidade do tipo 1 é gerada, o método BT aciona o procedimento Intraturmas-Interturmas (descrito na seção 8.). Esta sequência de construção e refinamento é repetida por GBTmax iterações.

6. Geração de uma Solução Inicial

A solução inicial é gerada de forma construtiva, via um procedimento parcialmente guloso, conforme a seguir se descreve.

Inicialmente, determinamos os horários mais críticos, isto é, aqueles que têm o menor número de professores disponíveis. A seguir, enumeramos e ordenamos todas as aulas, priorizando aquelas mais difíceis de serem alocadas, ou seja, aquelas que têm os professores com maior carga horária e maior número de horários indisponíveis. Formamos, então, uma lista restrita de candidatos dessas aulas, de tamanho $|RCL|$, e selecionamos, aleatoriamente, uma delas. Em seguida, procuramos alocá-la (usando a ordem de horários críticos) de forma que não haja, a princípio, nenhum tipo de inviabilidade (no caso, violação às restrições (b) e (e)). Não sendo possível, admitimos, agora, violação somente à restrição (e). Se ainda assim a alocação não for possível, a aula é alocada admitindo-se violação também à restrição (b). Toda vez que uma aula é alocada, atualizamos os horários críticos, bem como as aulas mais difíceis remanescentes.

Como todas as aulas são alocadas, ainda que com algum tipo de inviabilidade, a restrição (c) é automaticamente assegurada.

7. Busca Tabu

Partindo de uma solução inicial, gerada pelo procedimento construtivo, a metaheurística de Busca Tabu (BT) segue, iterativamente, explorando toda a vizinhança $N(Q)$ da solução corrente Q , através de movimentos definidos na seção 3, e guiados pela função objetivo expressa em 4.1. O algoritmo segue, então, para o vizinho Q' que produzir o menor valor $f(Q')$, independentemente de ele ser pior que o valor $f(Q)$ corrente. De forma a prevenir a ocorrência de ciclagem, a cada vez que um movimento é realizado, ele é armazenado em uma lista tabu T . Essa lista tabu contém os $|T|$ movimentos mais recentes e reduz o risco de se visitar uma das $|T| - 1$ últimas soluções visitadas anteriormente. Como a lista tabu pode ser muito restritiva [8], o algoritmo BT usa também uma função de aspiração $A(f(Q))$, onde $A(f(Q)) = f(Q^*) - 1$. Dessa forma, um movimento perde seus *status* tabu se produzir uma solução Q' cujo valor seja inferior ao da melhor solução, Q^* , obtida até então. Sempre que uma solução sem sobreposições é gerada (isto é, $f_1(Q) = 0$), acionamos o procedimento Intraturmas-Interturmas, com a condição de que uma solução de mesmo valor ainda não tenha sido submetida a II. Os movimentos reversos aos movimentos realizados pelo procedimento II são introduzidos na Lista Tabu. O procedimento BT continua sua busca a partir da solução produzida por II e é executado por *BTmax* iterações sem melhora no valor da melhor solução.

8. Procedimento Intraturmas-Interturmas

Intraturmas-Interturmas (II) é uma extensão do procedimento descrito em [2, 12]. Ele é acionado quando uma solução sem inviabilidade do tipo 1 está disponível. Primeiramente, ele procura zerar a parcela f_2 da função objetivo. Se bem sucedido, ele tenta, em uma segunda etapa, melhorar a componente f_3 respeitando as restrições (a), ..., (e), isto é, navegando no espaço das soluções viáveis. Como a forma de atuação de II é semelhante nos dois casos, mostraremos seu princípio de funcionamento apenas para o segundo caso.

Suponhamos, então, disponível uma solução sem inviabilidade. Assim, dado um quadro Q de horário de professores nessas condições ($f_1(Q) = f_2(Q) = 0$), definimos o grafo da turma j por $G_j = (V_j, A_j)$, onde V_j é o conjunto dos horários reservados para a turma j e A_j é o conjunto dos arcos orientados definido conforme a seguir:

$A_j = \{ (k, \bar{k}) : \text{a aula da turma } j \text{ lecionada no horário } k \text{ pode ser lecionada também no horário } \bar{k} \}$

Da construção do grafo G_j deduz-se que para um arco orientado (k, \bar{k}) pertencer ao conjunto A_j as seguintes condições precisam ser atendidas: (i) o professor do horário k precisa estar disponível no horário \bar{k} ; (ii) as restrições relativas à aula do horário k precisam estar, também, satisfeitas no horário \bar{k} .

A cada arco $(k, \bar{k}) \in G_j$ associamos um custo $\Delta f_i(k, \bar{k})$, o qual representa a variação do custo de se transferir o professor i do horário k para o horário \bar{k} , tendo em vista a componente f_3 da função objetivo. Dessa forma, o custo é obtido calculando-se a diferença entre os valores da função objetivo, relativa ao professor,

nas configurações nova e antiga, isto é:

$$\Delta f_i(k, \bar{k}) = f_i(\bar{k}) - f_i(k) \quad (8.1)$$

sendo $f(\cdot) = (\rho * f_3)(\cdot)$

A Tabela 1 mostra um fragmento de um quadro Q_1 de horário de professores. Cada linha i representa um professor ($i = P1, P2, P3, P4$) e cada coluna k um horário ($k = H1, H2, H3, H4, H5$) de um mesmo dia. Cada elemento (i, k) dessa tabela representa a atividade do professor i no horário k . A, B, C e D são turmas. Um traço (-) significa que o professor está indisponível, enquanto uma célula vazia indica que não há atividade no horário. A coluna f_i indica o valor da função objetivo de cada professor, expressa em 4.1 e 4.2, com $\rho = 1$, $\alpha_i = 1$ e $\beta_i = \gamma_i = 0 \forall i$. Para esse quadro de horário, tem-se $f(Q_1) = f_{P1} + f_{P2} + f_{P3} + f_{P4} = 1 + 1 + 0 + 0 = 2$.

Tabela 1: Quadro Q_1

	H1	H2	H3	H4	H5	f_i
P1	A		B	B		1
P2	B	C		A	A	1
P3		B	A	C	B	0
P4	C	A	C	D	-	0

A Figura 1 representa G_A , o grafo da turma A. Cada horário é representado por um vértice, ao qual está associado um professor. O arco $(H1, H5)$ de custo -1 indica que se o prof. $P1$ mudar sua aula do horário $H1$ para o horário $H5$, haverá uma diminuição no valor da função objetivo de 1 unidade ($\Delta f_{P1}(H1, H5) = f_{P1}(H5) - f_{P1}(H1) = 0 - 1 = -1$).

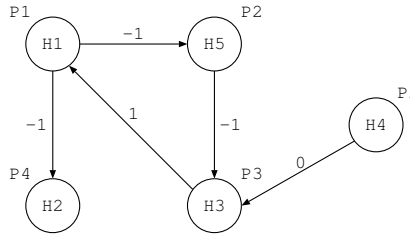


Figura 1: G_A , Grafo da Turma A

Para encontrar um quadro de horário com um valor menor para a função objetivo, é suficiente procurar um ciclo de custo negativo em G_j .

No exemplo considerado, a sequência de arcos $\{(H1, H5), (H5, H3), (H3, H1)\}$ forma um ciclo de custo total -1 ($= -1 + (-1) + 1$). Tal sequência define um conjunto de movimentos *intraturmas*.

Atualizado o grafo G_A e o quadro Q_1 com esse tipo de movimento, procuramos novamente pela existência de ciclos de custo negativo no grafo resultante da turma

A. Não existindo mais tais ciclos, a idéia, então, é repetir o procedimento para uma outra turma e assim sucessivamente até que não seja possível melhorar o quadro de horário dos professores através de movimentos intraturmas.

A existência de um ciclo de custo negativo pode, todavia, não garantir melhora no valor da função objetivo, assim como pode gerar soluções inviáveis do tipo 2, conforme mostrado em [12]. Essas situações, entretanto, só ocorrem quando um mesmo professor participa em mais de um arco no ciclo. Assim, faz-se necessário checar a viabilidade e o valor da função objetivo após os movimentos candidatos. De forma a encontrar outros ciclos de custo negativo em um grafo nessas condições, procedemos como segue: Escolhemos um arco qualquer do ciclo, $(k, \bar{k}) \in G_j$, e o inserimos em uma lista L de movimentos proibidos. A seguir, atualizamos o grafo da turma sob avaliação, excluindo de G_j os arcos pertencentes à L , e procuramos outro ciclo de custo negativo. Quando não mais for possível encontrar ciclos de custo negativo, passamos para uma outra turma e zeramos a lista L .

Ao final do procedimento Intraturmas podem restar, ainda, arcos de custo negativo nos grafos das turmas. A Figura 2, que considera os grafos de duas turmas, j e \bar{j} , ilustra tal situação.

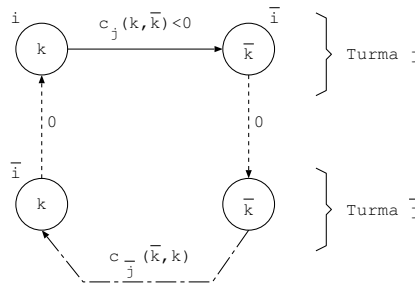


Figura 2: Grafos das turmas j e \bar{j} resultantes do procedimento Intraturmas

Nesta figura, há um arco de custo negativo na turma j , de k a \bar{k} , a saber $c_j(k, \bar{k})$, sinalizando que poderá haver uma melhora no valor da função objetivo se o professor i tiver sua aula do horário k transferida para o horário \bar{k} . Entretanto, essa transferência não pode ocorrer porque o professor \bar{i} , do horário \bar{k} , não está disponível no horário k (Nesse horário, ele está lecionando para a turma \bar{j}). A idéia, então, é a de trocar o horário dessas aulas do professor \bar{i} , de forma a permitir a procura de um ciclo de custo negativo envolvendo os grafos das duas turmas, conectados nos horários k e \bar{k} . Seja $c_{\bar{j}}(\bar{k}, k)$ o custo do caminho mínimo de \bar{k} a k em $G_{\bar{j}}$. A existência de um ciclo de custo negativo envolvendo as duas turmas pode ser observada verificando se a condição $c_j(k, \bar{k}) + c_{\bar{j}}(\bar{k}, k) < 0$ é satisfeita ao transferir o professor \bar{i} da turma j para a turma \bar{j} no horário \bar{k} e da turma \bar{j} para a turma j no horário k (arcos de custo nulo da figura 2, já que não há custos envolvidos na transferência de um professor de uma turma para outra em um mesmo horário).

Assim, para cada arco $(k, \bar{k}) \in G_j$, de custo negativo, pesquisamos a existência de um ciclo de custo negativo envolvendo esse arco de G_j , o grafo $G_{\bar{j}}$ e os arcos de

custo nulo que os conectam. Esta sequência de arcos define os chamados movimentos *interturmas*. Assim como em Intraturmas, faz-se necessário checar a viabilidade e o valor da função objetivo depois dos movimentos candidatos.

O procedimento Intraturmas-Interturmas consta, portanto, de duas etapas. Na primeira, aplicamos o procedimento Intraturmas, resultando n grafos G_j , $\forall j = 1, \dots, n$, com, possivelmente, arcos de custo negativo. Encerrada a primeira etapa, aplicamos o procedimento Interturmas.

Quando o procedimento II é acionado para tentar recuperar a viabilidade de uma solução que tem inviabilidade apenas do tipo 2, o custo de um arco (expressão 8.1) é avaliado considerando-se $f(\cdot) = f_2(\cdot)$. Encerrada essa tentativa, o custo do arco volta a ser avaliado com base na variação de custo da função 4.1.

9. Resultados Computacionais

Apesar de o PPHE ser um problema clássico da área de otimização, ainda não existe, de nosso conhecimento, um conjunto consagrado de problemas-teste com as características descritas na seção 2., o que se deve, em parte, à dificuldade de generalização do problema. No entanto, um esforço no sentido de reunir e padronizar problemas-teste nessa área está sendo feito pelo grupo WATT (*WorkGroup on Automated TimeTabling*, <http://www.asap.cs.nott.ac.uk/asap/watt>).

Sendo assim, para testar o algoritmo proposto, utilizamos 7 problemas-teste, sendo os dois primeiros extraídos de [9], enquanto os demais referem-se ao planejamento de aulas da Escola Estadual Dom Silvério, situada em Mariana (MG). Em todos os problemas são reservados $p = 25$ horários para cada turma.

O algoritmo GBT-II foi implementado na linguagem C e testado em um microcomputador Pentium II, de 350 MHz e 64 MB RAM, sob sistema operacional linux. Para determinar ciclos de custo negativo implementamos o Algoritmo de *Floyd* [3].

De forma a testar a eficácia do procedimento Intraturmas-Interturmas, comparamos o algoritmo proposto (GBT-II) com o algoritmo GBT, o qual não inclui tal procedimento. Para todos os problemas-teste tomamos os seguintes parâmetros, os quais foram calibrados tendo em vista a aplicação do algoritmo GBT na instância definida pelo problema 4: $|RCL| = \max\{1, (\#aulas \text{ ainda não alocadas})/10\}$, $BTmax = 500$, $GBTmax = 1$, $\omega = 100$, $\delta = 30$, $\rho = 1$, $\alpha_i = 3$, $\beta_i = 9$ e $\gamma_i = 1 \forall i = 1, \dots, m$.

A Tabela 2 apresenta algumas características dos problemas testados e sumariza os resultados obtidos. As colunas m , n e $\#aulas$ representam, respectivamente, o número de professores, turmas e aulas a serem alocadas. A coluna dens indica a densidade do quadro de horário, isto é, a razão entre a carga horária dos professores (somada ao número de horários indisponíveis) e o número de horários reservados para as aulas de todas as turmas. Os resultados da Tabela 2 são baseados em 25 execuções, cada qual iniciada com uma semente diferente de números aleatórios. Em cada execução, no entanto, a semente era a mesma para os dois algoritmos. Para cada conjunto de 25 execuções mostramos o valor médio da melhor solução $[f^*]$ e o tempo médio de CPU, no formato minuto:segundo.

Tabela 2: Resultados Computacionais

Prob.	m	n	#aulas	dens	GBT		GBT-II	
					$ f^* $	tempo	$ f^* $	tempo
1	8	3	75	0.57	204	00:28	203	00:25
2	14	6	150	0.50	368	01:53	356	01:41
3	16	8	200	0.70	491	02:17	481	01:24
4	23	12	300	0.87	749	06:01	741	06:02
5	31	13	325	0.42	831	15:36	826	08:09
6	30	14	350	0.48	847	16:10	810	09:40
7	33	20	500	0.61	1164	38:45	1101	29:00

De forma a mostrar a influência da inserção de II no algoritmo GBT, apresentamos, pela Figura 3, a evolução da melhor solução do problema 6 nos segundos iniciais de uma iteração GRASP. Apesar de específico ao problema 6, esse comportamento reflete o caso médio.

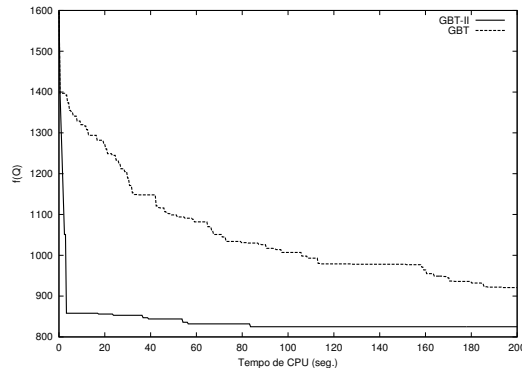


Figura 3: Comportamento típico dos algoritmos

10. Conclusões

Apresentamos uma metaheurística híbrida para resolver problemas de programação de horários em escolas. Enquanto o procedimento construtivo parcialmente guloso gera boas soluções iniciais e diversifica a busca, o procedimento de Busca Tabu (BT) navega, a partir delas, pelo espaço de soluções. Quando BT consegue gerar uma solução sem sobreposições, o procedimento Intraturmas-Interturmas (II) é acionado, visando a intensificar a busca. Esse último procedimento, conforme se deduz da Tabela 2, consegue soluções finais melhores em um menor espaço de tempo. Ademais, II consegue produzir melhores soluções mais rapidamente (vide Figura 3). Vale ressaltar, também, que o desempenho de GBT-II ainda pode ser melhorado,

uma vez que seus parâmetros não foram devidamente ajustados.

Finalmente, acrescentamos que o algoritmo GBT-II pode ser aplicado para tratar também de outros requisitos, além daqueles mencionados na seção 2., tais como: limite diário de aulas por professor, intervalo entre aulas de uma mesma matéria para uma mesma turma, preferência de professores por certos horários, etc. Basta, nesses casos, adicionar à função objetivo componentes que meçam a diferença entre a solução corrente e a desejada.

Referências

- [1] D. Abramson, Constructing school timetables using Simulated Annealing: sequential and parallel algorithms, *Management Science* **37** (1991), 98-113.
- [2] R. Alvarez-Valdes, G. Martin and J. M. Tamarit, Constructing Good Solutions for the Spanish School Timetabling Problem, *Journal of the Operational Research Society* **47** (1996), 1203-1215.
- [3] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, “Network flows: theory, algorithms and applications”, Prentice Hall, New Jersey, 1993.
- [4] A. Colormi, M. Dorigo and V. Maniezzo, Metaheuristics for high school timetabling, *Computational Optimization and Applications* **9** (1998), 275-298.
- [5] D. Costa, A tabu search algorithm for computing an operational timetable, *European Journal of Operational Research* **76** (1994), 98-110
- [6] S. Even, A. Itai and A. Shamir, On the complexity of timetabling and multi-commodity flow problems, *SIAM Journal of Computation* **5** (1976), 691-703.
- [7] T.A. Feo and M.G.C. Resende, Greedy randomized adaptive search procedures, *Journal of Global Optimization* **6** (1995), 109-133.
- [8] F. Glover and M. Laguna, “Tabu Search”, Kluwer Academic Publishers, Boston, 1997.
- [9] S.S. Mata, “O Problema do Horário na Escola de Segundo Grau: Modelagem e Implementação”, Tese de mestrado, Progr. de Eng. de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, 1989.
- [10] A. Schaerf, Tabu search techniques for large high-school timetabling problems, *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, 1996, 363-368.
- [11] A. Schaerf, A survey of automated timetabling, *Artificial Intelligence Review* **13** (1999), 87-127.
- [12] M.J.F. Souza, N. Maculan and L.S. Ochi, Melhorando Quadros de Horário de Escolas através de Caminhos Mínimos, *Tendências em Matemática Aplicada e Computacional* **1** (2000), 515-524.