

# MELHOR HORÁRIO PARA O SISTEMA CEFET CAMPOS

## Um aplicativo para automatizar a elaboração de horários das aulas

Sérgio Vasconcellos Martins\*

*A partir de trabalho de pesquisa no sistema CEFET Campos, foi desenvolvido um software, objetivando oferecer conforto e agilidade ao usuário para feitura automática de horários de aulas. Embora seu uso não esteja difundido no sistema, o aplicativo já está disponível para a comunidade, em versão preliminar. O programa computacional busca o melhor horário via emprego de heurística de otimização combinatória, respeitando as condicionantes. Atendendo a normas que regulam os projetos de pesquisa da instituição, o trabalho é aqui divulgado: o problema matemático subjacente e seu modelo conceitual são pormenorizados, bem como o algoritmo de solução concebido e as características e funções do software.*

**PALAVRAS-CHAVE:** Pesquisa operacional. Otimização combinatória. Heurística. Problema de horário. Desenvolvimento de *software*.

### 1. INTRODUÇÃO

Como resultado das iniciativas de pesquisas tecnológicas do sistema CEFET Campos, desenvolveu-se um *software* para elaboração automatizada de horário escolar foi desenvolvido no ano de 2003. O uso

\* Professor da Área de Indústria do CEFET Campos, Coordenador do Núcleo de Pesquisa Operacional e Doutor em Ciências de Engenharia.

deste tipo de ferramenta computacional ainda não faz parte da rotina do sistema e atualmente as programações das aulas são construídas manualmente, o que torna o processo moroso e penoso, também pela natureza dinâmica do problema, que reclama freqüentes reprogramações. O aplicativo visa a tornar mais ágil e confortável, para o responsável, o ato de montar horários de aulas de boa qualidade para os cursos regulares do sistema CEFET Campos – ou de outros eventos correlatos, que demandem recursos. A automatização no preparo de horários sucede pelo emprego de algoritmo heurístico de otimização combinatória, considerando restrições do problema, tais como: disponibilidade de turmas, professores e salas; preferência dos professores; distribuição uniforme das aulas na semana etc.

A construção de horário escolar pode ser modelada como um problema matemático de otimização combinatória (mundialmente referido como *timetable problem*, *TTP*) que é estudado há anos em universidades de todo o mundo e classificado como NP-difícil (KINGSTON, COOPER, 1996). Isto significa que a maioria de suas instâncias práticas não pode ser resolvida em tempo polinomial, implicando demasiado esforço computacional – o conjunto de dados de entrada de um problema de otimização estabelece uma instância em particular. Em função das singularidades e regulações do sistema de ensino de cada país, pode haver diferenças significativas de um caso para outro (DREXL; SALEWSKI, 1997. DIMOPOULOU; MILIOTIS, 2001). No estudo em tela, buscou-se guardar estrita relação com o sistema brasileiro típico (fundamental, médio e superior), particularmente em sintonia com a sistemática do CEFET Campos.

Com o propósito de relatar o projeto de pesquisa e seus resultados, este artigo obedece ao seguinte leiaute: na seção 2 descreve-se o problema matemático de otimização associado à construção de horário escolar e seu modelo conceitual; o algoritmo heurístico concebido para solução pode ser visto na seção 3, onde antes é introduzida a notação adotada para seus

parâmetros e variáveis; na seção 4, o aplicativo tem suas particularidades funcionais brevemente expostas; algumas considerações e sugestões podem ser vistas na seção 5; o apêndice, finalmente, retrata algumas de suas telas.

## 2. O PROBLEMA DO HORÁRIO ESCOLAR

Basicamente, o problema de horário escolar tipificado para abordagem pode ser descrito como a programação de um conjunto de aulas ao longo de uma semana letiva. A partir do conjunto de professores e do conjunto de turmas, uma aula a ser programada se define como cada uma das reuniões semanais exigidas do par professor-turma dentro de um conjunto de períodos (tempos) nos quais alguns recursos devem estar disponíveis e várias restrições adicionais têm que ser observadas. Entre estas restrições, podem ser citadas: (i) as respectivas demandas e disponibilidades de professores, disciplinas, turmas e salas; (ii) não se permite qualquer tipo de superposição de turmas, disciplinas e professores (por exemplo, dois ou mais professores não podem lecionar numa mesma turma ao mesmo tempo e duas ou mais turmas não podem demandar o mesmo professor concomitantemente); (iii) horas vagas e pré-designações; (iv) preferência dos docentes por períodos em que lecionar; (v) distribuição compacta das aulas etc. Com o objetivo de tornar o problema mais flexível, no presente estudo algumas restrições são relaxadas. É o bastante que um horário  $H$  obedeça às restrições (ii) e (iii) simultaneamente para ser assumido como viável. Agindo diferentemente, os problemas práticos ficariam excessivamente rígidos, com o que algumas alternativas de horário, não ótimas embora interessantes, não seriam consideradas. Adicionalmente, ao aceitar um horário antes inaceitável (por exemplo, porque algumas aulas não foram alocadas), o aplicativo vai conferir mais autonomia ao usuário, que poderá editar *a posteriori* um horário insatisfatório após negociação com os professores envolvidos, como é costumeiro na prática.

Matematicamente, o problema pode ser assim traduzido: seja uma classe (turma)  $c_i$  definida como um grupo de estudantes com exatamente o mesmo programa de disciplinas e  $p_j$  representando um professor. Os conjuntos  $C = \{c_1, \dots, c_m\}$  e  $P = \{p_1, \dots, p_n\}$  designam, respectivamente, a totalidade das  $m$  classes e dos  $n$  professores. A matriz  $m \times n$  de designação  $A=[a_{ij}]$  representa a atribuição de tempos de aula  $a_{ij}$  envolvendo a classe  $c_i$  e o professor  $p_j$  para todos  $i, j$ . O problema consiste em construir, quando possível, um horário global em  $T$  períodos. Cada classe e cada professor estão envolvidos em, no máximo, uma aula a cada tempo. Pode-se facilmente constatar que, expresso desta forma simplificada, o problema é viável, i.é., uma solução existe se e somente se  $\sum_i a_{ij} \leq T$  para todo  $j$  e  $\sum_j a_{ij} \leq T$  para todo  $i$ .

De modo a tornar o *TTP* tratável, o modelo é aqui adaptado ao conhecido problema de programação de tarefas (*jobs*). Mais exatamente, usa-se sua versão mais abrangente, referida como *PSP* (*project scheduling problem*) multimodo – cf. MARTINS, 2000. Assim sendo, se a carga horária semanal (*CHS*) de determinada disciplina para uma turma é distribuída ao longo da semana, cada reunião professor-turma é tomada como um *job*, podendo assumir diferentes durações. Como exemplo, uma *CHS* de 5 aulas semanais da disciplina *Língua Portuguesa* pode ser distribuída no esquema 3+2 representando dois *jobs* a serem processados: um de duração 3 tempos e o outro de duração 2 tempos. O que determina o esquema de distribuição é o modo de execução atribuído a uma *CHS*, como se vê mais adiante.

## 2.1. Modelo matemático

Três são os critérios de desempenho a serem minimizados na modelagem concebida: sobra de *jobs* (aulas necessárias mas não programadas), dispersão do horário, e discrepância entre as horas em que

os professores são alocados e sua real preferência para lecionar. Trata-se, por conseguinte, de problema de otimização da modalidade multicritério (múltiplos objetivos). Assim sendo, verifica-se *tradeoff* entre estes critérios: uma ação que beneficie um critério pode vir em detrimento dos outros. Por exemplo, pode acontecer que, para atender às preferências dos professores em um caso determinado, mais aulas restassem sem programação. De modo a avaliar o ponto de sub-ótimo, a estratégia adotada é a da estimação de ponto pela soma ponderada, conforme se segue.

Se se define o espaço viável  $HV$  como sendo o conjunto de todos os horários viáveis  $H$ , consoante a definição já estipulada, conceitualmente, o modelo matemático multicritério de otimização combinatória associado ao  $TTP$  reduz-se àquele apresentado na tabela 1. A função objetivo  $z$  (1) mede a discrepância existente entre um horário meramente viável e as reais necessidades da instituição escolar, i.é., horário ótimo ( $z$  deve ser nula, neste caso). Os coeficientes  $w_i$  são escalares que representam os pesos associados às funções  $f_i(H)$  e traduzem as decisões do usuário com relação ao grau de aceitabilidade do horário gerado. Sem perda de generalidade, assume-se que cada vetor de ponderação  $w \in \hat{R}^3$  é normalizado de modo que a soma de seus elementos resulte na unidade. Denotando-se  $W$  como o conjunto de todos os possíveis vetores de ponderação:

$$W = \{ w \in R^3 \mid w_i > 0, \sum_{i=1}^3 w_i = 1 \}.$$

Desse modo, a técnica de estimação de ponto pela soma ponderada busca o espaço viável a partir da combinação convexa dos objetivos.

As funções  $f_i(H)$ ,  $i = 1, 2$  e  $3$ , definem-se da seguinte forma:

- $f_1(H)$  diz respeito aos *jobs* que não foram escalonados (aulas restantes, sem terem sido programadas);
- $f_2(H)$  relaciona-se com a repartição das *CHS*'s das turmas na

semana letiva (distribuição uniforme das aulas: considera-se melhor, por exemplo, alocar uma  $CHS = 4$  no esquema 2+2 do que 3+1);

- $f_3(H)$  refere-se à preferência dos professores das horas em que lecionar.

$$\text{Minimizar } z = \sum_{i=1}^3 w_i \cdot f_i(H) \quad (1)$$

sujeito a :

$$H \in HV \quad (2)$$

---

**Tabela 1:** Modelo conceitual do problema de horário considerado

A restrição (2) circunscreve a busca somente entre os horários viáveis. Como é importante observar, o modelo matemático expresso na tabela 1 não é passível de tratamento por *softwares* comerciais de otimização, visto que é excessivamente simplificado: por exemplo, nada faz constar com respeito aos recursos disponíveis e exigidos. De qualquer modo, para instâncias reais, via de regra com centenas de *jobs* e recursos, não existe algoritmo de solução exata capaz de resolver o *TTP* em tempo computacional plausível. Para uma solução sub-ótima do *TTP* em consideração, um algoritmo meta-heurístico foi desenvolvido neste trabalho.

### 3. ALGORITMO META-HEURÍSTICO CONCEBIDO

Heurística é uma técnica que visa à resolução de problemas matemáticos de otimização por meio de busca de soluções sub-ótimas em um razoável custo computacional sem ser capaz de garantir otimalidade e sequer indicar o quão próximo do ótimo a solução encontrada está. Meta-heurísticas são procedimentos variantes que, tipicamente, exploram o conhecimento adquirido pela avaliação de soluções previamente visitadas no espaço viável. *Busca tabu*, *simulated annealing*, *GRASP* e algoritmos evolucionários (cf. MARTINS, 2000) são exemplos de meta-heurísticas.

Denominação mais geral para as abordagens genéticas, algoritmos evolucionários (ou evolutivos) são meta-heurísticas baseadas no princípio da evolução natural (cf. GOLDBERG, 1989). Sua aplicabilidade universal e destacada performance, quando voltados para uma vasta gama de problemas de otimização, fizeram irromper um notável interesse neste tipo de estratégia nos últimos anos. Têm experimentado largo emprego nos domínios, *e.g.*, da otimização contínua e discreta, *machine learning* e teoria dos jogos. O algoritmo genético clássico tem a seguinte estrutura básica:

**Para uma dada instância, faça:**

Geração da população inicial;  
Avaliação da adaptabilidade dos indivíduos.

**Repetir**

Cruzamento dos “pais” e geração dos descendentes;  
Mutação dos descendentes;  
Avaliação de adaptabilidade dos descendentes;  
Seleção dos “pais” para formação de nova população.

**Até satisfazer condição de parada**

**Retornar à melhor solução**

O algoritmo projetado para tratamento do *TTP* em questão é do tipo evolucionário (genético), uma vez que este já provou ser o de melhor desempenho para os problemas de programação de tarefas (HARTMANN, 1997). Para melhor compreensão de seu pseudocódigo, exibido adiante na tabela 2, seja a notação a seguir:

NI : quantidade de indivíduos de cada uma das sucessivas populações;  
 $d_{jm}$  : duração do *job j* processado no modo *m*;  
T : número total de tempos de aula na semana; por exemplo, para uma semana letiva de cinco dias com três turnos diários e cinco aulas por turno,  $T=5 \times 3 \times 5 = 75$  tempos;

- $CHS_{kl}$  : carga horária semanal da disciplina  $k$  para a classe  $l$ ;  
 $CHS$  : carga horária semanal da disciplina  $k$  para a classe  $l$ ;  
 $j$  : *job*, cada um correspondendo a uma subdivisão de cada  $CHS_{kl}$  após atribuição dos modos de execução;  
 $TJ$  : conjunto de todos os *jobs* a serem processados;  
 $CJ_i$  : conjunto dos *jobs* com execução completada na iteração  $i$ ;  
 $EJ$  : conjunto dos *jobs* sendo processados no tempo atual;  
 $VJ$  : conjunto dos *jobs* elegíveis para execução no tempo atual, ou seja, com recursos disponíveis;  
 $PC_j$  : período de conclusão do *job*  $j$ ;  
 $z_i$  : nível de adaptabilidade do indivíduo  $i$ , quer dizer, valor da função objetivo na iteração  $i$ .

Para ilustrar a distribuição dos *jobs* após atribuição dos modos de execução, considere-se uma turma exigindo 4 aulas de *Matemática* semanalmente. Neste caso a respectiva  $CHS$  tem tamanho 4 e pode ser subdividida em 4 *jobs*, com as seguintes configurações possíveis de duração para os *jobs*, em função do modo de execução atribuído: (1+1+1+1), (2+1+1+0), (2+2+0+0) e (3+1+0+0), (4+0+0+0). No caso, e. g., de (2+2+0+0), têm-se 2 *jobs* de duração 2 tempos de aula e mais 2 *jobs* de duração nula cada um. (É exatamente por admitir distintas configurações da distribuição da carga horária, nos dias da semana, que o problema se diz multimodo.) Assumindo assim a possibilidade de *job* com duração nula, o número de *jobs* corresponderá sempre ao tamanho da carga horária, independentemente do modo de execução escolhido –  $CHS = 4 =$  número de *jobs*, no exemplo. O número total de *jobs* a serem escalonados (tomando em conta todos os horários de todas as turmas), por conseguinte, permanece constante em qualquer iteração, com o que se racionaliza o procedimento computacional. A melhor escolha da distribuição corresponde ao menor custo para a função  $f_2(H)$ . No exemplo, a configuração (2+2+0+0) é, claramente, a mais conveniente porque mais equilibrada e, portanto, vai receber custo nulo.

Uma vez designados os modos para cada *CHS* aleatoriamente, um conjunto é formado com todos os *jobs* a serem programados. Ato contínuo, a cada iteração um novo indivíduo (horário) é gerado e a população inicial é montada após NI iterações. O esquema de geração atua do período *I* até o período *T*, procedendo ao incremento seqüencial unitário, cronologicamente. Em cada período *t*, escalonam-se todas as atividades que já podem ter seu processamento iniciado, ou seja, aqueles *jobs* que são viáveis em recursos, daí porque o esquema de geração é dito *em paralelo*. Após a montagem da população inicial, é importante submetê-la aos operadores *cruzamento* (dos pais), *mutação* e *seleção* no âmbito do algoritmo genético – estes operadores não são aqui detalhados. O processamento se repete nesta dinâmica com as populações descendentes, todas de tamanho NI, até que seja atendido o critério de parada (cf. seção 4). Ao emular o processo da seleção natural, o procedimento, espera-se, vai paulatinamente gerando indivíduos mais bem adaptados e isso se traduz em horários cada vez melhores. O operador *seleção* mede o grau de adaptabilidade de cada indivíduo *i* (valor da função objetivo  $z_i$ ), o que corresponde ao nível de aceitabilidade do horário construído. O melhor horário é retornado uma vez concluído o processamento.

## Início

Leia dados de entrada

### Processamento

Para  $i := 1, \dots, NI$ , faça

$CHS := \{ CHS_{1i}, \dots, CHS_{mi} \}$  - conjunto de cargas horárias

Atribua modo a cada  $CHS_{kl} \in CHS$  - para formar TJ

$TJ := \{ 1, \dots, J \}$  - conjunto inicial dos jobs

$CJ_i := \{ \}$  - conjunto dos jobs completados

$EJ := \{ \}$  - conjunto dos jobs em execução

Para  $t := 1, \dots, T$  faça

Compute $VJ$	- conjunto dos jobs elegíveis
Enquanto $VJ \neq \emptyset$ faça	
Selecione $j \in VJ$	- randomicamente
$PC_j := t + d_{jm} - 1$	- período de conclusão do job $j$
$EJ := EJ \cup \{j\}$	- atualiza $EJ$
$TJ := TJ \setminus \{j\}$	- atualiza $TJ$
Atualize $V$	
fim-enquanto	
Para todo $j \in EJ$ faça	
Se $PC_j = t$	
$CJ_i := CJ_i \cup \{j\}$	- atualiza $CJ_i$
$EJ := EJ \setminus \{j\}$	- atualiza $EJ$
fim-para	
fim-para	
Armazene $CJ_i$	
Compute $z_i$ de $CJ_i$	- valor da função objetivo
fim-para	
Algoritmo Genético	- gera população inicial
<b>Saída</b>	- geração dos descendentes
Registre o horário de menor $z$ encontrado	- o melhor horário é construído

**Fim**

---

**Tabela 2:** Pseudocódigo da meta-heurística idealizada

## 4. DESCRIÇÃO FUNCIONAL DO APLICATIVO

O aplicativo de construção automática de horários foi criado a partir da codificação e compilação de parte do algoritmo heurístico apresentado, via emprego do IDE Delphi 6 em ambiente Windows XP. Faz-se aqui, brevemente, a descrição das características funcionais do *software*, intitulado *MelHora* e doravante também assim referido. O que se segue deve ser acompanhado com as figuras do apêndice. A partir de interface amigável e de uso intuitivo (Fig. A1), o usuário alimenta a base de dados e passa a dispor dos recursos, como se expõe a seguir:

### 4.1. Dados de entrada

As seguintes informações devem ser cadastradas antes do processamento:

- parâmetros globais: instituição e semestre, dias, turnos e horários letivos (Fig. A2);
- turmas, disciplinas e salas (Fig. A3);
- professores, informando carga horária semanal máxima, disciplinas que podem lecionar, disponibilidades e preferências de horário (Fig. A3);
- carga horária semanal das disciplinas para cada turma;
- opcionalmente, pré-designação de aulas e/ou hora vaga para uma dada turma.

Antes do processamento propriamente dito, abre-se uma janela de diálogo (Fig. A4), permitindo que o usuário module devidamente, conforme suas conveniências, os pesos  $w_i$  associados aos critérios de otimização, como explicado na seção 2 (Modelo matemático), referentes a: (1) *Não designação* – aulas necessárias mas não alocadas; (2) *Distribuição das aulas* – compactação do horário; e (3) *Preferência dos professores* – períodos de predileção em que lecionar.

## 4.2. Recursos do aplicativo

Ao oferecer os recursos a seguir descritos, o *MelHora* também proporciona valiosa visão panorâmica das condições didáticas específicas da instituição ainda durante a entrada de dados:

- fluxograma ilustrativo, sugerindo roteiro com a seqüência dos passos de provimento dos dados para usuários não iniciados;
- em todas as telas, mostra os quantitativos para constante conferência;
- árvore de distribuição das disciplinas nas turmas;
- tela exclusiva com estatísticas da instância do problema (professores, turmas, disciplinas, salas etc.);
- na fase de pré-processamento, verifica pendências de modo a prevenir-se quanto a esquecimentos e inconsistências de dados (Fig. A4);
- finalmente, após processamento, gera simultaneamente os horários para turmas, professores e salas, em relatórios independentes e editáveis (Fig. A5). O conjunto das aulas não alocadas também pode ser exibido.

Um problema de otimização da natureza do *TTP*, no mais das vezes, exige grande esforço computacional até que se assegure encontrar uma alternativa satisfatória. A depender da disponibilidade, o tempo de processamento deve ser prefixado pelo usuário, desde alguns minutos até algumas horas, como um dos critérios de parada para o algoritmo. Com mais longo tempo de computação, crescem as chances de se encontrar um horário que melhor atenda às necessidades da instituição, evidentemente com a precondição de que a instância seja viável. A qualquer momento durante a execução, todavia, o processamento pode ser interrompido e o *MelHora* vai registrar e retornar o melhor horário encontrado até aquele instante. Para ajustes e aperfeiçoamento até se construir o horário definitivo,

arquivos com alternativas intermediárias podem ser salvos, comparados e editados pelo usuário após sucessivos processamentos.

## 5. CONCLUSÃO

Em face da necessidade de elaboração recorrente de horários em uma instituição de ensino, o *software* desenvolvido mostra-se oportuno como coadjuvante na otimização e prevenção de eventuais conflitos no emprego dos recursos – humanos e físicos –, quase sempre escassos. Na medida em que os horários são de difícil elaboração e estão sujeitos a constantes reprogramações, a ferramenta computacional, se e quando plenamente funcional, também vai contribuir para comodidade e presteza no ato de montar os variados horários, imprescindíveis ao bom desempenho de uma instituição educacional.

Convém ser reforçado que o programa não é de uso corrente no sistema CEFET, está em fase incipiente e ainda serão necessários testes exaustivos e ajustes antes que se possa dispor de uma ferramenta de fato eficaz. Nesse sentido, todos os interessados da comunidade são instados a contribuir para o aperfeiçoamento deste trabalho. Não obstante, simulações empreendidas de geração de horários, reais e fictícios, para depuração de *bugs* e comprovação de eficiência e adequação, em vista do escopo do projeto, comprovaram que o aplicativo em versão beta, ora disponível, já é capaz de construir horários de boa qualidade, mais ainda se aplicado a instâncias menos difíceis (menos de cem *jobs* e pouco restritas em recursos).

Futuramente o *MelHora* poderá incorporar itens não considerados atualmente, tais como incluir mais objetivos a serem otimizados (por exemplo, minimizar janelas), prioridades diversas (por exemplo, professor cuja alocação é prioritária por estar menos disponível), controle de mais recursos

(por exemplo, salas de aula) e funções outras (por exemplo, o próprio *software* sugerir ao usuário alterações que possam redundar em melhoria no horário). Sugestões são muito bem-vindas com relação a outros itens pedagógicos de interesse.

Toda a pesquisa consumiu quatro semestres e, malgrado reiteradas gestões de seu coordenador, o projeto transcorreu em ritmo insatisfatório, em razão de alguns percalços enfrentados, notadamente no último semestre. Desafortunadamente, o trabalho resultou truncado e o algoritmo genético não pôde ser inteiramente codificado, em grande medida por indisponibilidades do bolsista programador. Por isso, a heurística de busca implementada no código disponível apresenta caráter exclusivamente aleatório.

Como é habitual em projetos de desenvolvimento de *software* (PRADO, 2003), dificilmente o cronograma preestabelecido é cumprido, especialmente porque o escopo costuma ser pouco estável. É de bom alvitre que, em algum tempo, este trabalho venha a ser retomado e concluído, em vista de sua importância. Convém lembrar que a graduação em tecnologia da informação do CEFET Campos intitula-se “*Curso Superior de Tecnologia em Desenvolvimento de Software*”, o que só evidencia a especial relevância deste projeto. Colateralmente, tenciona-se consolidar a implantação do Núcleo de Pesquisa Operacional (NPO) no sistema CEFET Campos e futuras iniciativas por parte de outros pesquisadores, nesse campo, são fortemente encorajadas.

## APÊNDICE

Interface do aplicativo *MelHora*, exibindo algumas de suas funções e características.



Figura A1: Tela de abertura

The 'Parâmetros globais' dialog box has tabs for '1º Turno', '2º Turno', '3º Turno', and 'Dados da Instituição'. The '1º Turno' tab is active. It contains the following fields:

- 'Número de aulas diárias do turno': A dropdown menu with the value '5' selected.
- 'Número de dias com aula': A dropdown menu with the value '5 (Segunda a Sexta)' selected.
- 'Rótulo do Turno': A text input field containing the text 'Manhã'.

On the right side, there is a table titled 'Horários (Início e Fim)':

	Horários (Início e Fim)
1ª	07:30 a 08:20
2ª	08:20 a 09:10
3ª	09:10 a 10:00
4ª	10:20 a 11:10
5ª	11:10 a 12:00

At the bottom, there is a text box with the instruction: 'Aqui você define a quantidade de turnos e os horários de cada um deles, e se haverá ou não aulas aos sábados:'. Below this are two buttons: 'Ok' and 'Fechar'.

Figura A2: Cadastro dos parâmetros globais



Figura A3: Gerenciador de cadastro de dados específicos

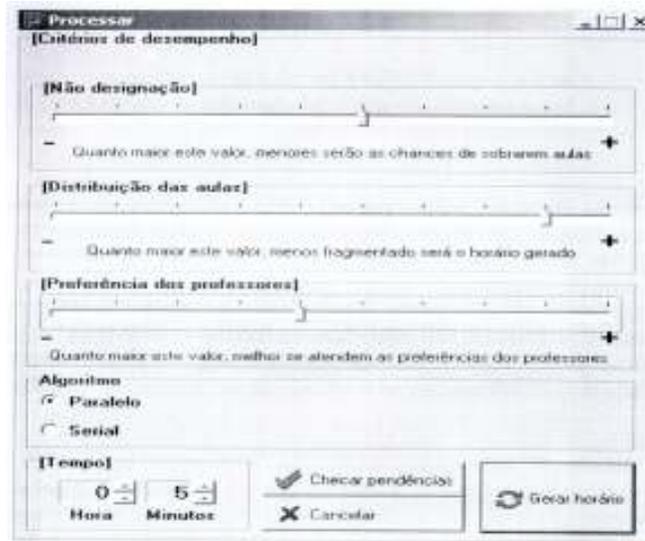


Figura A4: Tela onde se modulam os pesos dos critérios

Horário	Segunda	Terceira	Quarta	Quinta	Sexta	Sábado
07:30 a 08:30	Inglês M. Assis 103	Biologia C. Cavallini 103	Química J. C. de Carvalho 103	Física M. Bandeira 103	Matemática E. Cunha 103	
08:30 a 09:30	Inglês M. Assis 103	Biologia C. Cavallini 103	Química J. C. de Carvalho 103	Física M. Bandeira 103	Matemática E. Cunha 103	
09:30 a 10:00	Inglês M. Assis 103	Geografia J. Amado 103	Química J. C. de Carvalho 103	Física M. Bandeira 103	História C. Menezes 103	
10:20 a 11:30	Português F. Pereira 103	Geografia J. Amado 103	Literatura C. Luperini 103	Matemática E. Cunha 103	História C. Menezes 103	
11:30 a 12:00	Português F. Pereira 103	Geografia J. Amado 103	Literatura C. Luperini 103	Matemática E. Cunha 103	História C. Menezes 103	

**Figura A5:** Visualização de horários gerados para professores, turmas e salas

\*\*\*

**A better schedule for the CEFET Campos system:  
A software to automate the elaboration of class schedules**

Sérgio Vasconcellos Martins

*After research carried out on the CEFET Campos system, a software has been developed aiming at offering more comfort and agility to the user via the automatic making of class schedules. Although the program is not yet spread on the system, the application is available for the community in a preliminary version. The computer program searches for the best timetable by means of combinatorial optimization heuristic, considering the constraints. In accordance with institutional rules, the research is presented here as follows: the underlying mathematical problem and its conceptual model are detailed in the article, as well as the algorithm created for the program, and the features and functions of the software.*

**KEYWORDS:** *Operational research. Combinatorial optimization. Heuristics. Timetabling problems. Software development.*

## REFERÊNCIAS

DIMOPOULOU, M.; MILIOTIS, P. **Implementation of a university course and examination timetabling system.** European Journal of Operational Research, n. 130, 2001, p. 202-213.

DREXL, A.; SALEWSKI, F. **Distribution requirements and compactness constraints in school timetabling.** European Journal of Operational Research, n.102, 1997, p. 193-214.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning.** Massachusetts: Addison-Wesley Longman, 1989.

HARTMANN, S. **Project scheduling with multiple modes: a genetic algorithm.** Manuskripte aus den Instituten für Betriebswirtschaftslehre der, Alemanha: Universität Kiel, 1997.

KINGSTON, J. H.; COOPER, T. B. **The complexity of timetable construction problem.** Basser Department of Computer Science, Austrália: University of Sydney, 1996.

MARTINS, S. V. **Gerenciamento de projeto: meta-heurísticas para otimização do escalonamento de atividades na exploração e produção de petróleo.** Rio de Janeiro, 2000. 130p. Tese (doutorado) – Ciências de Engenharia, UENF.

PRADO, D. **Gerenciamento de Projetos nas Organizações.** Belo Horizonte: EDG, 2003.