

# UM ALGORITMO HÍBRIDO BASEADO EM ALGORITMOS MEMÉTICOS E RECONEXÃO POR CAMINHOS PARA RESOLUÇÃO DO PROBLEMA DE HORÁRIO ESCOLAR

**Alessandra Martins Coelho**

Centro Federal de Educação Tecnológica de Rio Pomba  
Av. Dr. José Sebastião da Paixão, s/n, Lindo Vale, Rio Pomba-MG  
ale\_contatos@ig.com.br

**Sérgio Ricardo de Souza**

Centro Federal de Educação Tecnológica de Minas Gerais  
Av. Amazonas, 7675, Nova Gameleira, Belo Horizonte-MG  
sergio@dppg.cefetmg.br

## Resumo

Este trabalho propõe uma metodologia híbrida entre algoritmos meméticos e a técnica de reconexão por caminhos, para solucionar o Problema de Horário Escolar. Considerou-se, para análise, os dados referentes aos professores do Centro Federal de Educação Tecnológica de Rio Pomba-MG nos anos de 2005 e 2006, para a programação de dois turnos em simultâneo. A metodologia proposta tem como objetivo satisfazer o maior número de professores quanto à quantidade de dias em que suas aulas são distribuídas; minimizar o número de janelas e quebras de aula; além de atender a um certo número de aulas duplas. Os resultados encontrados pelo algoritmo híbrido são comparados com os do método iterated local search.

**Palavras-Chaves:** Metaheurísticas; Problema de Horário Escolar; Algoritmo Memético.

## Abstract

This work provides a hybrid methodology between the memetics algorithms and the Path Relinking technique for solving an School Timetabling Problem. The analyzed source data came from the teachers staff of Centro Federal de Educação Tecnológica de Rio Pomba-MG from years 2005 and 2006, to the simultaneous scheduling two turns. The methodology objective is to satisfy the higher number of teachers requests, by reducing the number of days which its lessons are distributed; minimizing the number of empty windows and same classes being apart from each other, besides of reducing the number of double classes. The results found for the hybrid algorithm are compared with the ones of the method iterated local search.

**Keywords:** Metaheuristic; School Timetabling Problem; Memetic Algorithms.

## 1. INTRODUÇÃO

Problema de Horário é a denominação genérica para uma classe de problemas combinatórios, com grande relevância na área de Pesquisa Operacional. Consiste em construir quadros de horários para uma série de atividades, atendendo a um determinado conjunto de restrições.

Um horário é uma tabela de eventos organizada de acordo com o tempo em que esses eventos acontecem. Os eventos normalmente são encontros entre pessoas em um local específico. Por conseguinte, um horário especifica quais pessoas se encontram em qual local e a que horas.

Para se construir um horário várias exigências devem ser satisfeitas, de tal forma que

não exista nenhum evento programado ao mesmo tempo. Além disso, deve-se satisfazer, tanto quanto possível, aos desejos de todas as pessoas simultaneamente envolvidas.

Problemas de horário aparecem em diversos contextos, como, por exemplo, escala de empregados, eventos esportivos, horários educacionais e em problemas de transporte.

Desde a década de 60, os Problemas de Horário têm sido assunto de pesquisa por parte da comunidade científica, tanto devido ao aumento da complexidade dos problemas dessa classe, quanto na busca de melhores caminhos para solucioná-los. Economia e utilização eficiente de recursos são os vetores para o aperfeiçoamento da Geração de Horários.

Diversos algoritmos e heurísticas vêm sendo propostos com o objetivo de resolver diferentes aspectos do problema. Dentre esses métodos, pode-se citar as metaheurísticas *busca tabu*, *simulated annealing*, algoritmos genéticos, citados em [8] e algoritmos meméticos, em [2].

Este trabalho aborda o Problema de Horário Escolar, também conhecido como Problema Classe-Professor, que é uma variante do Problema de Horário Educacional. O problema proposto é tratado utilizando-se uma metaheurística híbrida, que combina as metaheurísticas *iterated local search*, algoritmos meméticos e a técnica de intensificação de reconexão por caminhos.

Este artigo está organizado como segue. Na seção 2 descreve-se o problema abordado. Nas seções 3 e 4 faz-se uma breve descrição do algoritmo memético; apresenta-se a forma de representação; a geração de uma solução inicial; as estruturas de vizinhança, as funções de avaliação e aptidão adotadas, bem como os operadores utilizados. A seção 5 apresenta os resultados encontrados. A seção 6 conclui este trabalho.

## 2. O PROBLEMA ABORDADO

O Problema de Horário Escolar, também conhecido como Problema Classe-Professor, trata da geração de quadros de horários das aulas semanais dos professores. As aulas são ministradas em um mesmo turno (manhã, tarde ou noite) e, para cada classe, há um conjunto de professores e uma carga-horária semanal para cada disciplina. O problema básico consiste em dado um conjunto de classes, um conjunto de professores, um conjunto de períodos e o número de aulas que cada professor deve ministrar a cada classe, evitar que os professores estejam alocados em mais de uma classe no mesmo horário; evitar que as classes não tenham aula com mais de um professor ao mesmo tempo; e fazer com que a carga-horária semanal de cada disciplina seja cumprida.

Os tipos de requisitos que envolvem o problema podem ser alterados de instituição para instituição, tornando-o de difícil generalização. Esses requisitos envolvem a disponibilidade dos recursos, as preferências pessoais, as características desejadas no *layout* do quadro de horários (como, por exemplo, a inexistência de lacunas no horário dos professores), dentre outros.

## 3. UMA REVISÃO DE ALGORITMOS MEMÉTICOS

Trata-se de uma metaheurística pertencente à classe dos algoritmos populacionais. Um algoritmo populacional utiliza vários indivíduos na procura de soluções factíveis no espaço de busca.

Os algoritmos meméticos têm sido amplamente estudados e aplicados a vários problemas de otimização encontrados na literatura, tais como o caixeiro viajante, programação de jogos e programação de horários educacionais.

O termo “algoritmo memético” foi introduzido por [6] para descrever um processo evolutivo que tenha busca local como parte decisiva na sua evolução. Essa busca pode ser caracterizada como sendo um refinamento local dentro de um espaço de busca, de modo que

um indivíduo pode ter seu nível de adaptação aumentado após passar por uma etapa de refinamento.

Segundo [5], a idéia básica que sustenta os algoritmos meméticos é a combinação de conceitos e estratégias de diferentes metaheurísticas, como a busca baseada em populações (nos algoritmos evolutivos) e técnicas de busca local, com a intenção de unir as vantagens encontradas em cada uma delas.

Todas as etapas dos algoritmos genéticos, como os operadores de seleção, recombinação e mutação estão presentes nos algoritmos meméticos. A diferenciação entre estas duas classes de algoritmos ocorre pela inclusão de uma etapa de otimização dos agentes (população de indivíduos autônomos), através da adição de operadores de busca local independentes, que particularizam o aprendizado para cada agente.

Uma diferença entre genes e memes está no processo de transmissão aos seus descendentes. Quando o meme é transmitido, ele será adaptado pela entidade que o recebe com base no seu conhecimento e para melhor atender às suas necessidades. Quanto aos genes, no processo de evolução, eles são transmitidos de uma maneira tal que o descendente gerado vai herdar muitas habilidades e características presentes em seus progenitores. Dessa maneira, os algoritmos genéticos são inspirados na tentativa de emulação computacional da evolução biológica, enquanto os algoritmos meméticos tentam fazer o mesmo em relação à evolução cultural.

Na evolução cultural, a informação envolvida encontra-se nos memes, de modo que as alterações surgem pela combinação, criação e reorganização das representações mentais (conscientes ou não) e pela possível ineficácia dos mecanismos de transmissão de informação. A replicação (fenótipo) ocorre quando essas representações mentais são transformadas em ações passíveis de imitação ou expressas através de alguma linguagem. A incorporação dessa nova informação por parte de algum indivíduo certamente alterará a pressão seletiva e a influência vinculada às limitações impostas pelo ambiente àquele indivíduo.

Conforme [4], em um ambiente que emprega algoritmo memético, os operadores de recombinação e mutação agem como estratégias de diversificação. Os indivíduos da população podem estar localizados em uma região do espaço de busca contendo um ótimo local, chamada base de atração do ótimo local.

Utilizando a informação contida na população, novos pontos de partida podem ser descobertos após a busca local. Os operadores de recombinação e mutação podem gerar indivíduos da população que estejam localizados em bases de atração de ótimos locais ainda não explorados, de modo que um novo pico deva ser alcançado (maximização) ou um vale deva ser explorado (minimização).

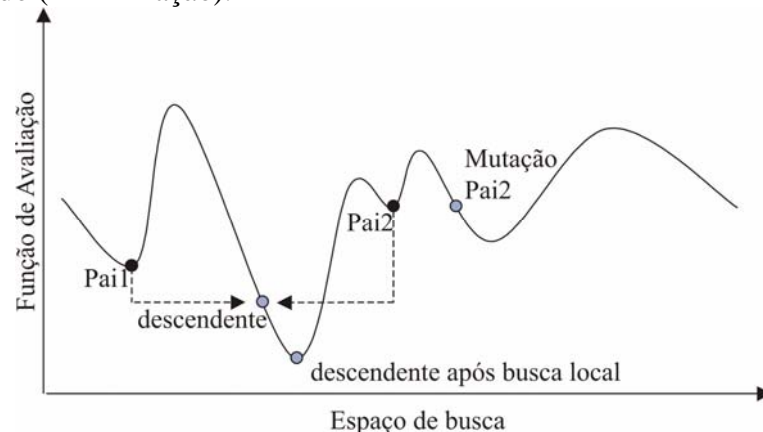


Figura 1: Representação dos operadores de recombinação e mutação agindo como estratégias de diversificação junto a algoritmos meméticos.

Utilizando o conceito de superfície de adaptação (*fitness*), a figura 1 ilustra esses eventos para um problema de minimização. Após a recombinação, o filho gerado pode possuir

uma adaptação baixa, mas um grande potencial para crescimento, de modo que uma busca local pode levar o descendente a assumir um valor de adaptação elevado. A mutação, por sua vez, pode levar a um pequeno aumento (ou decréscimo) da adaptação, por representar uma perturbação local junto à representação do indivíduo.

No caso de problemas de otimização com restrições, já na geração da população inicial podem surgir indivíduos infactíveis. Há, portanto, a necessidade de algum dispositivo de seleção que seja capaz de diferenciar entre indivíduos factíveis ou não, e, se necessário, utilizar uma função de reparação, para levar os agentes a factibilidade, conforme [7]. Uma vez criada a população, com todos os agentes factíveis, uma busca local é empregada, fazendo com que possam atingir um ótimo local dentro da região factível.

De maneira genérica, um algoritmo de busca local começa com um indivíduo  $s_0 \in S$  e tenta, continuamente, encontrar melhores elementos dentre os vizinhos. Em outras palavras, a finalidade será levar esse indivíduo para um local onde a função de adaptação tenha um valor melhor que o atual. Essa busca deverá ser realizada até que uma condição de parada seja satisfeita, sendo que essa condição deve ser atendida sempre que o processo de busca não for capaz de melhorar a solução atual.

#### 4. ALGORITMO HÍBRIDO MEMÉTICO-RECONEXÃO POR CAMINHOS APLICADO AO PROBLEMA DE PROGRAMAÇÃO DE HORÁRIOS

Nas subseções seguintes, é discutido o algoritmo híbrido memético-reconexão por caminhos, desenvolvido para a solução do problema de horário escolar objeto de interesse. O pseudocódigo completo do algoritmo está apresentado na figura 4.

##### 4.1. REPRESENTAÇÃO DE UMA SOLUÇÃO

Uma solução  $s$  (agente) é representada por uma matriz  $Q_{m \times p}$  de valores inteiros, na qual cada linha  $i$  representa a alocação semanal de aulas do professor  $i$ . Cada elemento  $q_{ik}$  indica a atividade do professor  $i$  no horário  $k$ . Os valores atribuídos a cada elemento devem pertencer ao conjunto  $\{-2, -1, 0, 1, 2, 3, \dots, n\}$ , com  $n$  igual ao número de classes, de modo que, caso  $q_{ik} = -2$ , o professor  $i$ , no horário  $k$ , está indisponível para a atividade de aula, mas poderá assumir carga horária, se necessário; caso  $q_{ik} = -1$ , o professor  $i$ , no horário  $k$ , está indisponível para a atividade de aulas e não poderá, em hipótese alguma, assumir encargos de aula. Caso  $q_{ik} = 0$ , o professor  $i$  está alocado para ministrar aulas para alguma classe  $(0, \dots, n)$  no horário  $k$ .

Caso, por fim,  $q_{ik} = \infty$ , o professor está disponível para a atividade de aulas, mas não possui aulas alocadas no horário  $k$ . Uma representação da matriz  $Q$  está apresentada na tabela 1.

Tabela 1: Representação do Quadro de Horários

prof	Semana Letiva																			
	Segunda				Terça				quarta				Quinta				sexta			
	h0	H1	h2	h3	h4	h5	h6	H7	h8	h9	h10	h11	h12	h13	H14	h15	h16	h17	h18	h19
1	-1	-1	1	1	-1	-1	5	5	-1	-1	-1	-1	-1	-1	0	0	-1	-1	4	4
2	-1	-1	11	11	-1	-1	INF	INF	-1	-1	9	9	-1	-1	INF	INF	-1	-1	-1	-1
3	-1	-1	-1	-1	INF	INF	INF	INF	INF	INF	INF	INF	4	4	INF	INF	4	4	4	4
4	9	0	INF	INF	INF	INF	INF	11	INF	9	4	1	5	11	INF	5	4	INF	1	0
.																				
.																				
m	4	INF	INF	11	INF	INF	INF	INF	INF	INF	11	INF	9	9	INF	5	-1	-1	-1	-1

##### 4.2. ESTRUTURA DE VIZINHANÇA

Uma solução  $s'$  é dita vizinha de  $s$  se for obtida a partir de um movimento de

realocação ou troca de aulas em  $s$ .

Dois movimentos de realocação foram definidos para a obtenção da vizinhança: o primeiro realiza, em cada linha  $i$  da matriz  $Q$ , a troca de dois valores distintos e não negativos. Esse movimento é identificado pela tripla  $\langle i, k_1, k_2 \rangle$ , na qual  $k_1$  e  $k_2$  representam os horários nos quais as atividades  $q_{ik_1}$  e  $q_{ik_2}$  do professor  $i$  serão permutadas.

O segundo movimento realiza, em cada linha  $i$  da matriz  $Q$  com aulas obrigatoriamente duplas, a troca de dois valores duplos distintos e não negativos. Este movimento é identificado pela tripla  $\langle i, k_{1a}k_{1b}, k_{2a}k_{2b} \rangle$ , na qual  $k_{1a}k_{1b}$  e  $k_{2a}k_{2b}$  representam os horários nos quais as atividades  $q_{ik_{1a}}q_{ik_{1b}}$  e  $q_{ik_{2a}}q_{ik_{2b}}$  do professor  $i$  serão permutadas.

### 4.3. FUNÇÃO DE AVALIAÇÃO

Em problemas reais, para determinar a qualidade de um quadro de horários, faz-se necessário considerar diferentes objetivos, que, geralmente, estão em conflito entre si. Por exemplo, deseja-se minimizar o número de dias que um professor vai à escola e, ao mesmo tempo, maximizar a quantidade de aulas duplas. No entanto, observa-se, na literatura, que a maioria dos Problemas de Horários abordam o uso de uma única função, que minimiza a soma dos pesos de penalidades para avaliar as soluções e só um horário (com a mais baixa penalidade total) é retornado, como resultado.

Para avaliar um quadro de horários, os requisitos do problema são separados em subconjuntos, cada qual representando uma função objetivo linear de penalidade. É assegurada uma maior importância aos requisitos essenciais. Os requisitos essenciais são aqueles que, se não forem satisfeitos, geram um quadro de horários inviável.

Considera-se, nesse trabalho, como requisitos essenciais: evitar que professores e classes estejam programados para mais de um encontro, em um mesmo horário; respeito ao limite de aulas diário de uma disciplina para uma determinada classe; e respeito às indisponibilidades dos professores e classes.

Os requisitos não-essenciais são aqueles que buscam melhorar a qualidade do quadro de horários. No entanto, se não forem satisfeitos, não geram um quadro de horários inviável. Considera-se, nesse trabalho, como requisitos não-essenciais: a minimização do número de dias que um professor vai à escola; a minimização do número de quebras de aulas; a minimização do número de janelas; e a maximização da quantidade de aulas duplas.

Dessa forma, uma solução  $s$  é avaliada com base na minimização da função  $f$ , conforme [9]:

$$f(Q) = \alpha f_1(Q) + \beta f_2(Q) + \gamma f_3(Q) \quad (1)$$

As duas primeiras componentes mensuram a viabilidade, ou não, do quadro de horários e, a terceira, o nível de satisfação dos professores quanto ao atendimento de seus requisitos pessoais.

A função  $f_1(Q)$  é representada pelo somatório do número de vezes que um professor ministra aulas para mais de uma classe no horário  $k$ , somada ao número de vezes que uma turma está sem atividade nesse mesmo horário  $k$ . A função  $f_2(Q)$  representa o somatório do número de vezes em que é violado o número máximo de aulas diário de uma mesma disciplina. Por fim, a função  $f_3(Q)$ , dada por

$$f_3(Q) = \sum_{i=1}^m (\alpha_i B_i + \beta_i J_i + \gamma_i V_i + \sigma_i D_i) \quad (2)$$

representa o somatório de vezes que as solicitações dos professores não são atendidas, ou seja, representa a ocorrência de lacunas entre duas aulas marcadas para uma mesma classe, no mesmo dia; se existem janelas no horário diário do professor; se os dias solicitados pelo

professor para a distribuição de suas aulas é desrespeitado; e se o número de aulas duplas solicitadas pelo professor, segundo a configuração de um dado curso, não é atendido.

Deve-se observar que uma solução  $s$  somente será viável se  $f_1(Q)$  e  $f_2(Q)$  forem iguais a zero. Os pesos  $\omega$ ,  $\delta$  e  $\rho$  dados às diversas medidas refletem a importância relativa da cada uma delas. Assim,  $\omega > \delta \gg \rho$ , de forma a privilegiar a eliminação das soluções inviáveis.

Do mesmo modo, dentre as soluções viáveis, procura-se obter soluções que compactam a agenda dos professores. Assim, os pesos atribuídos à  $\alpha_i$ ,  $\beta_i$  e  $\gamma_i$  devem ser maiores do que o peso atribuído à  $\sigma_i$ .

#### 4.4. GERAÇÃO DA SOLUÇÃO INICIAL

Optou-se pela geração de quadros de horários (agentes) aleatórios, de modo a diversificar o espaço de soluções, respeitando-se os horários de indisponibilidade dos professores, bem como o número máximo de aulas diárias de cada disciplina. O algoritmo possui uma fase de reparação, para levar os quadros de horários a factibilidade.

#### 4.5. OPERADOR LOCAL

Os otimizadores locais podem ser considerados como um operador e podem ser empregados em diferentes fases do algoritmo. Neste trabalho, utilizou-se como operador local o método baseado em trajetória *iterated local search*, que foi aplicado na população, a cada geração, antes da execução dos operadores de recombinação e mutação.

O método ILS é considerado o esquema mais geral entre as estratégias explorativas. Baseia-se na idéia de aplicar uma busca local em uma solução inicial qualquer até que se encontre um ótimo local, e então, perturbar a solução encontrada e reiniciar a busca local. Essa perturbação deve ser tal que possibilite a manutenção de características da região do ótimo local e, além disso, deve evitar um reinício aleatório. Por outro lado, a perturbação deve ser de tal monta que seja suficiente para escapar de um ótimo local e permitir a exploração de outras regiões do espaço de buscas. O método ILS, portanto, um método de busca local que procura focar a busca não no espaço completo de soluções, mas em um pequeno subespaço definido por soluções que são ótimas locais de determinado procedimento de otimização. A figura 2 ilustra o comportamento do método. Para uma revisão de métodos heurísticos, veja [1].

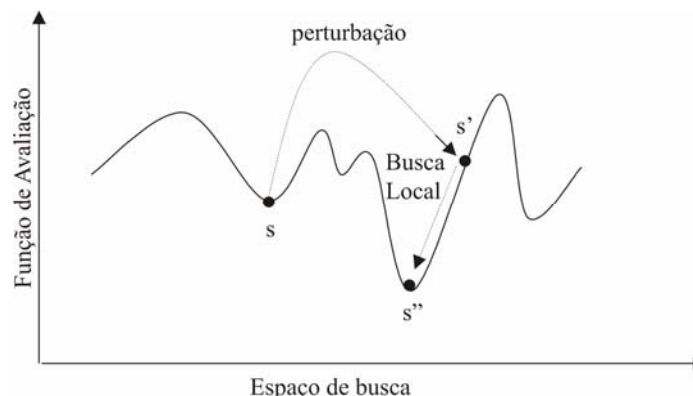


Figura 2: Representação da realização de uma perturbação em uma solução ótimo local, na busca por melhores soluções.

No presente trabalho, sua aplicação é feita na forma como segue. Partindo de uma solução viável pertencente à população, um professor é escolhido aleatoriamente e logo após é aplicada uma perturbação em seu horário. Foram considerados quatro níveis de perturbação.

Em seu nível mais baixo, uma perturbação consiste em uma troca aleatória de um valor distinto e não negativo, simples ou duplo, por outro valor, distinto e não negativo, simples ou duplo. A segunda perturbação realiza o procedimento por duas vezes e assim sucessivamente. Logo após a perturbação, a solução passa por uma busca local. Caso a solução encontrada após a busca local seja pior que a solução anterior, o movimento é desfeito, o nível de perturbação cresce e uma nova perturbação é aplicada à solução. Caso contrário, a solução encontrada passa a ser a solução corrente e o nível de perturbação atual é aplicado. Considera-se, como solução corrente, a melhor solução encontrada até então. O procedimento pára após um certo número de iterações.

Utilizou-se, como busca local associada ao método ILS, o método randômico não-ascendente, que é uma variante do método randômico de descida, diferindo deste por aceitar o vizinho gerado aleatoriamente, se ele for melhor ou igual à solução corrente. Assim, o método tem condições de percorrer caminhos de descida que passam por regiões planas. Se ocorrer melhora no valor da função objetivo do agente, ele será atualizado.

**4.6. OPERADOR DE RECOMBINAÇÃO**

O operador de recombinação é o responsável pelo processo de cooperação entre os agentes. Para a realização dessa tarefa, utilizou-se a idéia de reconexão por caminhos.

A técnica de reconexão por caminhos, proposta originalmente por [3], baseia-se na idéia de que entre duas soluções, denotadas por solução inicial  $s'$  e solução guia  $s''$ , pode-se traçar um caminho que as una, de modo que as soluções desse caminho contenham atributos dessas soluções. As características do caminho são especificações a respeito dos atributos que são acrescentados ou retirados para alcançar uma solução a partir da outra.

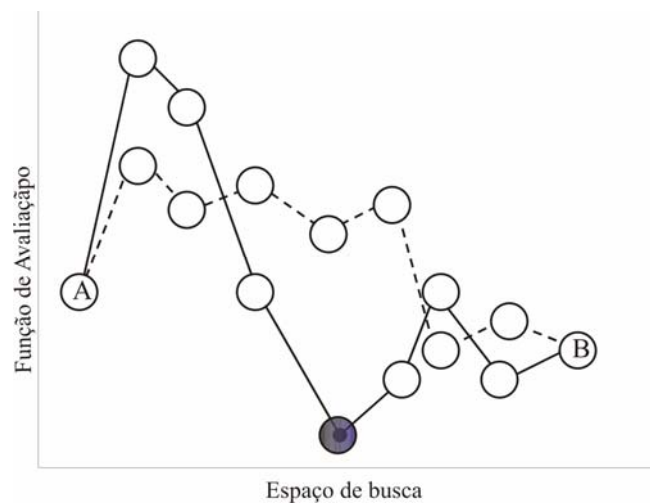


Figura 3: Representação da reconexão por caminhos entre duas soluções

A figura 3 mostra o mecanismo de reconexão por caminhos no caso de duas soluções: a solução A denominada solução inicial e a solução B denominada solução guia. Ambas soluções estão unidas por dois caminhos, um em traço descontínuo representando o caminho original que descreve o algoritmo de busca local para obter A e B, e outro em traço contínuo que representa o caminho obtido através da aplicação da técnica de reconexão por caminhos. No eixo das abscissas representamos os movimentos que se vão realizando, enquanto isso no eixo das ordenadas o valor da função objetivo, supondo um problema de minimização.

Tal como mostra a figura 1, a idéia na qual se baseia o método é que ao realizar movimentos da solução A até a solução B, guiando-se unicamente por incorporar à solução A propriedades da solução B, seja obtida alguma solução intermediária que melhore,

simultaneamente, o valor das soluções A e B.

Para gerar os caminhos é necessário selecionar movimentos que cumpram os seguintes objetivos: ao começar pela solução inicial, os movimentos devem introduzir de forma progressiva atributos pertencentes à solução guia, ou reduzir a distância entre atributos da solução inicial e solução guia. Primeiramente, considera-se a criação de caminhos que unam duas soluções selecionadas  $s'$  e  $s''$ , restringindo a atenção somente na parte do caminho situado entre ambas as soluções, gerando, assim, uma seqüência de soluções intermediárias

$$s' = s_1, s_2, \dots, s_n = s''.$$

Para reduzir o número de opções a considerar, a solução  $s_{i+1}$  deve ser criada em cada passo do caminho a partir da solução anterior  $s_i$ , selecionando um movimento dentre os disponíveis, deixando um menor número de movimentos candidatos para selecionar em um passo seguinte do caminho que dirige a exploração até  $s''$ , reduzindo-se, assim, a distância entre  $s'$  e  $s''$ .

Pode acontecer que em um novo caminho sejam encontradas soluções que não melhorem a qualidade das soluções extremas consideradas, no entanto algumas delas podem ser pontos de partida para alcançar outras soluções que sejam melhores que as originais. Por esta razão é interessante e valioso examinar os entornos das soluções ao longo do caminho construído, já que podem proporcionar novos pontos de partida para lançar uma busca adicional.

No presente trabalho, para cada filho gerado, seleciona-se aleatoriamente dentre os agentes da população,  $s'$  e  $s''$ . Para a inserção de atributos de  $s''$  em  $s'$  considerou-se como movimento a troca da programação semanal dos horários do professor. Assim, a programação semanal dos horários do primeiro e do último professor de  $s'$  é substituído pelo seu equivalente em  $s''$  e uma busca local é realizada. Se a solução encontrada for uma solução viável, esta será um possível filho gerado. No próximo passo, a programação semanal dos horários dos dois primeiros e dos dois últimos professores de  $s'$  são substituídos pelos seus equivalentes em  $s''$  e realiza-se a busca local. O processo é realizado até que  $s'$  encontre  $s''$ . Ao final do processo o operador retorna a melhor solução encontrada pela reconexão por caminhos.

Como busca local associada à técnica utilizou-se o método randômico não-ascendente.

#### 4.7. OPERADOR DE MUTAÇÃO

O operador de mutação é responsável por resgatar uma informação perdida ou mesmo a inclusão externa não contida em nenhum dos agentes. Basicamente, esse tipo de operador gerará um novo agente, mediante a modificação parcial de um agente existente. O movimento realizado pelo operador de mutação consiste em escolher aleatoriamente um dia da semana e trocar um horário de todos os professores por um outro horário, do mesmo dia, ou de outro, desde que a troca altere o valor da função objetivo. Esse tipo de movimento garante que o quadro de horários continuará viável após a mutação.



Figura 3: Pseudocódigo AlgoritmoHibrido aplicado ao Problema de Horário Escolar.

```

procedimento AlgoritmoHibrido(nagentes;maxDesvio; probReproducao; probMutacao)
1  pop: Agente[ ], popSobrevivente: Agente[ ];
2  desvio;
3  pop ← GeraPopInicial (nagentes, pop);
4  enquanto (desvio > maxDesvio) faca
5    ngeracoes ← ngeracoes +1;
6    para (i=1) ate (nagentes) faca
7      pop[i] ← operadorLocal(pop[i]);
8    fim-para;
9    AvaliaPop(pop);
10   para (filho=nagentes) ate (nagentes*2) faca //realizar reprodução
11     selecionaPais(pop);
12     se (randomico(0,1) < probReproducao) entao
13       pop[filho] ← operadorReproducao(Agente1,Agente2);
14     fim-se;
15   fim-para;
16   AvaliaPop(pop);
17   para (i=1) ate (nagentes*2) faca //realizar mutação
18     se (randomico(0,1) < probMutacao) entao
19       pop[i] ← operadorMutacao(pop[i]);
20     fim-se;
21   fim-para;
22   AvaliaPop(pop);
23   para (i=1) ate (nagentes) faca // definir população sobrevivente
24     popSobrevivente[i] ← selecao(pop);
25   fim-para;
26   para (i=1) ate (nagentes) faca //atualizar população
27     pop[i] ← popSobrevivente[i];
28   fim-para;
29   calculaDesvio;
30 fim-enquanto;
fim AlgoritmoHibrido.

```

#### 4.8. ESTRUTURA DE SELEÇÃO

A seleção dos agentes para a próxima geração é feita através da análise do seu grau de adaptação. Para esse fim, utiliza-se um mecanismo de roleta russa, no qual os mais aptos possuem maior probabilidade de sobreviverem, sendo que todos, mesmo os menos aptos, possuem alguma chance de sobrevivência. Os valores do grau de adaptação de cada agente  $s$  são convertidos em números, na faixa entre zero e um, através da função de adaptação:

$$f_{Aptidao}(s) = \frac{f_{Aptidao}(s)}{\sum_{i=1}^n f_{Aptidao}(s_i)} \quad (3)$$

Na equação 3, a função de aptidão avalia a possibilidade de sobrevivência de cada agente. É definida na forma:

$$f_{Aptidao}(s) = \bar{f}(pop) + \alpha\sigma(f(pop)) - f(s) \quad (4)$$

- $f_{Aptidao}(s)$  é o valor da função de aptidão da solução  $s$  ;
- $\bar{f}(pop)$  é o valor médio da função de custo da população, calculada como:

$$\bar{f}(pop) = \frac{\sum_{i=1}^n f(s_i)}{n} \quad (5)$$

sendo  $n$  o número de agentes da população.

- $\alpha$  é um parâmetro de ponderação da influência da diversidade da população sobre cada agente em particular;
- $\sigma(fpop)$  é o desvio padrão dos valores da função de custo da população;
- $f(s)$  é a função de custo da solução  $s$  .

A partir do processo de avaliação, pode-se encontrar o grau de adaptação dos agentes e eliminar aqueles que não são aptos ao meio.

## 5. RESULTADOS COMPUTACIONAIS

O algoritmo foi implementado na linguagem C++, usando o compilador C++ Builder 6.0, e testado em um microcomputador Pentium IV, 2.8 GHz, com 256MB de memória RAM. Inicialmente, foi submetido a uma bateria preliminar de testes, para calibrar os diversos parâmetros existentes (número de iterações máximo para o método de descida, número de iterações máximo utilizado no ILS, probabilidade de recombinação, probabilidade de mutação), bem como escolher os pesos mais adequados para a função de avaliação. Os resultados obtidos estão apresentados nas tabelas 2a e 2b e representam os valores efetivamente utilizados para a determinação dos melhores quadros de horários, segundo os termos propostos.

Tabela 2: Dados para implementação do Algoritmo.

(a) Parâmetros utilizados no Algoritmo

Parâmetro	Valor
Tamanho da População	10
MaxDesvio	0,1
Probabilidade de Reprodução	0,8
Probabilidade de Mutação	0,2
Iterações ILS	4
Iterações busca local	p x hx x nTurmas

(b) Pesos para função de avaliação

Peso	Valor
$\omega$	100
$\delta$	50
$\rho$	1
$\alpha$	5
$\beta$	2
$\gamma$	10
$\sigma$	1

Para avaliar o algoritmo, foram utilizados dados relativos à programação de horários dos cursos de ensino médio e ensino técnico do CEFET Rio Pomba, nos anos de 2005 e 2006.

No processo de elaboração dos horários dos professores dessa instituição de ensino, além dos requisitos essenciais para que um horário seja implementado, quais sejam, não ocorrer sobreposição de professores ou turmas e fazer com que a carga-horária semanal de cada disciplina seja cumprida, vários outros requisitos devem ser observados para a melhoria do quadro de horário, como:

- por razões pedagógicas, as aulas de educação física devem ser geminadas (ou duplas) e ocorrerem nos dois últimos horários diários;
- o número máximo de aulas diárias de uma mesma disciplina para uma mesma turma deve ser respeitado;
- o número de janelas na programação do horário dos professores deve ser diminuído;
- deve-se evitar a quebra de aulas no quadro de horários. Considera-se como quebra de aulas a ocorrência de aulas, para uma mesma classe, separadas por um intervalo;
- deve-se atender ao maior número possível de aulas geminadas;
- minimizar o número de dias em que cada professor ministra aulas;
- respeitar os horários de indisponibilidade dos professores.

Além das observações acima mencionadas, as classes de primeiro e segundo ano dos cursos técnicos das áreas de agropecuária possuem um dia da semana reservado para treinamento operacional. Neste caso específico, deve-se evitar que seja programadas aulas para a segunda-feira de manhã e sexta-feira à tarde.

A tabela 3 mostra os dados referentes a cada um dos problemas-teste.

Tabela 3: Dados dos problemas teste

Teste	Nº Prof.	Nº Turmas	Nº Horários/Turno	Duplas Requeridas	Quant. Aulas	Dens.
1	17	7	20	70	140	0,38
2	16	12	20	67	240	0,69
3	18	12	20	53	240	0,58
4	51	26	20	141	520	0,44

O problema 1 refere-se ao planejamento de aulas dos professores dos cursos técnico em informática e técnico em meio ambiente (turnos tarde e noite). Os problemas 2 e 3 referem-se ao planejamento das aulas dos professores do curso de ensino médio (turnos manhã e tarde) nos anos de 2005 e 2006. O problema 4 refere-se ao planejamento das aulas dos professores dos cursos técnicos da área de agropecuária, do curso técnico em informática e do ensino médio (turnos manhã e tarde). No problema um todas as aulas são geminadas. Nos demais problemas, considera-se que todos os professores com mais de duas horas-aula semanais para uma mesma turma gostariam de ter, pelo menos, uma aula geminada. Em todos os casos são gerados os quadros de dois turnos simultaneamente. A quantidade de aulas duplas requeridas pelos professores para cada um dos problemas teste encontra-se na coluna Duplas Requeridas. A densidade do quadro de horários para cada problema teste, dada pela razão entre a carga-horária dos professores, somada ao número de horários indisponíveis, e o número de horários reservados para a realização das aulas de todas as turmas é mostrada na coluna Dens.

Tabela 4: Solução Inicial

Teste	fo	Sobreposição de aulas	Excesso aulas/dia	Excesso Vezes	Janelas	Duplas não atendidas	Quebra aulas
1	8,3	0	0	0,8	0	0,3	0
2	216,7	0	0	1,8	34,4	31,9	19,6
3	215,9	0	0	2,3	28,5	33,9	20,4
4	322	0	0	3	49	59	27

Tabela 5: Solução Final

Teste	Fo	Sobreposição de aulas	Excesso /dia	Excesso Vezes	Janelas	Duplas não atendidas	Quebra aulas	Tempo (s)	Nº de Gerações
1	0	0	0	0	0	0	0	156,45	1
2	4	0	0	0	0	4	0	23921,28	35

3	3	0	0	0	0	3	0	21356,30	36
4	20	0	0	0	7	6	0	86415,18	24

As tabelas 4 e 5 apresentam, respectivamente, a média dos resultados da população inicial e o resultado final, para os 4 problemas-teste.

O comportamento do algoritmo híbrido em uma execução do problema 2, com relação à evolução da melhor solução, é mostrado na figura 4. Nesta figura a linha tracejada indica os valores da função de custo da população durante as primeiras gerações. A linha contínua indica a evolução da melhor solução gerada ao longo das gerações.

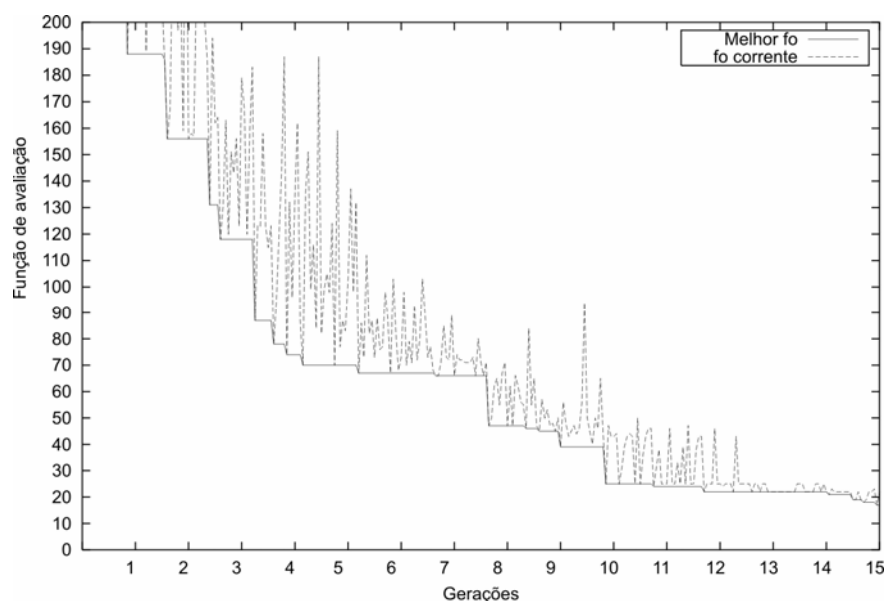


Figura 4: Comportamento do algoritmo híbrido em uma execução do problema 2

Para efeito de comparação, são apresentados na tabela 6 os resultados da função de avaliação para a aplicação exclusiva do algoritmo ILS aos mesmos problemas teste, para 200 iterações.

Tabela 6: Solução Final ILS

Teste	Fo	Sobreposição de aulas	Excesso/dia	Excesso/vezes	Janelas	Duplas não atendidas	Quebra aulas	Tempo (s)
1	0	0	0	0	0	0	0	32,28
2	25	0	0	1	1	2	3	1367,81
3	19	0	0	1	0	1	4	1382,75
4	633	0	0	16	56	64	41	2419,7

Constatou-se que o algoritmo híbrido consegue atingir soluções melhores que o algoritmo ILS puro, mesmo com o acréscimo de componentes, como no problema-teste 4. No problema-teste 1 ambos os algoritmos encontram ótimos resultados. Já os testes realizados com os problemas 2 e 3, apesar dos resultados do ILS serem bons, o mesmo poderia ser melhorado, se o algoritmo fosse executado por um número maior de iterações, o que faria com que os resultados dos dois algoritmos ficassem muito próximos. No entanto, a partir do momento que mais componentes são introduzidas ao problema, o algoritmo ILS já não consegue minimizar a insatisfação dos professores. Devido às diferenças entre um algoritmo populacional e um algoritmo não-populacional não temos como comparar o tempo gasto pelos algoritmos.

As tabelas 7 e 8 apresentam um dos resultados encontrados pelo algoritmo híbrido para o teste 2. Nesse resultado em específico pode-se observar que, de todos os requisitos solicitados para a melhoria do quadro de horários, o algoritmo não conseguiu atender apenas à configuração de aulas duplas solicitadas. Isso pode ser observado nas linhas referentes ao professor 12, no turno da tarde (1 dupla) e professor 14, no turno da manhã (1 dupla) e no turno da tarde (duas duplas).

Tabela 7: Quadro de Horário Ensino Médio 2005 - manhã

Prof	Semana Letiva																			
	Segunda				Terça				Quarta				Quinta				Sexta			
	H0	h1	h2	h3	h4	H5	h6	h7	h8	h9	H10	H11	h12	h13	h14	h15	h16	h17	h18	h19
0	I	I	1	1	I	I	5	5	I	I			I	I	4	4	I	I	0	0
1	I	I			I	I			I	I	9	9	I	I			I	I	11	11
2					4	4			F	F	F	F	4						4	4
3	5	4	0	0	9	9	11	11	1				F	F	F	F		4	1	5
4	I	I	I	I	I	I	I	I					11	4	0	0	1	1	5	9
5	0	0	5	5			1	1	5	5	0	1	F	F	F	F	5			
6	I	I	I	I	I	I	I	I	4	9					11	5	4	5		
7	I	I			I	I		0	I	I			I	I		1	F	F	F	F
8			4	4	5	1	0					0	1	9	5	11	F	F	F	F
9	F	F	F	F	1				9				0	1			0	11		
10					F	F	F	F		1	11	11		0	9	9	11	0	9	1
11	4				F	F	F	F	11	11	4	4	9	11			9	9		
12	11	11	9	9	11	11	9	9									F	F	F	F
13		5				5	4	4	I	I	I	I	I	I	I	I	F	F	F	F
14	1	1			0	0			0	0	1	5	5	5	1		F	F	F	F
15	9	9	11	11	I	I	I	I		4	5			I	I	I	I			

Tabela 8: Quadro de Horário Ensino Médio 2005 - tarde

Prof	Semana Letiva																				
	Segunda				Terça				Quarta				Quinta				Sexta				
	H0	H1	h2	H3	h4	H5	h6	H7	h8	h9	H10	h11	h12	h13	h14	h15	h16	h17	h18	H19	
0	I	I			I	I	2	2	I	I			I	I	3	3	I	I			
1	I	I			I	I	8	8	I	I	7	7	I	I	10	10	I	I	6	6	
2	7	6			6	6	7	7	F	F	F	F	7	7			6	6			
3				2	7	2	6	10	6	10	3	3	F	F	F	F	8	8	7		
4	I	I	I	I	I	I	I	I	8				6	10	2		7	2	3	3	
5	3						3	3			2	2	F	F	F	F	2				
6	I	I	I	I	I	I	I	I	7	6						6				7	
7			2	3	10											8	F	F	F	F	
8	8	2	3			7	10	6		3			2	6	7		F	F	F	F	
9	F	F	F	F	3	10			3	7						2		7	2	8	
10			10	10	F	F	F	F			8	8		2	8		3	3	10	2	
11	10	8	7	7	F	F	F	F			6	6	8	8	6	7	10	10			
12		10	8	8	8	8					10	10	10				F	F	F	F	
13			6	6					I	I	I	I	I	I	I	I	F	F	F	F	
14	2	3			2	3			2	2			3	3			F	F	F	F	
15	6	7			I	I	I	I	10	8				I	I	I	I			8	10

## 6. CONCLUSÕES

Este trabalho apresenta uma aplicação de um algoritmo evolutivo híbrido para o Problema de Horário Escolar. Considerou-se, para análise, os dados referentes aos professores

do Centro Federal de Educação Tecnológica de Rio Pomba nos anos de 2005 e 2006, para a programação de dois turnos em simultâneo.

Procurou-se, neste estudo, satisfazer ao maior número de professores quanto à quantidade de dias em que suas aulas são distribuídas, minimização do número de janelas e quebras de aula, bem como atender a um certo número de aulas duplas.

A qualidade da solução final está relacionada com uma boa estrutura de vizinhança; com a maneira de se avaliar os requisitos não atendidos e o tempo computacional disponibilizado para a execução. No caso dos testes realizados, o algoritmo híbrido proposto mostrou-se bastante eficaz, conseguindo encontrar, em todos os testes, quadros de horários de alta qualidade. O primeiro teste, referente aos cursos de Informática e Meio Ambiente, no qual todas as aulas são duplas, o algoritmo consegue encontrar sua melhor solução, na construção da população inicial.

O tempo de execução cresce consideravelmente à medida que se aumentam as componentes do problema. O esforço computacional do algoritmo híbrido, causado pela utilização da reconexão por caminhos no operador de recombinação e pelo operador local, é compensado pela alta qualidade da solução final. A utilização de reconexão por caminhos, na fase de reprodução, garante a viabilidade dos quadros de horários filhos e a qualidade dos mesmos, no que se refere ao atendimento da satisfação dos professores.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Blum, C. & Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, vol. 35, n. 3, pp. 268-308.
- [2] Burke, E. K. & Petrovic, S., 2002. Recent research directions in automated timetabling. *European Journal of Operational Research*, vol. 140, n. 2, pp. 266-280.
- [3] Glover, F., 1996. Tabu search and adaptative memory programming - advances, applications and challenges. In Barr, R., Helgason, R., & Kennington, J., eds, *Interfaces in Computer Sciences and Operations Research*, pp. 1-75. Kluwer Academic Publishers.
- [4] Merz, P. & Freisleben, B., 1999. A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem. In Angeline, P., ed, *1999 Congress on Evolutionary Computation (CEC'99)*, pp. 2063-2070, Piscataway, NJ. IEEE Press.
- [5] Moscato, P. & Cotta, C., 2003. An introduction to memetic algorithms. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, vol. , n. 19, pp. 131-148.
- [6] Moscato, P., 1989. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA.
- [7] Radcliffe, N. & Surry, P., 1994. Formal Memetic Algorithms. In Fogarty, T., ed, *Evolutionary Computing: AISB Workshop*, volume 865 of *Lecture Notes in Computer Science*, pp. 1-16. Springer-Verlag, Berlin.
- [8] Schaerf, A., 1999. A survey of automated timetabling. *Artificial Intelligence*, vol. 13, n. 2, pp. 87-127.
- [9] Souza, M. J. F., 2000. Programação de Horários em Escolas: Uma Aproximação por Metaheurísticas. Tese de Doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.