LEONARDO APARECIDO CISCON

O PROBLEMA DE GERAÇÃO DE HORÁRIOS: UM FOCO NA ELIMINAÇÃO DE JANELAS E AULAS ISOLADAS

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2006

LEONARDO APARECIDO CISCON

O PROBLEMA DE GERAÇÃO DE HORÁRIOS: UM FOCO NA ELIMINAÇÃO DE JANELAS E AULAS ISOLADAS

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:

Otimização e Heurísticas

Orientador:

Prof. Guilherme Bastos Alvarenga

LAVRAS

MINAS GERAIS – BRASIL

2006

Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca Central da UFLA

Ciscon, Leonardo Aparecido

O Problema de Geração de Horários: Um Foco na Eliminação de Janelas e Aulas Isoladas / Leonardo Aparecido Ciscon – Minas Gerais, 2006.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Heurísticas. 2. Otimização Combinatória. 3. Geração de Horários. 1. CISCON, L. A. II. Universidade Federal de Lavras. III. Título.

LEONARDO APARECIDO CISCON

O PROBLEMA DE GERAÇÃO DE HORÁRIOS: UM FOCO NA ELIMINAÇÃO DE JANELAS E AULAS ISOLADAS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 26 de Abril de 2006.

____
Prof. Fortunato Silva de Menezes

____
Prof. Ricardo Martins de Abreu Silva

____
Prof. Guilherme Bastos Alvarenga
(Orientador)

LAVRAS

MINAS GERAIS – BRASIL

Quando preciso tomar uma decisão, sempre pergunto:
"O que seria mais divertido?"
Peggy Walker

Dedico a Deus, por tudo.
Meus pais (José Tarcísio e Ana Maria) e irmãos (Carlos Eduardo e Giancarlo)
Por todo carinho, educação e dedicação.
Tios, tias, padrinhos, madrinhas e avó pelo apoio.
À Michelle por me fazer uma pessoa muito mais feliz.

AGRADECIMENTOS

Primeiramente, a minha Família,
Por tudo!

A toda a Turma 10, em especial Adriano (Bob), Anderson (Arroz),
Alessandro (Gordo-Careca), Daniel (Dedão), Ricardo (Lambari),
Tati (Tatiane), Renato (Alemão), Marcos (Kescara), Marcus (Carneiro),
Daniel (Garcia), Marlon, Maykel, Guilherme (Integér), Renato (Renatão), Fabrício
Por terem convivido comigo estes exatos 4 anos e 4 dias!

A todos da SWFactory e SWQuality
Por tudo que aprendi extra-classe e pelas amizades que vão durar para a vida toda!

Aos meus calouros da República: Guilherme (nariz), Henrique (orelha),
Heitor (barriga) e PC (careca)
Por terem sido meus divertidos e prestativos calouros!

A todos os professores, especialmente, Ana Cristina, Guilherme, Rêmulo e Olinda

A diretoria e todos os professores e funcionários da E. E. Padre Rogério Abdala

A algumas pessoas em especial: Vanessa, Breno, Humberto, Roberto, Robissson,
Márcio Henrique e Vanda Maria
Cada uma sabe exatamente o porquê de estar sendo agradecida!

A todos os meus amigos de Lavras, Varginha, Itajubá e Monsenhor Paulo!

Ao limite de R$ 200,00 do Banco do Brasil.
A todas as caronas que consegui.
Ao Frozen (antigo TheFlash) e ao Black-Label
Sem os quais eu não teria feito nenhum dos trabalhos da graduação!

E por fim... Agradeço muito a Michelle,
Por me mostrar que trabalho(s) e monografia não são tudo na vida!
E, claro, por me ensinar o melhor dos sentimentos!

RESUMO

Resumo: O Problema de Geração de Horários Escolares, conhecido na literatura como *Timetabling Problem*, trata da definição dos horários para todas as aulas de uma escola, considerando um conjunto limitado de horários e satisfazendo um conjunto restrições. A solução manual do problema além de ser trabalhosa e lenta, pode ocasionar soluções de qualidade muito ruim. Mesmo encontrar um quadro viável de horários é um problema NP-Difícil, dificultando o uso de técnicas exatas para instâncias de ordem mais elevada. Vários métodos de otimização e heurísticas têm sido propostos, mas a maioria encontra problemas ao lidar com um grande número de restrições, necessárias aos problemas do mundo real. Entre as técnicas mais recentes, as meta-heurísticas tem se destacado por possibilitar bons resultados em tempo aceitável de processamento. Busca Tabu, Recozimento Simulado e Algoritmos Genéticos têm sido as meta-heurísticas mais utilizadas. Entretanto, objetivos e restrições importantes têm sido desconsiderados, tais como: horários indesejáveis do professor, formação de janelas e aulas isoladas e prestígio do professor, dificultando a utilização dessas propostas de solução em problemas reais. Neste trabalho, dois importantes objetivos são tratados, a eliminação de janelas e aulas isoladas, buscando horários com mais aceitação dos professores. Esta generalização se mostrou essencial para a utilização da solução proposta na geração de horários para duas escolas escolhidas como estudo de caso. No contexto abordado, a utilização e aceitabilidade das soluções encontradas mostram a importância da generalização considerada e a validade da proposta atual.

Palavras-chaves: *timetabling*, algoritmos genéticos, otimização, geração de horários

ABSTRACT

Absctract: The *Timetabling problem* consists in scheduling a set of classes to a fixed number of time *slots* subject to a great number of constraints. The manual problem solution is painful and slow and many times the results aren’t satisfactory. Even to find a feasible timetable solution is NP-hard. Consequently, the use of exact methods is not appropriate to high orders instances. Many exact methods and heuristics have been proposed to approach the *timetabling* problem, however, the main problem from these methods have been to handle efficiently many real world constraints and objectives. From the recent approaches, the meta-heuristics highlight to find good quality results in reasonable execution time. Tabu Search, Simulated Annealing and Genetic Algorithm are the main meta-heuristic methods utilized. However, important constraints and objectives have been neglected, making hard the use of these solutions in the real world. In this paper, two important objectives are considered: the idle time windows and isolated classes minimization, possibiliting a timetable with higher quality of accepting. The generalization proposed in this paper has demonstrated very important in the solution for two studied case schools analyzed. In context, the real utilization and acceptability of the produced solutions show the generalization importance and current proposal importance.

Keywords: *timetabling*, genetic algorithms, heuristics

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG – Algoritmo Genético

AGGH – Algoritmo Genético de Geração de Horários

EEPRA – Escola Estadual Padre Rogério Abdala

EMAB – Escola Municipal Álvaro Botelho

GRASP – Procedimento de busca adaptativa gulosa e randomizada

RNA – Método Randômico Não Ascendente

SGBD – Sistema Gerenciador de Banco de Dados

UFLA – Universidade Federal de Lavras

UML – Linguagem de Modelagem Universal

1. INTRODUÇÃO

Um grande problema presente nas instituições educacionais em todo início de período letivo é a geração dos horários escolares. A programação dos horários torna-se uma tarefa difícil devido ao grande número de possibilidades e a necessidade de se respeitar uma série de restrições, muitas vezes conflitantes entre si.

As instituições de ensino geralmente necessitam de muitos dias programando os seus horários manualmente e chegam à maioria das vezes a um horário inviável. Geram-se horários não ideais à realidade da escola, além de grande esforço por parte de seus funcionários.

Uma solução que seja capaz de propor um horário com qualidade e que não viole as restrições é muito necessária, uma vez que é crescente o número de novos estabelecimentos de ensino e o problema estar presente em todos eles.

Adicionalmente, este é um problema desafiador em otimização combinatória. Devido a sua complexidade computacional e a conseqüente dificuldade de tratá-lo com algoritmos exatos em tempo hábil, normalmente estes problemas são tratados com métodos heurísticos. A família de problemas *timetabling* se caracteriza por um espaço de busca muito grande acompanhado de um conjunto de restrições que devem ser respeitadas.

O objetivo desse trabalho é apresentar um algoritmo genético para o tratamento do problema de geração de horários escolares, o *timetabling*, considerando o aspecto de eliminação de janelas e aulas isoladas no horário dos professores. Estes têm sido os grandes dificultadores na geração dos horários, bem como ponto fundamental na aceitabilidade pelos professores envolvidos. Janelas são os horários ociosos entre duas aulas de um mesmo turno na grade horária de um professor, ao passo que as aulas isoladas são aquelas presentes isoladamente em um dia no horário do professor. No contexto das escolas de ensino fundamental e médio é clara a importância das restrições citadas, contudo, não há referências a trabalhos que buscam a minimização simultânea destas, pelo que se tem conhecimento.

1.1. Objetivos e Justificativas

O tratamento do problema de geração de horários em escolas carece de bons trabalhos na literatura. Apesar de se encontrar ferramentas disponíveis, poucas tratam de maneira eficiente as restrições reais existentes em escolas. Com este trabalho objetiva-se:

- Apresentação de um modelo matemático acerca do problema;
- Implementação de um algoritmo que solucione o modelo apresentado considerando as restrições mais comuns para a geração de um horário de qualidade;
- Consideração, de forma simultânea, de mais duas importantes restrições: a eliminação de janelas e de aulas isoladas no horário do professor;
- Desenvolvimento de uma interface gráfica amigável que utilize este algoritmo.

1.2. Organização do Trabalho

Este trabalho está organizado da seguinte forma: O capítulo 2 traz a descrição do problema abordado no trabalho, métodos heurísticos, algoritmos genéticos além de uma breve revisão dos trabalhos correlatos. O capítulo 3 apresenta a metodologia do trabalho, explicando a proposta de solução que utiliza algoritmo genético considerando os objetivos já mencionados: janelas e aulas isoladas. Neste capítulo também é apresentado o sistema desenvolvido O capítulo 4 mostra os resultados obtidos e uma discussão relativa a estes resultados.   Já o capítulo 5 apresenta as conclusões e os trabalhos futuros. O capítulo 6 apresenta as referências bibliográficas consultadas. Por fim o capítulo 7 mostra um dos horários gerados pelo algoritmo para instâncias reais de uma escola de Monsenhor Paulo / MG.

2. REVISÃO DE LITERATURA

2.1. Descrição do Problema

O problema de geração de horários escolares, citado na literatura como uma das instâncias do *Timetabling Problem*, consiste em arranjar encontros entre professores e alunos em um período de tempo previamente fixado, tipicamente uma semana, de modo a satisfazer um conjunto de restrições que podem ser de vários tipos.

Segundo Souza (2000), a solução manual deste problema é uma tarefa árdua e normalmente requer vários dias de trabalho. A elaboração de um quadro de horários por esta via pode demandar duas semanas de trabalho em uma escola secundária ou até um mês em uma universidade. Ainda segundo Souza (2000), a solução obtida pode ser insatisfatória com respeito a diversos aspectos.

No entanto, alguns autores, entre eles, Schaerf (1999) e Werra (1985) acreditam que problemas de geração de horários não podem ser completamente automatizados. Há duas justificativas para isso: por um lado, há razões que não podem ser facilmente expressas em um sistema automatizado, que tornam um quadro de horário melhor que o outro. Por outro, uma vez que o espaço de soluções é vasto, a intervenção humana pode conduzir a busca em direção a regiões promissoras, nas quais o sistema, por si só, dificilmente teria condições de chegar.

Muitas variantes do problema têm sido propostas na literatura, e diferem umas das outras pelo tipo de instituição de ensino envolvida, universidades ou escolas médias, e pelo tipo de restrições impostas ao problema. Schaerf (1999) cita três classes de problemas:

- *School Timetabling*: seqüenciamento semanal das aulas de uma escola, evitando que professores e alunos tenham mais de uma aula simultaneamente. Basicamente, existe um conjunto de turmas, um conjunto de professores e um conjunto de horários reservados para a realização das aulas.
- *Course Timetabling*: diz respeito à alocação de aulas de uma universidade típica. Basicamente há um conjunto de disciplinas (Cálculo I, Engenharia de Software, Genética, etc.) e para cada disciplina um número de aulas. Há, também, um conjunto de cursos (Engenharia de Alimentos, Ciência da Computação, Agronomia, etc). Cada curso envolve um conjunto de disciplinas. Os alunos matriculam-se em turmas das disciplinas de seu curso. Uma turma de uma disciplina pode ter

estudantes de cursos diferentes. Há, por último, um conjunto de horários disponibilizados para a realização das aulas. O problema, então, trata do seqüenciamento semanal das aulas evitando a simultaneidade de disciplinas e respeitando os horários disponibilizados;

- *Examination Timetabling*: seqüenciamento de exames de um conjunto de cursos em uma universidade, evitando exames simultâneos de cursos com estudantes em comum, e espalhando os exames o máximo possível. Segundo Souza (2000), apesar da similaridade com o *course timetabling*, eles se distinguem, sobretudo pela natureza das restrições envolvidas. Entre as restrições típicas deste tipo de problemas, destaca-se: nenhum estudante pode fazer mais do que certo número de exames por dia, exames de certas disciplinas não podem preceder a exames de determinadas disciplinas, alguns exames têm que ser realizados em um mesmo horário.

O objetivo deste trabalho é produzir um método para tratar especificamente a variante citada como *School Timetabling*, porém com características em que ela se apresenta em escolas de ensino fundamental e médio no Brasil.

Tipicamente, estas escolas atendem a um determinado número de turmas que é limitado por sua capacidade física. As turmas são conjuntos disjuntos de estudantes que têm um mesmo currículo. Normalmente há mais turmas do que salas de aula, mas as escolas trabalham em mais de um período por dia, normalmente os períodos matutino, vespertino e noturno.

Cada turma possui uma relação de disciplinas que têm certo número de aulas dependendo do currículo do curso e série as quais a turma pertence. O caso mais comum é que quantidade de aulas de cada turma preenche completamente a semana, isto é, as turmas têm aulas em todos os horários de seu período, todos os dias da semana. Há casos, contudo, que podem ocorrer janelas também no horário da turma (Ciscon et al., 2005).

O número de horários do período e o número de dias da semana, multiplicados pelo número de turmas da escola nos dão o número total de aulas ministradas naquele período. Se forem considerados todos os períodos do dia teremos todas as aulas ministradas na escola (Ciscon et al., 2005).

Essas aulas são ministradas por um conjunto de professores. Cada professor tem seu próprio número de matérias, bem como as turmas para as quais lecionará. Muitas vezes os professores trabalham em mais de uma escola e em cada uma delas ministram um diferente número de aulas (Ciscon et al., 2005). O objetivo básico é fazer um quadro de horário, em geral semanal, de tal forma que: 1) As cargas horárias de todas as matérias sejam cumpridas; 2) Cada turma não tenha aula com mais de um professor ao mesmo tempo, e; 3) Um professor não dê aula para mais de uma turma em um mesmo horário.

Neste contexto, o presente trabalho foca a geração de horário minimizando a quantidade de janelas e aulas isoladas nos horários dos professores. Com isso busca-se a geração de um horário otimizado para o professor a fim de facilitar a execução de outras possíveis atividades.

2.1.1. Complexidade do Problema

O problema de programação de horários é NP-Difícil, conforme Even et al. (1976).

Cooper & Kingston (1996) demonstram para uma série de problemas de horários que aparecem comumente na prática estarem todos na classe NP-Difícil. Entretanto, algumas instâncias do problema podem ser resolvidas em tempo polinomial. Este é o caso do problema de programação de horários em escolas, em sua versão básica, quando turmas e professores estão sempre disponíveis. A maioria dos casos resolvíveis é, no entanto, muito especial e não incluem as restrições mais comuns.

Em vista disso, justifica-se a abordagem do problema de programação de horários por métodos heurísticos, os quais, não garantem a existência de uma solução viável, mesmo que essa exista, e tampouco sua otimalidade. Para o problema de geração de horários, conceitua-se como solução viável aquele horário que pode ser utilizado pela escola, não necessariamente sendo o mais adequado possível. Isto é, existem restrições de inviabilidade e restrições de qualidade, sendo que a satisfação apenas das primeiras restrições já qualifica um horário como viável.

2.2. Métodos Heurísticos

Em Problemas de Otimização Combinatória, cujo universo de dados é grande, existe um número muito extenso de combinações, tornando inviável a análise de todas possíveis soluções, visto que o tempo computacional para uma enumeração completa seria demasiadamente longo. Neste sentido, têm-se as heurísticas, também conhecidas como algoritmos heurísticos, que são métodos que compõem uma gama relativamente nova de soluções para Problemas de Otimização Combinatória.

Segundo Papadimitriou & Steiglitz (1982), as heurísticas são quaisquer métodos de aproximação sem uma garantia formal de seu desempenho.

As heurísticas, apesar de não garantirem encontrar a solução ótima para um problema, procuram por soluções consideradas de boa qualidade em um tempo computacional razoável.

2.2.1. Métodos de Busca Local

De acordo com Souza (2000), as técnicas de busca local em problemas de otimização constituem uma família de técnicas baseadas na noção de vizinhança. Em linhas gerais, a partir de uma solução inicial S_0 (a qual pode ser obtida por outra técnica ou gerada de forma aleatória), uma técnica de busca local navega pelo espaço de busca, passando, iterativamente, de uma solução para outra que seja vizinha. Dentre as técnicas existentes, Souza (2000) cita:

- Método de Descida
- Método Randômico de Descida
- Método Randômico Não Ascendente (RNA)

Para mais informações de métodos de busca local, consulte Souza (2000).

2.2.2. Meta-heurísticas

Ribeiro (1996) define meta-heurísticas como procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual tem que ser modelada para cada problema específico.

Contrariamente às heurísticas convencionais, as meta-heurísticas são de caráter geral e têm mais condições de escapar de ótimos locais.

Serão apresentadas, a seguir, algumas importantes meta-heurísticas.

2.2.2.1. GRASP (Greedy Randomized Adaptive Search Procedure)

Proposto por Feo & Resende (1995), o GRASP (Procedimento de busca adaptativa, gulosa e randomizada) é um método iterativo que consiste de duas fases: uma fase de construção, na qual uma solução é gerada, elemento a elemento, e de uma fase de busca local, na qual um ótimo local na vizinhança da solução construída é pesquisado, isto é, repete-se várias vezes a aplicação de busca local em soluções iniciais diferentes geradas aleatoriamente. A melhor solução encontrada ao longo de todas as iterações GRASP realizadas é retornada como resultado.

Segundo Feo & Resende (1995), em GRASP a aleatoriedade é utilizada em conjunto com informação heurística que leva em conta critérios de otimização relativos à solução parcial. A idéia de iteração, na sua forma básica, difere bastante das outras meta-heurísticas, pois a solução da iteração corrente não é resultado das anteriores, sendo fruto unicamente da semente aleatória da iteração.

No GRASP básico, não existe deterioração controlada da solução corrente como mecanismo de fuga do ótimo local, como ocorre por exemplo no Recozimento Simulado (deterioração controlada estocasticamente) ou Busca Tabu (deterioração controlada por estruturas de memória). Neste métodos, a idéia é começar de uma solução inicial e percorrer um único caminho dentro da vizinhança explorável, eventualmente deteriorando a solução corrente (Recozimento Simulado) ou o ótimo local alcançado (Busca Tabu) até que se pare em alguma solução, que será a final.

Ainda segundo Feo & Resende (1995), o procedimento GRASP procura conjugar bons aspectos dos algoritmos puramente gulosos, com aqueles dos procedimentos aleatórios de construção de soluções.

2.2.2.2. Recozimento Simulado

Técnica de busca local probabilística, proposta originalmente por Kirkpatrick et al. (1983), que se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. Isto é, conforme Noronha (2000) em analogia a física da matéria: levando um cristal a sua temperatura de fusão, as moléculas estão desordenadas e se agitam livremente. Ao resfriar-se a amostra de maneira infinitamente lenta, as moléculas vão adquirir a estrutura cristalina estável que tem um nível de energia mais baixo possível.

Conforme Aarts & Korst (1989) a analogia com a otimização (combinatória ou não) é bastante direta. Os estados da matéria são as soluções realizáveis, a quantidade objetiva substitui a energia, os estados metaestáveis da matéria sendo ótimos locais e a estrutura cristalina corresponde ao ótimo global.

Esta técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um *loop* que gera aleatoriamente, em cada iteração, um único vizinho *s'* da solução corrente *s*.

Downsland (1993) chama de Δ a variação de valor da função objetivo ao mover-se para uma solução vizinha candidata, isto é, $\Delta = f(s') - f(s)$, sendo que o método aceita o movimento, e a solução vizinha passa a ser a nova solução corrente, se $\Delta < 0$. Caso $\Delta >= 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de temperatura e que regula a probabilidade de aceitar soluções de pior custo.

Observa-se que a probabilidade de aceitar movimentos que degradam o valor da função objetivo decresce com a temperatura (T). Isto implica que quanto mais a temperatura cai menos movimentos degradantes tem uma chance de serem aceitos.

Segundo Downsland (1993), a temperatura T assume, inicialmente, um valor elevado T_0 e o procedimento pára quando a temperatura chega a um valor próximo de zero e nenhuma solução que piore o valor da função objetivo é mais aceita, isto é, quando o sistema está estável.

Mais informações em Downsland (1993) e Kirkpatrick et al. (1983).

2.2.2.3. Busca Tabu

Segundo Marinho (2005), o método de Busca Tabu, proposto independentemente por Glover (1986) e Hansen (1986) é um procedimento iterativo para a solução de problemas de otimização combinatória que aceita movimentos de piora para tentar escapar de ótimos locais distantes de um ótimo global.

Ainda segundo Marinho (2005), começando com uma solução inicial s_0, o método explora a cada iteração, um subconjunto V da vizinhança $N(s)$ da solução corrente s. Considerando o problema de minimização de uma função $f(.)$, o membro s' de V com menor valor nessa região torna-se a nova solução corrente mesmo que s' seja pior que s, isto é que $f(s') > f(s)$. No entanto, a melhor solução gerada até o momento, s^*, é sempre armazenada.

O critério de escolha da nova semente é utilizado para tentar escapar de um mínimo local. Esta estratégia, entretanto, pode fazer com que o algoritmo cicle, isto é, que retorne a uma solução já gerada anteriormente. De forma a evitar que isto ocorra, existe um mecanismo chamado memória de curto prazo ou lista tabu. O objetivo desta lista é tentar evitar movimentos que levem à regiões já visitadas do espaço de soluções, o que usualmente é alcançado pela proibição dos últimos movimentos realizados. Esses movimentos são armazenados nesta lista e permanecem proibidos (com status tabu), por um dado número de iterações.

Duas regras são normalmente utilizadas de forma a interromper o procedimento. Pela primeira, pára-se quando é atingido certo número máximo de iterações sem melhora no valor da melhor solução. Pela segunda, quando o valor da melhor solução chega a um limite inferior conhecido (ou próximo dele).

Mais informações em Marinho (2005), Glover (1986) e Hansen (1986).

2.2.2.4. Algoritmo genético

Goldberg (1989) conceitua Algoritmos Genéticos como algoritmos de busca baseados nos mecanismos da seleção natural e da genética natural. Eles combinam a sobrevivência da estrutura mais adaptada com a troca aleatória das informações genéticas para formar um novo espaço de busca. Segundo Goldberg (1989) eles exploram eficientemente as informações históricas para especular novos pontos de busca com uma esperada melhoria de desempenho.

Segundo Souza et al. (2002), a geração de horários é considerada um problema de decisão multicritério, porque para determinar a qualidade de uma programação faz-se necessário considerar diferentes objetivos (custos). Com isso, a utilização do Algoritmo Genético traz uma vantagem considerável, uma vez que um conjunto de indivíduos candidatos à solução do problema pode representar em um mesmo instante diversas regiões do espaço de busca.

Por ter sido, dentre as meta-heurísticas existentes, o escolhido para implementação neste trabalho, o Algoritmo Genético é mais bem descrito na seção a seguir.

2.3. Algoritmos Genéticos (AG)

Concebidos por Holland (1975), os Algoritmos Genéticos são baseados na teoria de evolução natural descritos por Darwin (1859). De acordo com Srivinas et al. (1994), atualmente, os AG têm se constituído ferramentas poderosas para resolver problemas onde o espaço de busca é muito grande e os métodos convencionais se mostram ineficientes.

Segundo Ribeiro Filho (2000), Algoritmos Genéticos são baseados num processo coletivo de aprendizagem dentro de uma população de indivíduos, cada um dos quais representando um ponto no espaço de busca de soluções para um dado problema. A população é inicializada e evolui através de gerações com o uso de operadores de seleção, reprodução e mutação. Durante o algoritmo, é feita uma avaliação da qualidade (adaptação) dos indivíduos e essa informação é usada para conduzir o processo evolutivo, favorecendo a seleção de indivíduos bem adaptados, para assim gerar indivíduos também bem adaptados. O mecanismo de cruzamento permite mesclar informações de indivíduos de uma geração e passa-las aos seus descendentes, e um processo de mutação introduz inovação na população.

Embora AGs sejam, atualmente, uma classe de algoritmos, pois existem variações na sua forma, será feito uma explicação breve do funcionamento básico destes algoritmos.

Antes, a idéia básica do AG pode ser descrita de maneira sucinta, no pseudo-código a seguir (Oliveira, 2004):

Algoritmo Genético Clássico:

1. P ← Geração da População Inicial
2. **enquanto** condição não satisfeita **faça**
3. P’ ← Seleção (P);
4. P ← Cruzamentos (P’);
5. P ← Mutações (P);
6. **fim enquanto**
7. Solução ← Melhor Indivíduo (P)

Goldberg (1989), Ribeiro Filho (2000) e Srivinas et al. (1994) foram consultados para um melhor detalhamento do método.

O AG processa estruturas que são representações de soluções do problema. As soluções são consideradas indivíduos, e as estruturas são representações de seus cromossomos.

Uma função associa as estruturas a valores numéricos que medem a qualidade das soluções. A formulação dessa função reflete os objetivos de otimização do problema. A qualidade das soluções é considerada como adaptação dos indivíduos.

Iterativamente, como se pode observar no pseudo-código apresentado anteriormente, o AG transforma a população em uma nova população, usando três operações genéticas definidas com base nos princípios fundamentais da evolução natural: seleção, reprodução (ou cruzamento) e mutação.

A seleção pode ser feita de inúmeras maneiras, mas normalmente as estruturas são selecionadas de modo proporcional à sua adaptação. Estruturas mais adaptadas são selecionadas com mais freqüência do que as menos adaptadas.

As estruturas selecionadas podem ser então reproduzidas de várias maneiras, dentre elas, o cruzamento, que é a associação entre duas estruturas para a geração de novos indivíduos.

Um processo de mutação sobre os novos indivíduos altera a formação de suas estruturas. O objetivo da mutação é manter variedade na nova população.

Tanto a escolha da forma de reprodução quanto da aplicação ou não da mutação são feitas de acordo com probabilidades que são parâmetros do AG.

Por fim, existe uma condição de parada do algoritmo, que pode ser baseado no número de iterações ou o tempo de execução.

2.4. Problema Básico de Geração de Horários

2.4.1. Formulação Matemática do Problema

Segundo Souza (2000), sejam os seguintes dados:

(1) Um conjunto finito *H* de horários semanais, por exemplo, das 10:00h até às 11:00h;

(2) Um conjunto de *m* professores, aos quais estão associados conjuntos $T_i \subseteq H$, representativos dos horários durante os quais o professor *i* está disponível para ensino;

(3) Um conjunto de *n* turmas e um conjunto $C_j \subseteq H$ onde C_j é o conjunto de horários durante os quais a turma *j* está disponível;

(4) Uma matriz R = $(r_{ij})_{mxn}$ de números inteiros não-negativos, os quais representam o número de aulas que cada professor *i* tem que ministrar para cada turma *j*;

Definição: *O problema básico de geração de horários em escola consiste em determinar a existência de uma função encontro, $f(i,j,k) : \{1,...,m\} \times \{1,...,n\} \times H \rightarrow \{0,1\}$, (onde $f(i,j,k) = 1$ se e somente se o professor i ensina para a turma j no horário k), satisfazendo:*

(a) $f(i, j, k) = 1 \Rightarrow k \in (T_i \cap C_j)$ *;*

(b) $\sum_{k \in H} f(i, j, k) = r_{i,j} \;\; \forall_i = 1,...,m \; e \; \forall_j = 1,...,n$ *;*

(c) $\sum_{i=1}^{m} f(i, j, k) \leq 1 \; \forall_j = 1,...,n \; e \; k \in H$ *;*

(d) $\sum_{j=1}^{n} f(i, j, k) \leq 1 \; \forall_i = 1,...,m \; e \; k = H$

A restrição (a) assegura que existe um encontro entre um professor *i* e uma turma *j* somente se ambos estiverem disponíveis no horário *k*. As restrições (b) garantem que o número total de encontros durante a semana entre o professor *i* e a turma *j* é exatamente o número requerido $r_{i,j}$. As restrições (c) impedem que alguma turma tenha mais do que um professor em um dado horário e o conjunto de restrições (d), por sua vez, assegura que nenhum professor ensina para mais de uma turma simultaneamente.

Pode-se considerar um grau de importância para as restrições. As restrições mais "fortes" são aquelas que garantem a viabilidade de uma solução, isto é, aquelas referentes aos conflitos de simultaneidade de aulas tanto para professores quanto para turmas.

Além dessas restrições consideradas "fortes", podem ser consideradas outras restrições mais "fracas" cuja não-satisfação não acarreta necessariamente inviabilidade da solução.

Assim sendo, uma solução *s* pode ser avaliada com base em duas componentes, uma de inviabilidade (*g(s)*), a qual mede o não atendimento aos requisitos essenciais (restrições "fortes"), e outra de qualidade (*h(s)*), a qual mede o não atendimento aos requisitos não-essenciais (restrições "fracas"). A função de custo de uma solução *s*, a qual deve ser minimizada, pode ser calculada, portanto, pela seguinte expressão: *f(s) = g(s) + h(s)*.

A componente *g(s)*, que mensura o nível de inviabilidade de uma solução *s*, é avaliada com base na expressão: $g(s) = \sum_{k=1}^{K} \alpha_k I_k$

Onde *K* é o número de medidas de inviabilidade, I_k é o valor da *k*-ésima medida de inviabilidade e α_k é o peso associado à essa *k*-ésima medida. Isto é necessário, pois há mais de uma medida de inviabilidade para o problema de geração de horários. No presente trabalho, referem-se à média de aulas por dia, número de aulas de cada disciplina por dia, choques no horário do professor e aos horários inviáveis. Estas restrições são apresentadas em detalhes na seção a seguir.

A componente *h(s)*, que mensura a qualidade de uma solução *s*, é avaliada com base na seguinte função: $$h(s) = \sum_{l=1}^{L} \beta_l Q_l$$

Onde *L* representa o número de medidas de qualidade, Q_l o valor da *l*-ésima medida de qualidade e β_l o peso associado a essa *l*-ésima medida. Assim como mencionado anteriormente, o contador *L* é necessário para indicar que existe mais de uma medida de qualidade no horário. Estas medidas são apresentadas em detalhes na seção a seguir.

Deve ser observado que uma solução *s* é viável se e somente se *g(s)* = 0. Nas componentes da função *f(s)* os pesos dados às diversas medidas refletem a importância relativa de cada uma delas e, sendo assim, deve-se tomar $\alpha_k >> \beta_l \quad \forall k, l$, de forma a privilegiar a eliminação das soluções inviáveis.

2.4.2. Restrições em horários reais

As componentes de inviabilidade e qualidade, mencionadas no modelo, são classificadas como restrições nos horários em problemas reais. Uma listagem com as restrições mais comuns encontradas em escolas secundárias e/ou abordadas em trabalhos clássicos da literatura é apresentada a seguir. As oito primeiras são consideradas neste trabalho. Optou-se pela escolha destas oito restrições por terem sido consideradas as mais importantes e as que mais refletem a realidade da maioria das escolas brasileiras.

1. Colisão por Professor: Essa restrição avalia se um professor lecionará mais de uma aula simultaneamente. É considerada como uma restrição “forte”. Na Figura 2.1 pode-se observar que a professora “Erotildes” deve dar aula tanto para o “2B” quanto para o “3A” no primeiro horário de segunda-feira, logo esta formulação torna este horário inviável.

Turma: 2B
SEGUNDA
MATEMÁTICA/Erotildes
FÍSICA /Toninho
BIOLOGIA /Dayse
QUÍMICA /Ana Paula
PORTUGUÊS /Lurdinha
Turma: 3A
SEGUNDA
MATEMÁTICA/Erotildes
PORTUGUÊS /Goreti
QUÍMICA /Ana Paula
BIOLOGIA /Dayse
HISTÓRIA /Marlene

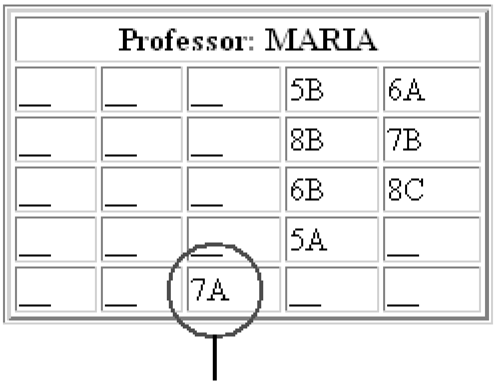
Figura 2.1 - Colisão por Professor

2. Janelas: Essa restrição verifica se a grade horária formada possui alguma “janela”, ou seja, se existem horários vagos em um determinado dia no horário do professor. Na Figura 2.2 pode-se notar a presença de duas janelas no horário da professora “Tanismara”, uma na terça-feira (3º horário) e outra na quarta-feira (2º horário).

Professor: TANISMARA				
__	__	8C	8C	__
__	6B	__	6B	__
__	__	6B	8C	__
8C	8C	__	__	__
6B	6B	__	__	__

Figura 2.2 - Janela no horário do Professor

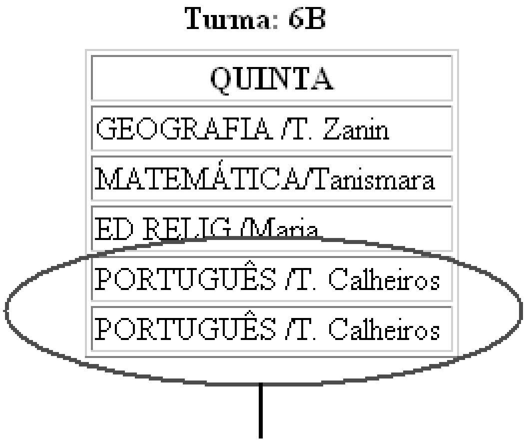
3. Aulas isoladas: Aulas presentes isoladamente na grade horária do professor. Por exemplo, na Figura 2.3 é mostrada a ocorrência de uma aula isolada no último horário de quarta-feira para a professora “Maria”.



Professor: MARIA				
__	__	__	5B	6A
__	__	__	8B	7B
__	__	__	6B	8C
__	__	__	5A	__
__	__	7A	__	__

Figura 2.3 - Aula isolada no horário do Professor

4. Blocos de Disciplinas (Aulas Geminadas): Esse critério avalia uma condição pedagógica baseada na forma pelas quais as disciplinas estão dispostas. Para um melhor aprendizado, convém que as disciplinas estejam dispostas em blocos de duas. Se a carga horária de uma disciplina são cinco aulas semanais, o ideal é que elas estejam dispostas em dois blocos de duas aulas e uma aula isolada. Entretanto esta restrição não é unanimidade entre as escolas. A Figura 2.4 exibe a ocorrência de aulas geminadas.



Turma: 6B

QUINTA
GEOGRAFIA /T. Zanin
MATEMÁTICA/Tanismara
ED RELIG /Maria
PORTUGUÊS /T. Calheiros
PORTUGUÊS /T. Calheiros

Figura 2.4 - Aulas Geminadas no horário do Professor

5. Vários Blocos por Dia: Um problema pode ocorrer durante a formação dos blocos. Trata-se do fato de dois blocos de uma mesma disciplina ficarem alocados em um mesmo dia, constituindo assim uma atitude pedagogicamente incorreta. É preciso evitar, por exemplo, quatro aulas de matemática em um determinado dia, pois os alunos e professores ficariam insatisfeitos com o horário gerado.

6. Preferências dos professores: Os professores geralmente trabalham em mais de uma escola e têm suas preferências de horário. Por isso, além de avaliar todos os critérios anteriores é preciso verificar se o professor está lecionando num horário de sua preferência. O presente trabalho considera os horários inviáveis (o professor não pode dar aula neste horário, considerada no trabalho como restrição “forte”) e os horários indesejáveis (horário que o professor prefere não dar sua aula, considerada como restrição “fraca”).

7. Indisponibilidade da turma: Uma aula não pode ser alocada a um horário a qual a turma não esteja disponível. Na Figura 2.5 pode-se observar que a turma “5A - Ensino Supletivo” não têm aulas no primeiro e nem no último horário de cada dia.

Turma: 5A - Ensino Supletivo				
SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
ARTES /Israel	HISTÓRIA /Marlene	HISTÓRIA /Marlene	ED. FÍSICA/Dayse	MATEMÁTICA/Marly
ED. FÍSICA/Dayse	MATEMÁTICA/Marly	PORTUGUÊS /Elisa	MATEMÁTICA/Marly	PORTUGUÊS /Elisa
PORTUGUÊS /Elisa	PORTUGUÊS /Elisa	MATEMÁTICA/Marly	GEOGRAFIA /Fabiola	GEOGRAFIA /Fabiola

Figura 2.5 - Indisponibilidade da turma

8. Média de aulas por dia: Esse critério avalia se o número de horários por dia está de acordo com a média de aulas. Essa média é calculada baseada na carga horária semanal de uma determinada turma. Se uma turma tem vinte e cinco aulas semanais, o ideal é que ela tenha cinco aulas por dia. A avaliação desse critério evita que uma turma tenha muitos horários em alguns dias e poucos horários em outros. Esta restrição foi considerada “forte” neste trabalho.

9. Aulas preferenciais: A configuração de aulas que são preferíveis nos primeiros horários ou nos últimos. Novamente baseado no aspecto pedagógico, algumas escolas preferem que as aulas consideradas mais difíceis, como Física ou Matemática sejam alocadas no início do horário.

10. Prestígio do professor: Por exemplo, um professor mais renomado ou um que trabalha a mais tempo na escola tem mais prestígio que um outro que seja menos renomado ou que tenha menos tempo de casa. Isto é, ele terá preferência na alocação de suas aulas.

11. Restrição de precedência: O quadro de horário de uma turma não pode conter algumas seqüências específicas de matéria. Na prática, evitam-se aulas “difíceis” em seqüência.

12. Distribuição de aulas: Aulas da mesma matéria precisam ser espalhadas tão uniformemente quanto possível ao longo da semana.

13. Pré-alocação de aulas: Algumas aulas com seus horários pré-alocados;

Souza (2000) afirma que, na prática, uma tabela de horário satisfazendo a todos esses requisitos pode não existir. As restrições (2 + 3 e 12) são contraditórias. Entretanto, faz-se necessário encontrar um compromisso conveniente entre professor e turma para gerar uma tabela de horária tão boa quanto possível, mesmo na presença de alguns conflitos.

2.5. Trabalhos Correlatos

Segundo Even et al. (1976), o problema de geração de horários é NP-Difícil e, em vista disso, vários métodos heurísticos têm sido experimentados.

Recentemente, técnicas baseadas em meta-heurísticas têm sido empregadas. São encontrados trabalhos com Recozimento Simulado em Abramson (1991) e Abramson (1999), Busca Tabu em Hertz (1992), Costa (1994), Schaerf (1996) e Schaerf (1999) e Algoritmos Genéticos em Colorni et al. (1998) e Abramson & Abela (1992) e combinação de métodos diferentes em Cooper & Kingston (1993).

Abramson (1991) aplicou Recozimento Simulado para resolver um problema de horários escolares na qual se considera, também, a alocação de salas. A função objetivo a ser minimizada é a soma do número de conflitos, em cada horário, de turmas, professores e salas, isto é, a soma do número de vezes, menos um, que cada turma, professor ou sala aparece em cada horário, se esse número for maior que zero.

Abramson (1999) compararam seis esquemas diferentes de resfriamento em um algoritmo de Recozimento Simulado para resolver problemas de horários em escolas. Um desses esquemas, que produziu soluções de melhor qualidade em um menor espaço de tempo, consiste em uma seqüência de reaquecimento, seguido de resfriamento, quando a temperatura atinge um determinado nível, o qual é determinado pelo próprio método.

Hertz (1992) aplicou Busca Tabu para resolver um problema onde as matérias são dividas em tópicos, os quais devem ser ministrados em certa ordem para uma dada turma ao longo de um conjunto de dias.

Costa (1994) e Schaerf (1996) aplicaram Busca Tabu para resolver um problema de horários em escolas (PHE) em que aparecem restrições de vários tipos. Schaerf (1996) representa uma solução por uma matriz, onde suas linhas representam os professores e as colunas, os horários reservados para as aulas. Cada elemento da matriz, referente a um dado professor e a um dado horário, contém o nome da turma para a qual o professor vai lecionar naquele horário. Um movimento consiste em trocar duas aulas de um dado professor ou mover uma aula para um horário diferente. Costa (1994), ao contrário, empregou um tipo de movimento diferente. Mais precisamente, ele permite somente a mudança de uma única aula para um horário diferente. Assim, nessa representação, um professor pode ensinar mais que uma aula ao mesmo tempo. A função objetivo nessas duas abordagens é uma soma de componentes que avaliam o atendimento aos diversos requisitos exigidos para o quadro de horário.

Colorni et al. (1998) aplicaram Algoritmos Genéticos ao PHE. Quadros inviáveis de horário também são incluídos no espaço de busca do algoritmo. A função objetivo considera requisitos didáticos, organizacionais e pessoais e incorpora, também, o número de inviabilidades. De forma a conduzir a busca em direção a quadros viáveis de horário, a função objetivo é hierarquizada, atribuindo-se às inviabilidades um peso maior do que o dos demais requisitos.

Em trabalhos anteriores considerando a realidade no Brasil, cabe citar Souza et al. (2002), que utiliza um algoritmo híbrido que inclui: GRASP, Busca Tabu e Algoritmo Genético. A inclusão das meta-heurísticas GRASP e Busca Tabu, visam, respectivamente, gerar uma população inicial de forma heurística (não aleatória) e o refinamento da melhor solução encontrada em cada geração do algoritmo genético. Os resultados estão limitados a uma comparação entre um método de geração com o refinamento final pela Busca Tabu e aqueles gerados com o mesmo algoritmo, porém sem o refinamento final.

Já em Timóteo (2002) é utilizado um Algoritmo Genético com formas adicionais de mutação, seleção e de cruzamento. Não considerando os requisitos de formação de janelas e aulas isoladas, o trabalho alcançou resultados interessantes, inclusive apresentando uma comparação entre várias alternativas de configuração possíveis.

2.6. Considerações Finais do Capítulo

Neste capítulo foi descrito de forma detalhada o problema a ser tratado no trabalho. Além disso, foram apresentados métodos heurísticos que são utilizados na literatura para a resolução do problema.

Foi observado, também, como os princípios da evolução natural foram utilizados na criação dos Algoritmos Genéticos para a obtenção de soluções de problemas de otimização combinatória.

Finalmente foi feito um estudo resumido dos trabalhos correlatos de maior importância na literatura relacionados ao tema deste trabalho.

3. METODOLOGIA

3.1. Modelo Tratado

Na seção 2.4, foi apresentado o modelo básico para o problema de geração de horários. Contudo, este modelo não reflete todas as restrições tratadas neste trabalho. Por isso, fez-se necessário uma adaptação ao modelo:

Sejam os seguintes dados:

(1) Um conjunto finito *H* de horários semanais, por exemplo, das 10:00h até às 11:00h;

(2) Um conjunto de *m* professores, aos quais estão associados conjuntos $T_i \subseteq H$, representativos dos horários durante os quais o professor *i* está disponível para ensino;

(3) Um conjunto de n turmas e um conjunto $C_j \subseteq H$ onde C_j é o conjunto de horários durante os quais a turma *j* está disponível;

(4) Uma matriz R = $(r_{ij})_{mxn}$ de números inteiros não-negativos, os quais representam o número de aulas que cada professor *i* tem que ministrar para cada turma *j*;

Função Objetivo:

Minimize $\sum \chi K + \sum \delta L + \sum \varepsilon M + \sum \varphi N + \sum \gamma O$ **sujeito a:**

(a) $f(i,j,k) = 1 \Rightarrow k \in (T_i \cap C_j)$ *;*

(b) $\sum_{k \in H} f(i,j,k) = r_{i,j} \ \forall_i = 1,\ldots,m \ e \ \forall_j = 1,\ldots,n$ *;*

(c) $\sum_{i=1}^{m} f(i,j,k) \leq 1 \ \forall_j = 1,\ldots,n \ e \ k \in H$ *;*

(d) $\sum_{j=1}^{n} f(i,j,k) \leq 1 \ \forall_i = 1,\ldots,m \ e \ k = H$

A restrição (a) assegura que existe um encontro entre um professor *i* e uma turma *j* somente se ambos estiverem disponíveis no horário *k*. As restrições (b) garantem que o número total de encontros durante a semana entre o professor *i* e a turma *j* é exatamente o número requerido $r_{i,j}$. As restrições (c) impedem que alguma turma tenha mais do que um professor em um dado horário e o conjunto de restrições (d), por sua vez, assegura que nenhum professor ensina para mais de uma turma simultaneamente.

Sendo:

K, L, M, N e O a quantidade de cada uma das medidas de qualidade;

χ, peso da penalidade: janelas no horário;

δ, peso da penalidade: aulas isoladas;

ε, peso da penalidade: blocos de disciplinas;

φ, peso da penalidade: monopólio de aulas por dia;

γ, peso da penalidade: horários indesejáveis.

3.2. Proposta de Solução

Dentre as meta-heurísticas citadas, o Algoritmo Genético foi a escolha devido a facilidade de abordagem do problema considerando a eliminação de janelas e aulas isoladas e também por estar entre as meta-heurísticas de destaque em problemas de otimização combinatória. Além disso, segundo Eiben & Smith (2003), AGs são o grupo de algoritmos evolucionários mais conhecido e estudado, o que contribui para sua utilização e aprendizagem.

Outro ponto importante a ser citado é a série de trabalhos correlatos que obtiveram relativo sucesso na utilização do AG como alternativa de solução para o problema de geração de horários.

A fim de adaptar o algoritmo genético para a resolução do problema abordado, foram necessárias definições e modificações. As seções a seguir descrevem estas alterações.

3.2.1. Função de Adaptabilidade

A partir do modelo proposto, sentiu-se a necessidade de se construir a função de adaptabilidade atendendo as exigências do Algoritmo Genético. Para tanto, as restrições (a) e (b) não foram incluídas como penalidades da função objetivo, pois estas são automaticamente atendidas devido à representação matricial dos indivíduos (será mais bem explicado nas seções 3.2.2 e 3.2.3). Já as restrições (c) e (d) foram convertidas para a penalidade I, cujo peso é α. A restrição de horários inviáveis (restrição “forte”) foi convertida para a penalidade J, cujo peso é β. Embora estas duas restrições sejam “fortes” foi necessário transformá-las em penalidades na função de adaptabilidade, pois, com a utilização das ponderações, o algoritmo, além de permitir a existência de soluções inviáveis nos horários iniciais, consideraria estas restrições na resolução do problema. Estes pesos possuem valores bem superiores aos outros, assim, o algoritmo busca minimizar primeiramente estas restrições.

Já as restrições “fracas” que são focadas neste trabalho (janelas e aulas isoladas) são tratadas pelas penalidades K e L cujos pesos são χ e δ e possuem pesos intermediários.

Os valores de cada um destes pesos e dos outros não citados serão mencionados no capítulo 4.

A seguir, a Função de Adaptabilidade proposta:

$$\textbf{\textit{Adaptabilidade}}_{\textbf{\textit{i}}} = \sum \alpha I + \sum \beta J + \sum \chi K + \sum \delta L + \sum \varepsilon M + \sum \varphi N + \sum \gamma O$$

Sendo:

I, J, K, L, M, N e O a quantidade de cada uma das medidas avaliadas pelo algoritmo;

α, peso da penalidade: choques entre professores;

β, peso da penalidade: horários inviáveis;

χ, peso da penalidade: janelas no horário;

δ, peso da penalidade: aulas isoladas;

ε, peso da penalidade: blocos de disciplinas;

φ, peso da penalidade: monopólio de aulas por dia;

γ, peso da penalidade: horários indesejáveis.

3.2.2. Representação de Soluções

Ao se implementar um AG, a primeira necessidade é criar uma representação eficiente para as soluções. A representação pode significar o bom ou mau desempenho do algoritmo.

Para representar a solução no problema da geração de horários, vários trabalhos foram avaliados, porém a forma mais intuitiva foi citada por Timóteo (2002), onde cada gene representa um *slot* (ex: terça-feira às 07h00min). Um cromossomo seria representado por uma matriz tri-dimensional (Número de Turmas X Número de Dias X Número de

Horários) e em cada célula da matriz seria armazenado um número inteiro que representaria a relação entre um professor e sua respectiva disciplina. A representação proposta tem a vantagem de não permitir, devido a sua codificação, que duas disciplinas referentes à mesma turma sejam lecionadas no mesmo horário. Porém, não garante a não ocorrência de colisões de professores e outras restrições.

Cada indivíduo é formado por uma matriz de três dimensões conforme mostrado na Figura 3.1. Segundo Timóteo (2002), é importante observar que dependendo do número de turmas e *slots* disponíveis, a representação de uma solução pode ser muito grande, requisitando, assim, maiores recursos computacionais.

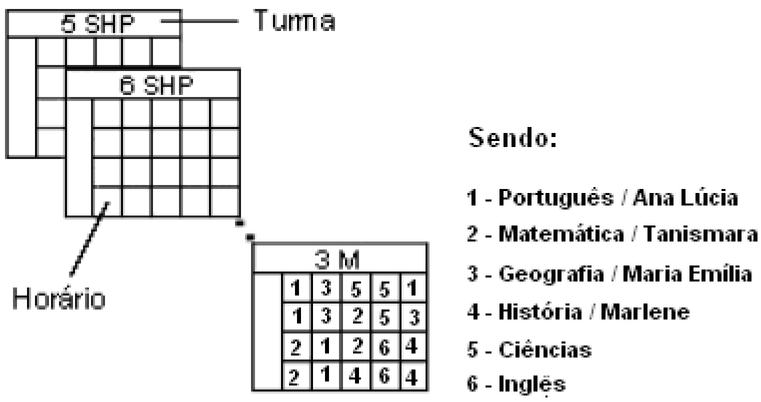


Figura 3.1 - Representação da Solução

Na verdade, um AG trabalha com uma população de indivíduos, e por isso a representação desta população seria, um conjunto de matrizes.

3.2.3. População Inicial

Inicia-se o AG com a geração da população inicial. Neste trabalho, optou-se pela geração de uma população inicial aleatória. Para cada turma, existe uma lista com as disciplinas que devem ser ministradas. As disciplinas dessa lista são sorteadas e atribuídas em cada *slot* de cada indivíduo. O processo continua até que a lista de disciplinas de todas as turmas seja percorrida. Segundo Goldberg (1989) o tamanho ideal para a população é entre 50 e 100 indivíduos. Neste trabalho, inicialmente, optou-se por uma população de 100 indivíduos.

Apesar de a população inicial ser gerada de maneira aleatória, a lista de disciplinas de cada turma e a representação da solução proposta (vide seção 3.2.2) impedem a geração de indivíduos totalmente inviáveis. Dentre as quatro restrições de inviabilidade consideradas neste trabalho (vide seções 2.4.1 e 2.4.2), duas delas (média de aulas por dia e número de aulas de cada disciplina) são tratadas já na geração da população inicial.

3.2.4. Avaliação da População

Segundo Mitchell (1996), na avaliação da população cada indivíduo da população sofre um processo de avaliação, com o objetivo de retornar seu *fitness* (grau de adaptação), ou seja, determinando o quão apto ele está para o meio em relação à população a que pertence.

Para a avaliação de indivíduos adotou-se nesse trabalho a abordagem baseada em penalidades. Para cada tipo de penalidade atribui-se um peso que influenciará negativamente na seleção do indivíduo para a reprodução, conforme função de adaptabilidade da seção 3.2.1.

As seguintes restrições de inviabilidade e de qualidade são avaliadas pelo Algoritmo Genético proposto:

- Choque de professores (restrição “forte”);
- Horário inviável (restrição “forte”);
- Janelas no horário dos professores (restrição “fraca”);
- Aulas isoladas (restrição “fraca”);
- Horário indesejável (restrição “fraca”);
- Aulas geminadas (restrição “fraca”);
- Ocorrência de Vários blocos (restrição “fraca”);
- Monopólio de aulas (restrição “fraca”); e
- Média de aulas por dia (restrição “forte”)

Cada uma das restrições consideradas possui um peso diferente. Como visto na função de adaptabilidade da seção 3.2.1, as restrições “fortes” possuem um peso elevado, a fim de serem minimizadas primeiro. As restrições focadas no trabalho possuem um peso intermediário. Já as outras restrições possuem um peso menor, e por isso, são as últimas a serem minimizadas pelo algoritmo.

Quando um indivíduo não possui nenhuma penalidade, seu *fitness* é retornado com o valor um, sendo este o valor conceitual para o limite superior neste problema.

3.2.5. Seleção dos Indivíduos Reprodutores

De acordo com Goldberg (1989) na etapa de Seleção, os indivíduos são selecionados para o cruzamento. Neste ponto, com base no grau de adaptação de cada um, efetua-se um processo onde os mais aptos possuirão uma maior probabilidade de se reproduzirem. Este é um passo importante de um AG, pois uma boa escolha de indivíduos é aquela que mescla indivíduos bons com indivíduos ruins.

Para o processo de seleção foi utilizado o método torneio. Antes da seleção, os indivíduos são avaliados separadamente e o resultado é a quantificação de seu grau de aptidão. Na implementação do torneio, formam-se aleatoriamente pares que realizam uma disputa entre si, sendo que o indivíduo mais apto é o vencedor do torneio e está selecionado para a fase de reprodução. No final deste processo, metade dos indivíduos está selecionada para esta fase. A metade não selecionada é descartada, isto é, ela não participará da formação da população seguinte.

3.2.6. Cruzamento

Goldberg (1989) define cruzamento, ou *crossover*, como a operação que realiza a troca do material genético entre os cromossomos, ou seja, aquela que irá combinar os genes dos indivíduos selecionados na etapa anterior para gerar novos indivíduos.

Optou-se neste trabalho o cruzamento com corte em um ponto, isto é, o cruzamento entre os horários selecionados no processo de seleção se dá a partir do sorteio de um ponto aleatório dentro da grade de horários. A partir deste *slot* sorteado, o material genético (disciplinas) dos indivíduos são trocados. Porém é preciso ressaltar que para o problema de geração de horários uma regra deve ser respeitada: o número de aulas que cada disciplina possui na turma deve se manter constante. Por exemplo, se uma turma possui cinco aulas de Português, após o cruzamento, este número deve ser mantido. Isto é, o cruzamento deve garantir que esta restrição “forte” não será violada. Para tanto foi proposta uma solução que somente troca as disciplinas quando há correspondência entre o primeiro e o segundo indivíduo. Para casos que não há correspondência, as disciplinas são mantidas em suas posições originais. Dessa forma, cada par de indivíduos geram dois novos descendentes.

Outra restrição “forte” que o cruzamento proposto não viola é a que trata da média de aulas por dia. Isto é garantido pela representação do indivíduo, definida na seção 3.2.2.

Entretanto, o cruzamento por si só não garante a eliminação completa das outras duas inviabilidades consideradas no trabalho: choque de professores e horários inviáveis. Ou seja, um cruzamento pode gerar indivíduos com choques e com horários inviáveis, contudo, estas inviabilidades são tratadas como penalidades na função de adaptabilidade.

Para realizar o cruzamento, dois indivíduos são escolhidos dentre aqueles selecionados para reprodução. Contudo, somente uma parcela dos pares é efetivamente cruzada. Neste trabalho, inicialmente, esta parcela foi definida como 60% dos indivíduos. Conseqüentemente, parte dos indivíduos, são mantidos, embora possam ser modificados na etapa seguinte de mutação.

3.2.7. Mutação

Depois que a etapa de cruzamento é realizada, pode acontecer a mutação. A mutação tem a intenção de prevenir que todas as soluções do problema sejam puramente combinações das soluções da população inicial. A operação de mutação muda aleatoriamente a descendência criada pelo cruzamento.

A mutação é realizada de forma totalmente aleatória e simples. Duas disciplinas dentro de uma mesma turma são sorteadas aleatoriamente e seus horários são trocados. Como o número de possibilidades de inversões de disciplinas (300, para 12 turmas x 25 aulas semanais, por exemplo) é muito elevado, este trabalho optou por realizar duas mutações em cada geração. Isto quer dizer que a cada iteração do algoritmo duas mutações são realizadas em toda a população.

Quanto às inviabilidades, a mutação comporta-se da mesma maneira do cruzamento. Isto é, mantém as restrições média de aulas por dia e número de aulas de cada disciplina invioláveis e não considera as outras duas restrições, podendo gerar indivíduos com choques e/ou com horários inviáveis.

A Figura 3.2 mostra como é realizada a mutação neste trabalho. No quadro de horários A é possível observar que duas disciplinas foram escolhidas (aleatoriamente). Já no quadro B, verificamos o mesmo quadro de horário após a mutação, isto é, as suas disciplinas foram invertidas de posições.

A

Turma: 3A				
SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MATEMÁTICA/Erotildes	MATEMÁTICA/Erotildes	PORTUGUÊS /Goreti	MATEMÁTICA/Erotildes	FÍSICA /Toninho
PORTUGUÊS /Goreti	BIOLOGIA /Dayse	GEOGRAFIA /Maria Emilia	PORTUGUÊS /Goreti	ED. FÍSICA/Nara
QUÍMICA /Ana Paula	PORTUGUÊS /Goreti	BIOLOGIA /Dayse	QUÍMICA /Ana Paula	BIOLOGIA /Dayse
INGLÊS /Efigênia	FÍSICA /Toninho	ED. FÍSICA/Nara	HISTÓRIA /Marlene	MATEMÁTICA/Erotildes
HISTÓRIA /Marlene	GEOGRAFIA /Maria Emilia	INGLÊS /Efigênia	ARTES /Israel	QUÍMICA /Ana Paula

B

Turma: 3A				
SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MATEMÁTICA/Erotildes	MATEMÁTICA/Erotildes	PORTUGUÊS /Goreti	MATEMÁTICA/Erotildes	FÍSICA /Toninho
PORTUGUÊS /Goreti	BIOLOGIA /Dayse	GEOGRAFIA /Maria Emilia	PORTUGUÊS /Goreti	ED. FÍSICA/Nara
QUÍMICA /Ana Paula	PORTUGUÊS /Goreti	HISTÓRIA /Marlene	QUÍMICA /Ana Paula	BIOLOGIA /Dayse
INGLÊS /Efigênia	FÍSICA /Toninho	ED. FÍSICA/Nara	HISTÓRIA /Marlene	MATEMÁTICA/Erotildes
BIOLOGIA /Dayse	GEOGRAFIA /Maria Emilia	INGLÊS /Efigênia	ARTES /Israel	QUÍMICA /Ana Paula

Figura 3.2 - Mutação

3.2.8. Seleção dos Sobreviventes

Antes da realização do cruzamento e da mutação, o algoritmo implementado utiliza-se do elitismo para garantir que os dois melhores indivíduos sobrevivam na população seguinte.

Como forma de seleção dos sobreviventes, optou-se que os indivíduos descendentes (filhos) sempre substituirão os indivíduos ancestrais (pais). Esta maneira é uma tentativa de se evitar que as novas populações sejam formadas apenas por indivíduos cuja média de fitness sejam superiores a seus ancestrais. Eiben & Smith (2003) afirma que a diversidade da população é medida pelo número de indivíduos diferentes, os quais devem incluir indivíduos ruins, médios e bons, pois isto expande o espaço de busca do algoritmo.

3.2.9. Condições de Parada

São as condições que determinam o fim do processo iterativo. Existem algumas maneiras de finalizar o processamento de um AG. Ribeiro Filho (2000) cita:

- Tempo;
- Número de Gerações;
- Convergência (95 % dos genes iguais entre os indivíduos).

Dentre as três condições de parada apresentadas, este trabalho faz uso do número de gerações para encerrar a execução do algoritmo. Inicialmente, foi escolhido o limite de 10.000 gerações. Contudo, após diversas execuções, foi notado que o número de 15.000

gerações oferecia maiores possibilidades de se encontrar horários de melhor qualidade e ainda dentro de um intervalo aceitável de tempo.

3.3. O Sistema Desenvolvido

Após o entendimento de todas as questões que envolvem o problema de geração de horários e o funcionamento do AG, o passo seguinte foi a modelagem e implementação das necessidades específicas do algoritmo genético para o tratamento do problema citado.

Além da implementação do algoritmo em si, com todos os seus operadores e parâmetros, foi desenvolvida também uma aplicação que utiliza este algoritmo e apresenta as demais funcionalidades, tais como: interface gráfica, acesso a banco de dados, entre outras.

Esta seção abordará esta aplicação desenvolvida, bem como seus dados de entrada, informações de saída e modos de operação.

A Figura 3.3 exibe a tela principal do sistema.

Figura 3.3 - Tela Principal do Sistema

3.3.1. Linguagens e Ferramentas utilizadas

O ambiente de desenvolvimento da aplicação é bastante diversificado. Tanto o algoritmo em si quanto a aplicação foram desenvolvidos utilizando-se a linguagem de programação JAVA™. A aplicação utiliza o Sistema Gerenciador de Banco de Dados (SGBD) livre MySQL para armazenar os dados de entrada e os parâmetros de execução do algoritmo. Os dados de saída, isto é, o horário gerado é exibido em linguagem HTML.

O sistema, inicialmente chamado de AGGH 1.0, possui uma interface gráfica a fim de facilitar a utilização do usuário. Foi utilizado o ambiente de desenvolvimento NetBeans 5.0 e o editor Java Gel.

A seguir é feita uma breve descrição de cada uma das tecnologias utilizadas.

3.3.1.1. Linguagem de Programação

Para implementação do algoritmo foi utilizada uma linguagem muito robusta e popular, a linguagem de programação Java.

Segundo Deitel et al. (2001), Java é uma poderosa linguagem de programação de computadores que permite a construção de poderosos sistemas de informação. A linguagem Java foi escolhida por apresentar várias qualidades dentre as quais se destacam:

- Linguagem robusta;
- Java é gratuita;
- Java é uma linguagem multi-plataforma, isto é, pode ser usada tanto em Windows, quanto Linux e outros sistemas operacionais;
- Java é de fácil aprendizado;
- Facilidade de construção de interfaces gráficas amigáveis;
- É uma linguagem de fácil integração, existe uma facilidade de comunicação com outros programas e protocolos, um exemplo disso é a fácil e bem realizada integração com o Servidor de Banco de Dados MySQL;
- Java é popular e já é a escolha para implementação de aplicativos baseados em Intranet, Internet e qualquer outro dispositivo que se comunique em uma rede (como telefones celulares, pagers, etc).

Mais informações em Deitel et al. (2001) e site do Java ([http://java.sun.com]).

3.3.1.2. Sistema Gerenciador de Banco de Dados

A utilização de um SGBD é necessária ao desenvolvimento do sistema, pois os dados de entrada (turmas, aulas, professores, preferências e restrições) serão armazenados em banco de dados.

O MySQL, um dos SGBD mais utilizados atualmente, tanto na área acadêmica quanto na área comercial, foi o escolhido devido a uma série de fatores. A versão utilizada foi o MySQL 3.23. Este software está disponível em [http://www.mysql.com].

Algumas características pelas quais foi escolhido esse SGBD:

- Execução muito rápida de comandos, provavelmente o mais rápido do mercado;
- Sistema de segurança simples e funcional;
- É gratuito para o público em geral.

Alem disso, o MySQL tem a capacidade de:

- Lidar com um número ilimitado de usuários;
- Manipular mais de cinqüenta milhões (50.000.000) de registros;

3.3.1.3. Ambiente de Desenvolvimento

Para o desenvolvimento do algoritmo, e posteriormente da aplicação foram utilizados os ambientes de desenvolvimento NetBeans 5.0 ([http://www.netbeans.org/]) e Gel ([http://www.gexperts.com/]). O primeiro tem ótimo suporte para a confecção de interfaces gráficas robustas que muito facilitam a operação do usuário. Já o Gel foi utilizado por ser mais leve e esteve mais presente durante a construção do algoritmo em si.

Ambos foram utilizados por serem produtos livres, populares no mercado e de fácil aprendizado. Para maiores informações, basta consultar os respectivos websites.

3.3.2. Modelagem do Sistema

Por se tratar de um sistema complexo, antes de iniciar a implementação do sistema fez-se necessário a sua modelagem. Segundo Elmasri & Navathe (2005) as metodologias de modelagem de dados de objetos como UML (Universal Modeling Language – Linguagem de Modelagem Universal) estão se tornando cada vez mais populares no projeto e engenharia de software. Essas metodologias vão além do projeto de um banco de dados, especificando o projeto detalhado dos módulos de software e suas interações, utilizando vários tipos de diagramas.

Foram construídos os seguintes diagramas:

- Diagramas de Caso de Uso
- Diagrama de Classes
- Diagrama de Seqüência
- Diagrama de Atividades
- Diagrama de Estados

Os principais deles são descritos nas subseções a seguir.

3.3.2.1. Diagrama de Caso de Uso: Cadastros e Lançamentos

Segundo Booch et al. (2001) a modelagem de um diagrama de caso de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema. Para sistemas que possuem um número elevado de funcionalidades, a construção destes diagramas visa facilitar o entendimento do problema, a documentação do que será desenvolvido bem como facilita o próprio desenvolvimento.

A Figura 3.4 descreve todas as funcionalidades que o usuário de perfil "Administrador" poderá executar na ferramenta.

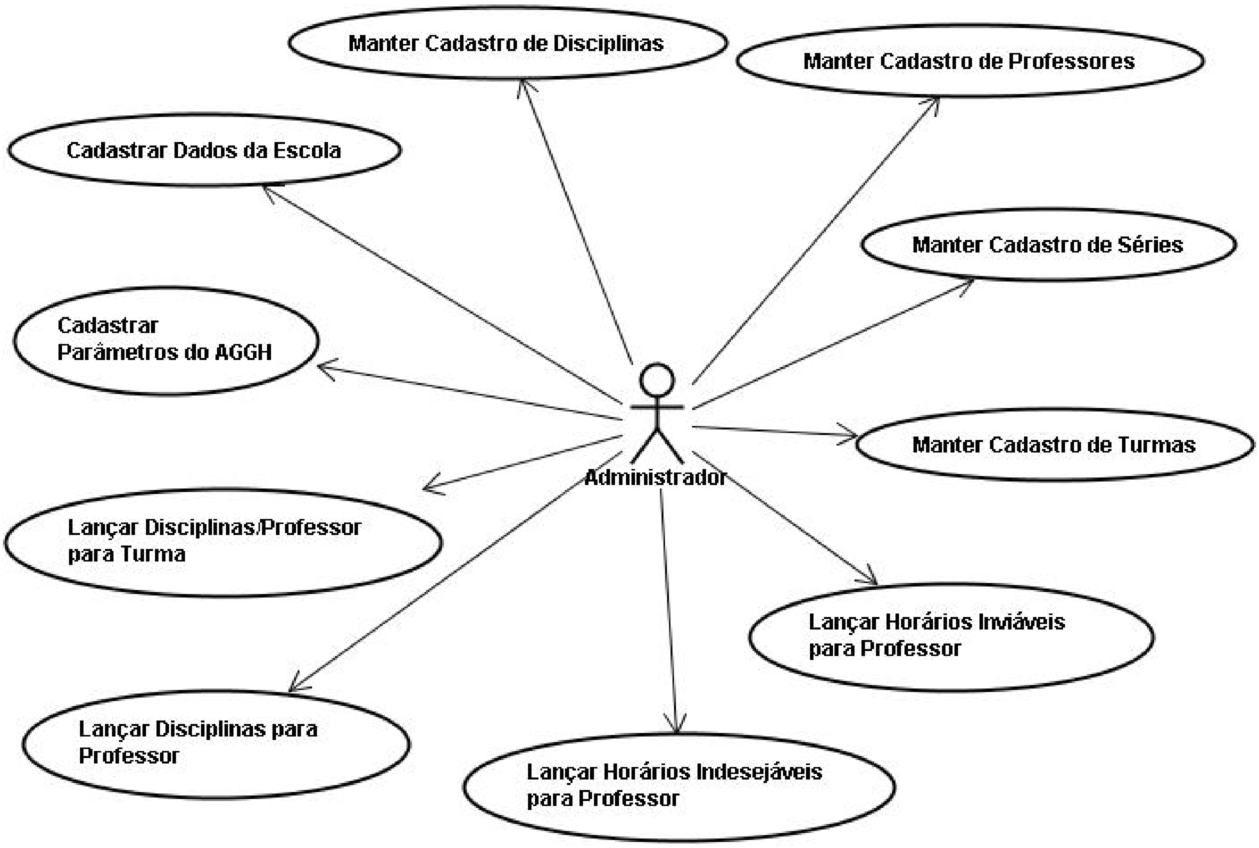


Figura 3.4 - Diagrama de Caso de Uso: Cadastros e Lançamentos, em UML

3.3.2.2. Diagrama de Caso de Uso: Geração de Horários

Já a Figura 3.5 apresenta as funcionalidades que o usuário de perfil **Consulta** pode executar. A seta partindo do usuário Administrador indica que este pode executar qualquer funcionalidade que o usuário Consulta tem acesso.

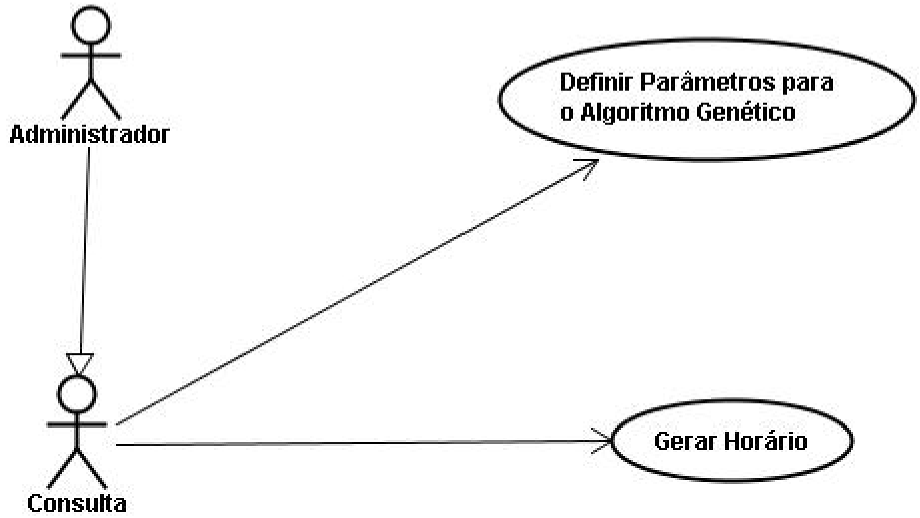


Figura 3.5 - Diagrama de Caso de Uso: Geração de Horários, em UML

3.3.2.3. Diagrama de Atividade: Lançar Disciplina para Professor

Booch et al. (2001) conceitua Diagrama de Atividades como diagramas que capturam ações e seus resultados. Eles focam o trabalho executado na implementação de uma operação e suas atividades numa instância de objeto.

A Figura 3.6 ilustra todas as atividades que o sistema deve executar para concretizar a operação de lançar disciplinas para professor. Veja que o diagrama é dividido em três partes, sendo que cada uma delas corresponde a uma das entidades do sistema.

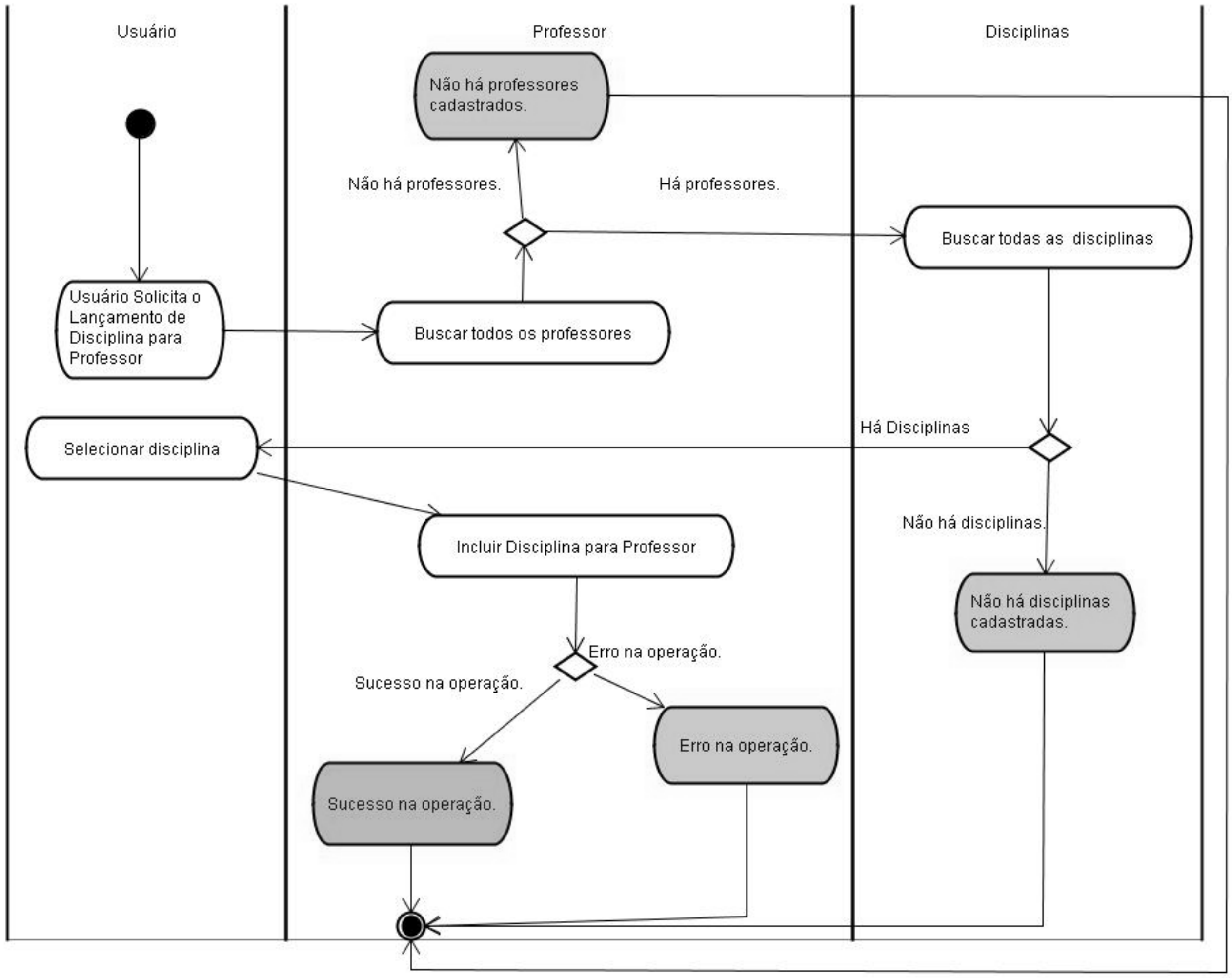


Figura 3.6 - Diagrama de Atividade: Lançar Disciplina para Professor, em UML

3.3.2.4. Diagrama de Classes: Classes Manager

Segundo Booch et al. (2001), Diagrama de Classes demonstram a estrutura estática das classes de um sistema, onde estas representam as "coisas" que são gerenciadas pela aplicação modelada. Ainda segundo Booch et al. (2001), uma classe num diagrama pode ser diretamente implementada utilizando-se uma linguagem de programação orientada a objetos, no caso deste trabalho, a Linguagem Java.

A partir da Figura 3.7 pode-se observar as classes manipuladores do sistema. Elas estão identificadas, descritas com seus métodos e relacionadas entre si. Geralmente um sistema possui mais de um diagrama, pois nem todas se encaixam em um diagrama específico.

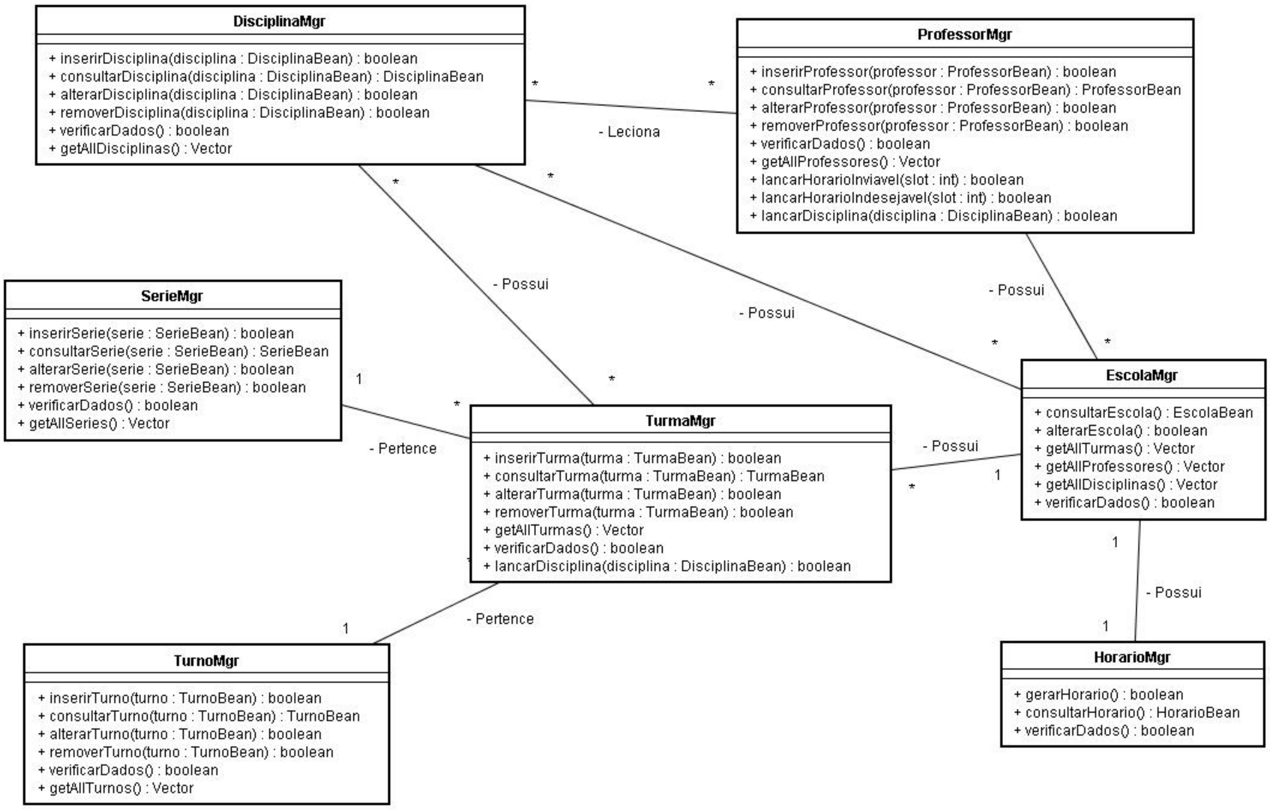


Figura 3.7 - Diagrama de Classes: Classes Manager, em UML

3.3.2.5. Modelagem de Dados

Por se tratar de uma aplicação de banco de dados, a modelagem de dados foi também construída. Segundo Elmasri & Navathe (2005) a modelagem conceitual é uma fase muito importante no planejamento de uma aplicação de um banco de dados bem sucedida.

Ainda segundo Elmasri & Navathe (2005), o modelo relacional representa o banco de dados como uma coleção de relações. Informalmente, cada relação se parece com uma tabela de valores.

Na Figura 3.8 tem-se o diagrama para o esquema do banco de dados relacional do sistema. Cada tabela é chamada de relação e cada cabeçalho de coluna é conhecido como atributo. Os atributos sublinhados em cada tabela indicam a chave-primária e a indicação “(FK)” exibe a chave-estrangeira em cada relação.

T_Disciplina	
CODIGO	NOME

T_Serie	
CODIGO	NOME

T_Turno	
CODIGO	NOME

T_Professor	
CODIGO	NOME

T_Escola	
CODIGO	NOME

T_Horario			
CODIGO	NUMERO_DIAS	NUMERO_HORARIOS_POR_DIA	CODIGO_ESCOLA (FK)

T_Turma					
CODIGO	NOME	ANO	CODIGO_TURNO	CODIGO_SERIE	CODIGO_ESCOLA (FK)

T_Disciplinas_Turma	
CODIGO_DISCIPLINA	CODIGO_TURMA (FK)

T_Disciplinas_Professor	
CODIGO_DISCIPLINA	CODIGO_PROFESSOR (FK)

T_Horario			
CODIGO	NUMERO_DIAS	NUMERO_HORARIOS_POR_DIA	CODIGO_ESCOLA (FK)

T_Professores_HorarioInviavel	
CODIGO_PROFESSOR (FK)	CODIGO_HORARIO_INVIAVEL

T_Professores_HorarioIndesejavel	
CODIGO_PROFESSOR (FK)	CODIGO_HORARIO_INDESEJAVEL

T_Disciplinas_Escola	
CODIGO_DISCIPLINA	CODIGO_ESCOLA (FK)

T_Professores_Escola	
CODIGO_PROFESSOR	CODIGO_ESCOLA (FK)

T_Parametros				
CODIGO	NOME	NUM_GERACOES	TAM_POPULACAO	TAXA_MUTACAO
P_CHOQUE	P_H_INVIAVEL	NUM_ELETISMO	TAXA_CRUZAMENTO	P_H_INDESEJAVEL
P_BLOCO	P_MONOPOLIO	P_JANELA	P_AULA_ISOLADA	P_MAX_DISC_DIA

Figura 3.8 - Diagrama de Esquema de Dados

3.3.3. Funcionalidades

Durante a construção da modelagem do sistema foram identificadas as funcionalidades indispensáveis para seu funcionamento. O desenvolvimento foi baseado nesta lista, que está descrita a seguir.

- Cadastro de Dados da Escola
- Cadastro de Disciplinas;
- Cadastro de Professores;
- Cadastro de Séries;
- Cadastro de Turmas;
- Lançamento de Disciplinas para Professor;
- Lançamento de Disciplinas/Professor para Turmas;
- Lançamento de Horários Inviáveis para Professor;
- Lançamento de Horários Indesejáveis para Professor;
- Relatórios Diversos;
- Configuração de Parâmetros do Algoritmo Genético; e
- Geração de Horário;

As subseções posteriores descrevem como proceder para utilizar o sistema.

3.3.4. Dados de Entrada

O sistema possui interface gráfica a fim de facilitar principalmente a inclusão dos dados de entrada e os parâmetros do algoritmo genético. Como dados de entrada, entende-se:

- Dados da Escola;
- O número e horários disponíveis;
- As disciplinas;
- Os professores
- As restrições nos horários dos professores;
- As turmas;
- A relação turmas, disciplinas e professores;

A Figura 3.9 exibe a tela de cadastro de professores. Para efetuar o cadastro de um professor, o usuário deve apenas preencher seu nome e clicar no botão “Cadastrar”. Além disso, é possível localizar, alterar e remover professores já cadastrados.

Figura 3.9 - Tela de Cadastro de Professores

A Figura 3.10 exibe a tela de lançamento de disciplinas para professor. Para tanto, o usuário deve apenas escolher um professor, uma disciplina e clicar no botão “Inserir”.

Figura 3.10 - Tela de Lançamento de Disciplinas para Professor

Já a Figura 3.11 mostra o lançamento de restrições no horário dos professores. Para efetuar o lançamento dos horários inviáveis, o usuário deve selecionar o professor desejado e clicar em cada uma das "*check-box*" correspondentes ao horário que o professor não pode dar aula. Na figura podemos observar que a professora "Terezinha Rezende" não pode dar aulas em todos os horários da sexta-feira.

Figura 3.11 - Tela Lançamento de Horários Inviáveis para Professor

A Tabela 3.1 exibe os parâmetros do algoritmo genético e do problema de geração de horários que devem ser incluídos no sistema antes da execução do algoritmo.

Tabela 3.1 - Parâmetros de Configuração

Parâmetros
Tamanho da População
Número de Gerações
Taxa de Mutação
Taxa de Cruzamento
Número de Indivíduos Preservados no Eletismo
Peso da Penalidade Choque de Professores
Peso da Penalidade Horário Inviável
Peso da Penalidade Horário Indesejável
Peso da Penalidade Bloco de Horário
Peso da Penalidade Monopólio de Aula
Peso da Penalidade Janela
Peso da Penalidade Aula Isolada
Máximo de Disciplinas por Dia
Número de Dias
Número de Horários
Número de Turmas

A Figura 3.12 mostra a tela de cadastro destes parâmetros. Os Parâmetros Número de Dias, Horários e Turmas foram implementados na funcionalidade de cadastro de dados da escola.

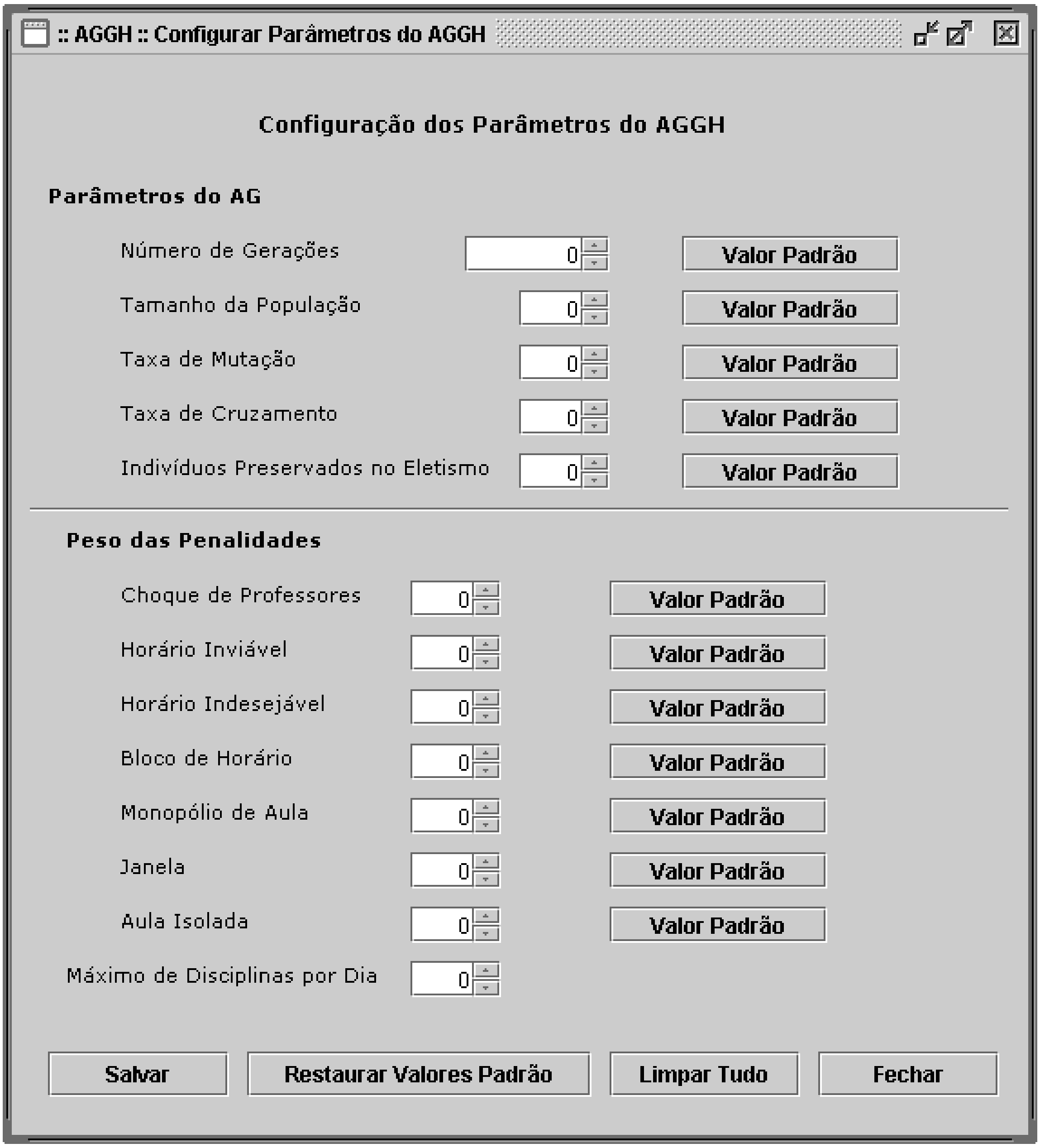


Figura 3.12 - Tela de Configuração dos Parâmetros do AGGH

3.3.5. Funcionamento

Com todas as entradas necessárias já armazenadas no sistema, o passo seguinte passa a ser a "calibragem" do algoritmo. Isto é, o usuário precisa definir os parâmetros do algoritmo (conforme Tabela 3.1). Geralmente este passo é feito apenas uma única vez, ou até que o algoritmo retorne soluções que se enquadrem nas necessidades do usuário.

Definidos os parâmetros, basta ao usuário iniciar a execução do algoritmo. A Figura 3.13 exibe a tela de execução do mesmo. Note a presença de um indicador de progresso.

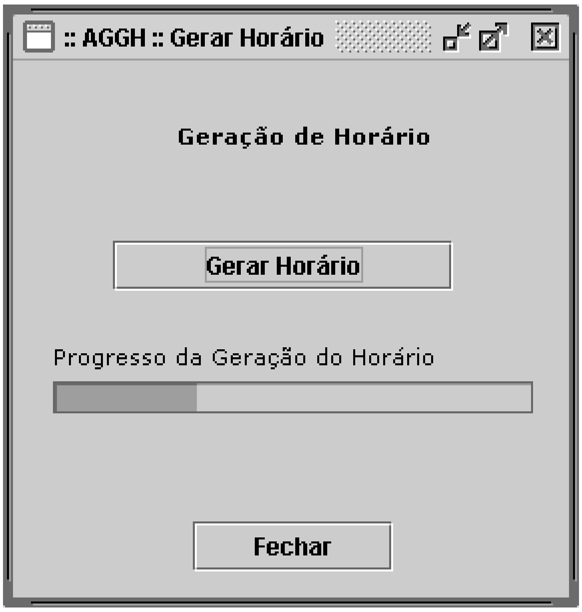


Figura 3.13 - Tela de Geração de Horário

3.3.6. Dados de Saída

Após a entrada dos dados e a execução do algoritmo, o horário gerado é exibido no formato HTML. Depois que um horário é gerado ele não pode ser modificado pelo sistema, somente manualmente.

A Figura 3.14 mostra parte de um horário das turmas gerado pelo sistema.

Turma: 1D				
SEGUNDA	**TERÇA**	**QUARTA**	**QUINTA**	**SEXTA**
HISTÓRIA /Marlene	QUÍMICA /Betânia	QUÍMICA /Betânia	FÍSICA /Graziela	MATEMÁTICA/Erotildes
PORTUGUÊS /Ana Lúcia	FÍSICA /Graziela	ED. FÍSICA/Nara	BIOLOGIA /Ana Silva	PORTUGUÊS /Ana Lúcia
ED. FÍSICA/Nara	PORTUGUÊS /Ana Lúcia	BIOLOGIA /Ana Silva	MATEMÁTICA/Erotildes	QUÍMICA /Betânia
INGLÊS /Marisa	PORTUGUÊS /Ana Lúcia	MATEMÁTICA/Erotildes	PORTUGUÊS /Ana Lúcia	HISTÓRIA /Marlene
BIOLOGIA /Ana Silva	GEOGRAFIA /Maria Emilia	INGLÊS /Marisa	GEOGRAFIA /Maria Emilia	FÍSICA /Graziela
Turma: 2C				
SEGUNDA	**TERÇA**	**QUARTA**	**QUINTA**	**SEXTA**
FÍSICA /Graziela	FÍSICA /Graziela	ED. FÍSICA/Nara	QUÍMICA /Betânia	QUÍMICA /Betânia
ED. FÍSICA/Nara	MATEMÁTICA/Erotildes	MATEMÁTICA/Erotildes	MATEMÁTICA/Erotildes	INGLÊS /Marisa
MATEMÁTICA/Erotildes	HISTÓRIA /Marlene	QUÍMICA /Betânia	BIOLOGIA /Ana Silva	PORTUGUÊS /Ana Lúcia
BIOLOGIA /Ana Silva	GEOGRAFIA /Maria Emilia	INGLÊS /Marisa	GEOGRAFIA /Maria Emilia	BIOLOGIA /Ana Silva
PORTUGUÊS /Ana Lúcia	PORTUGUÊS /Ana Lúcia	PORTUGUÊS /Ana Lúcia	PORTUGUÊS /Ana Lúcia	HISTÓRIA /Marlene
Turma: 3B				
SEGUNDA	**TERÇA**	**QUARTA**	**QUINTA**	**SEXTA**
ED. FÍSICA/Nara	HISTÓRIA /Marlene	FÍSICA /Toninho	BIOLOGIA /Ana Silva	ARTES /Israel
HISTÓRIA /Marlene	GEOGRAFIA /Maria Emilia	BIOLOGIA /Ana Silva	GEOGRAFIA /Maria Emilia	PORTUGUÊS /Elisa
PORTUGUÊS /Elisa	PORTUGUÊS /Elisa	ED. FÍSICA/Nara	QUÍMICA /Betânia	MATEMÁTICA/Erotildes
MATEMÁTICA/Erotildes	MATEMÁTICA/Erotildes	QUÍMICA /Betânia	GEOGRAFIA /Natali	QUÍMICA /Betânia
GEOGRAFIA /Natali	FÍSICA /Toninho	PORTUGUÊS /Elisa	PORTUGUÊS /Elisa	BIOLOGIA /Ana Silva
Turma: 3C				
SEGUNDA	**TERÇA**	**QUARTA**	**QUINTA**	**SEXTA**
ARTES /Israel	ED. FÍSICA/Adria	BIOLOGIA /Ana Silva	GEOGRAFIA /Maria Emilia	ED. FÍSICA/Adria
QUÍMICA /Betânia	QUÍMICA /Betânia	FÍSICA /Toninho	QUÍMICA /Betânia	PORTUGUÊS /Rose
BIOLOGIA /Ana Silva	GEOGRAFIA /Maria Emilia	PORTUGUÊS /Rose	PORTUGUÊS /Ana Lúcia	GEOGRAFIA /Natali
PORTUGUÊS /Ana Lúcia	GEOGRAFIA /Natali	PORTUGUÊS /Ana Lúcia	BIOLOGIA /Ana Silva	PORTUGUÊS /Ana Lúcia
FÍSICA /Toninho	MATEMÁTICA/Zezinho	MATEMÁTICA/Zezinho	MATEMÁTICA/Zezinho	PORTUGUÊS /Ana Lúcia

Figura 3.14 - Parte de um Horário Gerado - Visão para as Turmas

Já a Figura 3.15 exibe o mesmo horário, porém, na visão para os professores.

Professor: ANA LÚCIA				
__	__	__	__	__
1D	2J	__	2J	1D
2J	1D	__	3C	2C
3C	1D	3C	1D	3C
2C	2C	2C	2C	3C

Professor: MARLY				
__	__	__	__	__
__	1K	1K	1K	5A
__	5A	__	5A	__
1K	__	5A	__	__
__	__	__	__	__

Professor: MARLENE				
1D	3B	__	__	__
3B	5A	5A	__	__
__	2C	1K	__	__
__	1K	__	__	1D
__	__	__	__	2C

Professor: MARIA EMILIA				
__	__	__	3C	__
__	3B	__	3B	__
__	3C	__	__	__
__	2C	__	2C	__
__	1D	__	1D	__

Professor: ANA SILVA				
__	__	3C	3B	__
__	__	3B	1D	__
3C	__	1D	2C	__
2C	__	__	3C	2C
1D	__	__	__	3B

Professor: BETÂNIA				
__	1D	1D	2C	2C
3C	3C	2J	3C	2J
__	2J	2C	3B	1D
__	__	3B	__	3B
__	__	__	__	__

Professor: TONINHO				
__	__	3B	__	__
__	__	3C	__	__
__	__	2J	__	__
2J	2J	__	__	__
3C	3B	__	__	__

Professor: MARISA				
__	__	__	__	__
1K	__	__	__	2C
6A	__	__	__	__
1D	__	2C	__	6A
__	__	1D	__	__

Professor: NARA				
3B	__	2C	__	__
2C	__	1D	__	__
1D	__	3B	__	__
__	__	__	__	__
__	__	__	__	__

Professor: ELISA				
__	__	__	__	__
6A	6A	6A	__	3B
3B	3B	5A	1K	5A
5A	5A	1K	1K	1K
__	__	3B	3B	__

Professor: ADRIA				
__	3C	__	__	3C
__	__	__	__	__
__	__	__	__	__
__	__	__	__	__
__	__	__	__	__

Professor: ISRAEL				
3C	__	__	__	3B
5A	__	__	__	__
__	__	__	__	__
__	__	__	__	__
__	__	__	__	__

Professor: FABIOLA				
__	__	__	__	__
__	__	__	6A	1K
__	1K	__	2J	2J
__	6A	__	5A	5A
__	__	__	__	__

Professor: NATALI				
__	__	__	__	__
__	__	__	__	__
__	__	__	__	3C
__	3C	__	3B	__
3B	__	__	__	__

Professor: GRAZIELA				
2C	2C	__	1D	__
__	1D	__	__	__
__	__	__	__	__
__	__	__	__	__
__	__	__	__	1D

Professor: DAYSE				
__	__	__	__	__
__	__	__	5A	__
5A	__	__	6A	__
6A	__	__	__	__
__	__	__	__	__

Professor: DOUGLAS				
__	__	__	__	__
2J	__	__	__	__
1K	__	__	__	1K
__	__	__	__	2J
__	__	__	__	__

Professor: EROTILDES				
__	__	__	__	1D
__	2C	2C	2C	6A
2C	6A	6A	1D	3B
3B	3B	1D	6A	__
__	__	__	__	__

Professor: ZEZINHO				
__	__	__	__	__
__	__	__	__	__
__	__	__	__	__
__	__	2J	2J	__
__	3C	3C	3C	__

Professor: ROSE				
__	__	__	__	__
__	__	__	__	3C
__	__	3C	__	6A
__	__	6A	__	__
__	__	__	__	__

Figura 3.15 - Horário Gerado - Visão para Professores

Além disso, é exibido um relatório final com os dados sobre o horário gerado. A Figura 3.16 exibe este resumo.

Exame detalhado do horário:
Número de choques de professores: 0
Número de aulas em horário inviável para algum professor: 0
Número de aulas em horário indesejável para algum professor: 0
Número de vezes que houve blocos de disciplinas: 3
Número de aulas que estão sendo ministradas desrespeitando o limite diário de aulas de uma mesma disciplina: 3
Número de janelas no horário dos professores: 3
Número de aulas isoladas no horário dos professores: 11
QUALIDADE DO HORÁRIO: 0.81493305
Tempo de Execução: 22.58517 minuto(s).

Figura 3.16 - Dados do horário gerado

3.4. Considerações Finais do Capítulo

Foi apresentada, neste capítulo, a proposta de solução para o problema abordado. O motivo pela escolha do AG na resolução do problema, bem como a explicação de cada uma de suas partes e as decisões de implementação tomadas foram apresentadas. Todas as tecnologias e ferramentas utilizadas para o desenvolvimento do sistema e os passos para sua construção também foram detalhados neste capítulo.

Ficou clara a importância de uma boa escolha de representação de solução. Assim como a necessidade de construção da modelagem do sistema para facilitar seu desenvolvimento.

Foi mostrada também a aplicação desenvolvida, com muitas imagens da sua interface gráfica.

4. RESULTADOS E DISCUSSÃO

O algoritmo foi implementado em JAVA™ e testado em duas máquinas diferentes: Pentium III 1000 Mhz com 512 Mb de RAM e Pentium 4 1.8 Ghz com 256 Mb de RAM, ambos sob o sistema operacional Windows 2000.

Apesar de existirem vários trabalhos correlatos na literatura, optou-se neste trabalho pela utilização de casos reais para teste do algoritmo desenvolvido. Desde o início, já se tinha a expectativa de utilização dos horários gerados em duas escolas, sendo esta uma das principais motivações do presente trabalho.

Além disso, foi relatado por parte da diretoria das escolas toda a dificuldade de se gerar um horário manual, além do tempo que esta atividade requisitava, inclusive sendo exercida por mais de uma pessoa.

Para testar o algoritmo foram utilizadas, portanto, oito instâncias reais, sendo cinco para o ano de 2005 de duas escolas diferentes: Escola Estadual Padre Rogério Abdala (EEPRA - Monsenhor Paulo / MG) e Escola Municipal Álvaro Botelho (EMAB - Lavras / MG) e três para o ano de 2006 para a E. E. Padre Rogério Abdala.

A tabela a seguir resume os dados de todas as oito instâncias reais utilizadas no trabalho.

Tabela 4.1 - Características das Turmas utilizadas na Geração do Horário

Instância	Escola	Ano	Turno	Número de Turmas	Número de Disciplinas	Número de Professores
1	EEPRA	2005	Matutino	8	13	17
2	EEPRA	2005	Vespertino	8	13	20
3	EEPRA	2005	Noturno	8	14	18
4	EMAB	2005	Matutino	12	9	22
5	EMAB	2005	Noturno	11	5	14
6	EEPRA	2006	Matutino	7	13	17
7	EEPRA	2006	Vespertino	8	9	16
8	EEPRA	2006	Noturno	8	13	20

A partir da Tabela 4.1 podemos observar que o número de turmas das instâncias da EMAB é maior, contudo, o número de disciplinas é menor que das instâncias da EEPRA. Este contraste mostrou-se muito benéfico para o algoritmo, já que propiciou a avaliação do mesmo em aspectos diferentes na geração do horário para as escolas escolhidas. A seguir, neste capítulo, este fato será mais bem explicado.

Inicialmente foram utilizadas as instâncias um e dois visando a calibragem dos seguintes parâmetros: tamanho da população, número de gerações, taxa de mutação, taxa de cruzamento e o valor das penalidades usadas no cálculo do *fitness*.

A Tabela 4.2 mostra os valores escolhidos empiricamente:

Tabela 4.2 - Valores dos parâmetros e penalidades do Algoritmo Genético

Parâmetros	Valor
Tamanho da População	140
Número de Gerações	15000
Taxa de Mutação	2 mutações a cada geração
Taxa de Cruzamento	60%
Peso da Penalidade Choque de Professores	12,50
Peso da Penalidade Horário Inviável	12,75
Peso da Penalidade Horário Indesejável	1,25
Peso da Penalidade Bloco de Horário	2
Peso da Penalidade Monopólio de Aula	2
Peso da Penalidade Janela	3,5
Peso da Penalidade Aula Isolada	3,5
Máximo de Disciplinas por Dia	1

Entretanto, estes valores não foram unânimes para todas as instâncias. Como mencionado anteriormente, o contraste entre as duas escolas fez com que valores e pesos diferentes fossem utilizados para cada caso. A seguir, uma avaliação de cada uma das instâncias com seus respectivos resultados na geração do horário. E a Tabela 4.3 resume estes resultados.

Para cada instância foram realizadas em média 15 execuções do algoritmo.

A **instância um**, devido ao elevado número de aulas que cada professor possuía, ficou mais distante do limite superior conceitual Em alguns casos, havia professores que lecionavam 23 aulas, num total de 25 horários disponíveis. Isto reduzia drasticamente o espaço de busca, contudo, na maioria dos horários gerados, as restrições fortes estavam sempre zeradas, conforme pode ser observado a Tabela 4.3. A redução de janelas e aulas isoladas atingiu resultados satisfatórios, na medida em que superou a expectativa da diretoria da escola.

A **instância dois** possuía um menor número de restrições nos horários dos professores. Outra característica foi o fato de todas as aulas da disciplina de Educação Física serem realizadas somente nas duas últimas aulas de sexta-feira. A razão disso se refere ao fato destas aulas serem realizadas em outros dias da semana e em outro espaço físico fora da escola. Inicialmente o algoritmo não foi proposto para contemplar esta restrição (vide restrição 7 da subseção 2.4.2 - Restrições em horários reais), contudo, a adaptação do mesmo mostrou-se fácil e muito eficiente.

Para contornar esta necessidade que surgiu após o desenvolvimento do algoritmo, foi necessário apenas alocar um professor fictício para cada turma, sendo que este seria responsável apenas pela disciplina de “Educação Física” e para todos os outros professores de cada turma, foram definidos que os dois últimos horários de sexta-feira seriam inviáveis. Desta forma, durante a geração do horário, a disciplina de “Educação Física” ficava sempre alocada nos dois últimos horários de sexta-feira.

Apesar desta adaptação, conforme podemos constatar na Tabela 4.3, a instância dois retornou horários sem violação nas restrições fortes e com minimizações satisfatórias no número de janelas e de aulas isoladas.

Já **instância três** apresentou uma particularidade: quatro de suas turmas não possuíam nem a primeira e nem a última aula, por se tratarem de quatro turmas de ensino supletivo. Novamente foi necessária a implementação adicional da mesma restrição da instância 2, citada anteriormente, porém de uma forma diferente.

Desta vez não era possível colocar inviabilidades no horário de todos os professores, já que as outras quatro turmas possuíam seu horário de forma regular (com cinco aulas em um mesmo dia). Partindo dessa premissa, foi necessário alocar dois professores fictícios para cada turma do ensino supletivo, sendo que o primeiro deles teria como horários inviáveis os quatro últimos horários de cada dia, e o segundo teria como horários inviáveis os quatro primeiros horários de cada dia. Com isso, evitava-se a geração de choques entre estes professores enquanto os outros professores reais da turma seriam alocados nos horários válidos, isto é, nos horários 2, 3 e 4 das turmas supletivas e nos horários 1, 2, 3, 4 e 5 das turmas regulares.

Embora tenha sofrido esta adaptação, a instância três sempre atingia os valores de *fitness* mais elevados para as instâncias da EEPRA. Isto pode ser comprovado analisando os resultados da Tabela 4.3. Especial atenção para a considerável redução das aulas isoladas e principalmente das janelas.

Com o maior número de turmas entre as instâncias testadas, a **instância quatro** foi a que gastou o maior tempo de execução. Isto se deve ao fato da existência de elevado número de turmas, o que, de acordo com a representação dos indivíduos proposta na seção 3.2.2 leva ao maior número de matrizes e de operações envolvendo as mesmas. Atingiu resultados satisfatórios, porém o algoritmo não conseguiu eliminar em sua totalidade os horários indesejáveis. Isto se deve em parte ao excesso de restrições nos horários dos professores. Veja a Tabela 4.3 para maiores detalhes.

Já a **instância cinco**, que possuía um reduzido número de disciplinas, apresentou elevados valores de *fitness*. Estes resultados indicam que com menos restrições (maior espaço de busca) para as instâncias avaliadas, os *fitness* dos horários ficam mais próximos do limite superior conceitual. Através da Tabela 4.3, podemos observar os *fitness* retornados.

As instâncias de 2006 chamaram a atenção pelo excesso de restrições nos horários dos professores. Para todas elas foi necessária uma redução (ou alteração) nas preferências dos professores.

A distribuição das aulas para a **instância seis** mostrou-se muito parecida com a instância um. Isso facilitou inclusive o cadastro das aulas para a posterior geração do horário. Seu processamento foi o mais rápido comparando-se as outras duas instâncias de 2006. Isto se deve ao fato de esta instância possuir um número menor de turmas.

Assim como a instância seis, a **instância sete** mostrou-se praticamente a mesma do seu ano anterior (instância dois) em relação à distribuição das aulas. A alteração que ocorreu foi a substituição de uma turma de ensino médio por outra do ensino fundamental. Isto fez com que o período todo fosse composto apenas por turmas do ensino fundamental. Com isso o número de disciplinas e professores foi reduzido. Apesar disso, de acordo com a Tabela 4.3, a instância sete foi a que mais distante esteve do limite superior conceitual, sendo necessária por algumas vezes a redução de restrições para alguns de seus professores.

Ao contrário da sua instância equivalente em 2005, a **instância oito** apresentou geralmente valores de *fitness* bem abaixo da instância três. A geração do horário foi complicada pelo elevado número de professores e restrições associadas. Contudo, após a alteração de algumas inviabilidades dos professores, chegou-se a propostas de horários satisfatórias, de acordo com a diretoria da EEPRA. A Tabela 4.3 resume os melhores resultados encontrados para esta instância.

O fato de terem sido escolhidos duas escolas foi muito vantajoso, pois possibilitou o teste de duas visões diferentes de cada escola na geração de seus horários. Enquanto a escola de Monsenhor Paulo não desejava a geminação de aulas (formação de blocos de disciplinas), a escola de Lavras indicou esse agrupamento como necessário. Outra característica avaliada e que contribuiu para o teste do algoritmo foi o número diferente de professores e disciplinas para cada escola.

Nenhum dos horários gerados para a EEPRA violou as restrições “fortes” consideradas na função de adaptabilidade (vide seção 3.2.1): horários inviáveis e choques nos horários. A eliminação de janelas e aulas isoladas foi muito satisfatória, especialmente no resultado sete (instância três), onde o horário gerado possuía apenas duas janelas e três aulas isoladas no horário dos professores. Normalmente horários gerados de forma manual apresentam quatro ou cinco vezes mais estes números, de acordo com informações fornecidas pela escola. Este horário foi, inclusive, o que mais agradou a diretoria da EEPRA.

Quanto às outras restrições “fracas”, o algoritmo apresentou bons resultados. Já em relação às instâncias da EMAB, o algoritmo atingiu bons resultados, apesar do excessivo número de restrições. A grande formação de blocos de disciplinas foi um fator positivo nos horários gerados, conforme pode ser visto na Tabela 4.3. Este fato foi citado como relevante na qualidade dos horários, de acordo com a diretoria da EMAB.

Observou-se em ambas as escolas, especialmente para as instâncias de 2006 (EEPRA) a importância da interferência humana na escolha do melhor horário proposto pelo sistema dentro de 15 opções que foram produzidas. Fatores, como, prestígio de um determinado professor (não considerado no modelo) ou formação de aulas geminadas possuem um peso elevado na decisão final. Em virtude disso é possível afirmar a dificuldade de se desenvolver um sistema que resolva totalmente o problema de geração de horários em escolas.

A Tabela 4.3 apresenta um resumo dos melhores resultados obtidos.

Tabela 4.3 - Resultados alcançados pelo Algoritmo Genético

Resultado	Instância	*Fitness*	Tempo	Choques	Inviáveis	Indesejáveis	Blocos	Janelas	Aulas Isoladas
1	**1**	**0.638**	**21’17”**	**0**	**0**	**0**	**3**	**10**	**3**
2	1	0.642	20’50”	0	0	0	1	**9**	**4**
3	1	0.651	21’02”	0	0	0	2	**8**	**4**
4	**2**	**0.546**	**21’38”**	**0**	**0**	**0**	**9**	**3**	**1**
5	2	0.532	21’49”	0	0	0	8	**6**	**2**
6	2	0.528	21’26’	0	0	0	9	**7**	**3**
7	**3**	**0.913**	**21”12’**	**0**	**0**	**0**	**1**	**2**	**3**
8	3	0.819	21”02’	0	0	0	3	**1**	**7**
9	3	0.836	21”19’	0	0	0	1	**6**	**7**
10	4	0.598	28”54’	0	0	5	38*	**6**	**9**

11	4	0.598	26”31’	0	0	5	45*	**6**	**8**
12	4	0.625	26”23’	0	0	5	32*	**5**	**7**
13	5	0.781	24”03’	0	0	0	24*	**4**	**2**
14	5	0.781	24”47’	0	0	0	24*	**4**	**2**
15	5	0.699	23”59’	0	0	0	29*	**4**	**3**
16	6	0.597	18”35’	0	0	0	4	**7**	**2**
17	6	0.612	19”00’	0	0	0	3	**6**	**3**
18	6	0.588	18”50’	0	0	0	2	**5**	**6**
19	7	0.566	21”33’	0	0	0	7	**5**	**5**
20	7	0.601	21”10’	0	0	0	6	**8**	**4**
21	7	0.570	20”59’	0	0	0	6	**8**	**7**
22	8	0.603	21”18’	0	0	0	5	**6**	**6**
23	8	0.640	22”01’	0	0	0	4	**4**	**5**
24	8	0.592	21”50’	0	0	0	9	**5**	**4**

Os resultados em negrito foram os horários oficiais da Escola Estadual Padre Rogério Abdala em Monsenhor Paulo / MG no ano de 2005. Para o ano de 2006, as instâncias sublinhadas foram aquelas escolhidas pela diretoria da escola. Estes horários encontram-se nos apêndices deste trabalho.

Para os horários marcados com asterisco (*), era desejável que ocorresse a formação de blocos de disciplinas. Para que o algoritmo passe a desconsiderar esta restrição foi necessário apenas zerar o valor da penalidade referente à formação de blocos para as instâncias da EMAB.

4.1. Considerações Finais do Capítulo

Neste capítulo foram apresentadas todas as instâncias reais utilizadas no trabalho. Cada uma delas possuiu suas particularidades e, algumas delas, inclusive, restrições que inicialmente nem estavam previstas para tratamento do algoritmo.

Observou-se que o mais difícil na utilização de um algoritmo genético é na verdade a sua “calibragem”, isto é, a definição dos parâmetros corretos que melhores soluções retornam. Foi notado que para cada instância, uma configuração diferente era necessária.

Foi possível constatar a qualidade do algoritmo em situações diferentes que cada escola pode adotar.

5. CONCLUSÃO

Foi apresentada uma solução para um problema de geração de horários, considerando a eliminação simultânea de janelas e aulas isoladas. Para possibilitar uma solução de qualidade em tempo aceitável, foi proposto um algoritmo genético.

A partir de uma solução inicial aleatória e inviável é gerada uma solução de boa qualidade de acordo com as restrições impostas.

Conforme observado Tabela 4.3 é possível atestar a capacidade do algoritmo na eliminação de choques e inviabilidades nos horários dos professores, condições estas básicas para um horário viável. Outros requisitos para um bom horário também foram bem tratados pelo algoritmo implementado. Notou-se uma boa eliminação de janelas e aulas isoladas nos horários dos professores. Fato, este verificado, pelas duas últimas colunas da tabela citada (especialmente resultados 3, 4, 7, 8, 13 e 14) e pela afirmação por parte da diretoria de ambas as escolas.

Não foi possível comparar o horário gerado pela proposta apresentada com um horário gerado manualmente em virtude da impossibilidade de utilizar as mesmas restrições que as escolas usaram em anos letivos anteriores. Entretanto, verificou-se também a satisfação da diretoria da EEPRA e dos professores com os resultados alcançados com o algoritmo. Em comparação com horários gerados manualmente em anos anteriores, houve uma considerável redução no tempo gasto para gerar o mesmo. Enquanto que manualmente eram necessárias duas pessoas trabalhando durante, no mínimo, cinco dias, com a utilização do algoritmo, foi necessário apenas o processamento de sugestões de horários e uma pessoa responsável pela escolha do melhor horário para escola. Além disso, houve redução também no período de adaptação do horário gerado (nos horários gerados manualmente ocorreriam adaptações/correções por até dois meses) e relevante diminuição também no número de pedidos e reclamações envolvendo o horário gerado, caindo de em média 20 solicitações para apenas quatro pedidos, sendo que apenas dois ajustes manuais foram realizados.

Conclui-se, portanto, que o trabalho apresentado é uma importante colaboração para geração de horários escolares para instituições de ensino fundamental e médio. Adicionalmente, a importante capacidade de eliminação de janelas e aulas isoladas que o algoritmo proposto possui, foi fundamental para aceitação dos resultados.

5.1. Trabalhos Futuros

Como sugestão de trabalhos futuros, pode-se dividi-los em duas categorias: melhorias no algoritmo genético proposto e melhorias na aplicação desenvolvida.

Na geração da população inicial, poder-se-ia utilizar alguma heurística que substituísse a geração aleatória dos indivíduos. Com isso a população inicial poderia ter o número de inviabilidades reduzidas.

Em relação ao cruzamento e a mutação propostos, seria interessante a utilização de mecanismos que evitassem a geração de indivíduos inviáveis, isto é, o cruzamento e a mutação seriam implementados de maneira mais inteligente.

Cabe também como sugestão de melhoria futura, a adaptação do algoritmo proposto para a execução em paralelo. Assim, vários horários poderiam ser gerados ao mesmo tempo, ou inclusive, a geração de um único horário tornar-se-ia mais rápida.

A implementação de outras restrições, conforme citado na seção 2.4.2, traria mais aplicabilidade do algoritmo à realidade das escolas brasileiras. A restrição não tratada, "Prestígio do Professor" seria uma boa e mais indicada alternativa de melhoria.

Também como trabalho futuro, deve-se mencionar a integração desta aplicação, com outra que trata das questões de administração escolar, isto é, em uma única aplicação, seria possível, cadastrar as turmas, professores, disciplinas, alunos, lançamento de notas e faltas, além da geração do horário.

E por último, a mais importante proposta de trabalho futuro identificada, a construção de uma interface inteligente de geração de horários. O usuário seria capaz de utilizar esta interface para modificar o horário proposto pelo algoritmo. E esta interface seria a responsável por identificar possíveis choques e inviabilidades no horário e permitir ou não a alteração desejada pelo usuário.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AARTS, E.; KORST J., **Simulated Annealing and Boltzmann Machine**, John Wiley, 1989.

ABRAMSON, D. **Constructing schools timetables using simulated annealing: sequential and parallel algorithms**. Management Science, n. 37, v. 1, p. 98-113, 1991.

ABRAMSON, D.; ABELA, J. **A Parallel Genetic Algorithm for Solving the School Timetabling Problem**. In Proceedings of the 15th Australian Computer Science Conference. Hobart, Austrália, 1992.

ABRAMSON, D.; KRISHNAMOORTHY, M.; DANG, H. **Simulated annealing cooling schedules for the school timetabling problem**. Asia-Pacific Journal of Operational Research, v. 16, pp. 1-22, 1999.

AKARI, R. G. I., **O problema de roteamento de veículos e algumas meta-heurísticas**, Instituto Nacional de Pesquisas Espaçais – INPE, 1998.

BOOCH, G.; JACOBSON, I.; RUMBAUGH J.; OMG Unified Modeling Language Specification. Rational Software Corporation. v. 1.4. 2001.

CISCON, L. A., et al. **O Problema de Geração de Horários: um Foco na Eliminação de Janelas e Aulas Isoladas**. In: XXXVIII Simpósio Brasileiro de Pesquisa Operacional. Gramado - RS, 2005.

COLORNI A.; DORIGO, M.; MANIEZZO, V. **Meta-heuristics for high school timetabling**. Computational Optimization and Applications , v. 9, p. 275-298, 1998.

COOK S. A., **The Complexity of Theorem-Proving Procedures**, Symp. On Theory of Computing. 1971.

COOPER, T. B.; KINGSTON, J. H. **The solution of real instances of the timetabling problem**. The Computer Journal, v. 36, pp. 645-653, 1993.

COOPER, T. B.; KINGSTON, J. H. **The Complexity of Timetable Construction Problems** In Burke, E.K., Ross, P. (eds), Practice and Theory of Automated Timetabling, v. 1153, Lecture Notes in Computer Science, pp. 283-295, Springer-Verlag, Berlin, 1996.

COSTA, D. **A tabu search algorithm for computing an operational timetable**. European Journal of Operational Research, v. 76, p. 98-110, 1994.

DARWIN, C. **On the Origin of Species**. 1st edition, Harward University Press, MA, 1859.

DOWSLAND, K. A. **Simulated Annealing**, **Modern Heuristic Techniques for Combinatorial Problems**, Blackwell Scientific Publications, 20-69, 1993.

EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. Springer, Holanda, 2003.

ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados**. 4ª. Edição, Pearson Addison Wesley, São Paulo. 2005.

EVEN, S.; ITAI, A.; SHAMIR, A. **On the complexity of timetabling and multicommodity flow problems**. SIAM Journal of Computation, v. 5, pp. 691-703, 1976.

FEO, T. A.; RESENDE, M. G. C. **Greedy randomized adaptive search procedures**. Journal of Global Optimization, v. 6, pp. 109-133, 1995.

GLOVER, F. **Future paths for Integer Programming and links to Artificial Intelligence**. Computers and Operations Research, v. 5, pp. 553-549, 1986.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. The University of Alabama, 1989.

HANSEN, P. **The steepest ascent mildest descent heuristic for combinatorial programming**. In Congress on Numeriacal Methods in Combinatorial Optimization, Capri, Italy, 1986.

HERTZ, A. **Finding a feasible course schedule using tabu search**. Discrete Applied Mathematics, v. 35, pp. 225-270, 1992.

HOLLAND, J. H. **Adaptation in Natural and Artificial System**. University of Michigan Press, 1975.

KIRKPATRICK, S.; GELLAT, D. C.; VECCHE, M. P. **Optimization by Simulated Annealing**. Science, v. 220, pp. 671-680, 1983.

MARINHO, E. H. **Heurísticas Busca Tabu para o Problema de Programação de Tripulações de Ônibus Urbano**. UFF. Rio de Janeiro - RJ, 2005.

MITCHELL, M. **An Introdution to Genetic Algorithms**. Massachusetts Institute of Tecnology, 1996.

NORONHA, T. F. **Uma Abordagem sobre Estratégias Meta-heurísticas**. UFRGN, Natal - RN, 2000.

OLIVEIRA, H. C. B. **Algoritmo Evolutivo no Tratamento do Problema de Roteamento de Veículos com Janela de Tempo**. UFLA. Lavras - MG, 2005.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. USA: Dover Publications Inc., 1982.

RIBEIRO, C. C. **Meta-heuristics and Applications**. In Advanced School on Artificial Intelligence, Estorial, Portugal, 1996.

RIBEIRO FILHO, G. **Melhoramentos do Algoritmo Genético Construtivo e Novas Aplicações em Problemas de Agrupamento**. PhD thesis, INPE, São José dos Campos - SP. 2000.

RUSSEL, S. S.; NORVIG, P. **Artificial Intelligence. A modern approach**. 2ª edição, 2003.

SCHAERF, A. **Tabu search techniques for large high school timetabling problems**. In: Proceedings of the 30th National Conference on Artificial Intelligence, v. 1, pp. 363-368, 1996.

SCHAERF, A. **A survey of automated timetabling**. Artificial Intelligence Review, v. 13, pp. 87-127, 1999.

SILVA, A. S. N. **Estudo e Implementação, mediante Recozimento Simulado, do Problema de Alocação de Salas**. UFLA. Lavras - MG, 2005.

SOUZA, M. J. F. **Programação de horários em escolas: uma aproximação por meta-heurísticas**, Tese de Doutorado, Programa de Engenharia de Sistemas e Computação, UFRJ, Rio de Janeiro - RJ, 2000.

SOUZA, M. J. F. et al. **Um Algoritmo Evolutivo Híbrido para o Problema de Programação de Horários em Escolas**. UFOP, Ouro Preto - MG, 2002.

SRIVINAS et al. **Genetic Algorithmics: A survey**. IEEE Computer Society, p 17-26, 1994.

TIMÓTEO, G. T. S. **Desenvolvimento de um Algoritmo Genético para a Resolução do Timetabling**. UFLA. Lavras - MG, 2002.

WERRA, D de. **An introduction to timetabling**. European Journal of Operational Research Society, v. 19, pp 151-162, 1985.

7. APÊNDICES

7.1. Tabela para preenchimento de horários

Tabela 7.1 - Tabela de Preenchimento de Horários

Turno				
	Série 1	Série 2	...	Série N
Disciplina 1	<número de aulas> <Nome do Prof 1>	<número de aulas> <Nome do Prof 1>	<número de aulas> <Nome do Prof 1>	<número de aulas> <Nome do Prof 1>
Disciplina 2	<número de aulas> <Nome do Prof 2>	<número de aulas> <Nome do Prof 2>	<número de aulas> <Nome do Prof 2>	<número de aulas> <Nome do Prof 2>
Disciplina 3	<número de aulas> <Nome do Prof 3>	<número de aulas> <Nome do Prof 3>	<número de aulas> <Nome do Prof 3>	<número de aulas> <Nome do Prof 3>
...	...	...	...	...
Disciplina N	<número de aulas> <Nome do Prof N>	<número de aulas> <Nome do Prof N>	<número de aulas> <Nome do Prof N>	<número de aulas> <Nome do Prof N>

7.2. Horário Oficial de 2005 (EEPRA) gerado pelo Sistema

Corresponde ao resultado 7 da Tabela 4.3

Horário das Turmas – NOTURNO

Turma: 1D

SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
QUÍMICA /Giorgia	QUÍMICA /Giorgia	MATEMÁTICA/Marli	FÍSICA /Toninho	INGLÊS /Marisa
MATEMÁTICA/Marli	ED. FÍSICA/Ária	QUÍMICA /Giorgia	INGLÊS /Marisa	FÍSICA /Toninho
BIOLOGIA /Ana Silva	BIOLOGIA /Ana Silva	BIOLOGIA /Ana Silva	GEOGRAFIA /Vera Carin	PORTUGUÊS /Ana Lúcia
GEOGRAFIA /Vera Carin	HISTÓRIA /Liliane	PORTUGUÊS /Ana Lúcia	PORTUGUÊS /Ana Lúcia	ED. FÍSICA/Ádria
PORTUGUÊS /Ana Lúcia	PORTUGUÊS /Ana Lúcia	FÍSICA /Toninho	HISTÓRIA /Liliane	MATEMÁTICA/Marli

Turma: 2C

SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
PORTUGUÊS /Ana Lúcia	MATEMÁTICA/Erotildes	INGLÊS /Marisa	ED. FÍSICA/Ádria	GEOGRAFIA /Vera Carin
GEOGRAFIA /Vera Carin	QUÍMICA /Giorgia	MATEMÁTICA/Erotildes	BIOLOGIA /Ana Silva	BIOLOGIA /Ana Silva
MATEMÁTICA/Erotildes	PORTUGUÊS /Ana Lúcia	HISTÓRIA /Marlene	PORTUGUÊS /Ana Lúcia	HISTÓRIA /Marlene
BIOLOGIA /Ana Silva	ED. FÍSICA/Ádria	FÍSICA /Toninho	INGLÊS /Marisa	MATEMÁTICA/Erotildes
QUÍMICA /Giorgia	FÍSICA /Toninho	PORTUGUÊS /Ana Lúcia	QUÍMICA /Giorgia	PORTUGUÊS /Ana Lúcia

Turma: 3C

SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MATEMÁTICA/Marli	MATEMÁTICA/Marli	QUÍMICA /Giorgia	BIOLOGIA /Ana Silva	ARTES /T. Zanin
FÍSICA /Toninho	PORTUGUÊS /Ivolina	ED. FÍSICA/Nara	PORTUGUÊS /Ivolina	MATEMÁTICA/Marli
PORTUGUÊS /Ivolina	INGLÊS /Marisa	FÍSICA /Toninho	INGLÊS /Marisa	GEOGRAFIA /Maria Emilia
ED. FÍSICA/Nara	QUÍMICA /Giorgia	PORTUGUÊS /Ivolina	QUÍMICA /Giorgia	PORTUGUÊS /Ivolina
BIOLOGIA /Ana Silva	BIOLOGIA /Ana Silva	HISTÓRIA /Marlene	GEOGRAFIA /M. Emília	HISTÓRIA /Marlene

Turma: 3D

SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
FÍSICA /Toninho	PORTUGUÊS /Ivolina	ED. FÍSICA/Nara	PORTUGUÊS /Ivolina	MATEMÁTICA/Marli
PORTUGUÊS /Ivolina	INGLÊS /Marisa	BIOLOGIA /Ana Silva	QUÍMICA /Giorgia	GEOGRAFIA /Maria Emília
ED. FÍSICA/Nara	FÍSICA /Toninho	PORTUGUÊS /Ivolina	BIOLOGIA /Ana Silva	ARTES /T. Zanin
QUÍMICA /Giorgia	BIOLOGIA /Ana Silva	HISTÓRIA /Marlene	GEOGRAFIA /Maria Emília	HISTÓRIA /Marlene
MATEMÁTICA/Marli	MATEMÁTICA/Marli	QUÍMICA /Giorgia	INGLÊS /Marisa	PORTUGUÊS /Ivolina

Turma: 2A - SUPLETIVO

SEGUNDA	- TERÇA	- QUARTA	- QUINTA	- SEXTA
PORTUGUÊS /Ana Lúcia	- BIOLOGIA /Vânia	- MATEMÁTICA/Marli	- FÍSICA /Toninho	- PORTUGUÊS /Ana Lúcia
QUÍMICA /Giorgia	- MATEMÁTICA/Marli	- QUÍMICA /Giorgia	- QUÍMICA /Giorgia	- FÍSICA /Toninho
BIOLOGIA /Vânia	- FÍSICA /Toninho	- GEOGRAFIA /Vera Carin	- GEOGRAFIA /Vera Carin	- PORTUGUÊS /Ana Lúcia

Turma: 2B - SUPLETIVO

SEGUNDA	- TERÇA	- QUARTA	- QUINTA	- SEXTA
QUÍMICA /Giorgia	- FÍSICA /Toninho	- FÍSICA /Toninho	- GEOGRAFIA /Vera Carin	- GEOGRAFIA /Vera Carin
MATEMÁTICA/Marli	- QUÍMICA /Giorgia	- PORTUGUÊS /Lurdinha	- BIOLOGIA /Dayse	- PORTUGUÊS /Lurdinha
BIOLOGIA /Dayse	- PORTUGUÊS /Lurdinha	- QUÍMICA /Giorgia	- MATEMÁTICA/Marli	- FÍSICA /Toninho

Turma: 1A - SUPLETIVO

SEGUNDA	- TERÇA	- QUARTA	- QUINTA	- SEXTA
INGLÊS /Marisa	- MATEMÁTICA/Marli	- PORTUGUÊS /Ana Lúcia	- PORTUGUÊS /Ana Lúcia	- INGLÊS /Marisa
PORTUGUÊS /Ana Lúcia	- HISTÓRIA /Liliane	- MATEMÁTICA/Marli	- HISTÓRIA /Liliane	- BIOLOGIA /Ana Silva
PORTUGUÊS /Ana Lúcia	- PORTUGUÊS /Ana Lúcia	- BIOLOGIA /Ana Silva	- BIOLOGIA /Ana Silva	- MATEMÁTICA/Marli

Turma: 5A - SUPLETIVO

SEGUNDA	- TERÇA	- QUARTA	- QUINTA	- SEXTA
BIOLOGIA /Dayse	- HISTÓRIA /Liliane	- PORTUGUÊS /Lurdinha	- BIOLOGIA /Dayse	- ARTES /T. Zanin
GEOGRAFIA /Vera Carin	- PORTUGUÊS /Lurdinha	- GEOGRAFIA /Vera Carin	- PORTUGUÊS /Lurdinha	- MATEMÁTICA/Marli
MATEMÁTICA/Marli	- MATEMÁTICA/Marli	- MATEMÁTICA/Marli	- HISTÓRIA /Liliane	- PORTUGUÊS /Lurdinha

---

Horário dos Professores - NOTURNO

```
======================
Professor: ANA LÚCIA
======================

2C - __ - __ - __ - __
2A - __ - 1A - 1A - 2A
1A - 2C - __ - 2C - 1D
1A - 1A - 1D - 1D - 2A
1D - 1D - 2C - __ - 2C

======================
Professor: ANA SILVA
======================

__ - __ - __ - 3C - __
__ - __ - 3D - 2C - 2C
1D - 1D - 1D - 3D - 1A
2C - 3D - 1A - 1A - __
3C - 3C - __ - __ - __

======================
Professor: NARA
======================

__ - __ - 3D - __ - __
__ - __ - 3C - __ - __
3D - __ - __ - __ - __
3C - __ - __ - __ - __
__ - __ - __ - __ - __

======================
Professor: VERA CARIN
======================

__ - __ - __ - __ - 2C
2C - __ - __ - 2B - 2B
5A - __ - 5A - 1D - __
1D - __ - 2A - 2A - __
__ - __ - __ - __ - __

======================
Professor: LURDINHA
======================

__ - __ - __ - __ - __
__ - __ - 5A - __ - __
__ - 5A - 2B - 5A - 2B
__ - 2B - __ - __ - 5A
__ - __ - __ - __ - __

======================
Professor: MARLI
======================

3C - 3C - 1D - __ - 3D
1D - 1A - 2A - __ - 3C
2B - 2A - 1A - __ - 5A
5A - 5A - 5A - 2B - 1A
3D - 3D - __ - __ - 1D

======================
Professor: GIORGIA
======================

1D - 1D - 3C - __ - __
2B - 2C - 1D - 3D - __
2A - 2B - 2A - 2A - __
3D - 3C - 2B - 3C - __
2C - __ - 3D - 2C - __

======================
Professor: IVOLINA
======================

__ - 3D - __ - 3D - __
3D - 3C - __ - 3C - __
3C - __ - 3D - __ - __
__ - __ - 3C - __ - 3C
__ - __ - __ - __ - 3D

======================
Professor: LILIANE
======================

__ - __ - __ - __ - __
__ - 5A - __ - __ - __
__ - 1A - __ - 1A - __
__ - 1D - __ - 5A - __
__ - __ - __ - 1D - __

======================
Professor: DAYSE
======================

__ - __ - __ - __ - __
5A - __ - __ - 5A - __
__ - __ - __ - 2B - __
2B - __ - __ - __ - __
__ - __ - __ - __ - __

======================
Professor: MARLENE
======================

__ - __ - __ - __ - __
__ - __ - __ - __ - __
__ - __ - 2C - __ - __
__ - __ - 3D - 3D - 3D
__ - __ - 3C - 3C - 3C

======================
Professor: TONINHO
======================

3D - __ - __ - 1D - __
3C - 2B - 2B - 2A - 1D
__ - 3D - 3C - __ - 2A
__ - 2A - 2C - __ - 2B
__ - 2C - 1D - __ - __

======================
Professor: ÁDRIA
======================

__ - __ - __ - 2C - __
__ - 1D - __ - __ - __
__ - __ - __ - __ - __
__ - 2C - __ - __ - 1D
__ - __ - __ - __ - __

======================
Professor: EROTILDES
======================

__ - 2C - __ - __ - __
__ - __ - 2C - __ - __
2C - __ - __ - __ - __
__ - __ - __ - __ - 2C
__ - __ - __ - __ - __

======================
Professor: M. EMILIA
======================

__ - __ - __ - __ - __
__ - __ - __ - __ - 3D
__ - __ - __ - __ - 3C
__ - __ - __ - 3D - __
__ - __ - __ - 3C - __

======================
Professor: MARISA
======================

__ - __ - 2C - __ - 1D
1A - 3D - __ - 1D - 1A
__ - 3C - __ - 3C - __
__ - __ - __ - 2C - __
__ - __ - __ - 3D - __

======================
Professor: T. ZANIN
======================

__ - __ - __ - __ - 3C
__ - __ - __ - __ - 5A
__ - __ - __ - __ - 3D
__ - __ - __ - __ - __
__ - __ - __ - __ - __

======================
Professor: VÂNIA
======================

__ - __ - __ - __ - __
__ - 2A - __ - __ - __
__ - __ - __ - __ - __
2A - __ - __ - __ - __
__ - __ - __ - __ - __
```