

Sistema de Alocação de Horários de Cursos Universitários: Um Estudo de Caso no Departamento de Computação da Universidade Federal de Sergipe

F. Vieira & H. Macedo

Departamento de Computação, Universidade Federal de Sergipe, 49100-000, São Cristóvão, Brasil

(Recebido em 30 de julho de 2010; aceito em 04 de março de 2011)

A alocação da grade de horários em cursos universitários é um trabalho contínuo, repetitivo e de difícil solução. Trata-se do típico problema de se otimizar a distribuição das disciplinas a serem ofertadas e seus respectivos professores de forma a se respeitar variadas restrições impostas. A automatização de um sistema para este fim é um problema computacional difícil, sem solução algorítmica trivial. Problemas de otimização como este requerem soluções alternativas, como as buscas locais em um espaço de estados possíveis. Este artigo propõe uma solução baseada na técnica de Algoritmos Genéticos. Um sistema de apoio à decisão sobre a alocação de grades de horários para o Departamento de Computação da Universidade Federal de Sergipe (UFS) foi desenvolvido seguindo esta abordagem, e os resultados obtidos mostram aplicabilidade da proposta.

Palavras-chave: Alocação de horários; Algoritmos Genéticos; Sistema de apoio à decisão.

The allocation of the period schedules in university courses is a continuous job, repetitive and hard to be solved. It deals with the typical problem in optimizing the distribution of the subjects to be offered to their respective professors, in such way it may respect the varied restrictions imposed. The automation of a system for this purpose is a difficult computational issue, without plain algorithmical solvability. Optimization problems like these require alternative solutions, as the local searches in a space of possible states. This article purposes a solution based on the Genetic Algorithms technique. A support system for the decision concerning the allocation of period schedules for the Computer Science Department in the Federal University of Sergipe (UFS) was developed following this approach. Gotten results show the applicability of the proposal.

Keywords: Timetabling; Genetic Algorithms; Decision Support System.

1. INTRODUÇÃO

A geração de grade de horários é um problema antigo, enfrentado pelas instituições de ensino superior normalmente em todo início de período letivo. Chefes de departamento e coordenadores de curso dedicam horas (ou dias) de trabalho em busca de uma solução que, quase invariavelmente, não é ótima. De fato, trata-se de um problema combinatório complexo, pois o conjunto de interesses particulares da instituição, professores e alunos geram diversas condições conflitivas a serem atendidas.

A importância dessa atividade, suas dificuldades e pesquisas voltadas à busca para solução do problema remete à Gotlieb em 1963 e se estende em diversos artigos publicados em conferências e periódicos científicos [Schaerf, 1995].

Na literatura científica, o tema de trabalho é conhecido como *timetabling*. O problema de *timetabling* aplicado a sistemas de ensino consiste em agendar uma sequência de encontros (aulas, exames, bancas, palestras ou outro tipo de atividade escolar) entre instituição, professores e estudantes em um período de tempo prefixado, satisfazendo um conjunto de restrições de variados tipos [Paim e Greis, 2008]. Em boa parte das soluções propostas, técnicas de Inteligência Artificial, como as buscas locais em um espaço de estados possíveis, são utilizadas.

Existem algumas ferramentas comerciais que prometem a geração automatizada de grades de horários, entretanto, sua utilização é pouco frequente uma vez que este tipo de problema incorre em necessidades específicas de um determinado curso, em detrimento das soluções genéricas existentes.

Este artigo descreve o sistema de geração automática de grade de horários desenvolvido para atender as necessidades do Departamento de Computação (DCOMP) da Universidade Federal de Sergipe (UFS), em particular, do curso de Sistemas de Informação. O problema é formulado como uma busca local em espaço de estados e utiliza a metáfora da seleção natural e evolução para que a melhor solução seja gerada ao final do processo. Esta técnica de otimização é conhecida como Algoritmos Genéticos. A abordagem adotada permite a reprodução da solução de forma a atender restrições específicas de outros cursos e de outras instituições.

O restante do artigo está organizado da seguinte forma. Na seção 2, são apresentados e discutidos os principais tipos de problemas de *timetabling* e trabalhos relacionados ao seu estudo e solução. Uma breve explicação da *rationale* dos algoritmos genéticos também é feita nesta seção. A seção 3 descreve o método utilizado para solução do problema no DCOMP, o que inclui uma discussão dos pormenores da problemática, modelagem genética do problema e implementação da solução. Resultados obtidos são apresentados e discutidos na seção 4. Finalmente, na seção 5, são apresentadas algumas conclusões e perspectivas de trabalhos futuros.

2. ALOCAÇÃO DE GRADE DE HORÁRIOS (*TIMETABLING*)

Burke et al (2003) define o problema de *timetabling* através de quatro parâmetros: T, um conjunto finito de horários (*times*); R, um conjunto finito de recursos (*resources*); M, um conjunto de eventos (*meetings*); e C, um conjunto de restrições (*constraints*). O problema está em associar horários e recursos aos eventos, buscando satisfazer as restrições da melhor forma possível.

Normalmente, os problemas de *timetabling* são classificados de acordo com os eventos a que se propõe: área esportiva, hospitalar, educacional, entre outras. Em relação à área educacional, Schaerf (1995) classifica em três diferentes grupos: *school timetabling*, *course timetabling*, *examination timetabling*. Os dois primeiros se caracterizam pelo agendamento de um conjunto de aulas (ensino médio e cursos universitários, respectivamente), e o *examination timetabling*, para a marcação de avaliações.

A diferença entre estas formas de classificação está na quantidade do conjunto finito de determinados parâmetros. O *school timetabling* possui uma quantidade de recursos menor, pois uma turma já possui uma sala especificada e seu grupo de alunos, em sua maioria, é predeterminado, o que faz com que a complexidade seja reduzida. Em problemas de *course timetabling* existe um grande conjunto de restrições: disciplinas de uma mesma turma podem ser alocadas em salas diferentes, cada disciplina é constituída por alunos de diversos cursos, professores possuem restrições de horário por desempenharem outras atividades na instituição, cada disciplina possui conjuntos de pré-requisitos que devem ser respeitados, entre outras.

Já no caso do *examination timetabling*, as restrições são mínimas. Para cada disciplina existe uma avaliação, as mesmas não devem ocorrer em um mesmo dia e horário e a marcação de provas em dias consecutivos deve ser evitada.

Nos problemas de *timetabling* busca-se atender ao maior número de restrições possíveis e estas são de variados tipos: *edge constraints*, *ordering constraints*, *event-spread constraints*,

preset specifications and exclusions, capacity constraints, hard and soft constraint, e algumas outras que não podem ser generalizadas [Fang, 1994]. O enfoque deste trabalho entretanto está ligado as *hard constraints* e *soft constraints*.

Restrições *hard* são as que não podem ser violadas em qualquer hipótese, pois causariam a inviabilidade da grade de horário. Já as restrições *soft*, são restrições desejáveis, que devem ser preferencialmente seguidas; sua violação, no entanto, não inviabiliza a solução.

Existem algumas abordagens para modelagem do problema de *timetabling* e automatização da solução. Uma abordagem inicial assumia que cada evento tivesse exatamente um grupo de alunos associado a este, um professor nomeado, e um número de horários de livre escolha [Cooper e Kingston, 1996]. Atualmente, dentre as várias técnicas de resolução existentes, as que mais se destacam são as de algoritmos genéticos (AGs), satisfação de restrições, arrefecimento simulado, busca tabu, e algoritmos de colônias de formigas.

A ideia do desenvolvimento dos AGs partiu dos conhecimentos da biologia, através da teoria evolutiva de Darwin e dos princípios básicos da genética moderna desenvolvida por Mendel. A teoria que foi utilizada como base para o desenvolvimento desta técnica foi a de que uma população de indivíduos que se reproduzem sexualmente é resultado da mutação e recombinação dos genes “sobreviventes” das gerações anteriores [Timóteo, 2002].

Nessa técnica, diversos indivíduos diferentes são gerados aleatoriamente e somente os mais adaptados sobrevivem. O fundamento básico é criar indivíduos, avaliá-los, selecionar os mais aptos, realizar o cruzamento de seus “códigos genéticos”, e gerar novos indivíduos resultantes do processo, que possivelmente são mais adaptados.

Na figura 1 está ilustrado o ciclo de execução de um AG. Primeiramente, é gerada uma população inicial de forma aleatória. Os indivíduos são avaliados através de uma função de avaliação (*fitness*), que deve ser bem definida. Na etapa seguinte, uma condição de parada é testada, que pode ser uma quantidade fixa de gerações ou um valor médio de *fitness* para a população. Através do *fitness* de cada indivíduo é realizada a seleção para participar da etapa de reprodução, onde os indivíduos envolvidos trocam material genético (*crossover*), produzindo os seres que constituirão a próxima população. Não raramente ocorre mutações nos indivíduos criados, o que possibilita o surgimento de novas soluções. Por fim, uma nova população é composta e o processo se repete.

Os AGs tem como fundamento básico aplicar o processo de evolução natural como um paradigma de solução de problemas, a partir de sua implementação em computador [Zuben, 2000]. Em sua tese, Fang (1994) apresenta um framework para resolver problemas de *timetabling* modular em instituições educacionais através de AGs. Sua abordagem envolve três componentes: declaração das restrições do problema específico, construção de uma função de avaliação para o problema específico e a utilização de um algoritmo genético de problemas independentes para resolução do problema.

Na técnica de Satisfação de Restrições (CSP - *Constraint Satisfaction Problem*), um conjunto de variáveis e um conjunto de restrições são especificados. Cada variável tem um domínio não-vazio de valores possíveis. Cada restrição envolve algum subconjunto das variáveis e especifica as combinações de valores permitidas para aquele subconjunto. Apesar da técnica lidar diretamente com a definição do que são restrições, há uma certa limitação para aplicação de preferências a estas restrições, necessárias para modelagem das restrições *soft* do problema tratado. Bistarelli *et al* (1997) apresentam uma proposta de solução desta problemática com CSP.

Outros trabalhos, como o de Zampieri e Schaerf (2006), que utilizam Busca Tabu para resolver problemas de *examination timetabling*, o de Malim *et al* (2006), que faz uso de

algoritmos imunológicos artificiais para resolver problemas de *university timetabling*, e Eley (2006) que trata de problemas de *exam timetabling* com algoritmos de colônias de formigas, são exemplos de trabalhos. A combinação de diferentes técnicas para solução do problema também é utilizada. O trabalho de Souza et al (2002) propõe a utilização de um algoritmo evolutivo híbrido, onde a população inicial é gerada através do algoritmo GRASP, e os melhores indivíduos passam por um refinamento pelo método de Busca Tabu.

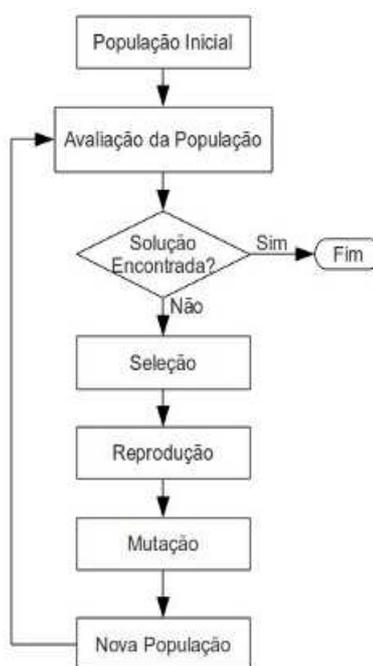


Figura 1 - Pipeline de funcionamento de um Algoritmo Genético

3. SISTEMA DE ALOCAÇÃO AUTOMÁTICA DE HORÁRIOS DO DCOMP

3.1. ALOCAÇÃO MANUAL DE HORÁRIOS

O Departamento de Computação da Universidade Federal de Sergipe (DCOMP) é composto por três cursos: Ciência da Computação, Sistemas de Informação e Engenharia da Computação. O departamento possui 28 professores, entre efetivos e substitutos, que são responsáveis por ministrar disciplinas desses 3 cursos. Atualmente, o DCOMP conta com aproximadamente 650 alunos, que estão dispostos entre os três turnos. As funções administrativas são divididas entre três coordenadores, um para cada curso, e acima deles estão o chefe e o subchefe do departamento. Toda esta estrutura é apoiada por 2 técnicos administrativos e 3 estagiários.

Um conceito presente é a ideia de créditos. Cada disciplina possui certa quantidade de créditos e cada crédito significa uma hora-aula semanal. Normalmente, um máximo de duas horas-aula de uma mesma disciplina acontece em um mesmo dia. Outro ponto importante é a existência de 3 tipos de disciplinas: as obrigatórias, as optativas e as eletivas. A primeira são as disciplinas que os alunos obrigatoriamente devem cursar para obter sua formação. As

optativas formam o currículo complementar; cada curso dispõe de uma quantidade predeterminada de créditos optativos que o aluno deve cumprir. As eletivas são disciplinas cujo conteúdo está mais distante do currículo pedagógico do curso, mas que podem ser cursadas e podem compor um limite percentual dos créditos optativos.

Atualmente, o desenvolvimento das grades de horários se dá da seguinte forma. Os coordenadores dos cursos do DCOMP elaboram sugestões de oferta que atendam as necessidades imediatas do curso que coordenam. Estas sugestões são enviadas ao chefe de departamento que analisa os pedidos dos coordenadores e tenta atender a todas as solicitações feitas. No entanto, muitas vezes, os horários propostos são conflitantes, neste momento se inicia a negociação entre os responsáveis, almejando a construção da estrutura de disciplinas obrigatórias que devem ser ofertadas. A partir deste momento as turmas começam a ser disponibilizadas no sistema acadêmico para que os alunos possam consultá-las.

Em seguida, há a criação de turmas adicionais que é decidida através do resultado da análise da retenção de alunos dos cursos e verificação de suas necessidades mais urgentes. Listas com sugestões de disciplinas optativas e horários sugestão são disponibilizadas na secretaria do departamento durante algumas semanas, visando computar interesses e intenções de matrícula por parte dos alunos. A partir de então, o chefe do departamento começa a adicionar novas turmas de forma manual, tentando atender o maior número possível de exigências, tanto por parte dos alunos, quanto por parte dos professores. Neste período a grade permanece disponibilizada para que os alunos comecem a construir seus horários. Uma última e não menos dificultosa etapa é então iniciada. De fato, quase invariavelmente, alunos percebem choques de horários e outras inconsistências. O trabalho é então retomado e a solução é continuamente refinada até que os problemas sejam completamente sanados.

3.2. ARQUITETURA DO SISTEMA

Com o conhecimento da estrutura organizacional do DCOMP e de posse do processo manual historicamente utilizado para elaboração da grade, seis casos de uso foram projetados para contemplar três diferentes visões do sistema, como pode ser observado na figura 2. A partir de profunda investigação dos pormenores que regem a construção de uma grade curricular no DCOMP, as restrições do tipo *hard* e do tipo *soft* foram especificadas e listadas a seguir.

Restrições *hard*:

- Todas as disciplinas ofertadas devem preencher o número de horas por semana estabelecidas pela estrutura curricular;
- O professor não pode lecionar em duas turmas diferentes em um mesmo dia e horário;
- Aulas de uma mesma turma não devem acontecer em um mesmo dia e horário;
- Todas as disciplinas obrigatórias do período vigente devem ser ofertadas;
- A oferta das disciplinas deve obedecer ao turno dos cursos.

Restrições *soft*:

- Todas as aulas de uma turma devem ser ofertadas, de preferência, em um mesmo horário durante a semana;
- Aulas de uma mesma turma não devem ser ofertadas em dias consecutivos, nem em um único dia;
- A preferência do professor em optar por lecionar apenas disciplinas de seu interesse particular deve ser respeitada.

O sistema foi implementado segundo o padrão arquitetural *Model-view-controller* (figura 3). este padrão provê maior maior independência para cada camada, facilitando o desenvolvimento, uma vez que alterações em uma camada não refletem nas demais. O pacote *model* contém classes responsáveis pelo acesso e gestão dos dados, formando uma camada de abstração com o SGBD. O pacote *Control* é o núcleo do sistema, que possui as regras de negócio presentes no departamento. *View* trata da interface com os usuários do sistema. O pacote *Engine* é reservado para os algoritmos de otimização, possibilitando soluções com baixo acoplamento.

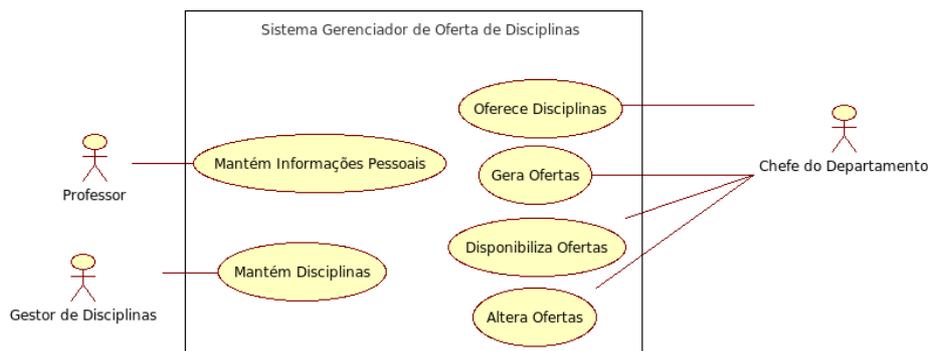


Figura 2: Casos de uso do sistema de alocação de horários do DCOMP.

Na figura 4 são ilustradas algumas classes presentes no pacote *model* e no pacote *control*.

3.3. MODELAGEM DO CROMOSSOMO

A modelagem do cromossomo é uma etapa de grande importância na formulação de um problema utilizando AGs. Uma boa definição do cromossomo minimiza a quantidade de testes necessários para avaliar um indivíduo.

Cada cromossomo é composto por todas as turmas que deverão ser ofertadas no período vigente (figura 5). Por si só, esta definição evita que a restrição *hard* de número 4 seja violada. Cada gene, representado por uma coluna, possui as informações sobre uma turma: disciplina, professor que a leciona, horários que ocorrerão as aulas e quantidade de vagas destinadas a cada curso.

O fato dos horários já estarem presentes no gene faz com que a primeira restrição *hard* seja plenamente atendida.

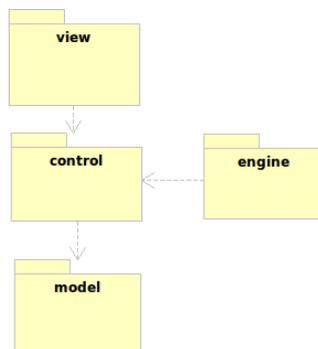


Figura 3: Organização do sistema de alocação segundo o padrão MVC. O algoritmo genético está presente no pacote *Engine*.

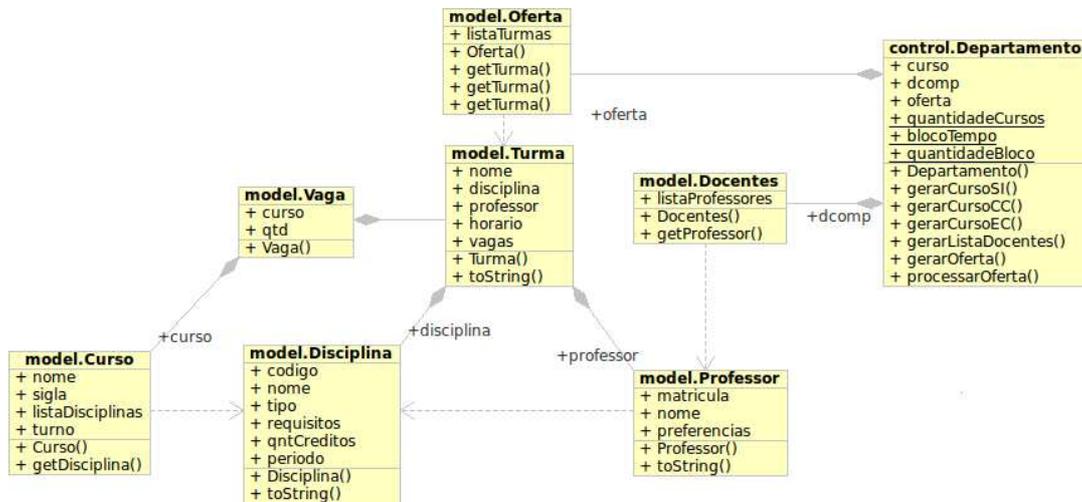


Figura 4: Classes dos pacotes model e control.

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|-----|------------|
| Disciplina | Disciplina | Disciplina | Disciplina | Disciplina | Disciplina | | Disciplina |
| Professor | Professor | Professor | Professor | Professor | Professor | ... | Professor |
| Horário[] | | Horário[] |
| Vagas[] | | Vagas[] |

Figura 5: Modelagem cromossômica de uma grade de ofertas. Cada gene armazena informações sobre a disciplina, o professor responsável, horários de aula e número de vagas.

3.4. IMPLEMENTAÇÃO DOS ALGORITMOS GENÉTICOS

Os indivíduos da população inicial foram criados com base na lista de todas as turmas que devem ser ofertadas no período vigente. Informação sobre o professor e horário da disciplina são gerados de forma aleatória em cada indivíduo, de forma a criar variabilidade genética na população. O professor é atribuído através da lista oficial de docentes do departamento e o vetor de horários é preenchido com valores inteiros de acordo com a tabela 1.

Tabela 1: Representação dos horários.

| Horário | Segunda | Terça | Quarta | Quinta | Sexta |
|----------------|---------|-------|--------|--------|-------|
| 07:00 às 09:00 | 0 | 1 | 2 | 3 | 4 |
| 09:00 às 11:00 | 5 | 6 | 7 | 8 | 9 |
| 11:00 às 13:00 | 10 | 11 | 12 | 13 | 14 |
| 13:00 às 15:00 | 15 | 16 | 17 | 18 | 19 |
| 15:00 às 17:00 | 20 | 21 | 22 | 23 | 24 |
| 17:00 às 19:00 | 25 | 26 | 27 | 28 | 29 |
| 19:00 às 21:00 | 30 | 31 | 32 | 33 | 34 |
| 21:00 às 23:00 | 35 | 36 | 37 | 38 | 39 |

Indivíduos são avaliados através da computação do somatório de valores de bonificação e penalização. O atendimento a uma determinada restrição representa um bônus associado a um peso pré-estabelecido para a mesma. O não-atendimento a uma restrição representa uma penalidade, também associada a um peso pré-estabelecido. Os valores dos pesos foram definidos considerando sua importância relativa em conversas com os especialistas no assunto: chefe do departamento e coordenadores de curso. A tabela 2 ilustra os pesos associados a cada restrição.

Na seleção dos indivíduos que farão parte da próxima geração é utilizado o método da *roleta russa*, onde todos os indivíduos podem ser selecionados para participar do cruzamento (*crossover*). No entanto, o *fitness* de cada um define a probabilidade disto acontecer. Uma vez ocorrida a seleção, é realizado o cruzamento entre os indivíduos. O método *single point crossover*, que seleciona um ponto aleatório do cromossomo para fazer o corte e trocar os materiais genéticos, foi escolhido (figura 6).

Tabela 2: Bonificações e Penalizações utilizadas

| Restrição | Bônus/Penalidade | Peso | Multiplicador |
|---|------------------|------|---|
| Disciplina atende horário do curso | Bônus | 3 | 1 * aula no horário |
| Evitar choque entre aulas de mesma turma | Bônus | 5 | 1 * turma sem choque |
| Professor dar aula em dois lugares ao mesmo tempo | Penalidade | 3 | 1 * choques encontrados |
| Aulas de uma turma devem ser ofertadas no mesmo horário | Bônus | 2 | 1 * horários que atendem a restrição |
| Aulas de uma turma não devem ser ofertadas em dias consecutivos, nem no mesmo dia | Penalidade | 2 | 1 * horários que não atendem a restrição |
| Professor optar por disciplinas de seu interesse | Bônus | 5 | 1 * disciplinas do interesse do professor que ele leciona |

4. EXPERIMENTOS E RESULTADOS

Para a realização dos experimentos com o sistema foram gerados dois grupos de testes, um com uma lista com 6 professores do departamento juntamente com as disciplinas que mais os interessavam, 12 turmas para serem ofertadas entre os 3 cursos, sendo dado maior foco para turmas do primeiro período de Sistemas de Informação. O outro grupo de teste conteve 8 professores tendo que ofertar 38 turmas do departamento de computação, dando maior ênfase as disciplinas de Ciência da Computação que é o curso que possui maior demanda por vagas.

Nos experimentos foram utilizados os seguintes parâmetros:

- População de 750 indivíduos;
- 200 gerações;
- Todos os indivíduos estão disponíveis para seleção;
- Taxa de *crossover* de 95%;
- Taxa de mutação de 5%;
- Método de seleção escolhido foi o da roleta;
- *Crossover* com apenas um ponto de corte;
- Não existe préprocessamento para eliminar indivíduos anômalos.

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|-----|------------|
| Disciplina | Disciplina | Disciplina | Disciplina | Disciplina | Disciplina | | Disciplina |
| Professor | Professor | Professor | Professor | Professor | Professor | ... | Professor |
| Horário[] | | Horário[] |
| Vagas[] | | Vagas[] |

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|-----|------------|
| Disciplina | Disciplina | Disciplina | Disciplina | Disciplina | Disciplina | | Disciplina |
| Professor | Professor | Professor | Professor | Professor | Professor | ... | Professor |
| Horário[] | | Horário[] |
| Vagas[] | | Vagas[] |

Figura 6: Crossover de um único ponto.

Com essas configurações foram realizados um conjunto de testes para perceber a convergência do algoritmo, analisando a evolução do melhor indivíduo da primeira geração com o melhor da última geração nos dois grupos de teste. Os dados são mostrados na tabela 3. É importante analisar que quanto maior o número de turmas a ser ofertada maior o ganho entre a primeira e a última geração. Isto ocorre porque há uma redução de recursos de tempo ocasionado pela quantidade de turmas que devem ser ofertadas, com isto a inserção ao acaso das turmas viola as restrições facilmente. Por isso a utilização de algoritmos de otimização trazem bons resultados.

Os gráficos da figura 7 e da figura 8 ilustram a curva de evolução do melhor e o pior indivíduo de cada geração. A convergência mais intensa ocorre até a quadragésima geração no grupo de teste 1 e até a sexagésima no grupo de teste 2. Isto mostra que quanto mais turmas, mais gerações serão necessárias para que a solução comece a convergir. A partir daí, o avanço é gradual e pouco perceptível. Outro fato interessante a se observar é que boas soluções são rapidamente aproveitadas pelo ambiente, já que não existe em qualquer momento no gráfico piora nos melhores indivíduos. Também vê-se uma certa oscilação nos piores indivíduos, mostrando que existe variação genética e possibilidade de evolução na população.

Tabela 3: Análise de convergência. Os valores representam o valor do fitness medido para o melhor indivíduo da respectiva geração.

| Melhor indivíduo da população inicial – Grupo de Teste 1 | Melhor indivíduo da última geração – Grupo de Teste 1 | Ganho (%) entre as gerações | Melhor indivíduo da população inicial – Grupo de Teste 2 | Melhor indivíduo da última geração – Grupo de Teste 2 | Ganho (%) entre as gerações |
|--|---|-----------------------------|--|---|-----------------------------|
| 159 | 217 | 36,47% | 302 | 501 | 65,89% |
| 154 | 219 | 42,20% | 300 | 494 | 64,66% |
| 158 | 221 | 39,87% | 307 | 496 | 61,56% |
| 159 | 223 | 40,25% | 313 | 485 | 54,95% |
| 159 | 221 | 38,99% | 300 | 490 | 63,33% |
| 159 | 213 | 33,96% | 307 | 495 | 61,23% |

Por último, foi analisado se a melhor solução encontrada que atende as restrições estabelecidas pelo departamento. Os dados encontrados estão dispostos na tabela 4 para o grupo de teste 1 e na tabela 5 para o grupo de teste 2. À primeira vista, observa-se que os resultados conseguidos foram positivos. Não houve oferta de disciplinas em turnos diferentes ao do curso, nem aulas de uma mesma turma no mesmo dia e horário. Não foi encontrado nenhuma alocação de um professor em turmas diferentes no mesmo dia e horário. Já em relação as restrições *soft*, nem todas foram plenamente atendidas. No grupo de teste 1: duas turmas não conseguiram se adequar à restrição que diz que uma turma deve ser ofertada no mesmo horário de dias diferentes. Outras duas turmas também não conseguiram evitar ser ofertadas em dias consecutivos, nem no mesmo dia. E por último, metade das turmas tiveram professores que lecionavam disciplinas que os interessavam. No grupo de teste 2 apresentou resultados bem parecidos se levarmos em consideração o aumento das ofertas: oito turmas não conseguiram se adequar à restrição de que uma turma deve ser ofertada no mesmo horário de dias diferentes. Outras oito turmas também não conseguiram evitar ser ofertadas em dias consecutivos, nem no mesmo dia. E por último, vinte duas turmas tiveram professores que estavam interessados em lecionar aquelas disciplinas.

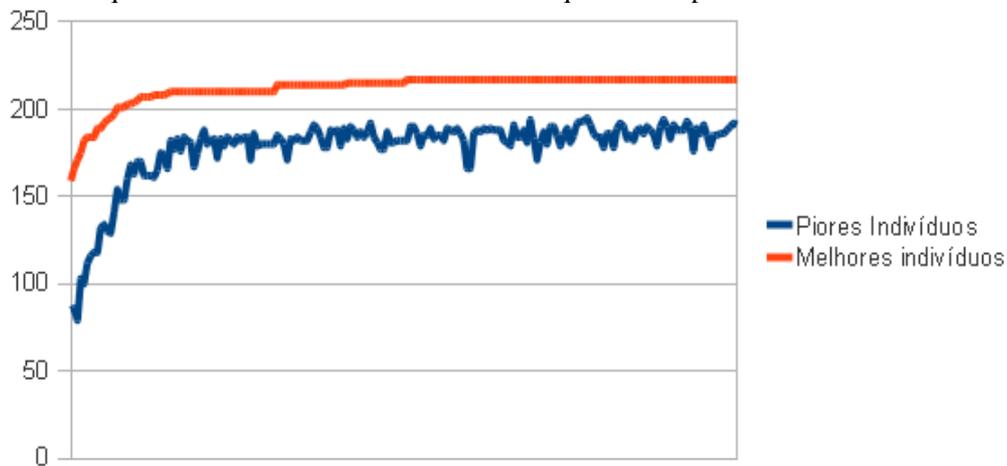


Figura 7: Diferença entre o melhor e o pior indivíduo em cada geração – Grupo de teste 1



Figura 8. Diferença entre o melhor e o pior indivíduo em cada geração – Grupo de teste 2

Tabela 4: Demonstrativo de ocorrências das restrições – Grupo de teste 1

| Restrição | Valores atingidos | Valores possíveis |
|---|-------------------|-------------------|
| Disciplina atende horário do curso | 35 | 35 |
| Evitar choque entre aulas de mesma turma | 12 | 12 |
| Professor ministrar aula em dois lugares ao mesmo tempo | 0 | 35 |
| Aulas de uma turma devem ser ofertadas no mesmo horário de dias diferentes | 14 | 16 |
| Aulas de uma turma não devem ser ofertadas em dias consecutivos, nem no mesmo dia | 2 | 16 |
| Turmas que possuíam professores que se interessavam pela disciplina | 6 | 12 |

Tabela 5: Demonstrativo de ocorrências das restrições – Grupo de teste 2

| Restrição | Valores atingidos | Valores possíveis |
|---|-------------------|-------------------|
| Disciplina atende horário do curso | 63 | 63 |
| Evitar choque entre aulas de mesma turma | 31 | 31 |
| Professor ministrar aula em dois lugares ao mesmo tempo | 0 | 63 |
| Aulas de uma turma devem ser ofertadas no mesmo horário de dias diferentes | 30 | 38 |
| Aulas de uma turma não devem ser ofertadas em dias consecutivos, nem no mesmo dia | 8 | 38 |
| Turmas que possuíam professores que se interessavam pela disciplina | 22 | 31 |

Estes resultados mostram que a solução é viável, entretanto a implantação no DCOMP se dará durante alguns períodos, o software será utilizado pela chefia como apoio ao método manual. Apenas após sólida comprovação de sua eficácia, este poderá vir a ser utilizado de forma oficial em detrimento ao método manual conservador.

5. CONCLUSÃO

Neste artigo foi proposta uma solução baseada em algoritmos genéticos. Este trabalho teve o intuito de apresentar a estrutura de um sistema de alocação de grade de horários, que utiliza algoritmos genéticos como técnica para resolução do problema, tendo como estudo de caso o Departamento de Computação da UFS. Apesar de bem estudado na literatura científica, este tipo de problema requer por vezes uma modelagem específica para cada estudo de caso.

Os experimentos realizados e as análises feitas deram validade a modelagem apresentada neste trabalho, mostrando a evolução da solução ao avanço das gerações. Estes resultados demonstram que é possível a criação de uma grade de horários automatizada útil e consistente para o DCOMP, reduzindo o tempo de criação da grade de horários inicial e o número de reclamações por motivo de choque de disciplinas.

Em trabalhos futuros é interessante a utilização de outras técnicas de otimização, comparando-as aos algoritmos genéticos. Também é possível a realização de análises em outros departamentos de forma a tentar generalizar uma solução para uma mesma instituição. Outra linha de trabalho é a criação de um módulo para análise da retenção de alunos, servindo como entrada de dados para o sistema.

-
1. BISTARELLI, S.; MONTANARI, U.; ROSSI, F. Semiring-based constraint satisfaction and optimization. *Journal of the ACM*, v. 44, n. 2, p. 201-236. 1997.
 2. BURKE, E.; WERRA, D. D.; KINGSTON, J. *Applications to timetabling. Handbook of Graph Theory*. p.445-474. CRC Press.
 3. COOPER, T. B.; KINGSTON, J. H. *The Complexity of Timetable Construction Problems*. , p. 283-295, 1996.
 4. FANG, H. Genetic Algorithms in Timetabling and Scheduling. 1994. Tese - Departament of Artificial Intelligence, University of Edinburgh, 1994.
 5. MALIM, M. R.; KHADER, A. T., MUSTAFA, A. *Artificial Immune Algorithms for University Timetabling*. Practice and Theory of Automated Timetabling. p. 234-245. 2006.
 6. PAIM, A. D.; GREIS, I. C. Abordagens para elaboração automatizada de tabela de horários acadêmicos., 2008.
 7. RUSSEL, S.; NORVIG, P. *Artificial Intelligence, A Modern Approach*. 2 ed. Prentice Hall, 2003.
 8. SCHAERF, A. A survey of automated timetabling A. Schaerf. , 1995.
 9. SOUZA, M. J.; COSTA, F. P.; GUIMARÃES, I. F. Um Algoritmo Evolutivo Híbrido para o Problema de Programação de Horários em Escolas. *Computer*, 2002.
 10. TIMÓTEO, G. T. Desenvolvimento de um algoritmo genético para a resolução de timetabling. , 2002.
 11. ZAMPIERI, A., SCHAERF, A. *Modelling and Solving the Italian Examination Timetabling Problem using Tabu Search*. Practice and Theory of Automated Timetabling. p. 487-491, 2006.