

JOÃO DAHER NETO

RESOLUÇÃO DO TIMETABLING UNIVERSITÁRIO UTILIZANDO MÉTODOS INTELIGENTES

LAVRAS - MG 2011

JOÃO DAHER NETO

RESOLUÇÃO DO TIMETABLING UNIVERSITÁRIO UTILIZANDO MÉTODOS INTELIGENTES

Monografia apresentada ao Colegiado do Curso de Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

Orientador

Prof. Dr. Joaquim Quinteiro Uchôa

LAVRAS - MG 2011

JOÃO DAHER NETO

RESOLUÇÃO DO TIMETABLING UNIVERSITÁRIO UTILIZANDO MÉTODOS INTELIGENTES

Monografia apresentada ao Colegiado do Curso de Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 9 de Novembro de 2011

Prof. Dr. José Monserrat Neto

Prof. Dr. Tales Heimfarth

Prof. Dr. Joaquim Quinteiro Uchôa

Orientador

LAVRAS - MG

2011



AGRADECIMENTOS

A Deus, por ter aberto todas as portas e mostrado o melhor caminho. Ao meu pai, Francisco, por abrir mão de todos os feriados e fins de semana para oferecer o suporte para minha graduação. A minha mãe, Rosangela, por sempre ser a cabeça inteligente ao administrar os inúmeros problemas e contratempos. Aos meus irmãos, Bruna e Thiago, por darem apoio e palavra amiga desde passei a morar longe de casa. A minha avó, Rosária, por dar todo o apoio mesmo sem saber até hoje o que eu realmente estudo. A minha namorada, Carol, pela companhia e paciência em todos os meus momentos dentro e fora da universidade. Ao meu colega, Hudson, por ser vital ajuda nos primeiros meses de graduação Aos demais colegas de turma, pela companhia nos estudos e na vida universitária. Aos funcionários do DRCA, DGTI e TecnoLivre, pela paciência e prestatividade no desenvolvimento da minha pesquisa.

RESUMO

A organização de um horário escolar semanal para uma instituição de ensino demanda considerável tempo e esforço por parte dos seus responsáveis até ser totalmente concluída. Não obstante, o resultado final nem sempre é satisfatório para os professores e alunos, onde, diversas vezes, obtém-se muitas aulas consecutivas de uma mesma disciplina ou professores encarregados de 2 turmas em um mesmo horário.

A partir dessa questão foram criados algoritmos computacionais para substituir o esforço mental da elaboração e propor horários que sejam capazes de conciliar professores, alunos, disciplinas e salas de aula em seus devidos horários. Este problema, tão comum em escolas e universidades, ganhou a definição de *timetabling* e vem sendo alvo das mais variadas meta-heurísticas para otimizar um bom resultado.

Este trabalho virá com a proposta do uso de um algoritmo inspirado no sistema imune humano, o *Clonalg*, para propor uma solução para as alocações temporal (horários) e física (salas de aula e laboratórios) para a Universidade Federal de Lavras. Para tal, tomou-se o cuidado de criar uma estrutura de dados favorável à execução do algoritmo em tempo hábil, visto que estar-se-á manipulando aproximadamente 3000 disciplinas distribuídas em mais de 200 lugares sob 15 tipos de restrições distintas.

À primeira vista não se pretende chegar a uma proposta ótima, o objetivo principal consiste em obter um horário inicial que seja passível de leves alterações manuais para se adequar à realidade da instituição, visto que até hoje este trabalho é feito completamente de forma manual.

Palavras-chave: Algoritmo Imunoinspirado; Course Timetabling; Clonalg.

ABSTRACT

Creating a weekly school timetable for an education institution demands significant time and effort by its empolyees until it's completely done. Though the result doesn't satisfy teachers and students, because it often creates successive classes of a single subject, or even teachers find themselves in charge of 2 classes at the same time.

So, computacional algorithms have been created in order to replace the mental effort on creating and offering timetables wich are able to arrange teachers, students, classes and classrooms in their due time. This problem, so usual in schools and colleges, has got the definition *timetabling* and it has been target of various meta-heuristics to optimize a plausible result.

This research will come with the proposal of using an algorithm inpired by the human immune system, the *Clonalg*, in order to offer a solution to alocate time (*timetabling*) and rooms (classroom and laboratories) in the Federal University of Lavras. Thus, a data structure has been carrefully created in order to allow the algorithm to execute under a reasonable time, since it will manipulate about 3000 disciplines distributed along more than 200 rooms under 15 different sorts of constraints.

At first sight, it is not intended to reach an optimal solution, the goal is bounded by an initial timetable, wich is liable to smooth manual changes in order to suit the reality of the instituion, since even today this job is done completly by hand.

Keywords: Immune Inspired Algorithm; Course Timetabling; Clonalg.

LISTA DE FIGURAS

Figura 1	Funcionamento de um Algoritmo Genético Típico 1		
Figura 2	Simplificação de um Algoritmo Inspirado no Sistema Imune		
Figura 3	Exemplo de Função com Máximo Local e Máximo Global		
Figura 4	Diagrama Simplificado de Classes da Estrutura Universitária		
Figura 5	Relação Entre Classes do Sistema Imune e Estrutura Universitária	37	
Figura 6	Probabilidade cada <i>Slot</i> de Tempo ser Sorteado	40	
Figura 7	Gráfico da Função de Fitness.	41	
Figura 8	Diferença Final e Inicial de Restrições Não Atendidas	57	
Figura 9	Impacto do Uso de Heurísticas no Ganho do Fitness	57	
Figura 10	Impacto do Uso de Heurísticas nas Restrições Não Atendidas		
Figura 11	Relação entre Fatores de Hipermutação e Fitness	59	
Figura 12	Relação entre Tamanho da População e Ganho de Fitness	59	
Figura 13	Relação entre Quantidade de Clones e Ganho de Fitness	60	
Figura 14	Relação entre Diversificação e Ganho de Fitness	61	
Figura 15	Relação entre Ranking e Ganho de <i>Fitness</i>	62	
Figura 16	Histórico de <i>fitness</i> das Células $\beta \in \omega$	63	

LISTA DE TABELAS

Tabela 1	Seção da Grade Curricular do 6º Módulo de Ciência da Computação	23
Tabela 2	Classificação de Restrições.	26
Tabela 3	Configuração de Pesos.	52
Tabela 4	Configurações de Parâmetros	53
Tabela 5	Ganho de Fitness e Número de Gerações	55
Tabela 6	Quantidade de Restrições Não Atendidas	56
Tabela 7	Configuração Ideal de Parâmetros	62
Tabela 8	Resultados da Simulação Intensa	63

SUMÁRIO

1	Introdução	12
2	O Timetabling	14
2.1	Comentários Iniciais	14
2.2	Geração de Horários Escolares	15
2.3	Abordagens para Resolução do Timetabling	16
2.3.1	Algoritmos Genéticos	17
2.3.2	Algoritmos Imunoinspirados	18
2.3.3	Subida de Encosta	20
2.3.4	Têmpera Simulada	21
3	O Timetabling na Universidade Federal de Lavras	22
3.1	Particularidades da Universidade Federal de Lavras	22
3.2	Processo Atual de Criação do Horário	24
3.3	Restrições	25
3.3.1	Choque de Horário	26
3.3.2	Choque de Professor	27
3.3.3	Bloco Excessivo.	27
3.3.4	Distribuição Semanal	27
3.3.5	Horários Esparsos	28
3.3.6	Sobrecarga de Professor	28
3.3.7	Revesamento de Horários	28
3.3.8	Lotação	29
3.3.9	Choque de Local	29
3.3.10	Locais Reservados	29
3 3 11	Locais Espaciais	30

3.3.12	Isolamento	30
3.3.13	Aulas Práticas	30
3.3.14	Turno	31
3.3.15	Proximidade	31
3.4	Comentários Finais	31
4	Resolução do Timetabling da UFLA	32
4.1	Fase Inicial.	32
4.1.1	Coleta de Informações	33
4.1.2	Elaboração da Estrutura	33
4.1.3	Relação com o Sistema Imune	36
4.2	Processo de Resolução	38
4.2.1	Geração da População Inicial	39
4.2.2	Avaliação das Células	39
4.2.3	Critério de Parada	42
4.2.4	Seleção Clonal	42
4.2.5	Hipermutação das Células	43
4.2.6	Heurísticas	44
4.2.7	Fatores de Hipermutação	46
4.2.8	Diversificação das Células	47
4.3	Simulação	48
4.3.1	Escopo da Simulação	48
4.3.2	Parâmetros de Simulação	49
4.4	Análise	54
4.4.1	Restrições Não Atendidas	55
4.4.2	Heurísticas	57
4.4.3	Fator de Hipermutação	58

4.4.4	Tamanho da População	59
4.4.5	Máximo de Clones	60
4.4.6	Diversificação	60
4.4.7	Ranking	61
4.5	Simulação Intensa	62
4.6	Comentários Finais	64
5	Conclusão	65
A	ANEXO: Interface web do Sistema - Início	69
В	ANEXO: Interface web do Sistema - Horário Individual	70
C	ANEXO: Interface web do Sistema - Horário de Oferta	71
D	ANEXO: Interface web do Sistema - Relatório de Erros	72

1 Introdução

Toda instituição de ensino exige a organização de um horário para distribuir seus professores, disciplinas e salas de aulas de forma a evitar diversos conflitos, tais como aulas simultâneas (alunos que têm aulas de 2 disciplinas no mesmo horário), choque de professores (professores são encarregados de 2 aulas ao mesmo tempo) e incompatibilidade de local com a disciplina (locais que não suportam o número de alunos). Tais restrições serão mostradas mais detalhadamente na Seção 2.2 deste trabalho.

Devido à complexidade desta tarefa, as escolas e universidades iniciam este trabalho com grande antecedência, definindo um funcionário responsável ou uma equipe, dependendo da proporção da instituição. No caso da Universidade Federal de Lavras, a qual foi objeto de estudo neste presente trabalho, a alocação de horários é feita por duas pessoas e chega a levar uma média de 4 meses até que seja obtido um horário inicial. No caso da Universidade, apenas parte desta tarefa é realizada manualmente, pois tem-se o auxílio do Sistema Integrado de Gestão¹, mas essa não é uma facilidade encontrada na maioria das instituições de ensino, as quais são obrigadas a realizar esta tarefa contando apenas com a capacidade de raciocínio dos encarregados.

Entretanto, mesmo depois do horário ser aplicado, o mesmo quase sempre sofre correções de erros que foram imperceptíveis aos olhos humanos, causando transtornos aos discentes e docentes prejudicados por horários incompatíveis com a realidade.

¹SIG, o Sistema Integrado de Gestão, implantado em dezembro/2010, informatizou grande parte das informações inerentes à organização interna da Universidade, tais como alunos matriculados, espaços disponíveis e horários pessoais.

A fim de abordar a problemática envolvida na resolução do *timetabling*, este trabalho está dividido em 5 capítulos, os quais tem suas abordagens resumidas a seguir:

- **Capítulo 2:** introduz os conceitos referentes à organização de um horário escolar, apresenta as variações do problema do *timetabling*, e mostra diferentes abordagens inteligentes utilizadas como tentativa de otimizar uma solução para o problema .
- **Capítulo 3:** descreve o processo de escalonamento de horários é realizado na Universidade Federal de Lavras, com todas as suas particularidades e restrições.
- **Capítulo 4:** mostra como o problema foi implementado, sua estrutura e relação com a abordagem inteligente, e como foi organizado o processo de simulação do sistema; além de realizar uma análise crítica dos resultados obtidos.
- **Capítulo 5:** aborda o que foi possível inferir sobre o desenvolvimento do sistema, os resultados do trabalho e quais melhorias podem feitas para tornar o sistema ainda mais eficiente.

2 O Timetabling

2.1 Comentários Iniciais

A alocação de horários é um problema do cotidiano da vida escolar que carece de bons algoritmos para gerar algum resultado plausível, economizando grande parte do esforço por parte dos funcionários encarregados de elaborar o horário escolar.

Considere que, de acordo com o Teorema 7.20 de (SIPSER, 2006), seja P o conjunto de problemas de decisão que podem ser resolvidos por uma máquina determinística em tempo polinomial; e seja NP a classe de problemas que podem ser verificadas em um tempo de execução polinomial, embora só possam ser resolvidos em tempo de execução polinomial por máquinas não-determinísticas. Pode-se concluir que, a não ser que P=NP (o que não é o considerado como verdade até então de acordo com (MR, 1979)), não há como determinar a melhor solução dentro do universo de possíveis soluções em tempo de execução polinomial para o problema de escalonamento de horários. Nesse contexto, o *timetabling* pertence à classe de problemas NP-Completo, segundo (EVEN, 1976), que representa o conjunto de problemas nos quais qualquer problema NP pode ser reduzido em tempo de execução polinomial, mas cuja solução pode ser verificada em tempo polinomial.

Portanto, obter um horário ótimo² para uma instituição é, teoricamente, inalcançável atualmente devido à limitação do modelo computacional binário. Isso se dá pelo fato do *timetabling* ser um problema NP-Completo, onde a gama de

²O termo "solução ótima", no contexto de Programação Matemática, é uma solução factível que fornece o menor valor à função objetivo, como citado em (SANTOS, 2010).

possibilidades de horários se torna absurdamente grande mesmo para pequenas instituições.

Como mostrado no trabalho de (TIMOTEO, 2002), uma escola de ensino fundamental com apenas 7 disciplinas e menos de 25 professores permite $8,1\times10^{100}$ arranjos diferentes no horário. Se nesse caso fosse aplicado um algoritmo de força bruta (método que testa uma solução e compara com as geradas anteriormente) para resolver este problema, o tempo de processamento (considerando que cada máquina execute 1 teste em 1ms) seria de $2,6\times10^{90}$ anos até que fosse obtido o melhor horário para a escola.

Nota-se, então, que a busca da melhor solução é inviável em tempo hábil, sendo possível apenas a aproximação de resultados, fator que justifica a popularização das adoções de técnicas heurísticas em problemas com essas características. Além disso, no caso de uma universidade, não é sequer necessário obter a solução ótima, basta ter uma solução possível de ser aplicada na realidade da instituição.

2.2 Geração de Horários Escolares

O problema de geração de horários escolares, denominado *Timetabling*, consiste em criar encontros entre diversas variáveis, podendo estas ser professores, alunos e salas de aula, em determinados espaços de tempo. Além disso, considerase algumas restrições impostas para criar tais encontros, as quais são definidas na Seção 3.3.

Há três variedades do *timetabling* para classificar³ os problemas de alocação de horário segundo suas características gerais. De acordo com (SCHAERF, 1999), essas classes de problemas são:

School *Timetabling*: trata-se da variação do *timetabling* utilizada na maioria das escolas, onde se precisa criar um conjunto semanal de aula onde alunos e professores não tenham mais de uma aula ao mesmo tempo.

Course *Timetabling*: trata-se da variação do *timetabling* utilizada em universidades, que consiste em criar um arranjo semestral de aulas, considerando alunos de cursos diferentes.

Examination *Timetabling*: trata-se da variação utilizada para agendar avaliações (exames) para alunos, evitando que um aluno tenha mais de um exame no mesmo horário e, além disso, favorecer uma maior distribuição temporal destes exames.

2.3 Abordagens para Resolução do Timetabling

Uma vez que o problema do *timetabling* é um problema NP-Completo, há diversas abordagens diferentes que podem ser utilizadas para tentar obter uma solução otimizada para a alocação de horários. A seguir, algumas técnicas serão citadas, a fim de mostrar o potencial que cada uma oferece.

³Embora sejam definidas classes para os problemas, os mesmos não são restritos a estas classes. Todos eles podem conter suas particularidades.

2.3.1 Algoritmos Genéticos

Algoritmo Genético é uma técnica de busca heurística que pertence a uma grande classe de algoritmos evolucionários, e é inspirada na evolução natural, como citado em (LIMA, 2008). Esta técnica particularmente baseia-se na estrutura genética e seu comportamento evolucionário para buscar a solução para algum problema de otimização⁴, devido ao grande número de restrições envolvidas ou espaço de busca amplo.

Como mostrado por (GOLDBERG, 1989), os algoritmos genéticos diferem-se de métodos tradicionais⁵ por apresentarem um paralelismo implícito, onde diversos indivíduos (denominados *genes*, no contexto dos AGs) são avaliados a fim de se obter uma solução aproximada para o problema.

O funcionamento do algoritmo genético, ilustrado na Figura 1, consiste em gerar uma população inicial de cromossomos, onde cada cromossomo representa uma possível solução para o problema. A cada iteração, os indivíduos são avaliados, e selecionados aos pares (através de diversas maneiras distintas) para o cruzamento⁶ e, finalmente, o novo indivíduo sofre uma mutação de seus genes. Desse modo, novos cromossomos distintos são criados a cada nova geração, permitindo evoluir a população até se obter um cromossomo que tenha uma solução aceitável para o problema.

Esta técnica pode ser aplicada no contexto do *timetabling*, para buscar uma horário otimizado para a instituição de ensino. Atualmente é a técnica mais utili-

⁴Problema de otimização é uma classe de problemas na qual a busca exaustiva, conhecido como força-bruta, mostra-se incapaz de obter a solução de forma eficiente.

⁵Métodos tradicionais, ou exatos, são modelos determinísticos para resolver o problema, totalmente voltados para um modelo matemático, operando um pontos discretos do problema.

⁶Cruzamento, no contexto de AGs, consiste na troca de informações entre indivíduos para gerar um novo indivíduo.

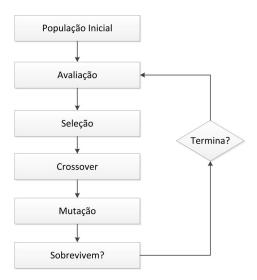


Figura 1: Funcionamento de um Algoritmo Genético Típico

zada para se resolver o problema, pois já se mostrau promissora quando aplicada na variação *School Timetabling*, como verificado nos resultados de (TIMOTEO, 2002).

2.3.2 Algoritmos Imunoinspirados

Algoritmos inspirados no sistema imune são uma técnica de computação evolutiva que se baseia no comportamento das células do sistema imune humano. Quando em contato com antígenos, o sistema evolui a fim de buscar uma solução para o problema. Diferentemente dos algoritmos genéticos, os imunoinspirados não misturam características das células, apenas fazem a mutação e clonagem individualmente das mesmas.

Um dos algoritmos imune mais conhecidos é o *CLONALG*, proposto por (SILVA, 2001), e será o algoritmo utilizado no sistema desenvolvido neste trabalho. De acordo com (UCHOA, 2009), a execução deste algoritmo dá-se da seguinte

forma: inicialmente uma população de células imunes é criada. Um treinamento é feito com estas células a fim de prepará-las para reconhecimento de antígenos. Depois, um novo antígeno é apresentado a cada célula do organismo e uma taxa de afinidade (denominada *bind*) é calculada para representar a qualidade da célula⁷. As células com pior capacidade de reconhecimento são eliminadas ou sofrem uma maior mutação que as demais.



Figura 2: Simplificação de um Algoritmo Inspirado no Sistema Imune

Dessa forma, o sistema evolui até atingir seu objetivo, que pode ser o de verificar uma sequência de entradas, muito utilizado na área de Segurança Compu-

⁷O termo *qualidade* representa quão capaz a célula é de identificar o antígeno.

tacional para Detecção de Intrusos (como mostrado na Figura 2); ou até obter-se uma célula que tenha uma qualidade determinada, utilizado para a resolução do *timetabling* por exemplo.

2.3.3 Subida de Encosta

A técnica de Subida de Encosta (ou *Hill Climbing*) é um método de busca heurística que consiste em gerar uma solução inicial e, incrementalmente, alterar elementos da mesma a fim de obter uma solução melhor. Cada alteração representa um novo estado, e o algoritmo verifica se tal alteração trará algum benefício para a solução atual: se a nova solução for melhor que a atual, troca-se para o novo estado; caso contrário, não troca de estado e toma-se a solução atual como solução do problema.

Como se trata de um algoritmo de busca local - ou seja, não se analisa "o todo", mas somente a situação atual - a Subida de Encosta está suscetível a não encontrar a melhor solução caso o problema em questão tenha máximos locais. Nesses casos, o algoritmo corre o risco de ater-se a algum máximo local, e impedir a evolução até o máximo global. Na Figura 3 pode ser vista a representação gráfica da equação de um problema que possui um máximo local, além do máximo global.

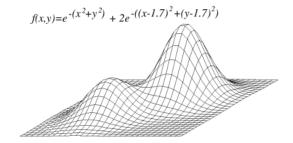


Figura 3: Exemplo de Função com Máximo Local e Máximo Global

Através das informações e resultados mostrados por (BURKE,), nota-se que este tipo de técnica de busca heurística poderia ser utilizada para resolver o problema do *timetabling*, uma vez que se pode tomar o horário criado no semestre anterior como solução inicial para o algoritmo e, a partir desta, buscar uma solução que atenda aos critérios do novo semestre.

2.3.4 Têmpera Simulada

A técnica de Têmpera Simulada, conhecida como *Simulated Annealing*, pode ser considerada um aperfeiçoamento da Subida de Encosta, citada na Subseção 2.3.3. A diferença é que a têmpera simulada permite transições para estados com resultados piores⁸ que o atual.

Dessa forma, a busca pela solução evita ater-se a máximos locais, aumentando as chances de se alcançar o máximo global do problema. Por outro lado, de acordo com (MOORE,), a elaboração de uma função de avaliação para um problema que utilize a técnica de Têmpera Simulada tem uma complexidade muito maior quando comparado à Subida de Encosta, uma vez que o algoritmo tem mais flexibilidade na transição de estados.

Até a presente data foram realizados poucos trabalhos utilizando esta abordagem, entretanto, como mostrado no trabalho de (THOMPSON, 1996), a técnica da Têmpera Simulada é utilizada para resolver com sucesso a versão mais simples do problema do *timetabling*: o *examination timetabling*.

⁸O termo *pior*, no contexto de algoritmos heurísticos, é usado para determinar uma solução que possui avaliação inferior a outra.

3 O Timetabling na Universidade Federal de Lavras

Neste trabalho, o problema de alocação de horários para a UFLA faz parte da classe de problemas *Course Timetabling*, uma vez que será necessário distribuir os alunos considerandos seus respectivos cursos. Além disso, outras diversas restrições serão avaliadas durante a busca por uma solução para o problema, as quais serão abordadas detalhadamente na seção a seguir.

3.1 Particularidades da Universidade Federal de Lavras

Para definir as particularidades do problema na Universidade Federal de Lavras, foi feito um acompanhamento dos responsáveis pela elaboração do horário na universidade, a fim de definir como o processo é feito a cada semestre, e externalizar o conhecimento tácito dos funcionários durante a resolução manual do problema.

A universidade é dividida em departamentos, e cada um deles é responsável por criar ofertas⁹ de disciplinas da área de atuação do mesmo para os diversos cursos. Essas ofertas são definidas internamente pelo chefe do departamento em conjunto com os coordenadores dos cursos relacionados. Nessa etapa define-se a carga horária de cada oferta e também já é levada em consideração a quantidade de possíveis alunos para cada disciplina. Em muitos casos é possível ofertar uma disciplina para vários cursos simultaneamente, sendo necessário agrupar várias turmas (chamado "composição de turmas").

⁹O termo "ofertas", no contexto deste trabalho, será usado para definir um conjunto de aulas de determinado assunto (disciplina) que é frequentado por um conjunto de alunos (turmas) pré-definido.

Paralelamente, a coordenação dos cursos organiza a grade curricular para seus respectivos cursos, o que consiste em definir como será o progresso de cada turma do primeiro ao último período de curso. Agrupam-se disciplinas por períodos, e criam-se pré-requisitos¹⁰ para cada disciplina, o que varia de curso para curso. Na Tabela 1, um exemplo real de uma matriz curricular constando algumas disciplinas do 6º módulo de Ciência da Computação.

Tabela 1: Seção da Grade Curricular do 6º Módulo de Ciência da Computação

Disciplina	Código	Pré-Req.	Mód.	С.Н. Т.	С.Н. Р.
Inteligência Artificial	GCC128	COM166	6	68	0
Interface Homem-Máquina	GCC121		6	68	0
Sistemas Operacionais	GCC116		6	68	0
Redes de Computadores	GCC125	COM168	6	68	0

Com as constantes reformas nas grades curriculares dos cursos, cada novo aluno é vinculado a uma determinada matriz curricular (a mais recente até então) no ato de sua matrícula.

Por fim, o DRCA¹¹ reúne tais informações citadas anteriormente e inicia o processo de alocação de horários. Deve-se ressaltar que os horários criados são válidos apenas para alunos regulares¹², os demais alunos definirão seus próprios horários do modo que seja mais conveniente, estando sujeitos à incompatibilidade (choques de horário, por exemplo) de determinadas disciplinas.

 $^{^{10}\}mathrm{S\~{a}o}$ disciplinas que precisam ter sido cursadas anteriormente para permitir a matrícula em uma determinada disciplina.

¹¹Departamento de Registro e Controle Acadêmico é um setor subordinado à Pró-Reitoria de Graduação da UFLA que se encarrega do controle do progresso dos alunos, bem como a elaboração dos horários.

¹²Discentes que não foram reprovados em nenhuma disciplina até então e, consequentemente, estão acompanhando a sequência de disciplinas propostas em sua grade curricular.

3.2 Processo Atual de Criação do Horário

Atualmente, todo o processo de definição de horários da UFLA é feito com o auxílio do SIG, que oferece desde o cálculo de estimativas de alunos em cada disciplina até a verificação de conflitos temporais e espaciais. Para elaborar o horário de um semestre, a equipe encarregada inicia o processo com uma antecedência de 3 a 4 meses do início do mesmo e, para tal, executa as seguintes tarefas para cada oferta da Universidade:

- 1º: Criar uma oferta para uma disciplina proposta por um departamento, incluindo a composição de turmas e carga horária propostas. Basicamente, é feito o cadastro das características de cada oferta no sistema.
- 2º: Definir um horário adequado para a oferta, levando em consideração todas as restrições temporais citadas anteriormente na Seção 3.3. Essa etapa é realizada baseada apenas no conhecimento tácito da equipe, devido à experiência adquirida em semestres anteriores.
- 3º: Definir um espaço para a oferta, observando atentamente o número de vagas ofertadas proposto pelo departamento responsável e comparando a estimativa de alunos informada pelo SIG. Isto é feito para garantir que a oferta da disciplina seja capaz de atender à estimativa de alunos. Além disso, considerar-se-á algumas das restrições espaciais citadas na seção seguinte.

Assim que um conflito é encontrado, seja ele na etapa um ou dois, o sistema acusa o erro, mas não oferece solução. Portanto, é feito um reajuste manual, procurando outra possibilidade de horário ou local para a oferta, novamente se baseando na tentava-e-erro. Muitas vezes há conflitos em cadeia, sendo necessário alterar uma segunda oferta para poder alocar a primeira, e assim por diante.

3.3 Restrições

Ao gerar uma proposta de horário, devem ser considerados diversos fatores limitantes (denominados *constraints* na literatura) que aplicam exigências a serem cumpridas.

Essas restrições pode ser classificadas em duas categorias de acordo com seu propósito:

Restrições Temporais: são aquelas que dizem respeito ao horário em que uma aula pode ser alocada (choques de horario, por exemplo); e

Restrições Espaciais: aquelas que tratam do local em que uma aula pode ser alocada (choques de local, por exemplo).

Também podem ser classificadas de acordo com sua importância, variando dependendo do interesse da instituição:

Restrições Obrigatórias: são aquelas que basta uma única ocorrência para tornar o horário inutilizável. São criadas com o propósito de tornar o horário funcional. Choques de horário, por exemplo, é um critério obrigatório porque é inaceitável mais de um professor lecionando na mesma sala no mesmo horário; e

Restrições Preferenciais: são restrições nas quais se deseja alcançar um número baixo de ocorrências, mas que não impedem o uso real do horário gerado. São criadas com o propósito de refinar o horário, tornando-o conveniente.

Na Tabela 2 é possível ver as classificações de restrições de acordo com seu propósito e importância¹³, as quais são detalhadas nas Subseções de 3.3.1 a 3.3.15.

Tabela 2: Classificação de Restrições

Restrição	Temporal	Espacial	
	Choque de Horário	Choque de Local	
	Choque de Professor	Além da Capacidade	
Obrigatória	Bloco Excessivo	Locais Reservados	
	Turno	Locais Especiais	
		Aulas Práticas	
	Distribuição Semanal		
Preferencial	Horários Esparsos	Isolamento	
Preferencial	Sobrecarga de Professor	Proximidade	
	Revesamento de Horários		

3.3.1 Choque de Horário

Um choque de horário ocorre quando disciplinas distintas estão sendo ofertadas em um mesmo horário, e algum fator impede a alocação simultânea das mesmas.

No caso da UFLA, como são considerados apenas os alunos regulares, só podem ocorrer choques entre disciplinas do mesmo período de um determinado curso. Por exemplo, um aluno do *i*-ésimo período do curso *X* deve ser capaz de cursar todas as disciplinas propostas para o *i*-ésimo módulo da grade curricular do curso *X*. Em outras palavras, o aluno regular não pode ter seu progresso atrasado devido a impossibilidade de se matricular em disciplinas.

 $^{^{13}}$ Neste trabalho, a importância é considerada de acordo com o interesse da Universidade Federal de Lavras, podendo variar para outras instituições.

3.3.2 Choque de Professor

Disciplinas distintas, mas ministradas por um único professor, não podem estar em horários iguais. Em hipótese alguma admite-se que um professor lecione mais de uma disciplina no mesmo horário.

Não obstante seja uma restrição obrigatória, no escalonamento de horários da UFLA esse critério é desconsiderado. A razão disto é o fato de que os professores ainda não estão definidos durante a criação de horário. Caso haja algum choque de professor, o departamento responsável solicita um rearranjo dos horários; portanto, ignorar-se-á tais critérios no sistema desenvolvido.

3.3.3 Bloco Excessivo

Disciplinas de 4 ou mais créditos não podem ser alocada em 4 ou mais horários consecutivos. Deve-se alocar em grupos de, no máximo, 3 créditos consecutivos. No caso de disciplinas de 3 créditos, fica a caráter da oferta definir se será criada uma única aula ou duas.

3.3.4 Distribuição Semanal

Aulas de disciplinas de 4 créditos ou mais devem ser distribuídas ao longo da semana, normalmente com 1 dia vago de intervalo. Desse modo, almeja-se manter o aluno diariamente na universidade e evitar que o mesmo tenha dias completamente sem aula.

3.3.5 Horários Esparsos

Conhecidos popularmente como "janelas", procura-se evitar que as aulas distintas tenham um grande espaço de tempo não ocupado entre si. Quanto mais consecutivas forem as aulas distintas, mais brevemente o aluno estará disponível para outras atividades acadêmicas.

3.3.6 Sobrecarga de Professor

Professores que lecionam em disciplinas no último horário da noite (21:00 às 22:40) tem preferência de não lecionar disciplinas no primeiro horário do dia seguinte (08:00 às 09:40). Isso porque, de acordo com o artigo 66 da CLT (Consolidação das Leis do Trabalho), os docentes devem ter um intervalo de, no mínimo, 11 horas entre a última aula de um dia e a primeira do próximo, denominado intervalo intra jornada.

Devido ao desconhecimento dos professores de cada oferta (motivo citado na Subseção 3.3.2), esse critério é desconsiderado no sistema proposto neste trabalho.

3.3.7 Revesamento de Horários

Disciplinas noturnas devem alternar entre o primeiro e segundo horários da noite, a fim de evitar que uma mesma disciplina mantenha-se sempre no último horário. Por exemplo, se uma determinada aula de uma oferta é alocada às 19:00 da segunda-feira, a próxima aula dessa oferta deve ser alocada às 21:00 de algum outro dia, e vice-versa.

3.3.8 Lotação

A capacidade do espaço a ser alocada uma oferta deve ser menor ou igual ao número de vagas da oferta da disciplina, ou seja, a sala de aula deve comportar todos os alunos matriculados na oferta.

Ao se alocar uma aula, toma-se o cuidado de deixar um espaço extra, a fim de comportar alunos não esperados, como alunos de turmas especiais¹⁴, ouvintes¹⁵ e alunos especiais¹⁶.

3.3.9 Choque de Local

Disciplinas distintas não podem ser ofertadas no mesmo local em um mesmo horário. É inaceitável que uma mesma sala seja ocupada por mais de uma oferta ao mesmo tempo.

3.3.10 Locais Reservados

Espaços físicos internos a departamentos só podem ser usados por cursos relacionados ao departamento. Por exemplo: é proibida a utilização de algum laboratório de informática do Departamento de Ciência da Computação (DCC) para aulas que não são ofertadas pelo próprio departamento.

¹⁴Aluno de turma especial é aquele matriculado em condição diferenciada que o impede de ser reprovado por falta (portanto não necessita frequentar as aulas), cabendo-lhe apenas atingir a nota necessária para aprovação através das avaliações.

¹⁵Ouvintes são alunos não matriculados, convidados, que assistem a aulas sem qualquer compromisso com frequência ou notas.

 $^{^{16}}$ Alunos especiais são estudantes matriculados em outra instituição de ensino que frequentam disciplinas isoladas na universidade.

3.3.11 Locais Especiais

Determinados espaços físicos só podem ser utilizados por disciplinas específicas.

Na UFLA, e em outras universidade onde o sistema possa ser aplicado, há casos de salas de aula projetadas para determinadas disciplinas e que não podem ser utilizadas por outras. No caso da UFLA, toma-se como exemplo a sala de aula 7 no Pavilhão 1 (PV1-07) com equipamentos para a disciplina de Desenho Técnico.

3.3.12 Isolamento

No período noturno, as aulas são alocadas preferencialmente em salas de aula próximas à área central¹⁷ da UFLA, por questões de segurança.

3.3.13 Aulas Práticas

Ofertas que tenham carga horária prática pré-definidas pelo departamento devem ter suas aulas práticas alocadas em espaços físicos destinados a estes fins, como laboratórios. As demais aulas teóricas devem ser necessariamente alocadas em salas de aula.

¹⁷Considera-se *área central* a região que envolve os pavilhões de aula próximos à Cantina Central, que são os Pavilhões de Aulas 1, 3, 5 e 6

3.3.14 Turno

As aulas devem ser alocadas em horários coerentes ao turno de todas as turmas que cursam a oferta. No caso de cursos diurno, o horário deve ser entre 07:00 e 17:40; para cursos noturno, de 19:00 às 22:40.

No sistema desenvolvido, a hora-aula foi modificada de 50 minutos para 1 hora, com propósito de facilitar a criação e manipulação do sistema.

3.3.15 Proximidade

Alunos e professores tem preferência de terem suas aulas alocadas em locais próximos ao departamento vinculado a seus cursos, a fim de evitar um grande deslocamento por parte dos envolvidos.

3.4 Comentários Finais

Neste capítulo foram abordados conceitos que envolvem o escalonamento de horários e salas de aula, bem como todas as restrições que envolvem a resolução do problema em sua variedade *course timetabling*. Além disso, foi mostrado, para cada restrição, quais serão consideradas durante o desenvolvimento deste trabalho.

4 Resolução do Timetabling da UFLA

4.1 Fase Inicial

A metodologia deste trabalho consistiu, inicialmente, em detectar o problema de alocação de horários na universidade, o que foi proposto pela desenvolvedora do atual sistema de gestão da UFLA (SIG), a TecnoLivre¹⁸.

Posteriormente, deu-se início ao levantamento de requisitos, através de constantes visitas aos setores responsáveis pela atual criação do horário para o acompanhamento de parte do processo de escalonamento manual de ofertas, mostrado na Subseção 4.1.1.

Nessa etapa foi possível elaborar a estrutura universitária compatível com todas as características relevantes ao processo (Seção 4.1.2). Essa fase foi cuidadosamente revisada e validada, pois trata-se da base de execução de todo o processo de resolução algorítmico do problema. Paralelamente, os dados reais da universidade foram solicitados aos setores responsáveis, para permitir testes e simulações mais próximas da realidade.

Em seguida, a estrutura do algoritmo inteligente foi implementada. Optouse por utilizar a linguagem PHP por ser orientada a objetos, fundamentada na web e, principalmente, por ser de fácil integração ao atual sistema de gestão da universidade, uma vez que pretende-se acoplar o sistema produzido neste trabalho.

¹⁸Cooperativa de prestação de serviços em tecnologia da informação com software livre, situada na UFLA. http://www.tecnolivre.com.br/

4.1.1 Coleta de Informações

Para permitir a realização de testes mais próximos da realidade da universidade, foi necessária a aquisição de informações referentes ao processo de escalonamento de horários na instituição. Através do apoio da TecnoLivre, os dados sobre disciplinas ofertadas e salas de aula foram fornecidos pelo DGTI¹⁹, o qual é responsável pela gestão das informações a respeito da instituição.

Também, mediante autorização do Pró Reitor de Graduação da UFLA, prof. João Chrysostomo de Resende Júnior, foi possível ter acesso às denominadas "planilhas de compactuação"²⁰, que informam as disciplinas que cada departamento é responsável por ofertar, com as turmas às quais a oferta é destinada.

4.1.2 Elaboração da Estrutura

A etapa de elaboração e criação da estrutura adequada é a parte mais crítica do projeto, já que trata de todo o controle e avaliação de informações do algoritmo.

Devido à complexidade do *course timetabling*, mais especificamente no contexto da UFLA, a estruturação do problema através da orientação a objetos (O.O.) mostra-se mais adequada, porque, como proporciona um nível alto de abstração, oferece facilidade de reutilização de código e diminuição do custo de manutenção, vantagens defendidas por (YOURDON, 1992).

A estrutura para controlar as informações da universidade relevantes à resolução do problema consiste em 9 classes extremamente inter-relacionadas, através

¹⁹Diretoria de Gestão de Tecnologia e Informação. http://www.dgti.ufla.br/site/

²⁰Planilhas criadas pela secretaria de cada departamento da UFLA, de acesso restrito à Pró Reitoria de Graduação.

de ponteiros (característico do paradigma estruturado a objetos), permitindo eficiência a consultas e verificação de relações entre os componente do sistema.

Na Figura 4 é apresentado o diagrama simplificado de classes da estrutura criada. Nesse caso, cada seta indica uma relação direta (ponteiros) entre atributos das classes. Segue-se uma breve descrição das classes apresentadas na Figura 4.

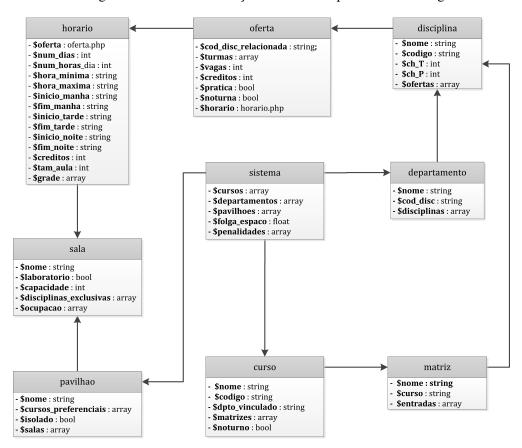


Figura 4: Diagrama Simplificado de Classes da Estrutura Universitária

departamento: contém informações sobre cada departamento da universidade, dentre elas, quais disciplinas o departamento é responsável por oferecer à universidade.

- disciplina: contém as cargas horárias da disciplina, e quais são as ofertas da disciplina.
- oferta: contém o turno da oferta (diurno/noturno), para quais turmas a oferta é destinada e qual o horário semanal da oferta especificamente.
- horário: armazena a grade semanal de uma oferta individualmente, além de ser a classe responsável por realizar as operações na grade, como criação e alteração aleatórios dos horários de aulas.
- curso: representa um determinado curso de graduação da universidade, armazenando dados como turno do curso e todas as matrizes curriculares ativas para os alunos de cada período.
- matriz: é a grade curricular de uma determinada turma²¹, contendo as disciplinas e os respectivos períodos (semestres) que são cursadas.
 - Os pré-requisitos (fortes, fracos e afins) das disciplinas são irrelevantes para o propósito do sistema, já que que os pré-requisitos apenas restringem se o aluno pode ou não cursar a disciplina, e o *timetabling*, por sua vez, necessitará alocar todas as aulas, independente de quem irá cursar a disciplina.
- pavilhão: foi criada com o propósito de agrupar as diversas salas de aula com características comuns, permitindo, também, aproximar-se mais da estrutura real da universidade. Armazena, além das salas de aula pertencentes ao pavilhão, quais cursos tem preferência²² de utilizarem salas deste pavilhão.

²¹No contexto de grade curricular, entende-se por "turma" o conjunto de alunos de um determinado curso que ingressam na universidade em um mesmo período.

²²Os critérios de preferência foram abordados nas restrições das Subseções 3.3.10 e 3.3.15.

Com propósito de simplificação, departamentos que possuem salas de aula internas (laboratórios de informática e experimentos, por exemplo) também foram considerados pavilhões.

- sala: mais fisicamente específica, um objeto desta classe guarda a capacidade da sala; se é um laboratório para aulas práticas ou uma sala de aula para aulas teóricas; e se trata-se de uma sala criada exclusivamente²³ para algumas disciplinas.
- **sistema:** essa classe representa, na realidade da instituição, o DRCA. É a classe principal, responsável por gerenciar todas as outras classes, coordenar a extração dos dados, execução e avaliação de todo o sistema.

4.1.3 Relação com o Sistema Imune

Neste trabalho optou-se por utilizar um tipo de algoritmo imune para resolver o problema do *timetabling*, o CLONALG. Essa opção foi feita por se tratar de uma abordagem ao problema pouco explorada e ser considerada, de acordo com (UCHOA, 2009), promissora em determinados problemas, quando comparada a algoritmos genéticos.

A representação do sistema universitário de horários da UFLA pode ser relacionada com os componentes do sistema imune (S.I.) da seguinte forma:

• Célula: no *timetabling*, uma célula representa uma proposta de horário, contendo todo o sistema (disciplinas, ofertas, salas, etc.) onde é feita uma sugestão de horário e local de aulas para todas as ofertas da universidade.

²³ A questão de salas exclusivas foi detalhada anteriormente no Capítulo 3.3.11.

- Antígeno: teoricamente, um antígeno deveria ser um horário completo e
 funcional, com o propósito de ser apresentado às células do sistema para
 servir de referência para determinar a afinidade, ou *fitness*, de cada uma.
 Mas como o objetivo é justamente encontrar esse horário, não há antígenos
 nesse sistema imune.
- *Bind*: representa a qualidade com a qual a célula reconhece um antígeno. Como não há antígenos, a taxa de "ligação" com o antígeno das células imune é determinada diretamente através da avaliação da qualidade das mesmas de acordo penalidades pré-determinadas (citadas na Seção 4.2.2).

A fim de representar cada um dos componentes do S.I., foram criadas duas classes, cujo diagrama de classes é mostrado na Figura 5. Nela, podemos perceber a ligação entre as duas grandes estruturas do sistema *timetabling*: o sistema imune e a estrutura universitária, através das classes celula e sistema. Segue-se também o propósito geral de cada classe.

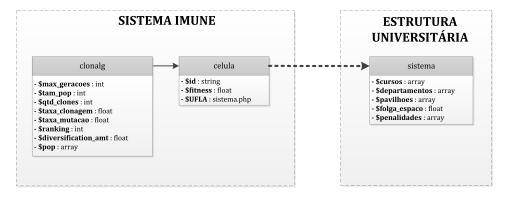


Figura 5: Relação Entre Classes do Sistema Imune e Estrutura Universitária

• célula: a menor unidade do sistema imune, essa classe representa um único indivíduo (uma célula). Além de armazenar seu próprio sistema uni-

versitário²⁴ com a proposta de horário, possui o valor do seu *fitness* e um identificador único²⁵.

• clonalg: representa o sistema imune propriamente dito, onde há a população de células e o gerenciamento de todo o processo evolutivo, desde a criação até a verificação de convergência através do critério de parada.

Na seção seguinte será mostrado detalhadamente o processo de evolução do sistema imune.

4.2 Processo de Resolução

Na Subeção 2.3.2 pôde-se ver que um algoritmo imunoinspirado pode ser divido em cinco etapas de execução:

Aquisição de dados: detalhada na Subseção 4.2.1, nesta fase os dados externos são incluídos no sistema e a população inicial é gerada.

Treinamento celular: como não há reconhecimento de antígenos (como exposto na Subseção 4.1.3), não há necessidade de realizar o treinamento das células, portanto essa fase é ignorada.

Critério de parada: a cada iteração, verifica-se se a população chegou ao critério de parada. No *timetabling* universitário, o critério de parada significa obter uma célula ideal. Esta etapa será detalhada na Subseção 4.2.3.

Avaliação: a Subseção 4.2.2 aborda a fase avaliação, que consiste em determinar a qualidade de cada uma das células.

²⁴Uma instância da classe sistema da Subseção 4.1.2.

 $^{^{25}}$ O uso do identificar único (id) em cada célula é utilizado na geração de relatórios de evolução do sistema, a fim de acompanhar o desenvolvimento das células.

Preparação para nova iteração: nessa etapa as células são modificadas a fim de permitir o progresso do sistema imune. Primeiramente é feita a seleção clonal, então as células são hipermutadas e novas células são criadas, cujos processos são mostrados nas Subseções 4.2.4, 4.2.5 e 4.2.8 respectivamente.

4.2.1 Geração da População Inicial

Inicialmente é criado um conjunto de células com horários aleatórios. Há imparcialidade na escolha dos *slots* de tempo, comprovada através do cálculo da variância entre a probabilidade de cada um dos oitenta²⁶ *slots* de tempo ser escolhido; quanto menor for este valor, mais garantia de aleatoriedade. A variância encontrada foi de aproximadamente $2,5*10^{-9}$ para uma amostra de 200.000 elementos, o que confirma a imparcialidade no sorteio.

Na Figura 6 pode-se perceber a probabilidade de cada *slot* de tempo ser sorteado. A aleatoriedade ideal seria representada no gráfico com secções de áreas iguais para cada *slot* de tempo.

4.2.2 Avaliação das Células

A fim de avaliar o cumprimento das restrições do problema (Seção 3.3), cada horário recebe uma contagem das ocorrências de descumprimento de alguma restrição.

Além da contagem de infrações às restrições, é feita uma normalização ponderada dessas ocorrências, que consiste em 2 etapas:

²⁶80 *slots* de tempo é obtido através da combinação de 5 dias na semana (segunda a sexta) com 16 possibilidades de horário cada (das 07:00 às 22:00).

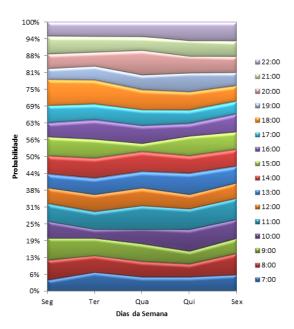


Figura 6: Probabilidade cada Slot de Tempo ser Sorteado

 Somar as ocorrências de cada restrição ponderando de acordo com o tipo de ocorrência, mostrada na Equação (1).

$$g_i(x) = \sum_{r=1}^{n} (N_{occ_r} * P_r)$$
 (1)

i: geração que se deseja verificar

n: quantidade de restrições avaliadas

 N_{occ_r} : quantidade de ocorrências de uma determinada restrição r.

 P_r : penalidade (peso) atribuído a uma determinada restrição r.

• Normalizar $g_i(x)$ através da Equação (2), a fim de gerar um número entre 0 e 1 para cada uma das células. Esse valor representa a qualidade do horário

sugerido, ou seja, o fitness da célula.

$$f_i(x) = \frac{1}{1 - g_i(x)} \tag{2}$$

i: geração que se deseja verificar

 $g_i(x)$: soma ponderada das ocorrências de restrições, obtida em (1).

Dessa forma, f(x) permite comparar células distintas, observando que, quanto mais próximo de 1, mais adequado é o horário sugerido pela célula, como pode ser visto no gráfico da Figura 7. Caso uma célula possua f(x) = 1, ou seja, 100% de *fitness*, pode-se concluir que não houve nenhuma ocorrência de restrição, ou seja, é um horário considerado ideal.

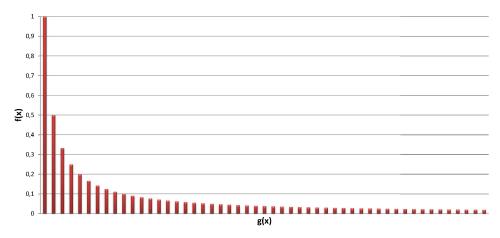


Figura 7: Gráfico da Função de Fitness

42

4.2.3 Critério de Parada

Para a proposta de horário ser considerada válida²⁷, algumas restrições exi-

gem que não haja nenhum descumprimento, enquanto outras aceitam uma deter-

minada quantidade de infrações. As restrições do primeiro tipo são denominadas

critérios obrigatórios, as do segundo tipo, critérios preferenciais.

É possível definir esses dois tipos de critérios da seguinte maneira:

Critério Obrigatório: $N_{occ} = 0$, e

Critério Preferencial: $N_{occ} < k$, onde

 N_{occ} : número de ocorrências de infrações ao critério

k: valor máximo aceitável de infrações ao critério

Esses critérios influenciarão no valor calculado do fitness da célula durante a

fase de avaliação (abordados mais à frente, na Subseção 4.3.2). Assim que uma

célula atingir o fitness máximo, o sistema para, informando que esta célula possui

um horário que atendeu a todas as restrições como fora configurado.

4.2.4 Seleção Clonal

A seleção clonal é o grande diferencial do *CLONALG*, daí a origem do nome.

De acordo com (BURNET, 1959), nessa fase é feito um ranqueamento dos indiví-

duos da população baseado no fitness de cada célula. Então é possível selecionar

²⁷O termo *válido*, no contexto deste trabalho, refere-se a um horário que possa ser utilizado sem problemas no cotidiano, não necessariamente um dos melhores horários possíveis.

apenas algumas células para serem replicadas (clonadas), em um processo chamado expansão clonal.

O processo de expansão clonal gera células idênticas em quantidade proporcional ao *fitness* da célula; ou seja, quanto maior o *fitness*, mais clones serão gerados. Dessa forma é possível multiplicar com maior velocidade aqueles indivíduos com maior afinidade.

4.2.5 Hipermutação das Células

Cada célula clonada sofre uma modificação de sua estrutura, denominada hipermutação, antes de serem adicionadas à população principal. A hipermutação nas células é feita através da alteração aleatória de horários e locais de aula a partir de um fator de hipermutação. Nesse processo, o fator de hipermutação h(x) define a probabilidade de cada aula ter seu horário e/ou local alterados, e a aleatoriedade escolhe um outro *slot* de tempo e/ou espaço para realocar a aula. Este fator pode ser obtido através da Equação 3.

$$h(x) = e^{-p*f_i(x)} \tag{3}$$

p: fator multiplicativo de mutação, intensificador ou inibidor da mutação $f_i(x)$: fitness celular, obtido na Equação (2).

4.2.6 Heurísticas

Heurísticas, no contexto de inteligência artificial, são regras que visam acelerar a convergência²⁸ de um algoritmo de otimização através da criação de diretrizes que norteiem a evolução do sistema, ao invés de deixar o algoritmo completamente à mercê da aleatoriedade.

Sendo assim, é possível complementar o *CLONALG* acrescentando heurísticas durante as etapas do algoritmo, como mostrado no trabalho de (BURKE, 1996). Pode-se criar regras que impeçam a criação de determinados horários como, por exemplo, não permitir que um horário para um curso noturno seja sequer alocado no turno diurno durante a criação ou hipermutação de uma célula.

Em um *course timetabling* há uma gama enorme de técnicas heurísticas que podem ser aplicadas. Entretanto, neste trabalho optou-se por usar apenas algumas heurísticas devido ao tempo de processamento demandado ao se julgar tais condições, dada a grande quantidade de informações e restrições.

Assim sendo, as técnicas heurísticas utilizadas foram:

• Coerência Interna da Oferta: cada oferta tem seu horário criado/modificado individualmente, portanto, durante a escolha aleatória do novo horário e/ou local, garante-se que não haja choque de horário entre aulas de uma mesma oferta. Por exemplo, uma aula da oferta X não pode ser alocada/alterada para outro dia e horário em que já haja outra aula da mesma oferta X.

²⁸Convergência pode ser entendido como a capacidade de evoluir o sistema imune em direção a um resultado desejado.

Isso evita choques de horários entre as aulas de uma única oferta, eliminando o excesso de colisões e permitindo que o algoritmo preocupe-se apenas em avaliar choques de horário entre ofertas distintas.

• Blocos de Aula: durante a sugestão e alteração de um horário de uma oferta, as aulas já são alocadas em blocos, considerando a duração máxima permitida para tal oferta. Na UFLA, o tamanho máximo de todas as aulas é de 2 créditos, com raríssimas exceções onde ofertas de 3 créditos têm uma única aula semanal de 3 créditos consecutivos.

A vantagem desta técnica é que remove a necessidade de avaliar uma das restrições citadas na Subseção 3.3, e o consumo de processamento para aplicar esta heurística é irrisório quando comparado ao processamento de se avaliar a restrição.

Turno de Horário: durante a criação de horários de aulas, restringe-se as opções de *slots* disponíveis para apenas aqueles em que os alunos cursam.
 Por exemplo, aulas de cursos diurnos serão sorteadas entre *slots* de 08:00 e 17:00; e cursos noturno, de 19:00 a 22:00.

Embora o custo computacional para essa heurística seja relativamente alto, a mesma permite um *fitness* inicial consideravelmente maior para as células. Toma-se como justificativa o exemplo uma oferta noturna: a probabilidade de ser sorteado um *slot* de tempo correto (no turno noturno) é de 25%; já com a heurística, 100%, eliminando totalmente a necessidade de avaliação desta restrição.

4.2.7 Fatores de Hipermutação

Abordagens anteriores do *timetabling* utilizam um único fator de hipermutação. Nesse trabalho, foram criados dois fatores de hipermutação com o propósito de evitar hipermutações ineficientes e aumentar a taxa de convergência do algoritmo.

Como o horário e local de uma aula podem ser alterados individualmente, foi diagnosticado que, em muitos casos, um único fator de hipermutação realiza uma transformação na grade horária incompatível com a restrição infringida.

Toma-se como exemplo uma aula prática de um curso diurno alocada na terçafeira às 14:00 em uma sala de aula²⁹. Essa aula infringe a restrição de que "aulas práticas devem ser alocadas em laboratórios". Por outro lado, o horário está
correto, já que trata-se de um curso diurno alocado às 14:00. Durante a fase de
hipermutação, o uso de um único fator implicará que o horário e o local podem
ser alterados. Então há a possibilidade do local ser alterado para um laboratório
(gerando convergência), mas, por outro lado, o horário pode ser alterado para o
período da noite, criando uma nova infração, já que trata-se de um curso diurno
(gerando divergência).

Dessa forma, o uso de dois fatores torna-se interessante, onde cada um deles é gerado de acordo com a necessidade de troca de horário ou de local, independentemente. No caso do exemplo anterior, o fator de hipermutação de horário seria bem menor que o de local, a fim de "incentivar" a troca de local e a manutenção do horário.

²⁹Considera-se "sala de aula" um espaço físico destinado a aulas teóricas, diferente de um laboratório de experimentos ou de informática, para aulas práticas.

4.2.8 Diversificação das Células

Para permitir uma cobertura maior do espaço de soluções do algoritmo, ao final de cada iteração é criado um conjunto de novas células completamente aleatórias. Assim, o sistema não fica restrito a evoluir sempre as mesmas células desde o início.

4.3 Simulação

A seguir, será mostrado como foi organizado o processo de simulação, desde a definição do seu escopo até a preparação dos parâmetros de simulação.

4.3.1 Escopo da Simulação

Como o objetivo deste trabalho é resolver o problema de alocação de horários e salas de aula para a toda a UFLA, foi necessário restringir o escopo da simulação, limitando a execução a alguns casos. Tal medida foi necessária devido ao fato de que que efetuar simulações que envolvessem todos os cursos e salas de aula da universidade impactaria fortemente no tempo de execução da fase de simulação do trabalho.

Portanto, nas simulações foram utilizados apenas os cursos de graduação em Ciência da Computação e Sistemas de Informação, por se tratarem de cursos integral e noturno, respectivamente. Essa escolha proporcionou a restrição necessária para fornecer um tempo de execução aceitável, além de cobrir todo o espaço de possibilidades de alocação de horários algoritmo.

A fim da simulação limitada manter a fidelidade para uma execução completa com todos os cursos, as opções de locais de aula também foram restringidas, permitindo somente a alocação de ofertas nas salas do Pavilhão 1 e nos laboratórios do Departamento de Ciência da Computação.

4.3.2 Parâmetros de Simulação

A seguir, será mostrado como foi definida a configuração das variáveis do sistema, suas representações e quais valores foram escolhidos para este trabalho.

Representação: O sistema criado, em conjunto com o *CLONALG*, possui grande variedade de configuração para a simulação. O algoritmo oferece a flexibilidade de configuração de 28 variáveis, divididas em quatro propósitos:

• Evolução da População: a evolução da população de células pode ser configurada através de 5 parâmetros:

pop_size: Quantidade máxima de células na população.

clone_amt: Quantidade máxima³⁰ de clones por célula.

mutation_tax: Constante de probabilidade de ocorrer mutação, definida como *p* na Subseção 4.2.5.

ranking: Quantidade de *n* melhores células envolvidas na seleção clonal.

new_amt: Quantidade de novas células a serem criadas no final de cada geração, a fim de garantir a diversificação.

• Critério de Parada: o encerramento da simulação pode ser definido através de 3 parâmetros:

acceptation: é o critério de parada que é utilizado para terminar a simulação quando todas as restrições obrigatórias já foram satisfeitas; definido por *k* na Subseção 4.2.3

³⁰Como mostrado na Subseção 4.2.4, o número de clones é proporcional aos *fitness* de cada célula.

- **time:** usado apenas para o desenvolvimento, é possível interromper a execução do sistema após um tempo pré-determinado através desta variável.
- **generation:** também com o propósito de desenvolvimento, este parâmetro para a simulação após um número específico de gerações evoluídas.
- Pesos para Avaliação: 11 parâmetros definem o grau de importância (de acordo com a instituição) para as restrições citadas na Subseção 3.3. Note que, das 15 restrições citadas, apenas 11 delas tem parâmetros de configuração; as demais foram descartadas por serem irrelevantes à instituição ou por terem sido solucionadas com as técnicas heurísticas da Subseção 4.2.6:

time_clash: peso aplicado à cada ocorrência de choque de horário.

room_clash: peso aplicado à cada ocorrência de choque de local.

spacing: peso aplicado à cada ocorrência de espaçamento inadequado.

gap: peso aplicado à cada ocorrência de janela.

swap: peso aplicado à cada ocorrência de não alternância em horários noturnos.

lab: peso aplicado à cada ocorrência de aulas prática que não estão alocadas em laboratórios.

shift: peso aplicado à cada ocorrência de aula alocada em horário fora do turno do curso.

isolation: peso aplicado à cada ocorrência de espaçamento inadequado.

- **closeness:** peso aplicado à cada ocorrência de aulas alocadas em locais distantes ao departamento que ofertou a disciplina.
- vip: peso aplicado à cada ocorrência de aulas exclusivas não alocadas no devido local.
- **load:** peso aplicado à cada ocorrência de aulas alocadas em locais que não suportam o número de alunos previstos.
- Ajuste de Instituição: 10 parâmetros permitem ajustar o sistema à realidade da instituição em questão. Em suma, configuram o horário semanal utilizado na universidade, como horário de início e término de cada turno e quantidade de dias letivos semanais. Esses parâmetros permitem a multi-aplicação do algoritmo, apenas sendo necessário ajustar os valores para a instituição analisada.

Definição: Os parâmetros citados anteriormente foram configurados na simulação deste presente trabalho da seguinte forma:

 Ajuste de Instituição: de acordo com as informações do DRCA, a grade horária semanal foi configurada se forma a oferecer 5 dias letivos semanais, com 17 slots diários (das 08:00 até as 23:00).

Os cursos diurnos tem suas ofertas iniciadas às 08:00 e alocadas, no máximo, às 17:00; excluindo o *slot* das 12:00 (intervalo de almoço). Os cursos noturnos se iniciam às 19:00 e são alocados, no máximo, no *slot* das 22:00.

O *slot* das 18:00 é alocado para aulas extras³¹, as quais estão fora do escopo deste projeto.

³¹Aulas extras podem ser ofertas para turmas de alunos repetentes ou por qualquer outro motivo.

 Pesos para Avaliação: através da consulta aos membros da atual equipe de elaboração do horário na UFLA e ao professor orientador, foi verificado o grau de importância de cada uma das restrições, a fim de definir um valor relativo³² para cada um dos parâmetros, cujos valores são mostrados na Tabela 3:

Tabela 3: Configuração de Pesos

Parâmetro	Peso Relativo
time_clash	1
room_clash	1
spacing	0,25
gap	0,33
swap	0,33
lab	1
shift	1
isolation	0,33
closeness	0,15
vip	1
load	1

• Evolução da População: os parâmetros de evolução de população são decisivos para a convergência satisfatória do algoritmo. E os seus ajustes são a grande dificuldade em qualquer algoritmo bioinspirado. Foram realizadas simulações com 15 configurações distintas (nomeadas de A até P) para permitir a sintonia fina dos parâmetros até se obter uma configuração ideal.

As configurações são criadas em trios, onde um único parâmetro é modificado para se analisar o impacto do mesmo. As configurações A, B e C, por exemplo, mantém fixo todos os parâmetros, exceto a taxa

³²O valor de cada peso varia continuamente de 0 a 1, representando quão mais importante uma determinada restrição é sobre as demais.

de mutação (mut_tax). Os valores que se mantém fixos são definidos de acordo com o melhor resultado do trio de simulação anterior.

As configurações de A até P são mostradas na Tabela 4, na qual o parâmetro analisado (variável) tem seu valor destacado com **negrito**.

Tabela 4: Configurações de Parâmetros

Config.	pop_size	clone_amt	mut_tax	ranking	new_amt
A	100	10	0,25	10	10
В	100	10	0,5	10	10
С	100	10	1	10	10
D	20	10	0,9	10	1
Е	100	10	0,9	10	10
F	200	10	0,9	10	10
G	25	3	0,9	10	10
Н	25	10	0,9	10	10
I	25	20	0,9	10	10
J	25	15	0,9	3	10
L	25	15	0,9	10	10
M	25	15	0,9	20	10
N	25	15	0,9	10	3
О	25	15	0,9	10	10
P	25	15	0,9	10	20

• Critério de Parada: Todas as simulações tiveram seu critério de parada³³ através do tempo de execução: 45 minutos de execução. Optouse por utilizar o tempo de execução em detrimento do número de gerações devido ao fato de que, para ser realmente aplicado na universidade, o tempo de convergência deve ser inferior ao atual tempo de criação de horário.

³³Veja os possíveis critérios de parada para o sistema na Subseção 4.2.3.

4.4 Análise

Foram realizadas 2 simulações para cada configuração citada na seção anterior, permitindo maior fidedignidade dos resultados obtido pela média dos mesmos, devido à aleatoriedade da execução do problema: totalizando 30 simulações. A máquina utilizada para os testes foi um Intel[®] Core2Duo³⁴ 2,53GHz, 3MB de cache e 4GB de memória RAM.

Depois de obtida a melhor configuração, foi feita uma simulação intensa com duração de 6 horas, a fim de avaliar a convergência do sistema.

Analisou-se, em cada simulação, o *fitness* da melhor e da pior célula de cada geração da população, denominadas β e ω , respectivamente. Foram considerados os *fitness* inicial e final destas células, nas gerações 0 e n, respectivamente, onde n é a primeira geração completa imediatamente após atingir o critério de parada.

Com os valores da melhor célula β é possível inferir quão eficiente foi a configuração do sistema através do cálculo do ganho de *fitness* da população $g_{fit}(\beta)$, ou seja, quanto a população convergiu, obtido pela Equação 4. A pior célula ω , por sua vez, indica se o algoritmo está estagnando em um determinado ponto, caso o *fitness* de ω aproxime-se de β .

$$g_{fit}(\beta) = f_n(\beta) - f_0(\beta) \tag{4}$$

 $f_i(x)$: fitness celular da célula x na geração i, obtido na Equação (2);

 β : a célula com melhor *fitness* da geração atual.

³⁴Processador *mobile* com 2 núcleos com 1 thread em cada, de acordo com http://ark.intel.com/products/37006/Intel-Core2-Duo-Processor-P8700-(3M-Cache-2_53-GHz-1066-MHz-FSB).

O ganho de *fitness* da população $g_{fit}(\beta)$ obtido em cada uma das configurações é mostrado na Tabela 5, juntamente com o número de gerações n atingido³⁵ pela configuração.

Tabela 5: Ganho de Fitness e Número de Gerações

$g_{fit}(\boldsymbol{\beta}) (10^{-6})$	n
8,730	16
1,660	13
24,01	13
24,14	14
7,578	14
12,83	15
9,887	38,5
20,25	15
20,00	8,5
10,68	11
12,43	10
15,02	10
15,46	11
17,16	8
9,675	7
	8,730 1,660 24,01 24,14 7,578 12,83 9,887 20,25 20,00 10,68 12,43 15,02 15,46 17,16

4.4.1 Restrições Não Atendidas

Também é importante analisar em cada simulação a sua capacidade de atender às restrições impostas pelo sistema. Foi verificado a quantidade de restrições obrigatórias não atendidas por cada configuração, tanto na Geração 0 quanto na Geração n; os resultados obtidos são mostrados Tabela 6. O sistema apenas estará próximo do resultado ideal caso este valor final seja zero.

³⁵O número de gerações varia entre as configurações porque o critério de parada escolhido foi o tempo de execução e, como cada configuração lida com quantidade diferentes de células, o tempo gasto para evoluir e analisar a população é diferente.

Tabela 6: Quantidade de Restrições Não Atendidas

Config.	Geração 0	Geração n	Diferença
A	1415	1416	+0,080%
В	1415	1385	-2,120%
С	1415	1421	+0,424%
D	1418	1409	-0,599%
E	1415	1456	+2,933%
F	1415	1389	-1,837%
G	1414	1408	-0,389%
Н	1415	1375	-2,826%
I	1428	1396	-2,206%
J	1415	1415	0,000%
L	1425	1461	+2,526%
M	1404	1410	+0,427%
N	1415	1422	+0,495%
О	1393	1395	+0,144%
P	1415	1347	-4,806%

Pelos resultados representados no gráfico da Figura 8 pôde-se perceber que, embora a algumas simulações tenham conseguido uma leve diminuição na quantidade de infrações, houve casos em que houve um aumento das mesmas (Configurações A, C, E, L, M, N e O). Em contrapartida, o *fitness* de cada uma dessas configurações aumenta (Tabela 5).

Esse fenômeno de aumento de erros acompanhado do aumento de *fitness* pode ser explicado pelo fato que o *fitness*, nesses casos, foi impulsionado por um alto grau de atendimento a restrições não obrigatórias, mas gerou um número maior infrações às restrições obrigatórias, justificando o aumento das mesmas.

Este tipo de acontecimento prejudica a convergência do algoritmo, e pode ter suas consequências amenizadas através de um ajuste melhor dos pesos para as restrições, reduzindo o valor relativo dos pesos das restrições opcionais, a fim de valorizá-las menos.

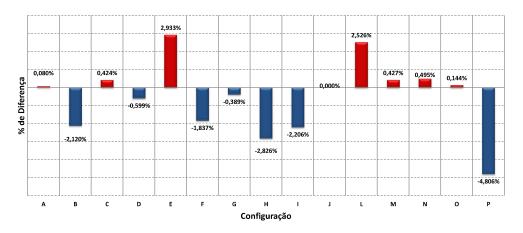


Figura 8: Diferença Final e Inicial de Restrições Não Atendidas

4.4.2 Heurísticas

O uso de técnicas heurísticas representou um impacto direto na velocidade e qualidade da convergência do sistema. Através das técnicas utilizadas na Subseção 4.2.6, o algoritmo teve um ganho de até 553% na velocidade de convergência, como pode ser visto na Figura 9. O gráfico reforça a necessidade de utilização - talvez até maior (a ser discutida no capítulo a seguir) - de técnicas heurísticas.

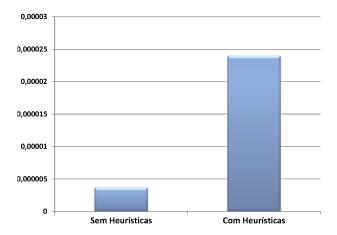


Figura 9: Impacto do Uso de Heurísticas no Ganho do Fitness

Além do ganho de *fitness*, a quantidade de infrações a restrições obrigatórias também é consideravelmente menor com o uso de heurísticas. Como mostrado na Figura 10, sem heurísticas, a população inicia o processo evolutivo com 2321 infrações em média, e termina, após os 45 minutos de simulação, com 2300 infrações, representando -0.85% de diminuição. Já o algoritmo com heurísticas, por sua vez, inicia com 1415 infrações em média e termina com 1347: -5% de diminuição.

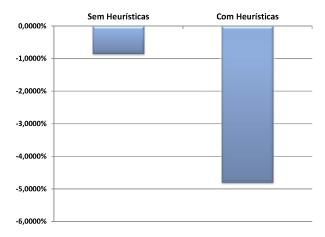


Figura 10: Impacto do Uso de Heurísticas nas Restrições Não Atendidas

4.4.3 Fator de Hipermutação

A variação do fator de hipermutação nas simulações mostrou um impacto grande nos resultados obtidos, inferido a partir dos resultados das configurações A, B e C. Essa relação pode ser visualizada através da Figura 11, onde nota-se que, "quanto maior [a taxa de mutação], melhor [o ganho de *fitness*]".

Dessa forma, chegou-se a um valor ideal para o fator de hipermutação: 90% de probabilidade.

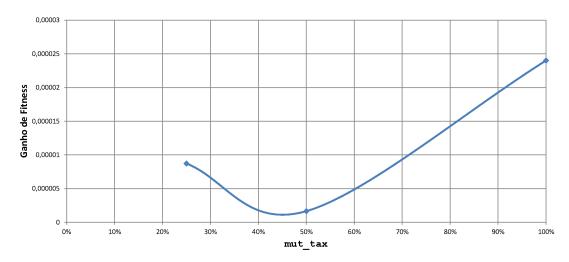


Figura 11: Relação entre Fatores de Hipermutação e Fitness

4.4.4 Tamanho da População

Através da Figura 12 (obtida das configurações, D, E e F) nota-se que, para populações acima de 20 células, o ganho de *fitness* diminui drasticamente, voltando a subir gradualmente acima de 100 células. Portanto, optou-se por utilizar, nas configurações seguintes, populações de tamanho 25.

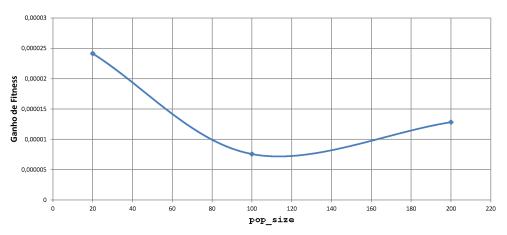


Figura 12: Relação entre Tamanho da População e Ganho de Fitness

4.4.5 Máximo de Clones

As configurações G, H e I analisaram o número máximo de clones por célula escolhida, cujo resultado é mostrado na Figura 13. É possível perceber que, à medida em que se aumenta o clone_amt, há um ganho de *fitness*. Entretanto, a partir de 10 clones/célula, o ganho de *fitness* mantém-se praticamente constanste.

Como a quantidade de clones impacta diretamente no tempo de execução, o ideal é utilizar a menor quantidade possível de clones/célula. Sendo assim, optouse por utilizar 15 clones por célula durante a fase de clonagem.

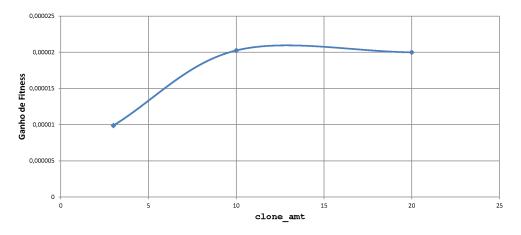


Figura 13: Relação entre Quantidade de Clones e Ganho de Fitness

4.4.6 Diversificação

Na Figura 14, obtida dos resultados das configurações J, L e M, o gráfico mostra que o ganho de *fitness* é diretamente proporcional³⁶ à quantidade de células de diversificação

 $^{^{36}}$ A relação "diretamente proporcional" é válida apenas para a região calculada ($2 \le x \le 20$), até que se prove sua extensão para outros valores.

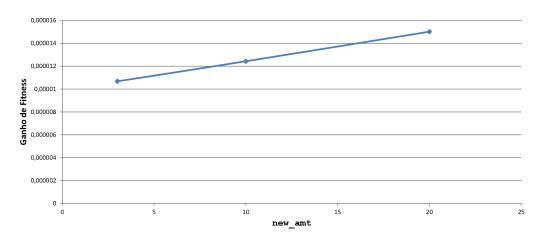


Figura 14: Relação entre Diversificação e Ganho de Fitness

Assim sendo, nas simulações seguintes, foram geradas 15 novas células ao término de cada geração para permitir a diversificação da população.

4.4.7 Ranking

A quantidade de células selecionadas na fase de Seleção Clonal, denominadas ranking, foi analisada nas configurações N, O e P. O resultado da relação entre o ranking e o ganho de *fitness* obtido é expresso na Figura 15.

Sobre este gráfico infere-se que o ganho de *fitness* passa a diminuir à medida em que se escolhe mais células para a seleção clonal, permitindo concluir que basta algumas células (entre 5 e 10 células) para garantir o progresso máximo do sistema. Portanto, 8 células para o ranking é considerado o valor que oferece melhor resultado.

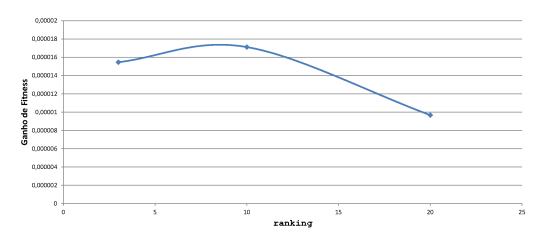


Figura 15: Relação entre Ranking e Ganho de Fitness

4.5 Simulação Intensa

Após a análise individual de cada parâmetro, chegou-se a uma configuração ideal, mostrada na Tabela 7.

Tabela 7: Configuração Ideal de Parâmetros

pop_size	clone_amt	mut_tax	ranking	new_amt
25	15	0,9	8	25

Com esta configuração, foi realizada um simulação mais intensa, com critério de parada temporal maior: 6 horas ininterruptas de execução. Esta simulação foi realizada em uma máquina com processador Intel[®] Core i7 930³⁷ 2,8GHz, 8MB de cache e 4GB de memória RAM.

Ao término da simulação, foram obtidos os seguintes resultados mostrados na Tabela 8.

 $^{^{37}} Processador \ desktop \ com \ 4 \ núcleos \ e \ 8 \ threads \ no \ total, \ de \ acordo \ com \ http://ark.intel.com/products/41447/Intel-Core-i7-930-Processor-(8M-Cache-2_ 80-GHz-4_80-GTs-Intel-QPI)$

Tabela 8: Resultados da Simulação Intensa

Fitness Co	elular	Restrições Não Atendidas		
Ganho (10 ⁻⁶)	$f_n(\boldsymbol{\beta})$	Diferença	Menor Valor	
24,71	0,000813	-2,1%	1387	

Foi criado o histórico de *fitness* das células β e ω de cada uma das 35 gerações atingidas a fim de verificar a capacidade de convergência do sistema. Este histórico é mostrado na Figura 16, sobre a qual podemos confirmar a capacidade de convergência do sistema criado.

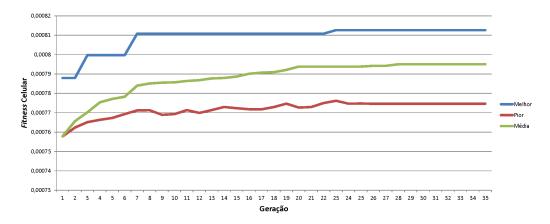


Figura 16: Histórico de *fitness* das Células β e ω

Através do histórico de *fitness* celular pode-se perceber três características:

Melhor Célula: a melhor célula β tem o valor do *fitness* constante durante várias gerações consecutivas, o que demonstra que mesmo a clonagem e mutação da melhor célula atual gera células de *fitness* inferior. O que revela a dificuldade em se obter, dentre as demais células, alguma célula com *fitness* melhor.

Pior Celula: ao contrário da melhor célula, a pior célula ω tem seu *fitness* alterado tanto pra mais quanto pra menos. A diminuição do *fitness* é caracte-

rística do *CLONALG*, que pode gerar célula ainda piores durante a fase de diversificação. Entretanto, de um modo geral a pior célula tende a manter seu *fitness* em crescimento.

Fitness Médio: por si só, o *fitness* da pior célula ω não permite inferir informações sobre o estado da população como um todo. Portanto foi calculado o valor do *fitness* médio, obtido pela média do *fitness* de todas as células da população. Através deste valor, pode-se acompanhar a convergência de toda a população, mostrada pelo crescimento da curva intermediária no gráfico da Figura 16.

4.6 Comentários Finais

O sistema desenvolvido teve um resultado muito satisfatório no que se refere à ganho de desempenho, proporcionando melhora de mais de 5 vezes quando comparado ao *CLONALG* simples, sem heurísticas.

Além disso, o trabalho desenvolvido contribui com diversas relações de impacto de cada parâmetro sobre o desempenho do algoritmo, através das curvas mostradas nas Subseções de 4.4.3 a 4.4.7. Estas curvas permitem definir novas configurações completas para o sistema desenvolvido sem que haja a necessidade de simulações para tal.

A simulação realizada na Subseção 4.5 confirma a capacidade de convergência do sistema, além de reforçar o desempenho promissor de algoritmos imunes quando aplicados no conceito de otimização de resultados no *timetabling*.

5 Conclusão

A melhor proposta para se resolver o problema de alocação de horários é através de técnicas inteligentes e, dentro da gama de algoritmos que se propõem a oferecer uma solução, os algoritmos imunes ainda são pouco explorados na literatura, quando comparado a outras técnicas, como algoritmos genéticos, por exemplo. Além disso, os algoritmos imunoinspirados mostram-se eficientes ao abordarem outros problemas de complexidade semelhante à do *timetabling*. Fato que demonstra que esta técnica pode oferecer excelentes resultados ao ser aplicada para resolver o problema do *timetabling* em uma instituição de ensino de grande porte e complexidade, como a Universidade Federal de Lavras.

Entretanto, a otimização de um resultado para o *course timetabling* é um processo longo e complexo, dado número exorbitante de possibilidades de combinações para criar um horário. Característica que é acrescida de outros diversos fatores complicantes, como as restrições obrigatórias e opcionais, tempo de execução e parâmetros de configuração.

Neste trabalho foi criado um sistema robusto e flexível que permitiu não só obtenção de uma configuração de parâmetros eficiente, mas também a criação de curvas que definem o impacto de alteração de cada parâmetro no desempenho do sistema.

Uma simulação mais completa, de duração maior, permitiu perceber que, embora o tempo de execução seja ainda muito longo, o sistema tem grande tendência a chegar a um resultado válido. Entretanto a otimização do sistema torna-se inevitável, a fim de tornar o uso do algoritmo vantajoso em relação ao atual processo de escalonamento de horários.

Almeja-se a modificação do sistema desenvolvido neste trabalho a fim de se aproximar mais de um resultado útil para a universidade. Propõe-se, desde já, comparar métodos inteligentes para se determinar a melhor abordagem para o problema do *course timetabling*.

A técnica da Subida de Encosta, mostrada brevemente na Subeção 2.3.3, mostra-se promissora na aplicação para o problema, uma vez que se tem uma solução inicial, não obstante esteja incompleta devido à adição e/ou modificação de ofertas para o semestre seguinte. Dessa forma, poder-se-ia evoluir o sistema inteligente a partir do horário de todas as ofertas do semestre atual.

Também é possível intensificar as técnicas heurísticas utilizadas, já que as que foram utilizadas mostraram impacto significante na convergência do algoritmo. Propõe-se um critério de hipermutação com avaliação mais profunda: analisando cada *slot* de tempo em que haja aula alocada a fim de verificar sua adequação às restrições impostas. Dessa forma será possível alterar com mais frequência apenas os horários incoerentes, evitando a atual modificação de horários que estão de acordo com as restrições, causada pela aleatoriedade do processo.

Além disso, para minimizar o tempo de execução, seria interessante paralelizar trechos do algoritmo, como já mostrado no trabalho de (ABRAMSON, 1991), principalmente o que se refere à avaliação das células, já que o mesmo se trata de um trecho não dependente.

Referências

ABRAMSON, D. Constructing school timetables using simulated annealing: Sequential and parallel algorithms. *Management Science*, v. 37, p. 16, 1991.

BURKE, E. K. A late acceptance strategy in hill-climbing for exam timetabling problems. *Automated Scheduling, Optimisation and Planning Group, School of Computer Science & IT*.

BURKE, E. K. Initialization strategies and diversity in evolutionary timetabling. *Evolutionary Computation*, v. 6, 1996.

BURNET, M. The clonal theory of acquired immunity. *University Press, Cambridge*, 1959.

EVEN, S. On the complexity of timetabling and multicommodity flowproblems. *SIAM Journal of Computation*, v. 5, 1976.

GOLDBERG, D. E. Genetic algorithms in search, optimization and machine learning. *The University of Alabama*, 1989.

LIMA, E. O. Algoritmo genético híbrido aplicado à otimização de funções. Trabalho de Conclusão de Curso, DCC/UFLA, 2008.

MOORE, A. W. Iterative improvement search - hill climbing, simulated annealing, walksat, and genetic algorithms.

MR, J. D. G. Computers and intractability - a guide to np-completeness. W. H. Freeman and Company, San Francisco, v. 1, 1979.

SANTOS, M. Programação matemática. *Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo*, 2010.

SCHAERF, A. A survey of automated timetabling. *Artificial Intelligence Review*, v. 3, 1999.

SILVA, L. N. de C. Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artficiais.

Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação,
Universidade Estadual de Campinas., 2001.

SIPSER, M. *Introduction to the Theory of Computation*. [S.1.]: Thomson Course Technology, 2006.

THOMPSON, K. A. D. J. M. Variants of simulated annealing for the examination timetabling problem. [S.l.]: Computer Science, Annals of Operations Research, 1996.

TIMOTEO, G. Desenvolvimento de um algoritmo genético para a resolução do timetabling. *Trabalho de Conclusão de Curso - DCC/UFLA*, 2002.

UCHOA, J. Q. *Utilização de Algoritmos Inspirados no Sistema Imune para*Detecção de Intrusos em Redes de Computadores. Tese (Doutorado) — UFMG, 2009.

YOURDON, P. C. . E. Análise Baseada em Objetos. [S.l.]: Campus, 1992.

A ANEXO: Interface web do Sistema - Início

Timetable UFLA

DEPARTAMENTOS	CURSOS	LOCAIS
DAE		
<u>DCC</u>	Ciência da Computação Sistemas de Informação	Salas do DCC
<u>DEX</u>		Pavilhão 3
<u>DCH</u>		Pavilhão 1
<u>PRG</u>		

Relatórios

Choques de Horário Choques de Espaço Distribuição de Aulas Adequação de Aulas

B ANEXO: Interface web do Sistema - Horário Individual

			Curso: G010		
	Segunda	Terca	Certodo: 5 Quarta	Quinta	Sexta
7:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
8:00	GCC107 (T) - 10A 10B 10C 11A 13A (PV308)	(vazio)	GCC106 (T) - 10A (PV306) GCC107 (T) - 10B (PV118)	(vazio)	GCC107 (T) - 10B (DCC02) GEX106 (T) - 10A (PV306)
9:00	GCC107 (T) - 104 108 10C 114 13A (PV308)	GCC106 (T) - 104 (PV117) GCC107 (T) - 10C 114 134 (DCC02)	GCC106 (T) - 104 (PV306) GCC107 (T) - 108 (PV118) GEX112 (P) - 104 108 2A (PV301)	(vazio)	GCC107 (T) - 108 (DCC02) GCC109 (T) - 104 (PV309) GEX106 (T) - 104 (PV306)
10:6	10:00 (vazio)	GCC106 (D - 104 (PV117) GCC107 (D - 10C 114 134 (DCC02) GEX112 (D - 108 (DCC05)	GEX112 (P) - 10A 10B 2A (PV301)	(vazio)	GCC107 (T) - 104 (PV303) GCC107 (T) - 10C 114 134 (PV103) GCC109 (T) - 104 (PV309)
11:6	11:00 (vazio)	GEX112 (T) - 10B (DCC05)	(vazio)	(vazio)	GCC107 (T) - 104 (PV303) GCC107 (T) - 10C 114 134 (PV103)
12:6	12:00 (vazio)	(vazio)	(vazio)	(vazio)	(vazio)
13:6	13:00 (vazio)	(vazio)	(vazio)	(vazio)	(vazio)
14:6	14:00 GCC110 (D - 104 (DCC01)	(vazio)	GEX106 (T) - 104 (PV107) GEX112 (T) - 104 10B 2A (PV111)	GCC110 (T) - 10.4 (PV302)	GCC107 (P) - 104 (DCC01) GCC108 (T) - 104 (DCC01) GCC109 (T) - 104 (PV109)
15:0	15:00 GCC110 (D - 104 (DCC01) GEX112 (P) - 108 (DCC03)	GCC107 (D - 104 (DCC04) GEX112 (D - 104 (PV109) GEX112 (D - 104 (PV312)	GEX106 (D <i>104</i> (PV107) GEX112 (D <i>104</i> 108 24 (PV111)	GCC108 (D - 104 (PV118) GCC110 (D - 10A (PV302)	GCC107 (P) - 104 108 10C 114 134 (PV308) GCC107 (P) - 104 (DCC01) GCC107 (P) - 108 (PV310) GCC108 (D) - 104 (PV109) GCC109 (D) - 104 (PV109)

GCC107 - Teorica (10A 10B 10C 11A 13A)

HORARIO

	Segunda	Terca	Quarta	Quinta	Sexta
7:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
8:00	PV308	(vazio)	(vazio)	(vazio)	(vazio)
9:00	PV308	(vazio)	(vazio)	(vazio)	(vazio)
10:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
11:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
12:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
13:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
14:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
15:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
16:00	DCC03	(vazio)	(vazio)	(vazio)	(vazio)
17:00	DCC03	(vazio)	(vazio)	(vazio)	(vazio)
18:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
19:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
20:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
21:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)
22:00	(vazio)	(vazio)	(vazio)	(vazio)	(vazio)

Choques de Horário

Foram detectados 168 choques de horário.

Curso: G010

	<u>1º Período</u>		
	9:00: GCC102 (T) GEX103 (T)		
Sexta	14:00: GCH101 (T) GEX103 (T)		
Sexia	15:00: GCH101 (T) GCH102 (T) GEX102 (T) GEX103 (T)		
	16:00: GCC101 (T) GCH102 (T) GEX102 (T)		
	2º Período		
Overte	15:00: GCC103 (T) GCC104 (T) GEX104 (T)		
Quarta	16:00: GCC104 (T) GEX104 (T)		
	10:00: GAE196 (T) GEX133 (P)		
Sexta	15:00: GCC105 (T) GCH104 (T) GEX104 (T)		
	16:00: GCC104 (P) GCH104 (T) GEX104 (T)		
	3º Período		
Terca	15:00: GCC107 (T) GEX112 (P) GEX112 (T)		
Terca	16:00: GCC107 (T) GEX112 (P) GEX112 (T)		
	9:00: GCC106 (T) GCC107 (T) GEX112 (P)		
Quarta	14:00: GEX106 (T) GEX112 (T)		
	15:00: GEX106 (T) GEX112 (T)		
Quinta	15:00: GCC108 (T) GCC110 (T)		
	9:00: GCC107 (T) GCC109 (T) GEX106 (T)		
Sexta	10:00: GCC107 (T) GCC107 (T) GCC109 (T)		
Sexia	14:00: GCC107 (P) GCC108 (T) GCC109 (T)		
	15:00: GCC107 (P) GCC107 (P) GCC108 (T) GCC109 (T)		
	4º Período		
	10:00: GCC111 (T) GCC114 (P) GEX108 (T)		