

Construção de um plano de estudos automatizado utilizando algoritmos genéticos

Victor M. G. Jatobá¹, Wellington L. S. Lacerda¹

¹União Metropolitana de Educação e Cultura – Faculdade de Ciência Exatas e Tecnologia (UNIME) – Lauro de Freitas – BA – Brasil

{victorjatoba10,wellington.lacerda}@gmail.com

Abstract. *Despite the difficulty and subjectivity found in the assembly of study plans, experts in the field of learning and human cognition raised some important issues to be met to mount a good plan. This assembly means scale optimally, subjects and workloads to be studied on day periods of a predetermined time. The adequacy and the need to use heuristic methods to solve these problems, the concepts of Genetic Algorithms have been used to automate the process of generating these plans. In order to demonstrate that the restrictions raised by the authors of reference were met, it was described the strategy used to achieve each one. Also, to prove the finite time automation in the generation of grids studies, the results of different crosses of data entries were analyzed. The most significant contribution of this work has been able to implement successfully the constraints raised by experts and using Genetic Algorithm, automate the process of generating a good plan of studies considering the users preferences.*

Keywords: *Genetic Algorithms, Artificial Intelligence, Scheduling Problems.*

Resumo. *Apesar da dificuldade e subjetividade encontrada na montagem de planos de estudos, especialistas na área de aprendizado e cognição humana levantaram alguns aspectos importantes a serem cumpridos para se montar um bom plano. Essa montagem se dá em escalonar, da melhor forma possível, matérias e cargas horárias a serem estudadas em períodos do dia de um tempo pré-determinado. Pela adequação e pela necessidade de se utilizar métodos heurísticos para solucionar problemas dessa natureza, foram utilizados os conceitos de Algoritmos Genéticos para automatizar o processo de geração desses planos. Como forma de demonstrar que as restrições levantadas pelos autores de referência foram alcançadas, foi descrito a estratégia utilizada para atingir cada uma. Também, para provar a automatização em tempo finito da geração das grades de estudos, foram analisados os resultados de diferentes cruzamentos de entradas de dados. A contribuição mais significativa deste trabalho foi conseguir implementar, com sucesso, as restrições levantadas pelos especialistas e, utilizando Algoritmos Genéticos, automatizar o processo de geração de um bom plano de estudos considerando as preferências dos usuários.*

Palavras-chave: *Algoritmos Genéticos, Inteligência Artificial, Problemas de escalonamento.*

1. Introdução

Em geral, problemas de otimização requerem métodos heurísticos para achar uma boa solução dentro de um período de tempo finito [Ellingsen e Penaloza 2003]. Os Algoritmos Genéticos (AG) são muito utilizados em problemas de natureza complexa, com difícil formulação matemática, ou com grande espaço de busca [Davis 1991], pois estes mantêm um equilíbrio entre os pontos conflitantes: exploração (aproveitamento das melhores soluções) e exploração (exploração do espaço de busca) conseguindo, assim, um aproveitamento mais satisfatório de melhores soluções e exploração do espaço de busca [Carvalho 2011].

Com isso, pela boa adequação dos AGs com problemas de escalonamento, este trabalho propõe-se em implementar estratégias para atender os critérios que definem um bom plano de estudo, levantados por especialistas na área de aprendizado e cognição humana e a partir dessa implementação utilizar AGs para automatizar o processo de escalonamento das disciplinas e suas respectivas cargas horárias a serem estudadas em cada período do dia, levando em consideração as preferências do usuário.

O funcionamento dos algoritmos genéticos se dá pela criação de uma população inicial aleatória, onde esta evolui aplicando três operações básicas: reprodução ou seleção, cruzamento e mutação [Ellingsen e Penaloza 2003]. Cada indivíduo representa um plano de estudos, sendo assim um candidato à solução. Após a geração da população inicial começa-se um ciclo, onde inicialmente é selecionado um grupo de indivíduos para serem cruzados (*crossover*). Posteriormente uma parte dessa população sofre uma mutação e logo depois os indivíduos passam por um processo de avaliação. Nesta etapa é aplicada uma função de *fitness* que pontua cada indivíduo, classificando-os como melhores ou piores entre si. Por fim, são selecionados os candidatos que tiveram o melhor *fitness*, retornando assim, para o início do ciclo ou parando caso tenha achado o indivíduo ideal ou o número de gerações tenha chegado ao limite pré-estabelecido. A grade ideal será aquela que minimizará mais as penalidades por não satisfazer as restrições menos e mais brandas. Nas próximas sessões serão apresentados a fundamentação teórica, a descrição do problema, a estratégia da solução utilizada para sanar o problema, bem como a descrição desta implementação, seguindo-se, assim, dos resultados e conclusões.

2. Planificação dos estudos

Zimmerman et al (1994) afirma que “[...] Na planificação das actividades de estudo, não se trata somente de aumentar o tempo de estudo, mas de promover a sua utilização mais adequada e eficaz, [...]”. Por conta disso, o primeiro impasse aqui encontrado foi achar, na literatura, algum especialista que definisse quais restrições deveriam ser atendidas para se gerar grades de estudos ideais para cada aluno. Mas, segundo Carita et al (1998), apesar da complexidade, existem aspectos importantes na etapa de planificação das atividades que deverão ser levados em consideração. Portanto, para o trabalho, foram adotados os pontos de definição presentes na Tabela 1 levantadas por esses autores de referência. Para cada ponto de definição é exposto a estratégia utilizada para satisfazê-lo. Esses pontos de definição são utilizados para traçar o perfil do aluno.

Tabela 1. Pontos de definição a serem identificados nos alunos, segundo Carita et al (1998) e a estratégia utilizada para alcançá-los

Pontos de definição	Estratégia utilizada para atingir o ponto
1. O ritmo pessoal de trabalho	Escolher em cada período do dia (manhã, tarde e noite) um grau de disponibilidade (nenhuma, baixa e alta).
2. As dificuldades de cada disciplina	Será atribuído peso às matérias a partir do nível de dificuldade do usuário.
3. As horas mais apropriadas para o estudo	Escolher em cada período do dia (manhã, tarde e noite) um grau de disposição intelectual e física (baixo, médio ou alto).
4. Os tempos dedicados ao lazer	O usuário informa quanto tempo ele irá querer reservar para não estudar.

Já na Tabela 2 encontram-se os critérios de avaliação também levantados por Carita et al (1998), que foram utilizadas para se gerar o plano de estudos ideal. Na sessão 7 estão descritas as estratégias utilizadas para satisfazer cada uma dessas restrições.

Tabela 2. Critérios de avaliação para o plano ideal segundo Carita et al (1998)

1. Não gastar todo o tempo de estudo numa só disciplina;
2. Ter em conta que algumas disciplinas necessitam mais tempo;
3. Deve começar-se pelas matérias de grau médio de dificuldade, seguindo-se pelas de maior dificuldade e finalizando com as mais fáceis;
4. Contemplar alguns minutos de intervalo entre o estudo de duas disciplinas, para descansar;
5. Assegurar que todas as disciplinas cadastradas estejam contempladas no plano.

3. Algoritmos Genéticos

Os algoritmos genéticos (AG), inicialmente propostos por Holland (1975), são algoritmos de busca baseados em mecanismos de seleção natural e genética [Vasconsellos 2011]. Ainda segundo Vasconsellos (2011), os AGs fundamentam-se na teoria da evolução das espécies de Darwin (1959) e no mecanismo de herança genética proposta por Mendel (1865). Pacheco (2005) expõe que o processo de evolução se dá em privilegiar os indivíduos mais aptos, onde estes possuem maior probabilidade de perpetuarem seus códigos genéticos que são representados pelos cromossomos. Os AGs então empregam esse princípio e evoluem populações de potenciais soluções codificadas através de cromossomos artificiais. O mesmo autor afirma que, o processo evolucionário envolve as etapas de avaliação, seleção, cruzamento (*crossover*) e mutação.

4. Definição do Problema

O propósito deste trabalho é utilizar estratégias de implementação para atender os critérios levantados por Carita et al (1998) e com isso, utilizar os AGs para automatizar a geração do plano de estudos ideal escalonando da melhor forma possível, matérias e suas respectivas cargas horárias a serem estudadas em cada período do dia considerando

o perfil do usuário que se deseja gerar o plano. Para identificar o perfil do aluno, este deverá informar qual o grau de disponibilidade e disposição intelectual e física (também aqui chamado de facilidade de aprendizado) para cada período do dia presente no plano. Além disso, deverão ser informadas quais as matérias que comporão o plano e o grau de dificuldade e importância que o usuário possui para cada uma. A Figura 1 mostra um exemplo de um plano de estudos gerado para três dias.

Dia 1:	Dia 2:	Dia 3:
Manhã: Matemática 2h História 1h	Manhã: Geografia 2h História 3h	Manhã: Geografia 2h
Tarde: Geografia 1h	Tarde:	Tarde: Português 1h
Noite: Português 4h	Noite: Matemática 4h	Noite: Matemática 2h Português 4h

Figura 1. Exemplo de um plano de estudos de três dias. No primeiro dia, no período da noite, deverão ser reservadas 4 horas para estudar português. O segundo dia à tarde, será reservado para não estudar nenhuma das matérias cadastradas e na manhã do terceiro dia, por exemplo, deverão ser reservadas duas horas para se estudar geografia.

5. Implementação

Um algoritmo genético foi construído com o intuito de resolver o problema de escalonamento de matérias e cargas horárias, melhor distribuindo-as em uma grade de horários considerando o perfil do usuário e respeitando as restrições utilizadas para se gerar um bom plano de estudos levantados por Carita et al (1998). Esse algoritmo também tem por finalidade automatizar o processo de geração do plano de estudos ideal. O AG empregado foi desenvolvido em JAVA utilizando o *framework* ECJ¹ [Luke 2002]. Para o funcionamento do algoritmo, foi preciso ser implementado, principalmente: (i) o método de cruzamento, (ii) o gene do indivíduo, (iii) o problema em si, que é onde se faz o cálculo para pontuar os indivíduos bons e ruins, (iv) a forma de mutação e (v) o método de criação da população inicial.

O método para gerar a população inicial cria matérias e cargas horárias aleatórias e as joga em períodos quaisquer. Considerou-se que uma pessoa teria uma noite de sono de 6 horas/dia, logo cada período do dia possuirá um máximo de 6 horas. Porém, para atender o critério 3 da tabela 2, diminui-se esse máximo em uma hora, portanto só haverá períodos com no máximo 5 horas reservadas para estudo. Assim haverá sempre no mínimo 1 hora para descanso a ser distribuída pelo aluno entre os estudos das disciplinas em cada período.

Um indivíduo (plano de estudos) basicamente possui um conjunto de planos diários (genes) representando o genoma deste. Um plano diário possui três períodos (manhã, tarde e noite) e cada período possui um conjunto de Matérias com suas respectivas cargas horárias a serem estudadas. A forma como foram representados esses dados encontra-se na subseção Estrutura de Dados.

¹ ECJ – em inglês *Evolutionary Computation Java-based*. É uma biblioteca de computação evolutiva em JAVA e pode ser baixada através do endereço <http://cs.gmu.edu/~eclab/projects/ecj/>

Na figura 2 é exposto o fluxograma do funcionamento básico do AG implementado. Primeiro são gerados de forma aleatória um número fixo de indivíduos que irão formar a população inicial e avaliados posteriormente. Esta avaliação gera um valor de *fitness* para cada indivíduo. Na condição que verifica se o algoritmo deve parar é averiguado se foi encontrado o indivíduo ideal ou o número de gerações atingiu o limite pré-estabelecido. Caso a condição seja falsa, passa-se para a etapa de seleção, cruzamento e mutação respectivamente. Logo após a população é atualizada e o processo retorna ao fluxo novamente.

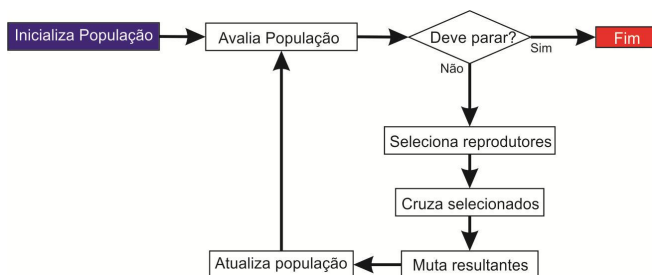


Figura 2. Fluxograma do funcionamento do Algoritmo Genético implementado.

São duas as formas de mutação. A primeira é mudando uma matéria para outra em um período de um dia aleatório. A outra é trocando o valor da carga horária de uma matéria também de forma aleatória.

O crossover ocorre de duas maneiras, por dia ou por período. Ou troca-se dois dias iguais (dia 1 do indivíduo 1 com o dia 1 do indivíduo 2, por exemplo) entre os indivíduos selecionados, ou troca-se dois períodos aleatórios de dias também aleatórios. Na figura 3 é apresentado um exemplo dos dois tipos de cruzamento.

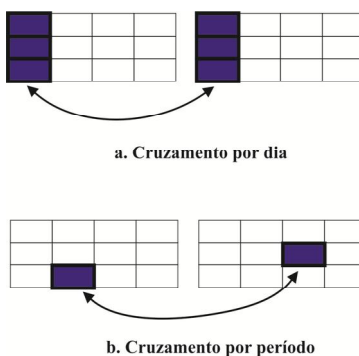


Figura 3. Formas de cruzamento. Cada tabela representa um indivíduo, onde as colunas representam os dias e as linhas os períodos do dia.

Para pontuar os indivíduos bons e ruins, foram implementadas nove restrições divididas em três classificações: *fixed*, *hard* e *soft* [Ellingsen e Penalosa 2003]. Estas possuem prioridade alta, média e baixa respectivamente. A descrição do cálculo para evoluir as populações encontra-se na sessão 6. Para atender as restrições levantadas pelos especialistas, primeiro teve que ser implementadas algumas restrições implícitas que davam base para as restrições de Carita et al (1998) existirem. São elas: (i) não alocar matérias em períodos do dia em que o aluno não tivesse disponibilidade; (ii) alocar matérias difíceis em períodos que o aluno possuísse maior facilidade de aprendizado; (iii) preencher o máximo possível o tempo em que o aluno tem disponível,

tanto com matérias, quanto com horas para o lazer; (iv) gerar planos de estudos diversificados, evitando planos diários iguais. A restrição i foi classificada como *fixed* e as ii, iii e iv como *hard*. As descrições das restrições da Tabela 2, que são as levantadas pelos especialistas, encontram-se na Sessão de Discussão.

A tabela 3 mostra como foram classificadas as matérias por nível de dificuldade. Cada matéria cadastrada conterà um nível de dificuldade que vai de 0 a 100, onde zero não tem dificuldade e 100 possui a dificuldade máxima.

Tabela 3. Classificação das matérias por nível de dificuldade.

Faixa de dificuldade	Classificação para cada faixa
0 >= n < 20	Fácil
20 >= n < 40	Fácil/Médio
40 >= n < 60	Médio
60 >= n < 80	Difícil/Médio
80 >= n <= 100	Difícil

5.4 Estrutura de Dados

Na figura 4, o indivíduo é representado pela classe *DayPlanGeneVectorIndividual* e o seu genoma nada mais é que um conjunto de *DayPlanGene*, que representa o gene do indivíduo. O gene nada mais é que um plano diário e ele é composto por três períodos do dia: manhã, tarde e noite. Um período do dia é um conjunto de *SubjectWorkload*, sendo este composto por um atributo chamado *workload* que é a quantidade de horas alocadas para a matéria específica e a matéria propriamente dita, retratado por um atributo da classe *Subject*. Por fim, esta classe representa a matéria, onde esta possui um identificador numérico, um nome e um nível de dificuldade que deverá estar entre 0 e 100.

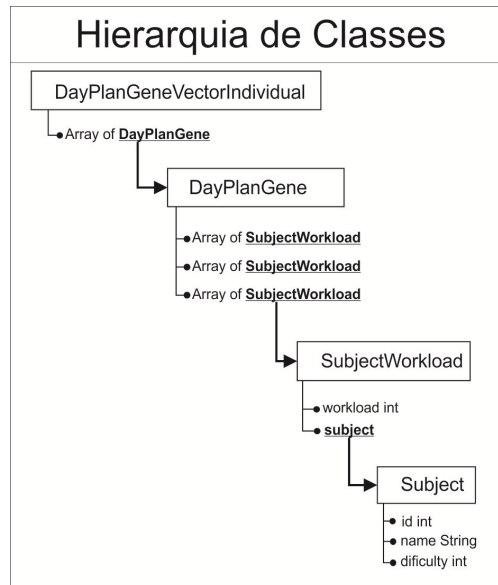


Figura 4. Representação da hierarquia de classes usada para representar o indivíduo e o seu respectivo gene.

Os dados de entrada que irão preencher as informações das classes descritas acima serão colhidos por quatro arquivos de entrada. Onde o primeiro deverá conter uma lista de matérias seguidas de um número entre 0 e 100 representando o grau de dificuldade de cada uma. Outros dois arquivos conterão respectivamente um grau de disponibilidade (baixo, médio ou alto) e de facilidade de aprendizado (bom, médio ou ruim) para cada período do dia (manhã, tarde e noite). Por fim o aluno deverá preencher em outro arquivo o seu nome e a quantidade de horas que ele pretende destinar para o lazer, ou seja, para não estudar. Essa quantidade de horas deverá ser calculada em relação a totalidade de horas disponível no plano de estudos.

6 Evolução / Processo de avaliação

A evolução acontece graças à função de *fitness*, que é responsável por retornar um valor que define o quão bom é cada indivíduo. Nessa função existem três variáveis nomeadas: *fixed*; *hard* e *soft*. Cada variável guarda a média da soma dos valores acumulativos de cada restrição pelo seu respectivo tipo. Em outras palavras, para obter a média na variável *fixed*, por exemplo, guarda-se a soma das quatro restrições tipo *fixed* dividido por 4. Na *hard*, as duas restrições tipo *hard* dividido por 2 e na *soft*, as três deste tipo dividido por 3. Como os valores acumulativos de cada restrição pode chegar no máximo a 100, o valor de cada variável dessa função também atinge no máximo esse valor. O cálculo do valor de *fitness* de um indivíduo é dado pelo resultado da equação $fixed + (hard * 0.7) + (soft * 0.3)$. Logo o valor de *fitness* máximo que um indivíduo pode atingir é de 200. Esse valor então é que distingue os indivíduos bons dos ruins, os quais darão continuidade ao processo evolutivo. Exemplificando, caso tenhamos um indivíduo cujo a média da soma dos valores das restrições tipo *fixed* foi 90, a média das do tipo *hard* foi 70 e as do tipo *soft* foi 100, o valor de *fitness* para este indivíduo será de 169.

7 Discussão

Existem diversas soluções de implementação que possam atender as estratégias para obter um plano de estudos ideal levantadas por Carita et al (1998). A seguir segue a descrição de como foram implementadas as soluções utilizadas por este trabalho para atender aos critérios de avaliação para o plano ideal presente na Tabela 2. Com isso, pode-se analisar a conformidade entre os planos de estudos gerados e as restrições levantadas pelos especialistas citadas na Sessão 1.

Para atender a restrição 1 da Tabela 2, que é não gastar todo o tempo de estudo em uma só disciplina, foi analisado cada período do dia não vazio (que possua pelo menos uma matéria alocada) presente no plano de estudos. Caso um período possua mais de uma matéria alocada então é atribuído 100 a um valor acumulativo ou 0 caso contrário. Esse valor acumulativo vai somando todos os valores para cada período não vazio. A pontuação para esta restrição é dada pela divisão do valor acumulativo pela quantidade de períodos analisados. Esta restrição foi classificada como baixa prioridade, ou seja, tipo *soft*.

A restrição 2, da Tabela 2, fala que algumas disciplinas necessitam de mais tempo que outras. Para isso, planos que alocam mais tempo para matérias que possuam uma maior dificuldade ou o contrário, menos tempo para as mais fáceis, deveriam ser privilegiados. Logo, na solução proposta, teve que ser verificado a porcentagem ideal de

horas que cada matéria deveria ter no plano de estudos. A figura 5 traz um exemplo hipotético do cálculo que mostra a porcentagem do quão ideal estão distribuídas as horas alocadas à matéria matemática em relação às outras matérias do subitem a. A primeira etapa é verificar a porcentagem ideal de horas para cada matéria que deverá ser alocado em relação ao total de horas disponível em todo o plano de estudos. O subitem b da figura 5 mostra como é feito o cálculo dessa porcentagem. Basicamente multiplica-se a dificuldade da matéria por cem e divide este resultado pelo total de horas que todo o plano de estudos tem disponível. A segunda etapa exemplificada no subitem c da figura 5 é ver quanto em horas, essa porcentagem anterior representa em relação ao total de horas disponível. A próxima etapa é calcular o quão próximo do ideal o plano está disponibilizando em horas de estudo para cada matéria (ver subitem d da tabela 1). Logo, as horas alocadas à matemática no exemplo da figura 5 estão a 45% do número de horas que seriam ideais para se estudar essa matéria. No final o valor acumulativo para esta restrição, será a soma de todas as porcentagens de cada matéria dividido pela quantidade de matérias disponíveis. Esta restrição foi classificada como tipo *hard*.

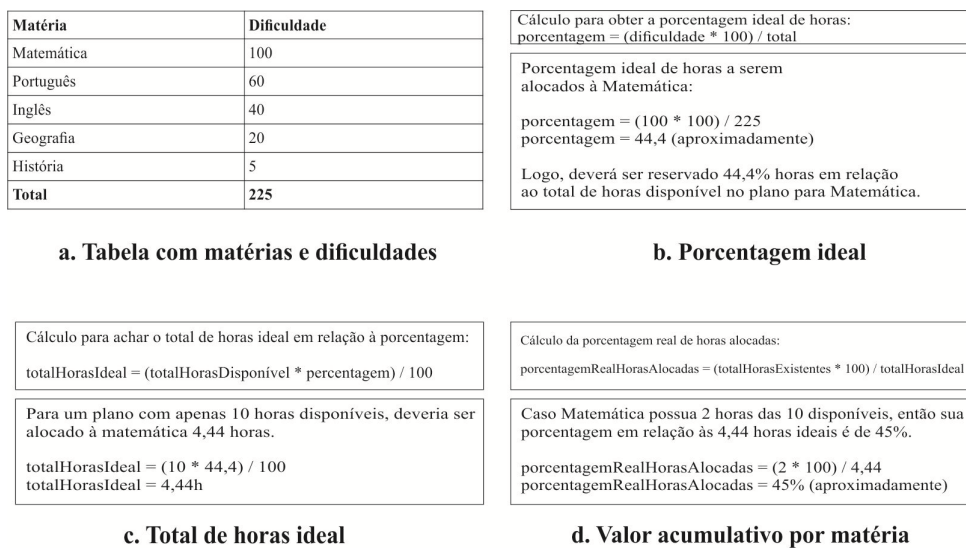


Figura 5. Um exemplo do cálculo da lógica da restrição 2, da Tabela 2.

No item 3, da Tabela 2, foi verificado se o plano de estudos está alocando as matérias com nível médio de dificuldade primeiro, seguindo-se das mais difíceis e finalizado com as mais fáceis. A classificação da tabela 3 foi utilizada para distinguir os níveis das matérias. Na solução, dividiu-se a quantidade de dias presentes no plano por três, ou seja, o primeiro terço da quantidade de períodos do plano de estudos foi reservado para os de nível médio, o outro terço para os difíceis e o último terço para os fáceis. Para o primeiro terço, foi verificado a porcentagem de matérias com nível médio presentes. O mesmo processo foi aplicado para os outros dois terços, onde o segundo terço foi em relação as matérias difíceis e o último em relação as fáceis. O valor acumulativo desta restrição é a soma das três porcentagens dividido por três. Esta restrição foi classificada como *soft*.

Na restrição 4, da Tabela 2, como foi descrito na sessão de Implementação, foi restringido a alocação de horas por período para 5 horas, logo, haverá sempre no mínimo 1 hora para descanso em cada período do dia.

Por último, para a restrição 5 é verificado se todas as matérias cadastradas pelo usuário estão presentes no plano de estudos retornando o valor 100 caso verdade. É atribuído o valor 0 caso esteja faltando alguma matéria no plano de estudos. Portanto para atender este critério, todas as matérias deverão estar presentes no plano de estudos almejado, caso contrário este plano não será o ideal. Com isso, esta restrição foi classificada como tipo *fixed*.

Assim, pode-se provar que todas as restrições da Carita et al (1998) consideradas por este trabalho para se gerar o plano de estudos ideal para cada usuário foram atendidas.

9 Conclusão

O principal achado deste trabalho foi conseguir gerar planos de estudos automatizados, implementando as restrições levantadas pelos especialistas Carita et al (1998) para se gerar o melhor plano de estudos, considerando suas preferências e utilizando algoritmos genéticos como estratégia de solução. Após diversas pesquisas, não foram encontrados, até o momento, nenhum outro trabalho correlato, que se pudesse realizar um estudo comparativo. Mas como prova da eficácia e validade do trabalho, foi descrito uma prova de conceito das estratégias de solução utilizadas para atender cada restrição levantada pelos especialistas referenciados, mostrando que estas estratégias estavam em conformidade com as restrições citadas.

Como forma de comprovação da automatização da geração das grades de horários em um tempo finito, foram executados diversos casos de testes considerando diferentes perfis de alunos e planos de estudos contendo 7, 15 e 30 dias. Cada caso de teste foi executado 50 vezes. O tempo médio para cada execução ficou em torno de 16 segundos. Não foi encontrada muita discrepância em relação ao desempenho do valor de *fitness* considerando diferentes perfis de usuários. Em relação à quantidade de dias, o *fitness* variou somente cerca de 10 pontos entre 7 e 30 dias. O resultado mais significativo em termos de discrepância de valores foi em relação ao tempo, pois a média deste para se gerar um plano de estudos de 15 dias, por exemplo, foi em torno de 40 segundos enquanto que para um plano de 30 dias essa média chegou a 1 min e 10seg. Essa análise pode ser visualizada na figura 6.

Os principais obstáculos aqui encontrados foram: (i) a dificuldade de encontrar na literatura especialistas que definissem claramente o que deveria ser feito para se gerar melhores grades, (ii) o estorvo na forma como estas restrições deveriam ser implementadas devido à subjetividade das suas interpretações e (iii) a construção do algoritmo em si já que o indivíduo é uma estrutura de dados complexa e não apenas um *array* de bits, acarretando assim na escrita de muitos códigos que não puderam ser aproveitados pelas classes pré-implementadas do ECJ, como a construção do gene e a forma de cruzamento e mutação, por exemplo.

Contudo, foi possível visualizar o funcionamento dos Algoritmos Genéticos para resolver problemas de *scheduling* e também, a eficácia na implementação das estratégias utilizadas para atenderem aos critérios dos especialistas aqui adotados para se gerar as grades de estudos ideais.

Nos trabalhos futuros, seria relevante utilizar outras técnicas de computação para a geração dos planos ideais considerando os mesmos parâmetros aqui utilizados, e assim

realizar uma comparação com os resultados deste trabalho, ou utilizar outras restrições levantadas por autores diferentes para se gerar melhores grades. Também, pode-se aplicar outras estratégias de solução para a implementação das restrições aqui produzidas, bem como construir uma interface gráfica onde os usuários pudessem acessar e construir suas grades de estudos de forma mais simples e amigável.

Qtd dias	7	15	30
Média de tempo	16seg	40seg	70seg
Média do <i>Fitness</i>	164,44	160,13	153,18

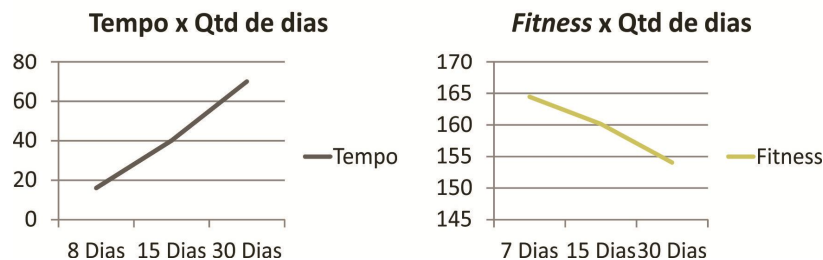


Figura 6. Diferença de desempenho em relação a quantidade de dias presentes no plano, considerando a média do tempo de execução e a média do valor de *Fitness* em 50 execuções.

Referências

- Carita, A.; Silva, A.; Monteiro, A. & Diniz, T. (1998). Como ensinar a estudar. Lisboa: Editorial Presença.
- Carvalho, M., B. (2011). Algoritmos Bio-inspirados para Minimização do Makespan do Problema de Escalonamento de Produção. Tese de doutorado, Faculdade de Engenharia elétrica e de computação, UNICAMP.
- Darwin, C. (1859). The Origin of Species, John Murray, Albermale Street, London.
- Ellingsen, K. e Penalzoza, M. (2003). A Genetic Algorithm Approach for Finding a Good Course Schedule.
- Davis, L. *Handbook of Genetic Algorithms*. (1991). Van Nostrand Reinhold, New York.
- Luke, S. (2002). ECJ: A Java-based evolutionary computation and genetic programming system, <http://cs.gmu.edu/~eclab/projects/ecj/>.
- MENDEL, G. (1865) Ensaios com Plantas Híbridas, Sociedade de História Natural de Brno.
- PACHECO, M., A., C. (2005) Algoritmos Genéticos: Princípios e Aplicações. ICA – Laboratório de Computação Aplicada – PUC-Rio.
- Vasconsellos, R. (2011). Um algoritmo genético para o problema de scheduling de projetos com restrição de recursos – Uma solução com gerenciamento de risco.
- Zimmerman, B.; Greenberg, D. & Weinstein, C. (1994). Self-Regulating Academic Study Time: A Strategy Approach. In D. Schunk & B. Zimmerman (Edt.s). Self-Regulation of Learning and Performance (pp. 181-199). New Jersey: Lawrence Erlbaum Associates.