

1 RESULTADOS OBTIDOS

Após o sistema implementado foram realizadas uma bateria de testes para que os valores parâmetros iniciais fossem definidos, os dados utilizados tem em sua composição o total de 10 salas e um curso com o total de 46 disciplinas e duas respectivas obrigаторiedades, serão utilizados como parâmetros iniciais os dados representados na Figura 1, no caso uma população com 140 indivíduos, 60% de crossover 20% de mutação, para entender melhor o andamento do algoritmo inicialmente será utilizada 400 gerações, o elitismo acontece em todos os testes realizados. Estes dados foram tirados como base na conclusão do trabalho de (CISCON, 2006). Os testes realizados tiveram uma máquina com a seguinte configuração, Intel(R) Core(TM) i5 3.40GHz com 16Gb de memória RAM sob o sistema operacional Windows 7 64bit.

Figura 1 – Parâmetros utilizados para o teste inicial

Parâmetros Algoritmo

Crossover: 0.6

Mutação: 0.3

Tamanho População: 140

Número de Gerações: 500

Elitismo: ☒ Sim ☐ Não

Fonte: Desenvolvido pelo autor

Ao realizar o primeiro teste, que é realizado com o uso de todos os parâmetros iniciais foi identificado que o algoritmo fica preso em planícies, ou seja fica preso por muito tempo em uma solução sem evolução da mesma. Neste caso podemos notar no gráfico representado pela Figura X que o algoritmo no começo tem uma evolução rápida, mas em um determinado ponto ele perde seu poder de evolução e começa a caminhar devagar em soluções que tem como valor da função objetiva em torno de 50% o que não faz da solução, uma solução que atenda ao objetivo deste trabalho.

Tentando identificar as possíveis causas desta aceleração de evolução do algoritmo foram realizados teste onde são retirados alguns parâmetros para identificar o que está acontecendo. Primeiramente foi realizado o teste onde o parâmetro de mutação não é utilizado. A figura X mostra o desempenho do algoritmo nesta situação. Podemos observar que em determinado ponto o algoritmo evolui rapidamente, porém o mesmo entra em uma planície e não consegue mais prosseguir com sua evolução, isso se deve ao fato de um super-indivíduo ter dominado a população passando as características dos seus genes para todos os indivíduos da população. Isto acontece pois não ocorre a mutação que é responsável pela troca de genes aleatoriamente de um indivíduo.

Logo em seguida foi realizado o teste onde o parâmetro de crossover não é utilizado. A figura X representa a evolução do algoritmo de acordo com a velocidade com que acontece a evolução de acordo com as mutações podemos observar que, a evolução através da mutação é bem lenta e o algoritmo encontra várias planícies durante a evolução tornando assim o processo para encontrar a solução bem demorado.

Comparando os gráficos três gráficos gerados apresentados na figura X podemos observar que o crossover é responsável por acelerar a evolução e a mutação faz com que o algoritmo

tenha uma nova diversidade de indivíduos para que não ocorra a parada do mesmo em planícies.

Para acelerar a evolução do algoritmo como solução foi implementada na mutação, a mutação com melhoria genética onde o algoritmo ao realizar este operador genético ao invés de apenas realizar uma troca genética, ele procura fazer a troca dos genes, enviando as características de um gene para um lugar mais adequado que respeite as restrições do problema.

A figura X representa a comparação entre o primeiro gráfico gerado com o parâmetro de mutação com a nova solução gerada através da utilização da melhoria genética. Podemos observar uma grande melhoria na evolução do algoritmo.

Inicialmente o método de mutação com melhoria genética troca apenas um dos genes, foram então realizados testes trocando um gene, quatro genes e 10% dos genes. O gráfico X mostra a comparação entre os três testes realizados. Podemos observar que quantos mais genes trocados durante a mutação maior é a evolução do algoritmo.

Após a escolha do número de genes a serem trocados durante a melhoria genética, foi realizado um novo teste que é a utilização do operador genético de mutação com o de crossover o resultado é apresentado na Figura X. Falar o que aconteceu.

Logo em seguida foi aumentado o número de gerações para descobrir se o algoritmo encontraria a melhor solução e o mesmo encontro em X. O resultado pode ser encontrado no APENDICE X

Após esta bateria de testes para descobrir sobre os passos dados pelo algoritmo foi realizado um teste com a carga completa que é XXXXXX.

Com o total de X horas e X gerações o algoritmo encontra a melhor solução que se encontra no APENDICE X.

REFERÊNCIAS

- AARTS, E.; KORST, J. Simulated annealing and boltzmann machines. New York, NY; John Wiley and Sons Inc., 1988.
- ARMENTANO, V. A.; BRANCHINI, R. M. Uma introdução à busca tabu. 2013.
- CAELUM. *Apostila do curso FJ-11 - Java e Orientação a Objetos*. 2013. Disponível em: <<http://www.caelum.com.br/apostila-java-orientacao-objetos>>. Acesso em: 29 set. 2013.
- CARTER, M. W.; TOVEY, C. A. When is the classroom assignment problem hard? *Operations Research*, INFORMS, v. 40, n. 1-Supplement-1, p. S28–S39, 1992.
- CISCON, L. A. O problema de geração de horários: Um foco na eliminação de janelas e aulas isoladas. 2006.
- DARWIN, C. On the origin of species by means of natural selection. 1859. *See also*: <http://www.literature.org/authors/darwin-charles/the-origin-of-species>, 1968.
- EVANS, J. R.; MINIEKA, E. *Optimization algorithms for networks and graphs*. [S.l.]: CRC Press, 1992.
- EVEN, S.; ITAI, A.; SHAMIR, A. On the complexity of time table and multi-commodity flow problems. In: IEEE. *Foundations of Computer Science, 1975., 16th Annual Symposium on*. [S.l.], 1975. p. 184–193.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, Elsevier, v. 13, n. 5, p. 533–549, 1986.
- GÓES, A. R. T. *Otimização na Distribuição da Carga Horária de Professores: método exato, método heurístico, método misto e interface*. Tese (Doutorado) — Dissertação de Mestrado, UFPR, 2005.
- GOLBARG, M. C.; LUNA, H. P. L. Otimização combinatória e programação linear. *Rio de Janeiro: Campus*, 2000.
- GOLDBERG, D. Genetic algorithms in optimization, search and machine learning. *Addison Wesley, New York. Eiben AE, Smith JE (2003) Introduction to Evolutionary Computing. Springer. Jacq J, Roux C (1995) Registration of non-segmented images using a genetic algorithm. Lecture notes in computer science*, v. 905, p. 205–211, 1989.
- HAMAWAKI, C. D. L. Geração automática de grade horária usando algoritmos genéticos: o caso da faculdade de engenharia elétrica da ufu. 2011.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: U Michigan Press, 1975.
- JONG, K. A. D. *An analysis of the behavior of a class of genetic adaptative systems*. Tese (Doutorado) — Dissertation Abstracts International, 36(10),5140B. (University Micro lms No. 76-9381), 1975.
- KIRKPATRICK, S.; JR., D. G.; VECCHI, M. P. Optimization by simulated annealing. *science*, Washington, v. 220, n. 4598, p. 671–680, 1983.
- LACERDA, E. G. de; CARVALHO, A. de. Introdução aos algoritmos genéticos. *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*, v. 1, p. 99–148, 1999.
- LOBO, E. L. M. Uma solução do problema de horário escolar via algoritmo genético paralelo. *Centro Federal de Educação Tecnológica de Minas Gerais*, 2005.

- LUCAS, D. C. Algoritmos genéticos: um estudo de seus conceitos fundamentais e aplicação no problema de grade horária. *Monografia de Graduação*, 2000.
- LUEZUTE, R. M.; KRIPKA, K. M. Simulated annealing aplicado ao problema de alocação de salas com deslocamentos mínimos. 2013.
- MARINHO, E. *Heurísticas busca tabu para o problema de programação de tripulações de ônibus urbano*. Tese (Doutorado) — Master's Thesis, Universidade Federal Fluminense, 2005.
- MENDES, W. *AngularJS um framework para facilitar sua vida*. 2013. Disponível em: <<http://www.slideshare.net/WilsonMendes/angularjs-um-framework-para-facilitar-sua-vida>>. Acesso em: 29 set. 2013.
- MICHALEWICZ, Z.; SCHOENAUER, M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, MIT Press, v. 4, n. 1, p. 1–32, 1996.
- MITCHELL, M. An introduction to genetic algorithms (complex adaptive systems). A Bradford Book, 1998.
- NASCIMENTO, A. S.; SILVA, R. M. S.; ALVARENGA, G. B. Uma aplicação de simulated annealing para o problema de alocação de salas. *INFOCOMP Journal of Computer Science*, v. 4, n. 3, p. 59–66, 2005.
- NORONHA, T. Uma abordagem sobre estratégias metaheurísticas. 2000. *Projeto Orientado—Universidade Federal do Rio Grande do Norte. Rio Grande do Norte*. Disponível em: <http://www.sbc.org.br/reic/edicoes/2001e1/cientificos/UmaAbordagemSobreEstrategiasMetaheurísticas.pdf>, 2003.
- OLIVEIRA, A. C. de. Uso do algoritmo genético e recozimento simulado para o problema de alocação de salas. *Monografia, Departamento de Ciência da Computação, Universidade Federal de Lavras*, 2006.
- OLIVEIRA, H. Algoritmo evolutivo no tratamento do problema de roteamento de veículos com janela de tempo. *Monografia, Departamento de Ciência da Computação, Universidade Federal de Lavras*, 2005.
- PINHEIRO, P.; OLIVEIRA, J. A. Um ambiente de apoio a construção de horário escolar na web: modelagem, implementação e aplicação nas escolas de ensino médio. *XXXIII Simpósio Brasileiro de Pesquisa Operacional*, Campos do Jordão, SP, 2001.
- PLAY! *The High Velocity Web Framework For Java and Scala*. 2013. Disponível em: <<http://www.playframework.com/>>. Acesso em: 29 set. 2013.
- POSTGRESQL. *Sobre o PostgreSQL*. 2013. Disponível em: <<http://www.postgresql.org.br/sobre>>. Acesso em: 29 set. 2013.
- RAO, S. S. *Optimization : theory and applications*. New Delhi: Wiley Eastern, 1984. ISBN 0-85226-780-0. Disponível em: <<http://opac.inria.fr/record=b1092847>>.
- RAUPP, M. Introdução à otimização linear. *LNCC, Rio de Janeiro. Notas de Aulas, Curso de Verão LNCC*, 2003.
- SCHAERF, A. A survey of automated timetabling. *Artificial intelligence review*, Springer, v. 13, n. 2, p. 87–127, 1999.
- SILVA, A. Estudo e implementação, mediante recozimento simulado, do problema de alocação de salas. *Monografia, Departamento de Ciência da Computação, Universidade Federal de Lavras*, 2005.
- SOUZA, M. J. F. Programação de horários em escolas: uma aproximação por metaheurísticas. *Rio de Janeiro*, 2000.

SOUZA, M. J. F.; MARTINS, A. X.; ARAÚJO, C. R. d. Experiências com simulated annealing e busca tabu na resolução do problema de alocação de salas. 2002.

STEIGLITZ, K.; PAPADIMITRIOU, C. H. Combinatorial optimization: Algorithms and complexity. *Prentice Hall, New Jersey., UV Vazirani (1984). On two geometric problems related to the travelling salesman problem. J. Algorithms*, v. 5, p. 231–246, 1982.

SUBRAMANIAN, A. et al. Aplicação da metaheurística busca tabu na resolução do problema de alocação de salas do centro de tecnologia da ufpb. *Anais do XXVI Encontro Nacional de Engenharia de Produção*, p. 1, 2006.

TIMÓTEO, G. T. S. *Desenvolvimento de um Algoritmo Genético para a Resolução do Timetabling*. 2005.

W3C. *HTML 4.01 Specification*. 2013 a. Disponível em: <<http://www.w3.org/TR/html4>>. Acesso em: 29 set. 2013.

W3C. *Cascading Style Sheets*. 2013 b. Disponível em: <<http://www.w3.org/Style/CSS>>. Acesso em: 29 set. 2013.

W3SCHOOLS. *JavaScript Tutorial*. 2013. Disponível em: <<http://www.w3schools.com/js/>>. Acesso em: 29 set. 2013.