

# 1 REFERENCIAL TEÓRICO

## 1.1 Problemas de Otimização

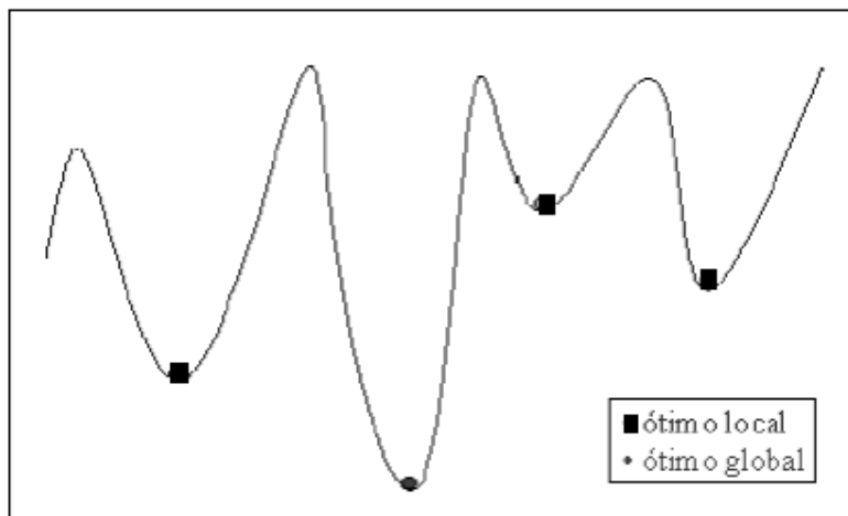
Otimização é o processo de encontrar a melhor solução, também chamada de solução ótima para um determinado problema (TIMÓTEO, 2005).

De acordo com (STEIGLITZ; PAPADIMITRIOU, 1982) a constituição de um problema de otimização se deve aos termos vizinhança, ótimo local e ótimo global. O termo vizinhança se trata de um subconjunto do conjunto do problema, ótimo local pode ser tratado como o melhor resultado em uma vizinhança, e ótimo global é a melhor solução encontrada no conjunto de acordo com a função objetivo.

Conforme cita (RAUPP, 2003), o problema de otimização combinatória pode ser denominado como a ação de maximizar ou minimizar uma função objetiva de diversas variáveis sujeita a um conjunto de restrições, dentro de contexto.

De acordo com a figura 1 pode se observar a relação entre ótimo local e ótimo global em um problema típico de minimização.

**Figura 1 – Representação de um problema de minimização com ótimos locais**



Fonte: (TIMÓTEO, 2005)

Segundo (STEIGLITZ; PAPADIMITRIOU, 1982), os problemas de otimização são divididos em duas categorias, problemas com variáveis contínuas e problemas com variáveis discretas. Problemas com variáveis discretas também podem ser conhecidos como Problemas de Otimização Combinatória (POC).

De acordo com (RAO, 1984) problemas do tipo POC tratam do estudo matemático para encontrar agrupamentos, arranjos ou a seleção ótima de objetos discretos, logo, não permitindo, a utilização de métodos clássicos de otimização contínua para sua resolução.

Segundo (GOLBARG; LUNA, 2000) a ocorrência de problemas de otimização combinatória podem acontecer em diversas áreas, projetos de sistemas de distribuição de energia elétrica, posicionamento de satélites, roteamento ou escalonamento de veículos, sequenciamento de genes e DNA, classificação de plantas e animais.

De acordo com (CISCON, 2006) em problemas de otimização combinatória, cujo universo de dados é grande e existe um grande número de combinações, o que torna inviável a

análise de todas soluções possíveis em um tempo adequado, utilizamos as heurísticas, também conhecidas como algoritmos heurísticos, que são métodos que compõem uma gama de soluções para problemas de otimização combinatória.

## 1.2 Heurística

O termo heurística é derivado do grego *heuriskein*, o que significa descobrir ou achar. De acordo com (TIMÓTEO, 2005) o significado da palavra em pesquisa operacional vai um pouco além da raiz etimológica. Segundo (STEIGLITZ; PAPADIMITRIOU, 1982), heurísticas são consideradas métodos de aproximação ou métodos de busca de solução, deve se levar em consideração que não exista uma garantia formal de seu desempenho e uma garantia de estas heurísticas que iram encontrar uma solução. As heurísticas, apesar de não garantirem encontrar a solução ótima para um problema, procuram por soluções consideradas de boa qualidade em um tempo computacional razoável.

Segundo (EVANS; MINIEKA, 1992) heurísticas são necessárias para implementação de problemas NP Difícil, caso deseje-se resolver tais problemas em um tempo razoável.

Ressalta-se que dentre as heurísticas, as chamadas meta-heurísticas, merecem especial atenção pois adotam técnicas para amenizar, a dificuldade que os métodos heurísticos têm de escapar dos ótimos locais. As meta-heurísticas podem partir em busca de regiões mais promissoras no espaço de soluções, além disto, as meta-heurísticas possuem grande abrangência, podendo ser aplicada à maioria dos problemas de otimização combinatória. (NASCIMENTO; SILVA; ALVARENGA, 2005)

Segundo (OLIVEIRA, 2006) uma heurística é a instanciamento de uma meta-heurística, ou seja, a aplicação da mesma em um problema específico de otimização.

Como exemplos de meta-heurísticas temos Busca Tabu (*Tabu Search*), Otimização por Colônias de Formigas (*Ant Colony Optimization*), Recozimento Simulado (*Simulated Annealing*) e Algoritmo Genético (*Genetic Algorithm*).

### 1.2.1 Busca Tabu

A metaheurística BT foi inicialmente desenvolvida por (GLOVER, 1986) como uma proposta de solução para problemas de programação inteira. A partir de então, o autor formalizou esta técnica e publicou uma série de trabalhos contendo diversas aplicações da mesma. A metaheurística BT utiliza uma lista contendo o histórico da evolução do processo de busca, de modo a evitar ciclagem; incorpora uma estratégia de balanceamento entre os movimentos aceitos, rejeitados e aspirados; e adota procedimentos de diversificação e intensificação para o processo de busca.

(SOUZA, 2000) e (WHITE; XIE; ZONJIC, 2004) ressaltam a existência de um mecanismo, relacionado com a Lista Tabu, que anula o status tabu de um movimento, denominado função de aspiração. Se um movimento pode proporcionar uma melhora considerável da função objetivo, então o status tabu é abandonado e a solução resultante é aceita como potencial vizinho.

### 1.2.2 Algoritmo da colônia de formigas

Achar uma referencia sobre o algoritmo da colonia

### 1.2.3 Recozimento Simulado

Técnica de busca local probabilística, proposta originalmente por (KIRKPATRICK; JR.; VECCHI, 1983), que se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. Isto é, conforme (NORONHA, 2003) em analogia a física da matéria: levando um cristal a sua temperatura de fusão, as moléculas estão desordenadas e se agitam livremente. Ao resfriar-se a amostra de maneira infinitamente lenta, as moléculas vão adquirir a estrutura cristalina estável que tem um nível de energia mais baixo possível. Conforme (AARTS; KORST, 1988) a analogia com a otimização (combinatória ou não) é bastante direta. Os estados da matéria são as soluções realizáveis, a quantidade objetiva substitui a energia, os estados metaestáveis da matéria sendo ótimos locais e a estrutura cristalina corresponde ao ótimo global. Segundo (REEVES, 1993), a temperatura  $T$  assume, inicialmente, um valor elevado  $T_0$  e o procedimento pára quando a temperatura chega a um valor próximo de zero e nenhuma solução que piore o valor da função objetivo é mais aceita, isto é, quando o sistema está estável.

Mais informações em (REEVES, 1993) e (KIRKPATRICK; JR.; VECCHI, 1983).

### 1.2.4 Algoritmos genéticos

Conforme cita (OLIVEIRA, 2005), os algoritmos genéticos foram introduzidos por (HOLLAND, 1975), com intuito de aplicar a teoria da evolução das espécies elaborada por (DARWIN, 1968) utilizando os conceitos da evolução biológica como genes, cromossomos, cruzamento, mutação e seleção na computação procurando explicar rigorosamente processos de adaptação em sistemas naturais e desenvolver sistemas artificiais (simulados em computador) que mantenham os mecanismos originais, encontrados em sistemas naturais.

Segundo (OLIVEIRA, 2005), o processo de evolução executado por um algoritmo genético corresponde a um procedimento de busca no espaço de soluções potenciais para o problema e, como enfatiza (MICHALEWICZ; SCHOENAUER, 1996), esta busca requer um equilíbrio entre dois objetivos aparentemente conflitantes: a procura das melhores soluções na região que se apresenta promissora ou fase de intensificação e a procura de outra região ou exploração do espaço de busca, também conhecida como diversificação.

Ainda segundo (OLIVEIRA, 2005), os algoritmos genéticos têm se mostrado ferramentas poderosas para resolver problemas onde o espaço de busca é muito grande e os métodos convencionais se mostraram ineficientes.

Mitchel (MITCHELL, 1998) cita que a terminologia biológica é muito importante para a compreensão do funcionamento dos algoritmos genéticos. Eis os principais termos:

- Cromossomo: estrutura que representa uma determinada característica da solução ou a própria solução;
- Gene: característica particular de um cromossomo. O cromossomo é composto por um ou mais genes.
- Alelo: valor de determinado gene;
- Locus: determinada posição do gene no cromossomo;
- Genótipo: estrutura que codifica uma solução. Um genótipo pode ser formado por um ou mais cromossomos;
- Fenótipo: decodificação ou o significado da estrutura;
- Fitness: significa aptidão. O quanto o indivíduo é apto para determinado ambiente;

As principais características que diferenciam os algoritmos genéticos de métodos tradicionais são (GOLDBERG, 1989):

- Parâmetros: os algoritmos genéticos trabalham com a codificação dos parâmetros e não com os parâmetros propriamente;
- Número de soluções: os algoritmos genéticos trabalham com uma população de indivíduos (representando um conjunto de soluções) e não com uma única solução;
- Avaliação das soluções: os algoritmos genéticos utilizam informações de custo ou recompensa penalizando ou premiando determinadas características das soluções;
- Regras: os algoritmos genéticos utilizam regras probabilísticas e não determinísticas;

O algoritmo genético é uma forma da estratégia gerar-e-testar realizando os testes baseados nos parâmetros da evolução biológica. Uma desvantagem notável é a variação dos operadores genéticos do algoritmo em cada problema. Dessa forma, para resolução de determinado problema, torna-se necessário um estudo particular a respeito do mesmo.

O algoritmo genético atua sobre uma população fazendo com que esta evolua de acordo com uma função de avaliação. O funcionamento é iterativo iniciando com a geração de uma população inicial que pode ser aleatória ou não, seguida do processo de avaliação, seleção, cruzamento e mutação, que ocorre a cada iteração até que seja atingido algum critério de parada. Os passos gerais de um algoritmo genético são ilustrados na figura Figura XXXX. Cada passo pode ser realizado de várias maneiras e pode variar de problema para problema (TIMÓTEO, 2005).

Figura Etapas de um Algoritmo Genético Básico

### 1.3 TimeTabling

Segundo (LUEZUTE; KRIPKA, 2013) os problemas de programação de horários (PPH), também conhecidos como *TimeTabling*, são os problemas que mais se destacam nas organizações acadêmicas. De acordo com (SCHAERF, 1999) estes problemas são divididos em três categorias *school timetabling*, *course timetabling* e *examination timetabling*.

*School TimeTabling*: Se trata basicamente da geração de horários semanais, em escolas de segundo grau, onde deve-se evitar os choques entre os horários das disciplinas e que cada professor receba apenas uma turma para cada horário. Neste caso o aluno recebe um número fixo de disciplinas a serem cursadas.

*Course TimeTabling*: Diz respeito à alocação de aulas de uma universidade típica. Neste problema os alunos podem escolher as matérias em que vão se matricular, portando o problema tem como objetivo minimizar os possíveis choques entre as disciplinas, professores e horários disponibilizados pela instituição de ensino.

*Examination TimeTabling*: Aborda o problema de programação de horários dos exames da instituição, de maneira que, disciplinas que tenham alunos em comum, distanciem o máximo possível as datas dos exames.

Segundo (PINHEIRO; OLIVEIRA, 2001) o problema de programação de horários vem sendo abordado desde a década de 60, sendo que os primeiros trabalhos a se destacarem foram realizados na década de 80.

O Problema de Alocação de Salas (PAS) também conhecido como *Classroom Assignment* é tratado como parte do problema de programação de cursos universitários *course timetabling*. Segundo (MARINHO, 2005) varias instituições universitárias se deparam com o PAS durante o início de cada semestre letivo, este problema é considerado NP-Difícil por (EVEN; ITAI; SHAMIR, 1975) e (CARTER; TOVEY, 1992), com isto, a determinação da solução ótima do

problema, em um período de tempo aceitável se torna uma tarefa difícil.

Segundo (LUVEZUTE; KRIPKA, 2013) o problema deve considerar que as disciplinas dos cursos universitários já tenham seus horários de início e de término definidos. O problema se resume então na alocação das disciplinas às salas desta universidade respeitando os horários destas disciplinas e as demais restrições exigidas.

Segundo (SOUZA, 2000) boa parte das universidades ainda resolvem este problema de forma manual, o que torna o processo árduo e demorado, podendo levar vários dias para ser concluído.

Uma vez que é de extrema dificuldade encontrar a solução ótima do PAS em tempo razoável, este problema é normalmente tratado através de técnicas heurísticas, que apesar de não garantirem encontrar a solução ótima do problema, são capazes de retornar uma solução de qualidade em um tempo adequado. As meta-heurísticas surgiram como uma alternativa para amenizar a dificuldade que os métodos heurísticos tem de escapar dos chamados ótimos locais.(NASCIMENTO; SILVA; ALVARENGA, 2005).

Segundo (EVEN; ITAI; SHAMIR, 1975) o PAS pertence a classe de Problemas de Otimização Combinatória (POC).

#### **1.4 Trabalhos Relacionados**

trabalho do marinho que usa tabu. trabalho da silvia que usa Recozimento Simulado (Simulated Annealing). trabalho da leonardo que usa Algoritmos Genéticos (AG). achar algum trabalho que utiliza o algoritmo das formigas. falar porque o trabalho do cara se assemelha ao meu trabalho.

## REFERÊNCIAS

- AARTS, E.; KORST, J. Simulated annealing and boltzmann machines. New York, NY; John Wiley and Sons Inc., 1988.
- 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.
- 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.
- 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.
- KIRKPATRICK, S.; JR., D. G.; VECCHI, M. P. Optimization by simulated annealing. *science*, Washington, v. 220, n. 4598, p. 671–680, 1983.
- LUVEZUTE, 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/UmaAbordagemSobreEstrategiasMetaheuristicas.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.
- REEVES, C. R. *Modern heuristic techniques for combinatorial problems*. [S.l.]: John Wiley & Sons, Inc., 1993.
- SCHAERF, A. A survey of automated timetabling. *Artificial intelligence review*, Springer, v. 13, n. 2, p. 87–127, 1999.
- SOUZA, M. J. F. Programação de horários em escolas: uma aproximação por metaheurísticas. *Rio de Janeiro*, 2000.
- 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.
- 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.
- WHITE, G. M.; XIE, B. S.; ZONJIC, S. Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, Elsevier, v. 153, n. 1, p. 80–91, 2004.