

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

Aplicação de Meta-Heurísticas à Engenharia de Produção

João Carlos Zayatz

TCC-EP-51-2012

Maringá - Paraná
Brasil

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

Aplicação de Meta-Heurísticas em Engenharia de Produção

João Carlos Zayatz

TCC-EP-51-2012

Trabalho de Conclusão de Curso apresenta ao Curso de Engenharia de Produção, do Centro de Tecnologia, da Universidade Estadual de Maringá – UEM.
Orientador(a): Prof^ª. Gislaine Camila Lapasini Leal

**Maringá - Paraná
2012**

DEDICATÓRIA

A minha família, amigos e professores.

AGRADECIMENTOS

A Deus, a minha família, a todos os meus amigos, colegas de trabalho e professores.

RESUMO

O presente trabalho apresenta Meta-heurísticas como alternativa de soluções para problemas em Engenharia de Produção, nos casos em que a complexidade aponta para a utilização de métodos aproximativos, que nem sempre retornam a solução ótima, mas sim uma resposta aceitável em tempo computacional viável. Três métodos são abordados: Algoritmos Genéticos, *Simulated Annealing* e *Particle Swarm*. Em um primeiro momento, é realizada a apresentação dos métodos, com a revisão da literatura. Em seguida, é demonstrada a aplicabilidade dos métodos em Engenharia de Produção, por meio de pesquisa bibliométrica. Por fim, uma implementação simples em código de programação exemplifica o funcionamento de uma Meta-heurística na prática.

Palavras-chave: Meta-heurística, Algoritmos Genéticos, *Particle Swarm*, *Simulated Annealing*.

SUMÁRIO

| | |
|--|------|
| LISTA DE ILUSTRAÇÕES | VIII |
| LISTA DE TABELAS | IX |
| LISTA DE ABREVIATURAS E SIGLAS | X |
| LISTA DE SÍMBOLOS | XI |
| 1 INTRODUÇÃO..... | 1 |
| 1.1 JUSTIFICATIVA..... | 2 |
| 1.2 DEFINIÇÃO E DELIMITAÇÃO DO PROBLEMA | 2 |
| 1.3 OBJETIVOS..... | 3 |
| 1.3.1 <i>Objetivo geral</i> | 3 |
| 1.3.2 <i>Objetivos específicos</i> | 4 |
| 1.4 METODOLOGIA | 4 |
| 2 REVISÃO LITERÁRIA..... | 6 |
| 2.1 A ABORDAGEM QUANTITATIVA E A PESQUISA OPERACIONAL..... | 6 |
| 2.2 MODELAGEM MATEMÁTICA..... | 9 |
| 2.3 OTIMIZAÇÃO GLOBAL | 10 |
| 2.4 OTIMIZAÇÃO COMBINATÓRIA | 11 |
| 2.5 A TEORIA DE N-COMPLETUDE..... | 13 |
| 2.6 MÉTODOS HEURÍSTICOS | 16 |
| 2.7 META-HEURÍSTICAS | 19 |
| 2.7.1 <i>Algoritmos Genéticos (AG)</i> | 21 |
| 2.7.2 <i>Representação Genética ou Codificação</i> | 22 |
| 2.7.3 <i>Geração da População Inicial</i> | 24 |
| 2.7.4 <i>Função de Aptidão</i> | 25 |
| 2.7.5 <i>Seleção de Indivíduos</i> | 26 |
| 2.7.6 - <i>Seleção pelo Método da Roleta</i> | 27 |
| 2.7.7 - <i>Seleção pelo método do torneio</i> | 28 |
| 2.7.8 - <i>Seleção pelo método da amostragem universal estocástica</i> | 29 |
| 2.7.9 - <i>Outros métodos de seleção</i> | 29 |
| 2.7.10 <i>Operadores Genéticos</i> | 30 |
| 2.7.11 <i>Cruzamento</i> | 30 |
| 2.7.12 - <i>Mutação</i> | 32 |
| 2.7.13 - <i>Parâmetros para algoritmos genéticos</i> | 32 |
| 2.8 <i>SIMULATED ANNEALING</i> | 33 |
| 2.8.1 <i>Descrição do Método</i> | 34 |
| 2.8.2 <i>Parâmetros para Simulated Annealing</i> | 36 |
| 2.9 <i>PARTICLE SWARM</i> | 37 |
| 2.9.1 <i>Algoritmo de otimização PSO</i> | 38 |
| 2.9.2 <i>Parâmetros de Confiança para um PSO</i> | 40 |
| 2.9.2 <i>Função Aleatória e Aceleração da Distância</i> | 41 |
| 2.9.3 <i>Limitador de Velocidade e Peso Inercial</i> | 41 |
| 3.1 CONTEXTUALIZAÇÃO | 43 |
| 3.2 META-HEURÍSTICAS UTILIZADAS EM ENGENHARIA DE PRODUÇÃO | 44 |
| 3.3 COLETA DE DADOS..... | 46 |
| 3.4 RESULTADOS E DISCUSSÕES..... | 46 |
| 3.4.1 <i>Aplicação de Meta-heurísticas em Engenharia de Produção</i> | 46 |
| 3.4.2 <i>Aplicação de Meta-heurísticas nas áreas da Engenharia de Produção</i> | 52 |
| 4 APLICAÇÃO PRÁTICA DE META-HEURÍSTICA..... | 58 |
| 4.1 ELEIÇÃO DO PROBLEMA | 58 |
| 4.2 META-HEURÍSTICA APLICADA AO PROBLEMA ELEITO | 59 |
| 4.3 TESTES DE VALIDAÇÃO | 64 |
| 4.4 RESULTADOS E DISCUSSÃO | 65 |
| REFERÊNCIAS | 69 |

ANEXO A..... 84
APÉNDICE A 89

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| FIGURA 1 – FLUXOGRAMA DA METODOLOGIA | 5 |
| FIGURA 2 – A ABORDAGEM DA ANÁLISE QUANTITATIVA | 7 |
| FIGURA 3 – FUNÇÃO CONVEXA | 9 |
| FIGURA 4 – FUNÇÃO OBJETIVO COM ÓTIMO LOCAL E GLOBAL | 11 |
| FIGURA 5 – DOIS TRAJETOS ENTRE 14 CIDADES | 12 |
| FIGURA 6 – RELAÇÃO ENTRE AS CLASSES DE PROBLEMAS | 15 |
| FIGURA 7 – PROBLEMA DE OTIMIDADE GLOBAL COM SOLUÇÃO INICIAL PARA UM ALGORÍTIMO HEURÍSTICO DE MELHORAMENTO | 19 |
| FIGURA 8 – SELEÇÃO PELO MÉTODO DA ROLETA | 28 |
| FIGURA 9 – SELEÇÃO PELO MÉTODO DA AMOSTRA UNIVERSAL ESTOCÁSTICA..... | 29 |
| FIGURA 10 – CRUZAMENTO DE UM PONTO..... | 31 |
| FIGURA 11 – CRUZAMENTO MULTIPONTOS..... | 31 |
| FIGURA 12 – CRUZAMENTO UNIFORME..... | 32 |
| FIGURA 13 – OPERAÇÃO DE MUTAÇÃO | 33 |
| FIGURA 14 – REVOADA DE PÁSSAROS | 38 |
| FIGURA 15 – GRÁFICO DO NÚMERO DE PUBLICAÇÕES POR ANO | 47 |
| FIGURA 16 – GRÁFICO DO NÚMERO DE PUBLICAÇÕES POR FONTE..... | 48 |
| FIGURA 17 – GRÁFICO DAS LINGUAGENS DE PROGRAMAÇÃO UTILIZADAS | 48 |
| FIGURA 18 – GRÁFICO DAS META-HEURÍSTICAS UTILIZADAS..... | 49 |
| FIGURA 19 – GRÁFICO DAS ÁREAS DE APLICAÇÃO..... | 50 |
| FIGURA 20 – APLICAÇÃO DE ALGORITMOS GENÉTICOS EM ENGENHARIA DE PRODUÇÃO | 50 |
| FIGURA 21 – APLICAÇÃO DE <i>SIMULATED ANNEALING</i> EM ENGENHARIA DE PRODUÇÃO..... | 51 |
| FIGURA 22 – APLICAÇÃO DE <i>PARTICLE SWARM</i> EM ENGENHARIA DE PRODUÇÃO | 52 |
| FIGURA 23 – GRÁFICO DA APLICAÇÃO DE META-HEURÍSTICAS EM À ENGENHARIA DE OPERAÇÕES E PROCESSOS DE PRODUÇÃO..... | 52 |
| FIGURA 24 – GRÁFICO DA APLICAÇÃO DE META-HEURÍSTICAS EM À LOGÍSTICA..... | 55 |
| FIGURA 25 – GRÁFICO DA APLICAÇÃO DE META-HEURÍSTICAS EM ENGENHARIA DA QUALIDADE | 56 |
| FIGURA 26 – PCV | 59 |
| FIGURA 27 – PRIMEIRA EVOLUÇÃO DO AG..... | 63 |
| FIGURA 28 – SEGUNDA EVOLUÇÃO DO AG..... | 63 |
| FIGURA 29 – RESULTADO DO AG APÓS A ÚLTIMA EVOLUÇÃO..... | 64 |

LISTA DE TABELAS

| | |
|---|----|
| TABELA 1 - PSEUDOCÓDIGO PARA UM AG | 22 |
| TABELA 2 - VALOR DE APTIDÃO PARA SEIS INDIVÍDUOS DE UMA POPULAÇÃO | 26 |
| TABELA 3 - VALOR DE APTIDÃO PARA SEIS INDIVÍDUOS DE UMA POPULAÇÃO E APTIDÃO RELATIVA..... | 27 |
| TABELA 4 - VALOR DE APTIDÃO PARA SEIS INDIVÍDUOS DE UMA POPULAÇÃO | 28 |
| TABELA 5- SIMULATED ANNEALING EM PSEUDOCÓDIGO..... | 36 |
| TABELA 6 - <i>PARTICLE SWARM</i> EM PSEUDOCÓDIGO..... | 39 |
| TABELA 7 - EXECUÇÃO DO AG PARA 7 A 12 VÉRTICES DO PCV | 65 |
| TABELA 8 - PESQUISA BIBLIOMÉTRICA..... | 90 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|---|
| ABEPRO | Associação Brasileira de Engenharia de Produção |
| AG | Algoritmo Genético |
| ENEGEP | Encontro Nacional de Engenharia de Produção |
| G&P | Gestão e Produção |
| GEPROS | Gestão da Produção, Operações e Sistemas |
| PCV | Problema do Caixeiro Viajante |
| PL | Programação Linear |
| PO | Pesquisa Operacional |
| PODes | Pesquisa Operacional para o Desenvolvimento |
| PSO | <i>Particle Swarm Optimization</i> |
| SA | <i>Simulated Annealing</i> |
| SCIELO | <i>Scientific Electronic Libray Online</i> |
| SIMPEP | Simpósio de Engenharia de Produção |
| SOBRAPO | Sociedade Brasileira de Pesquisa Operacional |

LISTA DE SÍMBOLOS

| | |
|--------------|--|
| f | função objetivo |
| Min | mínimo |
| $!$ | operador fatorial |
| f_{apt} | função de aptidão |
| E | em <i>Simulated Annealing</i> , a energia do átomo. |
| T | em <i>Simulated Annealing</i> , a temperatura |
| ΔE | variação da energia do átomo |
| P | em <i>Simulated Annealing</i> , a probabilidade de aceitação da configuração |
| K_b | constante de Boltzmann |
| T_0 | temperatura inicial |
| α | fator de ajuste |
| T_f | temperatura final |
| v_i | velocidade atual da partícula i ; |
| c_1, c_2 | parâmetros de confiança; |
| $rand$ | função aleatória; |
| $pbest_i$ | melhor posição que uma i partícula já obteve |
| $gbest$ | melhor posição encontrada pelas partículas no enxame |
| x_i | posição atual da partícula i |
| it | iteração atual. |
| $c_{1\ ini}$ | valor inicial para o parâmetro de confiança cognitivo |
| $c_{1\ fin}$ | valor final para o parâmetro de confiança cognitivo |
| $c_{2\ ini}$ | valor inicial para o parâmetro de confiança social |
| $c_{2\ fin}$ | valor final para o parâmetro de confiança social |
| R | em <i>Particle Swarm</i> é o número de iterações |
| v_{max} | velocidade máxima |
| w | peso inercial |

1 INTRODUÇÃO

Frente a um ambiente de negócios amplo e turbulento, a pesquisa passa a representar um recurso de grande poder para coletar, analisar e extrair informações valiosas de dados, tanto externos como internos às organizações (FREITAS e MOSCAROLA, 2002). Com efeito, cabe a aplicação de métodos de pesquisa quantitativos e qualitativos para apoio à tomada de decisão.

Para Lachtermacher (2004), a tomada de decisão pode ser entendida como o processo de identificar um problema ou uma oportunidade e selecionar uma linha de ação para resolvê-lo. Para tanto, muitas ferramentas em Pesquisa Operacional têm sido utilizadas, dentre elas, os modelos matemáticos, ideais em casos onde as grandezas são representadas por variáveis de decisão, e as relações entre as mesmas por expressões matemáticas.

Estes problemas, em que a qualidade da resposta pode ser medida por um número, aparecem em todas as áreas do conhecimento, sobretudo na Engenharia. Segundo Moreira (*apud* Marcondes Filho, 2008) a Programação Linear é um dos modelos matemáticos mais populares, estruturados para resolver problemas que apresentam variáveis que possam ser medidas e cujos relacionamentos possam ser expressos por meio de equações lineares. A este grupo de situações, é possível aplicar Algoritmos Exatos, que retornam uma solução ótima, mas nem sempre em tempo aceitável.

Todavia, muitos problemas em Engenharia de Produção são pertencentes à classe de problemas NP-Completo. Conforme Conselheiro (1999), a característica destes problemas é o aumento exponencial do tempo de solução de acordo com a dimensão do problema. Mesmo para problemas de dimensão reduzida, torna-se necessário o uso de técnicas que possam abreviar a busca por uma solução.

Diversos autores têm utilizado com sucesso Meta-heurísticas em problemas deste gênero. Estes métodos, ainda que não garantam encontrar uma solução ótima, fornecem uma solução viável, isto é, que se aproxime da solução ótima, e com um tempo computacional aceitável. Colin (2007) descreve Meta-heurísticas como sendo procedimentos generalistas que devem

ser adaptados a problemas com instâncias particulares, quando estes não são resolvidos apropriadamente com metodologias otimizantes exatas.

O presente trabalho busca explorar Meta-heurísticas para a resolução de problemas em Engenharia de Produção. Serão apresentados três métodos: Algoritmos Genéticos (AG), *Simulated Annealing* (SA) e *Particle Swarm* (PSO).

1.1 Justificativa

Morabito *et al.* (2008) aponta que modelos de Programação Linear podem utilizar em suas resoluções algoritmos que encontram uma solução ótima. No entanto, dependendo da complexidade do problema, esses métodos podem não ser capazes de fornecer uma solução ótima em tempo computacional aceitável. Neste sentido, existem diversos métodos não-exatos que podem ser utilizados para resolver modelos de programação discreta, como heurística e Meta-heurística.

Deste modo, cumpre elaborar um estudo exploratório em Meta-heurística, afim de identificar aqueles que têm apresentado os melhores resultados na resolução de problemas em Engenharia de Produção. Também, torna-se válido a elaboração de um estudo comparativo entre os resultados apresentados pelos métodos tradicionais e as soluções apresentadas pelas Meta-heurísticas.

1.2 Definição e delimitação do problema

Conforme Vieira (2006), problemas de otimização combinatória são definidos como um conjunto discreto de soluções, os quais possuem um número finito de elementos. Em Souza *et al.* (2009), é apresentada uma gama de soluções do gênero, tais como: alocação de recursos, corte de estoque, fluxo máximo em redes, programação da produção, programação de horários em escolas, dimensionamento de lotes, mistura de minérios, dentre outros.

Entretanto, alguns problemas de otimização combinatória não conseguem chegar a uma solução ótima em um tempo computacional baixo e de forma eficaz utilizando métodos determinísticos. Assim, uma determinada solução pode ser considerada aceitável se estiver

próxima da solução ótima. O bom desempenho de Meta-heurísticas nestas circunstâncias popularizaram os métodos para esse tipo de situação.

Em Marcondes Filho (2008), é realizada uma distinção entre Algoritmos Heurísticos e Meta-heurística. O primeiro, diz respeito a procedimentos para resolver problemas por meio de um enfoque intuitivo, em geral, racional, no qual a estrutura do problema possa ser interpretada e explorada inteligentemente para obter uma solução razoável em tempo aceitável. Podem ser construtivas ou de melhoramento, não tem prova de convergência e não garantem encontrar a solução ótima.

Por sua vez, Meta-Heurísticas são procedimentos heurísticos, mas exploram o espaço de soluções além da otimidade local, buscando boas soluções em novas regiões promissoras. Dentre as Meta-heurísticas, algumas que se destacam são os Algoritmos Genéticos, o *Simulated Annealing* e o *Particle Swarm*.

O interesse acadêmico por estas técnicas de otimização vem da aplicabilidade, comprovada pelos registros de trabalhos publicados nas mais diversas bases de dados, com resultados promissores.

Em um primeiro momento, será realizado o estudo exploratório do tema, com a apresentação dos algoritmos e sua abrangência de aplicação. Também, serão tabulados os resultados apresentados, para verificar quais algoritmos foram mais efetivos e em que tipo de situação. Por fim, uma implementação simples será realizada, demonstrando o funcionamento de uma Meta-heurística.

1.3 Objetivos

1.3.1 Objetivo geral

Elaborar um estudo exploratório de Métodos Heurísticos aplicados à Engenharia de Produção, apresentando as técnicas mais utilizadas na otimização de situações reais, e demonstrando seu funcionamento.

1.3.2 Objetivos específicos

Como objetivo específico, tem-se:

- Revisar a literatura sobre os métodos heurísticos e sua aplicabilidade;
- Demonstrar a viabilidade de utilização das Meta-heurísticas Algoritmos Genéticos, *Simulated Annealing* e *Particle Swarm* para a resolução de problemas de complexidade exponencial;
- Identificar as categorias de problemas da Engenharia de Produção que podem ser resolvidos através destes métodos;
- Ilustrar a aplicação de um dos métodos na resolução de um problema.

1.4 Metodologia

Foi realizada uma pesquisa básica acerca do tema proposto, com a finalidade de revisar a literatura a respeito de Meta-heurísticas e seu campo de aplicação. A busca de trabalhos publicados nas seguintes bases de dados visa demonstrar a aplicabilidade das técnicas selecionadas (Algoritmos Genéticos, *Simulated Annealing* e *Particle Swarm*) como metodologia de resolução de problemas: Encontro Nacional de Engenharia de Produção (ENEGEP), Simpósio de Engenharia de Produção (SIMPEP), Revista Pesquisa Operacional, Revista Gestão e Produção, Revista Pesquisa Operacional para o Desenvolvimento, Revista GEPROS, Revista Produção, *Scientific Electronic Libray Online* (SciELO), Biblioteca de TCC's do Departamento de Engenharia de Produção da Universidade Estadual de Maringá (DEP/UEM). Para tanto, foram considerados os trabalhos publicados nos últimos dez anos.

Os trabalhos publicados nestas bases de dados, que utilizaram as Meta-heurísticas citadas para resolução de problemas foram tabulados, a fim de identificar quais métodos têm sido aplicados e em quais áreas da Engenharia de Produção.

Por fim, elegeu-se um problema, por meio de pesquisa literária, que ilustra a aplicação de um dos métodos, de modo a deixar claro os passos do algoritmo e a forma de resolução. O fluxograma da metodologia é apresentado na Figura 1.

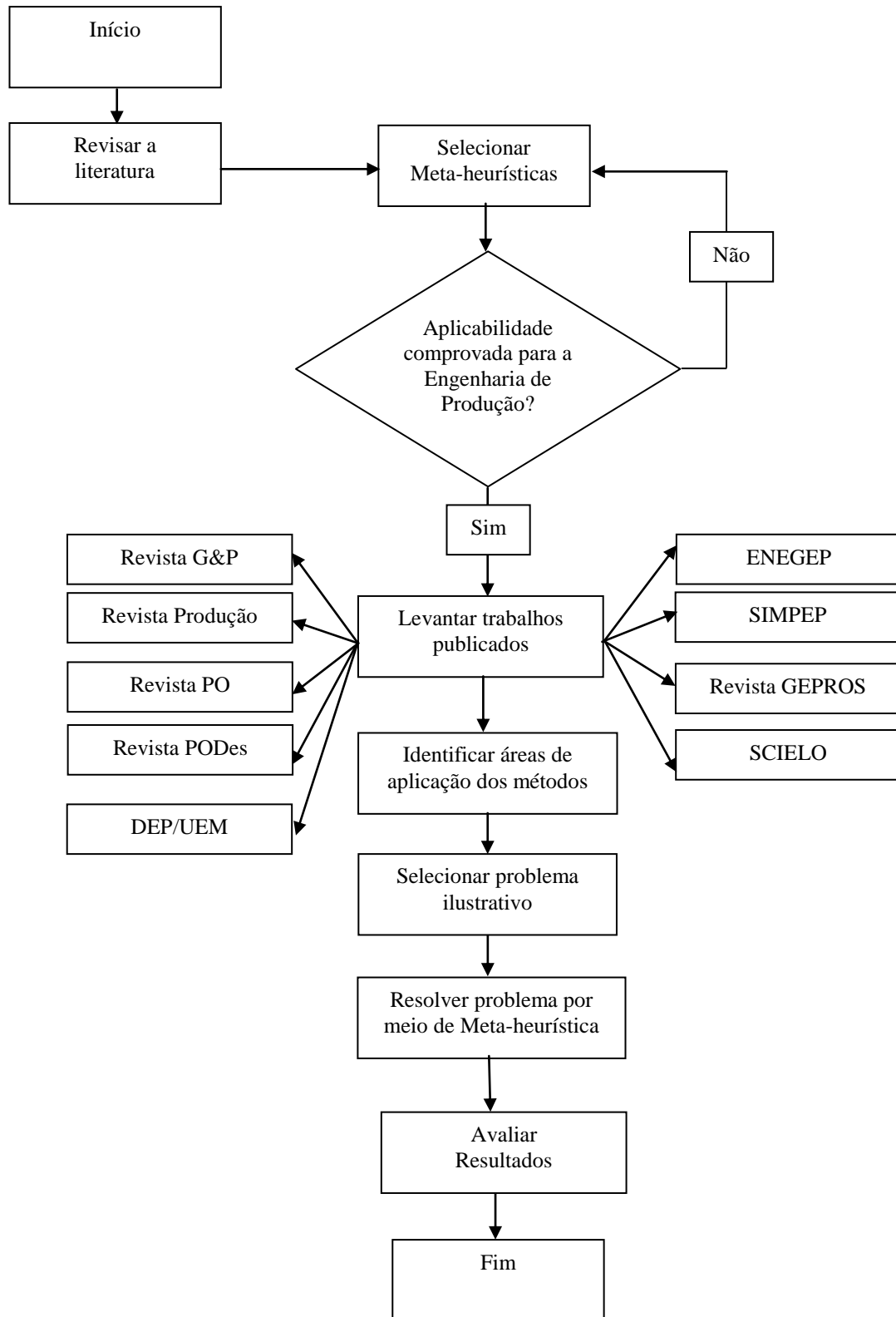


Figura 1 – Fluxograma da Metodologia

2 REVISÃO LITERÁRIA

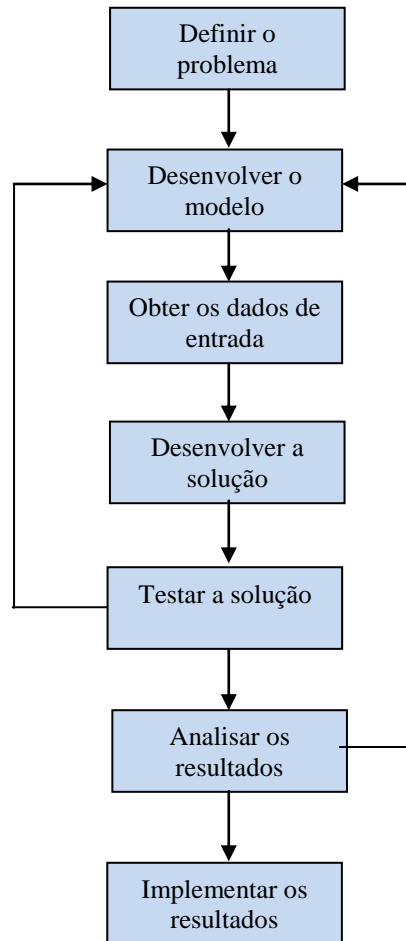
2.1 A Abordagem quantitativa e a Pesquisa Operacional

Conforme mencionado em Laugeni e Martins (2005), a partir dos trabalhos de Frederick Winslow Taylor, surgidos nos Estados Unidos, no final do século XIX, houve a sistematização do conceito de produtividade, que trata-se da busca incessante por melhores métodos de trabalho e processos produtivos, com o objetivo de se obter melhoria na produtividade com o menor custo possível. Esta procura passou a ser o ponto central de todas as empresas até hoje, diferenciando-se somente quanto às técnicas utilizadas. Com efeito, a seleção destas técnicas torna-se um importante diferencial competitivo, pois as organizações que obtiverem sucesso na escolha e execução dos métodos alcançarão os melhores resultados em termos de produtividade.

Para Hanna *et al.* (2010), na resolução de problemas, tanto fatores qualitativos quanto quantitativos devem ser considerados. A análise quantitativa é uma abordagem científica para a tomada de decisão onde, através do processamento e manipulação dos dados, gera-se a informação que apoiará a tomada de decisão. O resultado deve ser combinado com outras informações qualitativas do problema para se obter o posicionamento final com relação àquele cenário. A Figura 2 apresenta a sequência de ações para a resolução de um problema através da análise quantitativa, proposta pelos mesmos autores.

O processo de resolução tem início com a definição do problema. Em geral, um pesquisador, ou grupo de pesquisadores, utiliza a análise quantitativa para trazer soluções que signifiquem diminuição de custos ou aumento de lucros e que apontem soluções para problemas reais. Em seguida, tem início a confecção de um modelo para resolução. O modelo é uma representação do mundo real. Segundo Marcondes Filho (2008), modelos são representações simplificadas que preservam para determinadas situações uma equivalência da realidade, tornando os problemas mais fáceis e claros. Desenvolvido o modelo, são estabelecidos os dados de entrada, que devem ser precisos, pois dados equivocados produzem resultados enganosos (HANNA *et al.*, 2010). Através da manipulação do modelo, se desenvolve uma solução, procurando o melhor resultado possível. Tal solução é, então, testada. Se os dados estão

corretos, mas o resultado é inconciso com o problema, o modelo deve ser checado, a fim de investigar se possui lógica e representatividade com relação à situação real.



**Figura 2 – A abordagem da análise quantitativa
(HANNA, *et al.* 2010)**

Dando continuidade ao processo de resolução, deve ser procedida a análise dos resultados. Sendo o modelo apenas uma representação da realidade, a sensibilidade da solução a mudanças no modelo e nos dados de entrada é importante nesta etapa. Esse tipo de estudo é denominado análise de sensibilidade ou análise de pós-otimização (HANNA *et al.* ,2010). Finalmente, é procedida a implementação dos resultados.

Desde que os princípios da administração científica da produção foram criados, as técnicas de Pesquisa Operacional (PO) têm sido utilizadas, devido à sua forte estruturação lógica e amparo no ferramental matemático, para a resolução de problemas em diversas áreas. Conforme Arenales *et al.* (2007), para alguns autores, a PO é a abordagem científica para a

tomada de decisão, que procura determinar como melhor projetar e operar um sistema, usualmente sob condições que requerem a alocação de recursos escassos. Para Goldberg e Luna (2000), a PO congrega as mais consagradas técnicas para solução de modelos quantitativos, que podem ser expressos matematicamente, por meio de modelos estruturados de forma lógica e amparados no ferramental matemático de representação, objetivando claramente a determinação das melhores condições de funcionamento para os sistemas representados.

É possível encontrar aplicações para a PO nas mais diversas áreas do conhecimento, tais como Administração, conforme Hanna *et al.* (2010) e Lachtermacher (2004), Economia e Ciências Contábeis, apresentando em Lachtermacher (2004), Engenharia, demonstrado em Arenales *et al.* (2007), Estratégia, Finanças, Logística, Produção, Marketing e Vendas, encontrado no trabalho de Colin (2007), Sistemas Agroindustriais, descrito por Caixeta-Filho (2004), dentre várias outras aplicações descritas na literatura.

A resolução de problemas em Pesquisa Operacional pode ser dividida, segundo a Associação Brasileira de Engenharia de Produção (ABEPRO) (2012), nas seguintes subáreas:

1. Modelagem, simulação e otimização;
2. Programação matemática;
3. Processos decisórios;
4. Processos estocásticos;
5. Teoria dos jogos;
6. Análise de Demanda;
7. Inteligência computacional.

A abordagem utilizada na resolução de um problema depende da estrutura do mesmo e da modelagem proposta. Muitas vezes, uma solução exata pode ser encontrada, sendo a melhor existente e que maximiza o resultado de forma que nenhuma outra solução possível faria melhor. Isso pode ser realizado por meio de ferramentas de cálculo matemático. Todavia, existem várias situações práticas em que a complexidade do problema é tamanha que torna difícil encontrar tal solução. Desta forma, alternativas de resolução, como empregos de buscas

aleatórias, são ferramentas que trazem soluções nem sempre ótimas, mas aceitáveis aos problemas.

2.2 Modelagem Matemática

Conforme Arenales *et al.* (2007), uma situação real passível de ser descrita por relações matemáticas, dá origem a um modelo matemático do problema. A validação do modelo depende de a solução ser coerente com o contexto original.

Segundo Leal (2007), a otimização é o processo de encontrar a melhor solução dentre um conjunto de soluções para um problema, e ou ainda, é a técnica utilizada para minimizar (ou maximizar) uma função sujeita a restrições em suas variáveis. O problema geral consiste na composição de uma função objetivo que relacione diferentes variáveis do problema considerado sujeita, ou não, as restrições impostas. Trigueiros (2011) descreve um modelo matemático conforme segue abaixo.

$$\text{Minimizar ou maximizar} \quad f(x,y) \quad (1)$$

$$\text{Sujeito a} \quad h(x,y) = 0 \quad (2)$$

$$g(x,y) \leq 0 \quad (3)$$

Os valores x e y representam p variáveis reais e contínuas e q variáveis inteiras, respectivamente. Os vetores h e g representam n restrições de igualdade e m restrições de desigualdade, respectivamente (TRIGUEIROS, 2011). Quando todas as expressões envolvidas são lineares, é gerada uma função objetivo dita convexa, de tal forma que o ótimo local é também global. A Figura 3 ilustra esta situação.

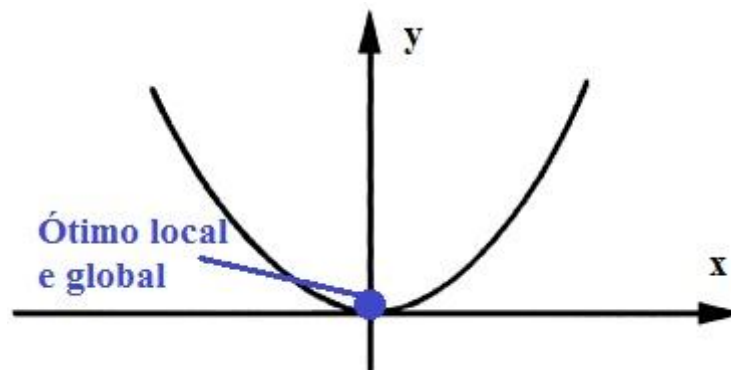


Figura 3 – Função Convexa

Em tais situações, frequentemente, métodos de programação linear são utilizados, tais como o Simplex. Todavia, conforme menciona Leal (2007), a utilização de técnicas de programação linear traz alguns problemas de robustez relacionados à continuidade das funções a serem otimizadas ou de suas restrições, existência de funções não-convexas, multimodalidade, existência de ruídos nas funções, necessidade de trabalhar com valores discretos para as variáveis, entre outros. Portanto, faz-se necessário o estudo de métodos alternativos para a solução destes problemas. Casos em que se enquadram tais fatos são descritos a seguir.

2.3 Otimização Global

Leal (2007), expõe que problemas de otimização podem ser divididos em duas categorias quanto à solução: ótimo local e ótimo global. Uma solução é ótimo local se ela é viável e se em sua vizinhança não possui soluções que sejam ao mesmo tempo viáveis e apresentem melhores valores da função objetivo. Uma solução é ótimo global se é viável e não há nenhuma outra solução viável com melhor valor da função objetivo.

Matematicamente, conforme Biscaia Junior (2002), uma solução ótimo local busca minimizar $f(x)$, procurando um x^* para o qual $f(x^*) \leq f(x)$ para todo x próximos de x^* . Encontrar um ótimo global seria o mesmo que encontrar um x^* que minimiza a função $f(x)$ para todos os valores possíveis de x .

Ainda, Cortes (1996), define ótimo local como o valor ótimo, máximo ou mínimo, da função objetivo em uma sub-região da região viável, e ótimo global como o valor ótimo, máximo ou mínimo, da função objetivo em toda a região viável. A Figura 4, proposta por Trigueiros (2011), mostra as soluções viáveis da função objetivo de um problema de minimização, a partir do qual é possível observar um ótimo local e um ótimo global.

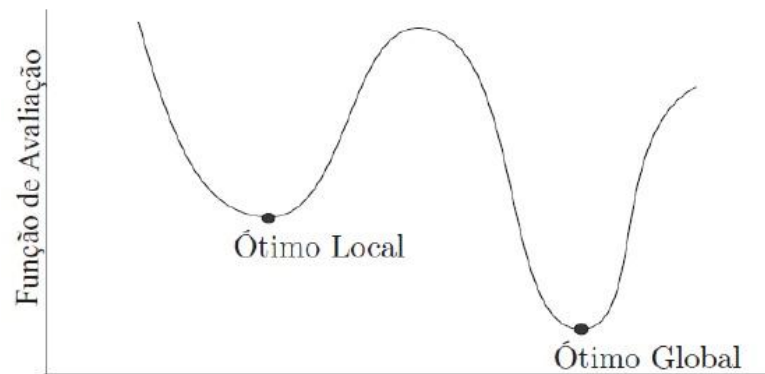


Figura 4: Função objetivo com ótimo local e global
(TRIGUEROS, 2011)

O problema de determinar o ótimo global é mais complexo do que identificar um ótimo local. Mas, sabendo que o ótimo local é uma solução viável e sua vizinhança não apresenta soluções melhores, se for possível delimitar a região de busca, ou usar indícios de pontos de partida para os quais uma metodologia de busca por soluções pode iniciar a procura pela solução, para a grande parte dos problemas, esta solução já será satisfatória.

Ótimos locais podem ser determinados por métodos de busca local simples, tais como o conhecido método por derivadas. Todavia, estes métodos perdem a eficiência quando a função possui uma grande quantidade de ótimos locais na vizinhança de busca, o que pode gerar uma armadilha, fazendo a solução convergir para um resultado que não é o melhor possível. Isso ocorre frequentemente quando há não-convexidade da função ou de suas restrições. Superar estas dificuldades exigem a aplicação de técnicas de otimização global que façam varreduras no espaço de busca para além de ótimos locais que não são ótimos globais.

2.4 Otimização Combinatória

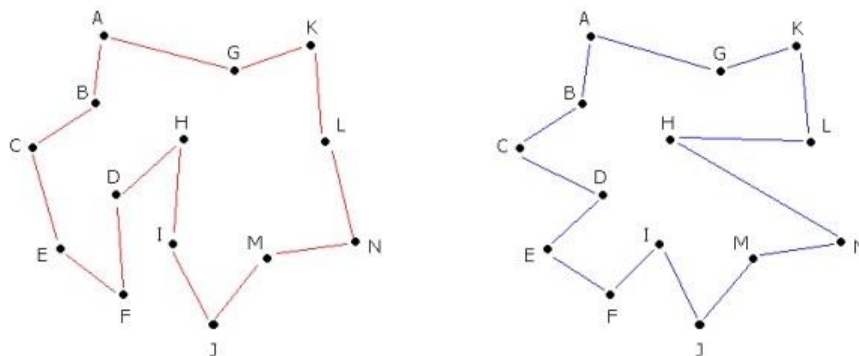
Condenossi (*apud* Marcondes Filho, 2008), descreve a otimização combinatória como um ramo da matemática que analisa problemas de otimização dentro de um limite de tempo. Otimizar significa avaliar as alternativas disponíveis de solução ao modelo e tomar aquela que gera os melhores resultados. A otimização combinatória busca encontrar o resultado que, dentro das restrições existentes, tem o menor custo possível.

Para o mesmo autor, em otimização combinatória, com as regras e restrições existentes, tem-se a possibilidade de enumerar todas as possíveis soluções, criando-se todos os subconjuntos existentes a partir do conjunto inicial e das regras de restrições. Na prática, isso é inviável para a maior parte dos problemas, devido ao elevado número de soluções possíveis.

Para ilustrar o caso, a situação mais conhecida que pode-se tomar como exemplo é o Problema do Caixeiro Viajante, ou PCV. Conforme Goldberg e Luna (2000), este problema é um dos mais tradicionais e conhecidos de programação matemática, possuindo grande aplicação prática, enorme relação com outros modelos e grande dificuldade de solução exata.

Melo (2006) esclarece que o PCV é um problema de roteamento que, resumidamente, consiste na elaboração de uma rota que deve cobrir pontos nos quais se encontra determinada oferta ou demanda. A rota deve se iniciar por determinado ponto que pode ser uma cidade, um posto de trabalho ou depósito, percorrer todos os demais pontos, passando por cada um deles uma única vez, e retornar ao ponto inicial, formando o chamado ciclo hamiltoniano. A ideia é procurar o caminho que tenha o menor custo.

A Figura 5, proposta por Biscaia Junior (2002), traz dois trajetos diferentes que ligam 14 cidades. O melhor trajeto que se poderia percorrer é aquele que traz o menor custo, em termos de combustível e tempo. Outros fatores, tais como as condições das estradas, também devem ser levados em conta.



**Figura 5 – Dois trajetos entre 14 cidades
(BISCAIA JUNIOR, 2002).**

O mesmo autor esclarece a dificuldade do problema: para n cidades, sendo a primeira cidade fixa, que é a origem do trajeto, o número total de escolhas possível é dado pela equação:

$$(n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1 \quad (4)$$

Ou, em notação fatorial:

$$R(n) = (n - 1)! \quad (5)$$

Quando para o problema não exista a preocupação de fixar a primeira cidade, ficando a busca restrita à melhor partindo de qualquer ponto, o número de combinações é dado por:

$$R(n) = n! \quad (6)$$

A estratégia de otimização combinatória consiste em gerar todas as $R(n)$ rotas, calcular o comprimento total de cada uma e ver qual tem o menor custo. É imediato que esta tarefa é feita com a ajuda do computador. Todavia, como a quantidade de soluções cresce fatorialmente, rapidamente o computador se torna incapaz de proceder os cálculos em tempo aceitável. Isso é devido ao caráter fatorial do problema. Mesmo com o aprimoramento dos recursos de *hardware* e *software*, números altos de pontos no grafo geram quantidades de combinações exorbitantes, impossibilitando listar todas as soluções e realizar as comparações necessárias para se encontrar a melhor resposta. É, também, o que ocorre nos demais problemas de Otimização Combinatória. Assim, a alternativa é adotar métodos que busquem o melhor resultado possível em tempo aceitável.

2.5 A Teoria de N-completude

Conforme Arenales *et al.* (2007), a PO tem sido chamada de “ciência e tecnologia de decisão”. Ciência, porque está relacionada a ideias e processos para articular e modelar problemas de decisão, determinando os objetivos do tomador de decisão e as restrições sob as quais se deve operar. Tecnologia, porque está relacionada a ferramentas de software e

hardware para coletar e comunicar dados, e organizar esses dados, usando-os para gerar e otimizar modelos e reportar resultados.

Um algoritmo é um processo sistemático para a resolução de um problema. O desenvolvimento de algoritmos é particularmente importante para problemas a serem solucionados em um computador, pela própria natureza do instrumento utilizado (MARKENZON e SZWARCFITER, 1994). Um algoritmo pode ser definido como uma sequência de passos para atingir um determinado objetivo, podendo ser utilizado como metodologia para encontrar a resposta a um problema proposto. Ao ser implementado em computador, através de linguagem de programação, torna-se uma importante ferramenta para a Pesquisa Operacional.

Todavia, conforme Goldberg e Luna (2000), apenas estabelecer um conjunto de instruções que soluciona certo problema decidível não significa que se tenha encontrado uma estratégia razoável, sob qualquer que seja o ponto de vista. O objetivo de um algoritmo é solucionar o problema de forma rápida e econômica. Assim, é necessário definir quais tipos de problemas existem e por quais tipos de algoritmos devem ser abordados.

Marcondes Filho (2008) divide algoritmos em determinísticos (a cada passo dado pelo algoritmo, o próximo passo apresentará somente uma solução) e não-determinísticos (fazem uma escolha aleatória do próximo passo, entre um número fixo de possibilidades e, conseqüentemente, o fluxo do algoritmo depende da alternativa selecionada, gerando um número finito de soluções). Para Trigueiros (2011), alguns métodos determinísticos simples são suficientemente capazes de resolver uma grande variedade de problemas. Contudo, quando o problema de otimização é complexo, quanto ao número de variáveis ou ao tamanho do espaço de busca, a aplicação de métodos determinísticos simples pode não avaliar efetivamente toda a região em estudo. A mesma autora destaca que, quando a utilização de métodos determinísticos é intratável computacionalmente, recomenda-se a utilização de abordagens não determinísticas de otimização.

Segundo Marcondes Filho (2008) e Cormen *et al.* (2002), os problemas podem ser divididos em quatro classes de complexidade, definidas da seguinte forma:

- P (*Polinomial Time*): problemas que podem ser resolvidos com algoritmos determinísticos polinomiais em função do tamanho da instância, a complexidade do problema cresce polinomialmente em função do tamanho da instância;
- NP (*NonDeterministic Polynomial Time*): problemas que podem ser resolvidos por algoritmos não-determinísticos polinomiais no tamanho da instância. A complexidade do problema cresce exponencialmente em função do tamanho da instância. Todo algoritmo determinístico é um caso particular de algoritmo não-determinístico e, portanto, $P \subseteq NP$;
- NP-Completo (NPC): problemas para os quais, até o momento, nenhum algoritmo de tempo polinomial jamais foi descoberto para resolver.
- NP-difícil (NPD): problemas para os quais a solução é, pelo menos, tão difícil quanto a solução de um problema NP-Completo.

A Figura 6 mostra a relação entre as classes de problemas. Fica claro, através de sua observação, que $P \subseteq NP$. Conforme Cormen *et al.* (2002), a classe NPC tem a surpreendente propriedade de que, se qualquer problema NP-Completo puder ser resolvido em tempo polinomial, então todo problema NP teria uma solução em tempo polinomial, isto é, $P=NP$. Porém, não há indícios de que isso será demonstrado. E, por outro lado, se qualquer problema NP não puder ser resolvido em tempo polinomial, então nenhum problema NPC pode ser resolvido em tempo polinomial. A maioria dos pesquisadores da área acredita nesta possibilidade, o que está de acordo com o contido na Figura 6.

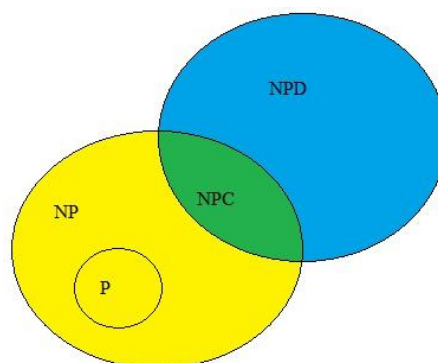


Figura 6 – Relação entre as classes de problemas
(CORMEM *et al.* 2002 *apud* MARCONDES FILHO, 2008)

O interesse na categoria de problemas NP-Completo reside no fato de que muitos problemas práticos pertencem a esta classe, sobretudo, na Engenharia. Por este motivo, torna-se importante o estudo e discussão de métodos que solucionem este tipo de problema. Cormen *et al.* (2002), realiza uma importante observação a respeito de problemas NP-completo, indicando as melhores metodologias

“Se puder estabelecer um problema como NP-Completo, você fornecerá uma boa evidência de sua intratabilidade. Como engenheiro, você faria melhor gastando seu tempo no desenvolvimento de um algoritmo de aproximação ou resolvendo um caso especial tratável, em vez de procurar por um algoritmo rápido que resolva o problema exatamente.” (CORMEN, *et al.* 2002).

Biscaixa Junior (2002) nomeia a categoria dos NP-Completo de “problemas não-determinísticos polinomiais completos”. O PCV, anteriormente apresentado, bem como os problemas de otimização combinatória, se encaixam em categorias NP-Completo ou NP-Difíceis.

2.6 Métodos Heurísticos

Segundo Russell e Norvig (2004) “heurística” é a busca informada, ou seja, aquela que utiliza conhecimento específico do problema, além da definição do próprio problema, podendo encontrar soluções mais eficientes do que uma estratégia sem informação.

Métodos Heurísticos, conforme apresenta Schwaarb (2005), são caracterizados por um caráter aleatório na busca do ótimo, um grande número de avaliações da função objetivo e por realizarem uma busca global em toda a região de interesse. Além disso, estes métodos não dependem de uma boa estimativa inicial dos parâmetros e não utilizam derivadas durante a otimização, o que garante a robustez do método, mesmo em problemas com parâmetros não significativos.

Para Marcondes Filho (2008), problemas de otimização combinatória devem ser tratados por algoritmos que, apesar de não garantir que a solução apresentada será ótima, forneçam uma solução viável, ou seja, que se aproxime da solução ótima com um tempo computacional aceitável.

Abordagens alternativas de otimização têm sido desenvolvidas com a utilização de métodos heurísticos. Sobretudo para problemas do tipo NP-Completo, estes métodos têm surgido com alta eficiência, conforme pode ser comprovado por inúmeros estudos apresentados na literatura. Tais métodos não garantem a determinação de soluções ótimas, mas eventualmente as encontram.

Afirma Biscaia Junior (2002) que, para problemas difíceis como a otimização global, os métodos heurísticos de busca têm sido largamente estudados e utilizados. Mesmo sem encontrar soluções ótimas, eles demonstram a melhor solução conhecida, o que pode ser mais do que o suficiente para o problema em questão.

Uma questão imediata em termos da utilização de métodos heurísticos diz respeito ao emprego destes métodos frente a um problema determinado. Conselheiro (2009) apresenta os seguintes fatores que tornam interessante a utilização de métodos heurísticos:

- i. inexistência de um método exato de resolução ou este requer muito esforço computacional;
- ii. quando há limitação do tempo, obrigando o emprego de métodos de resposta rápida à custa da precisão;
- iii. como passo intermediário de outro algoritmo para geração de uma solução inicial para métodos iterativos;
- iv. quando os dados são pouco confiáveis ou o modelo não descreve bem a realidade, abre-se mão de soluções exatas já que estas são uma aproximação da realidade;
- v. quando não se necessita da solução ótima, se os valores adquiridos pela função objetivo são relativamente pequenos pode não valer a pena o esforço por uma solução ótima. Pode ser suficiente a apresentação de uma solução melhor do que a atual.

Conforme Cordenossi (2008) e Marcondes Filho (2008), os métodos heurísticos podem ser divididos em três grupos: construtivos, de melhoramento e Meta-heurística. Uma heurística construtiva ou *cega* inicia com uma solução vazia e, passo a passo, insere novos componentes à solução, sendo que somente ao final das iterações do algoritmo se obtém uma solução, cujo valor se torna limitante para procura de novas soluções. Após um resultado ser inserido, não é possível mais retirá-lo. Para o exemplo do PCV citado, uma heurística construtiva consiste em tentar encontrar uma boa rota, considerando a cada interação somente o próximo passo, partindo de uma solução vazia e construindo a rota. Uma cidade é inserida de cada vez, até atingir a rota completa. Se uma cidade é inserida, não poderá mais ser retirada da rota.

Heurísticas de melhoramento, a partir de uma solução inicial, trabalham no melhoramento da solução atual, por meio da realização de passos sucessivos. Realizam inclusão e exclusão de novos resultados, pesquisando a vizinhança de busca das soluções em busca de maior qualidade de resposta. A vizinhança é tida como os novos espaços de busca, que podem ser alcançadas através da aplicação de regras ou funções de alterem a solução atual, gerando novos resultados. As heurísticas de melhoramento param, usualmente, quando nenhuma outra troca possível melhora o resultado atual, o que é considerado um ótimo local. Este ótimo local pode, eventualmente, ser o ótimo global, ou seja, a melhor solução possível para o problema, mas não há garantias em relação a este fato, como em todos os procedimentos heurísticos. Assim, pode acontecer do algoritmo parar e exibir como resposta um ótimo local, ignorando as possibilidades que poderiam levar o algoritmo a mais tarde encontrar um ótimo global. A Figura 7 apresenta uma adaptação para o problema de otimização global, incluindo uma solução inicial. Neste caso, o ponto ótimo local é uma armadilha para um algoritmo heurístico de melhoramento.

Para as situações que os métodos heurísticos tradicionais não conseguem englobar sem eu campo domínio, é possível encontrar esquemas alternativos de solução através de Meta-heurísticas.

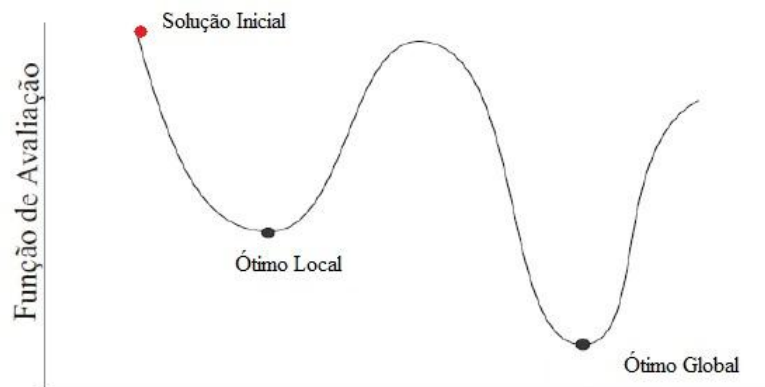


Figura 7: Problema de otimidade global com solução inicial para um algoritmo heurístico de melhoramento

(TRIGUEROS, 2011, e MARCONDES FILHO, 2008, adaptado)

2.7 Meta-heurísticas

As Meta-heurísticas são mecanismos de otimização eficientes e com grande aplicabilidade em diversas áreas na atualidade, devido ao seu caráter adaptativo e facilidade de implementação. Diferente dos métodos heurísticos tradicionais, exploram o espaço de busca evitando armadilhas de otimidade local e aumentam a chance de obter uma solução ótima global. Estas técnicas não apresentam a demonstração de que a solução encontrada é o ótimo global, mas sim uma solução de qualidade e aceitável em tempo computacional.

As duas últimas décadas do século XX trouxeram extraordinários avanços para a computação. Este fato favoreceu os trabalhos criados pelo ramo de Inteligência Artificial, possibilitando o desenvolvimento de programas que tratam o comportamento estocástico dos problemas, ou seja, que tratam o caráter não-determinístico dos casos com o auxílio de regras de probabilidade. Neste contexto, a implementação das Meta-heurísticas, com o auxílio de linguagens de programação, trouxeram importantes avanços para a criação de respostas à problemas de otimização global, problemas de otimização combinatória, problemas com muitos mínimos locais, dentre outros.

O método se inicia com a busca da melhor solução partindo de um espaço com soluções previamente apresentadas. Estas, podem ser definidas aleatoriamente ou com a ajuda de alguma heurística. A solução inicial não precisa ser viável (ou seja, não necessariamente

indicaria uma resposta aceitável ao problema) e muitas vezes, nem mesmo é uma boa solução. Em cada iteração, ocorre a alteração aleatória das variáveis de decisão do problema. O desfecho do método é a localização de uma solução aceitável ao problema, perante as restrições impostas e em tempo hábil.

Trigueiros (2011), divide as Meta-heurísticas em duas categorias:

- Meta-heurísticas de busca local: onde a pesquisa é feita por meio de operações aplicadas sobre a solução atual, em busca de outra solução de melhor qualidade da vizinhança. Possibilita percorrer todo o espaço de busca, considerando a vizinhança da solução atual como o conjunto de soluções viáveis do problema de otimização. A busca fornece uma solução no final de cada iteração, a partir da qual a busca prosseguirá na próxima iteração. Exemplos: *Tabu Search* (busca tabu), *Simulated Annealing* (Recozimento Simulado), *Variable Neighbourhood Search* (busca em Vizinhança Variável) e *Iterative Local Search* (busca Local Iterativa).
- Meta-heurísticas de busca global (populacional ou evolutiva): utilizam de uma população que evolui por meio de interações entre seus valores. O algoritmo armazena e fornece ao final de cada iteração um conjunto de boas soluções que são combinadas para gerar uma solução ainda melhor. Exemplos: *Genetic Algorithms* (Algoritmos Genéticos), *Memetic Algorithms* (Algoritmos Meméticos), *Ant Colony Optimization* (Otimização por Colônia de Formigas) e *Particle Swarm* (Exame de Partículas).

As técnicas de Algoritmos Genéticos, *Particle Swarm* e *Simulated Annealing* têm sido utilizadas em Engenharia de Produção com grande efetividade e resultados notórios. A seguir, estão descritos o comportamento destas técnicas. Para efeitos de nomenclatura, apenas a primeira Meta-heurística aqui apresentada será utilizada com a tradução em português, por assim ser apresentada pela maioria dos autores brasileiros. Em geral, as duas outras são utilizadas com o nome original em inglês.

2.7.1 Algoritmos Genéticos (AG)

Conforme Braga *et al.* (2005), Algoritmos Genéticos são algoritmos de busca global, desenvolvido pela chamada Computação Evolutiva, que faz uso de analogias à evolução natural proposta por Charles Darwin no século XIX e adaptados a esses modelos após a segunda metade do século XX. São programas computacionais que, tal qual a teoria de Darwin, propõe que os indivíduos com as melhores características têm maior chance de sobrevivência e de criarem descendentes mais aptos a sobreviver, que na prática são os melhores resultados para a resposta da situação proposta. Por sua vez, indivíduos com características não tão aptas, ou piores, tendem a desaparecer do grupo, ou seja, do conjunto de solução.

Os AGs são algoritmos não-determinísticos. Como outras Meta-heurísticas, garantem uma solução viável, nem sempre ótimo global, mas que possui condições de escapar de ótimos locais. Para Braga *et al.* (2005), os AGs, aparentemente simples, são capazes de resolver problemas complexos de uma maneira muito elegante e não limitados por suposições sobre o espaço de busca, relativas a continuidade, existência de derivadas, etc. Buscas em problemas reais são repletos de descontinuidades, ruídos e outros problemas. Por isso, ocorre grande número de aplicações práticas para AGs.

Segundo Michalewicz (1999, *apud* Leal, 2007), um algoritmo genético apresenta cinco aspectos fundamentais:

1. representação genética das soluções candidatas ou potenciais, ou processo de codificação;
2. geração da população inicial de soluções candidatas ou potenciais;
3. função de aptidão que avalie as soluções e classifique-as de acordo com sua adaptação ao meio;
4. operadores genéticos para a reprodução de novos indivíduos;
5. identificação dos valores para os vários parâmetros do algoritmo genético, tais como: tamanho da população, critérios de parada do algoritmo, método de seleção, taxa de mutação, entre outros.

Um AG pode ser representado, em pseudocódigo, conforme a Tabela 1.

Tabela 1 – Pseudocódigo para um AG

| Algoritmo Genético |
|---|
| <p>Início</p> <p>1) Gerar uma população inicial</p> <p>2) Avaliar a função objetivo dos indivíduos da população</p> <p>3) Repetir</p> <p style="padding-left: 20px;">Início</p> <p style="padding-left: 20px;">Cruzar os pais de modo que se reproduzam</p> <p style="padding-left: 20px;">Avaliar a função objetivo dos filhos gerados</p> <p style="padding-left: 20px;">Substituir os filhos julgados inadequados</p> <p style="padding-left: 20px;">Fim</p> <p>Até que o critério de parada seja atendido</p> <p>Fim</p> |

Fonte: Goldberg e Luna (2000)

As seções seguintes demonstram os itens que integram um AG: representação genética (ou codificação), geração da população inicial, função de aptidão, seleção de indivíduos, parâmetros e critério de parada.

2.7.2 Representação Genética ou Codificação

Conforme Braga *et al.* (2005), a representação genética é o primeiro passo para a utilização e um AG como mecanismo de resolução de um problema de otimização. O AG processa populações de indivíduos, intituladas Cromossomos. Um cromossomo é a forma de representar um indivíduo na população, que nada mais é do que uma estrutura de dados, em geral vetores de componentes. A todos os cromossomos reunidos dá-se o nome de espaço de busca.

Um Cromossomo é constituído por Genes, que são as variáveis do problema. A ausência ou presença de um Gene é a existência ou não de uma característica para aquela solução. Estes

Para ilustrar os passos de um AG, aqui, será utilizada codificação binária, devido a maior facilidade de visualização do processo.

2.7.3 Geração da População Inicial

A população inicial é o conjunto de cromossomos ou soluções, das quais o método partirá para buscar resultados mais promissores. Tradicionalmente, o mais comum é gerar a população de forma aleatória ou utilizar alguma heurística para determiná-la, conforme já foi exposto anteriormente. Ainda assim, cumpre apresentar as técnicas de geração de população inicial existentes, conforme apresentam Guimarães e Ramalho (2012). Para estes autores, as técnicas de inicialização podem ser divididas em quatro grupos:

1. inicialização aleatória da população: os N indivíduos são distribuídos aleatoriamente dentro do espaço de solução. Possui a desvantagem de possibilitar a ocorrência de regiões com concentração de pontos e outras sem indivíduos. Porém, permite uma maior variabilidade dos resultados obtidos;
2. inicialização através de uma heurística (inicialização determinística): utiliza uma heurística para a determinação da população, com a finalidade de distribuir uniformemente os indivíduos dentro do espaço de solução. Possui a desvantagem de cobrir bem o espaço de solução, mas a execução sucessiva não traz informação adicional, pois o estado de inicialização é sempre o mesmo;
3. inicialização aleatória com Nicho: uma região de cobertura para cada indivíduo é estabelecida, a qual se dá o nome de Nicho. Ao gerar a população inicial, se dois ou mais indivíduos aparecerem muito próximos, um deles é eliminado e o outro é gerado aleatoriamente. O processo é repetido até que todos os indivíduos estejam sozinhos em seus Nichos. Garante que não há indivíduos próximos uns dos outros, mas não garante que todo o espaço de solução esteja coberto;
4. inicialização implementada: faz-se uma mescla das técnicas anteriores. Inicialmente, gera-se uma população uniformemente distribuída dentro do espaço solução. Em

seguida, aplica-se uma perturbação aleatória em cada indivíduo. A perturbação aleatória é determinada pelo nicho de cada indivíduo. Assim, fica garantido uma cobertura ampla do espaço solução e a variabilidade característica da distribuição aleatória.

Outro fator relevante quanto a população inicial diz respeito ao seu tamanho. Leal (2007) argumenta que populações muito pequenas trazem problemas relacionados à perda de diversidade, encurtando as chances de obtenção do ótimo global. Populações maiores possuem maior chance de convergência para o valor desejado, pois a probabilidade da solução ótima ser constatada entre os elementos aumenta. Porém, o tempo de processamento também aumenta.

2.7.4 Função de Aptidão

Esta função é utilizada, conforme Braga *et al.* (2005) para o processo privilegie os indivíduos mais aptos. Para isso, a cada cromossomo é designado um valor de aptidão, dada pela função de aptidão $f_{apt}(x)$. Ela tem como entrada os valores dos Genes dos Cromossomos e como saída uma nota para a solução proposta. Os indivíduos com maiores notas terão maior chance de reprodução, mantendo suas características.

Segundo Goldberg e Luna (2005), a função de aptidão classifica os indivíduos por sua performance e por regras que permitirão que esses melhores indivíduos se perpetuem e se reproduzam. A Tabela 2 mostra uma lista de indivíduos avaliados segundo uma $f_{apt}(x)$, adaptado de Braga *et al.* (2005).

Conforme pode-se observar na Tabela 2, o indivíduo 1 possui maior valor de aptidão e, considerando como objetivo a maximização da função. Portanto, possui maiores chances de integrar o subconjunto que irá dar origem à próxima geração. O procedimento de escolha deste novo subconjunto que serão os pais é realizado através de métodos de seleção, conforme será apresentado a seguir.

Tabela 2 – Valor de aptidão para seis indivíduos de uma população

| Indivíduo x | Fapt(x) |
|---------------|-------------|
| 1 | 104 |
| 2 | 52 |
| 3 | 40 |
| 4 | 18 |
| 5 | 10 |
| 6 | 6 |

Fonte: Braga *et al.*, 2005, adaptado

Por fim, sobre a função de aptidão, Leal (2007) expõe os seguintes cuidados que se deve ter ao elaborar a função de aptidão. A função de aptidão é o que demanda maior custo computacional, pois constitui a comparação de um grande número de indivíduos, ainda que isso represente um número muito maior de comparações do que métodos de otimização combinatória. Por isso, é importante tomar os seguintes cuidados para evitar gastos computacionais desnecessários:

- não avaliar o mesmo indivíduo mais de uma vez;
- verificar se os pais são distintos aos filhos;
- manter a população com todos os cromossomos distintos entre si;
- criar uma memória para os algoritmos genéticos, a fim de verificar se um determinado indivíduo já foi criado.

2.7.5 Seleção de Indivíduos

É o processo de selecionar os indivíduos aos quais serão aplicados cruzamentos e mutações para formar a próxima geração. Como já exposto, os indivíduos mais qualificados possuem maior chance de integrar este grupo. Mas, o que se quer é explorar o espaço de busca para além da otimidade local. Assim, não se pode desprezar os indivíduos com menor aptidão, pois os mais aptos em uma geração não estão, necessariamente, mais perto da solução ótimo global. Por este motivo, os métodos de seleção procuram manter uma certa diversidade na

população, dando chance a todas as soluções promissoras, no entanto favorecendo os melhores resultados já encontrados.

2.7.6 - Seleção pelo Método da Roleta

Para iniciar o método, a Tabela 2 será retomada, incluindo uma nova coluna, onde será inserida a aptidão relativa, conforme a Tabela 3. A aptidão relativa pode ser calculada da seguinte forma:

$$\text{Aptidão Relativa} = \frac{f_{apt}(x)}{\sum_{i=1}^n f_{apt}(x)} \quad (9)$$

A aptidão relativa demonstra a probabilidade que o indivíduo tem de entrar no conjunto de solução. A soma de probabilidades será igual a 1. Isso é realizado para a construção do método da roleta, que consiste em colocar os indivíduos em uma roleta com áreas proporcionais à sua aptidão. Recebe maior fatia na roleta quem tiver melhor valor para $f_{apt}(x)$. A ilustração do método pode ser visualizada na Figura 8.

Tabela 3 – Valor de aptidão para seis indivíduos de uma população e aptidão relativa

| Indivíduo x | $f_{apt}(x)$ | Aptidão Relativa |
|-------------------------------------|--------------------------------|-----------------------------|
| 1 | 104 | 0,453 |
| 2 | 52 | 0,226 |
| 3 | 40 | 0,174 |
| 4 | 18 | 0,078 |
| 5 | 10 | 0,043 |
| 6 | 6 | 0,026 |

Fonte: Braga *et al.*, 2005, adaptado

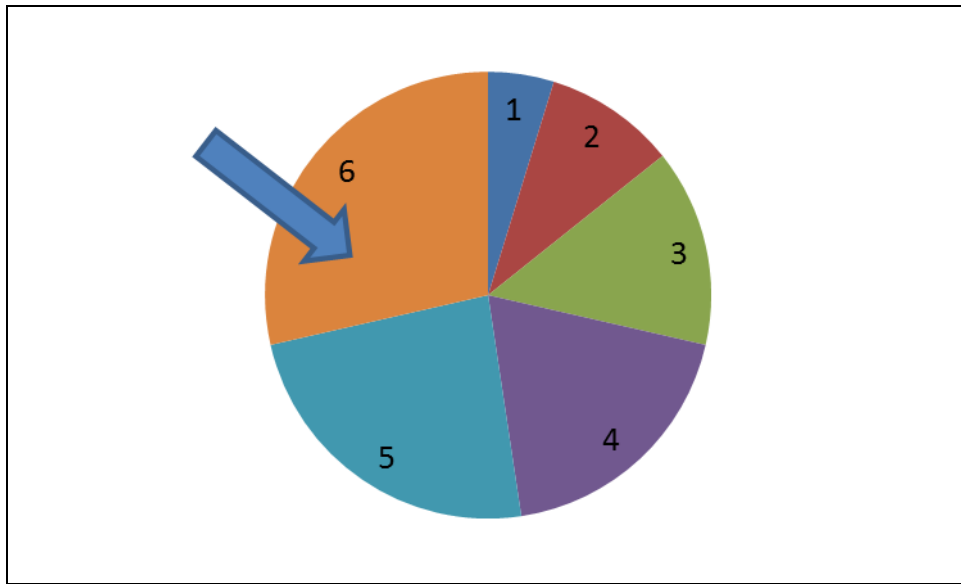


Figura 8 – seleção pelo método da roleta
(BRAGA *et al.*, 2005, adaptado)

A flecha representa o ponto onde a roleta para após ser girada e indica o indivíduo escolhido. Aos indivíduos selecionados é dado continuidade ao algoritmo.

2.7.7 – Seleção pelo método do torneio

Um conjunto de n indivíduos da população inicial são aleatoriamente escolhidos, com a mesma probabilidade. Dentre estes indivíduos, escolhe-se o mais apto para integrar o subconjunto do algoritmo. Em geral, $n = 3$ indivíduos é um número bastante usado. O processo é repetido até que o subconjunto esteja completo. A Tabela 4 ilustra o método por torneio.

Tabela 4 – Seleção por torneio

| <i>Candidatos escolhidos aleatoriamente</i> | Eleito |
|---|---------------|
| 1, 2, 4 | 1 |
| 3, 4, 6 | 3 |
| 1, 3, 6 | 1 |
| 2, 3, 5 | 2 |
| 2, 5, 6 | 2 |
| 1, 3, 6 | 1 |

Fonte: Braga *et al.*, 2005, adaptado

2.7.8 – Seleção pelo método da amostragem universal estocástica

Consiste em uma variação do método da roleta, mas incluindo mais de uma flecha de ponto de seleção, espaçadas com o mesmo ângulo entre elas. A roleta não é girada n vezes para escolher o n indivíduos que farão do subconjunto, mas uma única vez. Ou seja, há a necessidade de inclusão de n flechas. O método pode ser visualizado pela Figura 9.

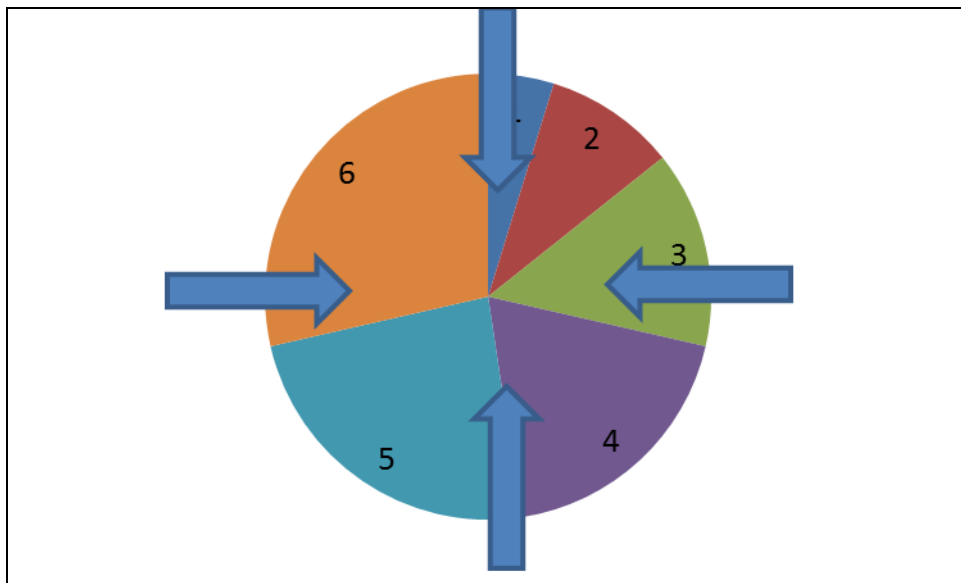


Figura 9 – seleção pelo método da amostra universal estocástica
(BRAGA *et al.*, 2005, adaptado)

2.7.9 – Outros métodos de seleção

Leal (2007) cita outros métodos de seleção, que são: *steady-state* (a população original é mantida, com exceção de alguns poucos indivíduos menos adaptados), seleção aleatória salvacionista (seleciona-se o melhor indivíduo e outros aleatoriamente), seleção elitista (seleciona-se uma porcentagem dos melhores indivíduos e os outros aleatoriamente ou por meio do método da roleta).

Os métodos aqui apresentados para Algoritmos Genéticos podem ser utilizados para a geração de populações de estudo para outras Meta-heurísticas.

2.7.10 Operadores Genéticos

Leal (2007) afirma que um algoritmo de otimização global deve ser capaz de explorar pontos inteiramente novos dentro do espaço de busca, bem como intensificar a busca nas regiões consideradas promissoras. Os operadores genéticos é que garantem este processo. Segundo Goldberg e Luna (2005), os genes devem, ser transformados por operadores genéticos em novos indivíduos. Existem operadores para realizar associações entre indivíduos ou em um mesmo indivíduo.

Segundo Braga *et al.* (2005), os operadores são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores. Para prevenir que os melhores indivíduos não desapareçam da população pela manipulação dos operadores genéticos, podem ser colocados diretamente na próxima geração. A isto, se dá o nome de elitismo.

Os operadores transformam a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. Os dois operadores mais apresentados pela literatura são cruzamento ou *crossover*, e mutação.

2.7.11 Cruzamento

Por Colin (2007), o cruzamento mistura genes de diversos cromossomos com o intuito de criar novos cromossomos. Para Goldberg e Luna (2005), o objetivo do cruzamento é formar novos indivíduos a partir da troca de genes entre os indivíduos pais. Isso permite que as novas gerações herdem as características dos pais, presentes na geração que as originou.

A operação de cruzamento é feita por codificação binária. Se é realiza por meio de codificação real, recebe o nome de recombinação. Braga *et al.* (2005) apresentam os seguintes tipos de cruzamento:

- I. Cruzamento de um ponto: é feita a escolha de um ponto nos cromossomos dos pais e a partir dele a informação genética é trocada. Os recortes são colados dando origem aos filhos, como é mostrado na Figura 10.

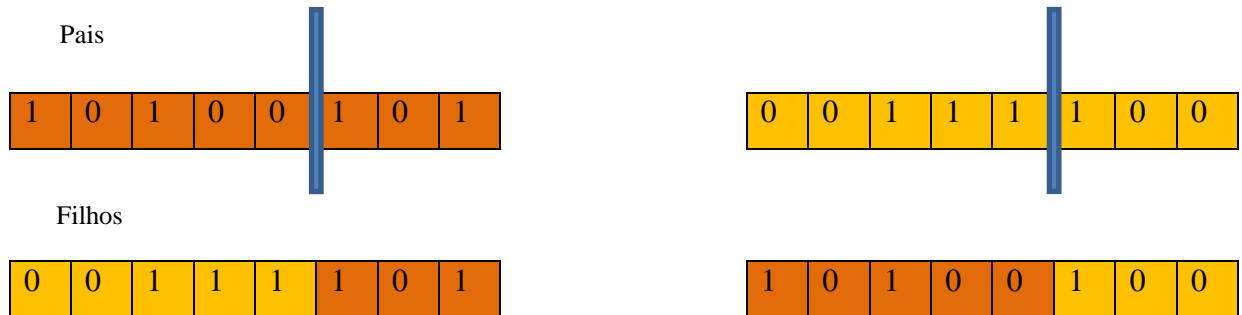


Figura 10 – Cruzamento de um ponto

- II. Cruzamento Multipontos: Generalização do modelo anterior, porém com mais de um ponto de cruzamento sendo utilizado. Esta exemplificado na Figura 11.

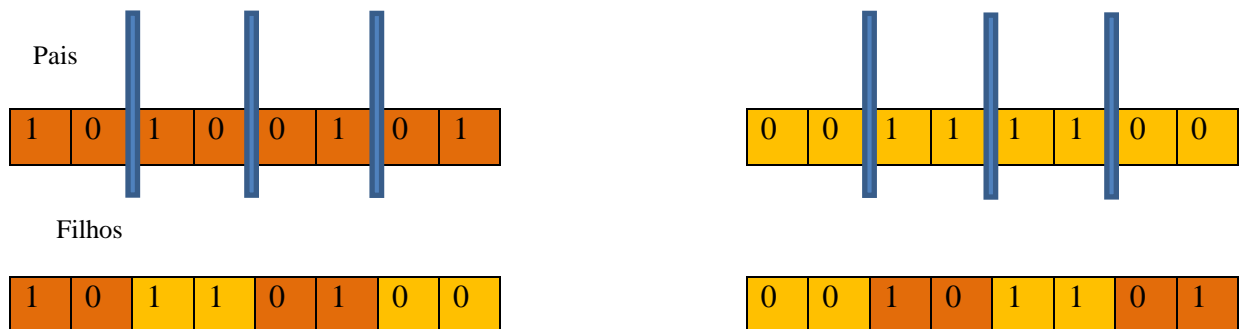


Figura 11 – Cruzamento multipontos

- III. Cruzamento Uniforme: Não há ponto de cruzamento, mas determina, utilizando uma máscara, que indica quais os genes de cada cromossomo que cada filho herdará. Um valor 1 na máscara indica que o Gene do pai A será herdado pelo filho C, e o gene correspondente do pai B será herdado pelo filho D. O inverso ocorre se o valor é 0 na máscara. A situação é ilustrada através da Figura 12.

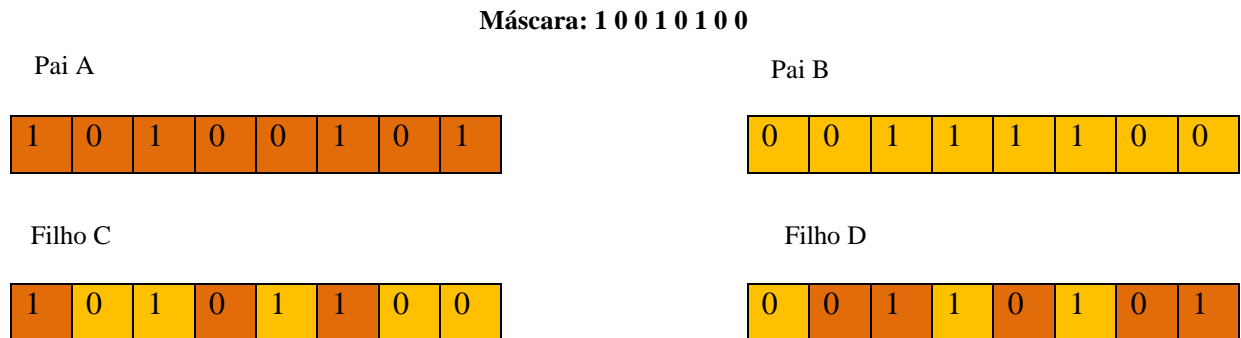


Figura 12 – Cruzamento Uniforme.

Leal (2007), apresenta mais dois cruzamentos, a saber: cruzamento por variável (o número de pontos de corte será determinado por variável, ou seja, se consideradas 4 variáveis tem-se 4 pontos de corte) e cruzamento entre vários indivíduos (seleciona-se o indivíduo base e depois, para cada variável, determina-se aleatoriamente um parceiro e um ponto de corte).

2.7.12 - Mutação

Colin (2007) apresenta o operador de mutação como aquele que seleciona um gene aleatoriamente e o muda para um alelo diferente, criando um novo cromossomo. Um alelo são as formas que um gene pode assumir, no caso da representação binária, os alelos são 0 e 1. A mutação introduz variabilidade na população e impede a queda em armadilhas de mínimos locais. A Figura 13 apresenta o exemplo da mutação de um cromossomo.

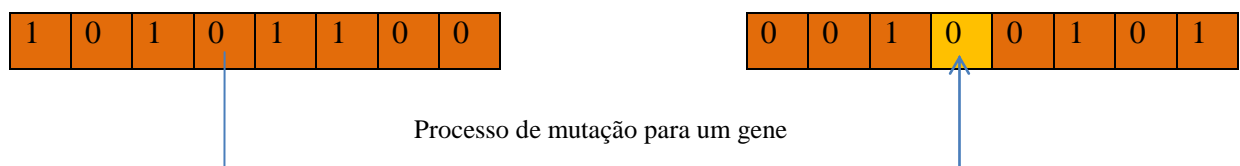


Figura 13 – Operação de Mutação

2.7.13 - Parâmetros para algoritmos genéticos

Os parâmetros são as características que afetam o desempenho do AG. Frequentemente, consideram-se fatores decisivos para o bom funcionamento destas Meta-heurísticas as seguintes características:

- a. Tamanho da população: o desempenho pode cair se a população é muito pequena e, por outro lado, exigirá recursos computacionais muito elevados se for grande demais;
- b. Taxa de cruzamento: indica a velocidade com a qual novos indivíduos são introduzidos na população. Se for muito alta, pode levar a estruturas com gerações que apresentam boas soluções saírem muito rápido da população;
- c. Taxa de mutação: quando baixa, faz com que a busca não fique presa em determinadas regiões do espaço de busca e gera a possibilidade de que mais pontos sejam explorados. Se esta taxa for muito alta, a mutação passa a ter um caráter aleatório que não será interessante;
- d. Intervalo de Geração: é o controle da porcentagem da população que será substituída para a próxima geração, já que, se for muito grande, a grande parte da população acaba sendo substituída, gerando perdas de boas soluções;
- e. Critério de Parada: é quando se deseja que o algoritmo pare. Existem diversas técnicas, sendo que um AG deve parar quando alcançar um determinado número de gerações, quando o ótimo da solução for conhecido e o algoritmo o atingiu e quando ocorre convergência, ou seja, não ocorre melhora com relação ao indivíduo mais apto.

2.8 Simulated Annealing

Simulated Annealing (Recozimento Simulado, ou Recozimento de Metais) é apresentado por Biscaia Junior (2002) como um processo de alteração de propriedades de um material metálico por aquecimento e resfriamento lento. Um metal é uma estrutura cristalina, onde sua organização molecular é disposta tridimensionalmente em rede e de forma bem definida, repetindo-se em toda a estrutura.

Quando o metal é resfriado, a estrutura cristalina, composta por cristais, é deformada, surgindo pontos de tensão. Cristais deformados têm maior energia do que os não deformados, devido à desorganização da estrutura nas interfaces entre os grãos. Esse quadro estrutural fica estável nesta temperatura, porém, encontrando possibilidade, os átomos tendem a se deslocar, em busca de um arranjo melhor, com menor energia e mais regular.

Um cristal perfeito é criado aumentando a temperatura do metal perto do ponto de fusão, onde os átomos terão liberdade para atingir uma configuração estável. Em seguida, a diminuição lenta da temperatura faz com que o material tenha tempo para configurar os átomos da melhor forma possível, levando a minimização da energia. Todo material possui uma estrutura atômica que é mínimo global, ou seja, que é a menor energia no zero absoluto. Mínimos globais ocorrem quando estruturas ficam estáveis em uma temperatura e não há mais mobilidade para rearranjar os átomos.

2.8.1 Descrição do Método

Conforme Oliveira (2006), em *Simulated Annealing*, cada configuração é definida por um conjunto de átomos de energia E e temperatura T . A cada passo do algoritmo, os átomos sofrem deslocamentos aleatórios, provocando uma alteração ΔE no sistema. Se a nova energia ΔE é menor ou igual a zero, o deslocamento é aceito, se não, o deslocamento é aceito dependendo da probabilidade da configuração dada por:

$$P(\Delta E) = e^{\left(\frac{\Delta E}{K_b T}\right)} \quad (10)$$

Onde K_b é a constante de Boltzmann, que relaciona energia e temperatura de moléculas e tem valor $1,32 \times 10^{-23} \text{ J/K}$.

Um número aleatório uniformemente distribuído no intervalo $[0, 1]$ é calculado e comparado com $P(\Delta E)$. A configuração é aceita se o número aleatório for menor do que $P(\Delta E)$, caso contrário, é rejeitado e a configuração anterior é utilizada como ponto de partida para o próximo passo. Diversos autores demonstram esta técnica de resolução, que é baseada no trabalho de Metropolis *et al.* (1953, *apud* Oliveira, 2006).

Conforme Becceneri *et al.* (2012), repetindo-se diversas vezes o procedimento básico anteriormente descrito, o movimento dos átomos do material na temperatura T é simulado. Substituindo a função objetivo pela equação da energia, e definindo-se as configurações atômicas como o conjunto de variáveis de projeto, o procedimento gera um conjunto de configurações de um problema de otimização a uma certa temperatura. Tal temperatura não passa de um parâmetro de controle e a constante de Boltzmann é um fator de escala que, em geral, se iguala à unidade.

Becceneri *et al.* (2012), prossegue a descrição do algoritmo, conforme os passos a seguir. Primeiramente, o sistema é fundido em elevada temperatura, em seguida, a temperatura cai até que ele congele e não ocorra melhoria no valor da função objetivo. Em cada temperatura, a simulação é conduzida em um número de vezes tal que o estado de equilíbrio seja atingido. O método segue conforme abaixo:

Sejam:

- x é a solução gerada na iteração atual;
- x^* é a melhor solução encontrada;
- f é a função objetivo;
- T_0 é a temperatura inicial;
- T é a temperatura atual;
- p é um número real, entre 0 e 1, gerado aleatoriamente.

Então:

- 1) Atribuir a x uma solução inicial;
- 2) Fazer $x^* = x$;
- 3) Definir T_0 ;
- 4) Verificar se as condições de parada foram satisfeitas;
- 5) Escolher x' vizinho de x ;
- 6) Calcular $\Delta E = f(x') - f(x)$;
- 7) Verificar se $\Delta E < 0$;
- 8) Se o passo anterior é verdadeiro, fazer $x = x'$. E se $f(x') < f(x^*)$ fazer $x^* = x'$;
- 9) Se o item 7 não procede, gera-se um número aleatório p' . Se $p' < e^{\frac{\Delta E}{T}}$, fazer $x = x'$.
- 10) Volta ao item 5.

- 11) Atualizar T ;
- 12) Retornar ao passo 4.

No início, à melhor solução, é atribuída a única solução conhecida. O pseudocódigo do algoritmo pode ser visualizado por meio da Tabela 5.

Tabela 5 – Simulated Annealing em Pseudocódigo

Simulated Annealing em Pseudocódigo

Início

Atribuir uma solução inicial

Fazer melhor solução receber a solução inicial

Definir a temperatura inicial

Enquanto a condição de parada não for satisfeita

Início

Comparar solução vizinha com a solução atual, substituir se a primeira for melhor

Comparar solução atual com a melhor solução, substituir se a primeira for melhor

Fim Enquanto

Atualizar temperatura

Repetir enquanto temperatura final não for alcançada

Fim

Fonte: Becceneri (2012), adaptado.

2.8.2 Parâmetros para Simulated Annealing

Becceneri *et al.* (2012) destaca que a solução inicial pode ser obtida de maneira aleatória, baseada na experiência do pesquisador.

O parâmetro da temperatura inicial deve ser suficientemente grande para que todas as transições sejam inicialmente aceitas. Este valor depende do tipo de problema e da instância analisada. Algumas alternativas são encontradas na literatura e, usualmente, a alternativa a seguir é considerada:

$$T_o = \ln f(x_o) \quad (11)$$

Onde $f(x_o)$ é o valor da função objetivo para a solução inicial.

Conselheiro (1999) destaca que, na teoria, a temperatura final chegará a zero. Porém, a probabilidade de aceitação e uma solução é praticamente nula para valores pequenos de T. Logo, a busca pode ser encerrada com valores maiores do que zero. O valor final do parâmetro de controle é geralmente uma função do número de soluções possíveis ou do número de soluções que se deseja analisar para uma mesma temperatura. Sendo normalmente 20 ou 30 soluções analisadas para uma mesma temperatura, que é corrigida por um fator de ajuste α correspondente de 80 a 99 de temperatura anterior, o valor final pode ser escrito como:

$$T_f = t_o \alpha^{30} \quad (12)$$

O mesmo autor cita outro critério, que consiste em interromper a busca após analisar um número de soluções sem que haja uma melhora de 2% na função objetivo, ou também, obedecendo critérios de tempo de busca.

2.9 Particle Swarm

Particle Swarm, ou em português, Enxame de Partículas, ou *Particle Swarm Optimization* (PSO), é uma método heurístico proposto por Kennedy e Eberhart (1995, *apud* Luz, 2008) inspirado no comportamento de algumas espécies animais, como revoada de pássaros e cardume de peixes. O método faz uma analogia ao comportamento cooperativo de troca de informação observado nestes grupos. O trabalho gerou a proposta de um algoritmo não-determinístico (estocástico) e que, apesar de muito robusto, mostrou-se relativamente simples de ser implementado computacionalmente.

O “enxame” faz referência a todo o grupo de indivíduos, e “partícula” se refere a um indivíduo específico. Durante o convívio social, estes grupos de animais trocam informações a respeito de rotas de deslocamento, localização de alimentos, existência de predadores etc.

Estas informações são compartilhadas e a partir do momento em que um indivíduo (partícula) passa a ter um determinado conhecimento, os demais passam a tê-lo também, levando a aprendizagem de todo o grupo (enxame). A Figura 14 demonstra o comportamento social de uma revoada de pássaros que, mesmo com o céu inteiro para voar, decidem adotar uma posição peculiar.



Figura 14 – Revoada de pássaros
(MIRROR, 2012)

Luz (2008) relata que a característica mais importante presente em diversos organismos vivos é a de avaliar estímulos, para classificá-los em positivos ou negativos. Então, o aprendizado pode ser visto como mudanças que possibilitem ao organismo melhorar a avaliação média de seu ambiente, a fim de definir o que é bom ou ruim.

2.9.1 Algoritmo de otimização PSO

Em um algoritmo PSO, as partículas procuram no espaço de busca uma solução viável a um problema proposto. Cada indivíduo (partícula) possui uma velocidade com a qual explora o espaço e uma memória para guardar qual é a melhor solução que já foi atingida. Também, é necessário guardar qual a melhor posição encontrada em toda a população. O pseudocódigo para um PSO pode ser representado conforme a Tabela 6.

Tabela 6 – *Particle Swarm* em Pseudocódigo

Particle Swarm

Início**Gerar** população inicial de partículas com posições e velocidades aleatórias**Enquanto** critério de parada não for satisfeito, **Para Toda** partícula **Faça****Início****Calcular** o valor da função objetivo para cada partícula**Comparar** resultado com *pbest*, atualizar *pbest* se o resultado atual for melhor**Comparar** valor obtido com *gbest*, atualizar *gbest* se o valor obtido for melhor**Atualizar** velocidade e posição da partícula**Fim Enquanto**

Fonte: Muller (2007) , Waintraub (2009) e Luz (2008), adaptado.

Muller (2007) apresenta um PSO conforme descrito a seguir. O espaço de representação será o bidimensional que foi originalmente proposto por Kennedy e Eberhart (1995, *apud* Luz, 2008) e possuem uma infinidade de aplicações com resultados promissores.

Sendo o espaço bidimensional, cada partícula possui uma posição no espaço, ou seja, uma coordenada (x, y) e uma velocidade v_i com a qual percorre o espaço. A cada iteração, a velocidade é atualizada, de acordo com a equação abaixo:

$$v_i^{it+1} = v_i^{it} + (c_1^{it} rand^{it} (pbest_i^{it} - x_i^{it})) + (c_2^{it} rand^{it} (gbest^{it} - x_i^{it})), \quad (13)$$

$$x_i^{it+1} = x_i^{it} + v_i^{it+1} \quad (14)$$

Em que:

 v_i : velocidade atual da partícula i ; c_1, c_2 : parâmetros de confiança;*rand* : função aleatória;*pbest_i* : melhor posição que a partícula i já obteve durante a busca;*gbest* : melhor posição encontrada pelas partículas no enxame; x_i : posição atual da partícula i ;*it* : iteração atual.

Esta é a versão padrão ou tradicional para um PSO. As velocidades são armazenadas em um vetor $V_i = (v_1, v_2, \dots, v_n)$, assim como as posições das partículas $X_i = (x_1, x_2, \dots, x_n)$ e as

melhores posições encontradas dentre todas as partículas
 $pbest_i = (pbest_{1i}, pbest_{2i}, \dots, pbest_{ni})$.

2.9.2 Parâmetros de Confiança para um PSO

Os parâmetros de confiança c_1, c_2 indicam, respectivamente, o quanto uma partícula confia na sua própria posição e o quanto confia no enxame. Estes parâmetros podem ser fixados ou variar em cada iteração. Em busca de melhores resultados, a cada problema, estes métodos são variam, haja vista que se trata de uma meta-heurística. Ratnaweera (2004, *apud*, Muller, 2007), propôs a utilização de equações para definir estes parâmetros. Para o autor, a cada iteração, tem-se:

$$c_1^{it} = (c_{1\text{ fin}} - c_{1\text{ ini}}) \frac{it}{R} + c_{1\text{ ini}} \quad (15)$$

$$c_2^{it} = (c_{2\text{ fin}} - c_{2\text{ ini}}) \frac{it}{R} + c_{2\text{ ini}} \quad (16)$$

Em que:

$c_{1\text{ ini}}$: valor inicial para o parâmetro de confiança cognitivo;

$c_{1\text{ fin}}$: valor final para o parâmetro de confiança cognitivo;

$c_{2\text{ ini}}$: valor inicial para o parâmetro de confiança social;

$c_{2\text{ fin}}$: valor final para o parâmetro de confiança social;

R = número de iterações;

it = iteração atual.

É imediato que a outra coordenada do ponto é fornecida pela função objetivo, formando o par (x,y) do ponto. O que se procura é, então, o ponto x que maximize ou minimize a função objetivo.

2.9.2 Função Aleatória e Aceleração da Distância

Conforme Muller (2007), a função aleatória *rand* é uma função randômica que gera aleatoriamente números entre 0 e 1, com a finalidade de manter a diversidade no enxame. A equação que atualiza a velocidade da partícula é composta por dois termos $(pbest_i^{it} - x_i^{it})$ e $(gbest^{it} - x_i^{it})$ que representam, respectivamente, a distância entre a melhor posição já encontrada pela partícula *i* e sua posição atual, e a distância entre a melhor posição encontrada pelo enxame e a posição atual da partícula *i*.

Em geral, conforme Muller (2007) e Waintraub (2009), o critério de parada comumente usado em PSO é o número de iterações.

2.9.3 Limitador de Velocidade e Peso Inercial

Muller (2007) e Waintraub (2009) defendem que a versão tradicional do PSO possui a necessidade de um controle de velocidade de partículas. Segundo os autores, um valor muito alto pode fazer com que a partícula ultrapasse uma posição ótima e um valor muito baixo pode ser insuficiente para alcançar o ótimo.

O valor máximo, ou v_{max} , é especificado pelo usuário de acordo com o problema, normalmente entre [-4,4]. Assim, se a velocidade v_i for superior a 4, a ela é atribuída o valor 4. Caso a velocidade encontrada for inferior a -4, seu valor passará a ser -4.

Para Muller (2007), tal limitador de velocidade, ainda que importante e muito encontrado na literatura, apresenta alguns empecilhos na implementação de um PSO:

- Rendimento a baixo do esperado em relação a outras Meta-heurísticas;
- Dificuldades relacionadas ao refinamento da solução por não haver, muitas vezes, ajuste da velocidade após encontrar a área em que o ótimo se localiza.

Uma alternativa é a utilização de um peso inercial w para controle de velocidade, que é uma carga na equação da velocidade proposta. A nova equação fica da seguinte forma:

$$v_i^{it+1} = wv_i^{t+1} + (c_1^{it}rand^{it}(pbest_i^{it} - x_i^{it})) + (c_2^{it}rand^{it}(gbest^{it} - x_i^{it})) \quad (17)$$

Para Luz (2008), esta constante tem influência direta sobre o controle de velocidade, conforme os valores a seguir:

- Se $w < 0,8$: o algoritmo opera em modo de busca local, explorando fortemente frações do espaço de busca;
- $w < 1,2$: o algoritmo trabalha no modo de exploração global, agindo sobre maior área do espaço, buscando espaços não explorados previamente;
- $0,8 < w < 1,2$: é o ponto intermediário para um bom balanço entre a busca global e local, e por consequência aqui se concentram os melhores valores de inércia para a maior parte dos problemas.

3 DESENVOLVIMENTO

3.1 Contextualização

Fleury (2008) aponta que é função do Engenheiro de Produção a criação de modelos que capturem as dimensões mais relevantes de problemas, quando estes tornam-se complexos demais para serem resolvidos somente com a experiência e intuição. Neste contexto, o Engenheiro de Produção deve conhecer as ferramentas matemáticas e estatísticas que subsidiem o processo de tomada de decisão.

Com frequência, problemas categorizados como NP-Completo ou NP-Difíceis surgem em situações reais, levando a necessidade de aplicação de Meta-heurísticas. Isso pode ser observado nos anais de congressos, trabalhos de graduação, dissertações e teses, contidas nas mais diversas bases de dados, comprovando a aplicabilidade destes métodos em várias áreas do conhecimento.

O Brasil possui alguns espaços de publicação em Engenharia de Produção e em Pesquisa Operacional de reconhecida importância. Dentre eles, é possível destacar:

- ENEPEP (Encontro Nacional de Engenharia de Produção): evento nacional organizado anualmente pela Associação Brasileira de Engenharia de Produção (ABEPRO) que, além de representar um espaço para a divulgação da produção científica, é classificado como fórum de discussão para a Engenharia de Produção no país;
- SIMPEP (Simpósio de Engenharia de Produção): organizado pelo Departamento de Engenharia de Produção da Universidade Estadual Paulista (DEP-UNESP), aborda temas em todas as áreas de Engenharia de Produção, possuindo alto nível de discussão. É considerado um dos maiores congressos sobre o tema no país, devido à qualidade das publicações que congrega;
- Revista GEPROS (Gestão da Produção, Operações e Sistemas): publicação da Universidade Estadual Paulista (UNESP), tem como foco divulgar artigos inéditos em Engenharia de Produção, por meio eletrônico e impresso;
- Revista Gestão & Produção: a “G&P” é uma publicação da Universidade Federal de Santa Catarina (UFSCAR) que aceita trabalhos inéditos, que representem avanços para a Engenharia de Produção;

- Revista Pesquisa Operacional: ou “Revista PO”, visa divulgar trabalhos em Pesquisa Operacional, afim de integrar e compartilhar conhecimentos técnicos na área. As publicações desta revista podem ser acessadas através do SCIELO. É pertencente à Sociedade Brasileira de Pesquisa Operacional (Sobrapo);
- Revista Pesquisa Operacional para o Desenvolvimento: ou “PODes” publica trabalhos e revisões de literatura em Pesquisa Operacional, em todas as suas subáreas. Engloba artigos em português, inglês e espanhol e é também pertencente à Sobrapo. Os artigos também podem ser acessados pelo SCIELO.

Outro importante espaço de publicação, mas que não se restringe à apenas publicações brasileiras é:

- SCIELO (*Scientific Electronic Libray Online*): é um modelo para a publicação eletrônica cooperativa de periódicos na Internet. Especialmente desenvolvido para responder as necessidades da comunicação científica nos países em desenvolvimento e particularmente na América Latina e Caribe (SCIELO, 2012). Foi organizado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e o Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde (BIREME), contando desde 2002 com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq);

Estas publicações podem ser livremente acessadas por meio da internet. Uma pesquisa sobre Meta-heurísticas em suas bases de dados levou à diversos trabalhos publicados, o que certifica a importância e qualifica o método como relevante para a Engenharia de Produção.

3.2 Meta-heurísticas utilizadas em Engenharia de Produção

Algoritmos Genéticos, *Simulated Annealing* e *Particle Swarm* são algumas das meta-heurísticas mais utilizadas na resolução de problemas em diversas áreas da Engenharia, conforme diversas publicações presentes na literatura. Além disso, são de fácil compreensão, pois fazem analogia com modelos da natureza e podem ser implementadas por meio de linguagens de programação ao alcance do pesquisador.

A Associação Brasileira de Engenharia de Produção (ABEPRO) divide a Engenharia de Produção nas seguintes subáreas:

- Engenharia de Operações e Processos de Produção;
- Logística;
- Pesquisa Operacional;
- Engenharia da Qualidade;
- Engenharia do Produto;
- Engenharia Organizacional;
- Engenharia Econômica;
- Engenharia do Trabalho;
- Engenharia da Sustentabilidade;
- Educação em Engenharia de Produção;

Com estas subdivisões da Engenharia de Produção, ao avaliar o cenário no qual um problema está inserido, é possível enquadrá-lo em uma destas subáreas. Um problema, para ser resolvido por meio de Meta-heurísticas, pode pertencer a qualquer uma delas, desde que seja possível criar um modelo que represente o problema e de que este se enquadre nos padrões para a resolução por estes métodos.

Os pseudocódigos apresentados na literatura para cada um dos três métodos descritos devem ser adaptados para todos os problemas e codificados em uma linguagem de programação que possibilite a comunicação das instruções ao computador. Algumas das linguagens de programação mais utilizadas atualmente são: Pascal, C, C++, Fortran, Visual Basic, Delphi, Java, entre outras.

3.3 Coleta de Dados

Por meio de pesquisa bibliométrica, os trabalhos publicados nos últimos dez anos nas principais base de dados do Brasi e algumas do exterior, que apresentaram Meta-heurísticas aplicadas à problemas práticos, foram levantados e podem ser encontrados no Apêndice A.

Nove base de dados foram consultadas: ENEGEP, SIMPEP, Revista GEPROS, Revista PO, Revista Produção, Revista G&P, Revista PODEs, SCIELO e DEP/UEM. Para esta última base de dados, foram consideradas as publicações na biblioteca de Trabalhos de Conclusão de Curso, disponível no sítio do Departamento de Engenharia de Produção da Universidade Estadual de Maringá. O interesse nesta base se deu devido ao fato de que, neste departamento, os estudos para a pesquisa foram desenvolvidos. A cada trabalho, foi atribuída a área da Engenharia de Produção a qual se enquadra. As publicações foram classificadas também quanto ao ano de divulgação e linguagem de programação utilizada para implementação, quando esta constava no texto.

As publicações das revistas que divulgam seus trabalhos na base de dados do SCIELO e que estão aqui citadas foram classificadas quanto a fonte pelo nome da revista. Assim, os trabalhos que possuem como fonte de referência o SCIELO são provenientes de diversas outras publicações que também contribuíram com a aplicação de Meta-heurística e estão registradas neste domínio.

3.4 Resultados e Discussões

3.4.1 Aplicação de Meta-heurísticas em Engenharia de Produção

A pesquisa ocorreu no período compreendido entre abril de 2012 e setembro de 2012. Um total de 102 publicações foram identificadas. Os resultados estão descritos a seguir.

Em um primeiro momento, foi realizada uma análise quanto ao número de publicações por ano. Este estudo visou verificar se há uma tendência de aumento ou diminuição do número de publicações no decorrer do tempo. Para esta análise, os trabalhos do ano de 2012 foram desconsiderados, haja vista que, até a conclusão do levantamento contido neste trabalho, as bases de dados não haviam concluído sua publicações para o presente ano. A Figura 15 apresenta o gráfico de barras para a quantidade de trabalhos publicados por ano.

Uma linha de tendência foi inserida ao gráfico de barras apresentado na Figura 14. A linha de tendência ascendente indica que, mesmo com a existência de altos e baixos no número de publicações no decorrer dos anos, se verifica uma evolução positiva no sentido do aumento do número de publicações.

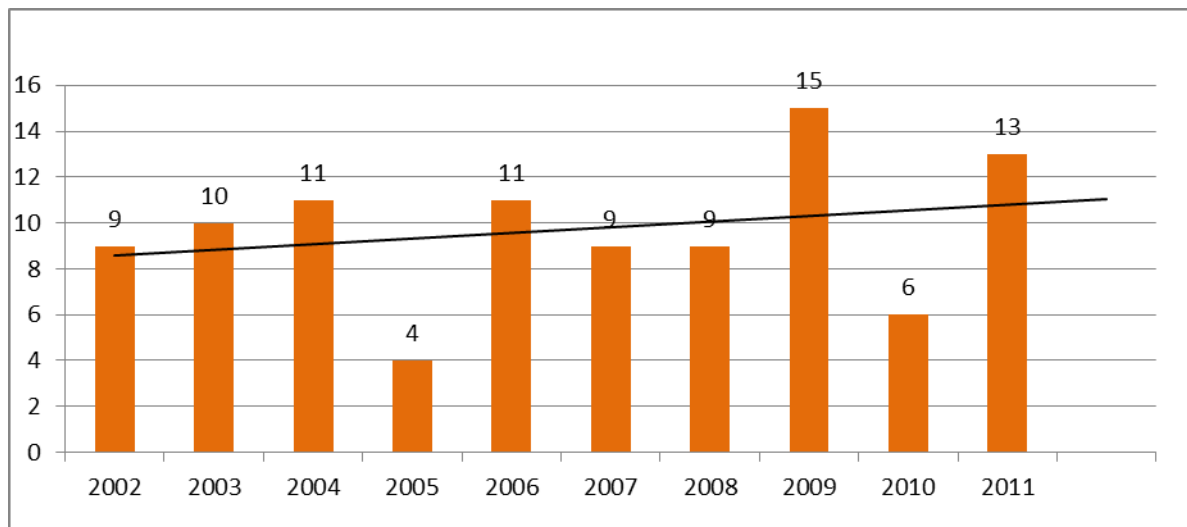


Figura 15 – Gráfico do número de publicações por ano

Dando continuidade à análise, foi verificada a distribuição dos trabalhos quanto às fontes de publicação. Esta análise visa identificar em quais bases de dados pode ser encontrado o maior número de publicações na área pesquisada. A Figura 16 mostra o gráfico de barras para o número de publicações por fonte. Do total de trabalhos avaliado, 39% foram publicados no ENEGEP, enquanto o SIMPEP obteve um total de 18%. Juntos, os dois eventos publicaram 47% dos trabalhos. Estes dois eventos são classificados como os maiores do país, em quantidade e qualidade de publicações para a Engenharia de Produção. Dado tal reconhecimento, pesquisadores de diversas instituições de ensino superior dedicam esforços para publicação de seus trabalhos nestes eventos, sob a forma de anais.

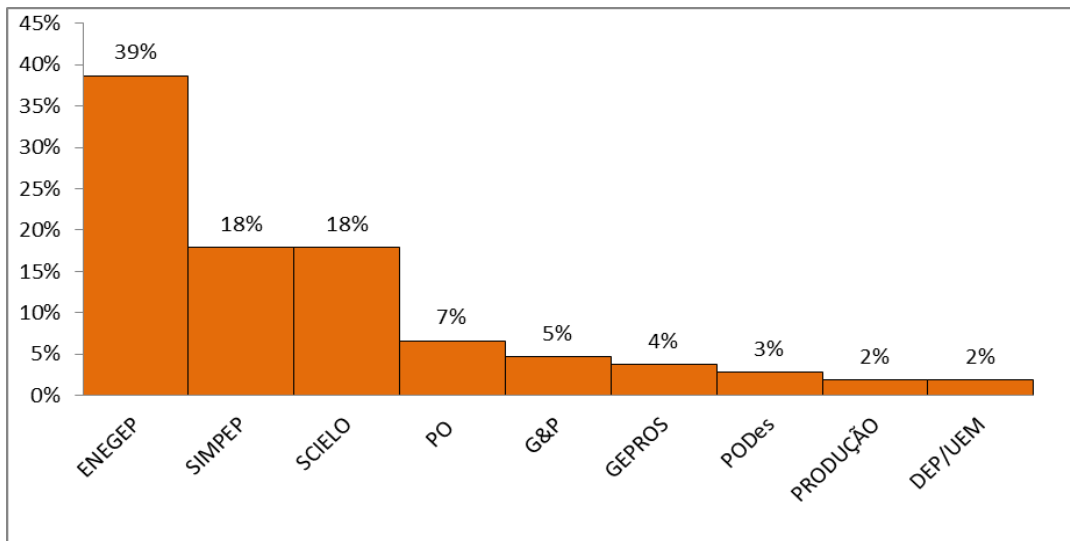


Figura 16 – Gráfico do número de publicações por fonte

Foi procedida, também, uma análise quanto as linguagem de programação que têm sido mais utilizadas na implementação dos métodos. Com esta verificação, busca-se reconhecer quais linguagens têm sido mais efetivas para a implementar as três Meta-heurísticas estudadas. Dentre os trabalhos em que foi possível identificar a linguagem de programação utilizada, o resultado estão contidos no gráfico da Figura 17.

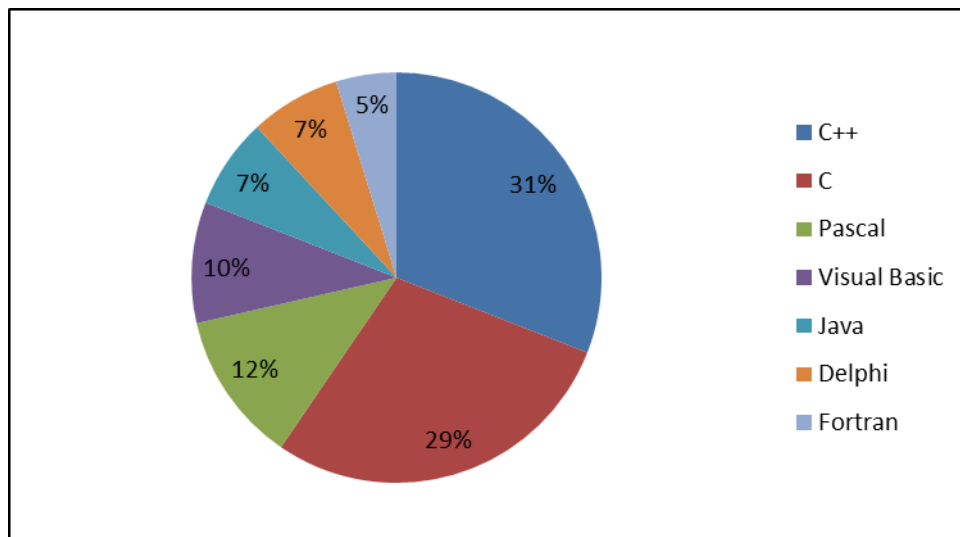


Figura 17 – Gráfico das linguagens de programação utilizadas

Como é possível observar, a grande parte das publicações utiliza linguagens tradicionais. C e C++ somam juntas 60%. Merece destaque a linguagem Pascal que, apesar de ter sido criada a

mais de 40 anos, ainda é utilizada em algoritmos robustos, como as Meta-heurísticas. Linguagens mais atuais, como Java e Delphi, foram identificadas em poucas publicações.

Também, foi feita a verificação da forma pela qual cada método está sendo utilizado, de uma forma geral, entre os trabalhos publicados. Conforme o gráfico da Figura 18, foi observado que 67% dos trabalhos utilizaram Algoritmos Genéticos como solução para os problemas proposto, o que demonstra que esta Meta-heurística é a mais difundida e efetiva dentre as três. Um fato que concorre a favor do Algoritmo Genético nesta comparação é o fato deste ser o mais tradicional dentre as três Meta-heurísticas estudadas.

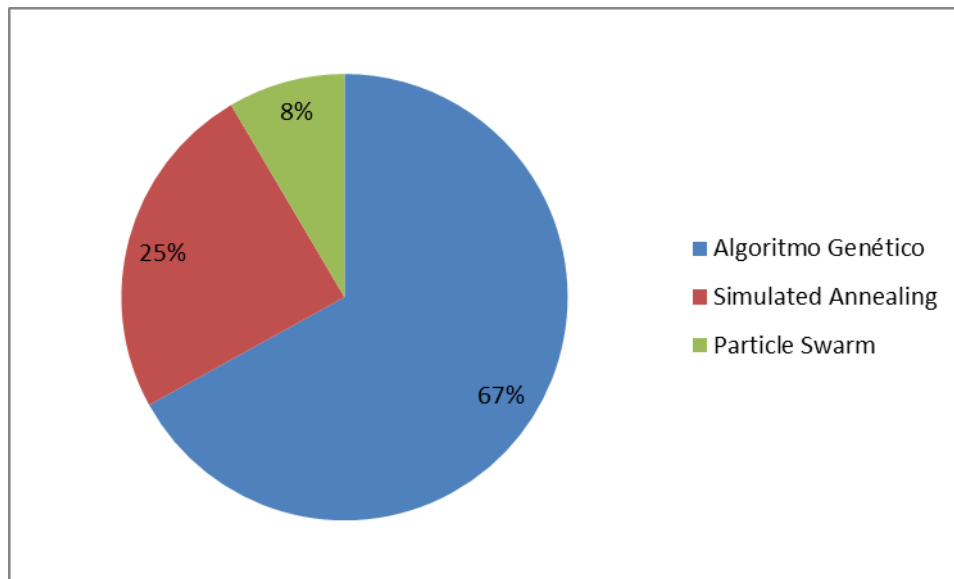


Figura 18 – Gráfico das Meta-heurísticas utilizadas

Outra análise possível é a verificação de quais áreas da Engenharia de Produção têm sido mais contempladas em termos de números de publicações em Meta-heurística. A Figura 19 apresenta o gráfico com estas informações. Do total, 60% das publicações foram aplicadas à Engenharia de Operações e Processos de Produção. Também, se destaca a área de logística, com 25% das publicações. Outras áreas em que foram encontradas aplicações para as Meta-heurísticas descritas foram Engenharia da Qualidade, 10%, Pesquisa Operacional, 5%, e Engenharia Organizacional, 1%.

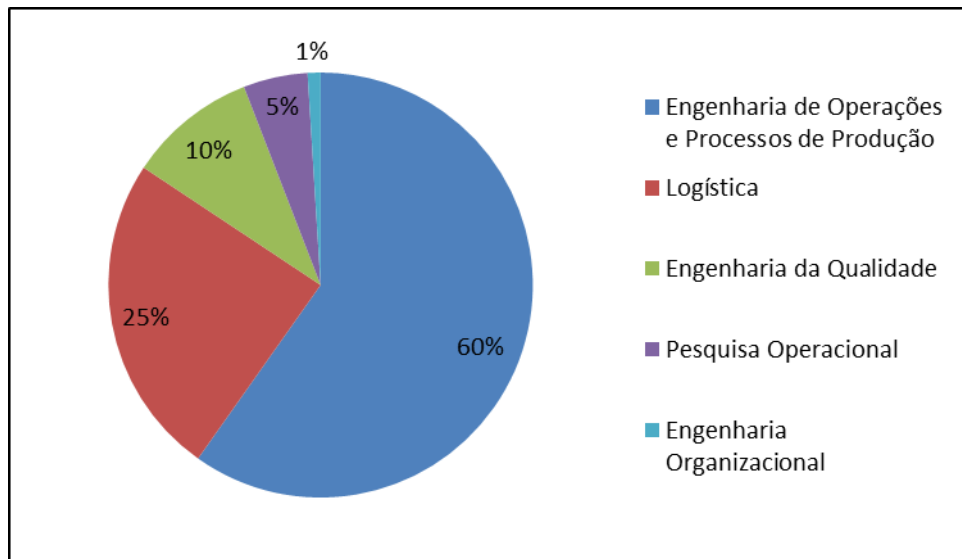


Figura 19 – Gráfico das áreas de aplicação

Os dados coletados demonstram, conforme a Figura 20, que 65% das publicações que utilizaram Algoritmos Genéticos pertencem à área de Operações e Processos de Produção, ao passo que 23% estão contidas na área de logística, mostrando uma tendência maior para aplicabilidade do método nestas áreas.

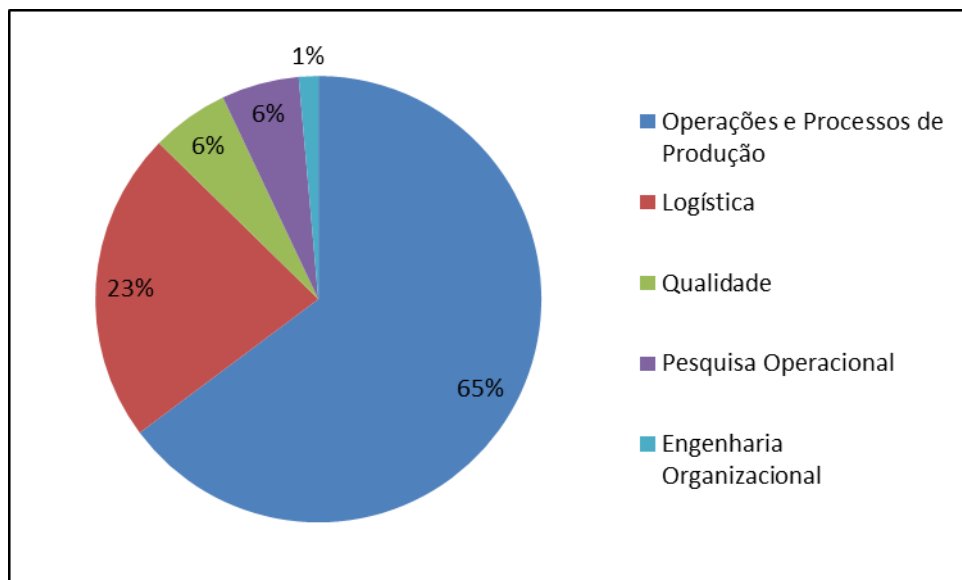


Figura 20 – Aplicação de Algoritmos Genéticos em Engenharia de Produção

Em concordância com o método anterior, verificou-se maior aplicabilidade para *Simulated Annealing* nas áreas de Operações e Processos de Produção e Logística, dentre as publicações que utilizaram este método. Todavia, houve aumento para 31% para a área de Logística em

relação ao AG, e diminuição para 50% para Operações e Processos de Produção, conforme pode ser verificado por meio do Gráfico da Figura 21.

Também, é possível observar um sensível aumento do percentual de trabalhos que utilizam esse método para a área de qualidade, em relação ao anterior, totalizando 15%. Não houve problemas identificados na área de Engenharia Organizacional e a área de Pesquisa Operacional é similar ao anterior.

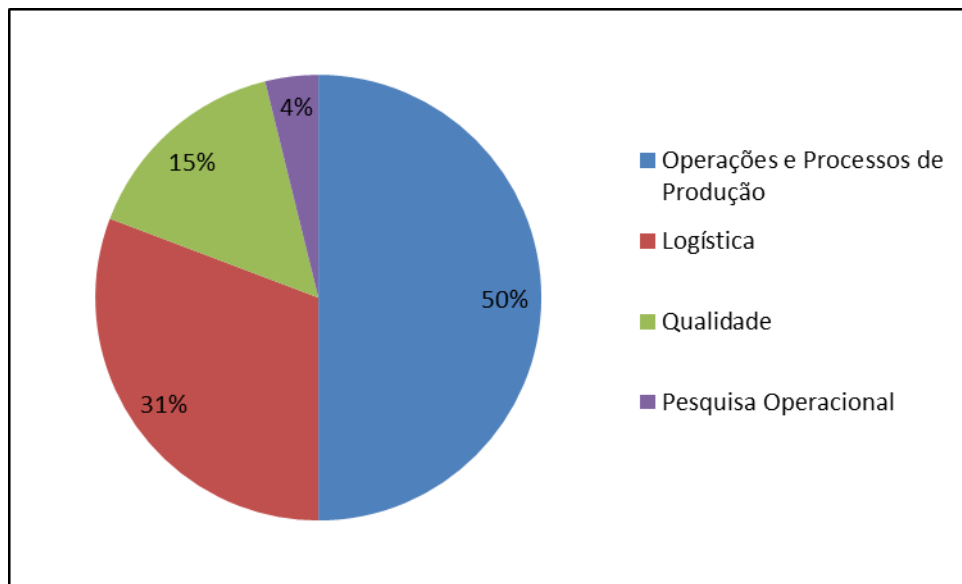


Figura 21 – Aplicação de *Simulated Annealing* em Engenharia de Produção

Do total de trabalhos publicados com a utilização do PSO, 56% foram aplicados à Engenharia de Operações e Processos de Produção, similar ao que ocorre no SA. Por sua vez, a área de Logística, com 22%, fica próxima percentual encontrado para o AG. Isso pode ser observado por meio do Gráfico da Figura 22.

Também, é possível notar crescimento do percentual de trabalhos publicados na área de Engenharia da Qualidade, para 22%, superior aos outros dois. Para o PSO, não foram registrados trabalhos na área de Pesquisa Operacional e Engenharia Organizacional.

De forma geral, os três métodos possuem maior tendência de aplicabilidade para as áreas de Operações e Processos de Produção, e Logística, nesta ordem.

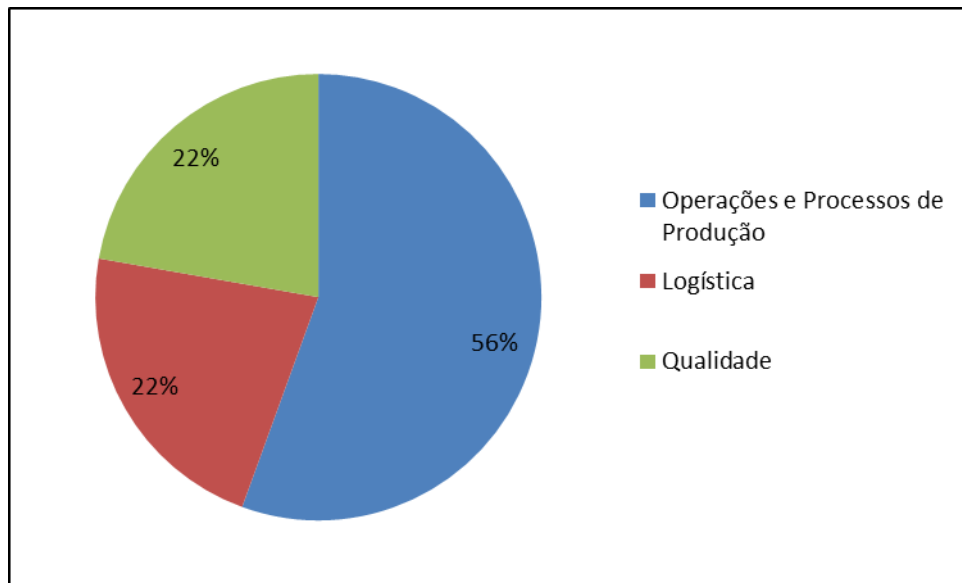


Figura 22 – Aplicação de *Particle Swarm* em Engenharia de Produção

3.4.2 Aplicação de Meta-heurísticas nas áreas da Engenharia de Produção

A distribuição das porcentagens de publicações por método para a área de Operações e Processos de Produção são apresentadas no gráfico da Figura 23. Por meio deste gráfico, é possível observar a predominância da utilização de Algoritmos Genéticos, em 72% das publicações.

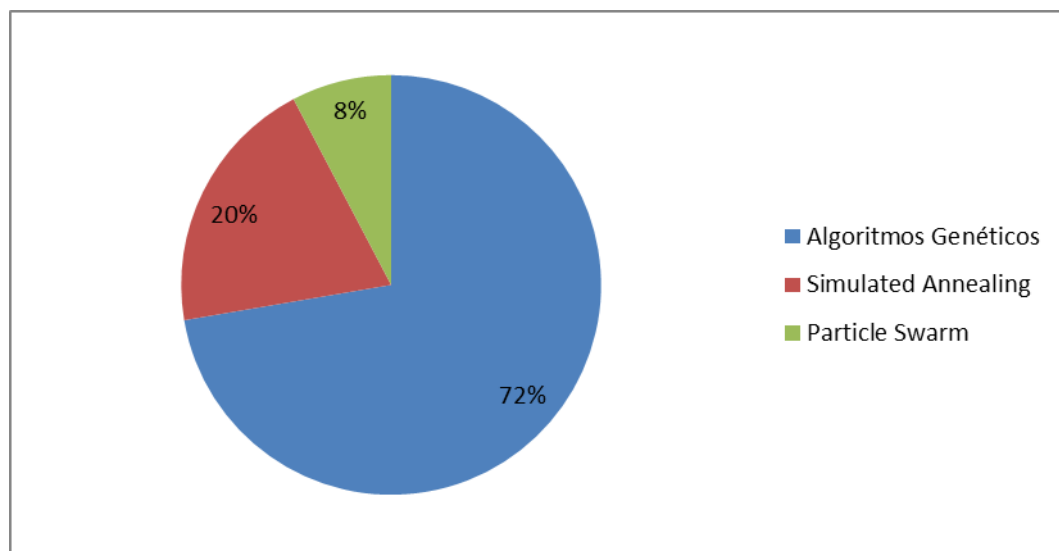


Figura 23 – Gráfico da aplicação de Meta-heurísticas em à Engenharia de Operações e Processos de Produção

Os trabalhos publicados com a utilização de Algoritmos Genéticos em Engenharia de Operações e Processos de Produção foram utilizados nos seguintes casos: Problema de formação de grupos e balanceamento de linhas de montagem SMT, em Candido e Silva (2002); Otimização do planejamento mestre da produção, em Gaspar Junior *et al.* (2002); Problema de empacotamento de Bins Tridimensionais, em Silva e Soma (2002); Problema de programação de horários escolares, em Costa *et al.* (2002), e Abensur e Oliveira (2012); Planejamento e otimização do desenvolvimento de um campo de petróleo, em Pacheco *et al.* (2002), e Argoud *et al.* (2004); Problema de agrupamento celulares de máquinas e peças, em Branco *et al.* (2003); Balanceamento de linhas de produção, em Mayerle e Santos (2003); Programação da produção em um processo de fundição de Zinco, em Laporte *et al.* (2003); Problema de determinação da localização de poços de petróleo, em Pacheco *et al.* (2003); Problema de despacho econômico e ambiental, em Baptista *et al.* (2003); Dimensionamento evolutivo de usinas hidroelétricas, em Carneiro e Silva Filho (2004); Otimização de uma célula de manufatura, em Galhardo *et al.* (2004); Problema de determinação de uma configuração para um sistema de co-geração de energia que utiliza gás natural como fonte de energia, em Goldbarb *et al.* (2005); Problema de localização e atividades econômicas, em Cortes e Paula Junior (2005); Determinação da operação ótima de sistemas hidrotérmicos de potência, em Carneiro *et al.* (2006); Localização de sistema servidos para cliente, em Lannoni e Morabito (2006); Problema de alocação de antenas de transmissão, em Cortes *et al.* (2006); Problema de despacho de energia elétrica, em Leal (2007); Problema de programação de operações em um ambiente de produção Flow Shop, em Lorena *et al.* (2007); Geração de arranjos físicos distribuídos, em Pitombeira Neto (2007); Problema de superação de disjuntores pela alocação ótima de dispositivos limitadores de corrente de curto-circuito, em Borges e Ferreira (2007); Modelagem para programação da produção de sistemas de manufatura com recursos compartilhados, em Deriz *et al.* (2007); Problema de sequenciamento de Produção, em Bittencout *et al.* (2007); Formação de módulos de arranjo físico, Argoud *et al.* (2008); Alocação de dispositivos de controle em uma rede de transmissão elétrica, em Volpi e Almeida (2008); Problema de configuração de redes de distribuição, em Bento e Kagan (2008); Escalonamento de processos distribuídos, em Silveira e Uber Junior (2008); Problema da programação da produção na indústria de fundição, em Behrend (2008); Minimização do número de objetos processados e o *setup* em um problema de corte unidimensional, em Golfeto *et al.* (2009); Simulação de Processos de produção de milho, em Araújo e Librantz (2009); Problemas de sequenciamento de curto-prazo para sistemas de

produção do tipo *jobshop*, Passos e Ilha (2009); Problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção, em Ribeiro *et al.* (2009); Problema de alocação de redundâncias, em Lins e Droguett (2009); Modelagem de confiabilidade de sistemas reparáveis, em Damaso e Garcia (2009); Problema de corte guilhotinado bidimensional, em Franceschette (2009); Problema de localização de Hubs com alocação simples sem restrição de capacidade, em Carvalho *et al.* (2010); Minização do número de objetos processados, em Golfeto *et al.* (2010); Escolha de misturas de fardos de fibras para o setor de fiação têxtil, em Silveira e Uber Junior (2011); Problema de planejamento florestal, em Barros Junior e Pereira (2011); Extração de fasores fundamental de tensão e corrente em sistemas que possibilitem uma proteção de distâncias mais rápida, em Coury *et al.* (2011); Problema de *Flowshop* flexível com penalidade de adiantamento e atraso, em Arroyo e Pereira (2011); Otimização do saldo de caixa, em Moraes e Nagano (2010); Gestão de estoque, em Araújo *et al.* (2009);

A técnica de Simulated Annealing foi utilizada em 20% dos trabalhos publicados em Engenharia de Operações e Processos de Produção. Os problemas resolvidos foram: Gestão de estoque, em Araújo *et al.* (2009); Problema de programação de tripulações, em Mapa *et al.* (2003); Problema de escalonamento com restrições de recursos e múltiplos modos de processamento, em Ahonen *et al.* (2006); Programação de *flowshop* permutacional, em Barros e Moccellini (2002); Problema de corte bidimensional com rotação de itens, em Gomes *et al.* (2009); Programação da produção, em Ribeiro (2003); Construção de mapas genéticos, em Faria *et al.* (2009); Otimização de estruturas, em Kripka (2004); Problema de agendamento automático de operações no refino e fundição, em Parada *et al.* (2011); Estimativa de propriedades radiativas em um meio unidimensional, em Lobato *et al.* (2010); Determinação do peso mínimo de perfis conformados a frio de seção U, em Kripka e Pravia (2011); Problemas de planejamento florestal com restrições de integridade, em Ribeiro *et al.* (2008); Estimação de valores de parâmetro característicos de materiais, em Hasselmann e Zhang (2011).

Para problemas em Engenharia de Operações e Processos de Produção, Particle Swarm foi identificado em 8% dos casos. Foram encontradas aplicações para: Problema de localização de facilidades, em Coelho e Mariani (2003), e Crossetti *et al.* (2006); Otimização de parâmetros de projeto, em Barakati (2012); Desenho de configuração de lasers de

bombeamento de amplificadores de Raman, em Chaves *et al.* (2011); Otimização do saldo de caixa, em Moraes e Nagano (2010);

A predominância da utilização de Algoritmos Genéticos também ocorre na área de Logística, conforme é possível por meio do gráfico da Figura 24.

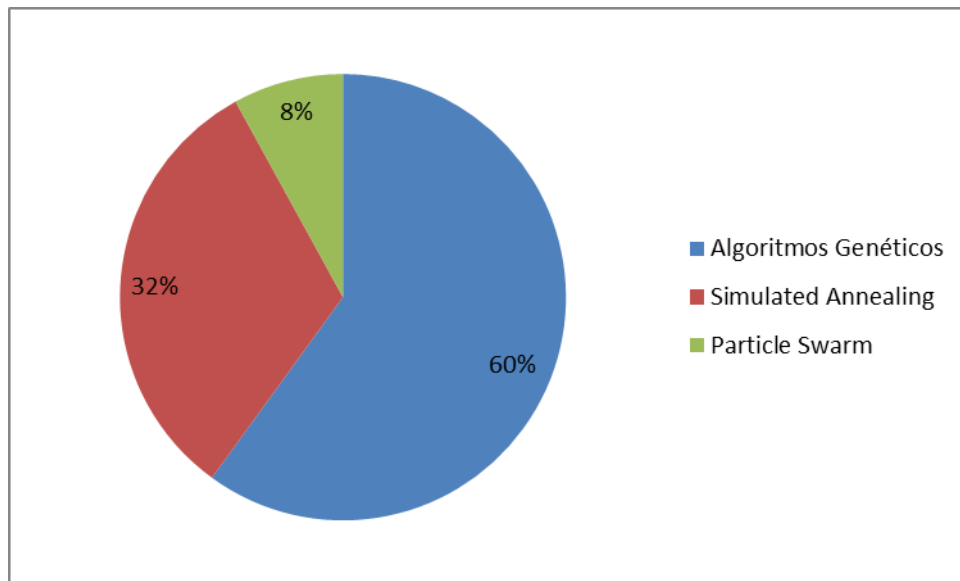


Figura 24 – Gráfico da aplicação de Meta-heurísticas à Logística

Foi observado a aplicação de AG em 60% dos casos. A aplicação de Algoritmos Genéticos em logísticas compreendeu os seguintes problemas: Problema de melhor localização de ambulâncias em um município, em Carvalho (2005); Problema do Caixeiro Viajante, em Cera *et.al.* (2004), Oliveira e Silva (2006), e Melo (2006); Problema de descarregamento de navios porta contêiners, em Chiwiacowsky e Monteiro (2012), Azevelo *et. al.* (2011), e Azevelo (2011); Problema de descarregamento de navios porta contêires integrado com o problema de roteamento de veículos, em Santos e Silva (2011); Estratégia de transmissão para o planejamento integrado de movimentação de gás natural *offshore*, em Arica *et al.* (2008); Problema das p-medianas, em Cordeiro *et al.* (2002); Problema da árvore geradora mínima, em Almeida *et al.* (2009); Problema de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas, em Belfiore e Yoshizaki (2006); Problema de localização de máxima cobertura, em Lorena (2006).

A metodologia do *Simulated Annealing* com aplicação em logística contou com 32% dos trabalhos publicados. Os problemas resolvidos com esta técnica foram: Problema do Caixeiro

Viajante, em Cera *et al.* (2004); Problema de programação de tripulações em um sistema de transporte público, em Alves *et al.* (2002); Problema de roteamento de veículos, em Castro *et al.* (2004), e Gomes Júnior (2005); Problema das p-medianas, em Aguilar *et al.* (2007); Problema de alocação de berços para atracação de navios, em Lorena *et al.* (2008), e Barros *et al.* (2009); Problema *dial-a-ride*, em Lorena e Mauri (2009).

Os problemas que utilizaram Particle Swarm em Logística representaram 8% dos casos. Os aplicações encontradas foram para: Problema do Caixeiro Viajante, em Aloise *et al.* (2006); Problema de máxima cobertura, em Prata (2012).

As publicações em Engenharia de Qualidade com as Meta-heurísticas pesquisadas podem ser visualizadas no gráfico da Figura 25.

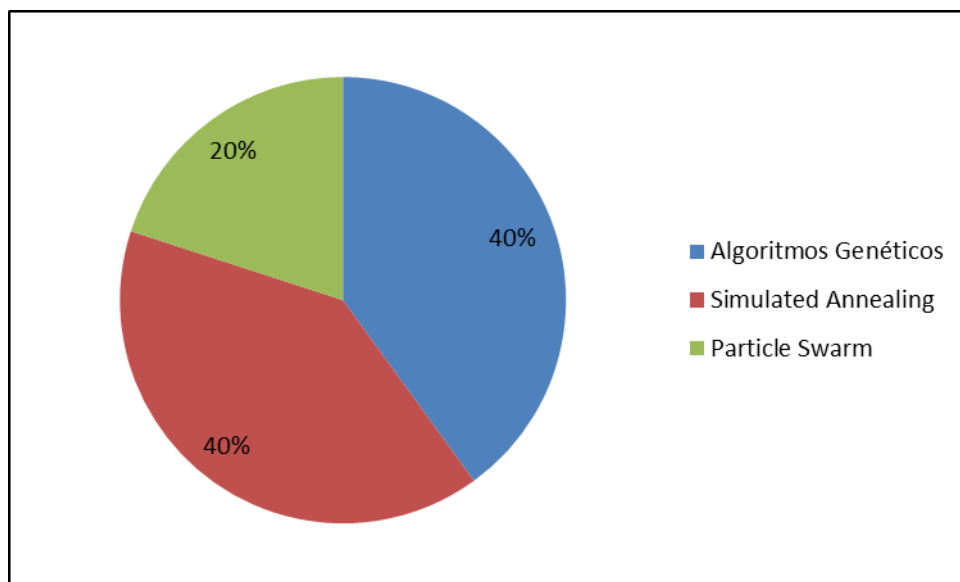


Figura 25 – Gráfico da aplicação de Meta-heurísticas à Engenharia da Qualidade

Pela análise deste gráfico, observa-se que Algoritmos Genéticos e *Simulated Annealing* predominam igualmente com relação aos trabalhos em Engenharia da Qualidade. Bassani *et al.* (2002), Bruxel Junior *et al.* (2004) e Ferrão *et al.* (2004) apresentam aplicações para controle de qualidade de medicamentos. Coury *et al.* (2012) utiliza AG para o problema de alocação de dispositivos indicadores de faltas em alimentadores primários de energia elétrica. Araújo *et al.* (2011) e Araújo *et al.* (2010) apresentam a utilização de *Simulated Annealing* em otimização de processos de secagem. Magalhães *et al.* (2011) utiliza o mesmo método aplicado à gráfico de controle de variáveis. Espinal *et al.* (2010) faz uso de *Simulated*

Annealing em um problema de purificação de etanol, durante o estudo da estrutura molecular da mistura etanol+água, que é obtida através deste método. O *Particle Swarm*, por sua vez, é abordado por Coelho *et al.* (2006) para quantificar falhas estruturais e por Ferrão *et al.* (2008) visando identificar alterações em Blendas de Biodiesel.

Com relação à aplicação dos métodos em Pesquisa Operacional, em geral, os trabalhos para este ramo da Engenharia de Produção dizem respeito à melhorias ou incrementações dos próprios métodos, ou combinações com outros, buscando metodologias eficazes de resolução de problemas.

Do total de cinco trabalhos encontrados em Pesquisa Operacional, quatro utilizaram Algoritmos Genéticos e um utilizou *Particle Swarm*. Borges e Lehrer (2002) apresentaram métodos novos de operadores de seleção utilizada por um AG, considerando o nível fenotípico com o genotípico, ao invés de considerar somente o material genético, além de apresentarem alternativas ao operador de recombinação. Por sua vez, Cosenza *et. al* (2004) traz uma proposta para solução do problema de inconsistência em matrizes decisórias utilizando AG. Tinós (2007) aplica AG à robôs móveis, nos quais a função de aptidão sofre alteração, contemplando o ramo da inteligência computacional. Miranda *et. al.* (2011) utilizam planejamento de experimentos para analisar a significância dos principais parâmetros de um AG clássico na otimização de um modelo de simulação a eventos discretos. Jiménez *et al.* (2002) utilizam *Simulated Annealing* como alternativa de resolução para problemas multiatributo discretos.

Um único trabalho foi encontrado para a área de Engenharia Organizacional, proposto por Moraes e Nagano (2009), que consiste na implementação de um algoritmo genético como apoio à tomada de decisão referente ao saldo ideal de caixa para uma empresa.

4 APLICAÇÃO PRÁTICA DE META-HEURÍSTICA

4.1 Eleição do Problema

Para ilustrar a aplicação de Meta-heurísticas em Engenharia de Produção, o Problema do Caixeiro Viajante (PCV) foi selecionado, devido as seguintes características:

1. grande aplicabilidade para problemas práticos, tais como programação de operações de máquinas em manufatura, programação de transporte entre células de manufatura, otimização do movimento de ferramentas de corte, otimização de perfurações de furos em placas de circuitos impressos, problema de roteamento de veículos, programação e distribuição de tarefas em plantas, trabalhos administrativos (GOLDBARG e LUNA, 2000);
2. diversidade e quantidade de trabalhos publicados que são modelados como PCV's;
3. para as três Meta-heurísticas estudadas, foram encontrados casos de aplicação para a PCV, comprovando sua importância em Engenharia de Produção;
4. pertencente à classe NP-Difícil (MELO, 2006) e, desta forma, conveniente para tratamento com Meta-heurísticas;

Diversas formulações podem ser encontradas para o PCV, conforme consta em Goldberg e Luna (2000). A formulação simples aqui considerada será uma adaptação das propostas de Biscaia Junior (2002) e Zanuz (2007): encontrar um caminho através de um grafo valorado que inicie e termine no mesmo vértice, passe por todos os pontos, volte ao vértice de origem e minimize o custo. A Figura 26 fornece uma ilustração para o problema.

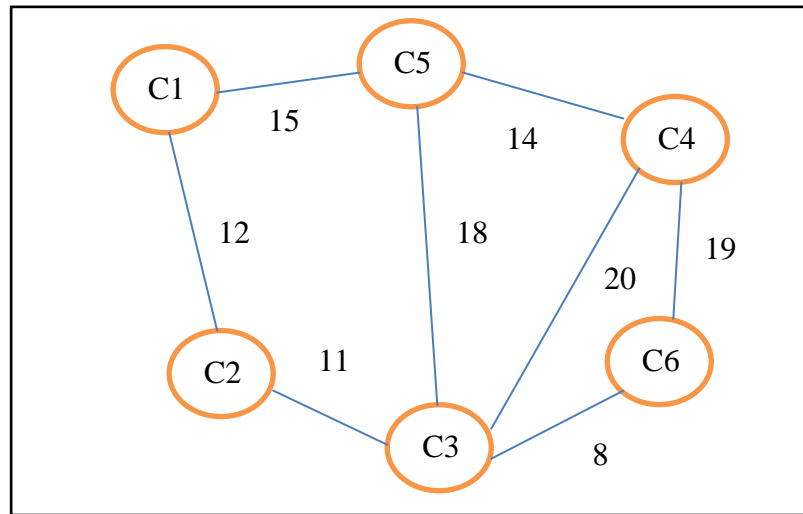


Figura 26 – PCV

(ZANUZ, 2007, Adaptado)

Para n vértices, o número de combinações possíveis, partindo de qualquer ponto não específico, é $R(n) = n!$ Para um número pequeno de vértices, um algoritmo determinístico é suficiente para retornar um resultado ótimo. Todavia, o número de combinações cresce fatorialmente com o aumento do número de vértices. O emprego de Meta-heurísticas é, então, uma alternativa para solução. Ainda que não se tenha garantia de encontrar o valor ótimo, há garantia de localização de um valor aceitável.

4.2 Meta-heurística aplicada ao problema eleito

Faria (2008) implementa um AG para resolução do PCV. O algoritmo proposto pelo autor é de simples implementação e fácil compreensão. Além disso, está disponível na página do autor o código para *download*.

Pelo código de Faria (2008), a partir de uma população de soluções gerada aleatoriamente, a cada evolução essas soluções são ordenadas em *Ranking*. Uma taxa de mortalidade é determinada e sobrevivem para a próxima geração os indivíduos que possuem as melhores soluções, dentro da taxa estabelecida. Os indivíduos sobreviventes sofrem cruzamento e mutação, gerando novos indivíduos para a população. O *Ranking* é organizado novamente e o processo se repete até que um número de evoluções estabelecido seja atingido.

Os principais métodos implementados no algoritmo de Faria (2008) estão descritos a seguir:

```

import java.util.Random;
//Importa a classe java.util.Random para gerar números aleatórios.
public class AlgoritmoGenetico {
//Declara a classe principal.
    public static int NUMERO_CIDADES = ;
    public static int NUMERO_POPULACAO = ;
//Declara o número de vértices e o tamanho da população inicial;
    public static void main(String[] args) {
        boolean mostrarEvolucao = true;
//Mostra as evoluções em tela durante a execução. Uma instrução False faria com que os
valores não fossem mostrados. A importância reside no fato de que a decisão de mostrar as
evoluções aumenta o tempo de processamento. Porém, fornece ao usuário um panorama do
comportamento do algoritmo.
        float taxaMortalidade = (float) ;
//Define a taxa de mortalidade.
        int numeroEvolucoes = ;
//Define o número de evoluções.
        int[][] mapa = { },
//Declara a matriz de adjacência, onde os valores devem ser declarados em linhas e colunas,
separados por vírgula.
        String[] cidades = { "", };
//Declara as variáveis que denominaram os vértices.
        int[][] cromossomos = new int[NUMERO_POPULACAO][NUMERO_CIDADES];
        int[] resultados = new int[NUMERO_POPULACAO];
//Variáveis que serão modeladas por meio do algoritmo.
        gerarCromossomosAleatoriamente(cromossomos);
//Gera as primeiras soluções aleatoriamente
        calcularResultado(cromossomos, resultados, mapa);
//Para as primeiras soluções, calcula o resultado.
        ordenar(cromossomos, resultados);

```

```

//Ordena soluções em Ranking.
    if (mostrarEvolucao)
        imprimir(cromossomos, resultados, cidades);
//Mostra as evoluções em tela.
    int i;
    for (i = 0; i < numeroEvolucoes; i++) {
//Determina quantas vezes será realizado o processo recursivo, conforme o número de
//gerações estabelecido.
        renovarCromossomos(cromossomos, resultados, taxaMortalidade);
//Método que atualiza o Cromossomo.
        calcularResultado(cromossomos, resultados, mapa);
//Calcula o resultado novamente.
        ordenar(cromossomos, resultados);
//Ordena cromossomo novamente.
        if (mostrarEvolucao) {
            System.out.println("Geracao: " + (i + 1));
            imprimir(cromossomos, resultados, cidades);
        }
        resultado(cromossomos, resultados, cidades);
    }
}
// Imprime resultados encontrados.

```

Na solução do problema que o autor disponibiliza em sua página, uma população de 10 soluções é gerada aleatoriamente. A cada evolução, os cromossomos são ordenados, deixando os melhores indivíduos primeiro. Uma taxa de mortalidade de 50% faz com que os 5 melhores permaneçam para a próxima evolução. Os cromossomos excluídos são substituídos pelo cruzamento dos 5 mais aptos. O cruzamento é aleatório entre dois pais também escolhidos aleatoriamente. Os genes são transmitidos aos filhos e os que não puderam ser copiados são substituídos por genes aleatórios, caracterizando mutação. O problema considera 8 cidades, que representam vértices de um grafo, mas pode ser expandido para um número maior ou menor, com alterações dos parâmetros e variáveis declaradas. O código completo está disponibilizado no Anexo A.

A implementação para um caso particular, a partir do código do autor, se dá alterando as variáveis do problema. Uma matriz de adjacência $n \times n$ deve ser fornecida pelo usuário. Cada vértice é representado por uma linha e uma coluna. O cruzamento entre uma linha e uma coluna é a distância de um vértice para outros. Com efeito, a matriz resultante possui diagonal nula, que representa a distância de um vértice para ele mesmo. A declaração da matriz no código fica assim designada:

```
Int [ ] [ ] mapa =      { { 0, 42, 61, 30, 17, 82, 31, 11, 80 },
                        { 42, 0, 14, 87, 28, 70, 19, 33, 67 },
                        { 61, 14, 0, 20, 81, 21, 8, 29, 56 },
                        { 30, 87, 20, 0, 34, 33, 91, 10, 34 },
                        { 17, 28, 81, 34, 0, 41, 34, 82, 57 },
                        { 82, 70, 21, 33, 41, 0, 19, 32, 65 },
                        { 31, 19, 8, 91, 34, 19, 0, 59, 31 },
                        { 11, 33, 29, 10, 82, 32, 59, 0, 43 },
                        { 80, 67, 56, 34, 57, 65, 31, 43, 0 } }
```

Neste caso, uma matriz 9×9 representa um grafo de 9 vértices. Um cromossomo é formado com uma sequência de vértices contemplando a todos sem repetir nenhum. Para execução, é necessário também declarar as variáveis:

```
public static int NUMERO_CIDADES = 9;
public static int NUMERO_POPULACAO = 20;
float taxaMortalidade = (float) 0.5;
String[ ] cidades = { "A", "B", "C", "D", "E", "F", "G", "H", "I" };
```

Neste caso, está definido o número de vértices igual a 9, a população de trabalho igual a 20 e a taxa de mortalidade 50%. Os vértices são nomeados para serem representados em tela. A solução inicial do algoritmo é mostrada a Figura 27.

| | |
|--|-----------------|
| A => H => I => G => C => D => F => B => E => A | Resultados: 261 |
| H => D => E => I => F => C => B => G => A => H | Resultados: 262 |
| B => F => E => D => A => H => C => I => G => B | Resultados: 321 |
| H => I => D => G => B => A => E => F => C => H | Resultados: 337 |
| E => A => F => D => C => I => H => B => G => E | Resultados: 337 |
| C => B => E => F => I => G => D => A => H => C | Resultados: 340 |
| A => C => G => F => B => E => D => H => I => A | Resultados: 353 |
| H => D => F => E => A => C => I => B => G => H | Resultados: 363 |
| D => A => E => F => C => G => H => I => B => D | Resultados: 373 |
| I => G => F => D => A => B => C => E => H => I | Resultados: 375 |
| B => E => I => G => H => C => D => F => A => B | Resultados: 381 |
| G => B => C => A => F => D => H => E => I => G | Resultados: 389 |
| F => H => B => G => I => C => E => D => A => F | Resultados: 398 |
| A => D => I => C => G => H => E => F => B => A | Resultados: 422 |
| I => E => B => F => A => D => G => C => H => I | Resultados: 438 |
| I => B => G => D => F => E => C => A => H => I | Resultados: 447 |
| D => I => E => F => A => B => C => H => G => D | Resultados: 449 |
| E => I => A => F => H => G => C => B => D => E | Resultados: 453 |
| D => B => E => H => G => I => A => C => F => D | Resultados: 482 |
| C => F => H => I => A => B => D => G => E => C | Resultados: 511 |

Indivíduos
mais aptos,
pais da
próxima
geração.

Menos aptos,
serão
substituídos
por filhos
gerados por
meio de
cruzamento e
mutação dos
pais.

Figura 27 – Primeira evolução do AG

Aplicados os operadores de cruzamento e mutação, a geração é montada, preservando os pais e apresentando os novos filhos. O Ranking é estabelecido novamente, conforme Figura 28.

| Geracao: 1 | |
|--|-----------------|
| A => H => I => G => C => D => F => B => E => A | Resultados: 261 |
| H => D => E => I => F => C => B => G => A => H | Resultados: 262 |
| A => H => G => F => C => D => I => B => E => A | Resultados: 276 |
| H => E => I => G => B => C => F => D => A => H | Resultados: 298 |
| A => F => E => B => C => D => H => I => G => A | Resultados: 300 |
| H => I => A => F => G => C => B => E => D => H | Resultados: 318 |
| C => D => F => B => A => E => G => I => H => C | Resultados: 319 |
| B => F => E => D => A => H => C => I => G => B | Resultados: 321 |
| E => A => F => D => C => I => H => B => G => E | Resultados: 337 |
| H => I => D => G => B => A => E => F => C => H | Resultados: 337 |
| C => B => E => F => I => G => D => A => H => C | Resultados: 340 |
| G => I => E => F => B => H => D => A => C => G | Resultados: 341 |
| A => C => G => F => B => E => D => H => I => A | Resultados: 353 |
| H => D => F => E => A => C => I => B => G => H | Resultados: 363 |
| D => A => E => F => C => G => H => I => B => D | Resultados: 373 |
| I => G => F => D => A => B => C => E => H => I | Resultados: 375 |
| A => D => B => C => I => F => E => G => H => A | Resultados: 397 |
| B => A => F => E => D => C => H => G => I => B | Resultados: 405 |
| H => F => G => D => A => E => C => I => B => H | Resultados: 426 |
| A => B => D => E => I => F => H => G => C => A | Resultados: 445 |

Figura 28 – Segunda evolução do AG.

O processo se repete até que o número de evoluções do usuário seja atingido. Essa declaração é também feita pelo usuário:

```
int numeroEvolucoes = 3000
```

Neste caso, foram definidas 3000 evoluções, que será o critério de parada. O resultado final é, então, gerado em tela, juntamente com a última evolução, conforme a Figura 29. Também, é fornecido o tempo de execução.

```
Geracao: 3000
D => I => G => F => C => B => E => A => H => D Resultados: 185
D => H => A => E => B => C => F => G => I => D Resultados: 185
F => C => B => E => A => H => D => I => G => F Resultados: 185
D => H => A => E => F => C => B => G => I => D Resultados: 198
F => H => A => E => B => C => D => I => G => F Resultados: 206
D => C => B => E => A => H => F => G => I => D Resultados: 206
D => H => A => E => B => C => G => F => I => D Resultados: 206
I => H => A => E => B => C => G => F => D => I Resultados: 207
F => H => A => E => B => C => G => I => D => F Resultados: 208
I => H => A => E => B => G => F => C => D => I Resultados: 212
D => H => A => F => E => C => B => G => I => D Resultados: 323
C => H => A => G => E => F => B => I => D => C Resultados: 337
G => F => D => I => E => C => B => A => H => G Resultados: 350
F => C => A => E => B => H => I => G => D => F Resultados: 358
I => H => C => E => B => G => A => F => D => I Resultados: 380
D => A => F => C => B => G => H => E => I => D Resultados: 398
F => I => A => H => E => B => D => C => G => F Resultados: 400
E => I => D => F => A => B => C => G => H => E Resultados: 411
I => C => A => B => E => H => G => F => D => I Resultados: 414
G => H => A => B => E => C => F => I => D => G Resultados: 432
D => I => G => F => C => B => E => A => H => D Resultado: 185
CONSTRUÍDO COM SUCESSO (tempo total: 11 segundos)
```

Figura 29 – Resultado do AG após a última evolução

4.3 Testes de Validação

Para analisar a validade do algoritmo de Faria (2008), foram implementadas matrizes para números pequenos de vértices, de 7 a 12, cujo os melhores resultados eram conhecidos. A taxa de mortalidade foi fixada em 0,5 para cada evolução, e o número de evoluções em 3000. Os resultados obtidos podem ser visualizados na Tabela 7.

Tabela 7 – Execução do AG para 7 a 12 vértices do PCV

| Nº de Vértices | Resposta Ótima | Resposta do AG | Desvio (%) | Tempo de Execução (S) | Tamanho da População |
|----------------|----------------|----------------|------------|-----------------------|----------------------|
| 7 | 70 | 70 | 0% | T < 1 | 20 |
| 8 | 80 | 80 | 0% | T < 1 | 20 |
| 9 | 90 | 90 | 0% | T < 1 | 20 |
| 10 | 100 | 100 | 0% | T < 1 | 20 |
| 11 | 110 | 110 | 0% | 4 | 250 |
| 12 | 120 | 120 | 0% | 9 | 400 |

O tempo de execução apresentado na Tabela 7 foi otimizado com a omissão da apresentação dos resultados parciais em tela. T < 1 representa um tempo inferior à 1 segundo de execução. O tamanho da população, inicialmente fixado em 20 cromossomos, não foi suficiente para convergir para a resposta para os dois últimos casos. Com o aumento da complexidade do problema, a resposta ótima foi obtida com o incremento de indivíduos à população de trabalho.

O código foi rodado no ambiente de desenvolvimento NetBeans IDE 7.2, da Oracle Corporation, em computador equipado com processador Intel® Core™ I5-2450M, 2,50 GHz e 4 GB de memória. O sistema operacional utilizado foi o Windows 7, da Microsoft Corporation.

4.4 Resultados e Discussão

O Algoritmo de Faria (2008) retornou a solução ótima conhecida para todos os casos estudados. Foi necessária a alteração do parâmetro Tamanho da População para os dois últimos casos, com o aumento da complexidade do Algoritmo. Isso possibilitou que novas soluções pudessem ser exploradas, uma vez que o aumento da população expande as possibilidades de resposta pesquisadas pelo AG. Houve, porém, aumento no tempo computacional de execução.

Outro parâmetro que pode fornecer resultados mais robustos é a alteração do número de evoluções, que pode ser necessário para números maiores de vértices na matriz de adjacência. Uma combinação destes dois parâmetros pode levar a melhores resultados com o crescimento do número de vértices, atentando para o tempo computacional de execução.

Outra alternativa interessante de validação do algoritmo seria a implementação de números maiores de vértices, comparando o tempo computacional com algoritmos exatos, ou com trabalhos que implementem outras metodologias para o PCV.

5 CONCLUSÃO

Este trabalho apresentou a aplicação de Meta-heurísticas em Engenharia de Produção. O estudo dos métodos apontou que sua utilização é indicada em problemas complexos, tais como NP-Completo, e que estes frequentemente aparecem em diversas áreas da Engenharia. Desta forma, o interesse no estudo destes métodos reside no fato de representarem soluções robustas, em tempo computacional viável, num contexto competitivo em que a busca por ferramentas de otimização por parte de organizações e pesquisadores, representa minimização de custos de processos e melhor aproveitamento de recursos.

As publicações mencionadas demonstraram a aplicabilidade dos métodos Algoritmos Genéticos, *Simulated Annealing* e *Particle Swarm* para as áreas de Engenharia de Operações e Processos de Produção, Logística, Engenharia da Qualidade, Pesquisa Operacional e Engenharia Organizacional. As duas primeiras, congregam a maior parte das publicações. Isto vai de encontro com a proposta de encontrar boas soluções como alternativas para problemas que envolvam grandes quantidades de possibilidades de resposta.

Foi observada uma tendência para o crescimento do número de publicações que utilizam Meta-heurísticas como alternativa de solução. Há, também, um grande número de publicações nos dois maiores eventos de Engenharia de Produção do Brasil (SIMPEP e ENEGEP), o que demonstra a dedicação dos pesquisadores em desenvolver soluções com estas metodologias.

O problema eleito demonstrou o funcionamento de um Algoritmo Genético para um caso simples do Problema do Caixeiro Viajante. O PCV é apresentado na literatura por diversos autores e os Algoritmos Genéticos foram as Meta-heurísticas que mais figuraram nas publicações, dado a sua tradição. O algoritmo de Faria (2008) retornou, em todos os casos, a resposta ótima em curto período de tempo. Foi verificado um aumento do tempo de processamento quando foi necessário alterar o tamanho da população, fato que possibilitou ao algoritmo pesquisar respostas que levaram a otimização do resultado.

Trabalhos futuros poderiam estudar a aplicação de outras Meta-heurísticas em Engenharia de Produção, bem como a hibridização de métodos, que frequentemente têm surgido também como alternativas eficazes de solução.

As dificuldades encontradas foram quanto ao estudo das linguagens de programação utilizadas, pois nem sempre esta informação está presente nos textos publicados. As implementações em código, em geral, não figuram nos trabalhos, ficando restrito apenas à apresentação dos resultados. Isso é justificado pelo valor que possui esta codificação. Livros específicos que abordem o tema de Meta-heurística isoladamente não foram encontrados, ficando restrito à capítulos que tratam do tema.

REFERÊNCIAS

ABENSUR, E.; OLIVEIRA, R.. Um método heurístico construtivo para o problema da grade horária escolar. **PODes**, v 4, n.2, p 230-248, mai-ago 2012.

AGUILAR, S.; DUTRA, V.; FERREIRA, G.. Metaheurística simulated annealing e Iterated Local Search aplicadas ao problema de p-medianas. **Anais. SIMPEP**, 2007.

AHONEN, H.; ALVARENGA, A.; LORENZONI, L.. Um algoritmo híbrido baseado em colônia de formigas e recozimento simulado para problemas de escalonamento com restrição de recursos e múltiplos modos de processamento.

ALMEIDA, F.; GUIMARÃES, E.; SOUZA, F.; VIANNA, D.. Algoritmos genéticos híbridos aplicados ao problema de árvores mínima com grupamentos. **Anais. SIMPEP**, 2009.

ALMEIDA, K.; VOLPI, N.. Algoritmo genético multicritério para alocação de dispositivos de controle em uma rede de transmissão elétrica. **Anais. ENEGEP**, 2008.

ALMEIDA, L.; PACHECO, M.; VELLASCO, M.. Algoritmos evolucionários na otimização de alternativas para o desenvolvimento de campos de petróleo. **Anais. ENEGEP**, 2003.

ALMEIDA, W. Métodos Heurísticos para o Problema de Localização de Concentradores. **Dissertação** (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais. 2009.

ALOISE, D.; OLIVEIRA, M.; SILVA, T.. Otimização discreta por nuvem de partículas aplicada ao problema do caixeiro viajante. **GEPROS**, ano 1, N.2, Abr/2006, p 87-95.

ALVES, J.; SILVA, G.; SOUZA, M.. Simulated annealing aplicado à programação da tripulação no sistema de transporte público. **Anais. ENEGEP**, 2002.

ALVES, W.; ARAÚJO, S.; LIBRANTZ, A.. Técnicas metaheurísticas aplicadas na otimização de parâmetros em um modelo probabilístico de gestão de estoques. **Anais. SIMPEP**, 2009.

ARAÚJO, S.; BENVENGA, M.; LIBRANTZ, A.; SANTANA, J.; TAMBOURGI, E.. Aplicação do simulated annealing na simulação e otimização do processo de secagem do malte de zea mays. **Eng. Agric. Jaboticabal**, v 31, n.5, p 940-953, set-out 2011.

ARAÚJO, S.; BEVENGA, M.; LIBRANTZ, A.; SANTANA, J.. Cinética e otimização do processo de secagem do malte de milho usando simulated annealing. **Anais**. ENEGEP, 2010.

ARAÚJO, S.; LIBRANTZ, A.. Algoritmos genéticos aplicados na simulação do processo de produção do malte de milho. **Anais**. ENEGEP, 2009.

ARENALES, M.; ARMENTO, V.; MORABITO, R.; YANASSE, H.. **Pesquisa Operacional – Para cursos de Engenharia**. Rio de Janeiro : Elsevier, 2007.

ARGOUD, A.; GONÇALVES FILHO, E.; GORGULHO JR., J.. Algoritmos genéticos para formação de células de fabricação. **Anais**. ENEGEP, 2004.

ARGOUD, A.; GONÇALVES FILHO, E.; TIBERTI, A.. Algoritmo genético de agrupamento para formação de módulos de arranjo físico. *GESTÃO E PRODUÇÃO*, v 15, n.2, p 393-405, mai-ago 2008.

ARICA, J.; IAMASHITA, E.; GALAXE, F.; *A planning model for offshore natural gas transmission*. **Pesquisa Operacional**, v 28, n.1, p 29-44, jan-abr 2008.

ARROYO, J.; CORTEZ, J.; MARQUES, T.. Um algoritmo genético para o problema de alocação de antenas de transmissão. **Anais**. ENEGEP, 2006.

ARROYO, J.; PEREIRA, A.. Metaheurística para problema de flowshop flexível com penalidades de adiantamento e atraso. **Anais**. SIMPEP, 2011.

AZEVEDO, A.; DEUS, N.; RIBEIRO, C.; SALLES NETO, L.. Resolução do problema de carregamento e descarregamento de contêineres num navio para múltiplos cenários via representação por regras e algoritmo genético. **Anais**. SIMPEP, 2011.

AZEVEDO, A.; RIBEIRO, C.; SALLES NETO, L.; SILVA, M.; SILVESTRE, M.. Resolução do problema de carregamento e descarregamento 3D de contêineres em terminais portuários, via representação por regras e algoritmo genético. **GEPROS**, Ano6, N.4, Out-Dez/2011, p 91-110.

AZEVEDO, A.; SILVA, M.; SILVESTRE, M.; SALLES N.; SILVESTRE, M.. Resolução do problema de carregamento e descarregamento 3D de contêineres em terminais portuários via representação por regras e algoritmos genéticos. **Anais**. SIMPEP, 2011.

BAPTISTA, E.; GOMES, M.; SAMED, M.. Uma abordagem eficiente para resolver os problemas de despacho econômico e ambiental. **Anais. SIMPEP**, 2003.

BARAKATI, S.; BANOOKH, A.. Optimal Design of Double Folded Stub Microstrip Filter by Neural Network Modelling and Particle Swarm Optimization. **Jornal of Microwaves, Optoelectronics and Electromagnetic Applications**, v 11, n.1, Jun 2012.

BARBOZA, A.; BORTOLOTTI, S.; NEVES JR., F.; SOUZA, R.. Algoritmo genético e PLIM aplicados ao problema de transferência e estocagem de produtos em uma indústria petrolífera. **Anais. SIMPEP**, 2008.

BARROS JR., A.; PEREIRA, G.. Aplicação de metaheurística algoritmo genético com Path relinking para a solução de um problema de planejamento florestal. **Anais. SIMPEP**, 2011.

BARROS, A.; MOCCELLIN, J.. Análise da flutuação do gargalo em Flowshop permutacional com tempos de setup assimétricos e dependentes da sequência. **GESTÃO E PRODUÇÃO**, v 11, n.1, p 101-108, jan-abr 2004.

BARROS, V.; COSTA, T.; LORENA, L.; OLIVEIRA, A..Problema de alocação de berços heterogêneos em portos graneleiros com restrições de estoques e condições favoráveis de maré. **Anais. SIMPEP**, 2009.

BECCENERI, J.. Meta-heurística e Otimização Combinatória: Aplicação em Problemas Ambientais. **Minicurso. INPE**, 2012. Disponível em <http://www.lac.inpe.br/ELAC/arquivos/MiniCurso_02ELAC2012.pdf>, acesso em 11 de agosto de 2012.

BECCENERI, J.; SILVA NETO, A.; SOEIRO, F.. **Recozimento Simulado. INPE**, 2010. Disponível em <<http://mtc-m19.sid.inpe.br/col/sid.inpe.br/mtc-m19%4080/2010/01.20.19.25/doc/cap5.pdf>>, acesso em 11 de agosto de 2012.

BEHREND, S.; BITTENCOURT, E.; BITENCOURT, C.; LANDMANN, R.; WYREBSKI, J.. Contribuição da inteligência artificial para a viabilização da estratégia de manufatura: um estudo de caso na indústria de fundição. **Anais. SIMPEP**, 2008.

BELFIORE, P.; CAMPOS, G.; YOSHIKAZI, H.. Algoritmos genéticos e computação paralela para problemas de roteirização de veículos com janelas de tempo e entregas fracionadas. **GESTÃO E PRODUÇÃO**, v 13, n.2, p271-281, mai-ago 2006.

BENTO, E.; KAGAN, N.. Algoritmos genéticos e variantes na solução de problemas de configuração de redes de distribuição. **Revista Controle & Automação**, v19, n.3, jul-set 2008.

BISCAIA JUNIOR, E. Otimização Heurística. **Notas de aula**. COPPE/UFRJ, 2002. Disponível em <http://www2.peq.coppe.ufrj.br/Pessoal/Professores/Evaristo/OTIMIZACAO%20HEURISTICA/>, acesso em 11 de agosto de 2012.

BITTENCOUT, F.; DUARTE, W.; EPAMINONDAS, L.. Uso do algoritmo genético para a solução de problemas de sequenciamento da produção. **Anais. ENEGEP**, 2007.

BORGES, C.; FERREIRA, C.. Solução do problema de superação de disjuntores pela alocação ótima de dispositivos limitadores de corrente de curto-circuito utilizando algoritmos genéticos. **Revista Controle & Automação**, v 18, n.4, out-dez 2007.

BORGES, P.; LEHRER, C.. Algoritmos genéticos com operação de seleção hawk-dove com estratégia TIT for TAT, e com operação de recombinação com taxas variáveis. **Anais. ENEGEP**, 2002.

BRAGA, A.; CARVALHO, A.; LUDEMIR, T.; Computação Evolutiva. *In* Computação Evolutiva – Fundamentos e Aplicações. Organização de Solange OliveiraRezende. – Barueri, SP: Manole, 2005.

BRANCO, R.; RODRIGES, G.; ROSA, L.. Tecnologia de grupo: algoritmos genéticos na solução de problemas de agrupamentos celulares de máquinas e peças. **Anais. ENEGEP**, 2003.

CAIXETA FILHO, J.. Pesquisa Operacional: Técnicas de Otimização Aplicadas a Sistemas Agroindustriais. 2 ed, Atlas, 2004.

CANDIDO, M.; SAMPAIO, R.; SILVA, K.. Uma abordagem híbrida para o problema de formação de grupos e balanceamento de linhas de montagem SMT. **Anais. ENEGEP**, 2002.

CARNEIRO, A.; CARVALHO, A.; LEITE, P.. Aplicação de algoritmos genéticos da determinação da operação ótima de sistemas hidrotérmicos de potência. **Revista Controle & Automação**, v 17, n.1, jan-mar, 2006.

CARNEIRO, A.; SILVA FILHO, D.. Dimensionamento Evolutivo de Usinas Hidroelétricas. **Revista Controle & Automação**, v 15, n.4, out-dez 2004.

CARNIERI, C.; ROSÁRIO, R.; STEINER, M..Proposta de solução para o problema das p-medianas na localização de unidades de saúde 24 horas. **Anais. ENEGEP**, 2002.

CARNIERI, C.; SMIDERLE, A.; STEINER, M.. Problema de cobertura de arcos – um estudo de caso. **Anais. ENEGEP**, 2003.

CARVALHO, C.; KOZEN, P.; FURTADO, J.; FERRÃO, M.; MOLZ, R.; BASSANI, I.; HUNING, S.. Otimização de métodos de controle de qualidade de fármacos usando algoritmos genéticos e busca tabu. **Pesquisa Operacional**, v 23, n.1, p 189-207, jan-abr 2003.

CARVALHO, E.; GUEDES, L.. Aplicação de inteligência computacional ao problema logístico de escoamento contigencial de tráfego aéreo. **Anais. SIMPEP**, 2007.

CARVALHO, R.; GOMES, B.; MARTINS, A.; PEDRO, O.. Algoritmo genético aplicado ao problema de localização de hubs com alocação simples sem restrição de capacidade. **Anais. SIMPEP**, 2010.

CARVALHO, S.; FIGUEIREDO, A.; LORENA, L.. Modelo TEAM de localização de ambulâncias – solução através do algoritmo genético construtivo. **Anais. ENEGEP**, 2005.

CASTRO, O.; MARTINS, A.; SOUZA, M.. Um método híbrido para a resolução do problema de roteamento de veículos. **Anais. ENEGEP**, 2004.

CERA, M.; MULLER, F.; SCHOPF, E.; TURCHETTI, R.; VAZ, C.; VIEIRA, L.. Metaheurísticas aplicadas na solução do problema do caixeiro viajante com demandas heterogêneas. **Anais. ENEGEP**, 2004.

CHAVES, A.. Modelagens Exata e Heurística para Resolução do Problema do Caixeiro Viajante com Coleta de Prêmios. **Monografia**. Departamento de Ciência da Computação, Universidade Federal de Ouro Preto-MG, 2003. Disponível em <

<http://www.decom.ufop.br/prof/marcone/Orientacoes/PCVCP-Exato-VNS.pdf>>, acesso em 11 de agosto de 2012.

CHAVES, D.; FIGUEIREDO, E.; BASTOS-FILHO, C.; MARTINS-FILHO, J.; SEGATTO, M.. Design of Distributed Optical-Fiber Raman Amplifiers using Multi-objective Particle Swarm Optimization. **Jornal of Microwaves, Optoelectronics and Electromagnetic Applications**, v 10, n.2, dec 2011.

CHIWIACOWSKY, L.; MONTEIRO, R.. Uma aplicação da meta-heurística algoritmo genético ao problema de carregamento de navios-contêineres. **PODes**, v 4, p 52-64, jan-abri 2012.

COELHO, L.; LOPES JUNIOR, V.; TEBALDI, A.. Detecção de falhas em Estruturas inteligentes usando otimização por nuvem de partículas: fundamentos e estudos de casos. **Revista Controle & Automação**, v 17, n.3, jul-set 2006.

COELHO, L.; MARIANI, V.. Algoritmo evolutivo híbrido: uma heurística promissora para o problema de localização de facilidade. **Anais. ENEGEP**, 2003.

COLIN, E.. Pesquisa Operacional: 170 **Aplicações em estratégia, finanças, logística, produção, marketing e vendas**. Rio de Janeiro: LTC, 2007. ISBN 978-85-216-1559-0

CONSELHEIRO, F. Estudo da Aplicação da Estratégia de Simulated Annealing aos Problemas de Programação da Produção em Unidades Batelada. **Dissertação** (Mestrado em Engenharia Química) – Universidade Estadual de Campinas. 1999.

CORDEIRO, M.; SHIMODA, E.; VIANNA, D.. Desenvolvimento de algoritmos genéticos para o problema das p-medianas utilizando operadores genéticos de cruzamento convencionais e não-convencionais. **Anais. SIMPEP**, 2009.

CORDENONSI, A.A. Ambientes, Objetos e Dialogicidade: Uma Estratégia de Ensino Superior em Heurística e Metaheurística. **Tese de Doutorado**: Programa de Pós-Graduação em Informática na Educação – UFRGS, 2008. Disponível em <http://www-usr.inf.ufsm.br/~andrezc/ia/heuristicas_construtivas.pdf>, acesso em 11 de agosto de 2012.

CORMEN, H.; LEISERSON, C.; RIVEST, R.; STEIN.; **Algoritmos – Teoria e Prática**. Tradução da 2ª ed. Vanderberg D. de Souza. – Rio de Janeiro: Elsevier, 2002 – 8ª reimpressão. ISBN: 85-352-0926-3.

CORREA, F.; LORENA, L..Aplicação da relaxação lagrangeana e do algoritmo genético construtivo na solução do problema probabilístico de localização-alocação de máxima cobertura. **GESTÃO E PRODUÇÃO**, v 13, n.2, p233-244, mai-ago 2006.

CORTES, J.; PAULA JR., G.. Uma abordagem para resolução do problema de localização de atividades econômicas. **Anais. ENEGEP**, 2005.

CORTES, M.. Algoritmo Genético em problemas de Programação não linear contínua. **Tese**. Florianópolis, SC, 1996.

COSTA, A.; BIAZI, E.; VITOR, J. Aplicação da Metaheurística PSO na Identificação de Pontos Influentes por meio da Função de Sensibilidade de Casos. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL. 2009, Cuiabá-MT. **Anais**. v.2 586 p. Disponível em <http://sbmac.locaweb.com.br/eventos/cnmac/xxxii_cnmac/pdf/320.pdf acesso> em 01/04/2012. Acesso em: 30 mar. 2012.

COSTA, F.; GUIMARÃES, I.; SOUZA, M..Um algoritmo evolutivo híbrido para o problema de programação de horários em escolas. **Anais. ENEGEP**, 2002.

COSTA, J.; COSENZA, C.; WANDERLEY, A.. Utilização de algoritmos genéticos em metodologia multicritério: uma solução para inconsistência matricial. **Anais. ENEGEP**, 2004.

COURY, D.; FLAUZINO, R.; SILVA, I.; USIDA, W.. Alocação eficiente de indicadores de faltas em um sistema de distribuição real usando computação evolutiva. **Revista Controle & Automação**, v 23, n.3, mai-jun 2012.

COURY, D.; OLESKOVICZ, M.; SOUZA, S.. Genetic algorithms applied to a faster distance protection of transmission lines. **Revista Controle & Automação**, v 22, n.4, jul-ago 2011.

CROSSETTI, G.; FURTADO, J.; MULLER, V.; NEIS, J.. Otimização do problema de layout de facilidades através da técnica enxame de partículas utilizando a modelagem attractor-repeller. **Anais. ENEGEP**, 2006.

CUNHA, C.; BONASSER, U.; ABRAHÃO, F. Experimentos Computacionais com Heurísticas de Melhoria para o Problema do Caixeiro Viajante. XVI Congresso da Anpet – Associação Nacional de Pesquisa e Ensino em Transportes. 2002, Natal-RN. **Anais**. Disponível em: <http://sites.poli.usp.br/ptr/ptr/docentes/cbcunha/files/2-opt_TSP_Anpet_2002_CBC.pdf> Acesso em: 01 abr. 2012.

DAMASO, V.; GARCIA, P.. *Testing and preventive maintenance scheduling optimization for aging systems modeled by generalized renewal process*. **PESQUISA OPERACIONAL**, v 29, n.3, p 563-576, set-dez 2009.

DERIZ, A.; KATO, E.; MORANDIN JR., O.; SANCHES, D.. Uma modelagem para programação da produção de sistemas de manufatura com recursos compartilhados utilizando algoritmos genéticos. **Anais. ENEGEP**, 2007.

DROGUETT, E.; LINS, I. *Multiobjective optimization availability and cost in repairable systems design via genetic algorithms and discrete event simulation*. **PESQUISA OPERACIONAL**, v 29, p 43-66, jan-abri 2009.

ESPINAL, J.; MEJÍAN, S.; MONDRAGÓN, F.; ORREGO, J.. Heteropentámeros (Etonol)₄-Água: Estudo Estrutural Y Termodinámico. **QUÍMICA NOVA**, v 33, N.4, 860-866, 2010.

FARIA, T.. Meta-heurística – **Algoritmo genético na resolução do Problema do Caixeiro Viajante**. 2008. Disponível em <<http://www.tuliofaria.net/metaheuristica-algoritmo-genetico-na-resolucao-do-problema-do-caixeiro-viajante/>>, acesso em 11 de agosto de 2012.

FARIA, T.; NOMELINI, Q.; SILVA, H.. Eficiência dos métodos de otimização simulated annealing delineação rápida em cadeia e ramos e conexões para construção de mapas genéticos. **Ciênc. Agrotec., Lavras**, v 44, n.6, p 1534-1537, nov-dez 2009.

FERRÃO, M.; FURTADO, J.; GERBASE, A.; RIZZI, M.; SAMIOS, D.. Otimização por enxame de partículas de modelos de regressão visando a identificação de adulterações nas blendas biodiesel/diesel. **Anais. ENEGEP**, 2008.

FERRÃO, M.; FURTADO, J.; NEUMANN,L.; ROSA, T.; SOUZA, J.. Aplicação do algoritmo genético na análise de medicamentos anti-inflamatórios industrializados. **Anais. ENEGEP**, 2004.

FERRÃO, M.; FURTADO, J.; NEUMANN,L.; ROSA, T.; SOUZA, J.. Otimização combinatória empregando algoritmo genético aplicada na análise multivariada de medicamentos manipulados. **Anais. ENEGEP**, 2004.

FLEURY, A.. Introdução à Engenharia de Produção. Organização: Mário Batalha, Ed. Campus, 2007.

FRANCESCHETTE, C..Uma abordagem heurística para o problema de corte guilhotinado bidimensional aplicado a uma situação de canibalização. **Anais. SIMPEP**, 2009.

FREITAS, H.; MOSCAROLA, J. Da observação à decisão: métodos de pesquisa e de análise quantitativa e qualitativa de dados. **RAE Eletrônica** – FGV – EAESP – São Paulo, v 1, n 1, jan-jun 2002. Disponível em: <<http://www.scielo.br/pdf/raeel/v1n1/v1n1a06.pdf>>. Acesso em: 01 abr. 2002.

GALHARDO, M.; MONTEVECHI, J.; SILVA, W.. Otimização de uma célula de manufatura utilizando simulação computacional. **Anais. ENEGEP**, 2004.

GASPAR JR. O.; SOARES, M.; VIEIRA, G.. Otimização do planejamento mestre da produção através de algoritmos genéticos. **Anais. ENEGEP**, 2002.

GOLBARG, M.; LUNA, H.; **Otimização Combinatória e Programação Linear – Modelos e Algoritmos**. Rio de Janeiro: Elsevier, 2000, 7ª reimpressão. ISBN: 85-352-0541-1.

GOLDBARG, M.; GOLDBARG, E.; MEDEIROS NETO, F.. Algoritmo evolucionário na determinação da configuração de custo mínimo de sistemas de co-geração de energia com base no gás natural. **PESQUISA OPERACIONAL**, v 25, n.2, p 231-259, mai-ago 2005.

GOLFETO, R.; MORETTI, A.; NETO, L..Algoritmo genético simbiótico para minimizar o número de objetos processados e o setup dado o número máximo de pilhas abertas num problema de corte de estoque. **PODes**, v 2, n.3, p 203-215, set-dez 2010.

GOLFETO, R.; MORETTI, A.; SALLES NETO, L.. *A Genetic algorithm applied to the one-dimensional cutting stock problem*. **PESQUISA OPERACIONAL**, v 29, n.2, p 365-382, mai-ago 2009.

GOMES JR.; MARTINS, A.; SOUZA, M.. Um algoritmo simulated annealing eficiente para o problema de roteamento de veículos com janela de tempo. **Anais. ENEGEP**, 2005.

GOMES, D.; OLIVEIRA, A.; SOUZA, M..Um algoritmo baseado em iterated local search e simulated annealing para a resolução de um problema de corte bidimensional com rotação de itens. **Anais. SIMPEP**, 2009.

GUIMARÃES, F.; RAMALHO, M.. Implementação de um Algoritmo Genético. **Trabalho Acadêmico**. Departamento de Engenharia Elétrica – UFMG. 2001. Disponível em <<http://www.cpdee.ufmg.br/~lusoba/disciplinas/ele037/t1.pdf>>, acesso em 11 de agosto de 2012.

HAESER, G.; GOMES–RUGGIERO, M.. Aspectos Teóricos de Simulated Annealing e um Algoritmo duas Fases em Otimização Global. **Departamento de Matemática Aplicada, IMECC, UNICAMP, Campinas, SP, Brasil..** Disponível em: <http://www.sbmec.org.br/tema/seletas/docs/v9_3/Hae_Gom.pdf>. Acesso em 30 mar. 2012.

HANNA, M.; RENDER, B.; STAIR, R.. *Análise Quantitativa para administração*. Porto Alegre : Bookman, 2010. ISBN 978-85-7780-632-4

HASSELMANN, F.; ZHANG, Y.; RAMIREZ, L.; Channel Characteristics in Tunnels FDTD Simulations and Measurement. **Journal of Microwaves, Optoelectronics and Electromagnetic Applications**, v 10, n.1, Jun 2011.

ILHA, V.; PASSOS, C..Uma abordagem multiagente para a solução de problemas do tipo jobshop. **Anais. ENEGEP**, 2009.

KRIPKA, M.. Discrete Optimization of Trusses by Simulated Annealing. **J. of the Braz. Soc. Of Mech. Sci. & Eng.** V XXVI, n.2, apri-jun 2004.

KRIPKA, M.; PRAVIA, Z.. Otimização numérica e experimental de perfis U formados a frio sujeitos à compressão. **REM: R. Esc. Minas**, Ouro Preto, jul-set 2008.

LACHTERMACHER, G. **Pesquisa Operacional na Tomada de Decisões: modelagem em Excel**. 2. Ed. – Rio de Janeiro: Elsevier, 2004. ISBN 85-352-1492-5.

LANNONI, A.; MORABITO, R.. Modelo hipercubo integrado a um algoritmo genético para análise de sistemas médicos emergenciais em rodovias. **GESTÃO E PRODUÇÃO**, v 13, n.1, p 93-104, jan-abr, 2006.

LAPORTE, L.; MARTINS, T.; SEIXAS FILHO, C.; MARTINS, T..Algoritmos genéticos aplicados à programação da produção de um processo de fundição de zinco. **Anais. ENEGEP**, 2003.

LAUGENI, F.; MARTINS, P.. **Administração da Produção**. 2 Ed. – São Paulo : Saraiva, 2005. ISBN 85-02-04616-0.

LEAL, G.. Extensão de um Algoritmo Cultural para problemas de despacho de Energia Elétrica. **Trabalho de Conclusão de Curso**. DEP-UEM, 2007.

LEITE, H.; RIBEIRO, C.; RODRIGUES, F.; SANTOS, H.; SOUZA, A.. Meta-heurística simulated annealing para solução de problemas de planejamento florestal com restrições de integridade. **R. Árvore**, Viçosa-MG, v 28, n.2, p 247-256, 2004.

LOBATO, F.; SILVA NETO, A.; STEFFEN JR., V.. A comparative study of the application of differential evolution and simulated annealing in radiative transfer problems. **J. of the Braz. Soc. Of Mech. Sci. & Eng.** V XXXII, n.5, 2010.

LORENA, L.; MAURI, G.. Uma nova abordagem para o problema dial-a-ride. **PRODUÇÃO**, v 19, n.1, jan-abr 2009.

LORENA, L.; NAGANO, M.; RIBEIRO FILHO, G.. Heurística evolutiva para a redução do estoque em processamento em sistemas de produção flow shop permutacional. **Anais. ENEGEP**, 2007.

LUGER, G. **Inteligência Artificial – Estruturas e Estratégias para a Resolução de Problemas Complexos**; trad. Paulo Engel. – 4. Ed. – Porto Alegre: Bookmann, 2004. ISBN 85-363-0396-4.

LUZ, E.. Estimação de Fonte de Poluição Atmosférica usando Otimização por Enxame de Partículas. Dissertação. Pós-graduação em computação aplicada – INPE, 2008. Disponível em <http://mtc-m17.sid.inpe.br/col/sid.inpe.br/mtc-17@80/2007/05.07.17.22/doc/publicacao.pdf>>, acesso em 11 de agosto de 2012.

MAGALHÃES, M.; MOURA NETO, F.. Economic-statistical design of variable parameters non-central chi-square control chart. **PRODUÇÃO**, v 21, n.2, p 259-270, abr-jun 2011.

MAPA, S.; SILVA, G.; SOUZA, M.; RODRIGUES, M.. Um estudo das heurísticas simulated annealing e VNS aplicadas ao problema de programação de tripulações. **Anais. ENEGEP**, 2003.

MARCONDES FILHO, W. Desenvolvimento e aplicação de algoritmos heurísticos ao problema de alocação de espaço físico em universidade. **Dissertação** (Mestrado em Ciência da Computação) – Universidade Estadual de Maringá. 2008.

MARKENZON, L.; SZWARCFITER, J.. Estruturas de Dados e seus Algoritmos. Editora LTC, 1994.

MAURI, G.; OLIVEIRA, A.; LORENA, L.. Heurística baseada no Simulated Annealing aplicada ao problema de alocação de berços. **GEPROS**, Ano3, N.1, jan-mar/08, p 113-127.

MAYERLE, S.; SANTOS, R.. Algoritmo genético para o balanceamento de linhas de produção. **Anais. ENEGEP**, 2003.

MELO, E.. Implementação de um Algoritmo Genético para uma Aplicação do Problema do Caixeiro Viajante. Trabalho de Conclusão de Curso, DEP-UEM, 2006. Disponível em <<http://www.dep.uem.br/tcc/arquivos/TCC-EP-33-06.pdf>>, acesso em 11 de agosto de 2012.

MIRANDA, R.; MONTEVECHI, J.; PINHO, A.. Utilização do planejamento de experimentos para análise de significância dos parâmetros de um algoritmo genético. **Anais. SIMPEP**, 2011.

MIRROR, D.. Revoada de Pássaros – Fotografia. In **Revista Globo Rural**. Disponível em <<http://revistagloborural.globo.com/GloboRural/0,6993,EEC1692378-1641,00.html>> acesso em 10 de agosto de 2012.

MORABITO, R., “Pesquisa operacional”, em: **Introdução à Engenharia de Produção**, M. Batalha (ed), Editora Campus/Elsevier, ISBN 978-85-352-2330-4, Rio de Janeiro, 157-182, 2008.

MORAES, M.; NAGANO, M.. Otimização do investimento em saldo de caixa: um complemento ao modelo de Miller e Orr. **Anais**. ENEGEP, 2009.

MORAES, M.; NAGANO, M.; Otimização do saldo de caixa: uma comparação entre algoritmos genéticos e particle swarm optimization. **Anais**. ENEGEP, 2010.

MULLER, V.. Otimização de Layouts Industriais através do Método Enxame de Partículas. **Dissertação**. Pós-graduação em Sistemas e Processos Industriais, Universidade de Santa Cruz do Sul, 2007. Disponível em <<http://btd.unisc.br/Dissertacoes/VivianeM.pdf>>, acesso em 11 de agosto de 2012.

OLIVEIRA, A.; SILVA, A.. Algoritmos genéticos: alguns experimentos com os operadores de cruzamento para o problema do caixeiro viajante assimétrico. **Anais**. ENEGEP, 2006.

OLIVEIRA, G.. Estudo e Aplicação da Evolução Diferencial. **Dissertação**. Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2006. Disponível em <http://www.bdtf.ufu.br/tde_arquivos/1/TDE-2007-05-03T065313Z-545/Publico/GTSOliveiraDISSPRT.pdf>, acesso em 11 de agosto de 2012.

PACHECO, M.; TÚPAC, Y.; VELLASCO, M..Planejamento e otimização do desenvolvimento de um campo de petróleo por algoritmos genéticos. **Anais**. ENEGEP, 2002.

PITOMBEIRA NETO, A.. Procedimento para a geração de arranjos físicos distribuídos baseado em otimização da simulação e algoritmos genéticos. **Anais**. ENEGEP, 2007.

PRADO, J.; SARAMAGO, S.; Otimização por Colônia de Partículas. **Revista Científica Eletrônica da Faculdade de Matemática – FAMAT**, Universidade Federal de Uberlândia – UFU – MG, Uberlândia-MG, N 4, abril de 2005. Disponível em <http://www.portal.famat.ufu.br/sites/famat.ufu.br/files/Anexos/Bookpage/Famat_Revista_04.pdf#page=87>. Acesso em: 31 mar. 2012.

PRATA, B.. Um algoritmo enxame de partículas para uma variante do problema de máxima cobertura. **GEPROS**, Ano 7, N.2, abr-jun/2012, p 139-148.

REZENDE, S.; PUGLIESI, J.; VAREJAO, F. Sistemas Baseados em Conhecimento. In: **Sistemas Inteligentes: fundamentos e aplicações**; organização Solange Oliveira Rezende. – Barueri, SP: Manole, 2005. Cap 9, p 225-248. ISBN 85-204-1683-7.

RIBEIRO, J.. Programação da Produção em uma indústria de Tintas. **Anais. SIMPEP**, 2003.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**; trad. PubliCare Consultoria. – Rio de Janeiro: Elsevier, 2004. ISBN 85-352-1177-2.

SANTOS, P.; SILVA, A.. Integração do Bin Packing Problem com o problema de roteamento de veículos utilizando um algoritmo genético. **Anais. SIMPEP**, 2011.

SCHWAAB, M.; Avaliação de Algoritmos Heurísticos de Otimização em Problemas de Estimção de Parâmetro. **Dissertação** (Mestrado em Engenharia Química) – Universidade Federal do Rio de Janeiro. 2005.

SILVA, J.; SOMA, N.. Um algoritmo genético aplicado ao problema de empacotamento de bins tridimensionais. **Anais. ENEGEP**, 2002.

SILVA, R.. Otimização do controle eletrônico do diagrama de radiação de arranjos de antenas usando algoritmos genéticos com codificação real. **Dissertação**. Programa de Pós-graduação em Energia Elétrica, UFRN, 2006.

SILVEIRA, R.; UBER JR., A.. Ferramenta de apoio a decisão na escolha de misturas de fardos de fibras para o setor de fiação têxtil utilizando algoritmos genéticos. **Anais. SIMPEP**, 2011.

SILVEIRA, R.; UBER JR., A.. Modelagem de um framework para escalonamento de processos distribuídos utilizando multiagentes e algoritmos genéticos. **Anais. SIMPEP**, 2008.

SOUZA, M.; MARTINS, A.; ARAÚJO, C. Experiências com Simulated Annealing e Busca Tabu na Resolução do Problema de Alocação de Salas. XXXIV **Simpósio Brasileiro de Pesquisa Operacional**. Rio de Janeiro, novembro de 2002. Disponível em:

<http://www.iceb.ufop.br/decom/prof/marcone/Disciplinas/InteligenciaComputacional/SBPO-2002-PAS-TC0106.pdf>. Acesso em: 29 mar. 2012.

SOUZA, M.; SOUZA, S.; RIBEIRO, F.. Uma metodologia adaptativa baseada em algoritmos genéticos para a resolução do problema de sequenciamento em uma máquina com penalização por antecipação e atraso da produção. **Anais. ENEGEP**, 2009.

TINÓS, R.. Comportamento auto-organizável em algoritmos genéticos aplicados a robôs móveis em ambientes dinâmicos. **Revista Controle & Automação**, v 18, n.1, jan-mar 2007.

TRIGUEIROS, D.. Reutilização de água em processos industriais: uma abordagem metaheurística. **Tese**. Programa de Pós-Graduação em Engenharia Química, UEM, 2011.

TRINDADE, A.; OCHI, L.. Um algoritmo evolutivo híbrido para a formação de células de manufatura em sistemas de produção. **PESQUISA OPERACIONAL**, v 26, p 255-294, mai-ago 2006.

VIEIRA, L.E. Algoritmo Evolutivo par o Problema do Caixeiro Viajante com Demandas Heterogêneas. **Dissertação** (Mestrado em Engenharia da Produção) – Universidade Federal de Santa Maria, Santa Maria, 2006. Disponível em <http://cascavel.cpd.ufsm.br/tede/tde_arquivos/12/TDE-2008-03-03T104411Z-1346/Publico/LUIS%20VIEIRA.pdf> Acesso em: 01 abr. 2012.

WAINTRAUB, M.. Algoritmos Paralelos de Otimização por Enxame de Partículas em Problemas Nucleares. **Tese de Doutorado**. COPPE – UFRJ, 2009. Disponível em <http://www.nuclear.ufrj.br/DScTeses/teses2009/tese_doutorado_marcelwaintraub.pdf>, acesso em 11 de agosto de 2012.

ZANUZ, L.. **O problema do Caixeiro Viajante**. 2007. Disponível em <<http://www.zanuz.com/2007/09/o-problema-do-caixeiro-viajante.html>>, acesso em 11 de agosto de 2012.

ZIVIANI, N.. **Projeto de Algoritmos – com implementações em PASCAL e C**. 2. Ed. São Paulo : Pioneira Thomson Learning, 2004.

ANEXO A

(Código proposto por Faria(2008) para o problema do Caixeiro Viajante, em linguagem Java)

```

import java.util.Random;
public class AlgoritmoGenetico {
    public static int NUMERO_CIDADES = 8;
    public static int NUMERO_POPULACAO = 10;
    public static void main(String[] args) {
        boolean mostrarEvolucao = true;
        float taxaMortalidade = (float) 0.5;
        int numeroEvolucoes = 3000;
        int[][] mapa = { { 0, 42, 61, 30, 17, 82, 31, 11 },
                        { 42, 0, 14, 87, 28, 70, 19, 33 },
                        { 61, 14, 0, 20, 81, 21, 8, 29 },
                        { 30, 87, 20, 0, 34, 33, 91, 10 },
                        { 17, 28, 81, 34, 0, 41, 34, 82 },
                        { 82, 70, 21, 33, 41, 0, 19, 32 },
                        { 31, 19, 8, 91, 34, 19, 0, 59 },
                        { 11, 33, 29, 10, 82, 32, 59, 0 } };
        String[] cidades = { "A", "B", "C", "D", "E", "F", "G", "H" };
        int[][] cromossomos = new int[NUMERO_POPULACAO][NUMERO_CIDADES];
        int[] resultados = new int[NUMERO_POPULACAO];
        gerarCromossomosAleatoriamente(cromossomos);
        calcularResultado(cromossomos, resultados, mapa);
        ordenar(cromossomos, resultados);
        if (mostrarEvolucao)
            imprimir(cromossomos, resultados, cidades);

        int i;
        for (i = 0; i < numeroEvolucoes; i++) {
            renovarCromossomos(cromossomos, resultados, taxaMortalidade);
            calcularResultado(cromossomos, resultados, mapa);
            ordenar(cromossomos, resultados);
            if (mostrarEvolucao) {
                System.out.println("Geracao: " + (i + 1));
                imprimir(cromossomos, resultados, cidades);
            }
        }
        resultado(cromossomos, resultados, cidades);
    }
    private static void resultado(int[][] cromossomos, int[] resultados, String[] cidades) {
        int i, i2;
        i=0;
        for (i2 = 0; i2 < NUMERO_CIDADES; i2++) {
            System.out.print(cidades[cromossomos[i][i2]] + " => ");
        }
        System.out.print(cidades[cromossomos[i][0]] + " ");
        System.out.println(" Resultado: " + resultados[i]);
    }
    public static void renovarCromossomos(int[][] cromossomos,
        int[] resultados, float taxaMortalidade) {
        int inicioExcluidos = (int) (taxaMortalidade * 10);

```



```

int i, i2 = 0;
for (i = inicioExcluidos; i < 10; i++) {
    boolean valido = false;
    while (!valido) {
        int[] c_tmp = resetaCromossomo();
        int pai1, pai2;
        pai1 = new Random().nextInt(inicioExcluidos);
        do {
            pai2 = new Random().nextInt(inicioExcluidos);
        } while ((pai1 == pai2)
            && (resultados[pai1] != resultados[pai2]));
        for (i2 = 0; i2 < (int) NUMERO_CIDADES / 4; i2++) {
            int pos;
            pos = new Random().nextInt(NUMERO_CIDADES);
            c_tmp[pos] = cromossomos[pai1][pos];
        }
        for (i2 = 0; i2 < (int) NUMERO_CIDADES / 4; i2++) {
            int pos = new Random().nextInt(NUMERO_CIDADES);
            if (c_tmp[pos] == -1) {
if (valorValidoNoCromossomo(cromossomos[pai2][pos],
c_tmp)) {
                c_tmp[pos] = cromossomos[pai2][pos];
            }
        }
    }
    for (i2 = 0; i2 < NUMERO_CIDADES; i2++) {
        if (c_tmp[i2] == -1) {
            int crom_temp = valorValidoNoCromossomo(c_tmp);
            c_tmp[i2] = crom_temp;
        }
    }
    valido = cromossomoValido(c_tmp, cromossomos);
    if (valido) {
        cromossomos[i] = c_tmp;
    }
}
}

private static int[][] gerarCromossomosAleatoriamente(int[][] cromossomos) {
    int[] c_tmp = new int[NUMERO_CIDADES];
    int i, i2;
    for (i = 0; i < NUMERO_POPULACAO; i++) {
        boolean crom_valido = false;
        while (!crom_valido) {
            crom_valido = true;
            c_tmp = resetaCromossomo();
            for (i2 = 0; i2 < NUMERO_CIDADES; i2++) {
                c_tmp[i2] = valorValidoNoCromossomo(c_tmp);
            }
        }
    }
}

```

```

        }
        crom_valido = cromossomoValido(c_tmp, cromossomos);
    }
    cromossomos[i] = c_tmp;
}
return cromossomos;
}
private static int[] resetaCromossomo() {
    int[] c = new int[NUMERO_CIDADES];
    int i;
    for (i = 0; i < NUMERO_CIDADES; i++) {
        c[i] = -1;
    }
    return c;
}
private static int valorValidoNoCromossomo(int[] c_tmp) {
    int crom_temp;
    boolean valido;
    do {
        crom_temp = new Random().nextInt(NUMERO_CIDADES);
        valido = true;
        for (int ii = 0; ii < NUMERO_CIDADES; ii++) {
            if (c_tmp[ii] == crom_temp)
                valido = false;
        }
    } while (!valido);
    return crom_temp;
}
private static boolean valorValidoNoCromossomo(int valor, int[] c_tmp) {
    int crom_temp = valor;
    boolean valido;
    valido = true;
    for (int ii = 0; ii < NUMERO_CIDADES; ii++) {
        if (c_tmp[ii] == crom_temp)
            valido = false;
    }
    return valido;
}
private static boolean cromossomoValido(int[] c_tmp, int[][] cromossomos) {
    int j, j2;
    boolean crom_valido = true;
    for (j = 0; j < NUMERO_POPULACAO; j++) {
        int n_iguais = 0;
        for (j2 = 0; j2 < NUMERO_CIDADES; j2++) {
            if (c_tmp[j2] == cromossomos[j][j2]) {
                n_iguais++;
            }
        }
        if (n_iguais == NUMERO_CIDADES)

```


APÊNDICE A

(Tabela da Pesquisa Bibliométrica)

Tabela 8 – Pesquisa Bibliométrica

| Título | Ano | Fonte | Método | Linguagem | Área da Engenharia de Produção |
|--|------------|------------------------------|----------------------------|------------------|---------------------------------------|
| <i>A Comparative Study Of The Application Of Differential Evolution And Simulated Annealing In Radiative Transfer Problems</i> | 2010 | SCIELO | <i>Simulated Annealing</i> | | Operações e Processos de Produção |
| <i>A Genetic Symbiotic Algorithm Applied To TheOne-Dimensional Cutting Stock Problem.</i> | 2009 | Revista Pesquisa Operacional | Algoritmos Genéticos | Fortran | Operações e Processos de Produção |
| <i>A Planning Model For Offshore Natural Gas Transmission.</i> | 2008 | Revista Pesquisa Operacional | Algoritmos Genéticos | | Logística |
| Algoritmo Evolutivo Híbrido: uma Heurística Promissora para o Problema de Localização de Facilidade. | 2003 | Enegep | <i>Particle Swarm</i> | | Operações e Processos de Produção |
| Algoritmo Genético Aplicado ao Problema de Localização de Hubs com Alocação Simples sem Restrição de Capacidade. | 2010 | SIMPEP | Algoritmos Genéticos | C++ | Operações e Processos de Produção |
| Algoritmo Genético de Agrupamento para Formação de Módulos de Arranjo Físico. | 2008 | G&P | Algoritmos Genéticos | | Operações e Processos de Produção |
| Algoritmo Genético Multicritério para | 2008 | Enegep | Algoritmos Genéticos | | Operações e Processos |

| | | | | | | |
|---|------|------------------------------|----------------------|--------------|--|-----------------------------------|
| Alocação de Dispositivos de Controle em uma Rede de Transmissão Elétrica. | | | | | | de Produção |
| Algoritmo Genético para Formação de Células de Fabricação. | 2004 | Enegep | Algoritmos Genéticos | Visual Basic | | Operações e Processos de Produção |
| Algoritmo Genético para o Balanceamento de Linhas de Produção. | 2003 | Enegep | Algoritmos Genéticos | | | Operações e Processos de Produção |
| Algoritmo Genético Simbiótico para Minimizar o Número de Objetos Processados e o Setup dado o Número Máximo de Pilhas Abertas num Problema de Corte de Estoque. | 2010 | PODes | Algoritmos Genéticos | Fortran | | Operações e Processos de Produção |
| Algoritmos Evolucionários na Determinação da Configuração de Custo Mínimo de Sistemas de Co-Geração de Energia com Base no Gás Natural. | 2005 | Revista Pesquisa Operacional | Algoritmos Genéticos | | | Operações e Processos de Produção |
| Algoritmos Evolucionários na Otimização de Alternativas para o Desenvolvimento de Campos de Petróleo. | 2003 | Enegep | Algoritmos Genéticos | | | Operações e Processos de Produção |
| Algoritmos Genéticos Aplicados à Programação da Produção de um Processo de Fundição de Zinco. | 2003 | Enegep | Algoritmos Genéticos | | | Operações e Processos de Produção |

| | | | | | |
|---|------|--------|----------------------|-----|-----------------------------------|
| Algoritmos Genéticos Aplicados na Simulação do Processo de Produção do Malte de Milho. | 2009 | Enegep | Algoritmos Genéticos | | Operações e Processos de Produção |
| Algoritmos Genéticos com Operação de Seleção <i>Hawk-Dove</i> com Estratégia <i>Tit For Tat</i> , e com Operação de Recombinação com Taxas Variáveis. | 2002 | Enegep | Algoritmos Genéticos | | Pesquisa Operacional |
| Algoritmos Genéticos e Computação Paralela para Problemas de Roteirização de Veículos com Janelas de Tempo e Entregas Fracionadas. | 2006 | G&P | Algoritmos Genéticos | C | Logística |
| Algoritmos Genéticos e Variantes na Solução de Problemas de Configuração de Redes de Distribuição. | 2008 | SCIELO | Algoritmos Genéticos | | Operações e Processos de Produção |
| Algoritmos Genéticos Híbridos Aplicados ao Problema da Árvore Mínima com Grupamentos. | 2009 | SIMPEP | Algoritmos Genéticos | C | Logística |
| Algoritmos Genéticos: Alguns Experimentos Com os Operadores de Cruzamento (“ <i>Crossover</i> ”) para o Problema do Caixeiro Viajante Assimétrico. | 2006 | Enegep | Algoritmos Genéticos | C++ | Logística |

| | | | | | |
|---|------|--------|----------------------------|--------|-----------------------------------|
| Alocação Eficiente De Indicadores de Faltas em um Sistema de Distribuição Real Usando Computação Evolutiva. | 2012 | SCIELO | Algoritmos Genéticos | | Qualidade |
| Análise da Flutuação do Gargalo em <i>Flow Shop Permutacional</i> com Tempos de <i>Setup</i> Assimétricos e Dependentes da Seqüência. | 2004 | G&P | <i>Simulated Annealing</i> | Delphi | Operações e Processos de Produção |
| Aplicação da Metaheurística Algoritmo Genético com <i>Path Relinking</i> para a Solução de um Problema de Planejamento Florestal. | 2011 | SIMPEP | Algoritmos Genéticos | Pascal | Operações e Processos de Produção |
| Aplicação da Relaxação Lagrangeana e do Algoritmo Genético Construtivo na Solução do Problema Probabilístico de Localização-Alocação de Máxima Cobertura. | 2006 | G&P | Algoritmos Genéticos | | Logística |
| Aplicação de Algoritmos Genéticos na Determinação da Operação Ótima de Sistemas Hidrotérmicos de Potência. | 2006 | SCIELO | Algoritmos Genéticos | | Operações e Processos de Produção |
| Aplicação de Inteligência Computacional ao Problema Logístico de Escoamento Contingencial de Tráfego Aéreo. | 2007 | SIMPEP | Algoritmos Genéticos | | Logística |
| Aplicação do Algoritmo Genético na Análise de Medicamentos Anti-Inflamatórios | 2004 | Enegep | Algoritmos Genéticos | | Qualidade |

| | | | | | | |
|---|------|--------|----------------------------|--------------|--|-----------------------------------|
| Industrializados. | | | | | | |
| <i>Application of Simulated Annealing In Simulation And Optimization Of Drying Process of Zea Mays Malt.</i> | 2011 | SCIELO | <i>Simulated Annealing</i> | | | Qualidade |
| <i>Channel Characteristics In Tunnels: FDTD Simulations And Measurement.</i> | 2011 | SCIELO | <i>Simulated Annealing</i> | | | Operações e Processos de Produção |
| Cinética e Otimização do Processo de Secagem do Malte de Milho Usando <i>Simulated Annealing.</i> | 2010 | Enegep | <i>Simulated Annealing</i> | | | Qualidade |
| Comportamento Auto-Organizável em Algoritmos Genéticos Aplicados A Robo. S Móveis em Ambientes Dinâmicos. | 2007 | SCIELO | Algoritmos Genéticos | | | Pesquisa Operacional |
| Contribuição da Inteligência Artificial para a Viabilização da Estratégia de Manufatura: Um Estudo de Caso na Indústria de Fundição. | 2008 | SIMPEP | Algoritmos Genéticos | Visual Basic | | Operações e Processos de Produção |
| Desenvolvimento de Algoritmos Genéticos Para o Problema das P-Medianas Utilizando Operadores Genéticos de Cruzamento Convencionais E Não-Convencionais. | 2009 | SIMPEP | Algoritmos Genéticos | C | | Logística |
| <i>Design Of Distributed Optical-Fiber Raman</i> | 2011 | SCIELO | <i>Particle Swarm</i> | | | Operações e Processos |

| | | | | | | | |
|--|------|------------------|--|----------------------------|------|--|-----------------------------------|
| <i>Amplifiers Using Multi-Objective Particle Swarm Optimization.</i> | | | | | | | de Produção |
| Detecção de Falhas em Estruturas Inteligentes usando Otimização por Nuvem de Partículas: Fundamentos e Estudo de Casos. | 2006 | SCIELO | | <i>Particle Swarm</i> | | | Qualidade |
| Dimensionamento Evolutivo de Usinas Hidroelétricas. | 2004 | SCIELO | | Algoritmos Genéticos | C++ | | Operações e Processos de Produção |
| <i>Discrete Optimization Of Trusses By Simulated Annealing.</i> | 2004 | SCIELO | | <i>Simulated Annealing</i> | | | Operações e Processos de Produção |
| <i>Economic-Statistical Design Of Variable Parameters Non-Central Chi-Square Control Chart.</i> | 2011 | Revista Produção | | <i>Simulated Annealing</i> | | | Qualidade |
| Eficiência dos Métodos de Otimização <i>Simulated Annealing</i> , Delineação Rápida em Cadeia e Ramos e Conexões para Construção de Mapas Genéticos. | 2009 | SCIELO | | <i>Simulated Annealing</i> | Java | | Operações e Processos de Produção |
| Extensão de um Algoritmo Cultural para Problemas de Despacho de Energia Elétrica | 2007 | DEP-UEM | | Algoritmos Genéticos | Java | | Operações e Processos de Produção |
| Ferramenta de Apoio a Decisão na Escolha de Misturas de Fardos de Fibras para o Setor | 2011 | SIMPEP | | Algoritmos Genéticos | | | Operações e Processos de Produção |

| | | | | | | |
|---|------|------------------------------|----------------------------|--------------|-----------------------------------|--|
| de Fiação Têxtil Utilizando Algoritmos Genéticos. | | | | | | |
| <i>Genetic Algorithms Applied To A Faster Distance Protection Of Transmission Lines.</i> | 2011 | SCIELO | Algoritmos Genéticos | | Operações e Processos de Produção | |
| <i>Heteropentámeros (Etanol)4-Agua: Estudio Estructural Y Termodinámico</i> | 2010 | SCIELO | <i>Simulated Annealing</i> | | Qualidade | |
| Heurística baseada no <i>Simulated Annealing</i> aplicada ao problema de alocação de berços. | 2008 | GEPROS | <i>Simulated Annealing</i> | C++ | Logística | |
| Heurística Evolutiva Para a Redução do Estoque em Processamento em Sistemas de Produção <i>Flow Shop</i> Permutacional. | 2007 | Enegep | Algoritmos Genéticos | C | Operações e Processos de Produção | |
| Implementação de um Algoritmo Genético para uma Aplicação do Problema do Caixeiro Viajante. | 2006 | DEP-UEM | Algoritmos Genéticos | Pascal | Logística | |
| Integração do <i>Bin Packing Problem</i> com o Problema de Roteamento de Veículos Utilizando um Algoritmo Genético. | 2011 | SIMPEP | Algoritmos Genéticos | Visual Basic | Logística | |
| <i>Interactive Simulated Annealing For Solving Imprecise Discrete Multiattribute Problems Under Risk.</i> | 2002 | Revista Pesquisa Operacional | <i>Simulated Annealing</i> | | Pesquisa Operacional | |
| Metaheurística <i>Simulated Annealing</i> para | 2004 | SCIELO | <i>Simulated Annealing</i> | Delphi | Operações e Processos | |

| | | | | | | | |
|--|------|--------|---|--------|--|--|-----------------------------------|
| Solução de Problemas de Planejamento Florestal com Restrições De Integridade. | | | | | | | de Produção |
| Metaheurísticas Aplicadas na Solução do Problema do Caixeiro Viajante Com Demandas Heterogêneas. | 2004 | Enegep | Algoritmos Genéticos e <i>Simulated Annealing</i> | | | | Logística |
| Metaheurísticas para Problema De <i>Flowshop</i> Flexível com Penalidades de Adiantamento e Atraso. | 2011 | SIMPEP | Algoritmos Genéticos | C++ | | | Operações e Processos de Produção |
| Metaheurísticas <i>Simulated Annealing</i> e <i>Iterated Local Search</i> Aplicadas ao Problema de P-Medianas. | 2007 | SIMPEP | <i>Simulated Annealing</i> | C++ | | | Logística |
| Modelagem de um <i>Framework</i> para Escalonamento de Processos Distribuído Utilizando Sistema Multiagentes e Algoritmos Genéticos. | 2008 | SIMPEP | Algoritmos Genéticos | Java | | | Operações e Processos de Produção |
| Modelo Hipercubo Integrado a um Algoritmo Genético para Análise de Sistemas Médicos Emergenciais em Rodovias. | 2006 | G&P | Algoritmos Genéticos | Pascal | | | Operações e Processos de Produção |
| Modelo <i>Team</i> de Localização de Ambulâncias - Solução Através do | 2005 | Enegep | Algoritmos Genéticos | | | | Logística |

| | | | | | | |
|--|------|------------------------------|----------------------|-----------------------|--|-----------------------------------|
| Algoritmo Genético Construtivo. | | | | | | |
| <i>Multiobjective Optimization Of Availability And Cost In Repairable Systems Design Via Genetic Algorithms And Discrete Event Simulation.</i> | 2009 | Revista Pesquisa Operacional | Algoritmos Genéticos | C++ | | Operações e Processos de Produção |
| <i>Optimal Design Of Double Folded Stub Microstrip Filter By Neural Network Modelling And Particle Swarm Optimization.</i> | 2012 | SCIELO | | <i>Particle Swarm</i> | | Operações e Processos de Produção |
| Otimização Combinatória Empregando Algoritmo Genético Aplicada na Análise Multivariada de Medicamentos Manipulados. | 2004 | Enegep | | Algoritmos Genéticos | | Qualidade |
| Otimização de Métodos De Controle de Qualidade de Fármacos Usando Algoritmo Genético e Busca Tabu. | 2003 | Revista Pesquisa Operacional | Algoritmos Genéticos | | | Qualidade |
| Otimização de uma Célula de Manufatura Utilizando Simulação Computacional. | 2004 | Enegep | | Algoritmos Genéticos | | Operações e Processos de Produção |
| Otimização discreta por nuvem de partículas aplicada ao problema do caixeiro viajante. | 2006 | GEPROS | | <i>Particle Swarm</i> | | Logística |
| Otimização do Investimento em Saldo de | 2009 | Enegep | | Algoritmos Genéticos | | Organizacional |

| | | | | | |
|--|------|--------|---|--|-----------------------------------|
| Caixa: um Complemento ao Modelo de Miller e Orr. | | | | | |
| Otimização do Planejamento Mestre da Produção Através de Algoritmos Genéticos. | 2002 | Enegep | Algoritmos Genéticos | | Operações e Processos de Produção |
| Otimização do Problema de Layout de Facilidades Através da Técnica Enxame de Partículas Utilizando a Modelagem <i>Attractor-Repeller</i> . | 2006 | Enegep | <i>Particle Swarm</i> | | Operações e Processos de Produção |
| Otimização do Saldo de Caixa: Uma Comparação entre Algoritmos Genéticos e <i>Particle Swarm Optimization</i> . | 2010 | Enegep | Algoritmos Genéticos E <i>Particle Swarm</i> | | Operações e Processos de Produção |
| Otimização Numérica e Experimental de Perfis U Formados a Frio Sujeitos à Compressão. | 2008 | SCIELO | <i>Simulated Annealing</i> | | Operações e Processos de Produção |
| Otimização por Enxame de Partículas de Modelos de Regressão Visando a Identificação de Adultrações nas Blendas Biodiesel/Diesel. | 2008 | Enegep | <i>Particle Swarm</i> | | Qualidade |
| Planejamento e Otimização do Desenvolvimento de um Campo de Petróleo por Algoritmos Genéticos. | 2002 | Enegep | Algoritmos Genéticos | | Operações e Processos de Produção |

| | | | | | |
|---|------|--------|----------------------------|-----|-----------------------------------|
| Problema de Alocação de Berços Heterogêneos em Portos Graneleiros com Restrições de Estoque e Condições Favoráveis de Maré. | 2009 | SIMPEP | <i>Simulated Annealing</i> | | Logística |
| Problema de Cobertura de Arcos – Um Estudo de Caso. | 2003 | Enegep | Algoritmos Genéticos | | Logística |
| Procedimento para a Geração de Arranjos Físicos Distribuídos Baseado em Otimização da Simulação e Algoritmos Genéticos. | 2007 | Enegep | Algoritmos Genéticos | C++ | Operações e Processos de Produção |
| Programação da Produção em uma Indústria de Tintas. | 2003 | SIMPEP | <i>Simulated Annealing</i> | C | Operações e Processos de Produção |
| Proposta de Solução para o Problema das P-Medianas na Localização de Unidades de Saúde 24 Horas. | 2002 | Enegep | Algoritmos Genéticos | | Logística |
| Resolução do problema de carregamento e descarregamento 3D de contêineres em terminais portuários, via representação por regras e algoritmo genético. | 2011 | GEPROS | Algoritmos Genéticos | C | Logística |
| Resolução do Problema de Carregamento e Descarregamento de Contêineres num Navio Para Múltiplos Cenários via Representação | 2011 | SIMPEP | Algoritmos Genéticos | C | Logística |

Por Regras e Algoritmo Genético.

| | | | | | |
|--|------|------------------|---|-----|-----------------------------------|
| <i>Scheduling Copper Refining And Casting Operations By Means Of Heuristics For The Flexible Flow Shop Problem.</i> | 2011 | SCIELO | <i>Simulated Annealing</i> | C++ | Operações e Processos de Produção |
| <i>Simulated Annealing</i> Aplicado à Programação da Tripulação no Sistema de Transporte Público. | 2002 | Enegep | <i>Simulated Annealing</i> | | Logística |
| Solução do Problema de Superação de Disjuntores Pela Alocação Ótima de Dispositivos Limitadores de Corrente de Curto-Circuito Utilizando Algoritmos Genéticos. | 2007 | SCIELO | Algoritmos Genéticos | | Operações e Processos de Produção |
| Técnicas Metaheurísticas Aplicadas na Otimização de Parâmetros de um Modelo Probabilístico de Gestão de Estoques. | 2009 | SIMPEP | Algoritmos Genéticos e <i>Simulated Annealing</i> | | Operações e Processos de Produção |
| Tecnologia de Grupo: Algoritmos Genéticos na Solução de Problemas de Agrupamentos Celulares de Máquinas e Peças. | 2003 | Enegep | Algoritmos Genéticos | | Operações e Processos de Produção |
| <i>Testing And Preventive Maintenance</i> | 2009 | Revista Pesquisa | Algoritmos Genéticos | | Operações e Processos |

| | | | | | | |
|---|------|-------------|----------------------------|--------|--|-----------------------------------|
| <i>Scheduling Optimization For Aging Systems Modeled By Generalized Renewal Process.</i> | | Operacional | | | | de Produção |
| Um Algoritmo Baseado Em Iterated Local Search E Simulated Annealing para a Resolução de um Problema de Corte Bidimensional com Rotação de Itens. | 2009 | SIMPEP | <i>Simulated Annealing</i> | C | | Operações e Processos de Produção |
| Um algoritmo enxame de partículas para uma variante do problema de máxima cobertura. | 2012 | GEPROS | <i>Particle Swarm</i> | Pascal | | Logística |
| Um Algoritmo Evolutivo Híbrido para o Problema de Programação de Horários em Escolas. | 2002 | Enegep | Algoritmos Genéticos | C++ | | Operações e Processos de Produção |
| Um Algoritmo Genético Aplicado ao Problema de Empacotamento de Bins Tridimensionais. | 2002 | Enegep | Algoritmos Genéticos | C | | Operações e Processos de Produção |
| Um Algoritmo Genético para o Problema de Alocação de Antenas de Transmissão. | 2006 | Enegep | Algoritmos Genéticos | Pascal | | Operações e Processos de Produção |
| Um Algoritmo Híbrido Baseado em Colônia de Formigas e Recozimento Simulado Para Problemas de Escalonamento com Restrição de Recursos e Múltiplos Modos de | 2006 | Enegep | <i>Simulated Annealing</i> | | | Operações e Processos de Produção |

| | | | | | | | | |
|--|----------------------------|------|--------|----------------------------|--------------|-----------------------------------|--|--|
| Processamento. | | | | | | | | |
| Um Algoritmo Eficiente para o Problema de Roteamento de Veículos com Janela de Tempo. | <i>Simulated Annealing</i> | 2005 | Enegep | <i>Simulated Annealing</i> | C | Logística | | |
| Um Estudo Das Heurísticas <i>Simulated Annealing</i> e VNS Aplicadas ao Problema de Programação de Tripulações. | <i>Simulated Annealing</i> | 2003 | Enegep | <i>Simulated Annealing</i> | C++ | Operações e Processos de Produção | | |
| Um Método Heurístico Construtivo para o Problema da Grade Horária Escolar. | | 2012 | PODes | Algoritmos Genéticos | Visual Basic | Operações e Processos de Produção | | |
| Um Método Híbrido para a Resolução do Problema de Roteamento de Veículos. | | 2004 | Enegep | <i>Simulated Annealing</i> | Não Consta | Logística | | |
| Uma Abordagem Eficiente para Resolver os Problemas de Despacho Econômico e Ambiental. | | 2003 | SIMPEP | Algoritmos Genéticos | | Operações e Processos de Produção | | |
| Uma Abordagem Heurística para o Problema de Corte Guilhotinado Bidimensional Aplicado a uma Situação de Canibalização. | | 2009 | SIMPEP | Algoritmos Genéticos | | Operações e Processos de Produção | | |
| Uma Abordagem Híbrida para o Problema de Formação de Grupos e Balanceamento de Linhas de Montagem Smt. | | 2002 | Enegep | Algoritmos Genéticos | | Operações e Processos de Produção | | |
| Uma Abordagem Multiagente para a | | 2009 | Enegep | Algoritmos Genéticos | | Operações e Processos | | |

| | | | | | | |
|--|------|------------------|----------------------------|--------|--|-----------------------------------|
| Solução de Problemas do Tipo <i>Jobshop</i> . | | | | | | de Produção |
| Uma Abordagem para Resolução do Problema De Localização de Atividades Econômicas. | 2005 | Enegep | Algoritmos Genéticos | Delphi | | Operações e Processos de Produção |
| Uma Aplicação da Metaheurística Algoritmo Genético ao Problema de Carregamento de Navios-Contêineres. | 2012 | PODes | Algoritmos Genéticos | C | | Logística |
| Uma Metodologia Adaptativa Baseada em Algoritmos Genéticos para a Resolução do Problema de Sequenciamento em uma Máquina com Penalização por Antecipação e Atraso da Produção. | 2009 | Enegep | Algoritmos Genéticos | C | | Operações e Processos de Produção |
| Uma Modelagem para Programação da Produção de Sistemas de Manufatura com Recursos Compartilhados Utilizando Algoritmos Genéticos. | 2007 | Enegep | Algoritmos Genéticos | | | Operações e Processos de Produção |
| Uma Nova Abordagem para o Problema <i>Dial-A-Ride</i> . | 2009 | Revista Produção | <i>Simulated Annealing</i> | C++ | | Logística |
| Uso do Algoritmo Genético para a Solução de Problemas de Sequenciamento da Produção | 2007 | Enegep | Algoritmos Genéticos | | | Operações e Processos de Produção |

| | | | | |
|--|--------|----------------------|-----|----------------------|
| Utilização de Algoritmos Genéticos em 2004 | Enegep | Algoritmos Genéticos | C++ | Pesquisa Operacional |
| Metodologia Multicritério: Uma Solução para Inconsistência Matricial. | | | | |
| Utilização do Planejamento de Experimentos 2011 | SIMPEP | Algoritmos Genéticos | | Pesquisa Operacional |
| Para Análise de Significância dos Parâmetros de um Algoritmo Genético. | | | | |
