

RICARDO MARTINS DE ABREU SILVA

**OTIMIZAÇÃO BASEADA EM COLÔNIA DE FORMIGAS APLICADA AO
PROBLEMA DE COBERTURA DE CONJUNTOS**

Tese apresentada à Pós-Graduação em Ciência da
Computação do Centro de Informática da
Universidade Federal de Pernambuco como
requisito parcial à obtenção do grau de Doutor em
Ciência da Computação

Orientador: Prof. Dr. Geber Lisboa Ramalho

RECIFE, 2003

RICARDO MARTINS DE ABREU SILVA

**OTIMIZAÇÃO BASEADA EM COLÔNIA DE FORMIGAS APLICADA AO
PROBLEMA DE COBERTURA DE CONJUNTOS**

Esta tese foi julgada adequada à obtenção do grau de Doutor em Ciência da Computação e aprovada em sua forma final pela Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco.

Recife, 2003

Prof. Dr. Álvaro José Periotto

COPPE – UFRJ, Brasil

Prof. PhD Aluizio Fausto Ribeiro Araújo

University of Sussex, Brighton, Inglaterra

Profa. Dra. Marcília Andrade Campos

Universidade Federal de Pernambuco, Brasil

Profa. Dra. Maria Cristina Falcão Raposo

Universidade Federal de Pernambuco, Brasil

Profa. PhD. Patrícia Cabral de Azevedo Restelli Tedesco

University of Leeds, UK

DEDICATÓRIAS

*Dedico esta tese a minha querida mãe, Sebastiana de Abreu Silva,
e à memória de meu pai, Rubens Martins Silva.*

AGRADECIMENTOS

Recebi muita ajuda para realizar esta tese. Prof. Geber Lisboa Ramalho, por acreditar em meu trabalho. Minha mãe, pelo seu amor incondicional. Minha família, sempre torcendo por mim. Flávia Loschi, pela paciência e incentivo. Meus amigos, com os quais sempre posso contar. Prof. Olavo de Carvalho, pela metanóia operada em mim. E principalmente a Deus, por todas as graças concedidas a este pobre pecador.

EPÍGRAFE

"Somente a consciência individual do agente dá testemunho dos atos sem testemunha, e não há ato mais desprovido de testemunha externa do que o ato de conhecer."

Olavo de Carvalho

RESUMO

O presente trabalho avalia o desempenho e o funcionamento da meta-heurística *Ant Colony Optimization – ACO* em instâncias de grande porte do problema de cobertura de conjuntos (*Set Covering Problem - SCP*). A meta-heurística *ACO* é um método de otimização baseado no comportamento de colônias de formigas reais e que vem obtendo resultados promissores em vários problemas combinatoriais. Entretanto, nós constatamos que, na maior parte dos artigos publicados pela comunidade *ACO*, a análise efetuada sobre as heurísticas não seguia um método de avaliação rigoroso, principalmente no que se refere à carência de estudos da influência dos parâmetros destas heurísticas sobre a qualidade dos resultados alcançados. Uma vez que eventuais descuidos ocorridos na etapa de avaliação de um algoritmo podem levar a conclusões equivocadas a respeito de seu desempenho, resolvemos utilizar um método de análise experimental para avaliar a adaptação da meta-heurística *ACO* em algum problema de otimização combinatorial previamente abordado pela comunidade *ACO*. O interesse em torno das instâncias de grande porte do problema de cobertura de conjuntos surgiu de sua complexidade (NP-Completo), e de sua capacidade de atender uma grande quantidade de problemas reais, os quais geralmente não possuem escala reduzidas na prática.

A principal contribuição deste trabalho, sobretudo com relação à surpresa do seu resultado em vista da literatura vigente, encontra-se na revelação da pouca importância do feromônio no método *ACO* em instâncias *SCP* de grande porte, assim como na exposição de teorias, baseadas no conceito da correlação da distância de adaptação, capazes de explicar não somente as causas responsáveis pela atuação do feromônio, mas também a melhoria oriunda das hibridizações (via busca local) do método *ACO* em *SCP*, a ponto deste último ser prescindível. Ou seja, chegamos à conclusão de que não se justifica a utilização do método *ACO* em instâncias *SCP* de grande porte, uma vez que existem técnicas mais simples e eficientes capazes de tratar este mesmo problema, como por exemplo, a busca local desenvolvida por Jacobs e Brusco (1995).

Palavras-chave: heurísticas de otimização, análise experimental, colônia de formigas artificiais.

ABSTRACT

The present work evaluates the performance and operation of the Ant Colony Optimization meta-heuristic (ACO) in large instances of the Set Covering Problem (SCP). The ACO meta-heuristic is a combinatorial optimization method based on the behavior of real ants colonies and that has obtained promising results in several combinatorial problems. However, we verified that, in mostly of the ACO community's work, the analysis of the heuristics didn't follow a rigorous evaluation method, mainly in what refers to the lack of studies about the influence of the parameters in the results reached by these heuristics. Once eventual negligence happened in the stage of algorithm evaluation can take to mistaken conclusions about your acting, we decided to use an experimental analysis method to evaluate the adaptation of the ACO meta-heuristic in some combinatorial problem previously approached by the ACO community. The interest around the large instances of the set covering problem came from your complexity (NP-complete), and of yours capacity to assist a great amount of real problems, usually with large scale in practice.

Despite of the effective literature, the main contribution of this work is in the revelation of the little importance of the ACO method's pheromone in large SCP instances, as well as in the exhibition of theories, based on the concept of the correlation of the adaptation distance, capable to explain not only the responsible causes for the pheromone's action, but also to explain the improvement from ACO method's hybridizations for SCP (through local search), to point of this last one to be dispensable. In other words, we reached the conclusion that the it is not justified the use of this method in large SCP instances, once simpler and efficient techniques exist capable to treat this same problem, as for instance, the local search developed for Jacobs and Brusco (1995).

Keywords: optimization heuristic, experimental analysis, colony of artificial ants.

SUMÁRIO

EPÍGRAFE	V
SUMÁRIO	IX
TABELAS	XI
FIGURAS	XII
QUADROS	XIII
1. INTRODUÇÃO	14
2. O PROBLEMA DE COBERTURA DE CONJUNTOS E SUAS HEURÍSTICAS	18
2.1 DEFINIÇÃO DO PROBLEMA DE COBERTURA DE CONJUNTOS	18
2.2 COMPLEXIDADE DO PROBLEMA DE COBERTURA DE CONJUNTOS	21
2.3 HEURÍSTICAS PARA O PROBLEMA DE COBERTURA DE CONJUNTOS	27
2.3.1 <i>Relaxação Lagrangeana de SCP e seu Modelo Dual</i>	31
2.3.2 <i>Heurísticas Lagrangeanas</i>	33
2.4 LS-ECC-SCP: A BUSCA LOCAL DE JACOBS E BRUSCO	34
2.4.1 <i>Módulo CONSTRUIR</i>	35
2.4.2 <i>Módulo BUSCAR</i>	37
2.5 SA-SCP: O RECOZIMENTO SIMULADO DE JACOBS E BRUSCO	40
2.6 GA-SCP: O ALGORITMO GENÉTICO DE BEASLEY E CHU	41
2.6.1 <i>População Inicial</i>	42
2.6.2 <i>Seleção por Torneio</i>	44
2.6.3 <i>Reprodução por Fusão</i>	44
2.6.4 <i>Mutação</i>	45
2.6.5 <i>Viabilização e Ajuste da Solução</i>	45
2.6.6 <i>Reposição Incremental</i>	46
3. OTIMIZAÇÃO BASEADA EM COLÔNIA DE FORMIGAS	48
3.1 EXPERIÊNCIA COM FORMIGAS REAIS	48
3.2 ANT SYSTEM E O PROBLEMA DO CAIXEIRO VIAJANTE	50
3.3 META-HEURÍSTICA <i>ACO</i> (<i>ANT COLONY OPTIMIZATION</i>).....	53
3.4 ALGORITMOS <i>ACO</i>	59
3.5 <i>MAX-MIN ANT SYSTEM</i> : A HEURÍSTICA DE STÜTZLE E HOSS	60
4. ACO E PROBLEMA DE COBERTURA DE CONJUNTOS	68
4.1 META-HEURÍSTICA <i>ACO</i> EM PROBLEMAS DE SUBCONJUNTOS.....	68
4.2 <i>ANT SYSTEM</i> E O PROBLEMA DE COBERTURA DE CONJUNTOS	69

4.3	<i>MAX-MIN ANT SYSTEM</i> E O PROBLEMA DE COBERTURA DE CONJUNTOS	73
4.4	<i>ACS-SCP</i> : O ALGORITMO DE HADJI, RAHOUAL, TALBI E BACHELET	78
4.5	HIBRIDIZAÇÕES SOBRE A HEURÍSTICA <i>AS-SCP</i>	82
4.6	<i>MAXMINLS-SCP</i> : A HIBRIDIZAÇÃO DA HEURÍSTICA <i>MAXMIN-SCP</i>	91
5.	MÉTODO EXPERIMENTAL	92
5.1	ESTRATÉGIA DE ANÁLISE	92
5.2	MÉTODOS ESTATÍSTICOS COMPARATIVOS.....	94
5.3	MÉTODO DE REGRESSÃO	99
5.4	DELINEAMENTO EXPERIMENTAL D-ÓTIMO	103
5.4.1	<i>Módulos CONSTRUIR e BUSCAR</i>	106
6.	RESULTADOS EXPERIMENTAIS.....	108
6.1	CONSIDERAÇÕES GERAIS.....	108
6.1.1	<i>Métricas de desempenho</i>	109
6.1.2	<i>Unidades experimentais</i>	109
6.1.3	<i>Ambiente computacional</i>	110
6.1.4	<i>Heurísticas do enfoque comparativo</i>	111
6.1.5	<i>Fatores e Níveis do enfoque causal</i>	111
6.1.6	<i>Relato dos resultados experimentais</i>	112
6.2	ANÁLISE DE <i>AS</i> SOBRE <i>SCP</i>	114
6.3	ANÁLISE DA RELAÇÃO PARÂMETROS-RESULTADOS DA HEURÍSTICA <i>AS-SCP</i>	118
6.4	ANÁLISE DE <i>MAX-MIN</i> SOBRE <i>SCP</i>	119
6.5	ANÁLISE DE VERSÕES HÍBRIDAS	121
6.6	ANÁLISE DAS COMPARAÇÕES COM OUTRAS HEURÍSTICAS	124
7.	DISCUSSÃO DOS RESULTADOS.....	125
7.1	CORRELAÇÃO DA DISTÂNCIA DE ADAPTAÇÃO EM <i>SCP</i>	126
7.2	SOBRE AS HEURÍSTICAS <i>AS-SCP</i> E <i>MAXMIN-SCP</i>	128
7.3	SOBRE AS VERSÕES HÍBRIDAS DE <i>AS-SCP</i> E <i>MAXMIN-SCP</i>	129
7.4	SOBRE A COMPARAÇÃO COM A BUSCA LOCAL DE JACOBS E BRUSCO	131
8	CONCLUSÃO	133
8.1	RESUMO	133
8.2	PRINCIPAIS DIFICULDADES ENCONTRADAS.....	135
8.3	CONTRIBUIÇÕES	135
8.4	TRABALHOS FUTUROS.....	137
	REFERÊNCIAS BIBLIOGRÁFICAS.....	138

TABELAS

Tabela 1: Heurísticas para o Problema de Cobertura de Conjuntos.	27
Tabela 2: Heurísticas nas instâncias das classes A, B, C e D.....	28
Tabela 3: Heurísticas nas instâncias das classes E, F, G e H.	29
Tabela 4: Fonte das instâncias de dimensões maiores de SCP.....	30
Tabela 5: Aplicações do Algoritmo <i>ACO</i>	59
Tabela 6: Descrição das conclusões oriundas do fluxograma da Figura 5	96
Tabela 7: Fonte das instâncias de dimensões maiores de <i>SCP</i>	110
Tabela 8: Tratamentos em cada etapa do enfoque comparativo na avaliação de <i>AS</i> sobre <i>SCP</i> ..	111
Tabela 9: Correlação da distância de adaptação em instâncias SCP da biblioteca OR-Library...	127

FIGURAS

Figura 1: Instância genérica do problema de cobertura de conjuntos.....	19
Figura 2: Representação binária de um indivíduo	42
Figura 3: Esquema das Experiências de Goss, Aron, Deneuborg e Pasteels	49
Figura 4: Estratégia de avaliação experimental de <i>AS</i> sobre <i>SCP</i>	94
Figura 5: Estratégia para comparação de dois algoritmos.....	97
Figura 6: Inferência Estatística: Duas Médias.....	98
Figura 7: Diagrama da Metodologia de Regressão.....	102
Figura 8: Estratégia para análise de pontos extremos.....	103
Figura 9: Formato geral de um diagrama de caixa (<i>boxplot</i>).....	113
Figura 10: <i>Boxplots</i> de <i>AS-Path-SCP-Pure</i> e <i>LS-ECC-SCP</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	114
Figura 11: <i>Boxplots</i> de <i>AS-Path-SCP-Pure</i> e <i>AS-SCP-Pure</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	115
Figura 12: <i>Boxplots</i> de <i>AS-SCP-Pure</i> e <i>AS-SCP</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	116
Figura 13: <i>Boxplots</i> de <i>AS-SCP</i> e <i>LSECCSCP</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	116
Figura 14: <i>Boxplots</i> de <i>AS-SCP</i> e <i>H-SCP</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	117
Figura 15: Desvio percentual médio dos resultados de <i>AS-SCP</i> em função de α e β	119
Figura 16: <i>Boxplots</i> de <i>AS-SCP</i> e <i>MAXMINSCP</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	120
Figura 17: <i>Boxplots</i> das hibridizações de <i>AS-SCP</i> e <i>MAXMIN-SCP</i> em (a) <i>scp4</i> , (b) <i>scp5</i> e (c) <i>scpa</i>	124
Figura 18: <i>Boxplots</i> das heurísticas <i>LS-ECC-SCP</i> , <i>AS-SCP</i> e <i>GA-SCP</i> em <i>scp4</i> , <i>scp5</i> e <i>scpa</i>	125

QUADROS

Quadro 1: Heurística LS-ECC-SCP.....	35
Quadro 2: Heurística SA-SCP de Jacobs e Brusco.....	40
Quadro 3: <i>AS-TSP: Ant System</i> no Problema do Caixeiro Viajante.	53
Quadro 4: Meta-heurística em pseudo-código.....	57
Quadro 5: Rotina <i>geração_e_atividade_das_formigas()</i> em pseudo-código.....	58
Quadro 6: Esquema geral da heurística <i>MMAS</i>	62
Quadro 7: <i>Ant System</i> no problema de cobertura de conjuntos.....	73
Quadro 8: MAXMINSCP: <i>MAX-MIN Ant System</i> para <i>SCP</i>	74
Quadro 9: Heurística ACS-SCP de Hadji, Rahoual, Talbi e Bachelet (HADJI et al. 2000).....	80
Quadro 10: Heurística <i>AS-ECC-SCP</i>	83
Quadro 11: Heurística <i>AS-ECC-LSK-SCP</i>	85
Quadro 12: Heurística <i>AS-ECC-LSK-LS-SCP</i>	86
Quadro 13: <i>AS-DIV-ECC-LSK-LS-SCP</i>	87
Quadro 14: Heurística <i>AS-ECC-LSKIN-SCP</i>	88
Quadro 15: <i>AS-ECC-LSKIN-LS-SCP</i>	89
Quadro 16: <i>AS-DIV-ECC-LSKIN-LS-SCP</i>	90
Quadro 17: Heurística <i>MAXMINLS-SCP</i>	91
Quadro 18: Heurística LS-SCP de Jacobs e Brusco.....	106

1. INTRODUÇÃO

Nos últimos anos, importantes atividades de pesquisa vêm ocorrendo no campo das Heurísticas Naturais, uma área que utiliza alguma analogia com sistemas naturais, ou sociais, para derivar métodos de solução não-determinísticos para problemas de otimização combinatoriais complexos (NP-Completo) (COLORNI, A., et al., 1996).

Em meio a este cenário, eis que surge em 1999 a nova meta-heurística de otimização baseada em colônia de formigas (*Ant Colony Optimization - ACO*) (DORIGO, M., CARO, G., 1999a) (DORIGO, M., CARO, G., 1999a) com suas promissoras adaptações em vários problemas combinatoriais: caixeiro viajante (STUTZLE, T.; DORIGO, M., 1999), arranjo seqüencial (GAMBARDELLA, L. M.; DORIGO, M., 1997), atribuição quadrática (MANIEZZO, V.; COLORNI, A.; DORIGO, M., 1999), escalonamento (COLORNI, A., et al. 1994), coloração de grafos (COSTA, D.; HERTZ, A., 1997), roteamento de veículos (BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C., 1999a), entre outros.

A meta-heurística *ACO*, apogeu do trabalho seminal desenvolvido por Dorigo (1992) em sua tese de doutorado, propõe o estudo de sistemas artificiais inspirados no comportamento de colônias de formigas reais (GOSS, S., et al., 1989), com o intuito de solucionar problemas de otimização combinatoriais (DORIGO, M., CARO, G., 1999a). A idéia central desta meta-heurística consiste em simular o processo de depósito e evaporação de uma substância (feromônio) excretada pelas formigas reais, como mecanismo de aprendizagem de um sistema multiagente (no caso, as formigas) proposto a solucionar um determinado problema combinatorial.

Entretanto, incentivados pelos trabalhos de Hooker (1996), Rardin e Uzsoy (2001), McGeoch (1999), Cohen (1995) e Barr (1995), nós constatamos (SILVA, R. M. A.; RAMALHO, G. L., 2001a) que, na maior parte dos artigos publicados pela comunidade *ACO*, a análise experimental

das heurísticas de otimização não seguia um método de avaliação, principalmente no que se refere à carência de estudos sobre a influência dos parâmetros das heurísticas na qualidade dos resultados observados. Geralmente, as análises estavam mais preocupadas em identificar **qual** a melhor heurística para determinado problema de otimização (**enfoque comparativo**), e não **como** esta heurística afinal trabalha (**enfoque causal**).

Uma vez que eventuais descuidos ocorridos na etapa de avaliação de um algoritmo podem levar a conclusões equivocadas a respeito de seu desempenho - criando obstáculos ao desenvolvimento de novos algoritmos com o propósito de melhorar as soluções outrora obtidas – então um estudo mais aprofundado da potencialidade de *ACO* seria bem-vindo.

Sendo assim, o objetivo deste trabalho consiste em utilizar um método de análise experimental para avaliar, segundo os enfoques comparativo e causal, a adaptação da meta-heurística *ACO* em algum problema de otimização combinatorial previamente abordado pela comunidade *ACO*. O propósito não consiste tanto em verificar a validade da hipótese da existência de alguma conclusão imprecisa divulgada por esta comunidade, mas sobretudo lançar conjecturas que expliquem o desempenho deste método e suas propriedades, que por sua vez auxiliarão o desenvolvimento de novos algoritmos com melhor desempenho.

O problema de otimização combinatorial adotado neste trabalho são instâncias de grande porte do problema de cobertura de conjuntos (*Set Covering Problem -SCP*). O interesse em torno deste problema decorre de sua complexidade e aplicabilidade. *SCP* é um problema NP-completo, o que segundo Pacca e Loureiro (2001, p. 479): “...*dificulta sobremaneira a expectativa de um bom desempenho de algoritmos exatos para problemas de grande porte...*”. Comumente modelado via programação linear discreta, *SCP* mostra-se capaz de atender uma grande quantidade de problemas práticos, tais como, a localização de estações rádio-base em um sistema celular de telecomunicações, a localização de maquinário em uma linha de produção, a localização dos componentes em um circuito integrado, entre outros. Cabe observar que para realizar os experimentos contamos com a base de testes para *SCP* desenvolvida por Beasley (1990a), pois foi a mesma utilizada pelas heurísticas presentes no trabalho de Caprara, Fischetti e Toth (2000).

Devido ao fato da maior parte dos algoritmos *ACO*, desenvolvidos para os mais diversos problemas combinatoriais, serem extensões da heurística *Ant System (AS)*, decidimos não apenas utilizá-la neste estudo da atuação da meta-heurística *ACO* em *SCP*, mas também usar duas de suas extensões: a heurística *MAX-MIN Ant System* desenvolvida por Stützle e Hoss (1998), e a heurística híbrida *ACS-SCP* desenvolvida por Hadji, Rahoual, Talbi e Bachelet (2000). A primeira devido à constatação de que explora de forma mais eficiente o espaço de soluções viáveis das instâncias *SCP* utilizadas nos experimentos. A segunda pelo fato de ser um trabalho correlato sobre a utilização do método baseado em colônia de formigas sobre *SCP*.

Algumas das mais recentes e eficientes heurísticas para *SCP* presentes na literatura (CAPRARA, A.; FISCHETTI, M.; TOTH, P., 2000), mais especificamente o algoritmo genético *GA-SCP* desenvolvido por Beasley e Chu (1996) e o recozimento simulado *SA-SCP* criado por Jacobs e Brusco (1995), também serão utilizados em nosso estudo, assim como também o método *LS-ECC-SCP* de busca local para *SCP* desenvolvido por Jacobs e Brusco (1995). Primeiro porque, dentre as heurísticas que apresentaram as soluções de melhor qualidade (CAPRARA, A.; FISCHETTI, M.; TOTH, P., 2000), *GA-SCP* e *SA-SCP* foram as que abordaram o problema de cobertura de conjuntos pelo modelo original (tal como ocorre nas adaptações *ACO* em *SCP*). Segundo, porque o algoritmo *LS-ECC-SCP* servirá de referência (controle) ao processo de análise das heurísticas avaliadas neste trabalho, pois o uso de novas heurísticas não é justificável se existe outra mais simples com desempenho melhor ou equivalente.

É importante deixar claro que o foco desta tese não é a criação de novos algoritmos baseados na meta-heurística *ACO* para resolver *SCP*, o que seria o caminho mais comum nas teses em computação atuais, mas de fazer uma avaliação do potencial da meta-heurística *ACO* (através das extensões e hibridizações das heurísticas *Ant-System* e *MAXMIN Ant System*) para o problema de cobertura de conjuntos, e de oferecer explicações mais detalhadas sobre a influência dos seus parâmetros.

A principal contribuição deste trabalho, sobretudo com relação à surpresa do seu resultado em vista da literatura vigente (HADJI, R. et al. 2000), encontra-se na revelação da pouca importância do feromônio no método *ACO* (mais especificamente nas variações e hibridizações das heurísticas canônicas *Ant-System* e *MAXMIN Ant System*) em instâncias *SCP* de grande porte, assim como na exposição de teorias, baseadas no conceito da **correlação da distância de adaptação** (a ser visto no Capítulo 7), capazes de explicar não somente as causas responsáveis pela atuação do feromônio, mas também a melhoria oriunda das hibridizações (via busca local) do método *ACO* em *SCP*, a ponto deste último ser prescindível. Ou seja, chegamos à conclusão de que não se justifica a utilização do método *ACO* em instâncias *SCP* de grande porte, uma vez que existem técnicas mais simples e eficientes capazes de tratar este mesmo problema, como por exemplo, a busca local desenvolvida por Jacobs e Brusco (1995). Portanto, ao invés de desperdiçar tempo em estudos sobre *ACO* em *SCP* (mais especificamente com relação às heurísticas *Ant System*, *MAX-MIN Ant System*, e suas hibridizações), é mais recomendável desenvolver outros métodos de otimização e/ou pesquisar em pormenor o comportamento de *ACO* neste tipo de problema.

Como relação às contribuições secundárias desta tese, podemos citar a constatação da viabilidade da técnica de delineamento D-ótimo (Seção 5.4) na construção de projetos experimentais, assim como, as adaptações da heurística *MAX-MIN Ant System* em *SCP* (Seção 4.3 e 4.6), inédita na comunidade *ACO*.

A estrutura do presente trabalho encontra-se organizada da seguinte maneira. O segundo capítulo apresenta o problema de cobertura de conjuntos, assim como o estado da arte relativo às heurísticas que abordam este tipo problema. O terceiro capítulo explana a meta-heurística de otimização baseada em colônia de formigas. O quarto capítulo adapta a meta-heurística *ACO* ao problema de cobertura de conjuntos, tendo por base as heurísticas *Ant-System* e *MAXMIN Ant System*. O quinto capítulo expõe o método de avaliação experimental adotado neste trabalho. Por fim, os resultados da análise experimental são descritos e discutidos no sexto e sétimo capítulo, respectivamente, enquanto as conclusões e futuros trabalhos servem de tema ao oitavo capítulo.

2. O PROBLEMA DE COBERTURA DE CONJUNTOS E SUAS HEURÍSTICAS

O presente capítulo tem por objetivo definir o problema de cobertura de conjuntos (Seção 2.1), determinar sua complexidade (Seção 2.2), e apresentar (Seção 2.3) algumas das mais recentes e eficientes heurísticas da literatura (CAPRARA; FISCHETTI; TOTH, 2000). Dentre as heurísticas que apresentaram as soluções de melhor qualidade¹, este capítulo ressalta as que abordaram o problema de cobertura de conjuntos segundo o mesmo modelo das adaptações de ACO em SCP. No caso, as heurísticas selecionadas foram a desenvolvida por Beasley e Chu (1996) (Seção 2.6), e a criada por Jacobs e Brusco (1995) (Seção 2.5). Ao lado do método da busca local para SCP desenvolvido por Jacobs e Brusco (1995)² (Seção 2.4), estas duas heurísticas formam o conjunto de algoritmos a serem comparados com a melhor adaptação da heurística Ant System (AS) em SCP.

2.1 DEFINIÇÃO DO PROBLEMA DE COBERTURA DE CONJUNTOS

Dado um conjunto de localidades e facilidades (respectivamente representadas por O e × na Figura 1.a), suponha que cada facilidade cubra (alcance) um determinado subconjunto de localidades a um custo específico (a cobertura de três facilidades encontra-se destacada na Figura

¹ O critério de qualidade está formalmente definido na seção 6.1.1, onde quanto mais próximo a solução construída pela heurística estiver do ótimo global do problema, melhor a sua qualidade.

² A busca local por melhoria para SCP de Jacobs e Brusco (1995) atuará como **método de controle**, ou seja, seu desempenho servirá de referência para os resultados das outras heurísticas. Afinal de contas, não se justifica o uso de uma nova heurística, se existe outra mais simples com desempenho melhor ou equivalente.

1.b). O **problema de cobertura de conjuntos** (*SCP*) consiste em selecionar um subconjunto de facilidades capaz de cobrir todas as localidades com o menor custo possível.

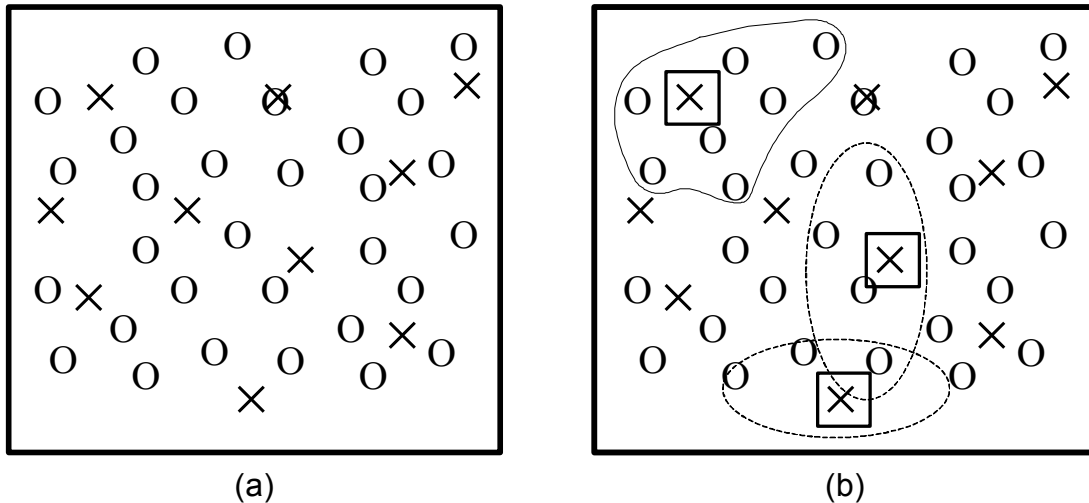


Figura 1: Instância genérica do problema de cobertura de conjuntos

Ou seja, dado um conjunto finito X de localidades e uma família F de subconjuntos de X , tal que $X = \bigcup_{S \in F} S$, o problema de cobertura de conjuntos consiste em encontrar um subconjunto de custo mínimo $C \subseteq F$, tal que $X = \bigcup_{S \in C} S$. Cabe observar que cada elemento $S \in F$ corresponde ao subconjunto de localidades coberta por alguma facilidade.

O problema de cobertura de conjuntos pode também ser definido da seguinte maneira. Seja $A = [a_{ij}]$ uma matriz $m \times n$ binária (0-1), $c = [c_j]$ um vetor real n -dimensional, $I = \{1, \dots, m\}$ o conjunto de linhas (localidades), e $J = \{1, \dots, n\}$ o conjunto de colunas (facilidades).

Considere que a coluna (facilidade) $j \in J$ cobre uma linha (localidade) $i \in I$ se $a_{ij} = 1$ (caso contrário $a_{ij} = 0$), e que o valor de $c_j > 0$, $j \in J$, representa o custo da coluna (facilidade) j .

O problema da cobertura de conjuntos consiste em encontrar um subconjunto $S \subseteq J$ de custo mínimo, tal que cada linha (localidade) $i \in I$ seja coberta por pelo menos uma coluna (facilidade) $j \in S$.

Sendo assim, SCP pode ser modelada da seguinte maneira:

$$\text{minimizar } \sum_{j \in J} c_j x_j \quad (2.1)$$

$$\text{sujeito a : } \sum_{j \in J} a_{ij} x_j \geq 1, \quad i \in I \quad (2.2)$$

$$x_j \in \{0,1\}, \quad j \in J \quad (2.3)$$

onde $x_j = 1$ se $j \in S$, e $x_j = 0$ caso contrário.

Em suma, a restrição (2.2) garante a cobertura de uma linha (localidade) por pelo menos uma coluna (facilidade), enquanto a restrição (2.3) define os valores permitidos às variáveis x_j .

A relevância do problema de cobertura de conjuntos está em sua capacidade para modelar problemas práticos reais presentes em vários empreendimentos públicos ou privados: “... *In short, the success or failure of both private and public sector facilities depends in part on the locations chosen for those facilities ...*” (DASKIN, 1995, p. 1). Pois, visto que o termo “facilidade” significa construções, equipamentos e serviços criados para um determinado propósito (CAMBRIDGE, 1995), então podemos considerar tais facilidades como sendo, por exemplo, hospitais (TOREGAS, et al., 1971), supermercados, fábricas, escolas, bibliotecas, corpo de bombeiros (WALKER, 1974), usinas de energia (KUBY, et al., 1993), postos de combustíveis, pontos de transporte urbano (GLEASON, 1975), componentes de um circuito integrado (FRANCIS; WHITE, 1974), máquinas de uma linha de produção (VERTER; DINCER, 1995), estabelecimentos comerciais, delegacias, empresas de limpeza urbana, estações rádio-base de um sistema de telecomunicações celular (MATEUS; LOUREIRO, 1998), entre outros. Para

maiores detalhes sobre problemas práticos modelados como SCP, consultar os trabalhos de: Schilling, Jayraman e Barkhi (1993); Balas (1983), e Ceria, Nobili e Sassano (1997).

2.2 COMPLEXIDADE DO PROBLEMA DE COBERTURA DE CONJUNTOS

O problema de cobertura de conjuntos é NP-Completo. Porém, antes de demonstrar a prova, tomemos inicialmente as seguintes considerações (CORMEN et al, 2001):

- O caráter NP-Completo não se aplica diretamente a problemas de otimização, mas a problemas de decisão, em que a resposta é simplesmente “sim” ou “não”. Entretanto, um problema de otimização pode ser formulado como um problema de decisão relacionado. Para isto, um valor k é imposto sobre o limite da função objetivo a ser otimizada. Por exemplo, no caso de SCP, o problema de decisão relacionado (SCP-DECISION) consiste em afirmar (sim ou não) se uma solução para SCP é **viável** e tem custo menor do que k .
- Uma solução é viável se satisfaz todas as restrições do problema relacionado. Por exemplo, no caso de SCP, uma solução é viável se satisfaz as restrições (2.2) e a função objetivo (2.1).
- Um dos aspectos convenientes de se concentrar em problemas de decisão é que eles tornam fácil o uso do mecanismo da teoria de linguagem formal. Nesse momento, vale a pena rever algumas definições dessa teoria. Um **alfabeto** Σ é um conjunto finito de símbolos. Uma **linguagem** L sobre Σ é qualquer conjunto de cadeias formadas por símbolos de Σ . Por exemplo, se $\Sigma = \{0,1\}$, o conjunto $L = \{10, 11, 101, 111, 1011, 1101, 10001, \dots\}$ é a linguagem de representações binárias de números primos. Denota-se a cadeia **vazia** por ε , e a **linguagem vazia** por \emptyset . A linguagem de todas as cadeias sobre

Σ é denotada por Σ^* . Por exemplo, se $\Sigma = \{0,1\}$, então $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$ é o conjunto de todas as cadeias binárias. Toda linguagem L sobre Σ é um subconjunto de Σ^* .

- Do ponto de vista da teoria de linguagem formal, o conjunto de instâncias para qualquer problema de decisão Q consiste simplesmente no conjunto Σ^* de todas as cadeias sobre o alfabeto $\Sigma = \{0,1\}$ ³. Tendo em vista que Q é completamente caracterizado pelas instâncias de problema que produzem uma resposta 1 (sim), podemos ver Q como uma linguagem L sobre $\Sigma = \{0,1\}$, onde $L = \{x \in \Sigma^* \mid Q(x) = 1\}$. Por exemplo, o problema de decisão SCP-DECISION tem a seguinte linguagem correspondente:

SCP-DECISION = $\{\langle X, F, k \rangle \mid X \text{ é um conjunto finito; } F \text{ é uma família de subconjuntos de } X, \text{ onde } \bigcup_{S \in F} S; k \geq 0 \text{ é um inteiro, e existe um subconjunto } C \subseteq F \text{ de tamanho mínimo}^4 \text{ cujos membros cobrem todo } X \}$.

- Um algoritmo A **aceita** uma cadeia $x \in \{0,1\}^*$ se, dada a entrada x , a saída do algoritmo é $A(x) = 1$. Um algoritmo A **rejeita** uma cadeia $x \in \{0,1\}^*$ se, dada a entrada x , a saída do algoritmo é $A(x) = 0$.
- A linguagem **aceita** por um algoritmo A é o conjunto $L = \{x \in \{0,1\}^* \mid A(x) = 1\}$, isto é o conjunto de cadeias que o algoritmo aceita.

³ A codificação binária é utilizada para representar instâncias do problema abordado por um programa de computador. Caracteres, polígonos, grafos, funções, pares ordenados, programas - todos podem ser codificados como cadeias binárias.

⁴ À título de simplificação estamos considerando $c_j = 1, j \in J$. Neste caso, o custo de uma solução corresponde ao tamanho do subconjunto C , denotado por $|C|$, e calculado pelo número de elementos que possui.

- Uma linguagem L é **decidida** por um algoritmo A , se toda cadeia binária em L é aceita por A e toda cadeia binária não pertencente a L é rejeitada por A .
- Uma linguagem L é **aceita em tempo polinomial** por um algoritmo A se ela é aceita por A e se, além disso, existe uma constante k tal que, para qualquer cadeia $x \in L$ de comprimento n , o algoritmo A aceita x no tempo $O(n^k)$.
- Uma linguagem L é **decidida em tempo polinomial** por um algoritmo A se existe uma constante k tal que, para qualquer cadeia $x \in \{0,1\}^*$ de comprimento n , o algoritmo decide corretamente se $x \in L$ no tempo $O(n^k)$.
- $P = \{L_1 \subseteq \{0,1\}^* \mid \text{existe um algoritmo } A \text{ que decide } L \text{ em tempo polinomial}\}$ constitui a **classe de complexidade** P .
- Um **algoritmo de verificação** A é um algoritmo de dois argumentos: uma cadeia de entrada comum x ; uma cadeia binária y chamada **certificado**.
- Um algoritmo A de dois argumentos **verifica** uma cadeia de entrada x se existe um certificado y tal que $A(x, y) = 1$.
- A **linguagem verificada** por um algoritmo de verificação A é definida por $L = \{x \in \{0,1\}^* \mid \text{existe } y \in \{0,1\}^* \text{ tal que } A(x, y) = 1\}$.
- Uma linguagem L pertence a classe NP se e somente se existe um algoritmo de tempo polinomial de duas entradas A e uma constante c tal que

$$L = \{x \in \{0,1\}^* \mid \text{existe um certificado } y \text{ com } |y| = O(|x|^c) \text{ tal que } A(x, y) = 1\}$$
- Dizemos que uma linguagem L_1 é redutível em tempo polinomial a uma linguagem L_2 , o que se escreve como $L_1 \leq_p L_2$, se existe uma função calculável de tempo polinomial

$f : \{0,1\}^* \rightarrow \{0,1\}^*$ tal que, para todo $x \in \{0,1\}^*$, $x \in L_1 \Leftrightarrow f(x) \in L_2$. Chamamos a função f de **função de redução**, e um algoritmo de tempo polinomial F que calcula f é chamado **algoritmo de redução**.

- Uma linguagem $L \subseteq \{0,1\}^*$ é NP-Completa (NPC) se: (a) $L \in NP$, e (b) $L' \leq_p L$ para todo $L' \in NP$. Se uma linguagem L satisfaz à propriedade (b), mas não necessariamente a propriedade (a), dizemos que L é NP-difícil.

Após as considerações acima citadas segue abaixo o lema, e sua respectiva prova, fundamental à demonstração de que uma linguagem é NPC (CORMEN et al., 2001).

LEMA 2.1

Se L é uma linguagem tal que $L' \leq_p L$ para alguma $L' \in NPC$, então L é NP-difícil. Além disso, se $L \in NP$, então $L \in NPC$.

PROVA

Tendo em vista que L' é NP-Completa, para todo $L'' \in NP$, temos $L'' \leq_p L'$. Por hipótese, $L' \leq_p L$ e, desse modo, por transitividade, temos $L'' \leq_p L$, o que mostra que L é NP-difícil. Se $L \in NP$, também temos $L \in NPC$.

Em outras palavras, reduzindo a L uma linguagem NP-Completa L' conhecida, reduzimos implicitamente toda linguagem em NP a L . Assim, o Lema 2.1 apresenta um método para provar que uma linguagem L é NP-Completa (CORMEN, et al., 2001):

1. Provar que $L \in NP$.
2. Selecionar uma linguagem NP-Completa conhecida L' .

3. Descrever um algoritmo que calcule uma função f que mapeie toda instância $x \in \{0,1\}^*$ de L' para uma instância $f(x)$ de L .
4. Provar que a função f satisfaz a $x \in L'$ se e somente se $f(x) \in L$ para todo $x \in \{0,1\}^*$.
5. Provar que o algoritmo que calcula f é executado em tempo polinomial.

Uma vez apresentados os pré-requisitos necessários à demonstração de que uma linguagem L é NP-Completa (considerações, lema e método), agora chega a vez de provar que $SCP \in NPC$, usando-se para isto a definição de SCP baseada na teoria de conjuntos (Seção 2.1, 1º. Parágrafo, p. 21). Desta maneira, seguindo as etapas do método acima citado, temos:

1. Dado C , conferir se todos os elementos de X pertencem a algum conjunto $S \in C$, e se $|C| \leq k$. O algoritmo que realiza esta tarefa precisa apenas percorrer as listas que representam os subconjuntos de S , para cada elemento da lista que representa o conjunto X . Sendo assim, a complexidade é de $\Theta(N)$, com $N = \sum_{S \in C} |S|$.
2. $L' = VCP$ -DECISION é uma linguagem NP-Completa (CORMEN, et. al, 2001). O acrônimo VCP representa o **problema de cobertura de vértices**. Uma **cobertura de vértices** de um grafo não orientado $G = (V, E)$ é um subconjunto $V' \subseteq V$ tal que, se (u, v) é uma aresta de G , então $u \in V'$ ou $v \in V'$ (ou ambos). O problema de cobertura de vértices consiste em encontrar uma cobertura de vértices de tamanho mínimo em um dado grafo não-orientado, onde o tamanho $|V'|$ é o número de vértices contidos em V' . O problema de decisão VPC -DECISION, relacionado ao problema de otimização VCP , tem a seguinte linguagem correspondente:

VCP-DECISION = $\{ \langle G, k \rangle \mid G = (V, E) \text{ é um grafo não-orientado; } k \geq 0 \text{ é um inteiro, e existe um subconjunto } V' \subseteq V \text{ tal que } |V'| \leq k, \text{ se } \forall (u, v) \in E, \text{ ou } u \in V' \text{ ou } v \in V' \text{ (ou ambos)} \}$.

3. Dado $\langle G(V, E), k \rangle \in \text{VCP-DECISION}$, definir F tal que cada elemento de F é um subconjunto associado a cada vértice $u \in V$, que contém u e todos os vértices v onde $(u, v) \in E$. Ou seja, $F = \{ \{u\} \cup \{v \mid (u, v) \in E\} \}$, para todo $u \in V$. Considerando $X = V$, então $\langle X, F, k \rangle \in \text{SCP-DECISION}$.
4. Se C é uma cobertura de vértices de $\langle G(V, E), k \rangle \in \text{VCP-DECISION}$, então os subconjuntos $S \in F$ correspondentes aos vértices em C , juntos compõem a cobertura de conjuntos de $\langle X, F, k \rangle \in \text{SCP-DECISION}$. Caso contrário, existiria algum vértice $u \in V$ que não estaria contido em nenhum destes subconjuntos $S \in F$. Contradizendo portanto a hipótese de que C é uma cobertura de vértices de $\langle G(V, E), k \rangle \in \text{VCP-DECISION}$, onde todo vértice $u \in V$ é incidente a uma aresta $(u, v) \in E$, com $u \in C$ ou $v \in C$.
5. O algoritmo que constrói F precisa apenas percorrer as listas de adjacências correspondentes a cada vértice de V do grafo $G(V, E)$. A representação de lista de adjacências de um grafo $G(V, E)$ consiste em um arranjo $Adj.$ de $|V|$ listas, uma para cada vértice de V . Para cada $u \in V$, a lista de adjacências $Adj[u]$ contém (ponteiros para) todos os vértices v tais que existe uma aresta $(u, v) \in E$. Portanto, se $G(V, E)$ é um grafo não orientado, a soma dos comprimentos de todas as listas de adjacências é $2|E|$ pois, se (u, v) é uma aresta não orientada, então u aparece na lista de adjacências de v , e vice-versa. Sendo assim, a complexidade em percorrer todas estas listas é de $\Theta(V + E)$.

2.3 HEURÍSTICAS PARA O PROBLEMA DE COBERTURA DE CONJUNTOS

Baseada no trabalho do Caprara, Fischetti, e Toth (2000), esta seção apresenta algumas das mais recentes e eficientes heurísticas da literatura, para resolução de instâncias de grande porte do problema de cobertura de conjuntos (Tabela 1).

Embora a maioria das técnicas de otimização presentes na Tabela 1 recorram às **heurísticas lagrangeanas** (Seção 2.3.3) para solucionar o **modelo dual da relaxação lagrangeana** (RARDIN, 1998) (Seção 2.3.2) de SCP (2.1) – (2.3), há outros métodos de otimização eficientes, tais como, o algoritmo genético de Beasley e Chu (1996) (Seção 2.6) e o recozimento simulado de Jacobs e Brusco (1995) (Seção 2.5), que abordam diretamente o modelo primal de SCP (2.1) – (2.3).

Tabela 1: Heurísticas para o Problema de Cobertura de Conjuntos.

Nome da Heurística	Modelo SCP	Técnica(s) de Otimização	Referências
Be	Modelo Dual da Relaxação Lagrangeana	Heurística Lagrangeana	Beasley J. E. (1990b)
LL	Relaxação Surrogate	Heurística Lagrangeana	Lorena L. A. N., e Lopes F. B. (1994)
CFT	Modelo Dual da Relaxação Lagrangeana	Heurística Lagrangeana	Caprara A., Fischetti M., e Toth P. (1995)
CNS	Extensão do Modelo Dual da Relaxação Linear	Heurística Lagrangeana	Ceria S., Nobili P., e Sassano A. (1995)
JaBr	Modelo Primal	Recozimento Simulado	Jacobs L. W., e Brusco M.J. (1995)
BaCa	Modelo Dual da Relaxação Lagrangeana	Heurística Lagrangeana	Balas E., e Carrera M. C. (1996)
BeCh	Modelo Primal	Algoritmo Genético	Beasley J. E., e Chu P. C. (1996)

Ha	Modelo Dual da Relaxação Lagrangiana	Heurística Lagrangeana	Haddadi S. (1997)
----	--	------------------------	-------------------

A Tabela 2 apresenta os resultados das heurísticas da Tabela 1, aplicadas em instâncias (coluna “Nome”) SCP das classes A, B, C e D (Tabela 4) da biblioteca de testes desenvolvida por Beasley (1990a), cujas soluções ótimas são previamente conhecidas (coluna “Ot.”). Para os algoritmos de Beasley (1990b) (coluna “Be”), Lorena e Lopes (1994) (“LL”), Balas e Carrera (1996) (“BaCa”), Beasley e Chu (1996) (“BeCh”), Haddadi (1997) (“Ha”) e Caprara, Fischetti e Toth (1995) (“CFT”), a Tabela 2 exibe os valores das melhores soluções encontradas (coluna “Sol.”), assim como o tempo gasto para encontrá-las (coluna “Tp.”).

Tabela 2: Heurísticas nas instâncias das classes A, B, C e D.

Problema		Be		LL		Baca		BeCh		Ha		CFT	
Nome	Ot.	Sol.	Tp.	Sol.	Tempo	Sol.	Tp.*	Sol.	Tp.*	Sol.	Tp.	Sol.	Tp.
A.1	253	255	36.0	254	2.7	258	39.0	253	222.4	254	22.1	253	82.0
A.2	252	256	44.2	255	2.9	254	40.9	252	327.9	253	20.2	252	116.2
A.3	232	234	28.1	234	2.6	237	28.6	232	127.0	234	21.7	232	249.9
A.4	234	235	33.5	234	2.4	235	36.3	234	45.5	234	16.8	234	4.7
A.5	236	237	19.0	238	2.2	236	26.2	236	23.7	236	17.4	236	80.0
B.1	69	70	28.4	70	3.0	69	29.0	69	20.0	69	26.4	69	4.0
B.2	76	77	40.8	76	4.0	76	29.0	76	11.6	76	28.3	76	6.1
B.3	80	80	25.4	81	4.4	81	35.1	80	709.7	81	28.0	80	18.0
B.4	79	80	37.0	81	4.3	79	29.0	79	29.9	79	29.7	79	6.3
B.5	72	72	26.0	72	4.1	72	32.6	72	5.3	72	26.6	72	3.3
C.1	227	230	42.4	227	4.0	230	116.2	227	187.9	228	32.7	227	74.0
C.2	219	223	66.0	222	4.1	220	56.1	219	40.7	223	34.0	219	64.2
C.3	243	251	75.1	251	4.9	248	61.7	243	541.3	245	35.9	243	70.2
C.4	219	224	63.4	224	5.4	224	68.1	219	144.6	223	32.8	219	61.6
C.5	215	217	39.9	216	4.1	217	64.6	215	80.6	216	30.2	215	60.3
D.1	60	61	40.9	60	4.8	61	36.6	60	13.8	61	45.9	60	23.1
D.2	66	68	52.7	68	3.5	67	46.6	66	198.6	66	49.3	66	22.0
D.3	72	75	55.8	75	5.8	74	47.2	72	785.3	73	48.6	72	22.6
D.4	62	64	36.5	63	4.8	63	39.8	62	73.5	62	48.0	62	8.3
D.5	61	62	36.7	62	4.5	61	36.2	61	79.8	62	42.5	61	10.3

Ot = solução ótima; Sol. = melhor solução gerada pela heurística; Tp = tempo gasto pela heurística para encontrar a solução Sol; Tp.* Tempo total de execução da heurística

Fonte: CAPRARA, A.; FISCHETTI, M.; TOTH, 2000.

Segundo a Tabela 2, as heurísticas BeCh e CFT geram a solução ótima de **todas** as instâncias das classes A, B, C e D (Tabela 4) da biblioteca de testes desenvolvida por Beasley (1990a). Entretanto, o tempo médio gasto para descobrir a melhor solução é de 50 segundos para CFT, e 180 seg. para BeCh.⁵

A Tabela 3 apresenta os resultados das heurísticas da Tabela 1, aplicadas em instâncias (coluna “Nome”) SCP das classes E, F, G e H (Tabela 4), cujas melhores soluções são previamente conhecidas (coluna “MS”), assim como, a melhor cota inferior (coluna “CI”).

Tabela 3: Heurísticas nas instâncias das classes E, F, G e H.

Problema		Be		BaCa		CNS		BeCh		JaBr		Ha		CFT		
Nome	MS	CI	Sol	Tp.	Sol	Tp.*	Sol	Tp.°	Sol	Tp.	Sol	Tp.°	Sol	Tp.*	Sol	Tp.
E.1	29	29	29	72.6	29	55.3	X	X	29	38.2	29	408.0	29	187.1	29	26.0
E.2	30	28	32	92.7	32	76.0	X	X	30	14647.7	30	408.0	32	199.8	30	408.0
E.3	27	27	28	92.7	28	80.9	X	X	27	28360.2	27	408.0	28	198.1	27	94.2
E.4	28	28	30	100.3	29	77.5	X	X	28	539.9	28	408.0	29	191.3	28	26.3
E.5	28	28	28	80.8	28	61.6	X	X	28	35.0	28	408.0	28	181.6	28	36.6
F.1	14	14	15	43.9	14	67.5	X	X	14	76.4	14	408.0	14	654.1	14	33.2
F.2	15	15	16	102.6	15	88.6	X	X	15	78.1	15	408.0	15	646.8	15	31.2
F.3	14	14	15	124.7	15	76.5	X	X	14	266.8	14	408.0	15	665.3	14	248.5
F.4	14	14	15	118.2	15	74.8	X	X	14	209.7	14	408.0	15	665.5	14	31.0
F.5	13	13	14	129.3	14	62.2	X	X	13	13192.6	14	408.0	14	684.7	13	201.1
G.1	176	165	184	287.8	183	325.6	176	4905.5	176	30200.0	179	408.0	181	191.3	176	147.0
G.2	154	147	163	204.9	161	370.1	155	4905.5	155	360.5	158	408.0	160	216.5	154	783.4
G.3	166	153	174	318.2	175	378.6	167	4905.5	166	7841.6	170	408.0	174	193.1	166	978.0
G.4	168	154	176	292.0	176	332.2	170	4905.5	168	25304.7	172	408.0	177	184.0	168	378.5
G.5	168	153	175	277.5	172	262.6	169	4905.5	168	549.3	168	408.0	173	192.0	168	237.2
H.1	63	52	68	317.7	68	488.4	64	4905.5	64	1682.1	64	408.0	66	479.9	63	1451.1
H.2	63	52	66	293.9	67	380.7	64	4905.5	64	530.3	64	408.0	67	462.3	63	887.0
H.3	59	48	65	325.1	63	443.1	60	4905.5	59	1803.5	61	408.0	62	462.4	59	1560.3
H.4	58	47	63	333.5	62	354.7	59	4905.5	58	27241.8	59	408.0	61	458.0	58	237.6
H.5	55	46	60	303.0	58	321.3	55	4905.5	55	449.6	55	408.0	56	452.3	55	155.4

⁵ As heurísticas CNS (CERIA; NOBILI; SASSANO, 1995) e JaBr (JACOBS; BRUSCO, 1995) não estão contidas na Tabela 2, porque as respectivas referências não relatam testes sobre estas instâncias.

MS = melhor solução da literatura; CI = melhor cota inferior conhecida; Sol. = melhor solução gerada pela heurística; Tp = tempo gasto para encontrar Sol.; Tp.* Tempo total de execução; Tp.º Limite de Tempo; X = Não há registros na literatura.

Fonte: CAPRARA, A.; FISCHETTI, M.; TOTH, 2000.

Para os algoritmos de Beasley (1990b) (coluna “Be”), Balas e Carrera (1996) (“BaCa”), Ceria, Nobili e Sassano (1995) (“CNS”), Beasley e Chu (1996) (“BeCh”), Jacobs e Brusco (1995), Haddadi (1997) (“Ha”) e Caprara, Fischetti e Toth (1995) (“CFT”), a Tabela 3 exibe os valores das melhores soluções encontradas (coluna “Sol.”), assim como o tempo gasto para encontrá-las (coluna “Tp.”).

De acordo com a Tabela 3, a heurística CFT gera novamente a solução ótima de todas as instâncias, mas agora das classes E, F, G e H (Tabela 4), também pertencentes à biblioteca desenvolvida por Beasley (1990a). O mesmo ocorre em BeCh, exceto em três casos particulares (G.2, H.1 e H.2).

As soluções de CNS e JaBr estão em média um pouco piores do que as apresentadas por CFT e BeCh, porém são bem melhores que as soluções de Be, BaCa e Ha (Tabela 3). Por fim, o tempo médio gasto para descobrir a melhor solução é de 200 segundos para Be e BaCa, 400 segundos para Ha e CFT, e 7500 para BeCh ⁶.

Tabela 4: Fonte das instâncias de dimensões maiores de *SCP*.

Problema	Quantidade de Localidades	Quantidade de Facilidades	Instâncias
A	300	3.000	A.1 a A.5
B	300	3.000	B.1 a B.5
C	400	4.000	C.1 a C.5
D	400	4.000	D.1 a D.5
E	500	5.000	E.1 a E.5
F	500	5.000	F.1 a F.5
G	1000	10.000	G.1 a G.5
H	1000	10.000	H.1 a H.5

⁶ Ceria, Nobili e Sassano (1995) e Jacobs e Brusco (1995) relatam apenas o limite de tempo para execução de seus algoritmos, os quais foram dispostos na coluna Tp da Tabela 3.

Fonte: BEASLEY; CHU, 1996.

2.3.1 RELAXAÇÃO LAGRANGEANA DE SCP E SEU MODELO DUAL

Considerando que um modelo de otimização tenha o seguinte formato geral:

$$\begin{aligned} & \min \text{ ou } \max f(x_1, x_2, \dots, x_n) \\ & \text{sujeito a :} \\ & g_i(x_1, x_2, \dots, x_n) \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b_i \quad i = 1, \dots, m \end{aligned} \quad (2.4)$$

onde **a função objetivo** f e **as restrições** g_1, \dots, g_m são funções definidas sobre **as variáveis de decisão** x_1, x_2, \dots, x_n , enquanto b_1, \dots, b_m são os **parâmetros** do modelo (RARDIN, 1998). Então, se f, g_1, \dots, g_m são **funções lineares** e existe pelo menos uma variável de decisão **discreta**, o modelo é classificado de **programação linear discreta** ou **inteira**. Ou seja, tal como ocorre em SCP (2.1) – (2.3), um modelo de programação linear discreto possui o seguinte formato (RARDIN, 1998):

$$\begin{aligned} & \min \text{ ou } \max \quad \sum_{j=1}^n c_j x_j \\ & \text{sujeito a} \\ & \sum_{j=1}^n a_{ij} x_j \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b_i \quad i = 1, \dots, m \end{aligned} \quad (2.5)$$

onde $\exists x_j$ discreto

onde c_j é o coeficiente da função objetivo associado à variável de decisão x_j , enquanto a_{ij} é o coeficiente de x_j na i -ésima restrição do modelo.

A relaxação lagrangeana relaxa algumas restrições da programação linear discreta (2.5), na medida em que as move para a função objetivo com o seguinte formato (RARDIN, 1998):

$$\dots + u_i \left(b_i - \sum_{j=1}^n a_{ij} x_j \right) \dots$$

onde u_i é um **multiplicador lagrangeano** definido da seguinte maneira. Caso a restrição relaxada tenha a forma $\sum_{j=1}^n a_{ij} x_j \geq b_i$, então $u_i \leq 0$ para maximização, ou $u_i \geq 0$ para minimização. Por outro lado, caso tenha a forma $\sum_{j=1}^n a_{ij} x_j \leq b_i$, então $u_i \geq 0$ para maximização, ou $u_i \leq 0$ para minimização. Por fim, para restrições da forma $\sum_{j=1}^n a_{ij} x_j = b_i$, então u_i está apto a receber qualquer valor, seja para problemas de maximização ou minimização.

Sendo assim, a **relaxação lagrangeana da programação linear do SCP** (2.1) – (2.3) assume o seguinte formato:

$$\begin{aligned} L(u) &= \min \sum_{j \in J} c_j x_j + \sum_{i \in I} u_i \left(1 - \sum_{j \in J} a_{ij} x_j \right) \\ &= \min \sum_{j \in J} c_j(u) x_j + \sum_{i \in I} u_i \end{aligned} \quad (2.6)$$

$$\text{sujeito a : } x_j \in \{0,1\}, \quad j \in J \quad (2.7)$$

onde $u \in R_+^m$ representa o **vetor de multiplicadores lagrangeanos** associado à Equação (2.3), e $c_j(u) = c_j - \sum_{i \in I_j} u_i$, o **custo lagrangiano** associado a coluna $j \in J$, com $I_j = \{i \in I \mid a_{ij} = 1\}$.

Fixado um vetor $u \in R_+^m$, a solução ótima de $L(u)$ (dada por $x_j(u) = 1$, se $c_j(u) < 0$; $x_j(u) = 0$, se $c_j(u) > 0$; e $x_j(u) \in \{0,1\}$, se $c_j(u) = 0$) determina uma **cota inferior** para o modelo original (primal) de SCP (2.1) – (2.3). Primeiro porque um termo não-positivo foi adicionado à função objetivo do modelo primal de SCP. Ou seja, para a mesma solução ótima x_j ($j \in J$), o valor de $L(u)$ há de ser menor ou igual ao valor da função objetivo do modelo primal de SCP (2.1). Segundo porque a remoção das restrições do modelo primal de SCP (2.3) pode apenas melhorar (ou manter) a solução ótima.

O **modelo dual relativo à relaxação langrangeana** (2.6) - (2.7) da **programação linear discreta de SCP** (2.1) - (2.3) consiste em encontrar um vetor de multiplicadores lagrangeanos $u^* \in R_+^m$ que **maximize** a cota inferior $L(u)$:

$$\max L(u) \tag{2.8}$$

a fim de que o valor da função objetivo de $L(u)$ (2.6), aplicada sobre a solução ótima x_j ($j \in J$) construída a partir de $u^* \in R_+^m$, se aproxime do valor ótimo do modelo primal de SCP (2.1) – (2.3).

Entretanto, nosso principal objetivo não consiste em computar a melhor cota inferior, mas sim orientar a busca por soluções para SCP que sejam próximas do valor ótimo. É justamente esta a tarefa das heurísticas lagrangeanas.

2.3.2 HEURÍSTICAS LAGRANGEANAS

Segundo Techapichetvanich e Bricker (1993), as heurísticas lagrangeanas são algoritmos projetados para serem inseridos em alguma técnica de otimização que trate o modelo dual da relaxação langrangeana de SCP (2.8), como por exemplo, o **método do subgradiente** sugerido por Held, Wolfe e Crowder (1974) e suas variações (BALAS; CARRERA, 1996) (CAPRARA;

FISCHETTI; TOTH, 1995) (CERIA; NOBILI; SASSANO, 1995). Dito de outro modo, as heurísticas lagrangeanas utilizam o vetor de multiplicadores lagrangeanos $u^* \in R_+^m$ que maximiza $L(u)$, e/ou a solução ótima x_j ($j \in J$) construída a partir de $u^* \in R_+^m$, para orientar a busca por soluções x'_j ($j \in J$) próximas à solução ótima do modelo primal de SCP.

A maior parte das heurísticas lagrangeanas dispostas na Tabela 1 utilizam o seguinte enfoque guloso (*greedy approach*). Dado que o método do subgradiente gera uma seqüência u^0, u^1, \dots de vetores de multiplicadores lagrangeanos não negativos, sendo u^k o vetor correspondente a k -ésima iteração deste método, então uma seqüência S^0, S^1, \dots de soluções viáveis ao modelo primal de SCP é construída da seguinte maneira. A cada iteração do método do subgradiente, um conjunto S é inicializado com as colunas da solução ótima x_j ($j \in J$) construída a partir do vetor de multiplicadores $u \in R_+^m$ atual. Em seguida, para cada linha i não coberta por S , a coluna com o menor custo original c_j em $J_i = \{j \in J \mid a_{ij} = 1\}$ é incluída em S . Por último, as colunas redundantes são removidas de S . Uma coluna j é redundante, se a solução $S - \{j\}$ ainda for viável ao modelo primal de SCP. Este processo se repete até que uma condição de parada seja satisfeita (número de iterações, qualidade da solução, entre outras). Por fim, a melhor solução até então encontrada é selecionada.

2.4LS-ECC-SCP: A BUSCA LOCAL DE JACOBS E BRUSCO

O objetivo desta seção é apresentar o método de otimização para *SCP* desenvolvido por Jacobs e Brusco (1995) (), cujo desempenho servirá de referência (controle) ao processo de análise das heurísticas estudadas neste trabalho.

Considerando os seguintes parâmetros:

- S_{opt} = solução ótima do problema;

- $Z(S)$ = custo da solução S ;
- TM = tempo de execução máximo;
- t = tempo de execução atual;

A heurística *LS-ECC-SCP* de Jacobs e Brusco (1995) constrói outra solução viável S' , a partir da atual solução viável S (Quadro 1, linha 3). Se o custo da solução S' for menor do que o custo da solução S , ou seja $Z(S') < Z(S)$, então S' torna-se a nova solução atual (Quadro 1, linha 4). Caso contrário, ou seja $Z(S') > Z(S)$, a solução atual S é mantida. O processo é repetido até que o tempo máximo TM pré-estabelecido seja alcançado, ou quando a solução ótima S_{opt} do problema for encontrada (Quadro 1, linha 2).

Quadro 1: Heurística LS-ECC-SCP.

1	CONSTRUIR uma solução inicial S viável
2	enquanto $((t < TM) \wedge (S \neq S_{opt}))$ faça
3	$S' \leftarrow$ BUSCAR (S);
4	se $(Z(S') < Z(S))$ então $S \leftarrow S'$;
5	retorne a solução S ;

Fonte: (JACOBS; BRUSCO, 1995).

A seguir os módulos **CONSTRUIR** e **BUSCAR**, presentes na heurística *LS-ECC-SCP* (Quadro 1), serão explanados em detalhes na seções 2.4.1 e 2.4.2, respectivamente.

2.4.1 MÓDULO CONSTRUIR

A partir dos seguintes parâmetros:

- I = conjunto de localidades (linhas);

- $J =$ conjunto de facilidades (colunas);
- $c_j =$ custo da facilidade j , $j \in J$;
- $R =$ sequência das facilidades, em **ordem natural**⁷, não contidas na solução S ;
- $w_i =$ a quantidade de facilidades que cobrem a localidade i , $i \in I$;
- $U =$ o conjunto das localidades descobertas (ou seja, onde $w_i = 0$);
- $a_{ij} = 1$, se a facilidade $j \in J$ cobre a localidade $i \in I$ (caso contrário $a_{ij} = 0$).

O módulo CONSTRUIR (Quadro 1, linha 1) é definido pelos seguintes passos:

- 1º. Inicializar R , $S = \emptyset$, $U = I$, e $w_i \leftarrow 0$, para todo $i \in I$;
- 2º. Selecionar aleatoriamente uma localidade i , $i \in U$;
- 3º. Selecionar a primeira facilidade j , $j \in R$, tal que $a_{ij} = 1$. Em seguida, mover a facilidade j de R para S ;
- 4º. Atualizar $w_i \leftarrow w_i + a_{ij}$ para todo $i \in I$. Atualizar U retirando as localidades com $w_i \neq 0$. Se $U = \emptyset$, avançar ao 5º passo, caso contrário retornar ao 2º passo;
- 5º. Examinar cada coluna k , $k \in S$, em ordem reversa (inversa a ordem natural). Se $w_i - a_{ik} \geq 1$ para todo $i \in I$, então mover k de S para R e atualizar $w_i \leftarrow w_i - a_{ik}$ para todo $i \in I$.

⁷ Ordem natural representa a disposição das facilidades em ordem crescente de custo c_j . Entretanto quando há mais de uma facilidade com mesmo custo, então a ordem destas facilidades é decrescente segundo o número de localidades que cobrem (JACOBS; BRUSCO, 1995).

Ou seja, a partir da escolha de uma localidade $i \in U$ (passos 1° - 2°), a primeira facilidade (na ordem natural) capaz de cobri-la será então adicionada à solução do problema (3° passo). Em seguida, o 4° passo não somente atualiza a cobertura das localidades, a fim de refletir a inclusão de j em S , como também redefine U apropriadamente. Por fim, se todas as localidades estiverem cobertas, as colunas redundantes serão eliminadas da solução S (5° passo).

2.4.2 MÓDULO BUSCAR

A partir dos seguintes parâmetros:

- I = conjunto de localidades (linhas);
- J = conjunto de facilidades (colunas);
- c_j = custo da facilidade j , $j \in J$;
- w_i = a quantidade de facilidades que cobrem a localidade i , $i \in I$;
- U = o conjunto das localidades descobertas (ou seja, onde $w_i = 0$);
- $N(S)$ = a quantidade de facilidades na solução S ;
- $Q(S) = \max (c_j \mid j \in S)$, ou seja, o máximo custo dentre as facilidades da solução S ;
- $D = \lceil \rho_1 * N(S) \rceil$ representa a quantidade de facilidades, dada como uma porcentagem ρ_1 de $N(S)$, a ser removida da solução S em cada iteração da heurística;
- d = contador da quantidade atual de facilidades removidas da solução S ;
- R = sequência das facilidades, em ordem natural, **não** contidas na solução S ;

- $E = \lceil \rho_2 * Q(S) \rceil$ representa o custo, dado como uma percentagem ρ_2 de $Q(S)$, usado como critério de seleção das facilidades candidatas à inclusão na solução S ;
- $R_E =$ conjunto das facilidades $j \in R$ em que $c_j \leq E$.
- $a_{ij} = 1$, se a facilidade $j \in J$ cobre a localidade $i \in I$ (caso contrário $a_{ij} = 0$);
- $\gamma_{ij} = 1$, se a facilidade $j \in R_E$ cobre a localidade $i \in U$ que ainda não foi coberta pela solução atual S (caso contrário $\gamma_{ij} = 0$);
- $\gamma_j = \sum_{i \in I} \gamma_{ij}$, quantidade de localidades $i \in U$ cobertas pela facilidade $j \in R_E$;
- $\xi_j = c_j / \gamma_j$, razão entre o custo da facilidade $j \in R_E$ e a quantidade de localidades $i \in U$ que cobre;
- $\xi_{\min} = \min_j (\xi_j \mid j \in R_E)$, menor razão existente entre os custos das facilidades $j \in R_E$ e quantidade respectiva de localidades $i \in U$ que cobre;
- $K =$ conjunto de facilidades tal que $\xi_j = \xi_{\min}$, $j \in R_E$.

O módulo BUSCAR (Quadro 1, linha 3) é definido pelos seguintes passos:

- 1º. Inicializar R ; $d \leftarrow 0$, $D \leftarrow \lceil \rho_1 * N(S) \rceil$ e $E \leftarrow \lceil \rho_2 * Q(S) \rceil$;
- 2º. Selecionar aleatoriamente uma facilidade k , $k \in S$;
- 3º. Transferir k de S para R . Atualizar $w_i \leftarrow w_i - a_{ik}$ para todo $i \in I$. Incrementar $d \leftarrow d + 1$. Se $d = D$, avançar para o 4º passo, caso contrário, retornar ao 2º passo;
- 4º. Construir U . Se $U = \emptyset$, então avançar para o 7º passo, caso contrário ir para o 5º passo;

5°. Construir R_E a fim de iniciar a montagem do conjunto K através dos seguintes passos:

a. $\gamma_{ij} \leftarrow 1$, se $w_i = 0$ e $a_{ij} = 1$ para todo $i \in I$, $j \in R_E$; caso contrário, $\gamma_{ij} \leftarrow 0$;

b. $\gamma_j \leftarrow \sum_{i \in I} \gamma_{ij}$, para todo $j \in R_E$;

c. $\xi_j \leftarrow c_j / \gamma_j$, para todo $j \in R_E$;

d. $\xi_{\min} \leftarrow \min_j (\xi_j \mid j \in R_E)$;

e. $K =$ conjunto de facilidades tal que $\xi_j = \xi_{\min}$, $j \in R_E$;

6°. Selecionar aleatoriamente uma facilidade $k \in K$ e movê-la de R para S . Atualizar $w_i \leftarrow w_i + a_{ik}$ para todo $i \in I$, e retornar ao 4° passo;

7°. Examinar cada coluna k , $k \in S$, em ordem reversa (inversa a ordem natural). Se $w_i - a_{ik} \geq 1$ para todo $i \in I$, então mover k de S para R e atualizar $w_i \leftarrow w_i - a_{ik}$ para todo $i \in I$.

O processo de busca começa pela retirada de $D = \lceil \rho_1 * N(S) \rceil$ facilidades da solução, ajustando em seguida as informações de cobertura apropriadamente (passos 1°-4°). Depois, facilidades com custo $c_j \leq \lceil \rho_2 * Q(S) \rceil$ são paulatinamente adicionadas à solução S , a fim de resgatar a viabilidade outrora perdida (passos 5°-6°). No caso, o critério de escolha da inclusão está baseado na razão do custo da facilidade pelo número de linhas descobertas que alcança, $\xi_j = c_j / \gamma_j$. Por fim, o processo de eliminação das colunas redundantes é executado (7° passo).

2.5SA-SCP: O RECOZIMENTO SIMULADO DE JACOBS E BRUSCO

Esta seção apresenta o método de recozimento simulado (*simulated annealing*) para SCP desenvolvido por Jacobs e Brusco (1995) (Quadro 2). Considerando os seguintes parâmetros:

- S_{opt} = solução ótima do problema; S^* = melhor solução até então encontrada;
- $Z(S)$ = custo da solução S ; TM = tempo de execução máximo; T = temperatura;
- t = tempo de execução atual; NI = número de iterações em um determinado valor de T ;
- f = fator de resfriamento, ou seja, fator de redução de T após NI iterações;

A heurística SA-SCP de Jacobs e Brusco (1995) cria outra solução viável S' , a partir da solução viável atual S (Quadro 2, linha 4). Se $\delta < 0$, então S' torna-se a solução atual (Quadro 2, linha 5). Caso contrário, ou seja $\delta > 0$, a probabilidade de S' tornar-se a solução atual é de $e^{-\delta/T}$.

O parâmetro $\delta = Z(S') - Z(S)$ representa a diferença de custo entre as soluções S' e S , enquanto T é a temperatura atual usada para controlar a aleatoriedade da busca. Sendo assim, quando T é grande, o expoente $-\delta/T$ tende a zero, implicando que a probabilidade de aceitar soluções viáveis, mas que não melhoram o custo, aproxima-se de 100% ($e^0 = 1$). Por outro lado, quando T é pequeno, a probabilidade de aceitar estas soluções decresce dramaticamente.

Quadro 2: Heurística SA-SCP de Jacobs e Brusco.

```
1 CONSTRUIR uma solução inicial  $S$  viável;  $S^* \leftarrow S$ ;  
2 enquanto  $((t < TM) \wedge (S \neq S_{opt}))$  faça  
3     para  $i = 1, \dots, NI$  faça  
4          $S' \leftarrow$  BUSCAR( $S$ );  $\delta \leftarrow Z(S') - Z(S)$ ;  
5         se  $(\delta < 0)$  então  $S \leftarrow S'$ ;  $S^* \leftarrow S'$ ;  
6         senão  $S \leftarrow S'$ , com probabilidade  $e^{-\delta/T}$ ;  
7     fim-do-para  
8      $T \leftarrow f * T$ ;  
9 fim-do-enquanto  
10 retornar a solução  $S^*$ ;
```


Fonte: (JACOBS; BRUSCO, 1995).

A heurística *SA-SCP* é repetida até que o tempo máximo TM pré-estabelecido seja alcançado, ou quando a solução ótima S_{opt} do problema for encontrada (Quadro 2, linha 2). Por fim, a solução de melhor custo até então encontrada será o ótimo aproximado (Quadro 2, linha 10).

Os módulos CONSTRUIR (Quadro 2, linha 6) e BUSCAR (Quadro 2, linha 9), presentes são os mesmos definidos nas seções 2.4.1 e 2.4.2, respectivamente.

Em suma, o recozimento simulado é uma extensão da busca local (Seção 2.4), que se caracteriza por também aceitar, mas com probabilidade $e^{-\delta/T}$, soluções S' que embora viáveis não melhoram o custo da solução anterior S .

2.6 GA-SCP: O ALGORITMO GENÉTICO DE BEASLEY E CHU

O algoritmo genético para *SCP* de Beasley e Chu (1996) possui os seguintes passos:

- 1°. Gerar uma população inicial com N soluções aleatórias (Seção 2.6.1). Em seguida, inicializar o índice de iteração para $t \leftarrow 0$;
- 2°. Selecionar duas soluções P_1 e P_2 pertencentes à população utilizando o processo de **seleção por torneio** (Seção 2.6.2);
- 3°. Combinar P_1 e P_2 para formar a nova solução C através do operador de **reprodução por fusão** (Seção 2.6.3);
- 4°. Mutar k facilidades da solução C , escolhidas aleatoriamente (Seção 2.6.4);
- 5°. Tornar C viável e, caso exista, remover suas facilidades redundantes (Seção 2.6.5). Incrementar $t \leftarrow t + 1$;

6°. Escolher uma solução com capacidade de adaptação inferior à capacidade média da população e substituí-la por C (Seção 2.6.6);

Se a solução ótima S_{opt} do problema não foi encontrada, e o tempo atual de execução for menor que o tempo máximo TM pré-estabelecido, então retornar ao 2°. passo. Caso contrário, **parar** e retornar a melhor solução obtida.

2.6.1 POPULAÇÃO INICIAL

A heurística de Beasley e Chu (1996) representa uma solução S para SCP através de um indivíduo formado por uma *string* binária de n - bits, sendo n o número de facilidades do problema. Caso o i -ésimo *bit* seja igual a um (1), então a facilidade i faz parte desta solução. Caso contrário, esta facilidade não foi selecionada (Figura 2).

colunas (genes)	1	2	3	4	5	...	n-1	n
<i>string</i> de bits	1	0	1	1	0	...	1	0

Figura 2: Representação binária de um indivíduo

Desta maneira, a partir dos seguintes parâmetros:

- I = conjunto de todas as localidades;
- J = conjunto de todas as facilidades;
- $J_i = \{j \in J \mid a_{ij} = 1\}$ conjunto de facilidades que cobrem a localidade $i, i \in I$;
- $I_j = \{i \in I \mid a_{ij} = 1\}$, conjunto de localidades cobertas pela facilidade $j, j \in J$;
- S = conjunto de facilidades em uma solução;
- U = conjunto de localidades não cobertas;

- w_i = número de facilidades em S que cobrem a localidade $i, i \in I$.

O processo de construção da população inicial da heurística GA-SCP é constituído pelos seguintes passos:

- 1º. Inicializar $n \leftarrow 1$;
- 2º. inicializar $S = \emptyset$; $w_i \leftarrow 0 (\forall i \in I)$; e o indivíduo $P_n[j] \leftarrow 0 (\forall j \in J)$;
- 3º. para cada localidade $i, i \in I$
 - a. aleatoriamente selecionar uma facilidade j em J_i ;
 - b. adicionar j a S e atualizar $w_i \leftarrow w_i + 1 (\forall i \in I_j)$;
- 4º. atualizar $Q \leftarrow S$;
- 5º. aleatoriamente subtrair alguma facilidade j do conjunto Q . Se $w_i \geq 2 (\forall i \in I_j)$, remover esta facilidade j do conjunto S e atualizar $w_i \leftarrow w_i - 1 (\forall i \in \beta_j)$;
- 6º. repetir o 5º passo até $Q = \emptyset$;
- 7º. $P_n[j] \leftarrow 1 (\forall j \in S)$. Se $n = N$, **parar**; caso contrário, incrementar $n \leftarrow n + 1$ e retornar ao 2º passo.

Cabe observar que este processo de construção da população inicial começa pela criação aleatória de uma solução S viável (passos 1º –2º). Em seguida, as colunas redundantes da solução S são eliminadas (passos 3º –5º) de forma aleatória. O processo se repete até que haja N soluções compondo a população (6º passo).

2.6.2 SELEÇÃO POR TORNEIO

O método de seleção por torneio opera a partir de dois conjuntos, T_1 e T_2 , cada qual com k indivíduos selecionados aleatoriamente da população. Em seguida, o indivíduo com a melhor capacidade de adaptação em T_1 , e outro indivíduo em T_2 , são selecionados para reprodução.

A aptidão do indivíduo i é calculada segundo a função objetivo de SCP (2.1):

$$f_i = \sum_{j=1}^n c_j s_{ij}$$

onde s_{ij} é o valor do j -ésimo bit (facilidade) da *string* correspondente ao i -ésimo indivíduo, enquanto c_j corresponde ao custo do bit (facilidade) j .

2.6.3 REPRODUÇÃO POR FUSÃO

A reprodução por fusão produz uma única solução C , através da seguinte combinação entre os indivíduos P_1 e P_2 :

1°. se $P_1[i] = P_2[i]$, então $C[i] \leftarrow P_1[i] = P_2[i]$;

2°. se $P_1[i] \neq P_2[i]$, então

a. $C[i] \leftarrow P_1[i]$, com probabilidade $p = f(P_2)/[f(P_1) + f(P_2)]$;

b. $C[i] \leftarrow P_2[i]$, com probabilidade $1 - p$;

2.6.4 MUTAÇÃO

A probabilidade de mutação no algoritmo genético de Beasley e Chu (1996) é variável. Ou seja, quanto mais o algoritmo genético converge, maior a probabilidade de mutação dos indivíduos. O intuito consiste em compensar a característica redução progressiva da exploração do espaço de soluções via reprodução, através do aumento da probabilidade de mutação à medida que o algoritmo genético converge.

Entretanto, seguindo as recomendações de alguns pesquisadores (BACK, 1993, *apud* BEASLEY; CHU, 1996) (De JONG, 1975, *apud* BEASLEY; CHU, 1996) resolvemos adotar neste trabalho a probabilidade de $1/n$ para mutação de um bit em um indivíduo de tamanho n . Até porque, em correspondência com Beasley e Chu, estes não mais sabiam como um dos parâmetros de seu processo de mutação variável era atualizado à medida que o algoritmo convergia.

Vale salientar que a probabilidade de mutação fixa é a única diferença entre o algoritmo genético original de Beasley e Chu (1996) e o implementado neste trabalho. Mesmo assim, como será visto no Capítulo 7, o algoritmo genético continua apresentando os melhores resultados, principalmente quando a escala das instâncias SCP aumenta.

2.6.5 VIABILIZAÇÃO E AJUSTE DA SOLUÇÃO

Uma solução S , fruto da reprodução por fusão e/ou mutação, pode acabar violando as restrições do problema de cobertura (2.2) e (2.3). Ou seja, algumas localidades i da matriz $\mathbf{a} = [a_{ij}]$ podem ainda estar descobertas, logo sendo necessária a adição de novas facilidades à solução S .

O processo de viabilização e ajuste de solução é composto pelos seguintes passos:

1º. inicializar $w_i \leftarrow |S \cap \alpha_i|, \forall i \in I$;

2º. inicializar $U = \{i \mid w_i = 0, \forall i \in I\}$;

- 3°. para cada linha i em U (em ordem crescente de i):
- encontrar a primeira coluna j (em ordem crescente de j) em α_i que minimize $c_j / |U \cap \beta_j|$;
 - adicionar j a S e incrementar $w_i \leftarrow w_i + 1, \forall i \in \beta_j$. Remover as linhas i do conjunto U que pertencem a β_j ;
- 4°. para cada coluna j em S (em ordem decrescente de j), se $w_i \geq 2, \forall i \in \beta_j$, subtrair a coluna j do conjunto S e decrementar $w_i \leftarrow w_i - 1, \forall i \in \beta_j$;
- 5°. retornar S como uma solução viável que contém nenhuma coluna redundante.

Cabe observar que a adição de colunas baseia-se na razão entre o custo da coluna e o número de linhas descobertas que alcança (passo 3.a). Em seguida, após restaurada a viabilidade da solução S , o processo de eliminação das colunas redundantes é ativado (passo 4-5).

2.6.6 REPOSIÇÃO INCREMENTAL

No método de reposição **incremental** dos indivíduos pela prole oriunda do processo reprodutivo, apenas parte da população é considerada. Justamente aquela formada pelas soluções com capacidade de adaptação abaixo da média da qualidade das soluções da população, donde serão escolhidas aleatoriamente as soluções que serão substituídas pela prole.

Uma vez que o presente capítulo definiu o problema de cobertura de conjuntos (Seção 2.1), determinou sua complexidade (Seção 2.2), e baseado no trabalho de Caprara, Fischetti e Toth (2000) apresentou algumas das mais recentes e eficientes heurísticas da literatura (Seção 2.3), inclusive destacando as que abordaram o problema de cobertura de conjuntos pelo modelo primal

(Seções 2.4 a 2.6). Agora, chega a vez da meta-heurística baseada em colônia de formigas desenvolvida por Dorigo (1992) ser explanada em detalhes no próximo capítulo.

3. OTIMIZAÇÃO BASEADA EM COLÔNIA DE FORMIGAS

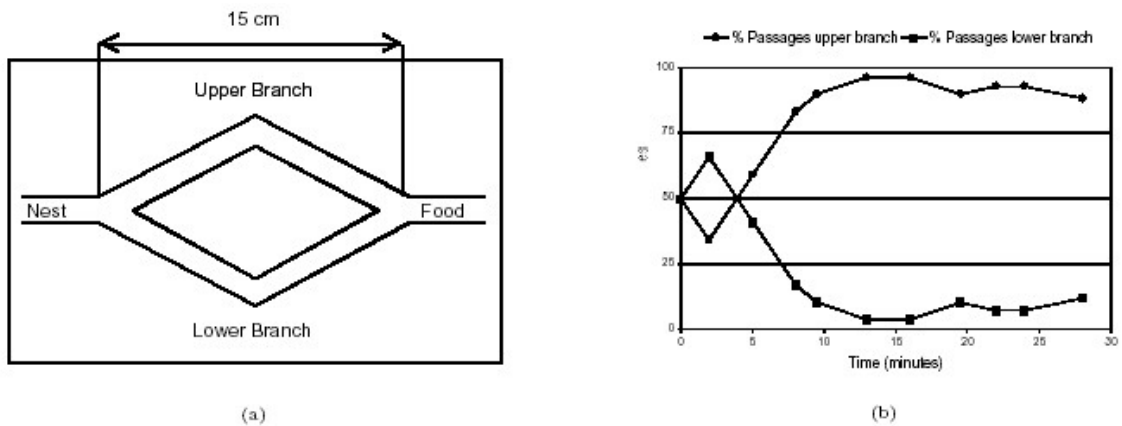
O presente capítulo apresenta uma série de instâncias (Seção 3.4-5) oriundas da meta-heurística *ACO* (Seção 3.3) (DORIGO; CARO, 1999a) (DORIGO; CARO, 1999b): apogeu do trabalho seminal (Seção 3.2) desenvolvido por Marco Dorigo (1992), sob a inspiração das experiências com formigas reais (Seção 3.1) realizadas por Goss, Aron, Deneubourg, e Pasteels (1989). O propósito é oferecer a oportunidade do leitor conhecer o método de otimização baseado em colônia de formigas concebido por Dorigo (1992), antes de adaptá-lo ao problema de cobertura de conjuntos (tema do próximo capítulo).

3.1 EXPERIÊNCIA COM FORMIGAS REAIS

Esta seção apresenta a experiência com formigas reais realizada por Goss, Aron, Deneubourg, e Pasteels (1989), que serviu de inspiração à criação do método de otimização de colônia de formigas. Esta experiência consistiu na submissão de uma colônia de formigas *Iridomyrmex humilis* a um fonte de alimento através de dois caminhos distintos (Figura 3).

O experimento foi planejado de maneira que as formigas, ao percorrerem ambas as direções (do ninho à fonte de alimento, e vice-versa), tinham sempre que optar por um ou outro caminho (Figura 3.a). Após uma fase transitória, a maioria das formigas acabava por escolher o mesmo caminho, assim como, a probabilidade de selecionar o caminho de menor distância aumentava, à medida que a diferença de tamanho entre os caminhos tornava-se maior.

Figura 3: Esquema das Experiências de Goss, Aron, Deneuborg e Pasteels



Fonte: (GOSS et al., 1989)

A explicação para tal comportamento está relacionada à substância, denominada **feromônio**, depositada pelas formigas durante sua locomoção. Quando as formigas chegam a um ponto de decisão, como a intersecção entre dois caminhos distintos, elas fazem uma escolha probabilística baseada na quantidade de feromônio presente nestes caminhos. Ou seja, quanto maior a quantidade de feromônio em um caminho, maior a probabilidade deste ser escolhido.

Inicialmente, não existe feromônio nos caminhos, portanto a probabilidade de escolha é a mesma para ambos. Entretanto, segundo Goss (1989, p. 580): “... *random fluctuations, after an initial transitory phase, cause a few more ants to randomly select one branch over the other ...*”. Sendo assim, quando os caminhos têm a mesma distância, a maior concentração de feromônio naquele caminho com um pouco mais de formiga, estimulará mais formigas a segui-lo; e assim cada vez mais, até que a grande maioria das formigas acabe optando pelo mesmo (Figura 3.b).

Quando os caminhos têm distâncias distintas, a taxa de chegada de formigas na fonte de alimentos, vindas pelo menor caminho, é maior do que a taxa de chegada de formigas vindas pelo maior caminho. Sendo assim, ao iniciar a viagem de retorno ao ninho, estas formigas hão de encontrar mais feromônio sobre o menor caminho. Este fato estimulará mais formigas a segui-lo,

e assim cada vez mais, até que a grande maioria das formigas acabe optando pelo mesmo. Este processo ocorre da mesma maneira no sentido inverso.

Em suma, de acordo com Goss (1989, p. 580): “... *in the case in which the bridge’s branches are of different length, the importance of initial random fluctuations is much reduced, and the stochastic pheromone trail following foraging behavior of the ants, coupled to differential branch length, is the main mechanism at work ...*”.

As próximas seções deste capítulo explicam como a experiência acima citada originou um método baseado em formigas capaz de solucionar problemas de otimização.

3.2 ANT SYSTEM E O PROBLEMA DO CAIXEIRO VIAJANTE

Aplicado pela primeira vez sobre o **problema do caixeiro viajante**, a heurística *Ant System (AS)* (DORIGO, 1992) foi o primeiro algoritmo baseado em colônia de formigas desenvolvido por Dorigo (Quadro 3).

Inicialmente, m formigas são distribuídas pelas cidades segundo algum critério pré-estabelecido, e todos os caminhos $(i, j) \in E$ são inicializados com a mesma quantidade $\tau_{ij}(1) > 0$ de feromônio (Quadro 3, linha 1).

Em seguida, cada formiga k ($k = 1, \dots, m$) seleciona as próximas cidades a serem visitadas (Quadro 3, linhas 5-6), através da seguinte regra de decisão probabilística:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, \text{ se } j \in N_i^k; \text{ senão, } p_{ij}^k(t) = 0. \quad (3.1)$$

Onde:

- $p_{ij}^k(t)$: probabilidade da cidade j ser escolhida pela formiga k , atualmente situada na cidade i , durante a t -ésima iteração da heurística AS ;
- $\tau_{ij}(t)$: intensidade do feromônio presente no caminho $(i, j) \in E$ na t -ésima iteração de AS ;
- α : parâmetro que regula a influência de $\tau_{ij}(t)$;
- $\eta_{ij} = \frac{1}{d_{ij}}$: visibilidade da cidade j com relação a cidade i , sendo d_{ij} : distância entre a cidade i e j ;
- β : parâmetro que regula a influência de η_{ij} ;
- $N_i^k(t)$: conjunto de cidades ainda não visitadas pela formiga k situada na cidade i durante a t -ésima iteração de AS ;

Este **processo de seleção da próxima cidade a ser visitada** é repetido até que todas as formigas tenham completado um caminho interligando todas as cidades uma única vez (Quadro 3, linha 4).

Depois, o **processo de depósito e evaporação de feromônio** (Quadro 3, linhas 11-12) ocorre da seguinte maneira:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3.2)$$

sendo $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$, com $\Delta\tau_{ij}^k(t) = \frac{Q}{L_k(t)}$, se a formiga k percorreu o caminho $(i, j) \in E$; caso contrário, $\Delta\tau_{ij}^k = 0$.

Onde:

- t : iteração atual da heurística AS ;
- $\rho \in [0,1]$: parâmetro que regula a redução de $\tau_{ij}(t)$;
- $\Delta\tau_{ij}(t)$: ganho total de feromônio no caminho $(i, j) \in E$, ocorrido na t -ésima iteração;
- m : número de formigas;
- $\Delta\tau_{ij}^k(t)$: ganho de feromônio no caminho $(i, j) \in E$, causado pela formiga k , na t -ésima iteração de AS ;
- Q : quantidade de feromônio excretada por uma formiga a cada iteração;
- $S_k(t)$: caminho completo que interliga todas as cidades uma única vez, descoberto pela formiga k na t -ésima iteração de AS ;
- $L_k(t)$: distância associada ao caminho completo $S_k(t)$ descoberto pela formiga k na t -ésima iteração de AS .

Todos os procedimentos acima citados são repetidos a cada iteração da heurística *Ant System*. Porém, quando o número máximo de iterações previamente estabelecido é alcançado (Quadro 3, linha 2), a heurística *Ant System* retorna a melhor solução até então encontrada (Quadro 3, linha 14).

Quadro 3: *AS-TSP: Ant System* no Problema do Caixeiro Viajante.

```

1 Fase de Inicialização
2 para  $t$  indo de 1 até um número máximo de iterações faça
3   para  $k$  indo de 1 até  $m$  faça
4     repita até a formiga  $k$  encontrar um caminho completo
5       selecione a próxima cidade a ser visitada segundo a regra
6       de decisão probabilística  $p_{ij}^k(t)$ ;
7     fim-do-repita
8     calcule a distância  $L_k(t)$  do caminho descoberto pela formiga  $k$ ;
9     se  $L_k(t) < L^*$  então  $S^* \leftarrow S_k(t)$ ;
10    fim-do-para
11    atualize  $\tau_{ij}(t+1)$ ,  $\forall (i, j) \in E$ , segundo processo de depósito
12    e evaporação de feromônio específico;
13  fim-do-para
14  retornar  $S^*$ ;

```

3.3 META-HEURÍSTICA *ACO* (*ANT COLONY OPTIMIZATION*)

O objetivo desta seção consiste em apresentar a meta-heurística *ACO* desenvolvida por Dorigo e Caro (1999a), cujas instâncias (os **algoritmos ACO**) podem ser utilizadas na busca por caminhos de custo mínimo em um grafo $G = (C, L, W)$, contanto que estes caminhos sejam viáveis segundo às restrições $\Omega(C, L, \theta)$, definidas abaixo.

O grafo $G = (C, L, W)$ e as restrições $\Omega(C, L, \theta)$, que juntos representam de maneira geral um problema de otimização discreto, são definidos da seguinte forma (DORIGO; CARO, 1999a):

- $C = \{c_1, c_2, \dots, c_{N_C}\}$ é o conjunto finito de N_C **componentes** do problema, como por exemplo, as **idades** do problema do caixeiro viajante (TSP);
- $L = \{l_{c_i c_j} \mid c_i, c_j \in C\}$ é o conjunto finito de possíveis **arestas** entre os elementos de C . Como por exemplo, a aresta $(i, j) \in E$ que interliga as cidades em TSP;
- W é o conjunto de **pesos** associados, ou a C , ou a L , ou a ambos. Por exemplo, a **distância** d_{ij} das arestas $(i, j) \in E$ que interligam as cidades em TSP;
- $\Omega(C, L, \theta)$ é o conjunto finito de **restrições** atribuídas aos elementos de C e L , com θ indicando que mudanças de restrição poderão ocorrer ao longo do tempo. Por exemplo, se uma determinada cidade for visitada em *TSP*, ela não poderá ser visitada novamente.
- os estados do problema são definidos por seqüências $s = \langle a_1, a_2, \dots, a_k, \dots \rangle$, com $a_i = c_k \in C$, $k = 1, \dots, N_C$;
- Caso c_1 seja o último componente do estado s_1 , então um estado s_2 é dito **vizinho** a s_1 , se somente se, $\exists c_2 \in C \mid (l_{c_1 c_2} \in L) \wedge (s_2 = \langle s_1, c_2 \rangle)$. Por exemplo, se existe em TSP uma aresta $(i, j) \in E$ entre as cidades $i, j \in E$, então o estado $s_2 = \langle s_1, j \rangle$ é vizinho ao estado $s_1 = \langle s_0, i \rangle$;
- Caso a seqüência $s = \langle a_1, a_2, \dots, a_k \rangle$, $(a_i, a_{i+1}) \in L$, seja viável por satisfazer as restrições $\Omega(C, L, \theta)$, então esta seqüência constituirá uma solução viável ψ do problema abordado;
- \tilde{S} é o conjunto de todas as soluções viáveis do problema;
- A cada solução viável ψ está associado um custo $J(\psi)$ dado em função dos custos $J(l_{c_i c_j}, \theta)$ das arestas $l_{c_i c_j} \in \psi$, ou dos custos $J(c_i, \theta)$ dos componentes $c_i \in C$, ou de

ambos. Por exemplo, dado que em TSP $J(l_{c_i c_j}, \theta) = d_{ij}$, onde d_{ij} é a distância entre a cidade i e j ; então $J(\psi) = \sum_{(i,j) \in \psi} d_{ij}$. Vale frisar que os custos $J(l_{c_i c_j}, \theta)$, e por conseguinte $J(\psi)$, podem estar parametrizados por alguma medida de tempo θ . Mas este não é o caso de TSP.

Em um algoritmo *ACO*, as formigas constróem soluções para o problema representado por G e Ω de forma incremental. Cada formiga k parte de uma solução parcial $\psi_k(1)$ composta de um único elemento pertencente a C , e segue adicionando novos elementos a $\psi_k(h)$ até alcançar uma solução viável ψ . A notação $\psi_k(h)$ representa a solução atual parcial construída em h passos pela formiga k , logo composta por h elementos pertencentes a C . Os componentes candidatos à adição em $\psi_k(h)$ são aleatoriamente selecionados a partir de uma vizinhança apropriadamente definida em torno do último elemento de $\psi_k(h)$. Esta escolha aleatória é feita aplicando-se uma política de decisão que faz uso de informações locais relacionadas aos vértices e/ou conexões visitadas. Após construir a solução viável ψ (ou enquanto a constrói), a formiga deposita feromônio sobre os componentes e/ou arestas visitadas, segundo à qualidade da solução ψ encontrada (ou que vem encontrando). Este feromônio depositado pela formiga é o que influenciará a construção da solução das demais.

O método de otimização baseado em colônia de formigas leva em conta as seguintes considerações (DORIGO; CARO, 1999a):

- 1º. A meta de uma formiga é a busca por soluções viáveis ψ de custo $J(\psi)$ mínimo. Por exemplo, em TSP a meta consiste na busca de soluções viáveis ψ que minimizem

$$J(\psi) = \sum_{(i,j) \in \psi} d_{ij};$$

- 2°. Uma formiga k tem uma **memória** M^k que armazena o caminho até então percorrido. A memória pode ser usada para: (i) construir soluções viáveis; (ii) avaliar a solução encontrada; e (iii) retornar pelo mesmo caminho percorrido;
- 3°. Uma formiga k no estado $s_r = \langle s_{r-1}, i \rangle$ pode mover-se para qualquer nó j , contanto que este nó pertença à **vizinhança** N_i do nó i em que se encontra. Além disso, a viabilidade com o estado s_r deve ser mantida. Ou seja, j deve pertencer a N_i^k , sendo $N_i^k = \{j \mid (j \in N_i) \wedge (\langle s_r, j \rangle \in \tilde{S})\}$;
- 4°. Uma formiga k localizada sobre o nó i seleciona o nó $j \in N_i^k$ segundo uma **regra de decisão probabilística**. Por exemplo, a regra p_{ij}^k (Equação 3.2) definida pela heurística *Ant System*;
- 5°. Pode-se atribuir um **estado inicial** s_s^k a uma formiga k , assim como uma ou mais **condições e^k de parada**. Por exemplo, em *AS_TSP* (Seção 3.2) seria ($s_s^k = i \in E$) e ($e^k \equiv$, linha 4), respectivamente.
- 6°. As formigas partem de seus respectivos estados iniciais em direção a um estado vizinho viável, e assim sucessivamente, até que pelo menos uma condição de término e^k seja alcançada por alguma formiga k ;
- 7°. A medida que se move do nó i para seu vizinho j , a formiga pode atualizar a trilha de feromônio τ_{ij} sobre a aresta (i, j) . Este processo denomina-se **atualização de feromônio passo-a-passo** (*online step-by-step pheromone update*);
- 8°. Uma vez construída uma solução, a formiga pode, ao retornar pelo caminho inverso ao percorrido, atualizar a trilha de feromônio. Este processo denomina-se **atualização de**

feromônio com atraso (*online delayed pheromone update*). Por exemplo, o processo definido na Equação 3.3 da heurística *Ant System*.

Dorigo e Caro (1999a) descrevem a meta-heurística *ACO* em pseudo-código, tal como disposto no Quadro 4.

Quadro 4: Meta-heurística em pseudo-código.

```
1 procedimento meta-heurística ACO()  
2   enquanto (condições de término  $e^k$  não satisfeitas) faça  
3     escalonar_atividades  
4         geração_e_atividade_das_formigas();  
5         evaporação_de_feromônio();  
6         atividades_opcionais();  
7     fim_escalonar_atividades  
8   fim_enquanto  
9 fim-procedimento
```

Embora as rotinas *geração_e_atividade_das_formigas*, *evaporação_de_feromônio* e *atividades_opcionais* sejam geralmente executadas em seqüência, a presença da primitiva **escalonar_atividades** (Quadro 4) indica a possibilidade de ocorrência da execução em paralelo. Assim como, a possibilidade de hibridização com outras técnicas de otimização (ex: o método de busca local), encontra-se prevista como atividade opcional na meta-heurística *ACO* (Quadro 4).

No Quadro 5, detalhes em pseudo-código da rotina *geração_e_atividade_das_formigas* (DORIGO; CARO, 1999a) são apresentados. A partir de um estado inicial (Quadro 5, linhas 3-4), uma formiga visita os nós do grafo G até que o estado alvo seja alcançado (Quadro 5, linha 5). A cada passo, a formiga k (situada no nó i) move-se para o próximo estado (Quadro 5, linha 7) de acordo com alguma regra de decisão (Quadro 5, linha 6); e se for o caso, já deposita feromônio sobre a aresta visitada (Quadro 5, linhas 8-10).

Quadro 5: Rotina *geração_e_atividade_das_formigas()* em pseudo-código.

```

1...procedimento geração_e_atividade_das_formigas( )
2      enquanto (recursos_disponíveis) faça
3          criar_e_inicializar formiga( );
4           $M \leftarrow$  atualizar_memória_da_formiga( );
5          enquanto (estado_corrente  $\neq$  estado_alvo) faça
6              próximo_estado  $\leftarrow$  aplicar_regra_de_decisão( );
7              mover_para_próximo_estado(próximo_estado);
8              se (atualização_de_feromônio_passo_a_passo) então
9                  depositar_feromônio_sobre_arco_visitado( );
10             fim_do_se
11              $M \leftarrow$  atualizar_memória_da_formiga( );
12          fim_do_enquanto
13          se (atualização_de_feromônio_com_atraso) então
14              para cada (arco_visitado  $\in \mathcal{A}$ ) faça
15                  depositar_feromônio_sobre_arco_visitado( );
16              fim_do_para
17          fim_do_se
18      fim_do enquanto
19 fim-procedimento

```

No caso da atualização de feromônio com atraso, cada formiga ao alcançar o estado alvo, voltará pelo mesmo caminho outrora percorrido, atualizando a trilha de feromônio (Quadro 5, linhas 13-17).

Após a geração e atividade das formigas, chega a vez do processo de evaporação (Quadro 4 - linha 5), seguido das possíveis atividades opcionais para auxílio no desempenho do algoritmo (Quadro 4 - linha 6). Por fim, todo o processo é repetido até que alguma formiga k satisfaça pelo menos uma das condições e^k de término (Quadro 4 - linha 2).

3.4 ALGORITMOS ACO

Cabe observar pela Tabela 5 que a meta-heurística ACO tem sido aplicada em vários problemas discretos de otimização, tais como, o problema do caixeiro viajante, a atribuição quadrática, o roteamento em redes de telecomunicações, o roteamento de veículos, etc. Entretanto, o algoritmo *Ant System (AS)* (DORIGO; MANIEZZO; COLORNI, 1991) ainda continua servindo de base à criação da maioria dos algoritmos ACO, tais como (Tabela 5): *Ant-Q*, *ACS*, *MMAS*, *AS_{rank}*, *AS-QAP*, *MMAS-QAP*, *ANTS-QAP*, *AS-FSP*, *AS-VRP*, *HAS-SOP*, *ANTS-FAP*, e *ACS-SCP*. Ou seja, todas as referências dispostas na Tabela 5, exceto *HAS-QAP* (GAMBARDELLA; TAILLARD, 1997) e *Ant-Net* (CARO; DORIGO, 1997), foram modificações e/ou extensões do algoritmo *AS*.

Tabela 5: Aplicações do Algoritmo ACO

Problema	Nome do Algoritmo	Ano	Referência Bibliográfica
Caxeiro Viajante	<i>AS</i>	1991	(DORIGO; MANIEZZO; COLORNI, 1991)
	<i>Ant-Q</i>	1995	(GAMBARDELLA; DORIGO, 1995)
	<i>ACS</i>	1996	(DORIGO; GAMBARDELLA, 1997)
	<i>MMAS</i>	1997	(STUTZLE; HOOS, 1998)
	<i>AS_{rank}</i>	1997	(BULLNHEIMER; HARTL; STRAUSS, 1999a)
Atribuição Quadrática	<i>AS-QAP</i>	1994	(MANIEZZO; COLORNI; DORIGO, 1999)
	<i>HAS-QAP</i>	1997	(GAMBARDELLA; TAILLARD, 1997)
	<i>MMAS-QAP</i>	1999	(STUTZLE; HOOS, 1999)
	<i>ANTS-QAP</i>	1998	(MANIEZZO, 1999)
Escalonamento	<i>AS-FSP</i>	1997	(STUTZLE, 1997)
Roteamento de Veículo	<i>AS-VRP</i>	1997	(BULLNHEIMER; HARTL; STRAUSS, 1999b)
Roteamento de Rede orientada à conexão	<i>Ant-Net</i>	1997	(CARO; DORIGO, 1997)
Ordenação Seqüencial	<i>HAS-SOP</i>	1997	(GAMBARDELLA; DORIGO, 1997)
Atribuição de Frequência	<i>ANTS-FAP</i>	1998	(MANIEZZO; CARBONARO, 2000)
Cobertura	<i>ACS-SCP</i>	2000	(HADJI, et al., 2000)

Fonte: (DORIGO; STUTZLE; CARO, 2000, p. 858).

Dada a importância de *AS*, decidimos não apenas utilizá-la no estudo da meta-heurística *ACO* em SCP (Capítulo 4), mas também utilizar uma de suas extensões: a heurística *MAX-MIN Ant System* (Seção 3.5) desenvolvida por Stützle e Hoss (1998), no intuito de melhorar seu desempenho.

3.5 MAX-MIN ANT SYSTEM: A HEURÍSTICA DE STÜTZLE E HOSS

Suponha o grafo $G = (C, L, W)$ e as restrições $\Omega(C, L, \theta)$ que representam um problema de otimização discreto, tal como descrito na seção 3.3, onde:

- $C = \{c_1, c_2, \dots, c_{N_c}\}$ é o conjunto finito de N_c componentes do problema;
- $L = \{l_{c_i c_j} \mid c_i, c_j \in C\}$ é o conjunto finito de possíveis conexões entre os elementos de C ;
- W é o conjunto de pesos associados, ou a C , ou a L , ou a ambos.

Pois bem, a heurística *MAX-MIN Ant System (MMAS)* difere da heurística *Ant System (AS)* em três aspectos fundamentais. O **primeiro aspecto** diz respeito à quantidade τ_{ij} de feromônio permitida nos caminhos $(i, j) \in L$. Em *MMAS*, τ_{ij} encontra-se limitado entre τ_{\min} e τ_{\max} . Sendo τ_{\min} o limite mínimo de feromônio consentido, enquanto τ_{\max} é o limite máximo.

O **segundo aspecto** que diferencia *MMAS* de *AS* é que, a cada iteração, apenas uma única formiga deposita feromônio sobre o caminho percorrido. Esta formiga pode ser aquela que encontrou a melhor solução da iteração atual. Ou então, a formiga que encontrou a melhor solução, dentre todas as iterações até então ocorridas.

A partir dos dois aspectos acima citados, para todo $(i, j) \in L$, o processo de depósito e evaporação de feromônio torna-se igual a:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}(t) \quad (3.3)$$

se $\tau_{\min} \leq (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}(t) \leq \tau_{\max}$.

Caso contrário, se $(1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}(t) > \tau_{\max}$, então $\tau_{ij}(t+1) = \tau_{\max}$. Senão, se

$(1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}(t) < \tau_{\min}$, então $\tau_{ij}(t+1) = \tau_{\min}$. Sendo $\Delta\tau_{ij}^{best}(t) = \frac{1}{J(\psi^{best})}$, se $(i, j) \in \psi^{best}$,

ou $\Delta\tau_{ij}^{best}(t) = 0$, se $(i, j) \notin \psi^{best}$. Onde:

- t : iteração atual da heurística *MMAS*;
- τ_{\max} : limite máximo de feromônio consentido sobre $(i, j) \in L$;
- τ_{\min} : limite mínimo de feromônio consentido sobre $(i, j) \in L$;
- $\rho \in [0,1]$: parâmetro que regula a redução de $\tau_{ij}(t)$;
- $\Delta\tau_{ij}^{best}(t)$: ganho de feromônio em $(i, j) \in L$, ocorrido na t -ésima iteração de *MMAS*;
- ψ^{best} : a melhor solução encontrada, seja esta a melhor solução encontrada na iteração atual (ψ^{ib}), ou a melhor solução encontrada dentre todas as iterações até então ocorridas (ψ^{gb});
- $J(\psi^{best})$: o custo da solução ψ^{best} .

Por fim, o **terceiro e último aspecto** que diferencia *MMAS* de *AS*, encontra-se na inicialização da quantidade de feromônio em τ_{\max} , para todas as arestas do grafo $G = (C, L, W)$ (Quadro 6, linha 1). Ou seja, $\tau_{ij}(1) = \tau_{\max}$, para todo $(i, j) \in L$.

Quadro 6: Esquema geral da heurística *MMAS*

```

1 Fase de Inicialização:  $J(\psi^{ig}) \leftarrow \infty$ ;  $\tau_{ij}(1) = \tau_{\max}$ , para todo  $(i, j) \in L$ .
2 para  $t$  indo de 1 até um número máximo de iterações faça
3    $J(\psi^{ib}) \leftarrow \infty$ ;
4   para  $k$  indo de 1 até  $m$  faça
5     repita até a formiga  $k$  encontrar uma solução viável  $\psi$ 
6       selecione o próximo nó a ser visitado, segundo a regra
7       de decisão probabilística  $p_{ij}^k(t)$ ;
8     fim-do-repita
9     calcule o custo  $J(\psi)$  da solução  $\psi$  descoberta pela formiga  $k$ ;
10    se  $J(\psi) < J(\psi^{gb})$  então  $\psi^{gb} \leftarrow \psi$ ;
11    se  $J(\psi) < J(\psi^{ib})$  então  $\psi^{ib} \leftarrow \psi$ ;
12  fim-do-para
13  atualize  $\tau_{ij}(t+1)$ ,  $\forall (i, j) \in E$ , segundo a Equação 3.3,
14  com  $\psi^{best} = \psi^{ib}$  e/ou  $\psi^{best} = \psi^{gb}$ .
15 fim-do-para
16 retornar  $\psi^{gb}$ ;

```

A utilização de apenas uma solução, seja ψ^{ib} ou ψ^{gb} , no processo de depósito e evaporação de feromônio (Equação 3.3), é uma forma de explorar os melhores resultados do espaço de soluções viáveis. Pois a quantidade de feromônio sobre as arestas $(i, j) \in \psi^{best}$ que formam as melhores soluções serão reforçadas. Entretanto, usando apenas $\psi^{best} = \psi^{gb}$, a **estagnação da busca** pode ocorrer. Ou seja, a busca pode convergir rapidamente para uma solução local ótima, não explorando devidamente o espaço de soluções viáveis (STUTZLE; HOSS, 1999).

Por outro lado, se $\psi^{best} = \psi^{ib}$, então o problema da estagnação pode ser atenuado, pois a melhor solução da iteração atual, geralmente muda significativamente de iteração a iteração. Logo, um número maior de arestas $(i, j) \in \psi^{best}$ pode ocasionalmente receber um reforço em sua quantidade de feromônio, ampliando assim o alcance de exploração do espaço de soluções viáveis (STUTZLE; HOSS, 1999).

Uma estratégia mista para o processo de depósito e evaporação de feromônio (3.3) também pode ser empregada. Por exemplo, utilizar $\psi^{best} = \psi^{ib}$ como procedimento padrão, mas tendo $\psi^{best} = \psi^{gb}$ a cada número fixo de iterações. Inclusive, segundo Stützle e Hoos (1998)⁸, a estratégia mista é a mais adequada para instâncias de grande porte do problema do caixeiro viajante (TSP), e do problema da atribuição quadrática (QAP).

A regra de decisão probabilística mantida por *MMAS* continua sendo a mesma de *AS*:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, \text{ se } j \in N_i^k; \text{ senão, } p_{ij}^k(t) = 0. \quad (3.4)$$

Onde:

- $p_{ij}^k(t)$: probabilidade do nó $j \in C$ ser escolhido pela formiga k , atualmente situada no nó $i \in C$, durante a t -ésima iteração da heurística *MMAS*;
- $\tau_{ij}(t)$: intensidade do feromônio no caminho $(i, j) \in L$, na t -ésima iteração de *MMAS*;
- α : parâmetro que regula a influência de $\tau_{ij}(t)$;
- η_{ij} : visibilidade do nó $j \in C$, com relação ao nó $i \in C$;

⁸ Neste trabalho, Stützle e Hoss hibridizam *MMAS* com o método de busca local.

- β : parâmetro que regula a influência de η_{ij} ;
- $N_i^k(t)$: conjunto dos nós ainda não visitados pela formiga k , atualmente situada em $i \in C$, durante a t -ésima iteração de *MMAS*;

Entretanto, cabe observar que a estagnação da busca no espaço de soluções viáveis também pode ocorrer se, a cada momento de escolha do próximo nó a ser visitado, a quantidade de feromônio no caminho que segue a um determinado nó é significativamente maior do que as outras opções possíveis. Pois, devido à regra de decisão probabilística descrita pela Equação 3.4, a maior parte das formigas vão preferir este nó específico, ao invés das outras alternativas. Com o passar do tempo, a quantidade de feromônio nos caminhos que interligam estes nós será cada vez mais reforçada. Logo, culminando em uma situação de estagnação.

Para dirimir o problema de estagnação acima descrito, a heurística *MMAS* limita a quantidade máxima (τ_{\max}) e mínima (τ_{\min}) de feromônio permitido nas arestas $(i, j) \in L$ do grafo $G = (C, L, W)$. O intuito consiste em evitar que a diferença relativa entre as quantidades de feromônio nas arestas $(i, j) \in L$ torne-se extrema durante o decorrer da execução do algoritmo.

Stützle e Hoss (1998) definem o limite máximo τ_{\max} de *MMAS* como sendo:

$$\tau_{\max}(t) = \frac{1}{\rho} \frac{1}{J(\psi^{gb})} \quad (3.5)$$

Visto que

$$\lim_{t \rightarrow \infty} \tau_{ij}(t) \leq \frac{1}{\rho} \frac{1}{J(\psi^{opt})} \quad (3.6)$$

Onde:

- $\tau_{\max}(t)$: a quantidade máxima de feromônio permitida em $(i, j) \in L$ durante a t -ésima iteração de *MMAS*;
- $\tau_{ij}(t)$: intensidade do feromônio no caminho $(i, j) \in L$, na t -ésima iteração de *MMAS*;
- $\rho \in [0,1]$: parâmetro que regula a redução de $\tau_{ij}(t)$;
- ψ^{gb} : a melhor solução encontrada, dentre todas as iterações até então ocorridas;
- $J(\psi^{gb})$: custo da solução ψ^{gb} ;
- ψ^{opt} : solução ótima do problema;
- $J(\psi^{opt})$: custo da solução ψ^{opt} .

Cabe observar que (3.6) determina um limite assintótico máximo para a quantidade de feromônio em uma aresta $(i, j) \in L$. Ou seja, dado que $1/J(\psi^{opt})$ é a quantidade máxima de feromônio depositado em uma aresta $(i, j) \in L$ após uma iteração qualquer de *MMAS*, então segundo o processo de depósito e evaporação de feromônio (3.3), a quantidade de feromônio em $(i, j) \in L$, até a t -ésima iteração de *MMAS*, é no máximo igual a:

$$(1 - \rho)^t \tau_{ij}(1) + \sum_{i=1}^t (1 - \rho)^{t-i} \frac{1}{J(\psi^{opt})}.$$

De acordo com a convergência de uma série geométrica infinita (LARSON; HOSTETLER; EDWARDS, 1995), quando $t \rightarrow \infty$ a equação acima converge para $\frac{1}{\rho} \frac{1}{J(\psi^{opt})}$. Como $\tau_{\max}(t)$

(3.5) é obtida pela substituição de ψ^{opt} por ψ^{gb} em $\frac{1}{\rho} \frac{1}{J(\psi^{opt})}$, significa que $\tau_{\max}(t)$ é uma estimativa deste limite assintótico máximo, atualizada a cada melhor solução global encontrada.

Quanto ao limite mínimo τ_{\min} para *MMAS*, Stützle e Hoss (1998) o definem da seguinte forma:

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)(1 - \sqrt[n]{p_{best}})}{(avg - 1)\sqrt[n]{p_{best}}}$$

Onde:

- $\tau_{\min}(t)$: a quantidade mínima de feromônio permitida em $(i, j) \in L$ durante a t -ésima iteração de *MMAS*;
- $\tau_{\max}(t)$: a quantidade máxima de feromônio permitida em $(i, j) \in L$ durante a t -ésima iteração de *MMAS*;
- n : número de componentes da solução do problema;
- $avg = n/2$: quantidade média de alternativas que a formiga leva em consideração, no momento de decisão do próximo nó a ser visitado;
- ψ^{gb} : a melhor solução em todas as iterações até então ocorridas;
- $p_{best} = p_{dec}^n$: probabilidade com que ψ^{gb} é encontrada após a convergência de *MMAS*;
- p_{dec} : probabilidade de escolher um componente $c_i \in \psi^{gb}$, $i = 1, \dots, n$.

Ou seja, de acordo com a regra de decisão probabilística de *MMAS* (Equação 3.4), levando-se em conta apenas a quantidade de feromônio presente em $G = (C, L, W)$ após a convergência de

MMAS, a probabilidade p_{dec} de uma formiga escolher um componente $c_i \in \psi^{gb}$ ($i = 1, \dots, n$) é igual a:

$$p_{dec} = \frac{\tau_{\max}(t)}{\tau_{\max}(t) + (avg - 1)\tau_{\min}}.$$

Resolvendo a equação acima para τ_{\min} temos:

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)(1 - p_{dec})}{(avg - 1)p_{dec}} = \frac{\tau_{\max}(t)(1 - \sqrt[n]{p_{best}})}{(avg - 1)\sqrt[n]{p_{best}}}.$$

Note que quando $p_{best} = 1$, então $\tau_{\min}(t) = 0$. Se ρ_{best} é muito pequeno, pode ocorrer de $\tau_{\min}(t) > \tau_{\max}(t)$. Nestes casos, $\tau_{\min}(t) = \tau_{\max}(t)$ o que significa apenas a utilização de η_{ij} .

O processo de seleção do próximo nó a ser visitado (Quadro 6, linhas 6-7) é repetido até que todas as formigas tenham encontrado uma solução viável ψ (Quadro 6, linha 5). Depois, o processo de depósito e evaporação de feromônio é disparado (Quadro 6, linhas 13-14).

Todos os procedimentos são repetidos a cada iteração da heurística *MMAS*. Porém, quando o número máximo de iterações previamente estabelecido for alcançado (Quadro 6, linha 2), a heurística *MMAS* retorna a melhor solução até então encontrada (Quadro 6, linha 16).

Uma vez que o presente capítulo apresentou uma série de algoritmos (Seção 3.4) da meta-heurística ACO (Seção 3.3), destacando as heurísticas *Ant System* (Seção 3.2) e *MAX-MIN Ant System* (Seção 3.5), agora chega a vez de adaptá-las ao problema de cobertura de conjuntos (tema do próximo capítulo).

4. ACO E PROBLEMA DE COBERTURA DE CONJUNTOS

Este capítulo apresenta as adaptações que realizamos da meta-heurística *ACO* sobre o problema de cobertura de conjuntos. Estas adaptações estão baseadas nas heurísticas *Ant-System* (Seção 3.2) e *MAXMIN Ant System* (Seção 3.5) e seguem os cânones adotados pela comunidade *ACO*. Tanto é assim que, por exemplo, o trecho correspondente às formigas no algoritmo híbrido de Hadji, Rahoual, Talbi e Bachelet (2000) é o mesmo adotado em nossa adaptação de *Ant-System* em *SCP*. No total, são dez algoritmos cuja motivação em utilizá-los é uma consequência do rumo que os experimentos tomaram neste trabalho, assunto abordado em mais detalhes no Capítulo 6.

A estrutura do presente capítulo encontra-se organizada da seguinte maneira. A seção 4.1 explica a atuação da meta-heurística *ACO* em problemas de subconjuntos, como é o caso de *SCP*. A seção 4.2 adapta a heurística *Ant System* a *SCP* (*AS-SCP*), enquanto a seção 4.3 ajusta *MAX-MIN Ant System* ao mesmo problema (*MAXMIN-SCP*). A seção 4.4 aborda o trabalho correlato de Hadji, Rahoual, Talbi e Bachelet (2000), cuja estrutura servirá de referência às hibridizações sobre *AS-SCP* (Seção 4.5). Por fim, a seção 4.6 apresenta a heurística *MAXMINLS-SCP*: a hibridização sobre *MAXMIN-SCP*.

4.1 META-HEURÍSTICA *ACO* EM PROBLEMAS DE SUBCONJUNTOS

A maior parte das aplicações da meta-heurística *ACO* tem se concentrado em resolver **problemas de ordenação**, como por exemplo, o problema do caixeiro viajante (*Traveling Salesman Problem – TSP*) (DORIGO; MANIEZZO; COLORNI, 1991), a atribuição quadrática (*Quadratic Assignment Problem - QAP*) (MANIEZZO; COLORNI; DORIGO, 1999), o roteamento em redes de telecomunicações (CARO; DORIGO, 1997), o roteamento de veículos (*Vehicle Routing Problem – VRP*) (BULLNHEIMER; HARTL; STRAUSS, 1999a), entre outros.

Entretanto, existe uma outra classe de problemas, denominada de forma geral como **problemas de subconjuntos** (*Subset Problems - SP*), em que a ordem dos elementos que formam uma solução não é levada em conta (LEGUIZAMÓN, G.; MICHALEWICZ, Z., 1999). Nesta classe encontram-se (RARDIN, 1998) (DASKIN, 1995): o problema da mochila múltipla (*Multiple Knapsack Problem - MKP*); o problema do conjunto independente máximo (*Maximum Independent Set Problem - MISPP*); o problema de cobertura de vértices (*Vertex Cover Problem - VCP*); o problema de cobertura de conjuntos (*Set Covering Problem - SCP*); o problema de cobertura máximo (*Maximum Covering Problem - MCP*); o problema das p-mediana (*P-Median Problem - PMP*), entre outros.

Segundo a formalização da meta-heurística ACO (Seção 3.3) realizada por Dorigo e Caro (1999a), o processo de depósito de feromônio ocorre sobre os arcos $(i, j) \in L$ do grafo $G(C, L, W)$ que representa o problema de otimização discreto abordado. Sendo assim, a quantidade de feromônio em um arco $(i, j) \in L$ está diretamente associada à probabilidade da seqüência $\langle i, j \rangle$ participar das soluções construídas pelas formigas vindouras. Porém, este *modus operandi* não é adequado aos problemas de subconjunto, pois são os componentes que interessam, e não a seqüência na qual podem estar dispostos. Ou seja, para abordar os problemas de subconjuntos, o processo de depósito de feromônio deve ocorrer sobre os nós $i \in C$ do grafo não-orientado e totalmente conectado $G(C, L, W)$.

A próxima seção explica como o depósito de feromônio da heurística *Ant-System* (Seção 3.2) pode ser modificado para ocorrer sobre os nós do grafo que representa o problema de cobertura de conjuntos.

4.2 ANT SYSTEM E O PROBLEMA DE COBERTURA DE CONJUNTOS

Dado um conjunto $J = \{1, \dots, m\}$ com m facilidades, e outro $I = \{1, \dots, n\}$ com n localidades. Foi visto (Seção 2.1) que o problema de cobertura de conjuntos consiste em encontrar um

subconjunto $S \subseteq J$ que minimize a função objetivo de SCP (2.1), e que satisfaça suas restrições (2.2) – (2.3). Ou seja, pelo fato das soluções SCP serem conjuntos, a ordem dos elementos que as formam não é relevante por definição. Logo, SCP pertence à classe SP .

O grafo $G = (C, L, W)$ não-orientado e totalmente conectado para SCP^9 , cujos nós $i \in C$ receberão feromônio, é construído da seguinte maneira:

- $C = J$ é o conjunto finito de facilidades do problema de cobertura de conjuntos;
- $L = \{(a, b) \mid a, b \in J \times J; a \neq b\}$ é o conjunto de todas as conexões entre duas facilidades $a, b \in J$ distintas;
- $W = \{c_j \mid j \in J\}$ é o conjunto de custos c_j associados a cada elemento de C , ou melhor, a cada facilidade $j \in J$;
- Seja $I = \{1, \dots, n\}$ o conjunto das localidades de SCP . Suponha que para cada facilidade $j \in J$ exista um subconjunto de localidades $I_j \subseteq I$ associado. Então, $\Omega(C, L, \theta)$ em SCP é formado pela restrição de que haja pelo menos um conjunto $S \subseteq J$, tal que $I = \bigcup_{j \in S} I_j$ (Seção 2.1).

Inicialmente, h formigas são distribuídas pelas facilidades segundo algum critério pré-estabelecido, e todas as facilidades $j \in J$ são inicializadas com a mesma quantidade $\tau_j(1) > 0$ de feromônio (Quadro 7, linha 1).

Em seguida, cada formiga k ($k = 1, \dots, h$) seleciona as próximas facilidades a serem visitadas (Quadro 7, linhas 5-6), através da seguinte regra de decisão probabilística:

⁹ O grafo foi especificado como sendo totalmente conectado para manter a associação com o experimento das formigas reais, entretanto na implementação o grafo é não-conectado.

$$p_{ij}^k(t) = \frac{[\tau_j(t)]^\alpha [\eta_j]^\beta}{\sum_{l \in N_i^k} [\tau_j(t)]^\alpha [\eta_j]^\beta}, \text{ se } j \in N_i^k; \text{ senão, } p_{ij}^k(t) = 0. \quad (4.1)$$

Onde:

- t : iteração atual da heurística *AS-SCP*;
- $p_{ij}^k(t)$: probabilidade da facilidade j ser escolhida pela formiga k , atualmente situada na facilidade i , durante a t -ésima iteração da heurística *AS-SCP*;
- $\tau_j(t)$: quantidade de feromônio sobre a facilidade $j \in J$ na t -ésima iteração;
- α : parâmetro que regula a influência de $\tau_j(t)$;
- $\eta_j = \frac{c_j}{|U \cap I_j|}$: visibilidade da facilidade j com relação à facilidade i . Sendo c_j , o custo da facilidade j ; U , o conjunto de localidades atualmente descobertas; I_j , o conjunto de localidades cobertas pela facilidade $j \in J$; e $|\cdot|$, o operador de cardinalidade de conjuntos;
- β : parâmetro que regula a influência de η_j ;
- $N_i^k(t)$: conjunto de facilidades ainda não visitadas pela formiga k , atualmente situada na facilidade i , nesta t -ésima iteração de *AS-SCP*;

Este processo de seleção da próxima facilidade a ser visitada é repetido até que todas as formigas tenham encontrado uma solução viável ψ para SCP (Quadro 7, linha 4).

Depois, o processo de depósito e evaporação de feromônio (Quadro 7, linhas 11-12) ocorre da seguinte maneira:

$$\tau_j(t+1) = (1 - \rho)\tau_j(t) + \Delta\tau_j(t) \quad (4.2)$$

sendo $\Delta\tau_j(t) = \sum_{k=1}^h \Delta\tau_j^k(t)$, com $\Delta\tau_j^k(t) = \frac{Q}{J[\psi_k(t)]}$, se a formiga k visitou a facilidade $j \in J$.

Caso contrário, $\Delta\tau_{ij}^k = 0$.

Onde:

- t : iteração atual da heurística *AS-SCP*;
- $\rho \in [0,1]$: parâmetro que regula a redução de $\tau_j(t)$;
- $\Delta\tau_j(t)$: ganho total de feromônio na facilidade $j \in J$, ocorrido na t -ésima iteração;
- h : número de formigas;
- $\Delta\tau_j^k(t)$: ganho de feromônio na facilidade $j \in J$, causado pela formiga k , na t -ésima iteração de *AS-SCP*;
- Q : quantidade de feromônio excretada por uma formiga a cada iteração;
- $\psi_k(t) \subseteq J$: solução viável para *SCP*, descoberta pela formiga k na t -ésima iteração;
- $J[\psi_k(t)] = \sum_{j \in \psi_k(t)} c_j$: custo da solução viável $\psi_k(t)$ sendo c_j , o custo da facilidade j .

Todos os procedimentos acima citados são repetidos a cada iteração da heurística *Ant System*. Porém, quando o número máximo de iterações previamente estabelecido é alcançado (Quadro 7, linha 2), a heurística *Ant System* retorna a melhor solução ψ^* até então encontrada (Quadro 7, linha 14).

Quadro 7: *Ant System* no problema de cobertura de conjuntos

```
1 Fase de Inicialização
2 para  $t$  indo de 1 até um número máximo de iterações faça
3   para  $k$  indo de 1 até  $m$  faça
4     repita até a formiga  $k$  encontrar uma solução viável  $\psi_k(t)$  para SCP
5       selecione a próxima facilidade  $j \in J$  a ser visitada segundo a regra
6       de decisão probabilística  $p_{ij}^k(t)$ ;
7     fim-do-repita
8     calcule o custo  $J[\psi_k(t)]$  da solução viável descoberta pela formiga  $k$ ;
9     se  $J[\psi_k(t)] < J(\psi^*)$  então  $\psi^* \leftarrow \psi_k(t)$ ;
10    fim-do-para
11    atualize  $\tau_j(t+1)$ ,  $\forall j \in J$ , segundo processo de depósito
12    e evaporação de feromônio;
13  fim-do-para
14 retornar  $\psi^*$ ;
```

4.3 MAX-MIN ANT SYSTEM E O PROBLEMA DE COBERTURA DE CONJUNTOS

O Quadro 8 apresenta a heurística *MAX-MIX Ant System* (Seção 3.5) para *SCP* (Seção 2.1). Dado $G = (C, L, W)$ e $\Omega(C, L, \theta)$ para *SCP* (Seção 4.2), todas as facilidades $j \in J$ são inicializadas com a mesma quantidade $\tau_j(1) = \tau_{\max}$ de feromônio; e h formigas são distribuídas pelas facilidades segundo algum critério pré-estabelecido (Quadro 8, linha 1).

Quadro 8: MAXMINSCP: *MAX-MIN Ant System* para SCP

```

1 Fase de Inicialização:  $J(\psi^{ig}) \leftarrow \infty$ ;  $\tau_j(1) = \tau_{\max}$ , para todo  $j \in J$ .
2 para  $t$  indo de 1 até um número máximo de iterações faça
3    $J(\psi^{ib}) \leftarrow \infty$ ;
4   para  $k$  indo de 1 até  $m$  faça
5     repita até a formiga  $k$  encontrar uma solução viável  $\psi$  para SCP
6       selecione a próxima facilidade a ser visitada segundo  $p_{ij}^k(t)$ ;
7     fim-do-repita
8     calcule o custo  $J(\psi)$  da solução  $\psi$  descoberta pela formiga  $k$ ;
9     se  $J(\psi) < J(\psi^{gb})$  então  $\psi^{gb} \leftarrow \psi$ ;
10    se  $J(\psi) < J(\psi^{ib})$  então  $\psi^{ib} \leftarrow \psi$ ;
11  fim-do-para
12  atualize  $\tau_j(t+1)$ ,  $\forall j \in J$  segundo o processo
13  de depósito e evaporação de feromônio;
14 fim-do-para
15 retornar  $\psi^{gb}$ ;

```

Em seguida, cada formiga k ($k = 1, \dots, h$) seleciona as próximas facilidades a serem visitadas (Quadro 8, linhas 6), através da mesma regra de decisão adotada por *AS-SCP* (Seção 4.2):

$$p_{ij}^k(t) = \frac{[\tau_j(t)]^\alpha [\eta_j]^\beta}{\sum_{l \in N_i^k} [\tau_l(t)]^\alpha [\eta_l]^\beta}, \text{ se } j \in N_i^k; \text{ senão, } p_{ij}^k(t) = 0.$$

Onde:

- t : iteração atual da heurística *AS-SCP*;
- $p_{ij}^k(t)$: probabilidade da facilidade j ser escolhida pela formiga k , atualmente situada na facilidade i , durante a t -ésima iteração da heurística *AS-SCP*;
- $\tau_j(t)$: quantidade de feromônio sobre a facilidade $j \in J$ na t -ésima iteração;
- α : parâmetro que regula a influência de $\tau_j(t)$;
- $\eta_j = \frac{c_j}{|U \cap I_j|}$: visibilidade da facilidade j com relação à facilidade i . Sendo c_j , o custo da facilidade j ; U , o conjunto de localidades atualmente descobertas; I_j , o conjunto de localidades cobertas pela facilidade $j \in J$; e $|\cdot|$, o operador de cardinalidade de conjuntos;
- β : parâmetro que regula a influência de η_{ij} ;
- $N_i^k(t)$: conjunto de facilidades ainda não visitadas pela formiga k , atualmente situada na facilidade i , nesta t -ésima iteração de *AS-SCP*;

Este processo de seleção da próxima facilidade a ser visitada é repetido até que todas as formigas tenham encontrado uma solução viável ψ para SCP (Quadro 8, linha 4).

Depois, o processo de depósito e evaporação de feromônio (Quadro 8, linhas 12 e 13) ocorre da seguinte maneira:

$$\tau_j(t+1) = (1 - \rho)\tau_j(t) + \Delta\tau_j^{best}(t) \quad (4.3)$$

Se $\tau_{\min}(t) \leq (1 - \rho)\tau_j(t) + \Delta\tau_j^{best}(t) \leq \tau_{\max}(t)$. Caso contrário, se $(1 - \rho)\tau_j(t) + \Delta\tau_j^{best}(t) > \tau_{\max}(t)$, então $\tau_j(t+1) = \tau_{\max}(t)$. Senão, se $(1 - \rho)\tau_j(t) + \Delta\tau_j^{best}(t) < \tau_{\min}(t)$, então $\tau_j(t+1) = \tau_{\min}(t)$. Sendo $\Delta\tau_j^{best}(t) = \frac{1}{J(\psi^{best})}$, se $j \in \psi^{best}$, ou $\Delta\tau_j^{best}(t) = 0$, se $j \notin \psi^{best}$.

Onde:

- t : iteração atual da heurística *MAXMIN-SCP*;
- $\tau_{\max}(t)$: limite máximo de feromônio consentido a $j \in J$ na t -ésima iteração;
- $\tau_{\min}(t)$: limite mínimo de feromônio consentido sobre $j \in J$ na t -ésima iteração;
- $\tau_j(t)$: quantidade de feromônio sobre a facilidade $j \in J$ na t -ésima iteração;
- $\rho \in [0,1]$: parâmetro que regula a redução de $\tau_j(t)$;
- $\Delta\tau_j^{best}(t)$: ganho de feromônio em $j \in J$, ocorrido na t -ésima iteração de *MAXMIN-SCP*;
- ψ^{ib} : a melhor solução encontrada na iteração atual;
- ψ^{gb} : a melhor solução encontra, dentre todas as iterações até então ocorridas;
- ψ^{best} : a melhor solução encontrada, seja esta ψ^{ib} ou ψ^{gb} ;
- $J(\psi^{best})$: o custo da solução ψ^{best} .

Todos os procedimentos são repetidos a cada iteração da heurística *MAXMIN-SCP*. Porém, quando o número máximo de iterações previamente estabelecido é alcançado (Quadro 8, linha 2), a heurística *MAXMIN-SCP* retorna a melhor solução até então encontrada (Quadro 8, linha 15).

A heurística *MAXMIN-SCP* adota a estratégia mista (Seção 3.5) para o processo de depósito e evaporação de feromônio, na medida em que utiliza $\psi^{best} = \psi^{ib}$ como procedimento padrão; mas a cada número fixo de iterações $\psi^{best} = \psi^{gb}$.

Para dirimir o problema de estagnação descrito na Seção 3.5, a heurística *MMAS* limita a quantidade máxima (τ_{max}) de feromônio permitida nas facilidades $j \in J$ do grafo $G = (C, L, W)$ da seguinte forma:

$$\tau_{max}(t) = \frac{1}{\rho} \frac{1}{J(\psi^{gb})}$$

Onde:

- $\tau_{max}(t)$: a quantidade máxima de feromônio permitida em $j \in J$ durante a t -ésima iteração de *MAXMIN-SCP*;
- $\rho \in [0,1]$: parâmetro que regula a redução de feromônio sobre as facilidades;
- ψ^{gb} : a melhor solução encontrada, dentre todas as iterações até então ocorridas;
- $J(\psi^{gb})$: custo da solução ψ^{gb} .

Quanto ao limite mínimo τ_{min} , *MAXMIN-SCP* o define da seguinte forma:

$$\tau_{min}(t) = \frac{\tau_{max}(t)(1 - \sqrt[n]{p_{best}})}{(avg - 1)\sqrt[n]{p_{best}}}$$

Onde:

- $\tau_{min}(t)$: a quantidade mínima de feromônio permitida em $j \in J$ durante a t -ésima iteração de *MAXMIN-SCP*;

- $\tau_{\max}(t)$: a quantidade máxima de feromônio permitida em $j \in J$ durante a t -ésima iteração de *MAXMIN-SCP*;
- n : número de componentes da solução ótima do problema;
- $avg = n/2$: quantidade média de alternativas que a formiga leva em consideração no momento de decisão do próximo nó a ser visitado;
- ψ^{gb} : a melhor solução em todas as iterações até então ocorridas;
- p_{best} : probabilidade com que ψ^{gb} é encontrada após a convergência de *MAXMIN-SCP*.

Note que quando $p_{best} = 1$, então $\tau_{\min}(t) = 0$. Se p_{best} é muito pequeno, pode ocorrer de $\tau_{\min}(t) > \tau_{\max}(t)$. Nestes casos, $\tau_{\min}(t) = \tau_{\max}(t)$ o que significa apenas a utilização de η_{ij} .

4.4 ACS-SCP: O ALGORITMO DE HADJI, RAHOUAL, TALBI E BACHELET

O objetivo desta seção consiste em apresentar *ACS-SCP*: o método baseado em colônia de formigas para *SCP*, desenvolvido por Hadji, Rahoual, Talbi e Bachelet (2000) (Quadro 9), cuja estrutura servirá de referência à hibridização de *AS-SCP* (Seção 4.5) e *MAXMIN-SCP* (Seção 4.6).

Dado $G = (C, L, W)$ e $\Omega(C, L, \theta)$ para *SCP* (Seção 4.2), todas as facilidades $j \in J$ são inicializadas com a mesma quantidade $\tau_j(1) > 0$ de feromônio; e h formigas são distribuídas pelas facilidades segundo algum critério pré-estabelecido (Quadro 9, linha 1).

Através do processo de **diversificação** (Quadro 9, linhas 4-5), cada formiga k ($k = 1, \dots, h$) recebe uma solução não-viável com NC_MAX facilidades aleatoriamente extraídas do conjunto $J = \{1, \dots, m\}$.

Com o intuito de tornar viável sua solução parcial inicial (Quadro 9, linha 6), cada formiga k ($k = 1, \dots, h$) seleciona as próximas facilidades a serem visitadas (Quadro 9, linhas 7-8), através da mesma regra de decisão probabilística da heurística *AS-SCP* (Seção 4.2):

$$p_{ij}^k(t) = \frac{[\tau_j(t)]^\alpha [\eta_j]^\beta}{\sum_{l \in N_i^k} [\tau_l(t)]^\alpha [\eta_l]^\beta}, \text{ se } j \in N_i^k; \text{ senão, } p_{ij}^k(t) = 0.$$

Onde:

- t : iteração atual da heurística *AS-SCP*;
- $p_{ij}^k(t)$: probabilidade da facilidade j ser escolhida pela formiga k , atualmente situada na facilidade i , durante a t -ésima iteração da heurística *AS-SCP*;
- $\tau_j(t)$: quantidade de feromônio sobre a facilidade $j \in J$ na t -ésima iteração;
- α : parâmetro que regula a influência de $\tau_j(t)$;
- $\eta_j = \frac{c_j}{|U \cap I_j|}$: visibilidade da facilidade j com relação à facilidade i . Sendo c_j , o custo da facilidade j ; U , o conjunto de localidades atualmente descobertas; I_j , o conjunto de localidades cobertas pela facilidade $j \in J$; e $|\cdot|$, o operador de cardinalidade de conjuntos;
- β : parâmetro que regula a influência de η_{ij} ;

- $N_i^k(t)$: conjunto de facilidades ainda não visitadas pela formiga k , atualmente situada na facilidade i , nesta t -ésima iteração de *AS-SCP*;

Após tornar sua solução parcial inicial viável, cada formiga dispara o processo de eliminação de facilidades redundantes (Quadro 9, linha 11), que serve de pré-requisito à etapa de atualização da melhor solução até então encontrada (Quadro 9, linhas 12 e 13).

Quadro 9: Heurística ACS-SCP de Hadji, Rahoual, Talbi e Bachelet (HADJI et al. 2000)

```

1 Fase de Inicialização
2 para  $t$  indo de 1 até um número máximo IT_MAX de iterações faça
3   para  $k$  indo de 1 até  $m$  faça
4     [construir uma solução não-viável para a formiga  $k$  contendo
5       NC_MAX facilidades escolhidas aleatoriamente];
6     repita até (a formiga  $k$  encontrar uma solução viável  $\psi_k(t)$  para SCP)
7       [selecione a próxima facilidade  $j \in J$  a ser visitada,
8         segundo a regra de decisão probabilística  $p_{ij}^k(t)$ ];
9     fim-do-repita
10    se ( $t=IT\_MAX$ ) então [atualizar  $\psi_k(t)$  via método de busca local];
11    [eliminar as facilidades redundantes em  $\psi_k(t)$ ];
12    [calcular o custo  $J[\psi_k(t)]$  da solução  $\psi_k(t)$ ];
13    se  $J[\psi_k(t)] < J(\psi^*)$  então [ $\psi^* \leftarrow \psi_k(t)$ ];
14    fim-do-para
15    [atualizar  $\tau_j(t+1)$ ,  $\forall j \in J$ , segundo processo de depósito e
16      evaporação de feromônio];
17 fim-do-para
18 [atualizar  $\psi^*$  via método de busca local];
19 retornar  $\psi^*$ ;

```

O processo de depósito e evaporação de feromônio (Quadro 9, linhas 12 e 13) é semelhante ao da heurística *AS-SCP* (Seção 4.2):

$$\tau_j(t+1) = (1 - \rho)\tau_j(t) + \Delta\tau_j(t)$$

sendo $\Delta\tau_j(t) = \sum_{k=1}^h \Delta\tau_j^k(t)$, com $\Delta\tau_j^k(t) = \frac{Q}{J[\psi_k(t)]}$, se a formiga k visitou a facilidade $j \in J$; caso contrário, $\Delta\tau_{ij}^k = 0$.

Onde:

- t : iteração atual da heurística *AS-SCP*;
- $\rho \in [0,1]$: parâmetro que regula a redução de $\tau_j(t)$;
- $\Delta\tau_j(t)$: ganho total de feromônio na facilidade $j \in J$, ocorrido na t -ésima iteração;
- h : número de formigas;
- $\Delta\tau_j^k(t)$: ganho de feromônio na facilidade $j \in J$, causado pela formiga k , na t -ésima iteração de *AS-SCP*;
- Q : quantidade de feromônio excretada por uma formiga a cada iteração;
- $\psi_k(t) \subseteq J$: solução viável para *SCP*, descoberta pela formiga k na t -ésima iteração;
- $J[\psi_k(t)] = \sum_{j \in \psi_k(t)} c_j$: custo associado a solução viável $\psi_k(t)$;
- c_j : custo da facilidade j .

Todos os procedimentos acima citados (Quadro 9, linhas 3-16) são retomados a cada iteração da heurística. Porém, quando o número máximo de iterações previamente estabelecido é alcançado (Quadro 9, linha 2), o método da busca local correspondente a *LS-ECC-SCP* (Seção 2.4) entra em ação, atualizando cada uma das soluções obtidas pelas formigas na última iteração do algoritmo. Dessas atualizações, a que tiver o melhor custo passará novamente pelo método de busca local (Quadro 9, linha 18), que por sua vez retornará o resultado final do algoritmo.

O trabalho de Hadji, Rahoual, Talbi e Bachelet (2000) não informa se algum método de análise experimental foi adotado na avaliação de *ACS-SCP*. Além disso, esta avaliação não ocorre segundo o enfoque causal, apenas de acordo com o enfoque comparativo, com este último sem nenhuma checagem a respeito de outras heurísticas para *SCP* presentes na literatura, como por exemplo, as descritas no artigo do Caprara, Fischetti e Toth (2000). Por fim, Hadji, Rahoual, Talbi e Bachelet (2000) relatam a viabilidade de *ACS-SCP* para instâncias *SCP* de grande porte.

4.5 HIBRIDIZAÇÕES SOBRE A HEURÍSTICA *AS-SCP*

Para se aproximar da qualidade dos resultados das heurísticas citadas na Seção 2.3, Hadji, Rahoual, Talbi e Bachelet (2000) recorreram à combinação da heurística *Ant System* com as técnicas de diversificação (*DIV*), eliminação de colunas redundantes (*ECC*) e busca local (*LS*) (Seção 4.4).

Compartilhando do mesmo propósito decidimos realizar esta mesma combinação sobre a heurística *AS-SCP*. Mas desta vez com o cuidado de averiguar a contribuição de cada técnica sobre a qualidade das soluções obtidas. Haja vista que alguma pode não estar melhorando significativamente os resultados.

Para dar cabo a tal tarefa, as hibridizações foram realizadas de forma gradativa. Primeiro, através da heurística *AS-ECC-SCP* (Quadro 10): *AS-SCP* (Seção 4.2) combinada com a técnica de eliminação de colunas redundantes (*ECC*). Depois, por meio da heurística *AS-ECC-LSK-SCP* (Quadro 11): método de busca local aplicado apenas na última iteração de *AS-ECC-SCP*. Em seguida, via a heurística *AS-ECC-LSK-LS-SCP* (Quadro 12): busca local aplicada não somente na última iteração de *AS-ECC-SCP*, mas também como tarefa final do algoritmo. Por último, a heurística *AS-DIV-ECC-LSK-LS-SCP* (Quadro 13): *AS-ECC-LSK-LS-SCP* dotada do método da diversificação.

Com o propósito de averiguar outras possibilidades de hibridizações via busca local, também construímos as heurísticas: *AS-ECC-LSKIN-SCP*(Quadro 14), *AS-ECC-LSKIN-LS-SCP*(Quadro 15), e *AS-DIV-ECC-LSKIN-LS-SCP*(Quadro 16), caracterizadas pelo fato da busca local ocorrer a cada iteração do algoritmo, ao invés de apenas atuar na última iteração. Estes algoritmos são extensões das heurísticas *AS-ECC-LSK-SCP*, *AS-ECC-LSK-LS-SCP*, e *AS-DIV-ECC-LSK-LS-SCP*, respectivamente.

Cabe observar que todas estas heurísticas são extensões de *AS-SCP*. Sendo assim, a diferença com relação a *AS-SCP* encontra-se destacada nas figuras auto-explicativas (Quadro 10 a Quadro 16) associadas a cada uma dessas hibridizações.

Quadro 10: Heurística *AS-ECC-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número máximo de iterações faça
3	para k indo de 1 até m faça
4	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
5	selecione a próxima facilidade $j \in J$ a ser visitada segundo a regra
6	de decisão probabilística $p_{ij}^k(t)$;
7	fim-do-repita
8	eliminar as facilidades redundantes em $\psi_k(t)$
9	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
10	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
11	fim-do-para
12	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
13	e evaporação de feromônio;
14	fim-do-para
15	retornar ψ^* ;

Quadro 11: Heurística *AS-ECC-LSK-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número IT_MAX máximo de iterações faça
3	para k indo de 1 até m faça
4	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
5	selecione a próxima facilidade $j \in J$ a ser visitada segundo a
6	regra de decisão probabilística $p_{ij}^k(t)$;
7	fim-do-repita
8	se ($t=IT_MAX$) então atualizar $\psi_k(t)$ via método de busca local
9	eliminar as facilidades redundantes em $\psi_k(t)$
10	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
11	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
12	fim-do-para
13	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
14	e evaporação de feromônio;
15	fim-do-para
16	retornar ψ^* ;

Quadro 12: Heurística *AS-ECC-LSK-LS-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número IT_MAX máximo de iterações faça
3	para k indo de 1 até m faça
4	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
5	selecione a próxima facilidade $j \in J$ a ser visitada segundo a regra
6	de decisão probabilística $p_{ij}^k(t)$;
7	fim-do-repita
8	se $(t=IT_MAX)$ então atualizar $\psi_k(t)$ via método de busca local
9	eliminar as facilidades redundantes em $\psi_k(t)$
10	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
11	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
12	fim-do-para
13	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
14	e evaporação de feromônio;
15	fim-do-para
16	atualizar ψ^* via método de busca local
17	retornar ψ^* ;

Quadro 13: *AS-DIV-ECC-LSK-LS-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número IT_MAX máximo de iterações faça
3	para k indo de 1 até m faça
4	construir uma solução não-viável para a formiga k contendo
5	NC_MAX facilidades escolhidas aleatoriamente;
6	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
7	selecione a próxima facilidade $j \in J$ a ser visitada segundo a regra
8	de decisão probabilística $p_{ij}^k(t)$;
9	fim-do-repita
10	se ($t=IT_MAX$) então atualizar $\psi_k(t)$ via método de busca local
11	eliminar as facilidades redundantes em $\psi_k(t)$
12	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
13	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
14	fim-do-para
15	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
16	e evaporação de feromônio;
17	fim-do-para
18	atualizar ψ^* via método de busca local
19	retornar ψ^* ;

Quadro 14: Heurística *AS-ECC-LSKIN-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número máximo de iterações faça
3	para k indo de 1 até m faça
4	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
5	selecione a próxima facilidade $j \in J$ a ser visitada segundo a regra
6	de decisão probabilística $p_{ij}^k(t)$;
7	fim-do-repita
8	atualizar $\psi_k(t)$ via método de busca local
9	eliminar as facilidades redundantes em $\psi_k(t)$
10	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
11	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
12	fim-do-para
13	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
14	e evaporação de feromônio;
15	fim-do-para
16	retornar ψ^* ;

Quadro 15: *AS-ECC-LSKIN-LS-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número máximo de iterações faça
3	para k indo de 1 até m faça
4	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
5	selecione a próxima facilidade $j \in J$ a ser visitada segundo a regra
6	de decisão probabilística $p_{ij}^k(t)$;
7	fim-do-repita
8	atualizar $\psi_k(t)$ via método de busca local
9	eliminar as facilidades redundantes em $\psi_k(t)$
10	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
11	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
12	fim-do-para
13	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
14	e evaporação de feromônio;
15	fim-do-para
16	atualizar ψ^* via método de busca local
17	retornar ψ^* ;

Quadro 16: *AS-DIV-ECC-LSKIN-LS-SCP*

1	Fase de Inicialização
2	para t indo de 1 até um número máximo de iterações faça
3	para k indo de 1 até m faça
4	construir uma solução não-viável para a formiga k contendo
5	NC_MAX facilidades escolhidas aleatoriamente;
6	repita até a formiga k encontrar uma solução viável $\psi_k(t)$ para SCP
7	selecione a próxima facilidade $j \in J$ a ser visitada segundo a regra
8	de decisão probabilística $p_{ij}^k(t)$;
9	fim-do-repita
10	atualizar $\psi_k(t)$ via método de busca local
11	eliminar as facilidades redundantes em $\psi_k(t)$
12	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
13	se $J[\psi_k(t)] < J(\psi^*)$ então $\psi^* \leftarrow \psi_k(t)$;
14	fim-do-para
15	atualize $\tau_j(t+1)$, $\forall j \in J$, segundo processo de depósito
16	e evaporação de feromônio;
17	fim-do-para
18	atualizar ψ^* via método de busca local
19	retornar ψ^* ;

4.6 MAXMINLS-SCP: A HIBRIDIZAÇÃO DA HEURÍSTICA MAXMIN-SCP

O Quadro 17 apresenta a heurística *MAXMINLS-SCP* (Seção 3.5) para *SCP*, com destaque para a única diferença com relação *MAXMIN-SCP*.

Quadro 17: Heurística *MAXMINLS-SCP*

1	Fase de Inicialização: $J(\psi^{ig}) \leftarrow \infty$; $\tau_j(1) = \tau_{\max}$, para todo $j \in J$.
2	para t indo de 1 até um número máximo de iterações faça
3	$J(\psi^{ib}) \leftarrow \infty$;
4	para k indo de 1 até m faça
5	repita até a formiga k encontrar uma solução viável ψ para <i>SCP</i>
6	selecione a próxima facilidade a ser visitada segundo $p_{ij}^k(t)$;
7	fim-do-repita
8	atualizar $\psi_k(t)$ via método de busca local
9	eliminar as facilidades redundantes em $\psi_k(t)$
10	calcular o custo $J[\psi_k(t)]$ da solução $\psi_k(t)$;
11	se $J[\psi_k(t)] < J(\psi^{gb})$ então $\psi^{gb} \leftarrow \psi_k(t)$;
12	se $J[\psi_k(t)] < J(\psi^{ib})$ então $\psi^{ib} \leftarrow \psi_k(t)$;
13	fim-do-para
14	atualize $\tau_j(t+1)$, $\forall j \in J$ segundo o processo
15	de depósito e evaporação de feromônio;
16	fim-do-para
17	retornar ψ^{gb} ;

Uma vez que as adaptações e hibridizações baseadas nas heurísticas *Ant System* e *MAXMIN Ant System* foram apresentadas, o próximo capítulo explica a estratégia utilizada para analisar todos os algoritmos descritos neste capítulo.

5. MÉTODO EXPERIMENTAL

Sabe-se que eventuais descuidos ocorridos na etapa de avaliação de um algoritmo podem levar a conclusões equivocadas a respeito de seu desempenho. Por este motivo, o presente capítulo trata da estratégia desenvolvida para analisar de forma rigorosa a meta-heurística *ACO* no problema de cobertura de conjuntos (SCP), mas que também pode ser adotada como método geral de análise experimental de quaisquer heurísticas de otimização.

As etapas (Seção 5.1) desta estratégia experimental são suportadas não apenas por métodos estatísticos comparativos (Seção 5.2) e de regressão (Seção 5.3), mas também por um método rigoroso de delineamento do projeto experimental (Seção 5.4).

5.1 ESTRATÉGIA DE ANÁLISE

Geralmente, o objetivo da análise experimental de heurísticas é de cunho **descritivo**, **causal**, ou **comparativo** (BARR, et al., 1995). No enfoque descritivo, a meta consiste em basicamente mensurar o **desempenho** da heurística segundo a qualidade da solução¹⁰ e o esforço

¹⁰ A **qualidade da solução** fornecida por uma heurística é geralmente medida por sua distância à solução ótima do problema abordado. Caso a solução ótima do problema não seja conhecida, a melhor solução até então encontrada na literatura especializada pode servir como referência de qualidade. Exemplos de métrica de qualidade são as mais variadas possíveis, como por exemplo, o desvio percentual $\frac{(y - y^*)}{y^*}$ para problemas de minimização, sendo y a solução fornecida pela heurística e y^* a solução ótima do problema (ou a melhor até então encontrada).

computacional¹¹ empregado. No enfoque causal o objetivo é investigar **como** e **quais** os fatores que influenciam de modo relevante o desempenho da heurística. Por fim, o enfoque comparativo confronta o desempenho da heurística investigada com o de outros métodos de otimização (HOOKER, 1996).

Com relação ao estudo sobre a meta-heurística *ACO* no problema de cobertura de conjuntos, nossa estratégia de análise experimental está organizada em quatro etapas distintas (Figura 4):

- 1°. **Análise das Adaptações de ACO sobre SCP:** etapa destinada ao desenvolvimento e análise de adaptações de *ACO* sobre *SCP* sem recorrer às atividades opcionais previstas na Seção 3.3 (Quadro 4, linha 6).
- 2°. **Análise da Relação Parâmetros-Resultado:** a melhor adaptação, descoberta na etapa anterior, passa por um exame mais detalhado, cujo objetivo consiste em descobrir qual a relação (caso exista) dos parâmetros usados neste algoritmo sobre as respostas observadas nos experimentos.
- 3°. **Análise das Versões Híbridas:** usando uma configuração propícia para os parâmetros avaliados na segunda etapa, versões híbridas de *ACO* sobre *SCP* são também criadas e comparadas.
- 4°. **Análise das Comparações com outras Heurísticas:** a melhor proposta descoberta nas etapas anteriores é confrontada com algumas das melhores heurísticas existentes na literatura sobre cobertura de conjuntos.

Os métodos estatísticos comparativos e de regressão que suportam estas etapas serão devidamente explanados nas duas seções seguintes.

¹¹ O **esforço computacional** representa os recursos (tempo, memória, estrutura de dados, etc.) utilizados pelo algoritmo durante a busca da solução do problema. Por exemplo, com relação ao tempo de execução, há métricas tais como: **tempo para encontrar a melhor solução**; **tempo de execução total**; e **tempo por fase**.

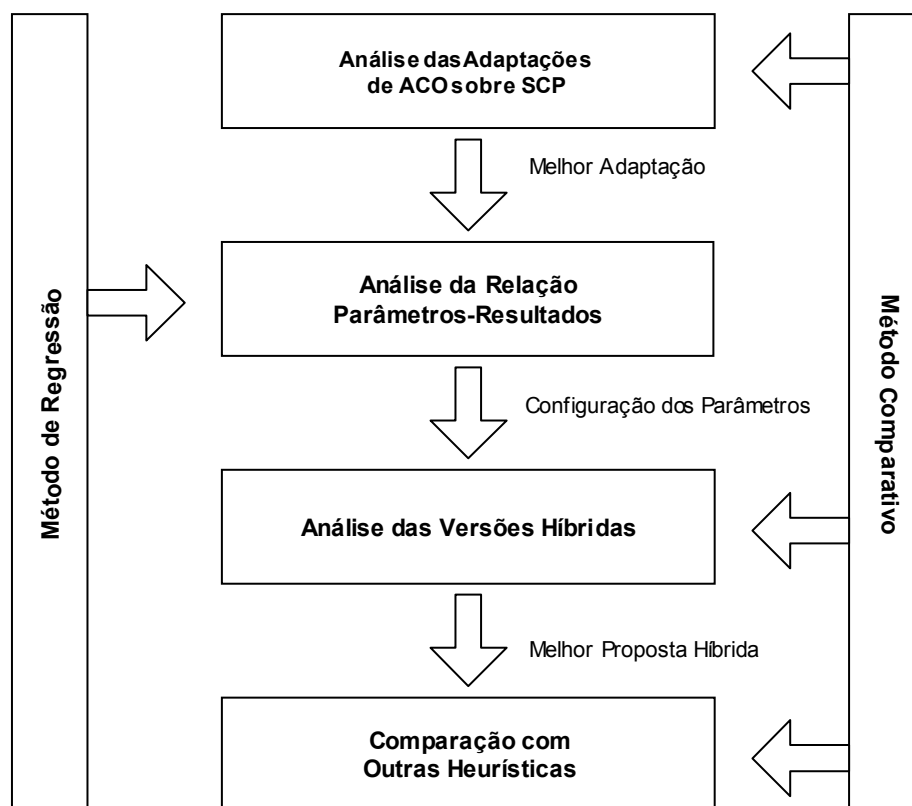


Figura 4: Estratégia de avaliação experimental de *ACO* sobre *SCP*

5.2 MÉTODOS ESTATÍSTICOS COMPARATIVOS

O esquema de decisão empregado na comparação de dois algoritmos encontra-se ilustrado na Figura 5 e na Figura 6. Por sua vez, as conclusões apresentadas neste esquema são descritas em detalhes na Tabela 6.

Basicamente, a idéia deste esquema consiste em avaliar a média e variância da qualidade das soluções e, caso seja necessário, também avaliar a média e variância do tempo decorrido para

obtê-las. O tempo será avaliado apenas quando não houver evidências suficientes para rejeitar a hipótese de igualdade da qualidade média dos resultados dos algoritmos avaliados.

A partir dos resultados oriundos da execução de um projeto experimental previamente estabelecido (Seção 5.4), o referido esquema de decisão segue por três passos distintos:

1. **Estatística Descritiva:** inicia a comparação dos algoritmos via representação pictórica dos resultados. Diagramas de caixa (*box-plot*) referentes à qualidade (ou tempo) associada a estes resultados são montados para cada um dos algoritmos avaliados. Caso haja sobreposição destes diagramas, então ir para o passo 2. Caso contrário, **parar:** a diferença entre a qualidade¹² das soluções propostas por cada algoritmo, ou entre o tempo consumido para alcançar tais soluções, é facilmente perceptível (Conclusão 1 - Figura 5, Descrição #1 -Tabela 6).
2. **Inferência Estatística sobre Duas Médias Populacionais:** havendo sobreposição dos diagramas de caixa, então surge a necessidade da aplicação de algum **método de inferência estatística** para comparação da média da qualidade (ou tempo) associada aos resultados destes algoritmos. Ou seja, sucedida a sobreposição dos diagramas, agora a questão consiste em decidir qual **método de inferência sobre duas médias populacionais** utilizar: teste Z, teste *t*-de-Student (teste T), teste de Mann-Whitney (teste U) (BHATTACHARYYA, G. K.; JOHNSON R. A., 1996) (HOEL, P. G., 1980) (MONTGOMERY, D. C., 1997) (RAMSEY, F. L.; SCHAFER, D. W., 1997). A Figura 6 apresenta o diagrama de decisão que trata desta questão¹³. Independente da indicação proposta por este diagrama, o resultado final de qualquer método de inferência estatística sobre duas médias populacionais consiste em refutar, ou não, hipóteses sobre estas médias. Sendo assim, caso **não** haja evidências suficientes para refutar a hipótese de igualdade da média de qualidade (ou tempo) associada aos resultados dos algoritmos,

¹² A qualidade das soluções é medida pelo desvio percentual, métrica devidamente explanada na seção 6.1.1.

¹³ Cabe observar que este diagrama é um detalhamento do processo intitulado: “Inferência Estatística: Duas Médias”, representado por um retângulo no fluxograma da Figura 5.

então ir para o passo 3. Caso contrário, **parar**: há evidências suficientes para apoiar a conjectura sobre a diferença de qualidade (ou tempo) agregada às soluções propostas por cada algoritmo (Conclusão 2 – Figura 5, Descrição #2 -Tabela 6).

3. **Comparação de Duas Variâncias**: quando **não** há evidências suficientes para refutar a hipótese de igualdade da média da qualidade (ou tempo) associada aos resultados dos algoritmos, então a próxima tarefa consiste em analisar as variâncias populacionais. Ou seja, descobrir se há, ou não, evidências suficientes para apoiar a conjectura sobre a diferença na variância das respostas destes algoritmos (Conclusão 3 - Figura 5, Descrição #3-Tabela 6).

Entretanto, quando há mais de dois algoritmos a serem comparados, então o **método de análise de variância** (ANOVA) (MONTGOMERY, 1996) é empregado. A saber, as suposições em torno da ANOVA são as seguintes: (i) as populações têm distribuição normal; (ii) as populações têm a mesma variância; (iii) as amostras são aleatórias e mutuamente independentes. Porém, se estas suposições em torno da ANOVA não forem satisfeitas, então o **teste de Kruskal-Wallis** é aplicado.

Tabela 6: Descrição das conclusões oriundas do fluxograma da Figura 5

Conclusão	Descrição	Notação
#1	A média (μ) da qualidade (Q), ou do tempo (T), associada às soluções do algoritmo A é melhor (+), ou pior (-), do que a do algoritmo B. ¹⁴	$A \xrightarrow{+/-\mu(Q/T)} B$
#2	A variância (σ^2) da qualidade (Q), ou do tempo (T), associada às soluções do algoritmo A é melhor (+), ou pior (-), do que a do algoritmo B. ¹⁵	$A \xrightarrow{+/-\sigma^2(Q/T)} B$
#3	Não há evidências de que a média (μ) da qualidade (Q) e do tempo (T) associados às soluções do algoritmo A e B não sejam equivalentes (\sim).	$A \xrightarrow{\sim} B$

¹⁴ Quanto menor o desvio percentual, ou o tempo, melhor o resultado.

¹⁵ Quanto menor a variância do desvio percentual, ou do tempo, melhor o resultado.

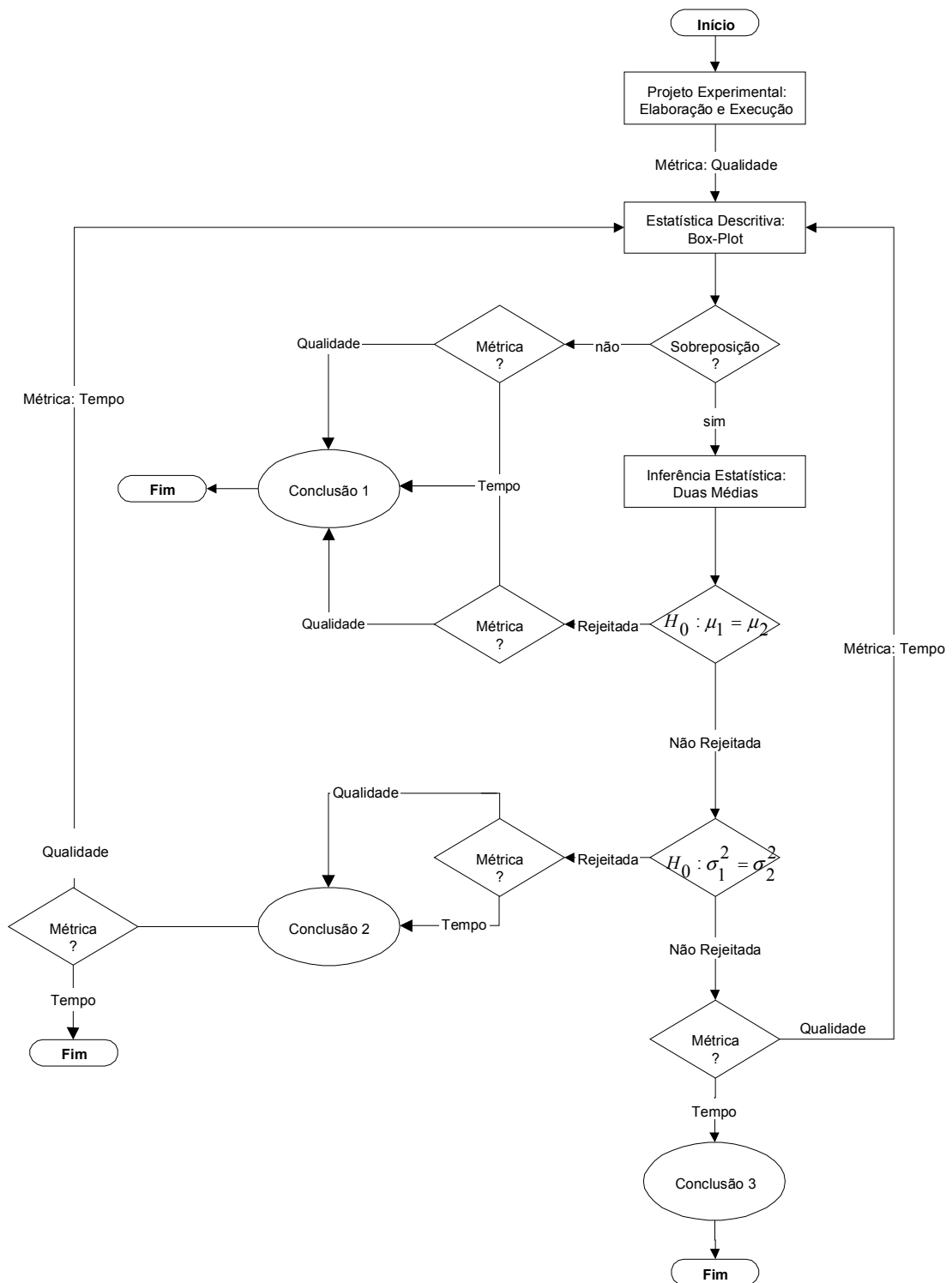


Figura 5: Estratégia para comparação de dois algoritmos.

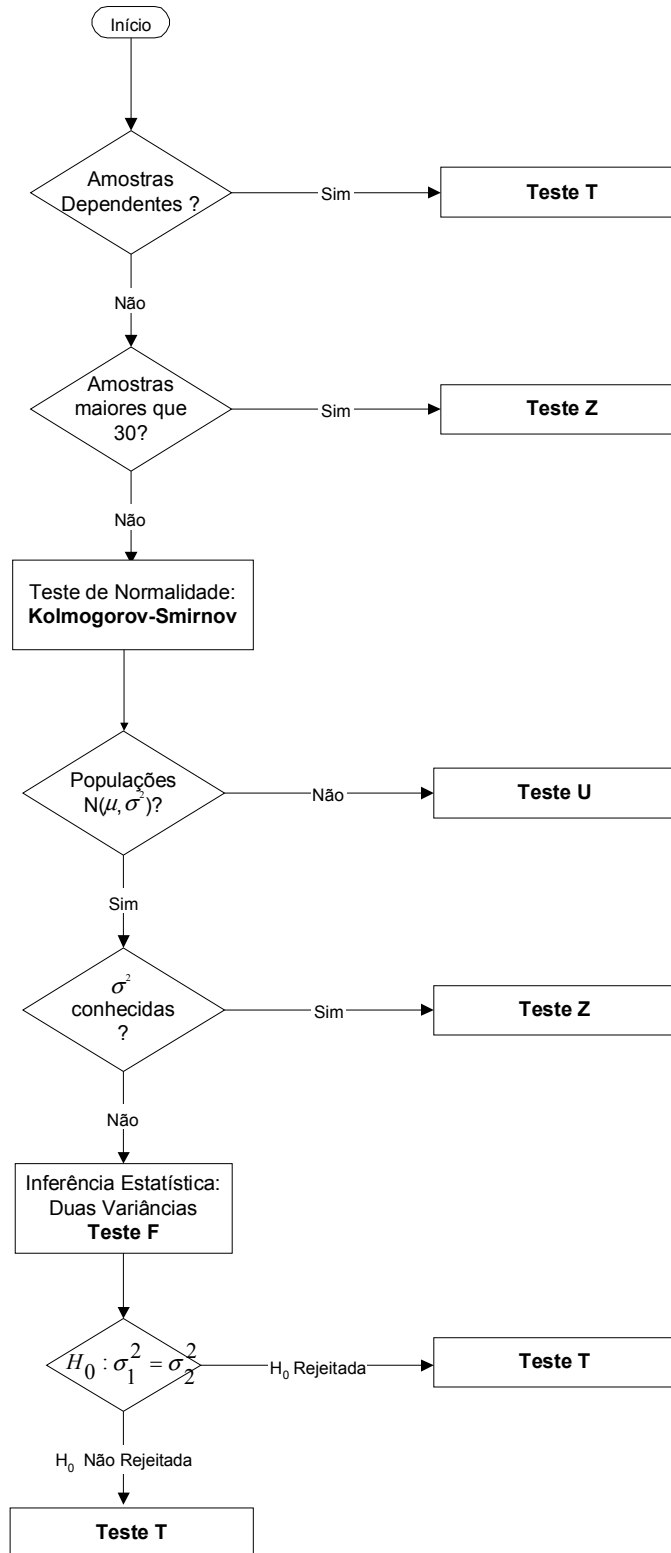


Figura 6: Inferência Estatística: Duas Médias

5.3 MÉTODO DE REGRESSÃO

A tarefa de descobrir relações causais (caso existam) entre valores atribuídos aos parâmetros de um algoritmo e a resposta gerada pelo mesmo, se adequa perfeitamente aos propósitos de uma técnica estatística denominada **análise de regressão**.

A análise de regressão é usada para descrever a distribuição dos valores de uma variável, a **variável resposta**, em função de outras variáveis, as **variáveis explanatórias**, ou **independentes**, ou **preditores**. Ou seja, a regressão de uma variável resposta sobre as variáveis explanatórias é uma relação matemática (**o modelo de regressão**) entre as médias das sub-populações de respostas (cada sub-população correspondente a cada um dos possíveis valores que as variáveis explanatórias possam assumir) e as variáveis explanatórias (RAMSEY; SCHAFER, 1997).

Neste sentido, a Figura 7 ilustra o diagrama do método de regressão empregado neste trabalho. Cabe observar que a partir dos resultados oriundos da execução de um projeto experimental, o referido diagrama segue por cinco passos distintos:

1. **Seleção de Variáveis Explanatórias:** a busca por um conjunto apropriado de preditores pode abranger uma grande variedade de modelos de regressão. Neste sentido, as **técnicas seqüenciais de seleção de variáveis** oferecem uma forma de explorar de modo sistemático alguns (mas não todos) destes modelos. Baseadas no processo de busca por melhoria (*hillclimbing*), a diferença entre estas técnicas seqüenciais de seleção de variáveis reside: (i) na definição de vizinhança utilizada; (ii) no critério de avaliação empregado; e (iii) na escolha do modelo de regressão inicial, donde o processo de busca se inicia. Como exemplo destas técnicas podemos citar: (i) a seleção *forward*; (ii) a eliminação *backward*; e (iii) a regressão *stepwise* (RAMSEY; SCHAFER, 1997);
2. **Análise de Regressão Linear Múltipla:** o objetivo consiste em encontrar um modelo linear múltiplo $Y = \beta_0 f_0(X_1, \dots, X_k) + \beta_1 f_1(X_1, \dots, X_k) + \dots + \beta_n f_n(X_1, \dots, X_k)$, com

$f_j(X_1, \dots, X_k)$ representando funções das variáveis explanatórias selecionadas pelo passo anterior, e β_j representando os **parâmetros** do modelo, também denominados **coeficientes de regressão** - que melhor traduza a relação (caso exista) entre os parâmetros usados no algoritmo e as respostas observadas nos experimentos;

3. **Análise de Resíduos:** após o desenvolvimento do modelo de regressão, as suposições consideradas na análise de regressão (**resíduos independentes, distribuídos normalmente, com média igual a zero e variância σ^2 constante**) devem ser averiguadas.

A independência dos resíduos pode ser verificada através do gráfico dos **resíduos versus a ordem** na qual os resultados foram produzidos. No caso, este tipo de representação pictórica não deve apresentar tendências. Ou seja, o gráfico deve apresentar uma “nuvem de pontos” homogênea.

A hipótese da distribuição normal dos resíduos pode ser averiguada através de dois gráficos: (i) o **histograma** dos resíduos, e (ii) o **gráfico de probabilidade normal** (*normal probability plot* ou *normal-scores plot*). Caso o contorno do histograma se assemelhe ao formato da distribuição normal, e o gráfico de probabilidade normal se aproxime de uma linha reta, então significa que não há evidências suficientes para refutar a hipótese de que a distribuição dos resíduos seja normalmente distribuída.

A suposição de que a variância dos resíduos é constante em todas as sub-populações de respostas (correspondentes a cada um dos possíveis valores que as variáveis explanatórias possam assumir) pode ser averiguada através do gráfico dos **resíduos versus os valores preditos** (*fitted values*) pelo modelo de regressão. Este tipo de representação pictórica também não deve apresentar tendências;

4. **Transformações de Variáveis:** a transformação da variável resposta, e/ou de alguma(s) variável(is) explanatória(s), caracteriza uma tentativa de sanar possíveis violações dos

pré-requisitos da análise de regressão (resíduos independentes, distribuídos normalmente, com média igual a zero e variância σ^2 constante).

5. **Análise de Pontos Extremos (*Outliers*)**: caso existam pontos extremos verificar quais (e por que) influenciam a análise de regressão, na medida em que sua ausência pode alterar o modelo de regressão previamente elaborado. Neste sentido, o presente trabalho adota a estratégia sugerida por Ramsey e Schafer (RAMSEY; SCHAFER, 1997) (Figura 8).

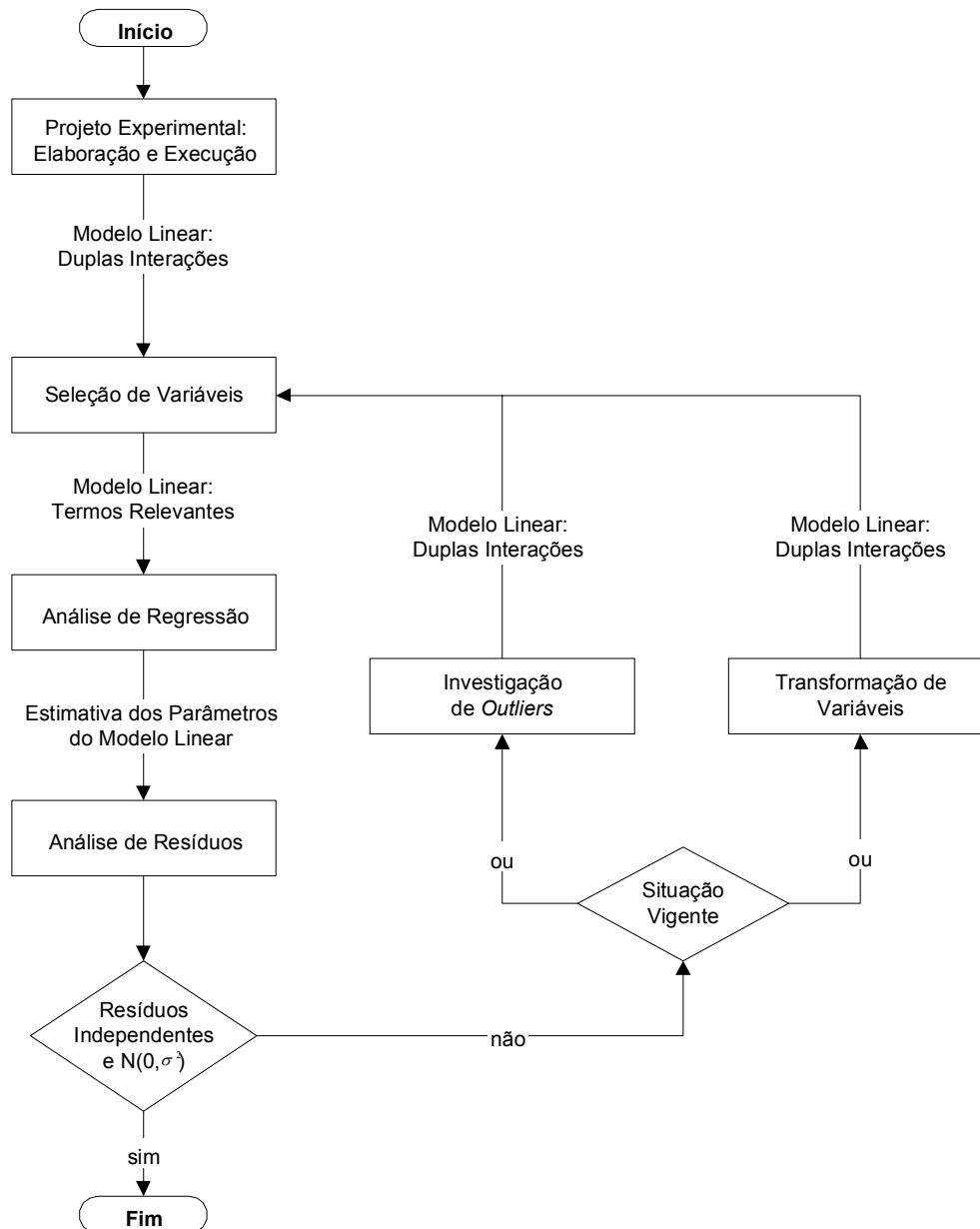


Figura 7: Diagrama da Metodologia de Regressão.

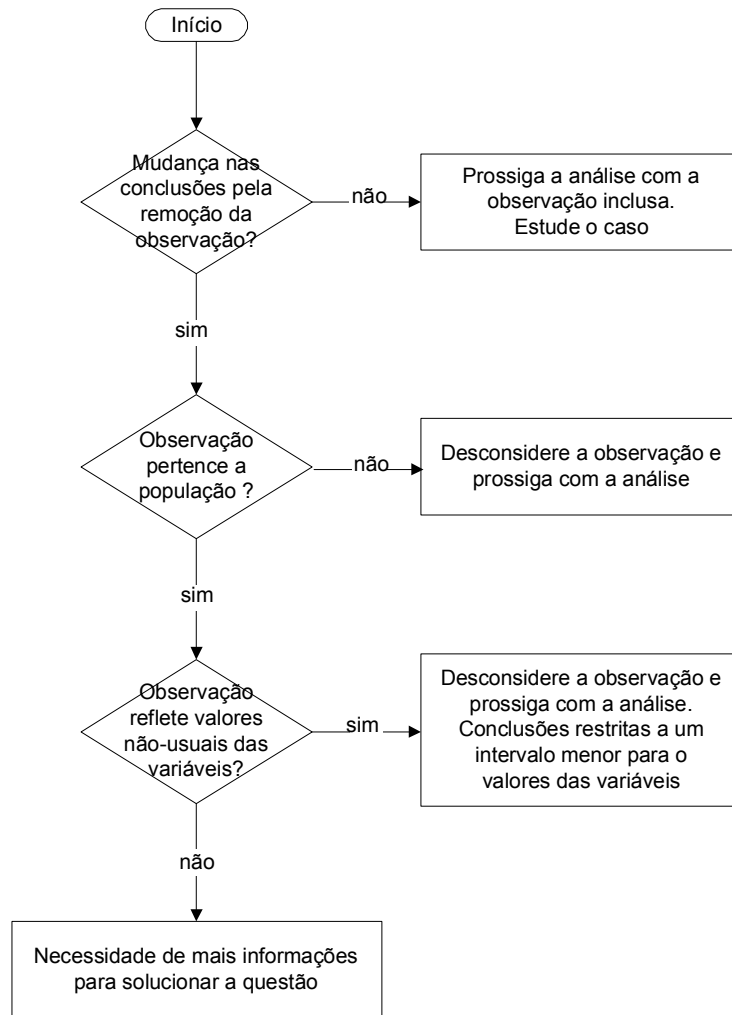


Figura 8: Estratégia para análise de pontos extremos.

5.4 DELINEAMENTO EXPERIMENTAL D-ÓTIMO

Além do método clássico da **blocação aleatória completa** (HICKS, 1993), usado para montar projetos experimentais que comparam heurísticas, o presente trabalho utiliza o **método de delineamento experimental D-ótimo** (ATKINSON; DONEV, 1992) para as análises de regressão.

Para minimizar a soma quadrática das diferenças entre as respostas observadas Y_j , $j = 1, \dots, n$, e suas respectivas estimativas fornecidas pelo modelo de regressão, o **método dos mínimos quadrados** calcula os coeficientes de regressão β segundo a equação:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

cuja variância $Var(\hat{\beta}_j)$ é diretamente proporcional ao inverso da **matriz informação de Fisher**:

$$Var(\hat{\beta}_j) \propto (\mathbf{X}^T \mathbf{X})^{-1}$$

O problema do **delineamento D-ótimo** consiste em maximizar o determinante da matrix de informação de Fisher, pois assim os elementos que compõem sua inversa serão minimizados, logo minimizando a variância dos coeficientes de regressão:

$$\max \det(\mathbf{X}^T \mathbf{X}) \Rightarrow \min (\mathbf{X}^T \mathbf{X})^{-1} \Rightarrow \min Var(\hat{\beta}_j), j = 1, \dots, n.$$

Dito de outro modo, o método do delineamento D-ótimo deve encontrar um projeto experimental (mais especificamente uma **matriz de projeto X**) que maximize o determinante da matriz de informação de Fisher. Tarefa que pode ser realizada através do método de otimização por busca local (Quadro 18).

Considerando os seguintes parâmetros:

- **X** : matriz de projeto atual;
- *IM* : quantidade de iterações máxima;
- *i*: iteração atual;

O método de delineamento D-ótimo (JOHNSON; WICHERN, 1982), a partir da matriz de projeto atual **X**, busca outra matriz de projeto **X'** (Quadro 18, linha 3). Se o determinante da

matriz de informação de Fisher, associada ao projeto \mathbf{X}' , for maior do que o determinante da matriz de informação de Fisher, associada ao projeto \mathbf{X} , então S' torna-se a nova solução atual (Quadro 18, linha 4). Caso contrário, ou seja, se $\det(\mathbf{X}'^T \mathbf{X}') < \det(\mathbf{X}^T \mathbf{X})$, então a matriz de projeto atual \mathbf{X} é mantida. O processo é repetido até que o número de iterações máximo IM pré-estabelecido seja alcançado (Quadro 18, linha 2).

Quadro 18: Heurística LS-SCP de Jacobs e Brusco

1	CONSTRUIR uma matriz de projeto inicial \mathbf{X} ; $i \leftarrow 0$;
2	enquanto $((i < IM))$ faça
3	$i \leftarrow i + 1$; $S' \leftarrow$ BUSCAR (S);
4	se $(\det(\mathbf{X}'^T \mathbf{X}') > \det(\mathbf{X}^T \mathbf{X}))$ então $S \leftarrow S'$;
5	retorne a solução S ;

Fonte: (JACOBS; BRUSCO, 1995).

5.4.1 MÓDULOS CONSTRUIR E BUSCAR

A partir dos seguintes parâmetros:

- L_j : conjunto finito de níveis atribuídos a variável independente X_j , $j = 1, \dots, r$;
- $P = L_1 \times L_2 \times \dots \times L_r$: conjunto de r -uplas representando todas as combinações possíveis entre os níveis das variáveis independentes X_1, \dots, X_r .

O módulo CONSTRUIR (Quadro 18, linha 1) monta uma matriz de projeto inicial \mathbf{X} , selecionando de forma aleatória n r -uplas distintas pertencentes a P . Por exemplo, a i -ésima linha de \mathbf{X} pode ser formada pela i -ésima r -upla sorteada de P .

O módulo BUSCAR monta uma nova matriz de projeto \mathbf{X}' , substituindo uma r -upla qualquer da matriz de projeto atual \mathbf{X} , por outra aleatoriamente selecionada do conjunto formada pelas r -uplas pertencentes a P , mas não presentes em \mathbf{X} .

Uma vez apresentada a estratégia de análise da meta-heurística *ACO* em *SCP*, o próximo capítulo relata os resultados obtidos através das heurísticas explanadas no capítulo 4.

6. RESULTADOS EXPERIMENTAIS

Este capítulo apresenta os resultados da estratégia (Seção 5.1) empregada na avaliação da meta-heurística *ACO*, em instâncias de grande porte do problema de cobertura de conjuntos. Primeiro, a infra-estrutura que suporta esta avaliação é apresentada (Seção 6.1). Depois, os resultados experimentais associados às heurísticas *AS-SCP* (Seções 6.2-3), *MAXMIN-SCP* (Seção 6.4), e suas respectivas hibridizações (Seção 6.5), são relatados. Por fim, a comparação com outras heurísticas é abordada (Seção 6.6). A explicação sobre o motivo da escolha destas heurísticas é uma consequência do rumo que os experimentos tomaram neste trabalho, e que será devidamente relatado no decorrer deste capítulo.

6.1 CONSIDERAÇÕES GERAIS

O objetivo desta seção consiste em apresentar algumas informações concernentes ao estudo da meta-heurística *ACO* em *SCP*, tais como: (i) métricas de desempenho (Seção 6.1.1); (ii) unidades experimentais (Seção 6.1.2); (iii) ambiente computacional (Seção 6.1.3); (iv) heurísticas do enfoque comparativo (Seção 6.1.4); (v) fatores do enfoque causal (Seção 6.1.5); e (vi) relato dos resultados experimentais (Seção 6.1.6).

6.1.1 MÉTRICAS DE DESEMPENHO

As métricas de desempenho adotadas neste trabalho são: (i) o **tempo**, em um intervalo de 10 minutos¹⁶, decorrido desde o final da fase de inicialização da heurística até o encontro de sua melhor solução; (ii) o **desvio percentual** $(Y - Y^*)/Y^*$, com Y representando o custo da melhor solução obtida pela heurística, e Y^* representando o custo da solução ótima do problema (ou o custo da melhor solução até então encontrada na literatura)¹⁷.

O desvio percentual normaliza a diferença entre o custo da melhor solução e o custo da solução ótima, logo tornando possível a comparação de resultados pertencentes a instâncias de SCP totalmente distintas. Nos problemas de minimização, tal como o problema de cobertura de conjuntos, quanto menor for o desvio percentual, melhor a qualidade da solução.

6.1.2 UNIDADES EXPERIMENTAIS

As instâncias do problema de cobertura de conjuntos, utilizadas como unidades experimentais na análise de *ACO* sobre *SCP*, são tiradas da base de dados *scpinfo* (Tabela 7), pertencente à biblioteca *OR-Library* desenvolvida por Beasley (BEASLEY, 1990a).

A **densidade** de uma instância de SCP é definida pela porcentagem de células do tipo $a_{ij} = 1$ presentes na matriz de cobertura $\mathbf{a} = [a_{ij}]$ do modelo primal de SCP (Equação 2.2). Ou seja,

$$\text{Densidade} = 100 \times \left[\frac{\sum_{i=1}^m \sum_{j=1}^n a_{ij}}{(m \times n)} \right] \% .$$

¹⁶ Intervalo maior que o triplo do tempo médio gasto pelas heurísticas apresentadas por Caprara, Fischetti e Toth (2000) (Seção 2.3).

¹⁷ O custo destas soluções é calculado pela função objetivo do problema de otimização abordado. No caso do problema de cobertura de conjuntos, a função objetivo é dada pela equação (2.1) da Seção 2.1.

Sendo assim, visando uma análise experimental sobre os problemas mais difíceis, as instâncias *SCP* selecionadas como unidades experimentais foram as de menor densidade (Tabela 7): *<scp41 a scp410; scp51 a scp510; e scp1 a scp5>*¹⁸.

Tabela 7: Fonte das instâncias de dimensões maiores de *SCP*

Problema	Quantidade de Localidades	Quantidade de Facilidades	Densidade (%)	Instâncias
Scp4	200	1.000	2	<i>Scp4.1 a scp4.10</i>
Scp5	200	2.000	2	<i>Scp5.1 a scp5.10</i>
Scp6	200	1.000	5	<i>scp6.1 a scp6.5</i>
Scpa	300	3.000	2	<i>scpa.1 a scpa.5</i>
Scpb	300	3.000	5	<i>scpb.1 a scpb.5</i>
Scpc	400	4.000	2	<i>scpc.1 a scpc.5</i>
Scpd	400	4.000	5	<i>scpd.1 a scpd.5</i>
Scpe	500	5.000	10	<i>scpe.1 a scpe.5</i>
Scpf	500	5.000	20	<i>scpf.1 a scpf.5</i>
Scpg	1000	10.000	2	<i>scpg.1 a scpg.5</i>
Scph	1000	10.000	5	<i>scph.1 a scph.5</i>

Fonte: (BEASLEY; CHU, 1996).

6.1.3 AMBIENTE COMPUTACIONAL

- *Hardware*: Processador AMD Athlon XP 1700+ com 256 MB de RAM;
- Sistema Operacional: Connectiva Linux 8;

¹⁸ As instâncias *SCP* das classes *Scpc* e *Scpg* não foram consideradas, pois os resultados obtidos não foram satisfatórios para a quantidade de formigas (50, 75, 100, 125 e 150) utilizadas em nossas análises experimentais.

- Linguagem de Programação: Java(TM) 2 Standard Edition, versão 1.4.1;
- Biblioteca de Computação Científica: The Colt Distribution, versão 1.0.3;
- Pacotes Estatísticos: R, MINITAB versão 11.13; S-PLUS 2000 Professional Versão 1 .

6.1.4 HEURÍSTICAS DO ENFOQUE COMPARATIVO

O enfoque comparativo ocorre na primeira, terceira e quarta etapa da estratégia de análise experimental de *ACO* sobre *SCP* (Seção 5.1), cujos tratamentos (heurísticas) encontram-se listados na Tabela 8.

Tabela 8: Tratamentos em cada etapa do enfoque comparativo na avaliação de *AS* sobre *SCP*

Etapas da Estratégia de Análise	Tratamentos (Heurísticas)
1º	<i>AS-Path-SCP-Pure; AS-SCP-Pure;</i> <i>AS-SCP; H-SCP; LS-ECC-SCP</i>
3º	<i>AS-SCP; AS-ECC-SCP; AS-ECC-LSK-SCP;</i> <i>AS-ECC-LSK-LS-SCP; AS-DIV-ECC-LSK-LS-SCP</i>
4º	<i>AS-ECC-LSK-SCP; GA-SCP;</i> <i>SA-ECC-SCP; LS-ECC-SCP</i>

6.1.5 FATORES E NÍVEIS DO ENFOQUE CAUSAL

O **enfoque** causal ocorre na segunda etapa do processo de análise experimental de *ACO* sobre *SCP* (Seção 5.1), cujos fatores, e respectivos níveis, são os seguintes:

- $\alpha \in [1, 2, 3, 4, 5]$: parâmetro que regula a influência do feromônio no processo de decisão probabilística do próximo nó (facilidade) a ser visitado pela formiga;

- $\beta \in [1, 2, 3, 4, 5]$: parâmetro que regula a influência das informações específicas do problema no processo de decisão probabilística do próximo nó (facilidade) a ser visitado pela formiga;
- $\rho \in [0.1, 0.3, 0.5, 0.7, 0.9]$: fator que regula a evaporação do feromônio;
- $Q \in [1.0, 3.25, 5.50, 7.75, 10.0]$: quantidade de feromônio que cada formiga carrega por iteração nas heurísticas baseadas em *Ant System*;
- $n \in [50, 75, 100, 125, 150]$: número de formigas empregadas no processo de otimização;
- $p_{best} \in [0.00005, 0.0005, 0.005, 0.05, 0.5]$: probabilidade da melhor solução até então encontrada ser selecionada após a convergência da heurística *MAXMIN-SCP*.

6.1.6 RELATO DOS RESULTADOS EXPERIMENTAIS

Na literatura é comum que a comparação de dois ou mais algoritmos seja relatada através de tabelas com as melhores respostas geradas pelos algoritmos analisados. Isto ocorre, por exemplo, em todos os trabalhos ilustrados pela Tabela 1 (Seção 2.3) e pela Tabela 5 (Seção 3.4) (SILVA; RAMALHO, 2001a). Entretanto, ao invés de apenas considerar os melhores resultados, o presente trabalho relata a comparação de algoritmos através de **diagramas em caixa** (*boxplot*).

Um diagrama em caixa (Figura 9) é um gráfico constituído de: (a) uma reta que se prolonga do menor ao maior valor observado nos experimentos; (b) um retângulo com retas traçadas no **primeiro quartil** (Q_1), na **mediana** (Q_2), e no **terceiro quartil** (Q_3); e (c) **valores extremos** (*outliers*) que eventualmente possam existir.

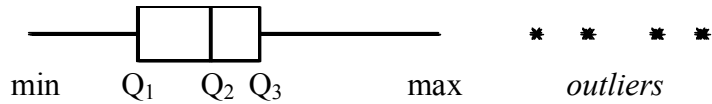


Figura 9: Formato geral de um diagrama de caixa (*boxplot*).

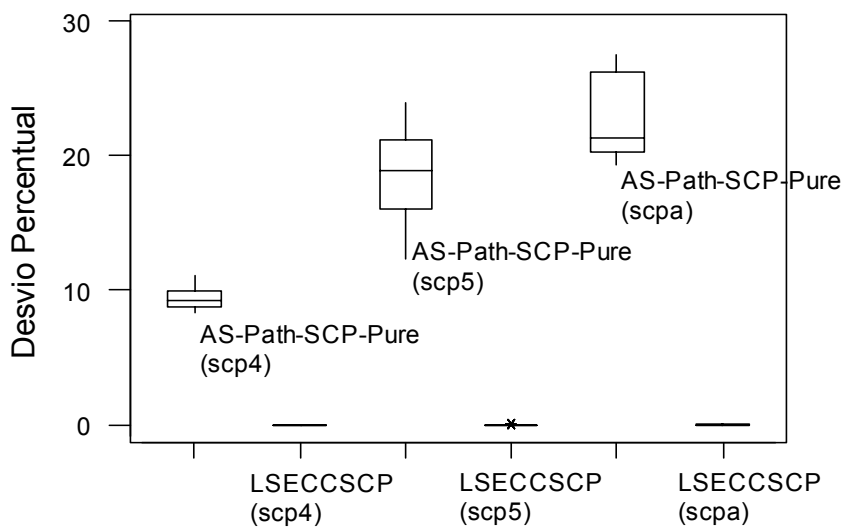
Os três quartis dividem as observações dispostas em ordem crescente em quatro partes iguais. Grosso modo, Q_1 separa os 25% valores inferiores dos 75% superiores; Q_2 separa os 50% inferiores dos 75% superiores; e Q_3 separa os 75% valores inferiores dos 25% superiores. Os valores extremos (*outliers*) são aqueles que estão muito afastados da maioria dos dados.

De acordo com a estratégia de análise experimental adotada neste trabalho (Seção 5.1), quando há sobreposição dos diagramas de caixa, significa que os métodos estatísticos comparativos (Seção 5.2) devem ser realizados. Para facilitar a leitura deste trabalho, apenas alguns destes testes são apresentados neste documento, e de forma reduzida. Ou seja, não estão descritos nesta tese os testes relativos às suposições em torno das inferências estatísticas, tais como, teste de normalidade, teste de homocedasticidade, teste de igualdade de variância, entre outros. Entretanto, todos os detalhes encontram-se dispostos no seguinte endereço: <http://www.comp.ufla.br/~rmas/tese>.

Com relação aos métodos de regressão (Seção 5.3), usados para descobrir a influência dos parâmetros sobre os resultados de um algoritmo, também apresentamos apenas algumas análises resumidas. Entretanto, todos os detalhes encontram-se dispostos no seguinte endereço: <http://www.comp.ufla.br/~rmas/tese>.

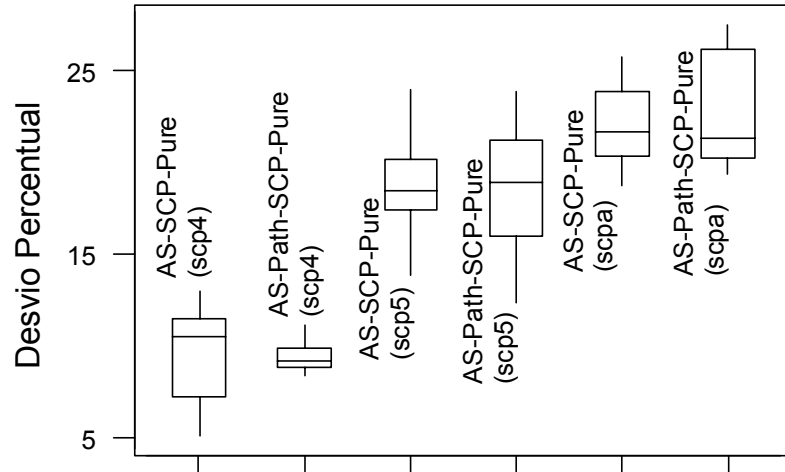
6.2 ANÁLISE DE AS SOBRE SCP

A primeira etapa da análise de AS em SCP (Seção 5.1) se inicia pela heurística *AS-Path-SCP-Pure* (Seção 4.1), cujos diagramas de caixas, relativos aos desvios percentuais de suas soluções, encontram-se distantes (Figura 10) dos diagramas de caixas próximos a zero, apresentados pelo algoritmo



Da mesma forma, os desvios percentuais das soluções da heurística *AS-SCP-Pure* (Seção 4.2) também encontram-se distantes (Figura 11) dos desvios percentuais das soluções de *LS-ECC-SCP*, pois segundo o teste-T, o desvio percentual médio de *AS-SCP-Pure* é equivalente ao desvio percentual médio de *AS-Path-SCP-Pure*. Ou seja, dado que $|T| < t_{0.025}$ nas classes *scp4*, *scp5*, e *scpa* (0.25, 0.24, e 0.39, respectivamente), as evidências não foram suficientes para rejeitar a hipótese de igualdade da qualidade média dos resultados de *AS-Path-SCP-Pure* e *AS-SCP-Pure*.

Apesar de apresentar desvios percentuais menores do que *AS-SCP-Pure* (Figura 12), a heurística *AS-SCP* (Seção 4.2) também não consegue melhores respostas que o algoritmo *LS-ECC-SCP*

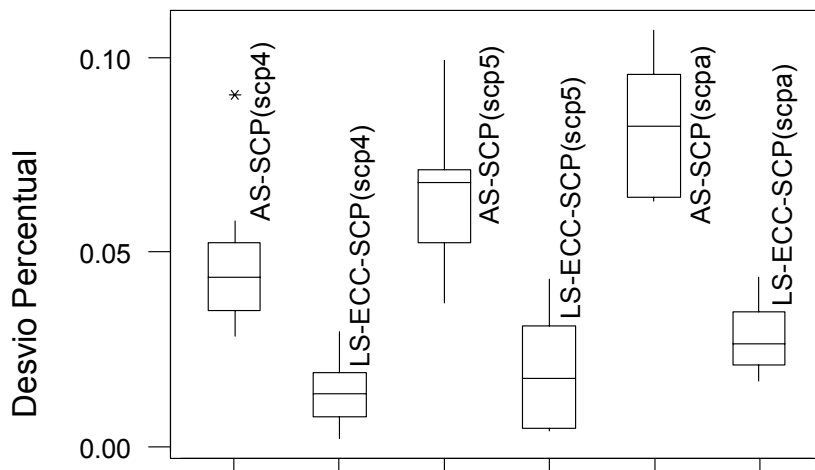
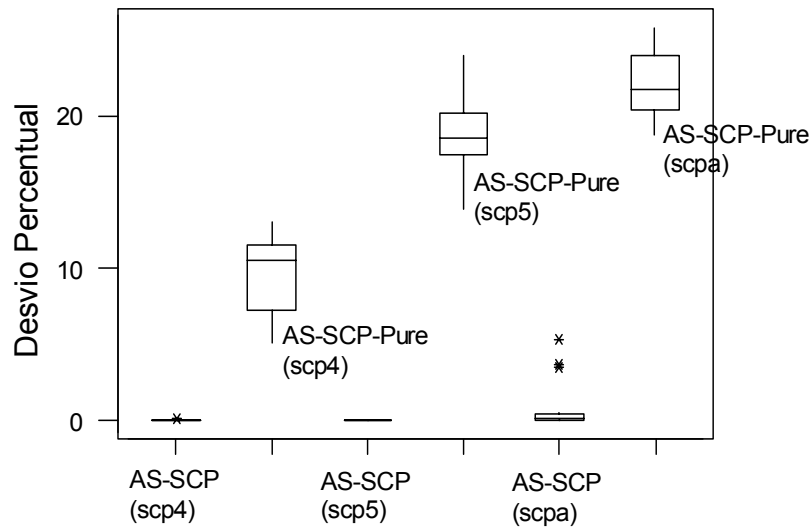


F:

Embora a melhoria obtida por SCP em problemas com grandes instâncias, que os apresentados por *AS-SCP-Pure* (Figura 12), paradoxalmente esta melhora excessiva acaba suscitando a hipótese de que o processo de depósito e evaporação de feromônio da heurística *Ant System*, talvez não seja, *per se*, um fator de aprendizagem muito decisivo em um sistema multiagente proposto a solucionar problemas de cobertura de conjuntos de grande porte.

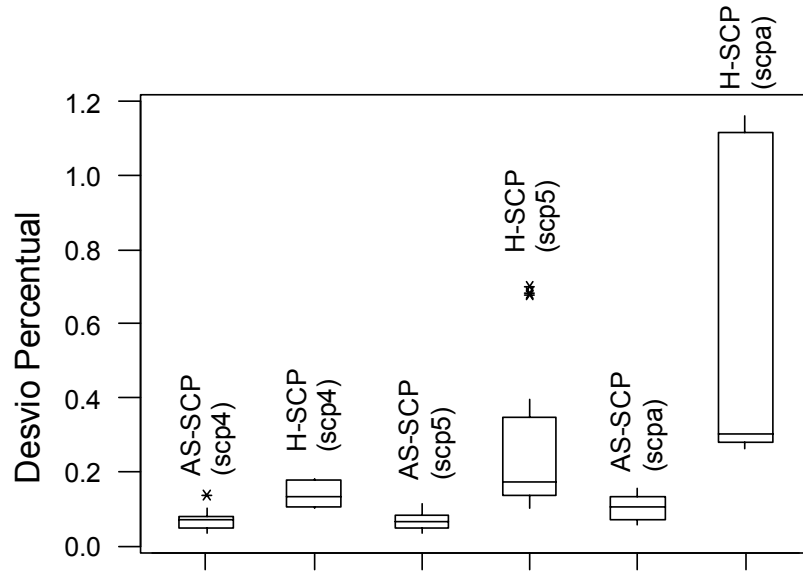
Para testar a hipótese acima citada, a heurística *AS-SCP* é comparada com o algoritmo *H-SCP*: uma alteração realizada sobre *AS-SCP*, onde a simulação do processo de depósito e evaporação de feromônio é descartada ($\alpha=0$). Ou seja, a única informação utilizada na escolha de uma facilidade consiste no parâmetro η_j (Seção 4.2): a quantidade de localidades que a facilidade j pode atender, dentre as localidades que ainda permanecem não-cobertas¹⁹. Caso não haja evidências suficientes para rejeitar a conjectura de igualdade entre os algoritmos *AS-SCP* e *H-SCP*, então a hipótese é fortalecida.

¹⁹ Dado que $H-SCP = AS-SCP_{(\alpha=0)}$, então *H-SCP* não está baseado na meta-heurística *ACO*, pois quando o parâmetro α de *AS-SCP* é igual a zero, o processo de depósito e evaporação de feromônio não é mais utilizado.



A comparação

de *AS-SCP* são menores do que os apresentados por *H-SCP*. Sendo assim, os indícios confirmam alguma relevância no processo de depósito e evaporação de feromônio para um sistema multiagente proposto a solucionar instâncias SCP de grande porte.



Mas mes

evaporação de feromônio em *AS-SCP* (Figura 14), a supressão deste processo em *H-SCP* provoca bem menos alterações nos desvios percentuais das soluções, do que aquelas causadas pela ausência do parâmetro η_j em *AS-SCP-Pure*.(Figura 12).

Este comportamento acaba suscitando uma segunda hipótese, a de que o processo de depósito e evaporação de feromônio da heurística *Ant System* sobre instâncias *SCP* de grande porte, talvez seja significativamente menos relevante do que o parâmetro η_j , que regula a quantidade de localidades que a facilidade j pode atender, dentre as localidades ainda não-cobertas.

Para verificar esta segunda hipótese, a Seção 6.3 emprega o método de regressão (Seção 5.3) sobre a heurística *AS-SCP*, a fim de descobrir qual a relação entre seus parâmetros e os resultados que gera.

6.3 ANÁLISE DA RELAÇÃO PARÂMETROS-RESULTADOS DA HEURÍSTICA *AS-SCP*

O emprego do método descrito na Seção 5.3 sobre a heurística *AS-SCP* em problemas de grande escala resultou, com coeficiente de determinação ajustado em torno de 85.8%, na seguinte equação de regressão²⁰:

$$(\text{desvio percentual})^{-0.67} = -3.51 + 4.07\beta + 2.48\rho + 0.0417\alpha Q - 0.375\beta\rho - 0.000661Qn - 0.493\beta^2.$$

Cabe observar que a influência do parâmetro β no desvio percentual dos resultados da heurística *AS-SCP* é significativamente maior que a apresentada pelo parâmetro α , pois a alteração do desvio percentual, causada quando apenas o parâmetro β é modificado, é maior do que quando apenas α é modificado.

Sendo assim, a última hipótese levantada na seção anterior é reforçada. Ou seja, as evidências indicam que na heurística *Ant System* sobre instâncias *SCP* de grande porte, a quantidade de feromônio presente na facilidade j é significativamente menos relevante do que o número de localidades ainda não-cobertas que esta facilidade é capaz de atender.

Esta hipótese é constatada pela Figura 15 que ilustra o desvio percentual médio, associado aos resultados da heurística *AS-SCP*, em função dos parâmetros α e β . Cabe observar que os piores resultados ocorrem quando β possui os menores valores.

²⁰ O valor de -0.67 para o expoente da variável dependente deste equação foi calculado através do método de Box-Cox (DRAPER, N. R.; SMITH, H., 1973).

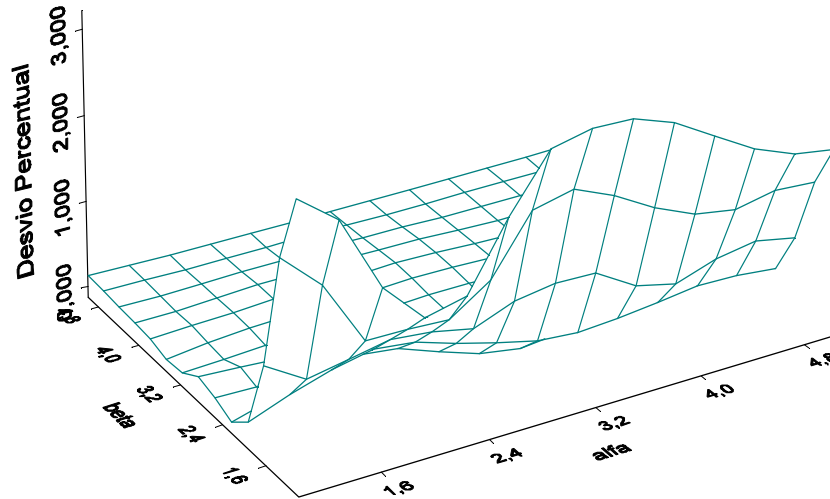


Figura 15: Desvio percentual médio dos resultados de *AS-SCP* em função de α e β .

6.4 ANÁLISE DE *MAX-MIN* SOBRE *SCP*

Uma vez constatada a hipótese de que, na heurística *Ant System* sobre instâncias *SCP* de grande porte, o parâmetro α é significativamente menos relevante do que o parâmetro β (Seção 6.3), resolvemos então aplicar a heurística *MAXMIN Ant System* em *SCP* (Seção 4.2) com o propósito de tentar aumentar a relevância do processo de depósito e evaporação de feromônio.

Para verificar se o propósito acima citado foi alcançado, a presente seção emprega o método de regressão (Seção 5.3) sobre a heurística *MAXMIN-SCP*, a fim de descobrir qual a relação entre seus parâmetros e os resultados que gera.

Com coeficiente de determinação em torno de 91.8% chegamos à seguinte equação de regressão:

$$\begin{aligned} \log(\text{desvio percentual}) = & 1.27 - 1.97\beta - 3.54\rho + 0.0217n - 0.0348\alpha\beta + 0.193\alpha\rho - 0.00102\alpha n \\ & + 0.254\beta\rho - 0.00806\rho n + 0.0279\alpha^2 + 0.206\beta^2 + 2.7\rho^2 - 0.000071n^2 \end{aligned}$$

Cabe observar que a relevância do parâmetro β ainda permanece, porém a influência do parâmetro α na qualidade dos resultados da heurística *MAXMIN-SCP* agora é maior do que a ocorrida na heurística *AS-SCP*. Sendo assim, as evidências indicam que o propósito de aumentar a relevância do feromônio, na resolução de problemas de cobertura de conjuntos de grande porte pela heurística *AS-SCP*, foi de fato alcançado através da heurística *MAXMIN-SCP*.

Mas será que a maior relevância do feromônio em *MAXMIN-SCP* resulta em resultados com menor desvio percentual que os apresentados pela heurística *AS-SCP*? Segundo a Figura 16, apesar do desvio percentual da heurística *MAXMIN-SCP* se aproximar mais da solução ótima em algumas instâncias da classe *scp4* e *scpa*, em média, o desvio percentual de suas soluções encontra-se equivalente à *AS-SCP*, exceto nas instâncias da classe *scpa*.

Dado que o algoritmo *LS-ECC-SCP* possui resultados com desvio percentual menor (Figura 12) do que os apresentados por *AS-SCP*, então por transitividade *LS-ECC-SCP* também possui resultados com desvio percentual menor do que os gerados por *MAXMIN-SCP*. Ou seja, não apenas *AS-SCP*, mas também *MAXMIN-SCP*, não consegue superar a qualidade (calculada pelo desvio percentual) dos resultados de *LS-ECC-SCP*.

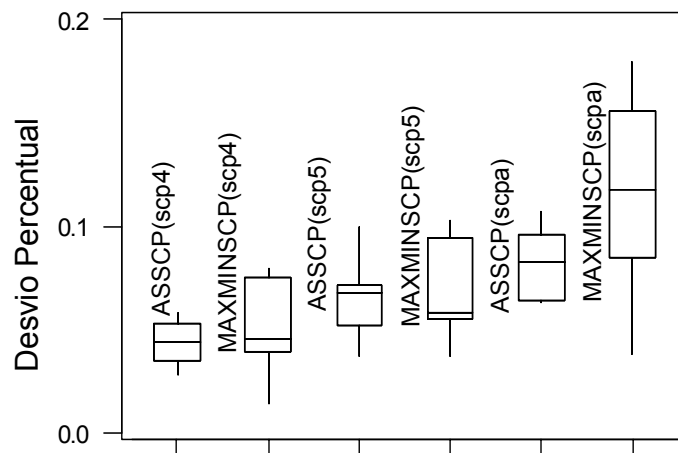


Figura 16: *Boxplots* de *AS-SCP* e *MAXMINSCP* em *scp4*, *scp5* e *scpa*.

6.5 ANÁLISE DE VERSÕES HÍBRIDAS

Devido às heurísticas *AS-SCP* e *MAXMIN-SCP* não terem encontrado soluções com desvios percentuais menores do que os apresentados pelos resultados de *LS-ECC-SCP*, recorreremos então às atividades opcionais previstas pela meta-heurística *ACO* (Seção 3.3), hibridizando as heurísticas *AS-SCP* e *MAXMIN-SCP* (Seções 4.3-4).

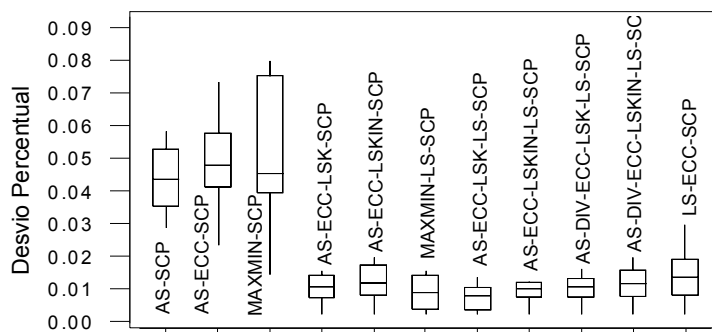
A Figura 17 resume a seqüência de hibridizações avaliadas nesta seção, tendo como referências limítrofes as heurísticas *AS-SCP*, *MAXMIN-SCP* e *LS-ECC-SCP*. Cabe observar que após o advento das versões híbridas das heurísticas *AS-SCP* e *MAXMIN-SCP* (Seções 4.3-4), os principais resultados verificados foram os seguintes:

1. A redução do desvio percentual dos resultados após hibridizações envolvendo o método da busca local é notória (Figura 17). Cabe observar que o desvio percentual das soluções dos algoritmos *ACO* em *SCP* que não possuem o método de busca local são significativamente maiores.
2. A presença do método de busca local na última etapa da heurística proporciona resultados com desvios percentuais menores do que aqueles obtidos em sua ausência. Este é o caso, por exemplo, das heurísticas *AS-ECC-LSK-LS-SCP* e *AS-ECC-LSKIN-LS-SCP*. O desvio percentual de seus melhores resultados é menor do que o desvio percentual das melhores soluções apresentadas por *AS-ECC-LSK-SCP* e *AS-ECC-LSKIN-SCP*, respectivamente. Também se observa que a mediana do desvio percentual de seus resultados é menor do que a mediana do desvio percentual das soluções das heurísticas *AS-ECC-LSK-SCP* e *AS-ECC-LSKIN-SCP*, respectivamente. Entretanto, vale ressaltar que por vezes a média do desvio percentual dos resultados são equivalentes. Isto ocorre, por exemplo, nas instâncias da classe *scp4*.
3. A modificação do local de atuação do método de busca local (em todas as iterações do algoritmo, ao invés da última iteração) resulta nas seguintes observações. O desvio

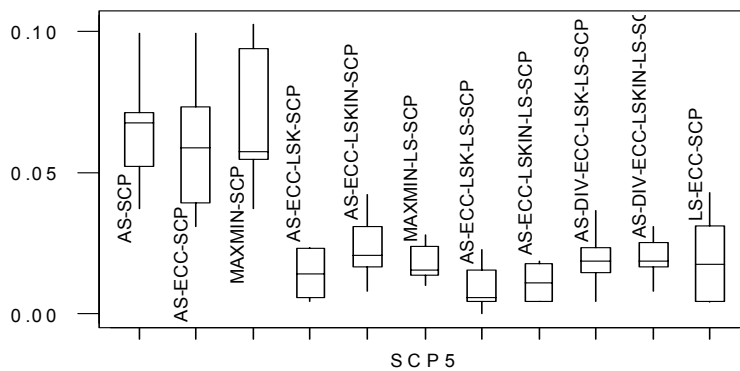
percentual dos melhores resultados das heurísticas *AS-ECC-LSKIN-SCP*, *AS-ECC-LSKIN-LS-SCP*, e *AS-DIV-ECC-LSKIN-LS-SCP* não é menor do que o desvio percentual das melhores respostas apresentadas por *AS-ECC-LSK-SCP*, *AS-ECC-LSK-LS-SCP*, e *AS-DIV-ECC-LSK-LS-SCP*, respectivamente. Exceto com relação a *AS-DIV-ECC-LSKIN-LS-SCP* em *scp5*, também se observa que a mediana do desvio percentual de seus resultados é menor do que a mediana do desvio percentual das heurísticas *AS-ECC-LSK-SCP*, *AS-ECC-LSK-LS-SCP*, e *AS-DIV-ECC-LSK-LS-SCP*, respectivamente. Entretanto, geralmente a média do desvio percentual dos resultados, após a modificação do local de atuação do método de busca local, é equivalente à média do desvio percentual antes desta modificação.

4. O emprego da técnica de diversificação não melhora a qualidade dos resultados das heurísticas já hibridizadas pela técnica de eliminação de colunas redundantes e pelo método da busca local. Isto é o que ocorre, por exemplo, com as heurísticas *AS-DIV-ECC-LSK-LS-SCP* e *AS-DIV-ECC-LSKIN-LS-SCP*. O desvio percentual de seus melhores resultados não é menor do que o desvio percentual das melhores respostas apresentadas por *AS-ECC-LSK-LS-SCP* e *AS-ECC-LSKIN-LS-SCP*, respectivamente. Também se observa que a mediana do desvio percentual de seus resultados é maior do que a mediana do desvio percentual dos resultados das heurísticas *AS-ECC-LSK-SCP* e *AS-ECC-LSKIN-SCP*, respectivamente. Entretanto, vale ressaltar que por vezes a média do desvio percentual dos resultados são equivalentes. Isto ocorre, por exemplo, nas instâncias da classe *scp4*.
5. A média do desvio percentual dos resultados da heurística *LS-ECC-SCP* desenvolvida por Jacobs e Brusco (1995) (Seção 2.4) mostra-se equivalente, ou menor, à média do desvio percentual das respostas das versões híbridas das heurísticas *AS-SCP* e *MAXMIN-SCP* (Seções 4.3-4).

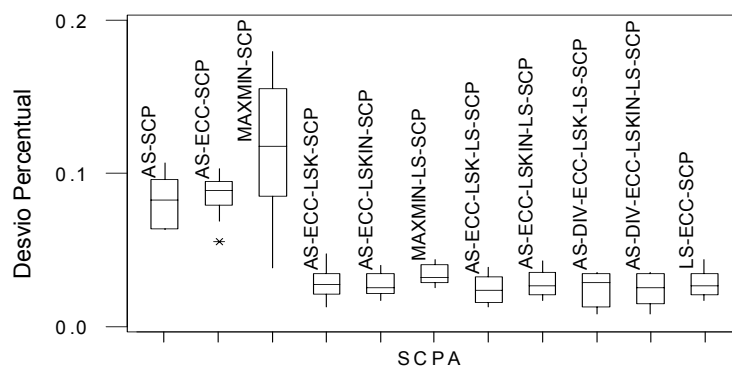
Vale a pena frisar que o tempo limite de 10 minutos (Seção 6.1.1) destinado aos experimentos não foi suficiente para a utilização do método da busca local, situado na última iteração da heurística *AS-ECC-LSKIN-LS-SCP*, sobre as instâncias SCP pertencentes à classe *scpa*.



(a)



(b)



(c)

Figura 17: *Boxplots* das hibridizações de *AS-SCP* e *MAXMIN-SCP* em (a) *scp4*, (b) *scp5* e (c) *scpa*.

6.6 ANÁLISE DAS COMPARAÇÕES COM OUTRAS HEURÍSTICAS

A Figura 18 ilustra a comparação entre as heurísticas *LS-ECC-SCP*, *AS-SCP* e *GA-SCP*. Em concordância com a revisão realizada por Caprara, Fischetti e Toth (2000) (Seção 2.3), a medida que a escala das instâncias *SCP* aumenta, o desvio percentual dos melhores resultados gerados pela heurística *GA-SCP* de Beasley e Chu (1996) tende a ser menor do que o desvio percentual das melhores respostas oriundas da heurística *SA-SCP* de Jacobs e Brusco (1995). Por sua vez, o desvio percentual dos melhores resultados da heurística *SA-SCP* tende a ser menor do que o desvio percentual dos melhores resultados gerados pela heurística *LS-ECC-SCP* de Jacobs e Brusco (1995).

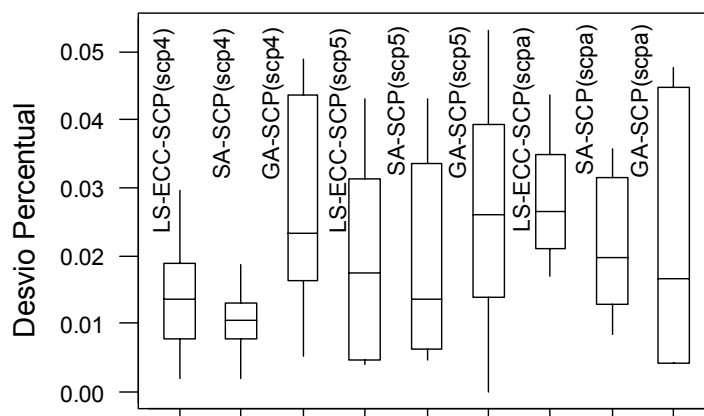


Figura 18: *Bo*

pa.

Cabe também observar (Figura 18) que, a medida que a escala das instâncias SCP aumenta, a mediana dos desvios percentuais das soluções geradas pela heurística *GA-SCP* tende a ser menor do que a mediana dos desvios percentuais das respostas da heurística *SA-SCP*. Por sua vez, a mediana dos desvios percentuais dos resultados de *SA-SCP* tende a ser menor do que a mediana dos desvios percentuais de *LS-ECC-SCP*.

Uma vez apresentados os resultados da estratégia de análise da meta-heurística *ACO* em problemas SCP de grande porte, o próximo capítulo trata de discutí-los, lançando hipóteses capazes de explicá-los.

7. DISCUSSÃO DOS RESULTADOS

Através do conceito de análise da correlação da distância de adaptação (Seção 7.1), o presente capítulo não apenas aborda (Seção 7.2) o menor desvio percentual alcançado pelos melhores resultados da heurística *MAXMIN-SCP* (Seção 4.4), quando comparados com o desvio percentual das melhores soluções da heurística *AS-SCP* (Seção 4.2), mas também discute (Seção 7.4) a

equivalência em torno da média do desvio percentual dos resultados do algoritmo *LS-ECC-SCP* de Jacobs e Brusco (Seção 2.4), quando comparados com a média do desvio percentual das versões híbridas de *MAXMIN-SCP* e *AS-SCP* (Seções 4.5-6). Além disso, trata das eventuais melhorias dos resultados ocorridas durante o processo de hibridização de *MAXMIN-SCP* e *AS-SCP* (Seção 7.3).

7.1 CORRELAÇÃO DA DISTÂNCIA DE ADAPTAÇÃO EM SCP

A **paisagem de adaptação** (*fitness landscape*) associada a um problema combinatorial é definida (STADLER, 1995): (i) pelo conjunto S de todas as soluções viáveis; (ii) pela função objetivo que atribui a cada $s \in S$ um valor de adaptação (custo) $f(s)$; e (iii) pela métrica $d(s, s')$ da distância entre as soluções $s \in S$ e $s' \in S$.

A **correlação da distância de adaptação** (FDC) é calculada pelo coeficiente ρ_{CD} de correlação entre os custos das soluções e suas respectivas distâncias ao ótimo global mais próximo. Ou seja, a partir das variáveis aleatórias C e D (respectivamente representando a adaptação de m soluções e suas correspondentes distâncias ao ótimo global mais próximo), $-1 \leq \rho_{CD} \leq 1$ é obtido pela seguinte equação:

$$\rho_{CD} = \frac{\sigma_{CD}^2}{\sigma_C \sigma_D}$$

onde σ_{CD}^2 representa a covariância entre C e D ; σ_C , o desvio padrão associado a C ; e σ_D , o desvio padrão de D .

Para o problema de cobertura de conjuntos, a correlação da distância de adaptação adotada por Finger, Stützle e Lourenço (2002) foi calculada da seguinte forma:

$$\rho_{CD} = \frac{\frac{1}{m} \sum_{i=1}^m (c_i - \bar{c})(d_i - \bar{d})}{\sqrt{\frac{1}{m} \sum_{i=1}^m (c_i - \bar{c})^2} \sqrt{\frac{1}{m} \sum_{i=1}^m (d_i - \bar{d})^2}}.$$

Onde:

- $C = \{c_1, \dots, c_m\}$ é o conjunto de custos associados a m soluções $s \in S$, e $D = \{d_1, \dots, d_m\}$ suas respectivas distâncias ao ótimo global mais próximo;
- $d(s, s') = \max(|s|, |s'|) - n(s, s')$ é a distância entre duas soluções $s, s' \in S$ viáveis em uma determinada instância *SCP*, sendo $n(s, s')$ a quantidade de facilidades pertencentes a ambas soluções s e s' , e $|s|$ o número de facilidades presentes na solução $s \in S$. Ou seja, se $d(s, s') = \max(|s|, |s'|)$ logo $s, s' \in S$ não possuem facilidades em comum, enquanto se $d(s, s') = 0$ então $s = s'$;
- $\bar{c} = \frac{1}{m} \sum_{i=1}^m c_i$ é a média de custo e $\bar{d} = \frac{1}{m} \sum_{i=1}^m d_i$ a média das distâncias.

Em seguida, através de um método de busca local, Finger, Stützle e Lourenço (2002) geraram $m = 1000$ soluções viáveis para cada instância *SCP* da biblioteca *OR-Library* (BEASLEY, 1990a), com o propósito de calcular a correlação da distância de adaptação para cada uma delas (Tabela 9)²¹.

Tabela 9: Correlação da distância de adaptação em instâncias *SCP* da biblioteca *OR-Library*.

Inst.	ρ_{CD}	Inst.	ρ_{CD}	Inst.	ρ_{CD}	Inst.	ρ_{CD}	Inst.	ρ_{CD}
A1	0.557	B4	–	D2	0.6306	E5	0.7602	G3	0.4355
A2	0.689	B5	–	D3	0.4653	F1	0.6475	G4	0.3989
A3	0.476	C1	0.5955	D4	0.5522	F2	0.7462	G5	0.4897

²¹ Finger, Stützle e Lourenço (2002) não calculam o FDC das instâncias *SCP* pertencentes às classes *scp4* e *scp5*.

A4	0.652	C2	0.5741	D5	0.7868	F3	0.4428	H1	0.3646
A5	0.602	C3	0.1355	E1	0.7202	F4	0.5802	H2	0.5301
B1	–	C4	0.6400	E2	0.2944	F5	0.2661	H3	0.4107
B2	–	C5	0.7409	E3	0.4699	G1	0.7667	H4	0.5582
B3	–	D1	0.7238	E4	0.5516	G2	0.5016	H5	0.6344

Fonte: (FINGER; STUTZLE; LOURENÇO, 2002).

Através da análise FDC das instâncias SCP pertencentes à biblioteca OR-Library (Tabela 9), a próxima seção aborda a melhoria verificada (Seção 6.4) sobre a heurística *AS-SCP* (Seção 4.2), operada por *MAXMIN-SCP* (Seção 4.4).

7.2 SOBRE AS HEURÍSTICAS *AS-SCP* E *MAXMIN-SCP*

O FDC é um importante critério de investigação sobre a importância da qualidade da solução na estratégia de otimização combinatorial. Isto ocorre porque quando $\rho_{CD} \rightarrow +1$, mais próximos os ótimos locais estarão dos ótimos globais, o que torna os custos das soluções cada vez mais aptos para orientar a busca por melhores soluções. Afinal, quanto melhor for este custo, é sinal de que estamos mais próximos de um ótimo global.

A heurística *AS-SCP* usufrui da alta correlação positiva das instâncias *OR-Library* (Tabela 9) apenas em seu processo de depósito de feromônio. Neste processo, a quantidade de feromônio depositada em uma facilidade está diretamente relacionada à qualidade da solução da qual fez parte. Ou seja, quanto melhor for a qualidade da solução, maior a quantidade de feromônio depositada nas facilidades que a compõe. Logo, maior a probabilidade dessas facilidades serem novamente escolhidas, o que aumenta as chances de uma solução ótima global ser encontrada na vizinhança próxima.

A heurística *MAXMINSCP* explora melhor a alta correlação positiva das instâncias *OR-Library* do que *AS-SCP*, pois em cada iteração de *MAXMINSCP* apenas uma única formiga realiza o depósito de feromônio: aquela que encontrou a melhor solução da iteração atual (*iteration-best*

ant), ou então a que encontrou a melhor solução desde o início da execução (*global-best ant*). Sendo assim, o reforço de feromônio sobre as facilidades que compõem as melhores soluções é maior do que o ocorrido em *AS-SCP*. Logo, a probabilidade de grande parte das facilidades que formam as melhores soluções ser escolhida, é maior em *MAXMINSCP* do que em *AS-SCP*. Situação que aumenta as chances de uma solução ótima global ser encontrada na vizinhança próxima às melhores soluções.

Desta forma, a heurística *MAXMINSCP* deve alcançar melhores resultados que *AS-SCP*, em instâncias *SCP* com alta correlação positiva. Hipótese esta confirmada quando a qualidade das soluções de *MAXMINSCP* foi comparada com os resultados de *AS-SCP* (Figura 16). Entretanto, vale frisar que a média da qualidade das soluções apresentadas por ambas mostrou-se estatisticamente equivalente (Seção 6.4).

7.3 SOBRE AS VERSÕES HÍBRIDAS DE *AS-SCP* E *MAXMIN-SCP*

Com o advento das versões híbridas das heurísticas *AS-SCP* e *MAXMIN-SCP* (Seções 4.3-4), os principais resultados (Figura 17) verificados na Seção 6.5 foram os seguintes:

1. A melhoria da qualidade dos resultados, após hibridizações com a busca local, é notória.
2. A presença do método de busca local na última etapa da heurística proporciona melhores resultados do que aqueles obtidos em sua ausência.
3. A modificação do local de atuação do método de busca local (em todas as iterações do algoritmo, ao invés da última iteração) não melhora a qualidade dos resultados.
4. O emprego da técnica de diversificação não melhora a qualidade dos resultados das heurísticas já hibridizadas pela técnica de eliminação de colunas redundantes e pelo método da busca local.

Em suma, o método da busca local aparece como principal responsável pela melhoria da qualidade dos resultados gerados pelas heurísticas *AS-SCP* e *MAXMIN-SCP*.

Segundo o *modus operandi* da busca local (Seção 2.4), algumas facilidades pertencentes a S são descartadas aleatoriamente, com o intuito de substituí-las apenas por facilidades que melhorem a função objetivo de *SCP* (Equação 2.1). Esta substituição ocorre de forma determinística, escolhendo-se a facilidade de menor custo que mais atende as localidades ainda não cobertas. Caso esta troca não resulte em uma solução viável para *SCP*, o processo de substituição continua. Caso contrário, a busca local é reiniciada, mas agora a partir da nova solução encontrada, e assim sucessivamente até que o limite máximo de iterações permitidas acabe.

De acordo com a Seção 7.2, a alta correlação positiva (Tabela 9) das instâncias *SCP* usadas como unidades experimentais (Seção 6.1.1) significa que os custos das soluções estão cada vez mais aptos a orientar a busca por melhores soluções, pois quanto melhor for este custo é sinal de que há um ótimo global próximo.

Portanto, a eficiência em explorar a vizinhança próxima a uma determinada solução S ; combinada ao fato das instâncias *SCP* usadas como unidades experimentais (Seção 6.1.1) possuírem em média uma alta correlação positiva (Tabela 9), fazem com que a busca local seja capaz de superar a qualidade dos resultados gerados pelas heurísticas *AS-SCP* e *MAXMIN-SCP*.

Desta forma, a presença do método de busca local na última etapa da heurística proporciona melhores resultados do que aqueles obtidos em sua ausência, porque há uma continuidade da busca local realizada na última iteração do algoritmo. Ou seja, aquela solução inicial (fornecida pela colônia de formigas) que se transformou na melhor solução até então encontrada (após a aplicação da primeira busca local), agora será contemplada com mais iterações da busca local final.

O raciocínio do parágrafo acima também se aplica à busca local que ocorre a cada iteração do algoritmo. Ou seja, a melhor solução até então encontrada (após a aplicação da busca local em

todas as iterações do algoritmo), agora também será contemplada com mais iterações da busca local final.

Entretanto, a modificação do local de atuação do método de busca local (ao invés da última iteração, agora em todas as iterações do algoritmo) não melhora a qualidade dos resultados, pois mesmo a busca local atualizando as soluções inicialmente encontradas pelas formigas a cada iteração do algoritmo, um certo tempo é necessário para que o depósito de feromônio surta efeito. Ou seja, é preciso um certo tempo para que haja uma melhoria da qualidade das soluções iniciais passadas à busca local pelas formigas a cada iteração do algoritmo. Inclusive, este tempo não apenas pode ser maior do que o limite de 10 minutos (Seção 6.1.1) previamente estipulado, como também há menos tempo disponível para as atividades realizadas pelas formigas. Afinal de contas, a cada iteração existe uma busca local por formiga que consome tempo.

Por fim, o emprego da técnica de diversificação não melhora a qualidade dos resultados das heurísticas já hibridizadas pela técnica de eliminação de colunas redundantes e pelo método da busca local, pois as soluções encontradas pelas formigas têm uma qualidade inferior, visto que a maior parte das facilidades que a compõe foi escolhida aleatoriamente. Conseqüentemente, a busca local não partirá de soluções com boa qualidade.

7.4 SOBRE A COMPARAÇÃO COM A BUSCA LOCAL DE JACOBS E BRUSCO

Apesar do desvio percentual dos melhores resultados obtidos por grande parte das versões híbridas das heurísticas *AS-SCP* e *MAXMIN-SCP* (Seções 4.3-4) serem menores (Seção 6.5) do que o desvio percentual do melhor resultado da heurística *LS-ECC-SCP* desenvolvida por Jacobs e Brusco (1995) (Seção 2.4, Quadro 1). Apesar da mediana do desvio percentual dos resultados obtidos por grande parte das versões híbridas das heurísticas *AS-SCP* e *MAXMIN-SCP* também serem menores do que a mediana do desvio percentual dos resultados da heurística *LS-ECC-SCP*.

A média do desvio percentual dos resultados da heurística *LS-ECC-SCP* mostrou-se equivalente, ou menor, à média do desvio percentual das soluções das versões híbridas das heurísticas *AS-SCP* e *MAXMIN-SCP*.

Isto significa que no contexto das instâncias *SCP* de grande porte (1000 a 3000 facilidades), pertencentes à biblioteca *OR-Library* desenvolvida por Beasley (1990a), a qualidade das soluções encontradas pelas formigas, que por sua vez servirão de solução inicial ao método de busca local, não supera a qualidade obtida pelo módulo CONSTRUIR (Seção 2.4.1) da heurística *LS-ECC-SCP*.

8 CONCLUSÃO

Nesta tese analisamos, através de um método experimental (Capítulo 5), a meta-heurística *ACO* (Seção 3.3) em instâncias de grande porte do problema de cobertura de conjuntos (Seção 2.1). O estudo envolveu adaptações para *SCP* da heurística *Ant System* (DORIGO, M., 1992) (Seção 3.2) e de duas de suas extensões: *MAX-MIN Ant System* (STUTZLE, T.; HOSS, H., 1998) (Seção 3.5) e *ACS-SCP* (HADJI, R., et al., 2000) (Seção 4.4).

Contrariamente a resultados anteriores (HADJI et al 2000), que mostram o bom desempenho de uma versão híbrida da heurística *Ant System* para *SCP* (chamada *ACS-SCP*), as evidências que encontramos indicam, entre outras coisas, que o maior responsável por este desempenho é o método de busca local contido em *ACS-SCP*. Tanto que se procedermos tal como Jacobs e Brusco (1995) em seu algoritmo *LS-ECC-SCP* (Seção 2.4), ou seja, se aplicarmos apenas o método de busca local contido em *ACS-SCP* (similar à heurística *LS-ECC-SCP*), o resultado (Capítulo 6) acaba sendo que a média do desvio percentual das soluções obtidas é equivalente à média do desvio percentual das soluções da própria heurística *ACS-SCP*. Como a heurística *LS-ECC-SCP* é bem mais simples que *ACS-SCP*, não se recomenda o uso desta última.

8.1 RESUMO

Para chegar a conclusão acima citada, uma série de testes foram realizados. Primeiro adaptamos, segundo os cânones adotados pela comunidade *ACO*, a heurística *Ant System* para *SCP* de acordo com as seguintes variações: (i) depósito de feromônio sobre os caminhos (*AS-Path-SCP-Pure*), (ii) depósito de feromônio sobre as facilidades (*AS-SCP-Pure*), (iii) similar ao trecho de *ACS-*

SCP (HADJI, et al. 2000) correspondente às formigas, ou seja, utilização de informações específicas do problema e depósito de feromônio sobre as facilidades (*AS-SCP*), e por último, (iv) a própria exclusão do processo de depósito e evaporação de feromônio (*H-SCP*). Nesta primeira fase, as evidências indicaram que a quantidade de feromônio presente nas facilidades é significativamente menos relevante do que as informações específicas do problema. Além disso, verificamos que nenhuma destas adaptações foram capazes de superar a qualidade dos resultados oriundos do método de busca local *LS-ECC-SCP*.

As evidências da maior relevância das informações específicas ao problema foram reforçadas através da análise de regressão da heurística *AS-SCP* (Seção 6.3). Sendo assim, com o propósito de aumentar a relevância do processo de depósito de feromônio, resolvemos então aplicar a heurística *MAX-MIN Ant System* em *SCP* (*MAXMIN-SCP*).

Mesmo com as evidências da análise de regressão de *MAXMIN-SCP* indicando que o propósito acima citado foi alcançado, esta heurística também não foi capaz de superar a qualidade dos resultados oriundos do método de busca local *LS-ECC-SCP*.

Devido às heurísticas *AS-SCP* e *MAXMIN-SCP* não terem encontrado soluções com desvios percentuais menores do que os apresentados pelos resultados de *LS-ECC-SCP*, recorreremos então às atividades opcionais previstas pela meta-heurística *ACO* (Seção 3.3), ou seja, partimos para as hibridizações das heurísticas *AS-SCP* e *MAXMIN-SCP* (Seção 4.3 e 4.4).

As hibridizações, inspiradas na heurística *ACS-SCP*, foram montadas através de combinações entre os métodos de eliminação de colunas (facilidades) redundantes (*ECC*), diversificação (*DIV*), *Ant System*, *MAX-MIN Ant System*, e busca local (*LS*). Entretanto, apesar da significativa melhoria da qualidade dos resultados propiciada pela utilização da busca local, nenhuma foi capaz de superar a média do desvio percentual dos resultados da heurística *LS-ECC-SCP*.

Após todos os testes acima citados, resolvemos então aplicar apenas o método da busca local (similar ao algoritmo *LS-ECC-SCP*) contido na heurística *ACS-SCP*. Observamos que a média do desvio percentual das soluções obtidas foi equivalente à média do desvio percentual das soluções

da própria heurística *ACS-SCP*. Porém, como a heurística *LS-ECC-SCP* é bem mais simples que *ACS-SCP*, não se recomenda o uso desta última.

8.2 PRINCIPAIS DIFICULDADES ENCONTRADAS

Uma das principais dificuldades encontradas neste trabalho foi a multidisciplinaridade exigida no processo de avaliação experimental de heurísticas de otimização. Basicamente, o analista deve dominar áreas tais como: análise, projeto, implementação e simulação de algoritmos; problemas NP-completos; pesquisa operacional; probabilidade e estatística; e delineamento de projetos experimentais. Além disso, o próprio processo de avaliação é árduo e exige uma infra-estrutura que nem sempre se encontra disponível.

8.3 CONTRIBUIÇÕES

As **principais** contribuições deste trabalho encontram-se listadas abaixo:

1. A revelação da pouca importância do processo de depósito de feromônio – aspecto fundamental do método *ACO* desenvolvido por Dorigo (1992) – nas heurísticas *Ant System* e *MAX-MIN Ant System* (e suas variações) em instâncias de grande porte do problema de cobertura de conjuntos. Mesmo nas versões híbridas destas heurísticas, o processo de depósito de feromônio mostrou-se pouco relevante, sendo o método de busca local baseado no trabalho de Jacobs e Brusco (1995) o principal responsável pela melhoria da qualidade dos resultados, quando comparados aos resultados das versões não-híbridas. Sendo assim, por ser mais simples e eficiente, recomenda-se a utilização da heurística *LS-ECC-SCP* para solução de instâncias de grande porte do problema de cobertura de conjuntos, ao invés das adaptações e hibridizações oriundas das heurísticas *AS-SCP* e *MAXMIN-SCP*.

2. A teoria de que a alta correlação da distância de adaptação das instâncias *SCP* da biblioteca *OR-Library* é a principal responsável pela qualidade inferior das soluções oriundas do método de otimização não-híbrido baseado em formigas, quando comparado com a heurística *LS-ECC-SCP* de Jacobs e Brusco (1995).

Teses que apontam limitações não são habituais, geralmente o que ocorre são novas propostas para solução de problemas antigos ou recentes. Entretanto, as contribuições deste tipo de trabalho também são importantes, como por exemplo, o **teorema da incompletude** de Gödel (GÖDEL, 1992) que prova ser impossível criar um sistema matemático completo e consistente (SINGH, 1999). Neste sentido, guardadas as devidas proporções, as conseqüências diretas oriundas das principais contribuições de nosso trabalho são as seguintes (Capítulo 1):

1. A advertência do provável desperdício de tempo em estudos sobre *ACO* em instâncias *SCP* de grande porte (mais especificamente com relação às heurísticas *Ant System* e *MAX-MIN Ant System*, com suas variações e hibridizações).
2. A recomendação para o desenvolvimento de outros métodos de otimização, ou mesmo pesquisas sobre o comportamento de *ACO* em problemas com características semelhantes.

Já em termos de contribuições **secundárias** temos:

1. A adaptação da heurística *MAXMIN Ant System* em *SCP* para aumentar a relevância do processo de depósito de feromônio.
2. Estudo, através de análises de regressão, da influência dos parâmetros das heurísticas *Ant System* e *MAXMIN Ant System* nos resultados gerados em instâncias de grande porte do problema de cobertura de conjuntos.
3. Constatação da viabilidade da técnica de delineamento D-ótimo na construção dos projetos experimentais, imprescindíveis à análise experimental das heurísticas de otimização.

4. Refutação das conclusões de Hadji, Rahoual, Talbi e Bachelet (2000) sobre o desempenho do método baseado em colônia de formigas em instâncias de grande porte do problema de cobertura de conjuntos.
5. Disponibilização via internet de toda infra-estrutura necessária à reprodução de todos os experimentos realizados neste trabalho.
6. Divulgação da análise experimental como alternativa de avaliação das heurísticas de otimização.

8.4 TRABALHOS FUTUROS

Como trabalhos futuros sugerimos as seguintes pesquisas:

1. Implementação e avaliação experimental da abordagem paralela da meta-heurística *ACO* em *SCP*. Neste sentido, já adaptamos uma versão paralela da heurística *Ant System* sobre *SCP* que podem ser vista no endereço: <http://www.comp.ufla.br/~rmas/tese>.
2. Análise experimental da viabilidade do emprego da meta-heurística *ACO* nas demais classes de problemas de otimização pertencentes à categoria do problema de localização de facilidades, tais como, *maximum covering location problem*, *p-center problem*, *p-median problem*, *fixed charge facility location problem* (DASKIN, M. S., 1995). Neste sentido, já adaptamos *Ant System* sobre os três primeiros problemas.
3. Análise experimental de outros problemas de subconjuntos no intuito de averiguar se as limitações de *ACO* se estendem a outras categorias de problemas. Neste sentido, já iniciamos algumas avaliações sobre o problema da mochila múltipla (*multiple knapsack problem* - MKP) que podem ser vista no endereço: <http://www.comp.ufla.br/~rmas/tese>.
4. Análise experimental do restante das heurísticas desenvolvidas pela comunidade *ACO*.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALASTAIR, J. W. An efficient method for generating discrete random variables with general distributions. **ACM Transactions on Mathematical Software**. v. 3, n. 3, p. 253-256, set. 1977.
- ATKINSON, A. C.; DONEV, A. N. **Optimum experimental designs**. Oxford: Clarendon Press, 1992. (Oxford Statistical Science Series, vol. 8).
- BACK, *apud* BEASLEY, J. E.; CHU, P. C. A genetic algorithm for the set covering problem. **European Journal of Operational Research**. n. 94, p. 392-404, 1996.
- BALAS, E. A Class of location, distribution e scheduling problems: modeling and solution methods. *In*: GRAY, P.; YUANZHANG, L. (Eds.). CHINESE-U.S. SYMPOSIUM ON SYSTEMS ANALYSIS. **Proceedings**. John Wiley and Sons, 1983.
- BALAS, E.; CARRERA, M. C. A dynamic subgradient-based branch-and-bound procedure for set covering. **Operations Research**. n. 44, p. 875-890, 1996.
- BARR, R. S. *et al.* Designing and reporting on computational experiments with heuristic methods. **Journal of Heuristics**. v. 3, n. 3, p 9-32, 1995.
- BEASLEY, J. E. OR-Library: distributing test problems by eletronic mail. **Journal of the Operational Research Society**. n. 41, p. 1069-1072, 1990a.
- _____. A lagrangian heuristic for set-covering problems. **Naval Research Logistics**. n. 37, p. 151-164, 1990b.
- BEASLEY, J. E.; CHU, P. C. A genetic algorithm for the set covering problem. **European Journal of Operational Research**. n. 94, p. 392-404, 1996.
- BHATTACHARYYA, G. K.; JOHNSON R. A. **Statistics: principles and methods**. [S.l]:John Wiley and Sons, 1996.
- BICKEL, P. J.; DOCKSUM, K. A. **Mathematical statistics: basic ideas and selected topics**. NJ: Prentice Hall, 1977. 492 p.
- BRUSCO, M. J.; JACOBS, L. W.; THOMPSON, G. M. A morphing procedure to supplement a simulated annealing heuristic for cost- and coverage-correlated weighted set-covering problems.

Working Paper. Operations Management and Information Systems Department, Northern Illinois University, 1996.

BULLNHEIMER, B.; HARTL, R. F.; STRAUSS C. An improved ant system algorithm for the vehicle routing problem. **Annals of Operation Research.** n. 89, p. 319-328, 1999a.

_____. A new rank version of the ant system: a computational study. **Central European Journal of Operations Research.** v. 1, n. 7, p. 25-38, 1999b.

CAMBRIDGE International Dictionary of English. Cambridge University Press, 1995.

CAPRARA, A.; FISCHETTI, M.; TOTH, P. A heuristic method for the set covering problem. **Operations Research.** v. 47, n. 5, p. 730-743, maio 1999.

_____. Algorithms for the set covering problem. **Annals of Operations Research.** v. 1, n. 98, p. 353-371, dezembro, 2000.

De JONG, *apud* BEASLEY, J. E.; CHU, P. C. A genetic algorithm for the set covering problem. **European Journal of Operational Research.** n. 94, p. 392-404, 1996.

DI CARO, G.; DORIGO, M. AntNet: a mobile agents approach to adaptive routing in communication network. *In:* NINTH DUTCH CONFERENCE ON ARTIFICIAL INTELLIGENCE. 12-13 novembro, Antwerpen, 1997. **Conferência.** Antwerpen, 1997.

CERIA, S.; NOBILI, P.; SASSANO, A. A lagrangian-based heuristic for large-scale set covering problems. **Mathematical Programming.** Série A., v. 81, n. 2, p. 215-228, abril 1998.

_____. Set Covering Problem. *In:* DELL'AMICO, M; MAFFIOLI, F.; MARTELLO, S. (eds.). **Annotated Bibliographies in Combinatorial Optimization.** John Wiley & Sons, 1998.

COHEN, P. R. **Empirical Methods for Artificial Intelligence.** MIT Press, 1995.

COLORNI, A. et al. Ant system for job-shop scheduling. **Belgian Journal of Operations Research, Statistics and Computer Science.** v.1, n. 34, p. 39-53, 1994.

COLORNI, A. et al. Heuristics from nature for hard combinatorial problems. **International Transactions in Operational Research.** vol. 1, n. 3. p. 1-21, 1996.

CORMEN et al. **Introduction to Algorithms.** 2. ed. MIT Press, 2001.

COSTA, D.; HERTZ, A. Ants can colour graphs. **Journal Operational Research Society.** n. 48, p. 295-305, 1997.

- DASKIN, M. S. **Network and discrete location**: models, algorithms, and applications. John Wiley & Sons, 1995.
- DORIGO, M. **Optimization, learning and natural algorithms**. 1992. Tese (Doutorado). Politecnico di Milano, Itália, 1992.
- DORIGO, M.; CARO, G. The ant colony optimization meta-heuristic. *In*: CORNE, D.; DORIGO M.; GLOVER, F. (eds.). **New ideas in optimization**. Londres, Reino Unido: McGraw-Hill, 1999.
- DORIGO, M.; CARO, G.; GAMBARDELLA, L. M. Ant algorithms for discrete optimization. **Artificial Life** . v. 2, n. 5, 1999.
- DORIGO, M.; GAMBARDELLA, L. Ant colony system: a cooperative learning approach to the travelling salesman problem. *In*: IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. **Conferência**. n.1 p. 53-66, 1997.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. Positive feedback as a search strategy. **Relatório técnico** 91-016. Itália: Departamento de eletrônica, Politécnico de Milão, 1991.
- _____. The ant system: optimization by a colony of cooperating agents. *In*: IEEE TRANSACTION SYSTEM MAN AND CYBERNETIC. **Conferência**. Série B, v. 1, n. 26, p. 29-41, 1996.
- DRAPER, N. R.; SMITH, H. **Applied regression analysis**. 2. ed. John Wiley & Sons, 1973.
- FINGER, M.; STUTZLE, T.; LOURENÇO, H. Exploiting fitness distance correlation of set covering problems. *In*: CAGNONI, S. et. al (eds.). **Applications of Evolutionary Computing**. Springer-LNCS. p. 61-71, 2002.
- FRANCIS, L. R.; WHITE, J. A. **Facility, layout and location**. Prentice-Hall, 1974.
- GAMBARDELLA, L. M.; DORIGO, M. HAS-SOP: hybrid ant system for the sequential ordering problem. **INFORMS Journal on Computing**. v. 3, n. 12, p. 237-255, 1997.
- GAMBARDELLA, L. M.; DORIGO M. Ant-Q: a reinforcement learning approach to the travelling salesman problem. *In*: 12TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING. **Proceedings**. Palo alto, Califórnia, p. 252-260, 1995.
- GAMBARDELLA, L. M.; TAILLARD, E. D.; DORIGO, M. Ant colonies for the QAP. **Journal of the Operational Research Society**. n. 50, p. 167-176, 1999.
- GAREY, M. R.; JOHNSON, D. S. **Computers and intractability**: a guide to the theory of NP-completeness. W. F. Freeman & Co., 1979.
- GLEASON, J. M. A set covering approach to bus stop location. **Omega**. n. 3, p. 605-608, 1975.

GÖDEL, K. **On formally undecidable propositions of principia matemática and related systems**. Dover Pubis, reprint edition, 1992.

GOSS, S. et al. Self-organized shortcuts in the argentine ant. **Naturwissenschaften**. n.76, p. 579-581, 1989.

GRAHAM, R. L. Bounds on multiprocessor timing anomalies. **SIAM Journal of Applied Mathematics**. n.17, p.416-429, 1969.

HADDADI, S. Simple lagrangian heuristic for the set covering problem. **European Journal of Operational Research**. p. 200-204, 1997.

HADJI, R. et al. Ant colonies for the set covering problem. *In: ANTS'2000 FROM ANT COLONIES TO ARTIFICIAL ANTS. 2TH INTERNATIONAL WORKSHOP ON ANT ALGORITHMS*. **Proceedings**. P. 63-66, Bruxelas, Bélgica, setembro, 2000.

HELD, M.; WOLFE, P.; CROWDER, H. D. Validation of subgradient optimization. **Mathematical Programming**, 1974.

HICKS, C. R. **Fundamental concepts in the design of experiments**. Oxford: Oxford University Press. 4. ed., 1993.

HOEL, P. G. **Estatística matemática**. [Introduction to Mathematical Statistics]. Trad. de Othon Guilherme Pinto Bravo. Rio de Janeiro: Guanabara Dois S. A., 1980.

HOOKER, J. N. Testing heuristics: we have it all wrong. **Journal of Heuristics**. v. 1, p. 33-42, 1996.

JACOBS, L. W.; BRUSCO, M. J. Note: a local-search heuristic for large set-covering problems. **Naval Research Logistics**. v. 42, p. 129-1140, 1995.

JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling**. John Wiley and Sons, 1991.

JOHNSON, R. A.; WICHERN, D. W. **Applied multivariate statistical analysis**. Prentice Hall, 1982

KARP, R. M. Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. **Mathematics of Operations Research**. n. 2, p. 209-224, 1977.

KUBY, M. et al. Planning china's coal and eletricity delivery system. **Interfaces**. 1995.

LARSON, R. E.; HOSTETLER, R. P.; EDWARDS, B. H. **Brief calculus with applications**. Massachusetts: D. C. Heath and Company. 4. ed., 1995.

- LEGUIZAMÓN, G.; MICHALEWICZ, Z. A new version of ant systems for subset problems. In: CONGRESS ON EVOLUTIONARY COMPUTATION. **Proceedings**. v. 2, p. 1459-1468, 1999.
- LORENA, L. N.; LOPES, F. B. A surrogate heuristic for set covering problems. **European Journal of Operational Research**. n. 79, p. 138-150, 1994.
- MANBER, U. **Introduction to algorithms**: a creative approach. Addison Wesley Publishing Company, 1989.
- MANIEZZO, V.; CARBONARO, A. An ants heuristic for the frequency assignment problem. **Future generation computer systems**. Amsterdam: Elsevier. n. 16, p. 927-935, 2000.
- MANIEZZO, V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. **INFORMS Journal on Computing**. v. 4, n. 11, p. 358-369, 1999.
- MANIEZZO, V.; COLORNI, A.; DORIGO, M. The ant system applied to the quadratic assignment problem. *IEEE transactions on knowledge and data engineering*. v. 11, n. 5, p. 769-778, 1999.
- MATEUS, G. R.; LOUREIRO A. **Introdução à Computação Móvel**. 11^a. Escola de Computação. COPPE/Sistemas, NCE/UFRJ, 1998.
- McGEOCH, C. C. Toward an experimental method for algorithm simulation. **INFORMS Journal on Computing**. v. 1, n. 8, 1999.
- MONTGOMERY, D. C. **Design and analysis of experiments**. 4. ed. John Wiley and Sons, 1997.
- RAMSEY, F. L.; SCHAFER, D. W. **The statistical sleuth**: a course in methods of data analysis. Duxbury Press, 1997.
- RARDIN, R.L. **Optimization in Operations Research**. Prentice-Hall, 1998.
- RARDIN, R. L. UZSOY, R. Experimental evaluation of heuristic optimization algorithms: a tutorial. **Journal of Heuristics**. v. 7, p. 262-304, 2001.
- RUSSELL, S.; NORVIG, P. **Artificial intelligence**: a modern approach. Prentice Hall, 1995.
- SCHILLING, D. A.; JAYARAMAN, V.; BARKHI, R. A review of covering problems in facility location. **Location Science**. v. 1, n. 1, p. 25-55, 1993.
- SILVA, R. M. de A.; RAMALHO, G. L. Going the extra mile ant colony optimization experimental evaluation. *In: 4TH METAHEURISTIC INTERNATIONAL CONFERENCE. Conferência*. 16-20 de julho de 2001a.

SILVA, R. M. de A.; RAMALHO, G. L. Ant system for the set covering problem. *In: INTERNATIONAL CONFERENCE ON SYSTEM, MAN, AND CYBERNETICS (IEEE SMC2001). Conferência.* Tucson, Arizona, EUA, 2001b.

SILVA, R. M. de A.; RAMALHO, G. L. *ASnode-SCP: a new version of the ant system for the set covering problem.* Anais do XXI Congresso da Sociedade Brasileira de Computação. v.1, p. 211, agosto, 2001c.

STADLER, P. F. Towards a theory of landscapes. *In: LOPES-PENA, R., et al. (eds.). Complex systems and binary networks.* Springer-Verlag, 1995.

STUTZLE, T. An ant approach to the flow shop problem. *In: 6TH EUROPEAN CONGRESS ON INTELLIGENT TECHNIQUES & SOFT COMPUTING (EUFIT'98).* Verlag Mainz, Aachen. **Proceedings.** v. 3, p. 1560-1564, 1997.

STUTZLE, T.; HOOS, H. Improvements on the ant system: introducing MAX_MIN ant system. *In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS.* Vienna, Austria. **Proceedings.** p. 245-249, 1998.

_____. MAX-MIN ant system and local search for combinatorial optimization problem. *In: META-HEURISTICS: ADVANCES AND TRENDS IN LOCAL SEARCH PARADIGMS FOR OPTIMIZATION.* Boston, MA: Kluwer Academic. p. 313-329, 1999.

SINGH, S. **O último teorema de Fermat:** a história do enigma que confundiu as maiores mentes do mundo durante 358 anos. [Fermat's last theorem]. Trad. de Jorge Luiz Calife. 6. ed. Rio de Janeiro: Record, 1999.

STUTZLE, T.; DORIGO, M. ACO algorithms for the traveling salesman problem. *In: EVOLUTIONARY ALGORITHMS IN ENGINEERING AND COMPUTER SCIENCE.* Wiley, 1999.

TECHAPICHETVANICH, K.; BRICKER, D. Investigation of lagrangian heuristics for set covering problems. **Relatório Técnico.** Department of Industrial Engineering. University of Iowa, 1993.

TOREGAS, C. et al. The location of emergency service facilities. **Operations Research.** n. 19, p. 1363-1373, 1971.

VERTER, V.; DINCER, M. C. Global manufacturing strategy. *In: DREZNER, Z. (ed.). Facility location: a survey of applications and methods.* p. 263-282, 1995.

WALKER, W. E. Using the set covering problem to assign fire companies to fire houses. **Operations Research.** n. 22, p. 275-277, 1974.