

**GRASP DUPLAMENTE REATIVA COM *PATH RELINKING* PARA O PROBLEMA  
GENERALIZADO DE  $p$ -MEDIANAS APLICADO À FORMAÇÃO DE *CLUSTERS*  
EM LOGÍSTICA DE LOCALIZAÇÃO**

**CAROLINE NASCIMENTO PARAJARA**

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE – UENF  
CAMPOS DOS GOYTACAZES – RJ  
JULHO – 2013

**GRASP DUPLAMENTE REATIVA COM *PATH RELINKING* PARA O PROBLEMA  
GENERALIZADO DE  $p$ -MEDIANAS APLICADO À FORMAÇÃO DE *CLUSTERS*  
EM LOGÍSTICA DE LOCALIZAÇÃO**

**CAROLINE NASCIMENTO PARAJARA**

“Dissertação apresentada ao Centro de  
Ciência e Tecnologia, da Universidade  
Estadual do Norte Fluminense, como  
parte das exigências para a obtenção  
do título de mestre em Engenharia de  
Produção.”

ORIENTADOR: Prof. Geraldo Galdino de Paula Junior, D. Sc.

CAMPOS DOS GOYTACAZES – RJ

JULHO – 2013

## FICHA CATALOGRÁFICA

Preparada pela Biblioteca do CCT / UENF

59/2013

Parajara, Caroline Nascimento

GRASP duplamente reativa com path relinking para o problema generalizado de p-medianas aplicado à formação de clusters em logística de localização / Caroline Nascimento Parajara. – Campos dos Goytacazes, 2013.

xiv, 131 f. : il.

Dissertação (Mestrado em Engenharia de Produção) -- Universidade Estadual do Norte Fluminense Darcy Ribeiro. Centro de Ciência e Tecnologia. Laboratório de Engenharia de Produção. Campos dos Goytacazes, 2013.

Orientador: Geraldo Galdino de Paula Junior.

Área de concentração: Pesquisa operacional.

Bibliografia: f. 125-131.

1. GRASP 2. p-MEDIANAS 3. PROBLEMA DE LOCALIZAÇÃO 4. METAHEURÍSTICA I. Universidade Estadual do Norte Fluminense Darcy Ribeiro. Centro de Ciência e Tecnologia. Laboratório de Engenharia de Produção II. Título

CDD 658.4034

**GRASP DUPLAMENTE REATIVA COM *PATH RELINKING* PARA O PROBLEMA  
GENERALIZADO DE  $p$ -MEDIANAS APLICADO À FORMAÇÃO DE *CLUSTERS*  
EM LOGÍSTICA DE LOCALIZAÇÃO**

**CAROLINE NASCIMENTO PARAJARA**

“Dissertação apresentada ao Centro de  
Ciência e Tecnologia, da Universidade  
Estadual do Norte Fluminense, como  
parte das exigências para a obtenção  
do título de mestre em Engenharia de  
Produção.”

Aprovada em 11 de JULHO de 2013.

Comissão Examinadora:

---

Eng. Edson Kenji Iamashita, D. Sc. – PETROBRAS

---

Prof. Rodrigo Tavares Nogueira, D. Sc. – UENF

---

Prof. José Ramón Arica Chávez, D. Sc. – UENF

---

Prof. Geraldo Galdino de Paula Junior, D. Sc. – UENF (Orientador)

*Dedico este trabalho ao meu avô Ciro.*

*“A persistência é o menor caminho para o  
êxito.”*

*Charles Chaplin*

## **Agradecimentos**

Agradeço em primeiro lugar a Deus que sempre esteve presente em minha vida, me iluminando e mostrando o caminho a ser seguido. Sem Ele nada sou!

Ao meu avô Ciro que, enquanto aqui esteve, sempre me incentivou de todas as formas para que estudasse.

Ao professor Galdino, não há palavras que demonstrem toda a minha gratidão por todo conhecimento transmitido e por toda sua paciência, dedicação, incentivo e por sua valiosa orientação.

A minha família por todo apoio, incentivo, compreensão e amor.

A todos os professores e funcionários do LEPROD.

Aos professores Rodrigo, Arica e ao engenheiro Edson Kenji por todas as sugestões e observações.

A Lucas pelo incentivo, companheirismo e compreensão.

Aos meus amigos Oscar Paz e Aline Viana pela amizade.

A UENF pela oportunidade do mestrado e a CAPES pelo apoio financeiro.

A todos que, de alguma forma, contribuíram para a realização deste trabalho.

## RESUMO

Este trabalho relata a experiência de formular, construir e testar um novo procedimento reativo para a GRASP, destinado à solução do problema generalizado de localização não-capacitado de  $p$ -medianas. São usados dois parâmetros de reação para controlar a busca de soluções na fase de construção da GRASP. Portanto, trata-se de uma metaheurística GRASP duplamente reativa, cujas características são aqui relatadas. O uso simultâneo dos dois parâmetros reativos permitiu criar uma disciplina para alocação dos clientes aos agrupamentos das medianas: clientes mais próximos, são alocados primeiro. O procedimento foi criado com o objetivo de aprimorar um método que já desfruta de boa reputação no ambiente das metaheurísticas em geral e, além disso, aplicá-lo a ensaios locais associados à logística de localização de instalações.

Junto dessa iniciativa, foi gerado um novo procedimento dedicado de *path relinking* para a GRASP aplicada ao problema tratado nesta dissertação. Conforme demonstram os ensaios realizados, trata-se de aprimoramento relevante que permite ao método exibir uma aproximação substancial com resultados de algoritmos exatos.

Foram realizados dois tipos de experimentos computacionais com o algoritmo desenvolvido. O primeiro experimento foi feito usando dados da biblioteca OR-Beasley, sem e com o procedimento de *path relinking*. O segundo experimento foi conduzido através de um ensaio local com a criação de *clusters*, usando 112 cidades brasileiras médias e grandes. Os resultados alcançados são interessantes e permitiram definir um índice logístico, aqui chamado de índice de vocação mediana, representado por índice  $vm$ , pelo qual pode-se representar o grau de proximidade de qualquer cidade brasileira com o aglomerado das demais cidades brasileiras médias e grandes.

O esforço de pesquisa permitiu consolidar o prestígio desta metaheurística e possibilitou ampliar o espaço por ela ocupado na solução de problemas combinatórios importantes de médio e grande portes que, com frequência, aparecem na economia.

Palavras-chave: GRASP,  $p$ -medianas, problema de localização, metaheurística.



## ABSTRACT

This work reports the experience in designing, building and testing a new procedure for the reactive GRASP, aimed at solving the general, uncapacitated  $p$ -medians location problem. Two parameters are used to control the reaction in the search for solutions at the construction phase of the GRASP. Therefore, it is a doubly reactive GRASP meta-heuristic, whose characteristics are reported here. The simultaneous use of two reactive parameters allowed to create a discipline for allocating customers to clusters associated to the medians: customers closer, are allocated first. The procedure was created with the goal of improving a method that already enjoys a good reputation in the environment of meta-heuristics in general and also apply it to locational trials associated to logistics of facility location.

Along this initiative, a new dedicated procedure was generated to accomplish the path relinking for the GRASP applied to the problem addressed in this dissertation. As shown by the tests, it is an important enhancement that allows the method to display a major approximation with results of exact algorithms.

There were two types of computational experiments with the developed algorithm. The first experiment was done using data from the OR-Beasley Library, with and without the path relinking procedure. The second experiment was conducted bringing into play a test with locational clustering, using 112 medium and large Brazilian cities. The results obtained are interesting and allowed to define a logistical index, here called median vocation index, represented by  $\text{index}_{vm}$ , by which one can represent the degree of proximity of any Brazilian city with the cluster of all others Brazilian medium and large cities.

The research effort allowed to consolidate the prestige of this meta-heuristic and also to expand the space it occupies in solving large and medium sized instances of important combinatorial problems that often appear in the economy.

Keywords: GRASP,  $p$ -medians, location problem, meta-heuristic.

# SUMÁRIO

<b>LISTA DE TABELAS</b>	<b>xii</b>
<b>LISTA DE FIGURAS</b>	<b>xiii</b>
<b>LISTA DE ALGORITMOS</b>	<b>xiv</b>
<b>I INTRODUÇÃO</b>	<b>1</b>
1.1 Objetivos . . . . .	4
1.2 Importância do trabalho . . . . .	5
1.3 Organização da dissertação . . . . .	5
<b>II PROBLEMAS DE LOCALIZAÇÃO</b>	<b>7</b>
2.1 Base histórica dos problemas de localização . . . . .	11
2.2 Modelos de localização . . . . .	12
2.2.1 Problema de fonte única e de múltiplas instalações . . . . .	12
2.2.2 $p$ -medianas e $p$ -centros . . . . .	13
2.2.3 O problema dinâmico de localização . . . . .	17
2.2.4 Problemas com múltiplos critérios . . . . .	18
2.2.5 Os problemas capacitado e não-capacitado de localização . . . . .	20
2.3 Abordagem genérica da complexidade dos problemas locais . . . . .	23
2.4 Complexidade do problema não-capacitado de localização . . . . .	23
2.4.1 O problema combinatório de recobrimento . . . . .	23
2.4.2 A equivalência entre o problema de recobrimento e o PLNC . . . . .	26
2.5 Métodos de solução . . . . .	27
2.5.1 Métodos exatos . . . . .	28
2.5.2 Introdução às metaheurísticas . . . . .	30
<b>III GRASP</b>	<b>36</b>
3.1 Introdução . . . . .	36
3.2 Cálculo da função gulosa da GRASP para $p$ -medianas . . . . .	42
3.3 GRASP reativa . . . . .	45

3.4	Novo procedimento reativo para a GRASP — a metáfora da peneira . . .	46
3.4.1	Variando os orifícios da tela da peneira . . . . .	46
3.4.2	Variando o tamanho da peneira . . . . .	47
3.5	Fase de busca local da GRASP reativa . . . . .	47
3.6	GRASP modificada: inclusão dos parâmetros de reação . . . . .	48
3.7	Complexidade assintótica da GRASP . . . . .	49
<b>IV</b>	<b>EXPERIMENTO COMPUTACIONAL – GRASP DUPLAMENTE REATIVA</b>	<b>53</b>
<b>V</b>	<b><i>PATH RELINKING</i></b>	<b>59</b>
5.1	Introdução . . . . .	59
5.2	Integração do <i>path relinking</i> com a GRASP — o que registra a literatura	61
5.3	O enfoque de <i>path relinking</i> sugerido e adotado neste trabalho . . . . .	63
5.4	Experimento comparativo com instâncias de OR Beasley usando <i>path relinking</i> . . . . .	71
<b>VI</b>	<b>O CÓDIGO COMPUTACIONAL</b>	<b>74</b>
6.1	Introdução . . . . .	74
6.2	A implementação da GRASP com <i>path relinking</i> . . . . .	76
<b>VII</b>	<b>CONCLUSÃO E PESQUISAS FUTURAS</b>	<b>77</b>
7.1	Conclusões . . . . .	77
7.2	Sugestões para trabalhos futuros . . . . .	80
<b>A</b>	<b>EXPERIMENTO COMPUTACIONAL COM O <i>PATH RELINKING</i></b>	<b>84</b>
A.1	Resultado de experimento locacional com as 112 cidades brasileiras médias e grandes para teste do <i>path relinking</i> . . . . .	84
<b>B</b>	<b>METAHEURÍSTICAS</b>	<b>99</b>
B.1	Introdução . . . . .	99
B.2	Espaço de busca e busca local como alicerces das metaheurísticas . . .	100
B.3	Algoritmos genéticos . . . . .	102
B.4	Busca tabu . . . . .	104
B.5	<i>Simulated annealing</i> . . . . .	107
B.6	Busca em vizinhança variável (BVV) . . . . .	111
B.7	Colônia de formigas . . . . .	115
B.8	Busca dispersa . . . . .	120
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>125</b>

## LISTA DE TABELAS

3.1	Matriz de distâncias para ilustrar o cálculo da função gulosa. . . . .	43
4.1	Avaliação da GRASP. Comparando com instâncias de OR Beasley. . . .	58
5.1	Resumo dos resultados com 112 cidades, 5 medianas, 2000 iterações.	66
5.2	Resumo dos resultados com 112 cidades, 5 medianas, 20000 iterações.	67
5.3	Quando piora o índice de vocação mediana, diminui o tamanho do <i>cluster</i> .	67
5.4	Cidades brasileiras, em ordem alfabética, na matriz de distâncias do DNIT.	68
5.5	Índice de vocação mediana das 112 cidades brasileiras médias e grandes.	69
5.6	Índice de vocação mediana das 112 cidades — ordenada pelo índice <sub>vm</sub> .	70
5.7	Avaliação da GRASP. Comparando com instâncias de OR Beasley usando <i>path relinking</i> . . . . .	73
6.1	Matriz de distâncias antes da aplicação do algoritmo de Floyd-Warshall, onde $\infty$ representa distâncias ainda desconhecidas. . . . .	75
6.2	Matriz de distâncias após a aplicação do algoritmo de Floyd-Warshall. .	76
A.1	Resultado (1) sem <i>path relinking</i> . . . . .	86
A.2	Resultado (2)           ”           . . . . .	87
A.3	Resultado (3)           ”           . . . . .	88
A.4	Resultado (4)           ”           . . . . .	89
A.5	Resultado (5)           ”           . . . . .	90
A.6	Resultado (6)           ”           . . . . .	91
A.7	Resultado (7)           ”           . . . . .	92
A.8	Resultado (8)           ”           . . . . .	93
A.9	Resultado (9)           ”           . . . . .	94
A.10	Resultado (10)           ”           . . . . .	95
A.11	26 cidades elite, candidatas a medianas na fase de <i>path relinking</i> . . . .	96
A.12	Resultado com <i>path relinking</i> . . . . .	97
A.13	Resultado com <i>path relinking</i> após 200000 iterações. . . . .	98

## LISTA DE FIGURAS

2.1	Exemplo — ilustração — de Localização de $p$ -medianas, ( $p = 9$ ). . . . .	14
2.2	Ilustração numérica da solução de um problema de $p$ -medianas. . . . .	15
2.3	Relacionando o problema do recobrimento ao PLNC . . . . .	26
3.1	Esquema geral de solução por via da GRASP, Günther et al. (2010). . . . .	37
3.2	Esquema construtivo da GRASP, Günther et al. (2010). . . . .	37
3.3	Elementos de construção da Lista Restrita de Candidatos. . . . .	46
3.4	Crescimento assintótico de funções de complexidade de tempo. . . . .	51
4.1	Problema de Localização de 60 pontos com 3 medianas. . . . .	53
4.2	Resultado do problema de 60 pontos com 3 medianas. . . . .	54
4.3	Localização das 3 medianas — 1, 19 e 37 — entre 60 pontos. . . . .	54
4.4	Experimento: comparação com Beasley – uma iteração GRASP. . . . .	56
4.5	Experimento: comparação com Beasley – 10000 iterações GRASP. . . . .	57
5.1	Dois caminhos: o original e o <i>path relinking</i> – em linhas sólidas. . . . .	60
5.2	Um esquema de <i>path relinking</i> sugerido em Talbi (2009). . . . .	62
5.3	Experimento: comparação com Beasley, usando <i>path relinking</i> . . . . .	72
B.1	Crossover de 1 ponto. Fonte: Yu & Gen (2012) . . . . .	103
B.2	Crossover de 2 pontos. Fonte: Yu & Gen (2012) . . . . .	103
B.3	Mutação: um único gene é trocado no cromossomo. . . . .	103
B.4	Esquema da BVV. . . . .	113
B.5	Comportamento da BVV com duas estruturas de vizinhança. . . . .	114
B.6	Esquema geral da busca dispersa. Fonte: Yu & Gen (2012) . . . . .	121
B.7	Esquema da busca dispersa. Fonte: Adaptado de Günther et al. (2010) . . . . .	123

## LISTA DE ALGORITMOS

1	Procedimento GRASP. . . . .	40
2	Fase de CONSTRUÇÃO. . . . .	41
3	Fase de MELHORIA. . . . .	41
4	Procedimento GRASP. . . . .	48
5	Fase de CONSTRUÇÃO. . . . .	48
6	Procedimento GRASP COM <i>PATH RELINKING</i> . . . . .	64
7	Algoritmo de Floyd-Warshall. . . . .	75
8	Busca Local. . . . .	101
9	Algoritmo genético. . . . .	104
10	Algoritmo de busca tabu. . . . .	107
11	Algoritmo de <i>simulated annealing</i> . . . . .	111
12	Busca em vizinhança variável. . . . .	114
13	Colônia de formigas. . . . .	120
14	Algoritmo de busca dispersa. . . . .	124

# CAPÍTULO I

## INTRODUÇÃO

Problemas de localização associam-se a um conjunto de instalações ou facilidades que minimizam o custo de satisfazer a demandas e obedecem a um elenco de restrições. Os quatro componentes principais que ajudam a descrever com simplicidade um problema locacional são:

- Instalações a serem localizadas
- Clientes, a respeito dos quais assume-se que já estão localizados ou instalados em pontos ou rotas
- Uma métrica a ser usada no tratamento das distâncias ou custos de transporte entre clientes e instalações
- Um espaço geográfico em que estão situados os clientes e os pontos candidatos para receber as instalações.

Modelos e decisões locacionais são de natureza econômica e visam a organizar o atendimento eficiente a algum tipo de demanda. Modelos locacionais são usados em aplicações que têm como meta e ênfase otimizar o custo ou tempo de atendimento, entre outros itens de interesse.

As instalações podem ser de natureza industrial, como fábricas e armazéns, *hubs* ou centros logísticos de recebimento, despacho e distribuição de mercadorias. Podem ser destinadas ao atendimento da população, como hospitais, escolas, bibliotecas, torres de transmissão de sinal de tv e de telefonia móvel, unidades de segurança pública e de combate a incêndios. Outros exemplos menos óbvios incluem instalações indesejadas que movimentam ou guardam produtos nocivos à natureza e à saúde humana, maximizando a distância entre essas instalações e o público em geral, como locais para depósito e tratamento de rejeitos e instalações nucleares. Além dessas, devem ser lembradas as que restringem o convívio social, como instituições corretivas e unidades prisionais. Nesse sentido, a análise locacional é um tema de interesse de

pesquisadores das áreas de engenharia, geografia, economia, pesquisa operacional, marketing, administração, matemática, políticas públicas, ciências sociais e ciência regional, entre outras.

A variedade de problemas que decorrem desse conceito compartilha características que se traduzem na escolha de facilidades individualmente posicionadas e na meta de atender a demandas de clientes em geral.

A nomenclatura dos problemas locacionais inclui a expressão “problema não-capacitado de localização” para designar o caso mais conhecido e pesquisado no contexto dos problemas de localização. Ela traz outras expressões que exibem a diversidade de problemas locacionais, como o “problema capacitado de localização”, a “localização de  $p$ -medianas” e de “ $p$ -centros” e a “localização em rede”.

Na formalidade o problema é definido como: um conjunto  $C$  de clientes e um conjunto  $I$  de instalações em potencial. Um custo fixo  $f_i \in \mathbb{R}_+$ , para cada instalação  $i \in I$ , e um custo  $c_{ij} \in \mathbb{R}_+$  de transporte, para cada  $i \in I$  e cada  $j \in C$ .

A partir destas informações, deve ser encontrado um subconjunto  $S \subset I$  de instalações (ditas instalações abertas) e uma aplicação  $\gamma : C \rightarrow S$ , que atribui clientes a instalações abertas, de modo a minimizar a soma dos custos de transporte com os custos fixos:

$$\sum_{j \in C} c_{\gamma(j)j} + \sum_{i \in S} f_i$$

que traduz a soma: (custo de serviço) + (custo de instalação).

Em muitas aplicações práticas, o custo dos serviços vem de uma métrica  $c$  sobre  $C \cup I$ . Isto é,  $c : C \cup I \rightarrow \mathbb{R}$ .

Especializando o tema para a indústria e a distribuição de seus produtos, o que se procura determinar nos problemas de localização é “onde” tais facilidades devem ser localizadas, “quais” fornecedores devem ser utilizados e com “quantos” centros de distribuição a empresa deve operar. Estudos que antecedem e sucedem os projetos locacionais procuram determinar o tipo de cliente ou de zonas de mercado que devem ser supridos a partir de cada centro de distribuição. Avaliam também os modais de transporte que podem ser usados para suprimento ou distribuição. São questões que apresentam forte interdependência com os problemas de localização.



No que registra a literatura, observa-se que a busca por um espaço maior e competitivo no mercado, progressivamente induz as empresas a investir em formas que representem um diferencial para o sucesso. Assim, é necessário realizar uma permanente reavaliação de processos, desde o desenvolvimento dos produtos até a entrega ao cliente. Neste momento, a logística torna-se o diferencial das empresas. Com o objetivo de permanecer no mercado as empresas buscaram formas de reduzir custos e aumentar a velocidade de seus processos, (JUNIOR & CECHIN, 2006).

Porter (1999) apresenta uma das razões que reforçam a necessidade da decisão locacional ser parte da estratégia global da empresa: “A localização afeta a vantagem competitiva através da influência sobre a produtividade e, em especial, sobre o crescimento desta”. Ele observa que uma empresa de qualquer setor (agricultura, vestuário, educação, etc.) pode tornar-se mais produtiva se forem incorporados métodos sofisticados, adotadas tecnologias avançadas e oferecidos produtos e serviços únicos. Potencialmente, todos os setores dispõem de condições para utilizar alta tecnologia e podem ser intensivos em conhecimento.

Uma boa localização pode fornecer vantagens estratégicas difíceis de serem superadas pela concorrência. Uma outra questão importante é que a localização representa um investimento de longo prazo que frequentemente está associada a um custo significativo no caso de realocação, ao contrário das estratégias de marketing, por exemplo, que podem ser facilmente modificadas em resposta a uma mudança ambiental. (GHOSH & CRAIG, 1983)

O projeto de um sistema logístico (transporte e localização) é uma questão estratégica para muitas empresas. Para Cortes (2003), a importância das decisões locais é frequentemente relacionada ao investimento elevado que tais decisões acarretam sobre os custos. Dessa forma, a localização representa um elemento fundamental para a vantagem competitiva das empresas.

A atividade de distribuição física dos produtos de uma empresa compreende toda a movimentação de bens entre a fábrica e os centros de distribuição. A última etapa nessa movimentação, dos centros de distribuição em direção aos consumidores, que pode ser definida como transporte local ou entrega, representa o elo mais caro da cadeia de distribuição (CHRISTOFIDES et al., 1981). Para esta etapa ser realizada

de maneira eficiente, a empresa deve desenvolver o planejamento e a execução desta atividade de transporte de forma racional e otimizada (BODIN et al., 1983).

O problema de escolher locais onde instalar unidades para fornecimento de produtos industriais e alocar clientes a elas é a questão central do projeto de um sistema de distribuição. A formulação de modelos e a construção de algoritmos para solução do problema subjacente é uma atividade que pode variar amplamente. Para ilustrar tal variedade, o problema de localização pode, dentre outras abordagens, ser tratado como um modelo de localização de  $p$ -medianas (generalizado ou não) ou como um problema de localização capacitado de dois estágios. Em qualquer deles o que se pretende é encontrar locais ótimos com vistas a instalar depósitos (centros de distribuição ou CD's) para servir aos clientes, sendo que no problema capacitado as instalações têm capacidades limitadas e os clientes exercem uma dada demanda. No problema simples (não-capacitado) considera-se que as instalações têm capacidade ilimitada e o fluxo ótimo de produtos ocorre entre a instalação e os clientes a ela atribuídos. No problema de dois estágios, procura-se garantir a atribuição ótima de clientes a CD's e o fluxo ótimo de produtos ocorre entre a fábrica e os CD's, e entre os CD's e os clientes a eles atribuídos. A relevância de um sistema logístico otimizado decorre do ganho de eficiência e competitividade, e do valor econômico que a empresa pode auferir em seu benefício.

Problemas de localização são, em geral, computacionalmente intratáveis, ou seja, não se pode garantir que sejam resolvidos de maneira exata por algoritmos em tempo polinomial determinístico, veja seções 2.3 e 2.4. Portanto, suas soluções podem ficar severamente comprometidas quanto a velocidade de emissão dos resultados. Este fato motiva a utilização de algoritmos heurísticos para proverem soluções viáveis tão boas quanto possíveis, eventualmente ótimas, para os problemas.

## 1.1 Objetivos

Os objetivos aqui definidos decorrem do que inspira o último parágrafo da introdução deste trabalho. Lá é criado um espaço para motivar aprimoramentos de métodos aproximados, ou não exatos, de solução dos problemas locacionais, em particular, dos modelos de localização não-capacitados de  $p$ -medianas. Os objetivos são definidos

como:

- Explorar variações de processos reativos, ainda inexistentes, da metaheurística GRASP. Em particular criar e experimentar um método duplamente reativo para a metaheurística em foco, explorando a forma cooperativa com que os parâmetros de reação podem agir no sentido de produzir um método de solução aprimorado
- Criar e experimentar uma nova forma de *path relinking* para a GRASP, especializada na solução de problemas de localização não—capacitados de  $p$ —medianas
- Ajustar o funcionamento da GRASP duplamente reativa com a nova modalidade de *path relinking* para o problema aqui tratado
- Demonstrar ou ilustrar a aplicação do método criado para simular a formação de *clusters* de vértices—ou cidades—atendidos por medianas, vértices ou cidades que centralizam o *hub* de atendimento aos clientes do *cluster*. Nesta demonstração o ensaio utilizará as cidades brasileiras médias e grandes, bem como as distância reais entre elas.

## 1.2 Importância do trabalho

Com o aumento de escala dos problemas de localização, as metaheurísticas ganharam um espaço generoso junto aos métodos de solução de problemas combinatórios. O crescimento de porte dos casos práticos que surgem na economia, continua a exigir esforço de pesquisa para aumentar a capacidade e eficiência dos métodos de solução. Além disso, problemas de localização possuem uma complexidade elevada. Portanto, processos mais eficientes envolvendo modelos e métodos precisam ser desenvolvidos com o intuito de melhorar a abordagem da complexidade inerente às decisões locais.

## 1.3 Organização da dissertação

O presente trabalho está organizado em 7 capítulos e 2 apêndices. Após este Capítulo introdutório, segue o Capítulo II onde é feito um estudo sobre a base teórica dos

problemas de localização. São apresentados também os principais tipos de problemas locais, sua complexidade, seus métodos de solução e uma revisão bibliográfica.

O Capítulo III descreve a metaheurística GRASP (“Greedy Randomized Adaptive Search Procedure”), onde é feito um detalhamento de sua complexidade assintótica, de sua fase de construção e de busca local, além da GRASP Reativa aqui desenvolvida.

O Capítulo IV faz um detalhamento do algoritmo — GRASP duplamente reativa — construído nesta dissertação, assim como realiza um experimento computacional em que se compara os resultados obtidos com as instâncias da biblioteca OR Beasley. O resultado da comparação com as instâncias de OR Beasley é apresentado através de gráficos e na forma de tabela.

O Capítulo V desenvolve e testa uma forma de *path relinking* dedicada à GRASP especializada em problemas de localização não-capacitados de  $p$ -medianas. Um índice logístico, aqui chamado de “índice de vocação mediana — índice<sub>vm</sub>” é definido neste capítulo. O ensaio computacional realizado com essa forma de *path relinking* usa, além das instâncias de OR Beasley, as cidades brasileiras médias e grandes, com as distâncias reais entre elas.

O Capítulo VI descreve o ambiente computacional e o código desenvolvido para os algoritmos produzidos e testados neste trabalho.

Finalmente, o Capítulo VII apresenta a conclusão desta pesquisa e faz sugestões para trabalhos futuros.

O Apêndice A resume os resultados do ensaio computacional com o *path relinking* usando as cidades brasileiras médias e grandes, e o Apêndice B faz uma apresentação de outras metaheurísticas que concorreram com a GRASP para serem adotadas neste trabalho.

## CAPÍTULO II

### PROBLEMAS DE LOCALIZAÇÃO

Existem várias formas de descrever, utilizar e definir problemas de localização, cada uma delas dando destaque ao aspecto que se deseja abordar. A descrição simplificada que aqui se faz, enquadra-se nesse elenco de possibilidades. Existem fatos comuns aos vários enfoques que podem ser adotados. O problema pode ser abordado de forma qualitativa ou quantitativa, ou ainda adotando uma abordagem híbrida dos dois enfoques. Localização de instalações tem sido assunto de interesse da engenharia de produção, engenharia de transporte, de administradores, membros da comunidade de pesquisa operacional e de executivos ligados à logística. Será feita uma pequena introdução ao tema e, posteriormente, será dado destaque a alguns aspectos do problema, em especial, ao enfoque quantitativo que é adotado neste trabalho.

A necessidade de fornecer com eficiência produtos provenientes de um conjunto de centros de distribuição supridos por fábricas, dá origem a um projeto que tem por base o problema de localização de instalações. Esta forma de introduzir o problema não deve induzir nenhuma exclusividade associada a produtos e fábricas, já que o problema de localização pode ser estudado em função de hospitais, escolas e outras unidades de atendimento à população em geral, como serviços baseados em localização que têm na telefonia móvel um bom exemplo. Entretanto, as abordagens práticas mais comuns consideram situações em que uma mercadoria — ou um misto delas — deve ser fornecida a partir de instalações selecionadas de um conjunto de locais elegíveis, para satisfazer a demanda de clientes espacialmente distribuídos. O problema envolve custos variáveis de transporte entre os centros fornecedores e os consumidores. São considerados ainda os custos fixos para manter as instalações em operação. Compete ao decisor escolher uma combinação de custo mínimo em termos de instalações a serem abertas e a determinação de quais clientes devem ser supridos pelo conjunto de instalações em operação. O termo facilidade é também usado neste contexto para designar, genericamente, fábricas, depósitos, centros de distribuição, torres de transmissão de sinal de TV, *hubs* associados ao transporte aéreo, torres

de recepção e transmissão de sinais associados à telefonia móvel, shopping centers, escolas, hospitais, entre outros. O termo “alocar” aparece com frequência neste contexto e serve para designar a atribuição ou associação de clientes às facilidades ou instalações que lhes atendem.

A análise locacional é vista também como um tópico importante no gerenciamento logístico. Neste ponto é necessário esclarecer o entendimento associado a este termo e afirmar, de acordo com Sule (2001), que a logística pode ser definida como a parte do processo de cadeia de suprimento que planeja, implementa e controla o fluxo e armazenamento eficiente e efetivo de bens, serviços e a informação que acompanha o processo, do ponto de origem ao ponto de consumo, visando atender a demanda do consumidor final. A localização de instalações surge como elo destacado nessa cadeia.

Em uma ligeira variação do tema, Lacerda (2000) menciona que, em geral, os estudos de localização tratam do problema de minimizar os custos de uma rede logística, estando esta sujeita a restrições de capacidade das instalações, tendo que atender a uma determinada demanda e devendo satisfazer certos limites de nível de serviço. Os dados de entrada para análise são as previsões de demanda para cada produto, as limitações de capacidade e as taxas de produção, as prováveis localizações das instalações, as possíveis ligações entre elas e os respectivos custos de transporte de cada modal.

Segundo Cortes & Paula Junior (2005), determinar a melhor localização para a instalação de filiais em um horizonte de planejamento é um desafio relevante para manter a competitividade de uma empresa, pois torna a escolha de onde localizar suas atividades econômicas uma das decisões estratégicas mais importantes no universo corporativo.

No início deste capítulo foi mencionada a existência dos enfoques qualitativo e quantitativo de abordagem do problema de localização. Para ilustrar o aspecto qualitativo começa-se lembrando que um dos primeiros atos que antecedem a análise locacional corresponde a avaliar a influência de alguns fatores inerentes a este tipo de problema. Em geral, a escolha de locais para instalações é uma decisão coletiva que envolve a participação de um grupo de executivos da empresa. É esse grupo que analisa

a importância dos fatores capazes de influenciar as decisões locacionais. Em geral, os itens auto-explicáveis, mais frequentes em análises dessa natureza, alguns inter-relacionados, incluem:

- Disponibilidade e preço de área para construção
- Infraestrutura de transporte, incluindo a variação de modais — rodoviário, aéreo, ferroviário, hidroviário e marítimo
- Oferta de mão de obra
- Proximidade de mercados
- Proximidade de fontes com oferta de matéria prima
- Proximidade de outras instalações de mesma natureza
- Atitudes favoráveis ou não da comunidade circunvizinha
- Disponibilidade de outros recursos como oferta de água e energia elétrica
- Disponibilidade e proximidades de parceiros terceirizáveis, reais ou eventuais
- Impostos e legislação.

O que o grupo de executivos procura com esta relação de itens é uma orientação para responder as perguntas a seguir, que aparecem no momento de implementar um projeto locacional.

- Onde devem ser localizadas as instalações?
- Como devem ser atribuídos ou alocados os clientes a estas facilidades, ou quais facilidades devem servir a quais clientes?
- Qual deveria ser a capacidade individual ou coletiva das instalações?
- Como deve ser desenvolvido o planejamento das facilidades de forma a permitir mudanças na produção, transporte e de outros itens como custos e tempos?
- Como devem ser planejadas as instalações dando-lhes condições de responder com rapidez as demandas de clientes ou usuários?

Além desses itens, outros pontos ligados ao aspecto qualitativo do problema devem ser lembrados. Em particular, deve ser mencionado que uma classificação importante para os problemas de localização relaciona-se ao horizonte de planejamento. Crainic (2002) cita os níveis de planejamento em que as diferentes decisões logísticas de localização se enquadram:

- **Nível estratégico:** são decisões que envolvem alta gerência e requerem grande investimento de capital, dentro de um horizonte de planejamento de longo prazo. Os dados são geralmente agregados e há um nível elevado de flexibilidade. Decisões estratégicas determinam as políticas gerais da empresa. Exemplos de decisões neste nível de planejamento envolvem a determinação do número de instalações em operação, suas respectivas localizações e capacidades.
- **Nível tático ou gerencial:** é o planejamento da rede logística no horizonte de médio prazo. Por exemplo, quando se define a alocação de clientes aos centros de distribuição, e destes às suas respectivas fábricas, a fim de aumentar o desempenho do sistema logístico. Neste nível, há uma mistura de dados agregados e detalhados, o nível de flexibilidade é intermediário, as políticas ainda são abstratas e as decisões são sensíveis a grandes variações dos dados e parâmetros do sistema, tais como variações sazonais de demanda. Outros exemplos de decisões dentro deste nível englobam a roteirização de veículos, a programação e o sequenciamento de atividades.
- **Nível operacional:** são as decisões de chão-de-fábrica, tomadas pelos gerentes locais, num horizonte de curto prazo, e em ambientes altamente dinâmicos, com baixa flexibilidade, onde o tempo de resposta deve ser mínimo. Possuir a representação detalhada das instalações, veículos e atividades é essencial. Como exemplos de decisões neste nível, pode-se citar a localização e movimentação de materiais dentro de armazéns, a programação de tripulações ou a alocação de recursos escassos, como por exemplo, a realocação de clientes em caso de parada de uma linha de produção em uma fábrica.

A avaliação qualitativa e as decisões que ela pode implicar, sobre a conveniência ou não dos vários itens locacionais são, possivelmente, de natureza fortemente subjetiva. Envolve preferências e gosto dos avaliadores, e analistas diferentes podem



atribuir pesos distintos aos vários itens envolvidos na avaliação. Surge, diante disso, a sugestão de que ela deva ser antecedida de uma análise ou avaliação quantitativa, em que se torna mínima a presença de subjetividades. Este trabalho explora o lado quantitativo da análise locacional — ainda a ser apresentado — sem desprezar o fato de que, na prática locacional, uma abordagem híbrida — compondo o qualitativo com o quantitativo — seja a mais aconselhada e adequada para projetos dessa natureza.

## 2.1 Base histórica dos problemas de localização

A análise dos problemas de localização é uma área em estudo desde o início do século passado. Historicamente, a teoria de localização foi formalmente iniciada por von Thünen, (1866) através da consideração de atividades agrícolas. Porém, deve-se a Weber (1909) a primeira tentativa de obtenção de uma teoria geral para a localização, (*apud* CORTES, 2003).

O modelo de von Thünen, (1866) sugere que a acessibilidade ao mercado urbano pode criar um sistema completo de uso da área rural. Esse modelo foi desenvolvido para a agricultura e supõe que um único produto é utilizado para abastecer um único centro urbano. O fator determinante no aluguel da terra é o custo de transporte. Quando tal custo é baixo, o aluguel da terra será alto, e vice-versa, (*apud* CORTES, 2003).

O modelo de Weber (1909) utiliza os seguintes pressupostos básicos: a distribuição não-uniforme das matérias-primas, uniformidade dos custos de transporte, homogeneidade do preço das mercadorias e mercados consumidores puntiformes. Nesse modelo, a localização de uma indústria será aquela que minimizar o custo total de transporte, tanto da matéria-prima quanto do produto final. Além do custo de transporte, os fatores locacionais de custo do trabalho e de aglomeração também, afetam a localização da atividade, (*apud* CORTES, 2003).

A publicação de Hakimi (1964) contribuiu para o crescimento da teoria de localização. Seu trabalho tinha o objetivo de determinar a localização de postos de serviços em redes e postos policiais em rodovias. Nesse trabalho, foram estabelecidos dois teoremas: o primeiro, determinando que existe um ponto em uma rede que minimiza a soma ponderada das distâncias mais curtas de todos os vértices a este ponto, o qual

vem a ser um dos vértices da rede. O segundo teorema determina que, para o caso de se escolher  $p$  pontos centrais (problema conhecido como  $p$ -mediana), existirá um conjunto de  $p$  pontos (vértices da rede) que minimizará a soma das distâncias ponderadas de todos os vértices aos mais próximos de  $p$  pontos da rede, (*apud* CORTES, 2003).

Desde então, a teoria da localização vem sendo desenvolvida ao longo dos anos e inspirando muitos pesquisadores em várias áreas de aplicação, como Engenharia, Economia, Pesquisa Operacional, Logística, Telefonia móvel, Marketing, Matemática, Ciência Regional, entre outros.

## 2.2 Modelos de localização

Conforme já mencionado, a formulação de problemas locacionais é uma atividade que pode variar amplamente. Já foram apresentados os componentes de uma abordagem qualitativa dos projetos de localização. Inicia-se aqui a apresentação dos aspectos quantitativos, usando conceitos, definições, modelos e algoritmos dedicados à solução do problema em foco.

### 2.2.1 Problema de fonte única e de múltiplas instalações

O problema de localização de uma única instalação, também conhecido como de fonte única, ocorre quando se pretende que haja apenas uma instalação, ou quando uma possível instalação estará tão isolada das demais que a demanda a ser atendida por ela pode ser considerada independente da demanda a ser atendida pelas demais instalações, permitindo a decomposição do problema de localização de  $n$  instalações em  $n$  problemas independentes de localização de uma instalação. Desta forma, essa hipótese certamente simplifica bastante o problema a ser resolvido, que passa a se basear na busca do local que permite otimizar uma função objetivo, seja ela voltada para a maximização dos lucros da empresa, para a minimização dos custos envolvidos, ou para objetivos específicos, como a minimização das distâncias ou dos tempos de transporte associados ao atendimento das demandas consideradas.

A modelagem do problema de múltiplas instalações certamente é mais complexa,

pois se deve buscar respostas para questões relacionadas a quantas instalações implantar, onde implantá-las, a capacidade de cada uma delas, a área de influência das mesmas e os modos de transporte a serem utilizados para o suprimento das mesmas e distribuição a partir de cada uma delas. A função objetivo, nesse caso, está associada, em geral, à minimização do somatório dos custos associados a cada uma das instalações, sujeita a restrições quanto ao porte mínimo ou máximo de cada instalação, distância entre elas, distâncias máximas de cada instalação aos pontos de demanda, além de limitações de capacidade de vias e dos veículos de transporte.

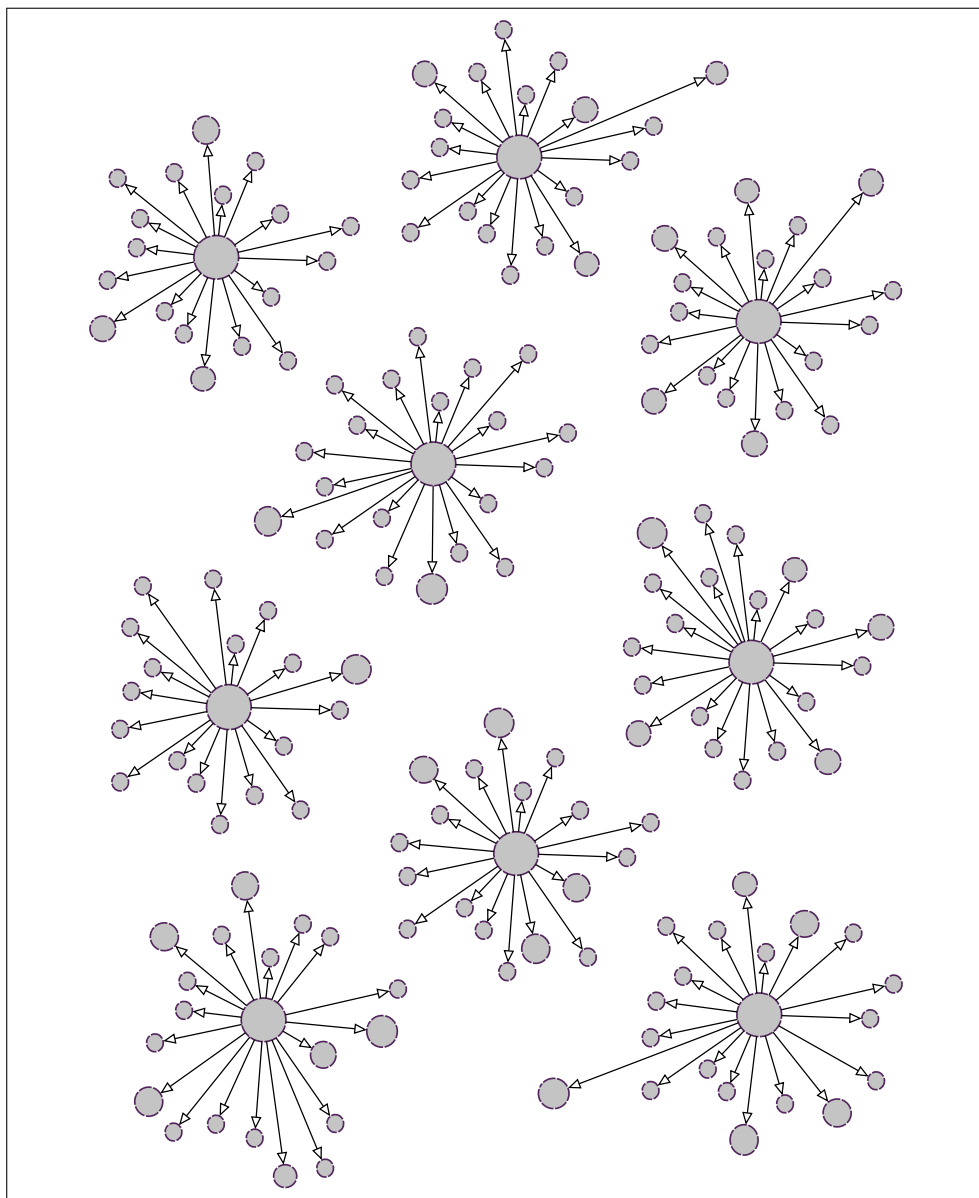
### 2.2.2 $p$ -medianas e $p$ -centros

Localizar  $p$ -medianas onde construir instalações constitui o problema de identificar  $p$  pontos numa rede que permitem minimizar a soma de todas as distâncias de cada ponto de demanda a sua mediana. O objetivo é minimizar a distância média para todos os pontos de demanda. É o maior grupo de problemas de localização de atividades e é conhecido como problema “minisoma”, mais apropriado para localizar instalações de interesse das empresas do setor privado — facilidades de fabricação e montagem, armazéns, etc.

A Figura 2.1 mostra vários pontos distribuídos em um plano. Nove desses — de onde partem as setas — foram escolhidos como medianas, formando assim, nove agrupamentos. Esse é um exemplo de problema de localização de  $p$ -medianas, cuja solução é objeto de pesquisa deste trabalho.

Segundo Pereira (2005), os dados relevantes para um problema de  $p$ -medianas são:

- Um número finito de pontos, com valores conhecidos de demanda, denominados pontos de demanda
- Um número finito de locais candidatos para a instalação de facilidades
- A distância entre cada ponto de demanda e os locais candidatos. Tais distâncias podem ser calculadas sobre a rede de caminhos que conectam os pontos, ou como distâncias euclidianas
- O número  $p$  de facilidades a serem instaladas.



**Figura 2.1:** Exemplo — ilustração — de Localização de  $p$ -medianas, ( $p = 9$ ).

A Figura 2.2 ilustra a solução numérica de um problema de localização não-capacitado de  $p$ -medianas, em que  $p = 10$ , e o número de vértices, pontos ou cidades envolvidos nesta instância, é igual a 900. Uma variação do problema de  $p$ -medianas, cuja solução foi objeto de pesquisa para este trabalho, é o problema generalizado de  $p$ -medianas, em que  $p$  é o número máximo de pontos em que se pretende posicionar as instalações e, em torno deles, organizar os *clusters* ou agrupamentos de clientes.

Voltando ao problema simples de  $p$ -medianas, tem-se que uma formulação matemática para ele pode ser a seguinte: Dado um grafo  $G = (N, A)$ , com  $|N| = n$ . O problema de  $p$ -medianas consiste em determinar  $p < n$  vértices (medianas) de modo a mini-

Solução de instância do problema não-capacitado de  $p$ -medianas: 900 cidades 10 medianas.

Dia: 07/07/2012, Hora: 21:05:14

Instância: 900 Cidades, 16200 Ligações entre cidades, 10 Medianas

A MELHOR SOLUÇÃO ENCONTRADA EM 10000 ITERAÇÕES, É:

Cluster da mediana 728:	23	59	74	117	155	176	224	260	261	271	277	282	292	306
	307	309	324	327	345	371	376	413	414	433	458	463	478	483
	511	532	542	572	581	618	646	648	650	665	695	727	758	765
	772	801	818	827	849									
Total de cidades neste cluster:	47, distância (custo) total neste cluster: 2931													
Cluster da mediana 762:	29	30	36	37	38	71	84	95	114	119	148	172	198	199
	203	205	233	236	242	253	262	266	267	273	276	288	316	317
	320	326	333	335	336	340	346	354	394	408	434	440	447	449
	498	504	505	506	515	541	547	560	576	580	587	621	627	634
	653	681	685	692	705	709	712	720	722	738	747	750	751	761
	763	764	771	787	794	812	829	848	850	851	876	881	891	
Total de cidades neste cluster:	83, distância (custo) total neste cluster: 6226													
Cluster da mediana 318:	50	67	104	125	166	187	231	234	256	319	331	343	344	348
	377	379	435	454	467	480	500	514	569	579	609	623	656	660
	668	680	684	726	739	741	744	760	777	843	861	899		
Total de cidades neste cluster:	40, distância (custo) total neste cluster: 2734													
Cluster da mediana 411:	28	33	34	43	44	45	80	83	99	109	113	120	122	123
	164	165	177	186	190	197	219	222	225	232	240	250	259	268
	286	298	302	313	325	347	351	362	369	406	407	410	412	418
	427	430	455	461	466	471	503	517	522	538	565	599	610	611
	617	628	636	637	672	691	696	703	706	719	730	734	736	740
	742	757	759	779	804	821	833	839	872	885	888			
Total de cidades neste cluster:	81, distância (custo) total neste cluster: 6304													
Cluster da mediana 392:	25	32	51	52	56	63	66	79	98	112	143	144	159	161
	163	209	220	243	312	322	328	357	382	391	393	398	402	415
	436	439	445	474	481	482	507	516	531	539	564	566	568	578
	585	608	629	644	647	679	688	711	716	743	745	767	775	782
	784	786	788	790	820	823	837	840	847	858	864	873	879	893
Total de cidades neste cluster:	70, distância (custo) total neste cluster: 8794													
Cluster da mediana 206:	15	16	17	18	26	27	35	39	40	58	73	77	81	89
	103	107	111	115	124	135	136	139	149	150	156	158	169	171
	178	181	193	195	204	207	210	216	245	251	252	263	264	270
	284	289	290	296	360	363	366	367	372	375	380	388	404	405
	421	424	437	457	462	470	496	502	520	529	530	537	543	551
	558	570	577	582	583	588	591	594	597	604	607	614	619	620
	624	642	652	654	657	667	674	676	677	683	693	718	724	735
	752	755	766	768	781	791	803	806	809	814	824	844	853	854
	868	870	875	878	884	886	894	896						
Total de cidades neste cluster:	120, distância (custo) total neste cluster: 8100													
Cluster da mediana 535:	1	2	3	10	11	12	13	24	41	42	53	57	61	69
	70	72	82	87	90	92	106	108	110	127	130	131	134	141
	142	152	153	160	162	174	179	183	184	191	192	194	196	201
	202	214	215	218	226	229	235	237	244	246	247	272	274	279
	281	300	301	304	311	330	337	338	342	353	355	358	364	373
	378	381	390	396	397	416	417	420	422	428	431	432	438	442
	443	444	448	452	459	460	469	479	484	487	499	501	508	509
	518	523	533	534	536	540	544	548	555	556	557	561	562	573
	589	592	605	622	625	626	630	631	633	638	655	664	666	669
	670	671	675	678	686	687	702	707	713	714	715	721	725	729
	733	737	749	753	754	769	773	783	785	789	792	793	798	800
	805	813	815	828	830	832	855	859	860	867	869	871	880	882
	883	887	890	892	898	900								
Total de cidades neste cluster:	174, distância (custo) total neste cluster: 13637													
Cluster da mediana 602:	21	22	46	47	48	60	76	85	86	102	121	126	137	138
	213	230	249	254	278	294	295	308	314	315	329	339	361	384
	387	389	401	425	429	464	468	473	477	488	491	492	495	513
	519	526	527	528	549	553	554	559	574	590	601	606	632	635
	641	645	651	708	717	774	795	807	816	817	825	846	862	863
Total de cidades neste cluster:	70, distância (custo) total neste cluster: 4540													
Cluster da mediana 132:	4	5	6	7	14	54	62	64	65	68	75	88	91	93
	96	116	128	129	133	145	151	154	157	168	170	173	175	185
	188	200	211	212	217	227	238	239	258	269	275	283	285	291
	293	299	303	310	334	341	352	359	368	370	386	395	400	409
	419	423	426	446	451	453	465	472	475	476	485	486	490	494
	510	546	552	567	571	575	584	593	595	596	598	600	603	613
	615	616	658	662	682	689	697	700	704	710	723	731	776	778
	780	796	802	810	822	826	831	834	835	836	838	841	845	865
	866	877	889	895										
Total de cidades neste cluster:	116, distância (custo) total neste cluster: 9043													
Cluster da mediana 101:	8	9	19	20	31	49	55	78	94	97	100	105	118	140
	146	147	167	180	182	189	208	221	223	228	241	248	255	257
	265	280	287	297	305	321	323	332	349	350	356	365	374	383
	385	399	403	441	450	456	489	493	497	512	521	524	525	545
	550	563	586	612	639	640	643	649	659	661	663	673	690	694
	698	699	701	732	746	748	756	770	797	799	808	811	819	842
	852	856	857	874	897									
Total de cidades neste cluster:	89, distância (custo) total neste cluster: 6501													

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 68810

Dia: 07/07/2012, Hora: 22:56:57

Tempo total gasto: 6703 segundos (111.717 minutos ou 1.86194 hora)

**Figura 2.2:** Ilustração da solução numérica de uma instância do problema de  $p$ -medianas ( $p = 10$ ), com 900 vértices — ou cidades.

mizar a soma das distâncias dos outros vértices do grafo à mediana mais próxima. A matriz de distâncias,  $D = (d_{ij})_{n \times n}$ , entre cada par de vértices deve ser conhecida *a priori*. Uma outra forma de apresentação do problema considera o particionamento de um conjunto  $N$  em  $p$  *clusters* disjuntos, minimizando — em conjunto — a distância entre cada elemento de cada *cluster* e a mediana que lhe corresponde.

Devido ao fato do problema de  $p$ -medianas permitir minimizar a soma de todas as distâncias de cada ponto de demanda a sua mediana, atualmente um grande número de aplicações são encontradas na literatura. Hakimi (1964) demonstrou as primeiras aplicações do problema de  $p$ -medianas para determinação de centros de comutação em redes de comunicações.

Tibes (2005) verificou a necessidade da população quanto à implantação de farmácias, atendendo às necessidades e ao crescimento populacional do município de Pato Branco. Foi proposta uma metodologia para distribuição espacial de Farmácias com o objetivo de minimizar a distância média de deslocamento dos usuários desde suas residências até a farmácia mais próxima. Como metodologia de solução adotada nesse trabalho, foram utilizados: o Algoritmo Genético e o Algoritmo de Teitz e Bart. Ao saber a quantidade de unidades a serem implantadas e realocadas, com as técnicas adotadas aplicadas ao problema das  $p$ -medianas, em pouco tempo se encontrava as melhores localizações para a implantação das Farmácias.

Em Lorena et al. (2001) é feito um relato da integração de um modelo de  $p$ -medianas aos SIGs ArcView, da ESRI (Environmental Systems Research Institute, Inc.) e de um sistema desenvolvido no INPE, denominado SPRING. Busca-se instalar três antenas de forma a fazer a maior cobertura do serviço de telefonia móvel no município de São José dos Campos-SP.

Em Resende & Werneck (2002b) foi abordado um problema de localização não-capacitado de  $p$ -medianas e é sugerida uma heurística multi-híbrida que combina elementos de várias metaheurísticas tradicionais como *scatter search* e busca tabu — que faz uso intenso do *path relinking*.

Bezerra (2008) apresenta um problema das  $p$ -medianas, em formulações sequenciais e paralelas onde o objetivo é encontrar os pontos onde instalar facilidades. Em seguida, gera-se a região de atendimento associada, conhecida como *cluster*. Na

geração dos *clusters*, é utilizada uma heurística de designação, a qual visa solucionar problemas de múltiplas facilidades, relacionando cada ponto de demanda à facilidade mais próxima. A geração de *clusters* é refinada pela heurística de localização–alocação.

Localizar  $p$ –centros constitui o problema de procurar numa rede  $p$  locais de forma que cada ponto de demanda recebe o serviço a partir da instalação do centro (local) mais próximo. Em outras palavras, procura-se minimizar a distância (ou custo) entre a atividade e o local mais distante dentro da rede. O objetivo é minimizar a distância máxima para todos os pontos de demanda. Este problema equivale ao de cobrir todos os pontos de uma área com  $p$  círculos de menor raio possível. É conhecido também como problema “minimax”, mais apropriado para localizar instalações de interesse do setor público – representações, hospitais, escolas, corpo de bombeiros, etc.

### 2.2.3 O problema dinâmico de localização

O problema dinâmico de localização lida com a possibilidade de expandir ou reduzir a capacidade máxima disponível em qualquer das instalações que operam em dado período do horizonte de planejamento. Existem também problemas de localização dinâmica em que surge a necessidade, no horizonte de planejamento, de desativar instalações e ativar outras facilidades em pontos distintos do espaço ou região de localização.

Owen & Daskin (1998) destacam que, devido a sua natureza estratégica, os problemas de localização necessitam também que características dinâmicas e/ou estocásticas sejam consideradas. A formulação dinâmica trata a questão do horizonte de tempo envolvido na localização. A formulação estocástica tenta captar as incertezas nos parâmetros de entrada do problema, podendo ser tratada de duas formas: através da abordagem probabilística e da abordagem de cenários.

Em Cortes & Paula Junior (2005) descreve-se um modelo de planejamento de localização. Esse trabalho considera que a análise de localização da atividade econômica é realizada em nível macro-geográfico e que, uma vez instalada, a atividade não poderá mudar de local durante o horizonte de tempo em estudo, ou seja, não é aceita uma realocação. Assume-se, também, que as condições futuras são as mais

prováveis de ocorrer e que as funções objetivo podem ser estabelecidas a partir de determinados fatores locais. Os objetivos do modelo são fundamentados nos fatores que mais influenciam a decisão local de atividades econômicas: dois fatores quantitativos (custo e proximidade dos mercados consumidores) e um fator qualitativo (benefícios agregados). Para solução do problema, foi proposto um algoritmo genético que considera uma função escalar relacionada a um problema auxiliar com um único objetivo (formada pela soma ponderada dos três objetivos) para ser uma medida de desempenho das soluções. Considerou-se, também, a correção e penalização das soluções que não respeitam as restrições do problema. O algoritmo genético de correção e penalização foi comparado com um algoritmo genético tradicional (sem correção e penalização). Sob as mesmas condições, o algoritmo de correção e penalização alcançou uma melhor qualidade de soluções elite.

#### 2.2.4 Problemas com múltiplos critérios

No decorrer da abordagem de vários tipos de problemas de localização verificou-se a necessidade prática de considerar simultaneamente vários critérios, o que deu origem aos problemas multiobjetivos ou com critérios múltiplos de localização.

Segundo Cortes (2003), a empresa que decidir por uma análise multicritério poderá melhorar sua vantagem competitiva, pois ao incorporar à análise local uma grande variedade de aspectos de decisões relevantes obtém-se uma representação mais próxima da situação real. No entanto, como as implicações das decisões locais são, em geral, de longo prazo, uma localização pode tornar-se um erro de investimento no futuro devido às mudanças ambientais, mesmo considerando uma abordagem de múltiplos critérios. Para tornar tal empreendimento lucrativo, é conveniente selecionar locais que não apenas tenham boa representação de acordo com o momento atual, mas que continuarão sendo lucrativos pelo período em que a atividade estiver em operação ou pelo menos durante o horizonte de tempo considerado no planejamento, mesmo em situações de mudanças ambientais, migrações populacionais e evolução das tendências de mercado. Nesse contexto, determinar as melhores localizações em um horizonte de planejamento é um importante desafio para a manutenção da competitividade da empresa, representando um dos mais importantes determinantes



para o sucesso.

Nesse mesmo trabalho, um problema de localização de atividades econômicas de base tecnológica é representado por um modelo linear 0-1, multiobjetivo e dinâmico. A formulação propõe minimizar os custos de implantação das atividades econômicas, minimizar o tempo de acesso dessas atividades aos seus clientes e maximizar a obtenção de benefícios agregados às instalações e suas conexões, durante o horizonte de planejamento. Além disso, o processo de localização atende a um conjunto de restrições operacionais, de orçamento, bem como à exigência de interligação dos períodos de tempo associados ao aspecto dinâmico do problema. Para solução do modelo, foi proposto um algoritmo genético e de sua adaptação à abordagem multiobjetivo do problema tratado nessa tese. Foram utilizados operadores de correção, apropriados e particulares, assim como incluiu a penalização das soluções inviáveis através de uma verificação sequencial das restrições. Ao final do processo evolutivo, as soluções não-dominadas, extraídas do conjunto das melhores soluções, são fornecidas à avaliação do decisor. Experimentos computacionais extensos foram realizados e seus resultados mostraram-se promissores, tanto em relação a aproximações de uma solução do problema em si, quanto com vistas à vantagem de se utilizar, via metáfora biológica, a inteligência da natureza em favor do cálculo de uma solução de problemas de modelagem computacional.

Lopes & de Almeida (2008), descrevem o problema de localização e, particularmente, da localização de unidades de serviço, segundo as diferentes visões encontradas na literatura: as variantes do problema e os métodos de resolução mais difundidos. Foi proposta a utilização do método multicritério SMARTER (Simple Multi-Attribute Rating Technique using Exploiting Rankings) para a solução de problemas, de modo a considerar, durante o processo decisório, os múltiplos atributos envolvidos e a subjetividade inerente à definição da localização de unidades de serviço. Uma aplicação numérica com números fictícios, porém num contexto realístico foi realizada onde o método SMARTER mostrou-se bastante eficaz na resolução do problema, pois se apresenta de forma simples e metodologicamente estruturado no que concerne ao tratamento multicritério do problema.

Um modelo multiobjetivo é proposto em Cortes (1998) para abordar de forma mais realista um problema de localização de plataformas de produção de petróleo, como

também dos poços produtores, cujos objetivos são: minimizar os custos de construção/instalação das plataformas e os custos de conexão do poço à plataforma; maximizar a produção de petróleo de um dado campo petrolífero; e minimizar o impacto de possíveis danos ambientais. Um problema monobjetivo auxiliar é construído considerando a função escalar ponderada aumentada de Tchebychef para definir o objetivo. Para auxiliar a resolução de tal problema é apresentado um algoritmo iterativo, baseado no método de enumeração implícita, que determina um conjunto de soluções eficientes para serem avaliadas pelo decisor.

### 2.2.5 Os problemas capacitado e não-capacitado de localização

Os problemas de localização também podem ser classificados como capacitados e não-capacitados. Os problemas que levam em conta a capacidade das instalações são ditos capacitados.

Um modelo simples do problema capacitado de localização é representado pelas relações (2.1) a (2.6), em que  $I$  é o conjunto de pontos elegíveis onde instalações poderão ser abertas. A escolha desse subconjunto de pontos de  $I$  em que as instalações serão abertas, é uma tarefa do algoritmo.  $J$  é o conjunto de clientes a serem atendidos pelas, possivelmente diversas, instalações abertas.

$$\text{minimizar } \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} + \sum_{i \in I} f_i y_i \quad (2.1)$$

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \quad (2.2)$$

$$x_{ij} \leq y_i, \quad \forall i \in I, \forall j \in J \quad (2.3)$$

$$\sum_{j \in J} D_j x_{ij} \leq B_i y_i, \quad \forall i \in I \quad (2.4)$$

$$y_i \in \{0, 1\}, \quad \forall i \in I \quad (2.5)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.6)$$

O significado da função e das relações do modelo é o que segue. (2.1) é a função critério representada pela soma de uma parcela de custos (ou distâncias) de transporte entre os clientes e as facilidades, com outra parcela, sendo esta de custos fixos das instalações. (2.2) garante que cada cliente será atendido por uma única

instalação. (2.3) representa a lógica pela qual clientes só serão atendidos a partir de instalações abertas ( $y_i = 0 \Rightarrow x_{ij} = 0$ , ou, se a instalação  $i$  está fechada, não há fluxo de bens ou serviços de  $i$  para o cliente  $j$ ). (2.4) é a restrição de capacidade em que a quantidade total demandada pelos clientes não pode superar a capacidade da instalação que os atende. Em (2.5),  $y_i = 1$  se a  $i$ -ésima instalação é aberta,  $y_i = 0$ , caso contrário. Em (2.6),  $x_{ij} = 1$  se a instalação  $i \in I$  atende o cliente  $j \in J$ , e  $x_{ij} = 0$ , caso contrário.

O modelo representado pelas linhas de (2.1) a (2.6), sem a restrição (2.4), é conhecido como problema de localização não-capacitado. Nesse caso, a capacidade das instalações não é levada em conta. Para mais detalhes, veja Amorim (2011) e Resende & Werneck (2002b). Acrescentando ao modelo (2.1) a (2.6) a restrição  $\sum_{i \in I} y_i = p$ , dá origem a uma versão do problema de localização de  $p$ -medianas. Se a restrição acrescentada for  $\sum_{i \in I} y_i \leq p$ , o modelo representa o problema generalizado de localização de  $p$ -medianas em que serão abertas, no máximo,  $p$  instalações. Como descritos, estes são os problemas capacitados de localização de  $p$ -medianas. Sem a restrição (2.4), tem-se os problemas não-capacitados de  $p$ -medianas.

O problema de Localização Capacitado de Custo Fixo busca, segundo Filho et al. (2005), definir um conjunto de facilidades (a partir de um conjunto de candidatas) que atenderão um outro conjunto de clientes geograficamente dispersos, considerando a minimização global dos custos relativos à demanda — distância entre as facilidades e os clientes — e a soma dos custos fixos das facilidades utilizadas.

Uma variação do problema generalizado de  $p$ -medianas, agora envolvendo capacidades, é apresentada pelas relações (2.7) a (2.17), de acordo com a descrição a seguir. Tem-se um conjunto  $I$  de fábricas já instaladas, um conjunto de pontos candidatos a *hubs* (CD's ou armazéns intermediários), e um conjunto  $K$  de clientes com localização conhecida. O modelo a seguir é o de um problema de localização cujo objetivo é o de escolher, no máximo,  $p$  pontos para instalação de *hubs* visando o atendimento dos clientes finais.

Um caso particular do modelo — (2.7) a (2.17) — surge quando  $|I| = 1$ , conhecido como “problema generalizado de localização de  $p$ -medianas de fonte única”. No contexto industrial, este modelo se associaria à existência de uma única fábrica que

produz e fornece para todos os clientes.

$$\text{minimizar } \sum_{i \in I} \sum_{k \in J} d_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} d_{jk} x_{jk} + \sum_{j \in J} f_j y_j \quad (2.7)$$

$$\sum_{i \in I} x_{ij} \geq 1, \quad \forall j \in J \quad (2.8)$$

$$\sum_{j \in J} x_{jk} = 1, \quad \forall k \in K \quad (2.9)$$

$$\sum_{j \in J} y_j \leq p, \quad (2.10)$$

$$x_{ij} \leq y_j, \quad \forall i \in I, \forall j \in J \quad (2.11)$$

$$x_{jk} \leq y_j, \quad \forall j \in J, \forall k \in K \quad (2.12)$$

$$\sum_{i \in I} G_i x_{ij} \leq B_j y_j, \quad \forall j \in J \quad (2.13)$$

$$\sum_{k \in K} H_k x_{jk} \leq B_j y_j, \quad \forall j \in J \quad (2.14)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (2.15)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.16)$$

$$x_{jk} \in \{0, 1\}, \quad \forall j \in J, \forall k \in K \quad (2.17)$$

A função critério (2.7) minimiza o custo (distância) de transporte das fábricas para os *hubs*, somado ao custo de transporte dos *hubs* aos clientes finais, e somado ao custo fixo dos *hubs*.

A restrição (2.8) garante para cada *hub* pelo menos uma fábrica que lhe atenda. A restrição (2.9) garante para cada cliente um único *hub* que lhe atenda. (2.10) é a restrição generalizada de  $p$ -medianas. (2.11) impede que qualquer fábrica forneça para um *hub* não aberto e a (2.12) impede que qualquer cliente seja atendido por um *hub* fechado. A restrição (2.13) garante que o total enviado por todas as fábricas  $i \in I$  a um *hub*  $j \in J$  não supera a capacidade do *hub* em questão. (2.14) garante o fornecimento pelo *hub*  $j \in J$  a seus clientes em quantidade não superior à capacidade do *hub* em questão. As restrições (2.15), (2.16) e (2.17) restringem as variáveis a valores binários.

## 2.3 Abordagem genérica da complexidade dos problemas locacionais

Em geral os problemas de localização possuem complexidade elevada e envolvem um grande volume de dados. Essa complexidade é devida ao fato de a análise lidar com um conjunto muito grande de variáveis de decisão que se influenciam mutuamente. Uma situação comum ocorre quando se busca a minimização dos custos e a maximização da qualidade. Nesse ponto, a redução dos custos pode implicar na queda da qualidade — o que possibilita não atender a um dos objetivos. Nesses casos, é necessário utilizar ferramentas de apoio à decisão que considerem o *tradeoff* entre os fatores relevantes que afetam a tomada de decisão. Além disso, o número de possíveis alternativas a serem analisadas e comparadas é muito alto, mesmo para problemas de porte modesto. É comum ter que se trabalhar com centenas de produtos, centenas de locais em potencial para terminais, centros de distribuição ou fábricas, dezenas de fornecedores, múltiplos modais de transporte e milhares de clientes.

Esses números dão uma ideia do volume de dados a ser manuseado, já que a análise requer informações detalhadas sobre a demanda, custos de transporte, custos e taxas de produção, custos fixos das instalações, localização dos clientes, localização dos atuais e prováveis pontos de estocagem e suprimento etc.

Apesar das dificuldades parecerem grandes, atualmente estão disponíveis vários métodos computacionais que facilitam a modelagem e solução deste tipo de problema.

## 2.4 Complexidade do problema não-capacitado de localização

A complexidade do problema não-capacitado de localização será examinada relacionando-o ao problema de recobrimento que é um tópico comum no contexto da otimização combinatória ou da otimização em números inteiros.

### 2.4.1 O problema combinatório de recobrimento

O problema do recobrimento pode ser definido a partir de:

- Um conjunto  $X = \{1, 2, \dots, n\}$  de requisitos que devem ser totalmente atendidos ou cobertos
- Um conjunto de atividades — diga-se  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m$  — em que algumas atividades, ou todas, cobrem o conjunto de todos os requisitos de  $X$
- Um custo associado a cada atividade

Qualquer atividade  $\mathcal{A}_i$  pode não cobrir o conjunto  $X$  de todos os requisitos. Entretanto, uma combinação de  $\mathcal{A}_i$ 's pode cobrir  $X$  inteiramente. Por exemplo, considera-se o conjunto de requisitos

$$X = \{1, 2, 3, 4, 5, 6\}$$

Considera-se as atividades

$$\mathcal{A}_1 = \{1, 2, 6\}, \mathcal{A}_2 = \{1, 3, 5\}, \mathcal{A}_3 = \{1, 5\}, \mathcal{A}_4 = \{4, 5, 6\}, \mathcal{A}_5 = \{3, 4, 6\}$$

Observa-se que nenhuma das atividades  $\mathcal{A}_1, \mathcal{A}_2$  e  $\mathcal{A}_4$ , isoladamente, cobre  $X$ . Entretanto,

$$\mathcal{A}_1, \mathcal{A}_2, \text{ e } \mathcal{A}_4$$

correspondem a uma combinação de atividades que, em conjunto, cobrem  $X$ . Neste caso,

$$X = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_4$$

Em relação a um problema de localização,  $X$  poderia ser o conjunto dos pontos, nós, vértices ou cidades com o requisito de que devem ser atendidos por algum CD<sup>1</sup>.  $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$  e  $\mathcal{A}_5$  são atividades de atendimento a possíveis *clusters* de pontos, nós, vértices ou cidades — ou seja, um CD poderia associar-se a  $\mathcal{A}_1$ , outro CD corresponderia a  $\mathcal{A}_2$ , e assim por diante.

O problema do recobrimento consiste em determinar uma combinação de atividades  $\mathcal{A}_i$ 's que pode, em conjunto, cobrir todos os requisitos em  $X$  e, ainda, minimizar uma função critério que traduza o custo de recobrimento.

---

<sup>1</sup>Centro de distribuição.

O problema de recobrimento traduzido por um modelo de otimização em números inteiros binários, torna-se:

$$\text{minimizar } \sum_{i=1}^m c_i x_i \quad (2.18)$$

$$\sum_{i=1}^m a_{ij} x_i \geq 1, \quad j = 1, \dots, n \quad (2.19)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, m \quad (2.20)$$

onde  $c_i$  é custo da atividades  $\mathcal{A}_i$ . A relação (2.19) traduz o fato de que para cada requisito  $j$  deve ser encontrada pelo menos uma atividade  $\mathcal{A}_i$  que o cubra. Usando o exemplo dado,  $a_{ij}$  tem o significado a seguir.

$$a_{ij} = \begin{cases} 1, & j \in \mathcal{A}_i \\ 0, & j \notin \mathcal{A}_i \end{cases} \quad \text{relativo ao exemplo isto significa} \quad (2.21)$$

1	2	3	4	5	6	← $j$
1	1	0	0	0	1	← $\mathcal{A}_1$
1	0	1	0	1	0	← $\mathcal{A}_2$
1	0	0	0	1	0	← $\mathcal{A}_3$
0	0	0	1	1	1	← $\mathcal{A}_4$
0	0	1	1	0	1	← $\mathcal{A}_5$
⋮	⋮	⋮	⋮	⋮	⋮	⋮

A matriz  $\mathcal{A} = (a_{ij})$ , (2.21), para o exemplo usado, tem  $m = 2^6 = 64$  linhas ou 63 linhas quando se exclui  $\mathcal{A}_i = \emptyset$ . A restrição (2.19) exige, em cada coluna  $j$  da matriz (2.21), a presença de pelo menos uma entrada igual a 1 para garantir que o elemento  $j \in \{1, 2, 3, 4, 5, 6\}$  seja coberto por um dos  $\mathcal{A}_i$ 's. Na solução,  $x_i = 1$  acarreta que, para um dado  $j \in \{1, 2, \dots, n\}$ ,  $j \in \mathcal{A}_i$ .  $x_i = 0$ , caso contrário.

Mais detalhes sobre a definição do problema de recobrimento estão em Nemhauser & Wolsey (1988) e em Chen et al. (2011).

“O problema do recobrimento de custo mínimo é NP–Hard.” A demonstração pode ser encontrada em Cheriyan & Ravi (1998), Garey & Johnson (1979) ou Papadimitriou & Steiglitz (1998).

## 2.4.2 A equivalência entre o problema de recobrimento e o PLNC

Considera-se o modelo

$$\text{minimizar } \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \quad (2.22)$$

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \quad (2.23)$$

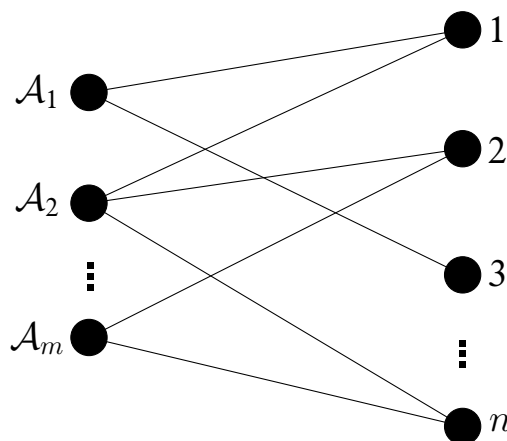
$$x_{ij} \leq y_i, \quad \forall i \in I, \forall j \in J \quad (2.24)$$

$$y_i \in \{0, 1\}, \quad \forall i \in I \quad (2.25)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.26)$$

O problema definido por (2.22) a (2.26), é conhecido como “de localização não-capacitado — PLNC”. Em Cheriyan & Ravi (1998) é demonstrada uma “equivalência” do PLNC ao problema de recobrimento no sentido de que uma solução ótima do PLNC fornece uma solução ótima para o problema de recobrimento. Esta equivalência ocorre de acordo com a descrição a seguir.

Em relação a uma instância do problema de recobrimento, pode-se construir uma instância do PLNC tal que uma solução ótima desse PLNC provê uma solução ótima para o problema de recobrimento. Em primeiro lugar, constrói-se um grafo bipartido, conforme Figura 2.3, baseado na instância do problema de recobrimento:



**Figura 2.3:** Relacionando o problema do recobrimento ao PLNC

para cada ponto  $j$  existe um nó  $j$  do “lado direito” do grafo bipartido, e para cada conjunto  $A_i$ ,  $A_i \subseteq \{1, 2, \dots, n\}$ , existe um nó  $A_i$  do “lado esquerdo” do grafo bipartido.



Existe uma aresta de  $\mathcal{A}_i$  para  $j$  se, e somente se, o ponto  $j$  pertence ao conjunto  $\mathcal{A}_i$  — isto é, se, e somente se, o conjunto  $\mathcal{A}_i$  cobre o ponto  $j$ . A instância do PLNC é como segue: o conjunto de clientes é  $\{1, 2, \dots, n\}$ . O conjunto de locais em potencial é  $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m\}$ . O custo  $c_{ij}$  é considerado zero se a aresta de  $\mathcal{A}_i$  para  $j$  está presente no grafo bipartido, ou ele é  $\infty$ , caso contrário.

Assim, o PLNC é o problema de abrir um número mínimo de instalações tais que cada cliente (ponto)  $j$  possa ser atribuído a (ou coberto por) uma instalação (conjunto)  $\mathcal{A}_i$  adjacente a ele — isto é,  $\mathcal{A}_i$  contém/atende  $j$ . Vê-se que uma solução da instância do PLNC (conjunto de locais a serem abertos) é ótima se, e somente se, a solução correspondente da instância do problema de recobrimento é ótima. Como o problema de recobrimento é NP-Hard, ver em Cheriyan & Ravi (1998), Nemhauser & Wolsey (1988) e em Chen et al. (2011), a equivalência entre os dois problemas permite concluir que “o PLNC é NP-Hard”.

## 2.5 Métodos de solução

Uma grande variedade de problemas difíceis de otimização pode ser encontrada em diferentes áreas do cotidiano como nas telecomunicações, logística, economia, transporte e produção. Teoricamente é possível resolver esses problemas de otimização enumerando todas as soluções e avaliando cada uma delas com relação a função objetivo, porém, na prática não é possível adotar uma estratégia deste tipo, dado o grande número de alternativas de soluções que podem existir, até mesmo para instâncias de pequeno porte. Além disso, frequentemente existem restrições de tempo para obter soluções com vistas a tomar decisões oportunas e, nesta perspectiva, métodos exatos<sup>2</sup> não devem ser adotados. Nas últimas décadas um grande esforço foi feito para desenvolver métodos que não requerem explicitamente enumerar todas as soluções. No entanto, para um grande número de aplicações, processos exatos continuam a ser intratáveis. Do ponto de vista prático, nessas situações, é desejável desenvolver algoritmos de solução que possam fornecer soluções de boa qualidade com esforço razoável. A qualidade das soluções pode ser testada usando limites. Este fato tem motivado o desenvolvimento de métodos aproximados, onde o objetivo é en-

---

<sup>2</sup>Veja a Seção 2.5.1.

contrar soluções de boa qualidade em quantidade razoável de tempo computacional. Um avanço notável na capacidade de resolver a difícil aproximação da otimalidade em problemas de otimização linear em números inteiros foi conduzido nos últimos anos. Esse avanço não é só devido à inovação tecnológica da informática, mas também à combinação de diversas metodologias, tais como novas heurísticas, melhorando a modelagem, teoria dos planos de corte e algoritmos de decomposição em otimização inteira. Embora esses métodos não tenham a otimalidade necessariamente garantida, eles podem fornecer soluções de boa qualidade de forma eficiente. Muitas vezes, a sua eficácia depende da forma como eles podem ser adaptados a uma situação específica de explorar a estrutura do problema em mãos (GARCÍA, 2001).

Esta seção apresenta uma breve revisão de alguns métodos de solução para problemas de otimização linear inteira.

### 2.5.1 Métodos exatos

Métodos exatos em geral produzem soluções com variáveis mistas – contínuas e inteiras. Os procedimentos exatos de maior sucesso são os de geração de colunas e os métodos enumerativos do tipo *branch & bound*, *branch & cut* e *branch & price*.

A metodologia *branch & bound* é utilizada para vários problemas NP-completos, como os problemas do caixeiro viajante e da mochila. É uma abordagem baseada na divisão do problema em vários subproblemas. A cada iteração, um subproblema novo é selecionado e processado. Um relaxamento desse subproblema atual é feito para se obter limites inferior e superior para ele. Se o limite é pior do que a melhor solução viável conhecida até então, o subproblema ativo é cortado. Mas, se a solução para o problema relaxado também é viável para o problema original, a melhor solução viável é atualizada e o subproblema é podado porque foi resolvido. Se o limite computado for melhor do que a melhor solução conhecida porém inviável com relação ao problema original, a ramificação é feita. O processo termina quando a lista de subproblemas fica vazia.

A eficiência do *branch & bound* depende muito da relaxação usada para computar limites e também da capacidade de identificar soluções primais de boa qualidade a fim de evitar que o número de subproblemas gerados torne-se rapidamente muito

grande. Diferentes técnicas de relaxamento, tais como relaxação Lagrangeana, que muitas vezes fornece limites superiores aos obtidos com a relaxação de otimização linear, são frequentemente utilizados no *branch and bound*. Além disso, diferentes tipos de heurística primal são utilizados a fim de proporcionar soluções viáveis de boa qualidade numa fase precoce do processo de enumeração.

O método *branch & cut* é uma variação do *branch & bound*. No *branch & cut*, vários cortes são gerados e impostos em cada um dos nós da árvore de ramificação para obter um limite mais exato para o nó antes de podá-lo e ramificá-lo.

Souza & Asada (2012) apresentam um algoritmo *branch & cut* com novas inequações específicas ao problema de planejamento da expansão de redes de transmissão de energia elétrica. A técnica pode ser resumida em adicionar cortes válidos ao modelo inicial nos subproblemas gerados pela estrutura *branch and bound*, usando um critério de separação. Todas as inequações propostas no trabalho são válidas tanto para os modelos lineares como para os modelos não lineares do problema. Os testes computacionais mostraram a eficiência do método proposto quando aplicado a subsistemas reais brasileiros e ao sistema colombiano.

Outra variação do *branch & bound* é o método *branch & price*, que tem o objetivo de gerar colunas em cada nó de uma árvore de busca *branch & bound* para geração de novas variáveis não-básicas.

Pereira (2005) apresenta a implementação de um algoritmo *branch and price* para resolver problemas de localização de facilidades baseados no modelo matemático do problema de  $p$ -medianas. A abordagem tradicional de geração de colunas é comparada com uma nova proposta, onde o critério de custos relativos empregado na seleção de colunas é modificado pelo multiplicador da relaxação lagrangeana / surrogate. A eficiência da nova abordagem foi comprovada por testes computacionais envolvendo instâncias com até 900 vértices. Também foram realizados estudos com dados reais de problemas de máxima cobertura, formulados como problemas de  $p$ -medianas, cuja esparsidade nos vetores de custos e na matriz de restrições representa grande dificuldade para métodos baseados em geração de colunas.

Em Longo (2005) é feita uma exploração de técnicas mais recentes da área de programação linear inteira e suas aplicações a problemas de roteamento de veículos.

Técnicas básicas de decomposição de problemas de programação linear e linear-inteira e de geração de colunas são revisadas e uma proposta de reformulação de problemas de programação linear-inteira alternativa àquela que gera o tradicional problema mestre de Dantzig–Wolfe, geralmente utilizados em abordagens por geração de colunas. A resolução de problemas de programação linear inteira neste contexto também é tratada e com a descrição do algoritmo *branch and bound* e das variações *branch and cut*, *branch and price* e *branch and cut and price*. Uma técnica denominada de Geração Projetada de Colunas e sua aplicação ao problema de Roteamento de Veículos com Restrição de Capacidade é apresentada. Nesse trabalho também é abordada a resolução do problema de Roteamento de Veículos sobre Arcos, através de sua transformação no primeiro problema citado e o uso de um algoritmo *branch and cut and price*. Finalmente, é proposto um novo problema na área de redistribuição de veículos de aluguel, para o qual é proposta uma formulação segundo uma abordagem por geração de colunas. São apresentados, ainda, procedimentos para a geração de colunas e resultados computacionais obtidos com um algoritmo *branch and price* para essa formulação.

### 2.5.2 Introdução às metaheurísticas

Dada a importância do tema “metaheurísticas” neste trabalho, no APÊNDICE B foi feita uma descrição detalhada, embora não exaustiva, a respeito deste tópico, apresentando o pseudocódigo das principais metaheurísticas em uso e abordadas na ampla literatura que existe sobre elas. A introdução que aqui se faz torna-se necessária dentro da sequência escolhida para descrever a pesquisa que é a base desta dissertação.

Metaheurísticas são procedimentos com capacidade de produzir soluções aproximadas de boa qualidade para os problemas em geral e, em particular, para os de localização. A expressão “soluções aproximadas de boa qualidade” deve ser acompanhada da informação de que é frágil o compromisso desses procedimentos em alcançar o ótimo do problema em questão. Entretanto, as soluções produzidas são plenamente satisfatórias para projetos de engenharia.

Segundo Bueno (2000), métodos heurísticos são procedimentos mais simples e flexí-

veis que os métodos exatos. Por isso, são menos exigentes em termos de tempo de execução e ocupação de memória. Ele diz ainda que na construção de uma heurística, é necessário encontrar o compromisso adequado entre a qualidade das soluções geradas e o esforço computacional. Pode-se estabelecer algumas características para uma boa heurística:

- capacidade de encontrar soluções de qualidade
- baixa probabilidade de gerar solução de má qualidade
- rapidez de processamento
- simplicidade de implementação
- ausência de restrições em relação a soluções de partida – se possível, aceitar soluções iniciais ineficazes.

Entre as metaheurísticas com maior destaque, pode-se citar GRASP — que será examinada no Capítulo III — busca tabu, algoritmos genéticos, *simulated annealing*, busca em vizinhança variável e colônia de formigas, entre outras, que são abordadas em detalhe no APÊNDICE B.

Para Glover & Laguna (1997) a metaheurística busca tabu é baseada na premissa de que a resolução de problemas, a fim de qualificar-se como inteligente, deve incorporar memória adaptativa e exploração da resposta. O recurso de memória adaptativa da busca tabu permite a implementação de procedimentos que são capazes de explorar o espaço solução economicamente e de forma eficaz. Uma vez que escolhas locais são guiadas por informações coletadas durante o processo, a busca tabu contrasta com projetos sem memória que dependem fortemente de processos semi-aleatórios que implementam uma forma de amostragem. Exemplos de métodos sem memória incluem heurísticas semi-gulosas e as abordagens de destaque “genéticas” e “*annealing*” inspiradas em metáforas da biologia e da termodinâmica (física). A memória adaptativa também contrasta com os projetos das típicas memórias rígidas das estratégias do *branch and bound*.

O algoritmo genético, proposto inicialmente por Holland (1975a), tem o intuito de aplicar a teoria da evolução das espécies de Darwin, utilizando os conceitos de gene, cromossomo, cruzamento, mutação e seleção, em qualquer problema de otimização. É

um procedimento adaptativo que simula a evolução de uma população de indivíduos, cada um deles representando uma possível solução de um dado problema de otimização. Cada indivíduo é caracterizado por um cromossomo que codifica as características relevantes ao problema. A população evolui iterativamente através de processos de seleção dos indivíduos mais bem adaptados, segundo uma dada função objetivo. Estes indivíduos têm seus cromossomos combinados de uma forma que possibilite a obtenção de novas gerações de indivíduos cada vez mais aptos. Para que seja induzida uma maior diversidade entre os indivíduos, são utilizados, também, mecanismos de mutação. Esse mecanismo de evolução induz a busca em diferentes áreas do espaço de solução, o que faz com que o algoritmo não fique preso a um ótimo local, favorecendo assim, a busca do ótimo global. Porém, por trabalhar com uma população de indivíduos e não com um único, o processo pode ser lento, além da necessidade de ajuste dos seus parâmetros para beneficiar sua eficácia. A probabilidade de sobrevivência no meio ambiente desse novo indivíduo depende da herança de características boas ou ruins adquiridas. Ao fim de todo o processo, o melhor indivíduo encontrado corresponderá a uma solução heurística para o problema.

Segundo Rezende (2005), os algoritmos genéticos diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos:

- Trabalham com a codificação do conjunto de parâmetros e não com os próprios parâmetros
- Trabalham com uma população e não com um único ponto
- Utilizam informação de custo e recompensa, e não derivadas de outro conhecimento auxiliar
- Utilizam regras de transição probabilísticas e não determinísticas.

Bezerra (2008) apresenta um estudo comparativo entre aplicações das metaheurísticas algoritmos genéticos (genetic algorithm), otimização por colônia de partículas discretas (discrete particle swarm optimization) e otimização por saltos de rãs (jumping frog optimization) à solução do problema das  $p$ -medianas, em formulações sequenciais e paralelas. A metodologia utilizada consiste na aplicação da metaheurística ao problema das  $p$ -medianas, a fim de encontrar os pontos de facilidades e, em seguida,

gerar a região de atendimento associada, conhecida como *cluster*. Na geração dos *clusters*, é utilizada uma heurística de designação que visa solucionar problemas de múltiplas facilidades, relacionando cada ponto de demanda à facilidade mais próxima. A geração de *clusters* é refinada pela heurística de localização–alocação (location–allocation heuristic). Testes computacionais, realizados com instâncias da literatura, mostram que as soluções encontradas através da metaheurística otimização por saltos de rãs são superiores, em todos os casos, às encontradas através das metaheurísticas algoritmos genéticos e otimização por colônia de partículas discretas.

Em Cortes (2003), os algoritmos genéticos são utilizados para solução de um problema de localização de atividades econômicas de base tecnológica representado por um modelo linear 0-1, multiobjetivo e dinâmico.

A metaheurística *scatter search* (busca dispersa) opera em um conjunto de soluções chamado de “conjunto de referência” combinando essas soluções para criar outras, de forma a melhorar a solução atual. Ao contrário do algoritmo genético, o tamanho do conjunto de referência é relativamente pequeno. O algoritmo combina mecanismos de exploração com uma tentativa explícita de forçar a diversidade para o sistema. O algoritmo começa com um conjunto de indivíduos iniciais semeados fornecidos pelo usuário. Em seguida, o algoritmo tenta produzir um grande número de indivíduos aleatórios, a partir das sementes, que são muito diferentes uns dos outros. Estes, além das sementes, constituem a população.

Resende & Werneck (2002b) apresentam uma heurística multi-híbrida que combina elementos de várias metaheurísticas tradicionais como *scatter search* e busca tabu (que faz uso intenso do *path relinking*) para o problema de localização não-capacitado de  $p$ –medianas. Os resultados empíricos sobre instâncias da literatura evidenciam a robustez do algoritmo que executa, pelo menos, tão bem como outros métodos, e, muitas vezes melhor, tanto em termos de tempo de processamento quanto na qualidade da solução. Em todos os casos, as soluções obtidas pelo método estavam dentro de 0,1% dos limites superiores mais conhecidos.

Em Glover et al. (2003) são apresentados vários trabalhos que utilizam a abordagem *scatter search*.

A metaheurística *simulated annealing*, introduzida por Cerny (1985), é baseada no

processo físico de *annealing*, que lhe serve de metáfora, denotando um processo térmico para obter estados de baixa energia de um sólido submetido a calor elevado. Primeiro, a temperatura é aumentada para um valor máximo, onde as partículas se organizam de forma aleatória. Depois, a temperatura é cuidadosamente diminuída até a temperatura ambiente, onde as partículas ficam dispostas de forma altamente estruturadas e a energia do sistema é mínima. As soluções são representadas pelos diversos estados físicos do sólido, e o custo, pela energia do sistema.

Em Haeser & Gomes-Ruggiero (2008) é descrita a teoria da estratégia *simulated annealing* e um método híbrido foi proposto para otimização global de problemas contínuos canalizados. A cada etapa deste método é realizada uma fase heurística, na qual o *simulated annealing* foi empregado, e uma fase local, na qual foi utilizado o método GENCAN. O objetivo desse procedimento foi explorar as propriedades de obtenção do ótimo global de *simulated annealing*, e acelerar esta estratégia acionando um procedimento de otimização local que possua boas propriedades de convergência para ótimos locais. Os resultados numéricos obtidos atestaram a eficiência do procedimento.

A metaheurística GRASP (Greedy Randomized Adaptive Search Procedure) (ver em Feo & Resende (1995)) vem produzindo excelentes resultados na solução de problemas combinatórios importantes. É um procedimento iterativo do tipo *multistart* em que cada iteração GRASP consiste de duas fases: uma de construção que produz soluções viáveis, e outra de busca local (ou de melhoria) em que um ótimo local é procurado na vizinhança da solução construída. Variações da GRASP como RGRASP (GRASP REATIVA), conforme García (2001), e GRASP com religamento de caminhos (*path relinking*), de acordo com Resende & Werneck (2002a), Boudia et al. (2007), Caires et al. (2010) e Frinhani (2011) têm produzido resultados interessantes neste contexto.

Filho et al. (2005) apresentam o desenvolvimento de um algoritmo híbrido para o Problema de Localização Capacitado de Custo Fixo (PLC), e também um sistema de testes desenvolvido para experimentar instâncias da OR-Library. Uma explicação detalhada da GRASP pode ser vista no trabalho. Os resultados encontrados na resolução de várias instâncias da OR-Library apresentaram-se muito bons, tanto para a GRASP Simples quanto para a GRASP Reativa (Adaptativa), sendo que esta última apresen-



tou, no geral, resultados melhores e com um tempo computacional menor.

Em Lamosa et al. (2010) foi resolvido um problema de posicionamento de sensores, considerando limitação na quantidade a ser instalada. Este pode ser visto como um problema de localização de facilidades de cobertura máxima, onde as facilidades são os sensores e o objetivo é determinar o posicionamento dos mesmos de forma a maximizar a área de cobertura de uma região. Neste trabalho, foi considerado que as facilidades possuem características distintas. Com isso, busca-se determinar o melhor posicionamento das diferentes facilidades, respeitando a limitação individual existente. Um algoritmo GRASP foi utilizado como método de solução. O algoritmo foi aplicado em uma região considerando somente 2 casos de teste com o objetivo de ilustrar a aplicação do método e bons resultados foram encontrados em um tempo computacional aceitável.

Nogueira et al. (2006) usam a GRASP para solução do problema da seleção de portfólios para o pequeno investidor (PSPPI), sob a ótica da eficiência média-variância, formulado como um problema de programação inteira quadrática, de difícil solução computacional — NP-Hard.

Leite et al. (2000) utilizam uma GRASP-3D para resolver o problema de preencher um container, com máxima ocupação possível, por itens de faces retangulares, em quantidades pré-definidas, de modo que os espaços vazios após o preenchimento sejam mínimos. Considera-se que os itens são de pesos próximos e de baixa densidade, para que não haja desequilíbrio durante o processo de empilhamento. É permitida a rotação dos itens no preenchimento do espaço.

Em Rangel et al. (2000) é realizada uma modificação na heurística GRASP aplicada ao problema quadrático de alocação, com uma proposta para aceitar ou não a solução inicial gerada na fase de construção, evitando uma busca que, eventualmente, exigiria muito esforço computacional. Tal proposta é baseada no cálculo dos custos normalizados em um intervalo de limites das soluções do problema quadrático de alocação.

# CAPÍTULO III

## GRASP

### 3.1 Introdução

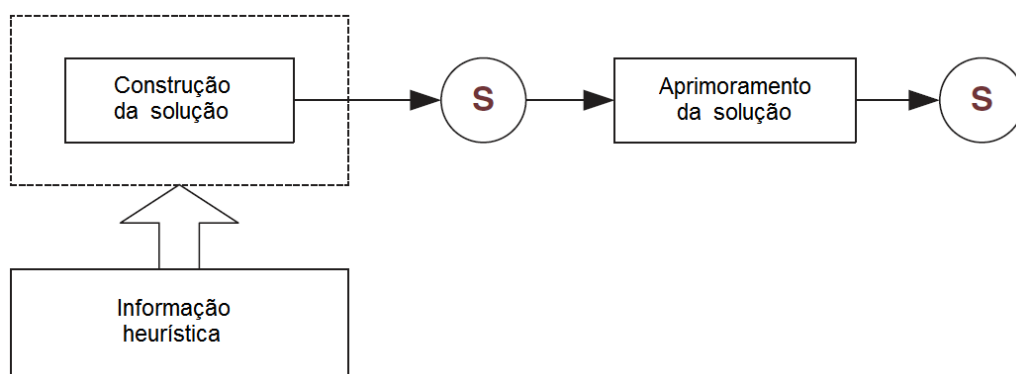
A GRASP (Greed Randomized Adaptive Search Procedure) é uma metaheurística gulosa do tipo multipartida<sup>1</sup> que adquiriu prestígio entre os métodos de solução aproximada por resolver com eficiência, em termos de tempo e ocupação de memória, problemas combinatórios de porte variado. Trata-se de procedimento em que a solução é paulatinamente construída. Cada iteração da GRASP é executada através de duas fases. Uma de construção e outra de busca local, (FEO & RESENDE, 1995). Na construção uma solução viável é produzida usando um algoritmo guloso aleatório, enquanto que na fase seguinte é aplicado um procedimento heurístico de busca local a partir da solução construída na primeira fase. Em geral, a GRASP faz a busca através de repetida construção de soluções. Isto é, em cada iteração a GRASP inicia com uma solução vazia e, iterativamente, adiciona elementos à solução parcial e retorna uma solução completa, que é, então, aprimorada por outro método — o de busca local. A metaheurística GRASP surgiu da necessidade de conciliar a simplicidade de ideias das heurísticas míopes ou gulosas com a limitação destas de gerarem a mesma e única solução cada vez que acionadas. De acordo com Hart & Shogan (1987), ideia semelhante à GRASP já havia sido explorada e apresentada por via de um procedimento que ficou conhecido como heurística semigulosa. Nessa proposta um enfoque guloso multipartida foi explorado, sem contudo usar procedimentos de busca local.

A Figura 3.1 representa o esquema geral da GRASP, em que estão destacadas as fases de construção da solução e o aprimoramento da solução construída. Observa-se que durante a fase de construção somente informação heurística é usada. Isto é, nenhuma informação do histórico de busca é incorporada pela fase de construção. A Figura 3.2 detalha, em particular, o esquema construtivo da GRASP, onde se vê

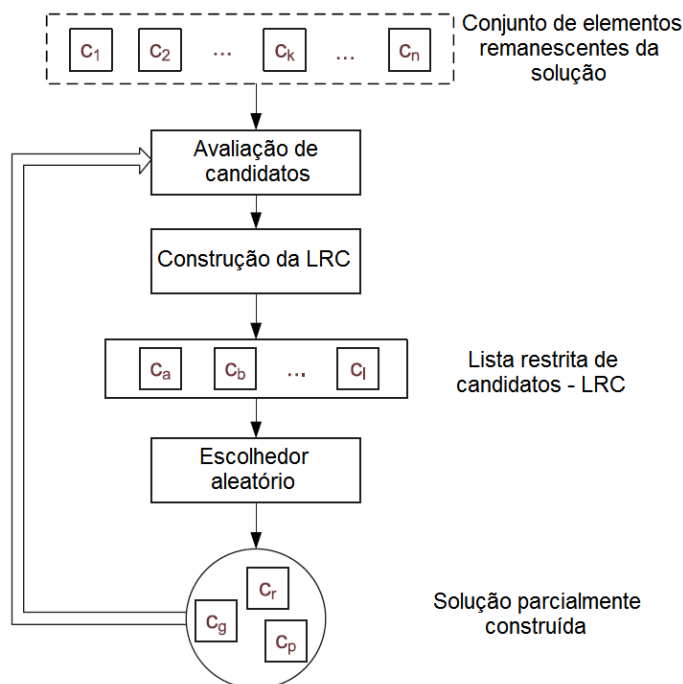
---

<sup>1</sup>Quando se usa um procedimento multipartida — ou *multistart* — são gerados pontos alternativos de partida, isto é, são filtrados os pontos de início. Em seguida, aciona-se o algoritmo de busca local e, finalmente, são examinadas as condições de parada.

que o processo guloso de construção pode ser conduzido por uma heurística que, a cada iteração, inclui elementos numa lista — ordenada de forma decrescente — até que ela atinja determinada cardinalidade, diga-se  $q \in \mathbb{N}$ ,  $q > 0$ . A partir dessa lista é gerado um subconjunto de elementos para compor a “Lista Restrita de Candidatos” — LRC. Essa lista é um componente fundamental da GRASP, e ela aparece no processo como a guardiã do lado probabilístico dessa metaheurística. O processo construtivo da GRASP amortece o ímpeto de gula dos procedimentos puramente gulosos.



**Figura 3.1:** Esquema geral de solução por via da GRASP, Günther et al. (2010).



**Figura 3.2:** Esquema construtivo da GRASP, Günther et al. (2010).

O passo de busca local é descrito como segue: como não há garantia de que a solução encontrada seja um ótimo local, uma busca é conduzida com vistas a aprimorar a solução, procurando na sua vizinhança uma solução melhor que a substitua. A

busca é dita “local” porque ela é conduzida na vizinhança da solução para a qual se procura uma melhoria.

A ideia de randomizar um algoritmo guloso permite dotar o método de aptidão para gerar várias soluções distintas. Isso torna o procedimento mais inteligente por permitir que a busca local possa ser aplicada várias vezes — uma para cada solução construída aleatoriamente. Este esquema se repete até que se alcance um dado número de iterações e a melhor solução encontrada seja apresentada como resultado final. Observa-se que as iterações são independentes, de modo que as soluções geradas não sofrem influência de outras soluções, nem as influencia. Essa independência das soluções sucessivamente geradas, libera o procedimento de realizar buscas na memória. Este enfoque torna-se vantajoso se a heurística construtiva alcançar, no espaço de busca, regiões promissoras que lhe permitam gerar ótimos locais de boa qualidade.

Os pontos a seguir ajudam a explicar, com reflexões adicionais, os aspectos fundamentais da GRASP.

- Como a GRASP é um método construtivo, ela começa com uma solução “vazia” e adiciona elementos a uma solução parcial até que ela fique completa, caracterizando aí o seu lado guloso. Após a fase de construção a solução pode ainda ser aprimorada em termos de busca na sua vizinhança, de modo que a fase de construção seja usualmente seguida por uma fase de aprimoramento/melhoria da solução, a qual é exclusivamente uma busca local.
- Os métodos gulosos não executam busca em memória o que, por um lado, agrega-lhes certa vantagem em termos de ocupação de espaço e gasto de tempo. Eles constroem cada solução de forma iterativa, avaliando os elementos remanescentes da solução parcial, adicionando-os de acordo com o desempenho — os elementos são adicionados desde que a solução seja aprimorada com a participação deles. Quando isso não é mais possível, o processo de construção é interrompido e a solução final é retornada. Não há, portanto, busca em memória na fase construtiva.
- A GRASP usa a assim chamada LRC que autoriza o algoritmo a fazer uma amostragem de soluções distintas a partir do espaço de soluções e, portanto,

desempenha, a partir daí, uma busca. A principal ideia do algoritmo consiste em relaxar a condição de “adicionar o elemento de melhor desempenho”. Ao invés disso, uma lista de candidatos promissores é construída. A partir desta lista um elemento é escolhido aleatoriamente e adicionado a atual solução parcial. Cria-se assim, uma tendência na direção de boas soluções e, ao mesmo tempo, soluções diferentes são construídas.

- O argumento registrado em Feo & Resende (1995) é expresso da seguinte forma: “Pelo uso de uma LRC são impostas variações nas soluções e na qualidade de cada uma delas. Considerando o tamanho da LRC como sendo 1, isto é, sempre é adicionado o elemento de melhor desempenho, a mesma solução é construída em cada iteração. Portanto, a variância na qualidade da solução é zero. Escolhendo aleatoriamente um elemento da LRC, é feita uma amostragem de soluções distintas e, portanto, a qualidade da solução final varia dentro de uma certa extensão. Assim, algumas soluções construídas são piores e outras melhores do que aquela solução de qualidade média.”
- Existe uma função de avaliação de candidatos,  $g$ , que determina a mudança na qualidade da solução que é induzida pelo respectivo elemento candidato. Matematicamente, a função  $g$  mapeia em números reais cada um dos elementos  $c_i$  do conjunto  $C$  dos elementos ainda não adicionados, fato traduzido por  $(g : C \rightarrow \mathbb{R})$ , o que reflete a mudança na função objetivo da solução parcial atual. Para a construção da lista necessita-se conhecer  $g_{\min}$  e  $g_{\max}$ , onde  $g_{\min} = \min\{g(c_i)\}, \forall c_i \in C$  e  $g_{\max} = \max\{g(c_i)\}, \forall c_i \in C$ . Portanto, sabe-se qual é o incremento mínimo e o máximo na qualidade da solução. Esta informação é usada para limitar o número de candidatos à LRC.
- A lista restrita de candidatos pode ser construída com base em dois critérios:
  - Critério 1: baseado na cardinalidade da LRC. Por este critério o método inclui os  $k$  melhores candidatos. Isto é, se  $k = 10$ , os 10 melhores elementos são incluídos na LRC.
  - Critério 2: baseado no valor. Por este critério usa-se um parâmetro  $\alpha$ ,  $\alpha \in [0, 1]$ , para restringir a entrada dos elementos candidatos na LRC.

Olhando para o critério 2 no caso de um problema de minimização, todos os

elementos candidatos que têm um limiar  $\mu = g_{\min} + \alpha(g_{\max} - g_{\min})$  são incluídos na LRC — de modo que o custo  $g(c_i)$  de um elemento deve estar no intervalo  $g(c_i) \in [g_{\min}, \mu]$  para ser incluído na LRC. Observa-se que se  $\alpha = 0$  o método de construção é guloso, e se  $\alpha = 1$  a construção é puramente aleatória.

- O tamanho da LRC tem influência importante no desempenho da GRASP. Em Resende & Ribeiro (2003) são descritos três enfoques diferentes para a escolha do  $\alpha$  no caso do critério 2 — baseado no valor — a ser usado.
  - Escolhe-se  $\alpha$  aleatoriamente a partir de uma distribuição uniforme e discreta de probabilidade — esta foi comprovada a alternativa mais promissora porque ela usa valores diferentes de  $\alpha$  que produzem alta variância.
  - Escolhe-se  $\alpha$  a partir de uma distribuição não-uniforme, decrescente e discreta de probabilidade
  - Fixa-se  $\alpha$  em um valor próximo à escolha puramente gulosa — esta foi comprovada a pior alternativa nos experimentos conduzidos por Resende & Ribeiro (2003).

A seguir, a metaheurística GRASP é apresentada através do Algoritmo 1.

---

**Algoritmo 1** GRASP.

---

**executa:** GRASP ( $f(\square)$ ,  $g(\square)$ ,  $N(\square)$ ,  $GRASP_{max}$ ,  $s$ )

1  $f^* \leftarrow \infty$

2  $s^* \leftarrow s$

3 **para** ( $iter = 1, \dots, GRASP_{max}$ ) **faça**

4   Construção( $g(\square)$ ,  $\alpha$ ,  $s$ )

5   BuscaLocal( $f(\square)$ ,  $N(\square)$ ,  $s$ )

6   **se** ( $f(s) < f^*$ ) **então**

7      $s^* \leftarrow s$

8      $f^* \leftarrow f(s)$

9   **fim se**

10 **fim para**

11  $s \leftarrow s^*$

12 Retornar  $s$

13 **fim** GRASP

---

O pseudo-código da fase de Construção é expresso pelo Algoritmo 2.

---

**Algoritmo 2** Construção.
 

---

**executa:** GRASP-CONSTRUÇÃO ( $g(\square)$ ,  $\alpha$ ,  $s$ )

- 1  $s^* \leftarrow \emptyset$
- 2 Inicializar o conjunto  $C$  de candidatos
- 3 **enquanto**  $C \neq \emptyset$  **faça**
- 4    $g_{min} = \min \{g(\theta) : \theta \in C\}$
- 5    $g_{max} = \max \{g(\theta) : \theta \in C\}$
- 6    $LRC = \{\theta \in C : g(\theta) \leq g_{min} + \alpha(g_{max} - g_{min})\}$
- 7   Selecionar, aleatoriamente, um elemento  $\theta \in LRC$
- 8    $s \leftarrow s \cup \{\theta\}$
- 9   Atualizar o conjunto  $C$  de candidatos
- 10 **fim enquanto**
- 11 Retornar  $s$
- 12 **fim** GRASP-CONSTRUÇÃO

---

Finalmente, a fase de Busca Local, ou Melhoria, é traduzida pelo Algoritmo 3.

---

**Algoritmo 3** Melhoria.
 

---

**executa:** GRASP-MELHORIA ( $f(\square)$ ,  $N(\square)$ ,  $s$ )

- 1  $s^* \leftarrow s$      $\{s^* \text{ é a melhor solução encontrada até o momento}\}$
- 2  $V = \{s' \in N(s) : f(s') < f(s)\}$
- 3 **enquanto**  $|V| > 0$  **faça**
- 4   Selecionar  $s \in V$
- 5   **se**  $f(s) < f(s^*)$  **então**
- 6      $s^* \leftarrow s$
- 7      $V = \{s' \in N(s) : f(s') < f(s)\}$
- 8   **fim se**
- 9    $s \leftarrow s^*$
- 10 **fim enquanto**
- 11 Retornar  $s$
- 12 **fim** GRASP-MELHORIA

---

A GRASP tem sido utilizada em problemas de otimização em números inteiros, tanto na sua concepção original quanto em algoritmos GRASP híbridos. Variações da GRASP, como RGRASP (GRASP REATIVA), conforme García (2001), e GRASP com religamento de caminhos (*path relinking*), de acordo com Resende & Werneck (2002a), Boudia et al. (2007), Caires et al. (2010) e Frinhani (2011) têm produzido resultados interessantes neste contexto.

### 3.2 Cálculo da função gulosa da GRASP para $p$ -medianas

Na construção da função gulosa da GRASP aparece um parâmetro  $\alpha$ , onde  $0 \leq \alpha \leq 1$ , que exerce o papel de controlador do tamanho da LRC. Como já foi dito, para  $\alpha = 1$ , o comportamento do algoritmo é puramente aleatório, e para  $\alpha = 0$ , tem-se um comportamento puramente guloso. Esse parâmetro controla, então, o grau de miopia e aleatoriedade do algoritmo. Se a cardinalidade da LRC for pequena, menor será o espaço de soluções analisado. Conseqüentemente, menores serão as chances de se escapar de um ótimo local. O  $\alpha$  pode ser, então, considerado como o principal parâmetro da GRASP, pois a qualidade das soluções depende fortemente de seu ajuste.

A função gulosa necessária na fase de construção da GRASP, relativa à  $i$ -ésima mediana,  $i \leq p$ , é definida por

$$g_i(C_i) = g_{min}^i + \alpha(g_{max}^i - g_{min}^i) \quad (3.1)$$

onde  $g_{min}^i$  é escolhido de tal maneira que  $g_{min}^i < g_{max}^i$  e

$$g_{max}^i(C_i) = \frac{\sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}, \quad |C_i| > 1 \quad (3.2)$$

ou

$$g_{max}^i(C_i) = \frac{f_i + \sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}, \quad |C_i| > 1 \quad (3.3)$$

Tanto na função (3.1), quanto nas funções (3.2) e (3.3),  $C_i$  representa um possível agrupamento da mediana  $i$ .  $\Delta_{ij}$  é a diferença entre a distância da mediana  $i$  à cidade de menor distância, e a distância da mediana  $i$  à cidade  $j$ . No caso da função (3.3),  $f_i$  é o custo fixo<sup>2</sup> da mediana  $i$ . Em qualquer dos dois casos, o agrupamento (com sua mediana) será tão melhor quanto menor for a média dos custos representada pelas funções.

Nas fórmulas (3.2) e (3.3), o  $-1$  no denominador se refere ao nó (cidade) mais próximo da mediana, que não participa do somatório. Para ela — cidade mais próxima — a

<sup>2</sup>Custos fixos são gerados por dispêndios necessários para se manter uma instalação aberta, como por exemplo, para custear contas de água e luz, mão de obra, aluguel, limpeza, equipamentos, etc.



distância adicional, em relação a distância mínima, é zero ( $\Delta_{ij} = 0$ ).

Para ilustrar o uso da função gulosa, considera-se uma matriz de distâncias para 10 cidades, de acordo com a Tabela 3.1. Aleatoriamente a cidade 8 é escolhida candidata a mediana.

A cidade mais próxima é a 3 que está a uma distância 5 da cidade 8. As demais cidades para serem atendidas por 8, percorrerão esses 5 mais uma distância adicional: a cidade 1 está distante 12:  $5 + 7$ . A cidade 2 está distante 18:  $5 + 13$ , e assim por diante. A seguir são feitos os outros cálculos com auxílio da Tabela 3.1.

**Tabela 3.1:** Matriz de distâncias para ilustrar o cálculo da função gulosa.

	1	2	<span style="border: 1px solid black; padding: 2px;">3</span>	4	5	6	7	<span style="border: 1px solid black; padding: 2px;">8</span>	9	10
1	0	*	*	*	*	*	*	*	*	*
2	*	0	*	*	*	*	*	*	*	*
3	*	*	0	*	*	*	*	*	*	*
4	*	*	*	0	*	*	*	*	*	*
5	*	*	*	*	0	*	*	*	*	*
6	*	*	*	*	*	0	*	*	*	*
7	*	*	*	*	*	*	0	*	*	*
→ <span style="border: 1px solid black; padding: 2px;">8</span>	12	18	<span style="border: 1px solid black; padding: 2px;">5</span>	6	14	9	10	0	8	40
	$5 + 7$	$5 + 13$	$5 + 0$	$5 + 1$	$5 + 9$	$5 + 4$	$5 + 5$	<i>mediana</i>	$5 + 3$	$5 + 35$
9	*	*	*	*	*	*	*	*	0	*
10	*	*	*	*	*	*	*	*	*	0

- $g_{max}^8 = \frac{\sum_{j \in C_8} \Delta_{8j}}{|C_8| - 1} = \frac{7 + 13 + 1 + 9 + 4 + 5 + 3 + 35}{8} = \frac{77}{8} = 9,625 \rightarrow 10$
- $g_{min}^8 = 0$
- $\alpha = 0,8$
- $g^8 = g_{min}^8 + 0,8 \times (g_{max}^8 - g_{min}^8) = 0,8 \times 10 = 8$

Escolha dos clientes da mediana candidata 8, com  $\alpha = 0,8$ . Em cada linha usa-se

$$\frac{\sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}.$$

- $\frac{1}{1} = 1 < 8$ , escolhe o 4

- $\frac{1+3}{2} = 2 < 8$ , escolhe o 9
- $\frac{1+3+4}{3} = 2, \dots < 8$ , escolhe o 6
- $\frac{1+3+4+5}{4} = 3, \dots < 8$ , escolhe o 7
- $\frac{1+3+4+5+7}{5} = 4 < 8$ , escolhe o 1
- $\frac{1+3+4+5+7+9}{6} = 5, \dots < 8$ , escolhe o 5
- $\frac{1+3+4+5+7+9+13}{7} = 6 < 8$ , escolhe o 2
- $\frac{1+3+4+5+7+9+13+35}{8} = 9,625 \not< 8$ , NÃO escolhe o 10

Neste caso, o agrupamento da mediana 8 é  $\{3, 4, 9, 6, 7, 1, 5, 2\}$ .

Escolha dos clientes da mediana candidata 8, com  $\alpha = 0,5$ . Neste caso,  $g^8 = g_{min}^8 + 0,5 \times (g_{max}^8 - g_{min}^8) = 0,5 \times 10 = 5$ .

- $\frac{1}{1} = 1 < 5$ , escolhe o 4
- $\frac{1+3}{2} = 2 < 5$ , escolhe o 9
- $\frac{1+3+4}{3} = 2, \dots < 5$ , escolhe o 6
- $\frac{1+3+4+5}{4} = 3, \dots < 5$ , escolhe o 7
- $\frac{1+3+4+5+7}{5} = 4 < 5$ , escolhe o 1
- $\frac{1+3+4+5+7+9}{6} = 5, \dots \not< 5$ , NÃO escolhe o 5

Neste caso, o agrupamento da mediana 8 é  $\{3, 4, 9, 6, 7, 1\}$ . Observa-se que quanto menor for o  $\alpha$  mais se restringe a escolha de cidades clientes da mediana candidata.

Os primeiros projetos de metaheurística GRASP trabalhavam com um  $\alpha$ ,  $0 \leq \alpha \leq 1$ , que era calibrado até se tornar um parâmetro fixo do método. Com a evolução da pesquisa pertinente, foram adotados esquemas reativos em que é permitido o  $\alpha$  sofrer variações durante as sucessivas iterações do algoritmo.

### 3.3 GRASP reativa

Segundo García (2001), uma das possíveis desvantagens dos algoritmos baseados em metaheurísticas é a falta de robustez, no sentido de que sua eficácia pode ser dependente da escolha de valores dos parâmetros. Isso geralmente implica em uma tarefa considerável para “ajustar” esses valores, a fim de obter melhor desempenho. Isto levou ao estudo dos chamados métodos reativos, que, em essência, são algoritmos baseados em metaheurísticas de auto-ajuste.

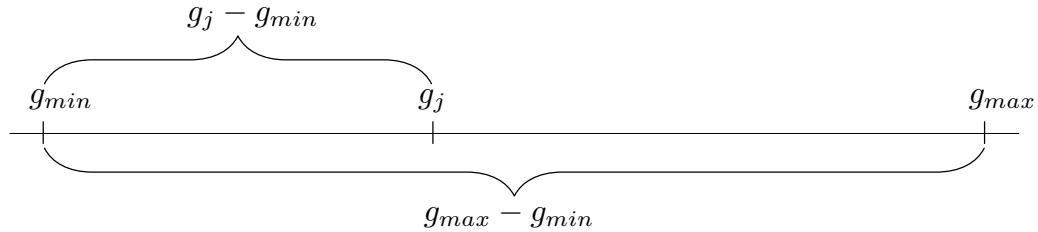
Como já se disse, na fase de construção da GRASP padrão, uma lista restrita de candidatos (LRC) é usada para conter os elementos que melhor se ajustam para serem adicionados à solução, que é construída a partir da LRC vazia, de acordo com uma avaliação da função gulosa. O objetivo desta fase é um equilíbrio entre qualidade e variedade de soluções.

Na fase de melhoria (busca local) da GRASP, o algoritmo compara o custo de alocação de cada cliente ao seu *cluster* atual com o custo de uma hipotética alocação aos outros *clusters*. Os clientes mudariam de *cluster* se essa comparação demonstrasse um ganho. Ao se mover um cliente de um *cluster* para outro ele deixa de incorrer em um custo para o *cluster* antigo e passa a incorrer em um custo, relativo ao novo *cluster*. Para ocorrer uma mudança, a diferença entre o que se ganha e o que se perde com a troca deve diminuir o custo de alocação dos clientes em questão. Outras operações que podem ser estabelecidas nessa fase de melhoria consistem em possível eliminação/fechamento de algumas medianas/*clusters*, possível criação de novas medianas/*clusters*, respeitando a restrição  $\sum_{i \in I} y_i \leq p$ , no caso de  $p$ -medianas, e assim por diante.

García (2001) considera  $g$  como a função gulosa e a LRC é definida para ter um número fixo de elementos ou — uma opção mais flexível — conter todos os elementos dentro de uma determinada distância do principal candidato como uma função de  $g$ . Além disso, se neste segundo caso a proximidade (medida por  $g$ ) de um candidato para o elemento superior é avaliada de acordo com a faixa de valores obtidos por todos os elementos (ao contrário em relação ao valor do candidato principal), o valor limite pode ser expresso como  $\alpha \times (g_{max} - g_{min})$ , onde  $\alpha \in [0, 1]$ . Por isso, para um dado valor de  $\alpha$ ,  $0 \leq \alpha \leq 1$ , conforme ilustra a Figura 3.3, a LRC pode ser definida

como:

$$\text{LRC} = \{j : (g_j - g_{\min}) / (g_{\max} - g_{\min}) \leq \alpha\}$$



**Figura 3.3:** Elementos de construção da Lista Restrita de Candidatos.

É possível alguma variação na maneira de se construir um esquema reativo para a GRASP. Em Boudia et al. (2007),  $\kappa$  é o número fixo de valores permitidos para  $\alpha$  e  $L = \{\beta_1, \beta_2, \beta_3, \dots, \beta_\kappa\}$  a lista desses valores. Ambos são determinados durante alguns testes preliminares. O valor de  $\alpha$  é selecionado aleatoriamente em  $L$  antes da construção de uma solução viável e a probabilidade de escolher diferentes valores  $\beta_t$ ,  $t = 1, \dots, \kappa$ , é progressivamente modificada em favor do tamanho da LRC.

### 3.4 Novo procedimento reativo para a GRASP — a metáfora da peneira

O procedimento reativo desenvolvido neste trabalho para a GRASP é conduzido provocando uma variação do  $\alpha$ ,  $0 \leq \alpha \leq 1$ , durante o processo iterativo, na fase de construção do método. Além do  $\alpha$ , um outro parâmetro,  $\gamma$ , aparecerá nas operações do algoritmo e a explicação do papel deste novo parâmetro será dada posteriormente, a exemplo do  $\alpha$ , através de uma metáfora, no contexto apropriado.

#### 3.4.1 Variando os orifícios da tela da peneira

Uma das possibilidades da GRASP reativa consiste em começar com o  $\alpha$  bem pequeno visando com isso escolher/alocar para cada mediana os nós (clientes) mais próximos. Em seguida o  $\alpha$  é aumentado progressivamente, permitindo que clientes mais distantes de cada mediana sejam alocados. O  $\alpha$  funcionaria como uma espécie de controlador dos orifícios da tela da peneira que é iniciada com uma tela bem fina.

No início, o material peneirado é constituído de grãos muito pequenos — ou finos (que representam os clientes mais próximos a serem alocados). A tela da peneira é progressivamente substituída por outra tela mais grossa, ou seja, com espaços maiores entre os fios da tela. O material peneirado vai se tornando progressivamente granulado com grãos maiores (representando os clientes mais distantes a serem alocados por último). Cria-se com isso uma disciplina de alocação: clientes mais próximo são alocados primeiro.

### 3.4.2 Variando o tamanho da peneira

Simultaneamente ao processo de aumentar progressivamente os orifícios da tela da peneira, fazendo o  $\alpha$  crescer no intervalo  $[0, 1]$ , o tamanho da peneira, em si, é também controlado de acordo com  $g = g_{min} + \alpha(g_{max} - g_{min})$ , em que  $g_{min}$  é uma função de duas entidades: um parâmetro  $\gamma$  e  $g_{max}$ , de acordo com

$$\begin{aligned} g_{min} &= g_{min}(\gamma, g_{max}) \\ &= \gamma \cdot g_{max}, \quad 0 \leq \gamma < 1 \end{aligned}$$

Com a alteração iterativa do parâmetro  $\gamma$ , o valor da diferença  $g_{max} - g_{min}$  se altera também de forma iterativa, pois, a cada vez que  $g_{max}$  é calculado, o  $g_{min}$  é atualizado. Cria-se com isso um processo GRASP<sub>2</sub> REATIVA, ou GRASP “duplamente reativa”, em que a reação se dá, simultaneamente, pelos dois procedimentos descritos.

## 3.5 Fase de busca local da GRASP reativa

Segundo Rangel et al. (2000), a fase de Busca Local é necessária pelo fato das soluções iniciais da GRASP não serem necessariamente ótimos locais. O objetivo da fase de Busca Local é, então, melhorar a solução criada na fase construtiva. Essa busca por melhores soluções se dá consultando a estrutura de vizinhança,  $Viz(s)$ , da solução  $s$  construída, através de sucessivas trocas da solução atual, sempre que uma melhor solução é encontrada. Várias são as trocas: pode-se fechar uma planta e alocar seus clientes à outras plantas; pode-se fechar uma planta e abrir uma que estava fechada; mudar um cliente de uma planta para outra; trocar de planta dois clientes

entre si, etc. Mais detalhes sobre essas operações estão em García (2001).

### 3.6 GRASP modificada: inclusão dos parâmetros de reação

Os algoritmos 4 e 5, que são os algoritmos 1 e 2 alterados, incorporam as modificações que tornam a GRASP duplamente reativa. Esta modificação afeta somente a parte inicial do algoritmo e a fase de construção. Na implementação, o intervalo  $[0, 1]$  é discretizado para fornecer valores numéricos aos parâmetros de reação.

---

#### Algoritmo 4 GRASP.

---

**executa:** GRASP ( $f(\square)$ ,  $g(\square)$ ,  $V(\square)$ ,  $MAX_{GRASP}$ ,  $s$ )

- 1  $f^* \leftarrow \infty$
  - 2  $s^* \leftarrow s$
  - 3 **para** ( $iter = 1, \dots, MAX_{GRASP}$ ) **faça**
  - 4   Construção( $g(\square)$ ,  $0 \leq \alpha \leq 1$ ,  $0 \leq \gamma < 1$ ,  $s$ )
  - 5   BuscaLocal( $f(\square)$ ,  $V(\square)$ ,  $s$ )
  - 6   **se** ( $f(s) < f^*$ ) **então**
  - 7      $s^* \leftarrow s$
  - 8      $f^* \leftarrow f(s)$
  - 9   **fim se**
  - 10 **fim para**
  - 11  $s \leftarrow s^*$
  - 12 Retornar  $s$
  - 13 **fim GRASP**
- 

---

#### Algoritmo 5 Construção.

---

**executa:** GRASP-CONSTRUÇÃO ( $g(\square)$ ,  $0 \leq \alpha \leq 1$ ,  $0 \leq \gamma < 1$ ,  $s$ )

- 1  $s^* \leftarrow \emptyset$
  - 2 Inicializar o conjunto  $C$  de candidatos
  - 3 **enquanto**  $C \neq \emptyset$  **faça**
  - 4    $g_{max}(\theta) = \max\{g(\theta) : \theta \in C\}$
  - 5    $g_{min}(\theta) = \gamma \cdot g_{max}(\theta)$ ,  $\theta \in C$
  - 6    $LRC = \{\theta \in C : g(\theta) \leq g_{min}(\theta) + \alpha(g_{max}(\theta) - g_{min}(\theta))\}$
  - 7   Selecionar, aleatoriamente, um elemento  $\theta \in LRC$
  - 8    $s \leftarrow s \cup \{\theta\}$
  - 9   Atualizar o conjunto  $C$  de candidatos
  - 10 **fim enquanto**
  - 11 Retornar  $s$
  - 12 **fim GRASP-CONSTRUÇÃO**
-

Por exemplo,  $0 \leq \alpha \leq 1$  assume os valores do conjunto  $\{0, 0.1, \dots, 0.9, 1\}$ , no decorrer das iterações. Do mesmo modo,  $0 \leq \gamma < 1$  assume os valores de  $\{0, 0.1, \dots, 0.9\}$ . No final, os parâmetros  $\alpha$  e  $\gamma$  tentam manter o melhor valor que encontraram durante o processo. Embora não registradas aqui, outras discretizações — mais e menos finas — foram testadas/experimentadas no decorrer da elaboração do novo procedimento.

### 3.7 Complexidade assintótica da GRASP

Declarado por Donald Knuth em Sedgewick & Flajolet (2013) “Modelos matemáticos têm sido uma inspiração crucial para toda a atividade científica, embora eles sejam somente idealizações aproximadas dos fenômenos do mundo real. Dentro do computador, tais modelos são mais relevantes como nunca o foram, porque os códigos computacionais criam um mundo artificial em que os modelos matemáticos frequentemente se aplicam com precisão.”

Códigos computacionais expressam algoritmos e esses têm sua ordem de complexidade. O tema é objeto desta pequena introdução a seguir como preâmbulo para que se declare a complexidade assintótica da GRASP.

O termo “complexidade” refere-se em geral, aos requisitos de recursos necessários para que um algoritmo possa resolver um problema computacional. (LEAL, 2002).

A avaliação do desempenho de um algoritmo é feita estimando o tempo de execução e a memória utilizada por ele, sendo o tempo de execução o parâmetro mais utilizado. O tempo de execução depende dos dados de entrada, do código gerado, do “hardware” utilizado e da complexidade do algoritmo implementado. A abordagem mais adotada é relacionar o tempo de execução diretamente aos dados de entrada, especificamente, ao seu comprimento ou tamanho, entendido, aqui, como o número de bits necessários à codificação. (NOGUEIRA, 2002).

Segundo Leal (2002), os requisitos de tempo são frequentemente considerados como fatores dominantes para determinar se um particular algoritmo é eficiente o bastante para ser útil na prática.

No mesmo trabalho ainda é afirmado que a *complexidade de tempo* para um algoritmo é uma função definida sobre os tamanhos das instâncias do problema, que considera

o requisito de tempo exigido por entradas de um mesmo tamanho, avaliando a quantidade de tempo necessária para o algoritmo resolver problemas desse tamanho.

Define-se formalmente, então, o tamanho da instância  $I$  de um problema  $\Pi$ , como o comprimento da entrada dessa instância, dado pelo número de símbolos na descrição de  $I$ , obtida de um esquema de codificação do problema  $\Pi$ , (LEAL, 2002).

A determinação da quantidade de recursos que um algoritmo precisa para resolver uma instância qualquer do problema é necessária para a análise da complexidade associada à resolução do problema específico.

Com o objetivo de desenvolver uma teoria tão geral quanto possível, independente do modelo de computação escolhido, e capaz de capturar a complexidade inerente do problema, estuda-se a razão de crescimento do requerimento de recurso mais que a sua própria avaliação. Na prática, sempre que as instâncias não excedem um tamanho fixo, um algoritmo com uma rápida razão de crescimento poderia ser preferível a outro com menor razão de crescimento. Entretanto, se o interesse é estudar como a complexidade cresce quando o tamanho das instâncias torna-se maior, pode-se formalizar tal comportamento através da notação de *ordem assintótica*, (LEAL, 2002).

**Definição 3.7.1** *Seja  $g : \mathbb{N} \rightarrow \mathbb{N}$  uma função. A classe de ordem assintótica de  $g$ , consiste de todas as funções  $f : \mathbb{N} \rightarrow \mathbb{N}$  para as quais existem constantes  $k$  e  $n_0$  tal que  $f(n) \leq k \cdot g(n)$ , para qualquer  $n \in \mathbb{N}$ ,  $n > n_0$ .*

Assim, para expressar que “ $f$  é da ordem assintótica de  $g$ ,” escreve-se, simplesmente, que  $f \in \mathcal{O}(g)$ . Nesse caso, diz-se ainda que “ $g$  domina  $f$  assintoticamente”. Portanto, de acordo com a Definição 3.7.1,  $\mathcal{O}(g)$  representa o conjunto de todas as funções que são, assintoticamente, dominadas por  $g$ .

A título de ilustração, considera-se a função  $f : \mathbb{N} \rightarrow \mathbb{N}$ , dada por  $f(n) = \sum_{i=1}^n i^2$ , e  $g : \mathbb{N} \rightarrow \mathbb{N}$ , tal que  $g(n) = n^3$ . Tem-se:

$$\begin{aligned} f(n) &= 1^2 + 2^2 + \cdots + n^2 = \frac{1}{3}n(n + \frac{1}{2})(n + 1) = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \\ &\leq |(\frac{1}{3} + \frac{1}{2n} + \frac{1}{6n^2})n^3| \leq |(\frac{1}{3} + \frac{1}{2} + \frac{1}{6})| n^3, \text{ se } n > 1. \end{aligned}$$

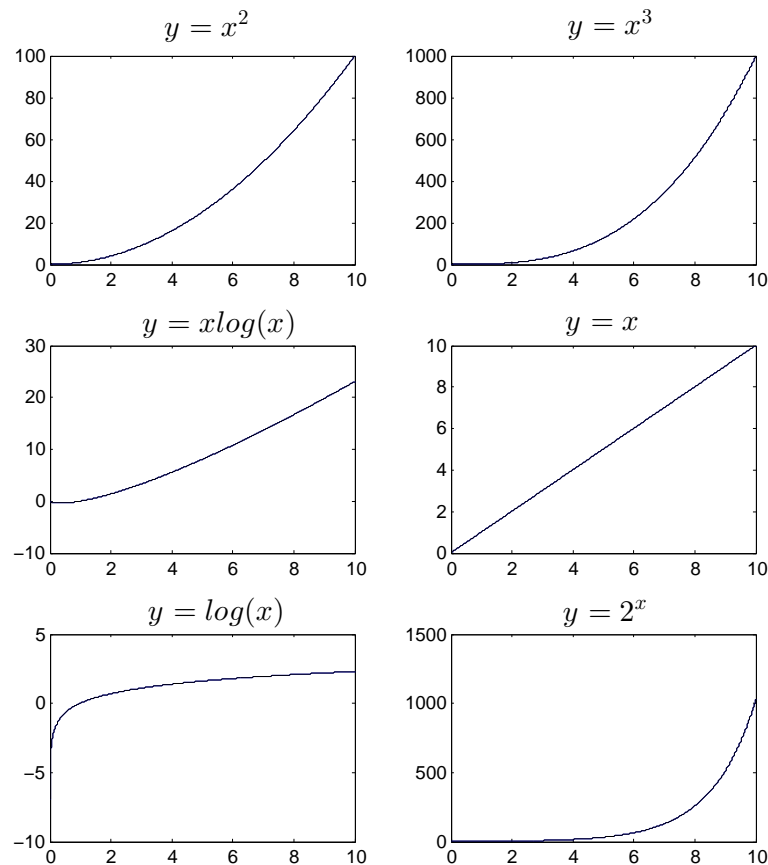
Fazendo  $k = \frac{1}{3} + \frac{1}{2} + \frac{1}{6}$  e  $n_0 = 1$ , tem-se que,  $n > n_0$  implica  $f(n) \leq k \cdot g(n)$  e, portanto,  $\sum_{i=1}^n i^2 \in \mathcal{O}(n^3)$ , ou  $f \in \mathcal{O}(g)$ .



Se  $f(n)$  e  $g(n)$  representam as complexidades de tempo para dois algoritmos, então, para problemas de tamanho mínimo igual a  $n_0$ , o tempo de execução do algoritmo de complexidade  $f(n)$  nunca excederá  $k$  vezes o tempo de execução do algoritmo de complexidade  $g(n)$ , para a mesma instância do problema.

**Definição 3.7.2** A complexidade assintótica de tempo  $T(n)$  de um algoritmo, é de ordem  $\mathcal{O}(g(n))$ , se existem constantes positivas  $k$  e  $n_0$ , tais que  $T(n) \leq k \cdot g(n)$ , para qualquer  $n > n_0$ .

As complexidades assintóticas mais comuns podem ser visualizadas no gráfico abaixo.



**Figura 3.4:** Crescimento assintótico de funções de complexidade de tempo.

Com exceção de  $\mathcal{O}(2^n)$ , todas as ordens assintóticas da Figura 3.4 são limitadas por um polinômio. Observa-se ainda que, para  $n$  suficientemente grande, tem-se

$$\mathcal{O}(1) < \mathcal{O}(\log n) < \mathcal{O}(n) < \mathcal{O}(n \log n) < \mathcal{O}(n^2) < \mathcal{O}(n^3) < \mathcal{O}(2^n)$$

Segundo Nogueira (2002), em alguns casos, um determinado algoritmo se constitui

de partes ou rotinas independentes. Sendo assim, a avaliação de sua complexidade assintótica ( $T(n)$ ) passa pela chamada Regra das Somas.

**Regra das Somas:** Se determinado algoritmo  $A$  é constituído de partes independentes  $A_1$  e  $A_2$ , por exemplo, com complexidades  $T_1(n)$  e  $T_2(n)$ , respectivamente, de ordem  $\mathcal{O}(f(n))$  e  $\mathcal{O}(g(n))$ , então sua complexidade  $T(n) = T_1(n) + T_2(n)$  será de ordem  $\mathcal{O}(\text{máximo}\{f(n), g(n)\})$ .

A pequena introdução feita nesta seção sobre complexidade assintótica, teve a finalidade de registrar aqui a terminologia utilizada, a seguir, na declaração da complexidade da GRASP.

Segundo (NOGUEIRA, 2002), a complexidade assintótica do algoritmo é determinada, preponderantemente, pela verificação das complexidades dos procedimentos *construir solução* e *melhorar solução*, pois é nesses procedimentos que ocorre um maior número de instruções.

Existem três parâmetros na GRASP que têm influência na sua complexidade assintótica. São eles:  $MaxIter$ ,  $|C|$  e  $|V|$ .  $MaxIter$  é o número máximo de iterações da GRASP. Para cada uma dessas iterações, a fase de construção executa o laço de instruções  $|C|$  vezes, em que  $C$  é a lista restrita de de candidatos. Portanto, o laço é executado um número de vezes igual a  $MaxIter \times |C|$ .  $V$  é a vizinhança da solução a ser aprimorada na fase de melhoria. Portanto, com o mesmo raciocínio anterior, o laço é executado um número de vezes igual a  $MaxIter \times |V|$ . Se o tamanho do problema é  $n$ , tem-se que  $|C| \leq n$  e  $|V| \leq n$ . Assim,  $\max\{MaxIter \times |C|, MaxIter \times |V|\} \leq MaxIter \times n$  e, portanto, como  $MaxIter$  é uma constante, tem-se que a complexidade assintótica da GRASP é  $\mathcal{O}(n)$ . Portanto, linear.

Na GRASP duplamente reativa existem mais dois parâmetros:  $0 \leq \alpha \leq 1$ ,  $0 \leq \gamma < 1$ . Para cada  $\alpha$ , todos os  $\gamma$  são usados. Dividindo-se os intervalos de  $\alpha$  e  $\gamma$  em  $m$  subintervalos, tem-se que o laço para todos os  $\alpha$  e todos os  $\gamma$  é executado  $m^2$  vezes. Portanto, a complexidade da GRASP duplamente reativa, como um todo, torna-se  $\mathcal{O}(m^2n)$ .

Em (NOGUEIRA, 2002), é mostrado que “a complexidade assintótica da GRASP para o problema quadrático da mochila é de ordem  $\mathcal{O}(n^2)$ ”.

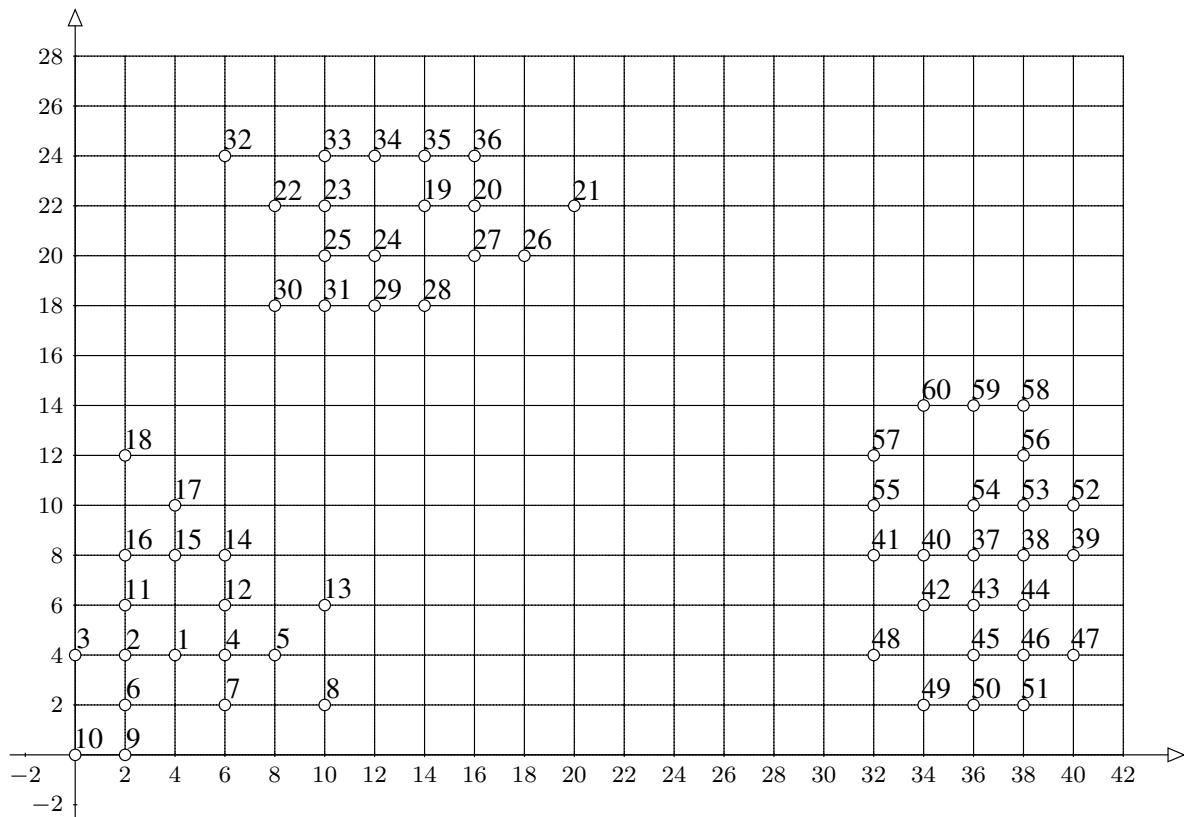
# CAPÍTULO IV

## EXPERIMENTO COMPUTACIONAL – GRASP

### DUPLAMENTE REATIVA

O experimento computacional foi desenvolvido no ambiente C++ do Visual Studio 2010, usando computador baseado em processador de múltiplos núcleos.

Para testar a qualidade das soluções fornecidas pelo algoritmo, foi criado um primeiro problema com 60 pontos distribuídos em um plano, conforme mostra a Figura 4.1. Foram feitas 169 ligações entre esses pontos. Apesar da matriz de distâncias ser simétrica, o grafo não é completo. Foi utilizado, então, o algoritmo Floyd–Warshall (ver Capítulo VI) para encontrar o caminho de menor distância entre todos os pontos da rede, completando assim, a matriz de distâncias.



**Figura 4.1:** Problema de Localização de 60 pontos com 3 medianas.

Como se percebe, os *clusters* já estavam visivelmente formados. O primeiro *cluster*

vai do ponto 1 ao ponto 18. O segundo *cluster* do ponto 19 ao ponto 36. Já o terceiro, do ponto 37 ao 60. O que se pretendia com o teste, era verificar se realmente os *clusters* formados pelo algoritmo seriam os mesmos da Figura 4.1. Os resultados comprovam a eficiência do algoritmo. Veja a Figura 4.2 com o resultado numérico e a Figura 4.3 com as medianas indicadas por pontos escuros.

Dia: 24/07/2012, Hora: 11:28:45

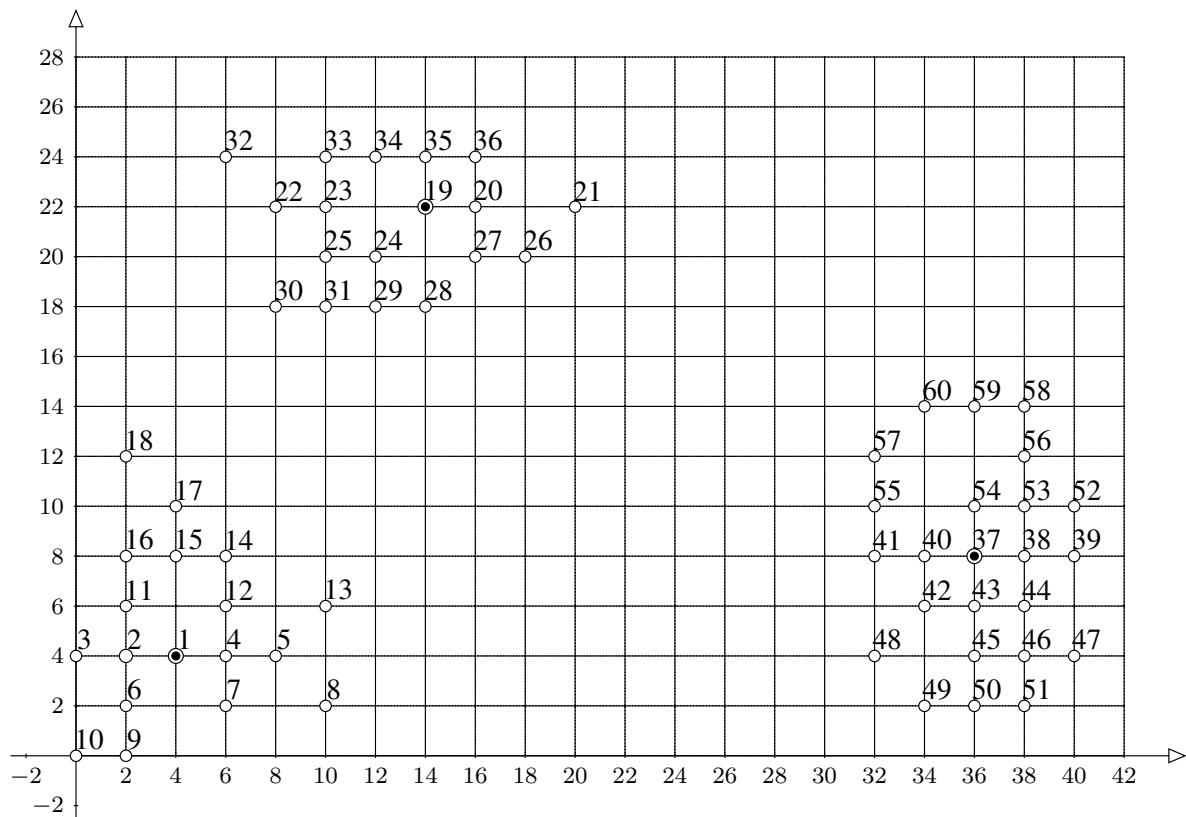
A MELHOR SOLUÇÃO ENCONTRADA EM 50000 ITERAÇÕES, (com 16 trocas para uma melhor solução), É:

Cluster da mediana 1:	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	18											
Total de cidades neste cluster:	17, distância (custo) total neste cluster: 127													
Cluster da mediana 37:	38	39	40	41	42	43	44	45	46	47	48	49	50	51
	52	53	54	55	56	57	58	59	60					
Total de cidades neste cluster:	23, distância (custo) total neste cluster: 162													
Cluster da mediana 19:	20	21	22	23	24	25	26	27	28	29	30	31	32	33
	34	35	36											
Total de cidades neste cluster:	17, distância (custo) total neste cluster: 111													

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 400

Dia: 24/07/2012, Hora: 11:40:06

**Figura 4.2:** Resultado do problema de 60 pontos com 3 medianas.



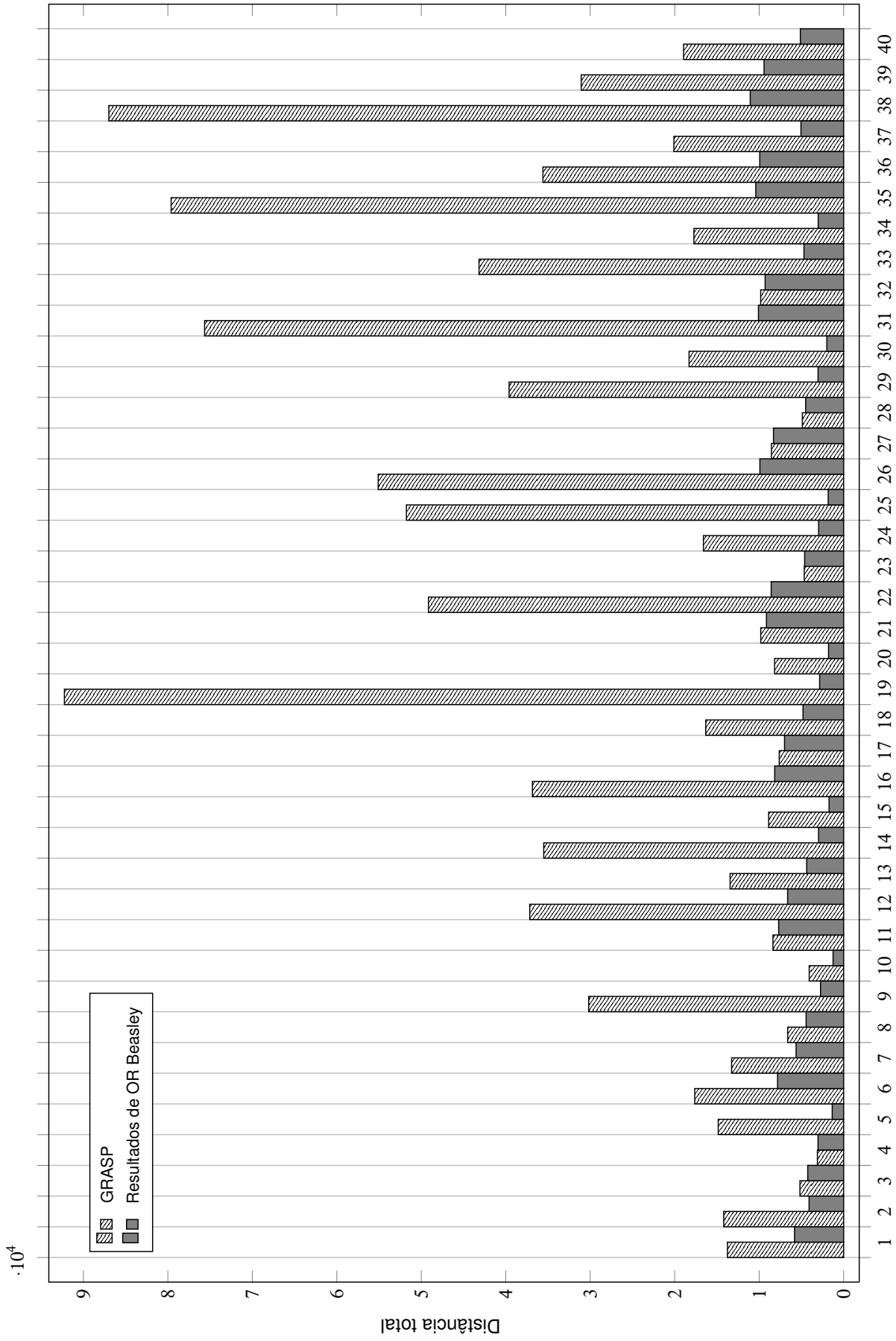
**Figura 4.3:** Localização das 3 medianas — 1, 19 e 37 — entre 60 pontos.

Foi utilizada também a biblioteca de problemas teste de *OR-Beasley*, Beasley (1990).

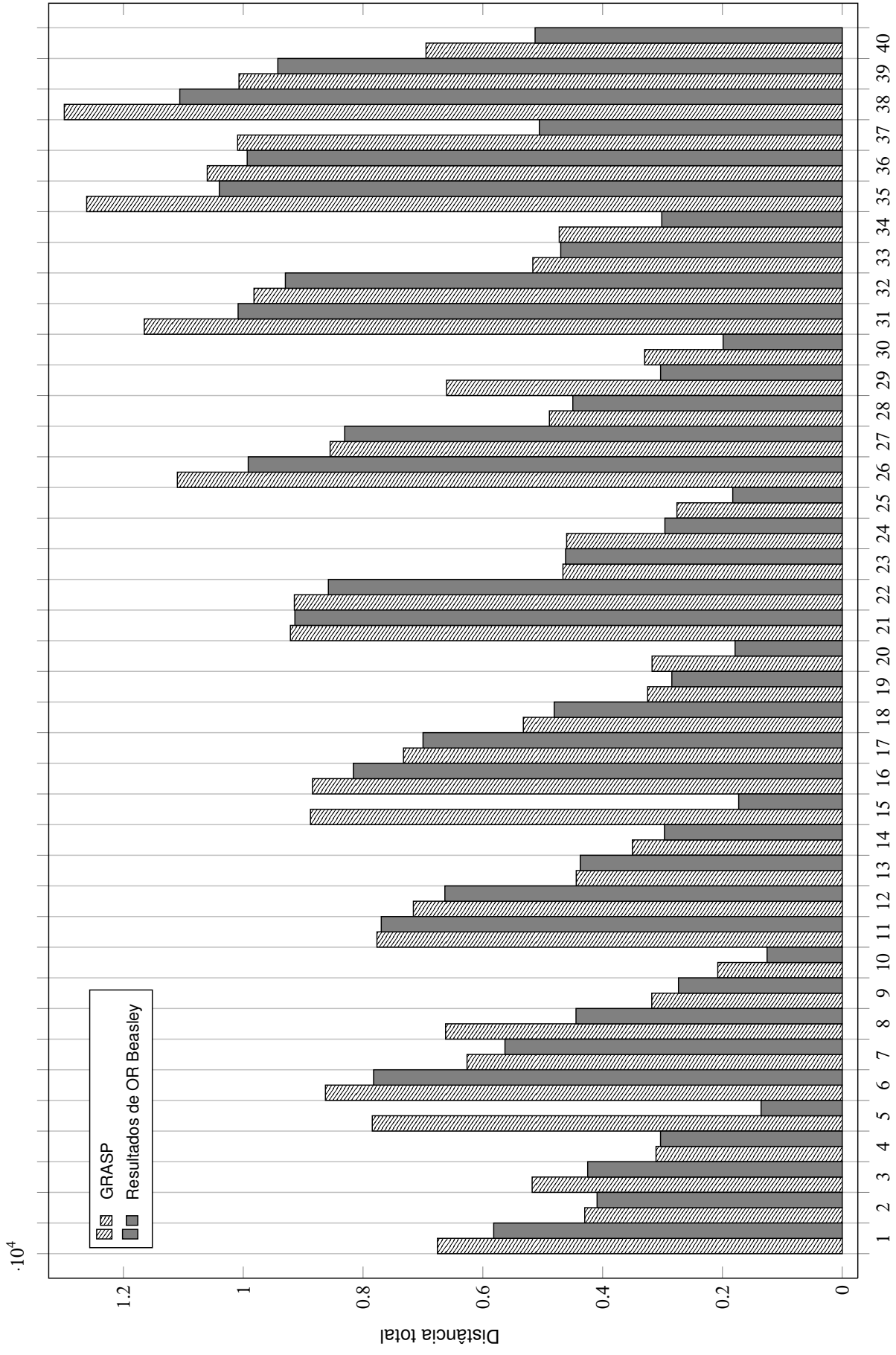
Esta biblioteca é composta por 40 instâncias diferentes de problemas não-capacitados, onde o número  $n$  de vértices varia de 100 a 900 e o número  $p$  de medianas varia entre 5 e 90. Todas estas instâncias já foram resolvidas até a sua otimalidade. Mais uma vez, as matrizes de distância não eram completas e o algoritmo de Floyd–Warshall foi utilizado para encontrar o caminho de menor distância entre todos os pontos da rede, completando assim, as matrizes de distâncias.

A parcela de custos fixos da função objetivo não foi utilizada nos testes — os dados da biblioteca OR Beasley não fornecem essa informação. As Figuras 4.4 e 4.5, em escalas diferentes para preservar a razão de aspecto, mostram os dois casos extremos dos experimentos. A Figura 4.4 mostra os resultados do experimento que executa uma única iteração com cada uma das 40 instâncias de OR Beasley. A Figura 4.5 mostra os resultados do experimento com a execução de 10000 iterações da GRASP sobre cada uma das mesmas instâncias. Outras avaliações foram feitas com número de iterações abaixo e acima de 10000 iterações.

Nas Figuras 4.4 e 4.5, a barra mais à esquerda nos gráficos — relativa a cada instância — mostra o desempenho da GRASP. A Tabela 4.1 mostra os resultados numéricos e obtidos dos experimentos com uma única iteração e com 10000 iterações.



**Figura 4.4:** Experimento: comparação do valor da distância total com Beasley – uma única iteração GRASP.



**Figura 4.5:** Experimento: comparação do valor da distância total com Beasley – 10000 iterações GRASP.

**Tabela 4.1:** Resultados do experimento para avaliar a GRASP, comparando com instâncias da biblioteca OR Beasley.

Instância	Número de Cidades	Número Máximo de Medianas	Trocas Para Uma Melhor Solução, Em 10000 Iter GRASP	Distância Total Para Uma Única Iter GRASP	Distância Total Para 10000 Iter GRASP	Tempo Médio Por Iteração GRASP (seg)	Solução Ótima OR Beasley
1	100	10	14	13758	6758	0.03999	5819
2	100	10	13	14199	4299	0.03999	4093
3	100	20	10	5177	5177	0.07000	4250
4	100	33	16	3109	3109	0.11000	3034
5	100	5	14	14846	7846	0.02999	1355
6	200	10	4	17628	8628	0.09000	7824
7	200	20	12	13264	6264	0.16000	5631
8	200	40	15	6622	6622	0.28000	4445
9	200	5	4	30180	3180	0.05999	2734
10	200	67	15	4078	2078	0.44990	1255
11	300	100	13	8368	7768	1.01000	7696
12	300	10	4	37160	7160	0.16000	6634
13	300	30	14	13441	4441	0.34990	4374
14	300	5	4	35504	3504	0.11000	2968
15	300	60	3	8878	8878	0.63000	1729
16	400	10	15	36844	8844	0.25000	8162
17	400	133	14	7625	7325	1.78000	6999
18	400	40	11	16324	5324	0.62000	4809
19	400	5	15	92249	3249	0.18000	2845
20	400	80	12	8175	3175	1.12000	1789
21	500	100	15	9813	9213	1.76000	9138
22	500	10	15	49147	9147	0.36000	8579
23	500	167	13	4663	4663	2.80900	4619
24	500	50	11	16601	4601	0.99000	2961
25	500	5	7	51760	2760	0.28000	1828
26	600	10	11	55100	11100	0.50000	9917
27	600	120	11	8550	8550	2.56900	8307
28	600	200	14	4889	4889	3.07000	4498
29	600	5	8	39606	6606	0.38000	3033
30	600	60	10	18299	3299	1.45000	1989
31	700	10	15	75653	11653	0.64990	10086
32	700	140	12	9821	9821	3.51000	9297
33	700	5	12	43166	5166	0.52990	4700
34	700	70	12	17725	4725	1.98000	3013
35	800	10	14	79614	12614	0.83990	10400
36	800	5	7	35601	10601	0.69990	9934
37	800	80	15	20094	10094	2.57900	5057
38	900	10	15	86988	12988	1.07000	11060
39	900	5	6	31071	10071	0.89990	9423
40	900	90	13	18949	6949	2.32900	5128



# CAPÍTULO V

## *PATH RELINKING*

### 5.1 Introdução

De acordo com Glover et al. (2003), publicado em Glover & Kochenberger (2003), “o *path relinking*, ou religamento de caminhos, foi sugerido como enfoque para integrar estratégias de intensificação e diversificação num esquema de busca. O enfoque pode ser visto como uma instância – altamente focada – de uma estratégia que procura incorporar atributos de soluções de alta qualidade (soluções elite), procurando contemplar ou favorecer esses atributos nos movimentos selecionados.”

Um procedimento equivalente seria construir caminhos usando uma combinação de várias soluções até que todas fossem examinadas. Um tal enfoque se enquadra também no que é denominado ou referido em geral como *path relinking*.

O *path relinking* também se associa à ideia de reinterpretar as combinações lineares de pontos no Espaço Euclidiano como caminhos entre e além das soluções no âmbito da vizinhança. O caminho entre duas soluções no espaço de busca pode produzir soluções que compartilham atributos comuns a soluções previamente existentes. A sequência de soluções vizinhas no espaço de decisão é gerada a partir da solução inicial, indo na direção da solução alvo. A melhor solução encontrada na sequência é retornada.

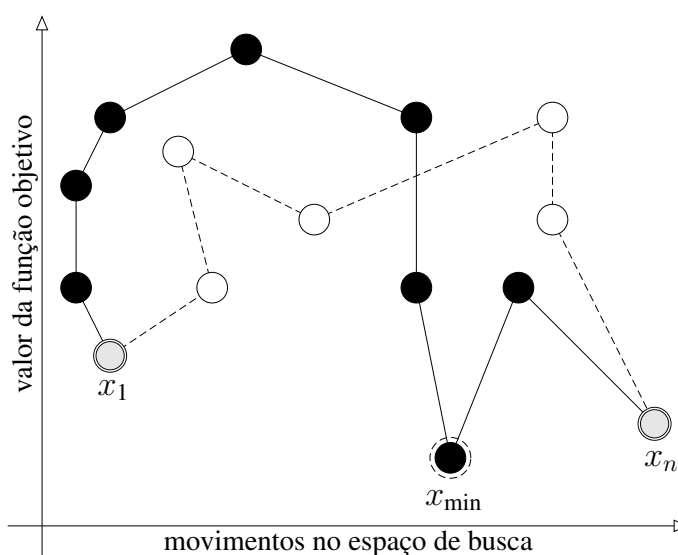
Intensificação e diversificação em qualquer método de busca devem ser tratadas com a ideia de se manter um equilíbrio entre elas durante o processo. A proposta do *path relinking* foi feita como um instrumento capaz de garantir, ou pelo menos permitir a integração equilibrada dessas estratégias de intensificação e diversificação.

Como ato de esclarecimento, o *path relinking* é um enfoque que pode ser usado na fase terminal de uma metaheurística “quase como uma pós-otimização, ou pós-melhoria no caso da metaheurística GRASP” para gerar novas soluções por via do exame de trajetórias que ligam soluções de alta qualidade – novas ou já existentes –

a partir de um ponto inicial que é uma dessas soluções, chamada de solução inicial. Desse ponto inicial em diante o procedimento explora o espaço de busca gerando um caminho que leva a outras soluções, chamadas de soluções guia. A seleção dos movimentos nesta fase é feita explorando os atributos de soluções já conhecidas, em particular, aqueles das soluções guia. Enfim, o *path relinking* pode ser visto como uma estratégia cuidadosamente focada que atua no sentido de incorporar atributos de soluções de alta qualidade, tentando privilegiar esses atributos nos movimentos que são progressivamente selecionados.

A motivação para o termo *path relinking* surge da capacidade que o enfoque tem para gerar um novo caminho entre soluções previamente ligadas por uma série de movimentos executados durante o processo de busca. Essa motivação se sustenta também pela virtude do método em gerar um caminho entre soluções previamente ligadas a outras soluções mas não necessariamente a cada outra solução.

A Figura 5.1 mostra dois caminhos possíveis – isto é, duas sequências de movimentos – ligando a solução  $x_1 =$  “início” à solução  $x_n =$  “alvo”, para ilustrar o religamento (relinking) entre soluções. O caminho original é representado por segmentos de linha



**Figura 5.1:** Dois caminhos: o original e o *path relinking* – em linhas sólidas.

interrompida e decorre da “operação normal” de um algoritmo que produz seu resultado na forma de uma série de movimentos que conduzem do ponto  $x_1$  ao ponto  $x_n$ , isto é, gera a sequência (“início” =  $x_1$ ,  $x_2$ , ...,  $x_n$  = “alvo”), enquanto que o caminho de religamento é representado pelos segmentos de linha sólida.

Os caminhos são gerados com base na seleção de movimentos e adotam uma trajetória em que os passos intermediários são determinados por alguma função de avaliação. Por exemplo, um enfoque comum usado é selecionar um movimento que otimiza o valor da função objetivo no sentido de soluções locais. Entretanto, durante o *path relinking* o objetivo maior constitui-se da incorporação de atributos das soluções guia, ao mesmo tempo em que registra os valores da função objetivo.

A Figura 5.1 mostra uma solução (o nó escuro associado a  $x_{min}$ ) alcançada pelo caminho de linha sólida. Além disso, o caminho religado pode encontrar soluções possivelmente melhores do que a solução guia inicial — ou provê “entradas férteis” para alcançar outras soluções, de alguma forma, melhores. Em geral, as soluções guia podem ser constituídas a partir de uma seleção de soluções anteriores chamadas de “soluções elite”.

## 5.2 Integração do *path relinking* com a GRASP — o que registra a literatura

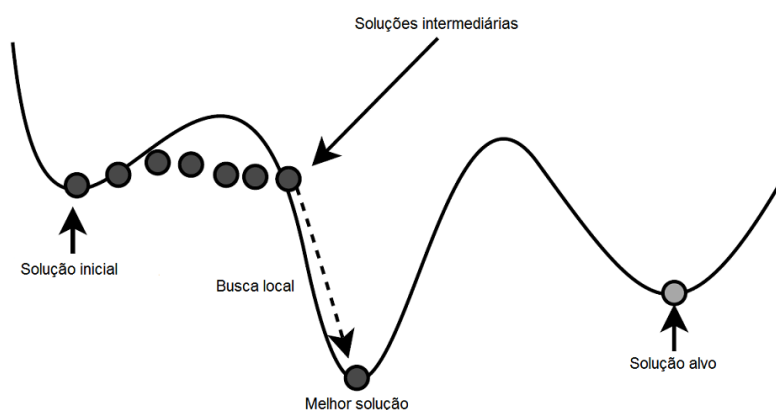
Laguna & Martí (1999) propuseram um método que combina GRASP e *Path Relinking* para desenvolver um procedimento capaz de competir, em velocidade, com heurísticas simples e em qualidade com metaheurísticas complexas. O procedimento híbrido proposto utiliza a GRASP como um método *multistart* que armazena um pequeno conjunto de soluções elite de alta qualidade para serem usadas com o propósito de guia. Vem sendo progressivamente reconhecido que o aprimoramento produzido pelo enfoque do *path relinking* sugere o mérito em potencial de se combinar estratégias de guia com métodos *multistart* como a GRASP, possivelmente pela oportunidade de injetar diversidade no processo e pela possibilidade de intensificar o procedimento de busca.

Em Talbi (2009) a estratégia do *path relinking* é descrita como um procedimento que pode ser aplicado a um conjunto de soluções elite calculadas pelas operações de uma metaheurística. Neste caso o *path relinking* pode ser visto como uma tarefa de intensificação motivada pelos atributos de um dado elenco de soluções elite. Soluções intermediárias, criadas durante o processo, tanto podem ser melhores do que a solução inicial ou do que a solução alvo, quanto podem servir como bons pon-

tos de início — ou de entrada — para processos adicionais de busca.

Considera-se a seguinte metaheurística híbrida que integra o conceito de *path relinking* com a GRASP. A GRASP mantém um conjunto de soluções elite encontrado durante as diferentes iterações da metaheurística. A cada iteração da GRASP, o *path relinking* é aplicado, a partir da nova solução gerada pela GRASP, a uma dada solução dentre as soluções elite. A solução alvo pode ser escolhida aleatoriamente a partir do conjunto elite. A Figura 5.2 ilustra um tal procedimento. Considerando na Figura 5.2 a solução inicial, a solução alvo e a melhor solução como representantes das três “bacias de atração”, observa-se a aplicação de um procedimento de busca local a uma solução intermediária no *path relinking*. Num cenário representado pela estrutura de um “grande vale”, uma solução intermediária entre dois ótimos locais — inicial e alvo — pode ser guiada na direção de uma terceira “bacia de atração” que pode conter uma solução ótima local melhor. Como experiência advinda da literatura, pode-se afirmar que os papéis da solução inicial e da solução guia são intercambiáveis. Cada solução pode ser induzida a possibilitar movimentos simultâneos na direção da outra como forma de gerar combinações.

A seguir será apresentado o enfoque do *path relinking* gerado e usado neste trabalho.



**Figura 5.2:** Um esquema de *path relinking* sugerido em Talbi (2009): observa-se que a aplicação de um procedimento de busca local a uma solução intermediária no *path relinking* pode conter uma solução ótima local melhor.

### 5.3 O enfoque de *path relinking* sugerido e adotado neste trabalho

O procedimento de *path relinking* construído para este trabalho será descrito a seguir com destaque para novidade que ele introduz em função das características da GRASP e do problema de localização em que ele foi aplicado. Em sequência, o procedimento foi testado com o mesmo conjunto de dados que vem sendo utilizado até aqui e, finalmente, uma aplicação foi feita através do ato de simular a formação de *clusters* com as cidades brasileiras médias e grandes, produzindo um ensaio sobre como utilizar esses recursos na solução de problemas de localização de  $p$ -medianas.

A primeira providência foi criar um conjunto de soluções elite através de um conceito que aqui foi denominado de “vocação mediana”. Este conceito, que será detalhado no decorrer desta descrição, fundamenta-se em um índice — “índice<sub>vm</sub>” — que estabelece o quanto cada cidade, suposta candidata a mediana, está próxima das demais cidades do grupo. Antes de construir esse índice, o instrumento previamente utilizado foi soma das distâncias de cada cidade às demais. Posteriormente, as cidades, supostas candidatas a mediana, foram ordenadas, em ordem crescente, de acordo com a distância total de cada uma delas às demais. Vários problemas menores foram construídos e resolvidos, considerando somente uma porcentagem do total de candidatas a medianas. Diga-se,  $\delta\% \times \text{Total de candidatas a mediana}$ , para o primeiro problema,  $2\delta\% \times \text{Total de candidatas a mediana}$  para o segundo problema, e assim por diante, até que  $k \times \delta\%$  se tornasse igual a 100%, isto é, todos os supostos candidatos a medianas. Cuidado foi tomado para que em nenhum dos problemas menores a quantidade de medianas candidatas ficasse inferior ao total máximo de medianas que se desejava localizar. A solução de todos esses problemas menores produziu um aglomerado de medianas, do qual participaram todas as medianas escolhidas como solução dos problemas menores. Esse aglomerado de medianas foi aceito, empiricamente, como um conjunto de medianas associado às soluções elite do problema de localização. Tal aceitação empírica da condição “elite” das soluções assim obtidas, ficou aguardando que experimentos comprovassem a presença de tal atributo. O passo final foi aplicar a esse conjunto de soluções elite um procedimento de *path relinking* que explorou a intensificação e a diversificação do processo de busca. Ou seja, um conjunto de novas soluções foi gerado nas várias iterações da GRASP para produzir a solução final

que se desejava melhor do que as soluções encontradas antes da intensificação do *path relinking*.

O Algoritmo 6 mostra o pseudocódigo do procedimento de *path relinking* aqui adotado.

---

**Algoritmo 6** GRASP COM *PATH RELINKING*.

---

**executa:** GRASP COM *PATH RELINKING* ( $s_k$ ,  $\text{MAXk}_{\text{PATHRELINKING}}$ ,  $X_1, \dots, X_{\text{MAXk}_{\text{PATHRELINKING}}}$ ,  $Y$ ,  $\mathcal{S}_{\text{PATHRELINKING}}$ )

- 1 //  $X_k$ , para  $k = 1, \dots, \text{MAXk}_{\text{PATHRELINKING}}$ , contém dados originais  
// do problema de localização como candidatos a medianas
- 2 //  $Y$  acumula as soluções elite como dados para o *path relinking*
- 3  $Y \leftarrow \emptyset$
- 4 **para** ( $k = 1, \dots, \text{MAXk}_{\text{PATHRELINKING}}$ ) **faça**
- 5 //  $X_k$  recebe, a cada iteração, uma parte (porcentagem) cumulativa  
// de vértices como candidatos a medianas, até o total de vértices
- 6  $X_k \leftarrow \left( \frac{k}{\text{MAXk}_{\text{PATHRELINKING}}} \times \text{DadosORIGINAIS} \right)$
- 7  $s \leftarrow s_k$
- 8 GRASP( $X_k$ )
- 9  $s_k \leftarrow s$
- 10  $Y \leftarrow Y \cup s_k$
- 11 **fim para**
- 12 GRASP( $Y$ )
- 13  $\mathcal{S}_{\text{PATHRELINKING}} \leftarrow s$
- 14 Retornar  $\mathcal{S}_{\text{PATHRELINKING}}$
- 15 **fim** GRASP( $Y$ )

---

A forma mais efetiva de mostrar o funcionamento deste modelo de *path relinking* é através de uma aplicação do problema generalizado de  $p$ -medianas, o que passa a ser feito através da formação de *clusters* com as cidades brasileiras, médias e grandes, que aparecem na matriz de distâncias do DNIT<sup>1</sup>.

A Tabela 5.4 mostra a relação, em ordem alfabética, das 112 cidades brasileiras que participaram do experimento. A Tabela 5.5 mostra a distância total de cada cidade às demais relacionadas, assim como o índice de vocação mediana associado a cada uma delas. A Tabela 5.6 mostra as cidades ordenadas pelo índice<sub>vm</sub>.

Pela Tabela 5.6, vê-se que a cidade brasileira mais próxima das demais — cidades médias e grandes — é Ribeirão Preto-SP, com índice<sub>vm</sub> igual a 1,328. A mais distante

---

<sup>1</sup>Departamento Nacional de Infraestrutura de Transportes. Autarquia responsável pelos sistemas de transporte: rodoviário, aquaviário e ferroviário do Brasil. Divisão de Planejamento, Coordenação e Gerência de Sistemas Rodoviários. <http://www.dnit.gov.br/>, acessado em 06/03/2013.

das demais é Boa Vista-RR, com  $\text{índice}_{vm}$  igual a 5,017.

O  $\text{índice}_{vm}$  foi construído de acordo com

$$\text{índice}_{vm}(\text{“cidade”}) = \frac{\sum(\text{distâncias da “cidade” às outras cidades})}{\text{Número total das outras cidades} \times 1000}$$

Por exemplo,

$$\text{índice}_{vm}(\text{Manaus}) = \frac{\sum(\text{distâncias de Manaus às outras cidades})}{\text{Número total das outras cidades} \times 1000} = 4.241$$

$$\text{índice}_{vm}(\text{São Paulo}) = \frac{\sum(\text{distâncias de São Paulo às outras cidades})}{\text{Número total das outras cidades} \times 1000} = 1.365$$

No experimento projetado o  $\delta$  foi considerado igual a 10. Isto significa que o primeiro problema menor foi resolvido com 10% das cidades com menor distância total às demais 111 cidades, como candidatas a mediana. O segundo problema menor foi resolvido com 20% dessas cidades, e assim por diante, até que 100% das cidades tornaram-se candidatas a mediana na solução do problema de localização em foco. Portanto, foram resolvidas 9 instâncias menores<sup>2</sup> mais a instância original, isto é, 10 instâncias. O Apêndice A mostra 10 tabelas — Tabela A.1 a Tabela A.10 — com o resultado dessas instâncias. A Tabela A.11 mostra um conjunto de 26 cidades constituindo o agregado elite de candidatas a mediana, definido durante a solução das 10 instâncias, conforme já descrito. Finalmente, a Tabela A.12 mostra o resultado final usando o procedimento de *path relinking*. A Tabela 5.1 mostra um resumo das soluções de cada uma das 10 instâncias. A melhor solução sem o *path relinking* ocorreu na Tabela A.2, com distância total igual 68551, entre as medianas e as cidades que a elas se ligam. O resultado com *path relinking* — Tabela A.12 — teve distância total igual 62393. A diferença entre os dois resultados se torna expressiva se for considerado que os dois passaram por um processo de “otimização no sentido das metaheurísticas”. Em outras palavras, o *path relinking* trabalha sobre um resultado já otimizado, ou o tem como referência.

<sup>2</sup>O termo “menor” aqui significa que somente uma parte dos vértices/cidades foi aceita como conjunto de candidatas a mediana.

**Tabela 5.1:** Resumo dos resultados com 112 cidades, 5 medianas, 2000 iterações.

Instância	Trocas para melhor solução	Distância total	Tempo médio por iteração (seg)
1	8	108493	0,03250000
2	9	68551	0,06450000
3	3	68613	0,09950000
4	8	68899	0,13349999
5	7	70795	0,16850001
6	4	71048	0,20299999
7	5	71546	0,24050000
8	2	73940	0,28650001
9	8	72485	0,33600000
10	4	75445	0,38350001

Experimento semelhante, agora com 20000 iterações para resolver os problemas menores, tem o resultado resumido na Tabela 5.2. Neste caso, a melhor solução sem o *path relinking* é a distância total igual a 66086. Com *path relinking* o valor melhora para 62393. Os valores da distância total sem o *path relinking* são diferentes, mostrando que um número maior de iterações influenciou o resultado. Entretanto, a distância total com *path relinking* foi o mesmo nos dois casos apresentados. O caso de 20000 iterações para resolver cada um dos problemas menores, gerou uma elite de 22 cidades, diferente do caso anterior, de 2000 iterações, que gerou uma elite de 26 cidades. Nos dois casos as cidades elite foram candidatas a mediana na fase de intensificação do *path relinking*.

Outras curiosidades aparecem. Embora com fator de intensificação do *path relinking* igual a 10, o que provocou 200000 iterações nessa fase de intensificação, a solução foi a mesma, e a única diferença foi a ordem em que o algoritmo definiu as medianas com os *clusters* associados, de acordo com a Tabela A.13. Suspeita-se que esta seja uma solução ótima para este número de medianas. Um fato que chama atenção é o de Ribeirão Preto-SP, cidade brasileira com melhor índice de vocação mediana, ser a mediana do maior *cluster* — 57 cidades. Tal característica aparece com destaque nesta observação. Além disso, a Tabela 5.3 mostra uma possível correlação entre o tamanho do *cluster* e o índice de vocação mediana da cidade em questão: na medida em que piora o índice<sub>vm</sub>, diminui o tamanho do *cluster* atendido por aquela mediana. Esta possível correlação entre tamanho do *cluster* e o índice<sub>vm</sub> não será estudada neste trabalho.



**Tabela 5.2:** Resumo dos resultados com 112 cidades, 5 medianas, 20000 iterações.

Instância	Trocas para melhor solução	Distância total	Tempo médio por iteração (seg)
1	3	108493	0,03335000
2	6	68551	0,06615000
3	9	66362	0,10200000
4	6	66086	0,14135000
5	13	68300	0,18560000
6	10	68631	0,22485000
7	19	67489	0,27375001
8	8	67329	0,37114999
9	14	70217	0,41319999
10	13	69277	0,47815001

**Tabela 5.3:** Quando piora o índice de vocação mediana, diminui o tamanho do *cluster*.

Mediana	Número de cidades no <i>cluster</i>	Índice de vocação mediana
Ribeirão Preto-SP	57	1,328
Feira de Santana-BA	19	1,871
Porto Alegre-RS	16	2,056
Terezina-PI	11	2,454
Porto Velho-RO	4	3,347

Uma série de outros experimentos, não divulgados aqui, foram feitos variando alguns parâmetros:

- Fazendo o número de medianas igual 2, 4, 6, 8 e 9
- Variando o total de iterações da GRASP
- Variando o fator de intensificação do *path relinking*.

**Tabela 5.4:** Cidades brasileiras, em ordem alfabética, na matriz de distâncias do DNIT.

Número	Cidade	Número	Cidade	Número	Cidade
1	Americana	39	Guaratinguetá	77	Porto Alegre
2	Anápolis	40	Ilhéus	78	Porto Seguro
3	Aracajú	41	Imperatriz	79	Porto Velho
4	Araçatuba	42	Ipatinga	80	Presidente Prudente
5	Araraquara	43	Itabira	81	Recife
6	Bagé	44	Itabuna	82	Ribeirão Preto
7	Barretos	45	Itajaí	83	Rio Branco
8	Barbacena	46	Itaparica	84	Rio de Janeiro
9	Barreiras	47	Itumbiara	85	Rio Verde
10	Bauru	48	João Pessoa	86	Rondonópolis
11	Belém	49	Joinville	87	Salvador
12	Belo Horizonte	50	Juazeiro do Norte	88	Santa Maria
13	Blumenau	51	Juiz de Fora	89	S. do Livramento
14	Boa Vista	52	Lages	90	Santos
15	Brasília	53	Limeira	91	São Carlos
16	Cachoeiro de Itap.	54	Londrina	92	S. José do Rio Preto
17	Caldas Novas	55	Maceió	93	S. José dos Campos
18	Campina Grande	56	Manaus	94	São Luís
19	Campinas	57	Marília	95	São Lourenço
20	Campo Grande	58	Maringá	96	São Paulo
21	Campos dos Goyt.	59	Montes Claros	97	Sta. Maria da Vitória
22	Caruaru	60	Mossoró	98	Sinop
23	Cascavel	61	Natal	99	Sobral
24	Caxias do Sul	62	Novo Hamburgo	100	Sorocaba
25	Chuí	63	Ourinhos	101	Taubaté
26	Corumbá	64	Palmas	102	Teresina
27	Criciúma	65	Paranaguá	103	Tubarão
28	Cuiabá	66	Parnaíba	104	Tucuruí
29	Curitiba	67	Passo Fundo	105	Uberaba
30	Dourados	68	Patos de Minas	106	Uberlândia
31	Feira de Santana	69	Paulo Afonso	107	Uruguaiana
32	Florianópolis	70	Pelotas	108	Valença
33	Fortaleza	71	Petrolina	109	Vilhena
34	Foz do Iguaçu	72	Petrópolis	110	Vitória
35	Franca	73	Picos	111	V. da Conquista
36	Guaranhuns	74	Poços de Caldas	112	Volta Redonda
37	Goiânia	75	Ponta Grossa		
38	Gov. Valadares	76	Ponta Porã		

**Tabela 5.5:** Índice de vocação mediana das 112 cidades brasileiras médias e grandes.

Número de Ordem	Cidade	Distância total	Índice <sub>vm</sub>	Número de Ordem	Cidade	Distância total	Índice <sub>vm</sub>
1	AMERICANA-SP	373571	3,365	57	MARÍLIA-SP	388102	3,496
2	ANÁPOLIS-GO	349736	3,150	58	MARINGÁ-PR	415095	3,739
3	ARACAJÚ-SE	232642	2,095	59	MONTES CLAROS-MG	312953	2,819
4	ARAÇATUBA-SP	384912	3,467	60	MOSSORÓ-RN	317272	2,858
5	ARARAQUARA-SP	370828	3,340	61	NATAL-RN	301164	2,713
6	BAGÉ-RS	520224	4,686	62	NOVO HAMBURGO-RS	475959	4,287
7	BARRETOS-SP	362978	3,270	63	OURINHOS-SP	391800	3,529
8	BARBACENA-MG	342656	3,086	64	PALMAS-TO	218699	1,970
9	BARREIRAS-BA	301376	2,715	65	PARANAGUÁ-PR	413393	3,724
10	BAURU-SP	382036	3,441	66	PARNAÍBA-PI	362049	3,261
11	BELÉM-PA	311650	2,807	67	PASSO FUNDO-RS	464060	4,180
12	BELO HORIZ-MG	153667	1,384	68	PATOS DE MINAS-MG	334335	3,012
13	BLUMENAU-SC	434125	3,911	69	PAULO AFONSO-BA	256007	2,306
14	BOA VISTA-RR	556965	5,017	70	PELOTAS-RS	254483	2,292
15	BRASÍLIA-DF	158872	1,431	71	PETROLINA-PE	268474	2,418
16	C. DE ITAPEMIRIM-ES	333516	3,004	72	PETRÓPOLIS-RJ	347597	3,131
17	CALDAS NOVAS-GO	343551	3,095	73	PICOS-PI	298725	2,691
18	CAMPINA GRANDE-PB	285137	2,568	74	POÇOS DE CALDAS-MG	361194	3,254
19	CAMPINAS-SP	150832	1,358	75	PONTA GROSSA-PR	416291	3,750
20	CAMPO GRANDE-MS	190504	1,716	76	PONTA PORÃ-MS	452165	4,073
21	CAMPOS DOS GOYTAC-RJ	341316	3,074	77	PORTO ALEGRE-RS	228249	2,056
22	CARUARU-PE	271607	2,446	78	PORTO SEGURO-BA	295473	2,661
23	CASCADEL-PR	441668	3,978	79	PORTO VELHO-RO	371620	3,347
24	CAXIAS DO SUL-RS	217509	1,959	80	PRES. PRUDENTE-SP	396563	3,572
25	CHUÍ-RS	535793	4,826	81	RECIFE-PE	273953	2,468
26	CORUMBÁ-MS	468708	4,222	82	RIBEIRÃO PRETO-SP	147471	1,328
27	CRICIÚMA-SC	458262	4,128	83	RIO BRANCO-AC	429826	3,872
28	CUIABÁ-MT	221652	1,996	84	RIO DE JANEIRO-RJ	167323	1,507
29	CURITIBA-PR	169685	1,528	85	RIO VERDE-GO	371832	3,349
30	DOURADOS-MS	448614	4,041	86	RONDONÓPOLIS-MT	410760	3,700
31	FEIRA DE SANTANA-BA	207691	1,871	87	SALVADOR-BA	217617	1,960
32	FLORIANÓPOLIS-SC	195533	1,761	88	SANTA MARIA-RS	494601	4,455
33	FORTALEZA-CE	308267	2,777	89	S. DO LIVRAMENTO-RS	531354	4,786
34	FOZ DO IGUAÇÚ-PR	456646	4,113	90	SANTOS-SP	159368	1,435
35	FRANCA-SP	359849	3,241	91	SÃO CARLOS-SP	372180	3,352
36	GUARANHUNS-PE	260734	2,348	92	S. J. DO R. PRETO-SP	379612	3,419
37	GOIÂNIA-GO	157562	1,419	93	S. J. DOS CAMPOS-SP	154141	1,388
38	GOV. VALADARES-MG	308359	2,778	94	SÃO LUÍS-MA	297497	2,680
39	GUARATINGUETÁ-SP	360363	3,246	95	SÃO LOURENÇO-MG	339087	3,054
40	ILHÉUS-BA	305398	2,751	96	SÃO PAULO-SP	151522	1,365
41	IMPERATRIZ-MA	379122	3,415	97	S. MARIA DA VIT-BA	350263	3,155
42	IPATINGA-MG	314507	2,833	98	SINOP-MT	486850	4,386
43	ITABIRA-MG	320016	2,883	99	SOBRAL-CE	342933	3,089
44	ITABUNA-BA	273897	2,467	100	SOROCABA-SP	378116	3,406
45	ITAJAÍ-SC	429899	3,872	101	TAUBATÉ-SP	363215	3,272
46	ITAPARICA-BA	270544	2,437	102	TERESINA-PI	272819	2,457
47	ITUMBIARA-GO	359269	3,236	103	TUBARÃO-SC	452019	4,072
48	JOÃO PESSOA-PB	284830	2,566	104	TUCURUÍ-PA	419672	3,780
49	JOINVILLE-SC	181321	1,633	105	UBERABA-MG	361764	3,259
50	JUAZEIRO DO NORTE-CE	294508	2,653	106	UBERLÂNDIA-MG	150823	1,358
51	JUIZ DE FORA-MG	161184	1,452	107	URUGUAIANA-RS	524785	4,727
52	LAGES-SC	445211	4,010	108	VALENÇA-BA	280867	2,530
53	LIMEIRA-SP	368502	3,319	109	VILHENA-RO	515323	4,642
54	LONDRINA-PR	164132	1,478	110	VITÓRIA-ES	180256	1,623
55	MACEIÓ-AL	257475	2,319	111	V. DA CONQUISTA-BA	276382	2,489
56	MANAUS-AM	470779	4,241	112	VOLTA REDONDA-RJ	351591	3,167

**Tabela 5.6:** Índice de vocação mediana das 112 cidades — ordenada pelo índice<sub>vm</sub>.

Número de Ordem	Cidade	Distância total	índice <sub>vm</sub>	Número de Ordem	Cidade	Distância total	índice <sub>vm</sub>
82	RIBEIRÃO PRETO-SP	147471	1,328	8	BARBACENA-MG	342656	3,086
19	CAMPINAS-SP	150832	1,358	99	SOBRAL-CE	342933	3,089
106	UBERLÂNDIA-MG	150823	1,358	17	CALDAS NOVAS-GO	343551	3,095
96	SÃO PAULO-SP	151522	1,365	72	PETRÓPOLIS-RJ	347597	3,131
12	BELO HORIZ-MG	153667	1,384	2	ANÁPOLIS-GO	349736	3,150
93	S. J. DOS CAMPOS-SP	154141	1,388	97	S. MARIA DA VIT-BA	350263	3,155
37	GOIÂNIA-GO	157562	1,419	112	VOLTA REDONDA-RJ	351591	3,167
15	BRASÍLIA-DF	158872	1,431	47	ITUMBIARA-GO	359269	3,236
90	SANTOS-SP	159368	1,435	35	FRANCA-SP	359849	3,241
51	JUIZ DE FORA-MG	161184	1,452	39	GUARATINGUETÁ-SP	360363	3,246
54	LONDRINA-PR	164132	1,478	74	POÇOS DE CALDAS-MG	361194	3,254
84	RIO DE JANEIRO-RJ	167323	1,507	105	UBERABA-MG	361764	3,259
29	CURITIBA-PR	169685	1,528	66	PARNAÍBA-PI	362049	3,261
110	VITÓRIA-ES	180256	1,623	7	BARRETOS-SP	362978	3,270
49	JOINVILLE-SC	181321	1,633	101	TAUBATÉ-SP	363215	3,272
20	CAMPO GRANDE-MS	190504	1,716	53	LIMEIRA-SP	368502	3,319
32	FLORIANÓPOLIS-SC	195533	1,761	5	ARARAQUARA-SP	370828	3,340
31	FEIRA DE SANTANA-BA	207691	1,871	79	PORTO VELHO-RO	371620	3,347
24	CAXIAS DO SUL-RS	217509	1,959	85	RIO VERDE-GO	371832	3,349
87	SALVADOR-BA	217617	1,960	91	SÃO CARLOS-SP	372180	3,352
64	PALMAS-TO	218699	1,970	1	AMERICANA-SP	373571	3,365
28	CUIABÁ-MT	221652	1,996	100	SOROCABA-SP	378116	3,406
77	PORTO ALEGRE-RS	228249	2,056	41	IMPERATRIZ-MA	379122	3,415
3	ARACAJÚ-SE	232642	2,095	92	S. J. DO R. PRETO-SP	379612	3,419
70	PELOTAS-RS	254483	2,292	10	BAURU-SP	382036	3,441
69	PAULO AFONSO-BA	256007	2,306	4	ARAÇATUBA-SP	384912	3,467
55	MACEIÓ-AL	257475	2,319	57	MARÍLIA-SP	388102	3,496
36	GUARANHUNS-PE	260734	2,348	63	OURINHOS-SP	391800	3,529
71	PETROLINA-PE	268474	2,418	80	PRES. PRUDENTE-SP	396563	3,572
46	ITAPARICA-BA	270544	2,437	86	RONDONÓPOLIS-MT	410760	3,700
22	CARUARU-PE	271607	2,446	65	PARANAGUÁ-PR	413393	3,724
102	TERESINA-PI	272819	2,457	58	MARINGÁ-PR	415095	3,739
44	ITABUNA-BA	273897	2,467	75	PONTA GROSSA-PR	416291	3,750
81	RECIFE-PE	273953	2,468	104	TUCURUÍ-PA	419672	3,780
111	V. DA CONQUISTA-BA	276382	2,489	45	ITAJAÍ-SC	429899	3,872
108	VALENÇA-BA	280867	2,530	83	RIO BRANCO-AC	429826	3,872
48	JOÃO PESSOA-PB	284830	2,566	13	BLUMENAU-SC	434125	3,911
18	CAMPINA GRANDE-PB	285137	2,568	23	CASCADEL-PR	441668	3,978
50	JUAZEIRO DO NORTE-CE	294508	2,653	52	LAGES-SC	445211	4,010
78	PORTO SEGURO-BA	295473	2,661	30	DOURADOS-MS	448614	4,041
94	SÃO LUÍS-MA	297497	2,680	103	TUBARÃO-SC	452019	4,072
73	PICOS-PI	298725	2,691	76	PONTA PORÃ-MS	452165	4,073
61	NATAL-RN	301164	2,713	34	FOZ DO IGUAÇU-PR	456646	4,113
9	BARREIRAS-BA	301376	2,715	27	CRICIÚMA-SC	458262	4,128
40	ILHÉUS-BA	305398	2,751	67	PASSO FUNDO-RS	464060	4,180
33	FORTALEZA-CE	308267	2,777	26	CORUMBÁ-MS	468708	4,222
38	GOV. VALADARES-MG	308359	2,778	56	MANAUS-AM	470779	4,241
11	BELÉM-PA	311650	2,807	62	NOVO HAMBURGO-RS	475959	4,287
59	MONTES CLAROS-MG	312953	2,819	98	SINOP-MT	486850	4,386
42	IPATINGA-MG	314507	2,833	88	SANTA MARIA-RS	494601	4,455
60	MOSSORÓ-RN	317272	2,858	109	VILHENA-RO	515323	4,642
43	ITABIRA-MG	320016	2,883	6	BAGÉ-RS	520224	4,686
16	C. DE ITAPEMIRIM-ES	333516	3,004	107	URUGUAIANA-RS	524785	4,727
68	PATOS DE MINAS-MG	334335	3,012	89	S. DO LIVRAMENTO-RS	531354	4,786
95	SÃO LOURENÇO-MG	339087	3,054	25	CHUÍ-RS	535793	4,826
21	CAMPOS DOS GOYTAC-RJ	341316	3,074	14	BOA VISTA-RR	556965	5,017

## 5.4 Experimento comparativo com instâncias de OR Beasley usando *path relinking*

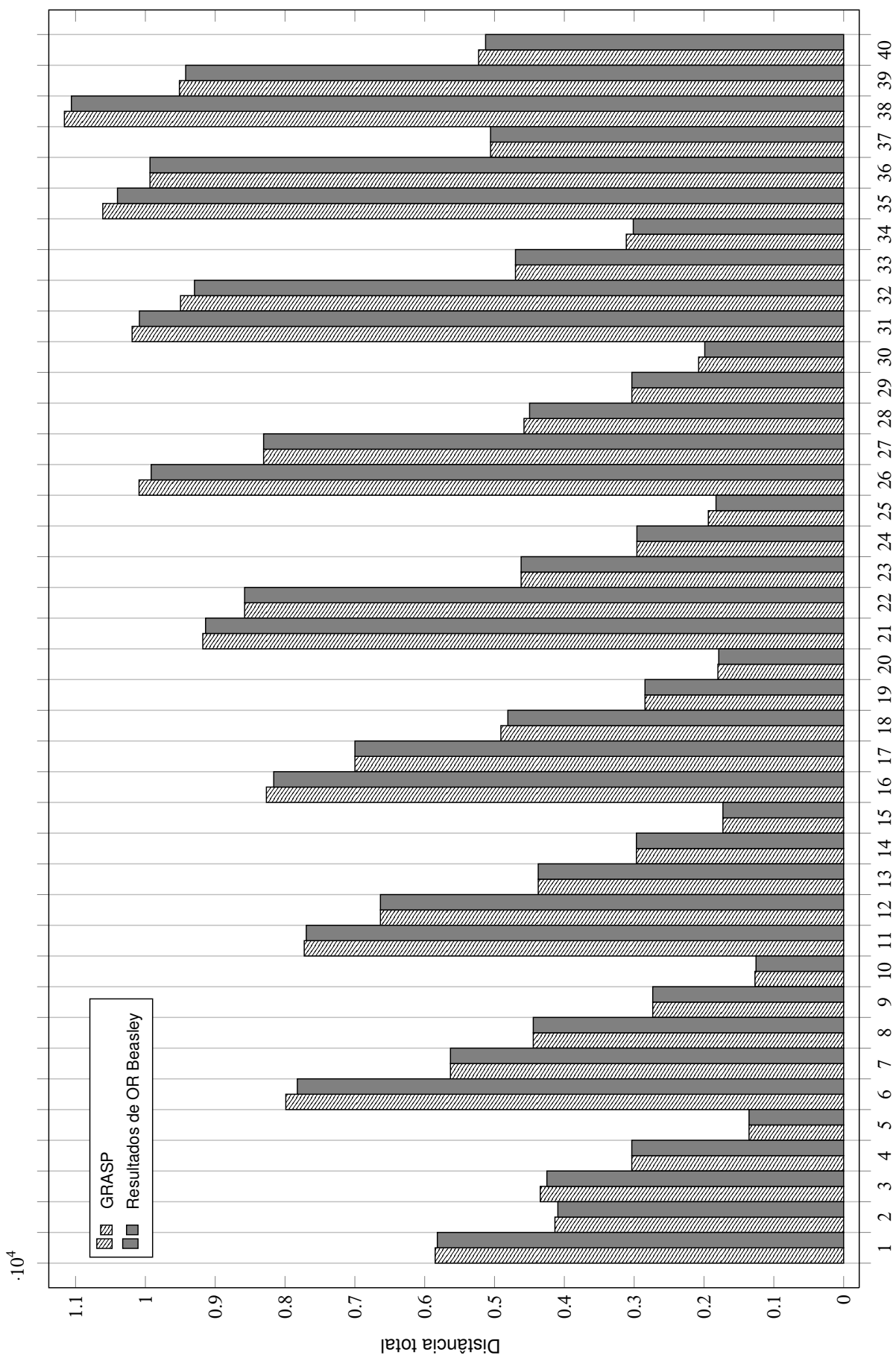
O experimento aqui relatado foi executado antes do ensaio de formação de *clusters* com as cidades brasileiras. A inversão da ordem no momento de apresentar os resultados deve-se ao fato de que parece mais simples mostrá-los e explicar o procedimento de *path relinking* adotado, a partir do experimento com as cidades.

Toda a descrição feita na seção 5.3 para o teste com as cidades, do procedimento de *path relinking* aqui adotado, é semelhante a que poderia ter sido dada para a experiência com os dados da biblioteca de problemas/instâncias de OR Beasley. Em particular, o conceito de “índice<sub>vm</sub>” surgiu no momento de usar os dados da referida biblioteca associados ao procedimento de *path relinking*.

Assim, vários subproblemas foram construídos com cada instância de OR Beasley, considerando somente uma porcentagem  $t\% \times Total\ de\ candidatas\ a\ mediana$ , para o primeiro problema,  $2t\% \times Total\ de\ candidatas\ a\ mediana$  para o segundo problema, e assim por diante, até que  $k \times t\%$  se tornasse igual a 100%, ou seja, todos os possíveis candidatos a medianas. Como já descrito anteriormente, foi tomado o cuidado para que em nenhum dos problemas menores a quantidade de medianas candidatas ficasse inferior ao total máximo de medianas que se desejava localizar.

Dessa forma foi produzido um aglomerado de medianas — acompanhadas dos respectivos *clusters* — do qual participaram as medianas soluções dos subproblemas de cada instância de OR Beasley. Esse aglomerado passou a ser tratado como uma “elite” de soluções e serviu de base para o procedimento de *path relinking* aqui adotado. Foi criado um parâmetro com a dupla função de intensificar e diversificar a busca executada pela GRASP quando aplicada a partir dos dados formados pela elite de soluções. Esse parâmetro de intensificação simplesmente amplificava o número de iterações GRASP a partir dos dados da elite.

Os resultados desse experimento são apresentados pela Figura 5.3 e pela Tabela 5.7, dando uma ideia clara do aprimoramento de que o *path relinking* foi capaz, atendendo ao desejo de alcançar soluções melhores do que aquelas obtidas antes da intensificação do *path relinking*, conforme mostra a Figura 4.5.



**Figura 5.3:** Experimento: comparação do valor da distância total com Beasley, usando *path relinking*.

**Tabela 5.7:** Resultados do experimento para avaliar a GRASP, comparando com instâncias da biblioteca OR Beasley usando *path relinking*.

Instância	Número de Cidades	Número Máximo de Medianas	Distância Total Para Uma Única Iteração GRASP <sup>a</sup>	Distância Total Com <i>Path Relinking</i>	Tempo Médio Por Iteração GRASP (seg)	Solução Ótima OR Beasley
1	100	10	13758	5850	0.04329	5819
2	100	10	14199	4135	0.05647	4093
3	100	20	5177	4345	0.09845	4250
4	100	33	3109	3034	0.12500	3034
5	100	5	14846	1355	0.04012	1355
6	200	10	17628	7989	0.10111	7824
7	200	20	13264	5631	0.18564	5631
8	200	40	6622	4445	0.27043	4445
9	200	5	30180	2734	0.06342	2734
10	200	67	4078	1270	0.45590	1255
11	300	100	8368	7725	1.97000	7696
12	300	10	37160	6634	0.21000	6634
13	300	30	13441	4374	0.38746	4374
14	300	5	35504	2968	0.12983	2968
15	300	60	8878	1729	0.64912	1729
16	400	10	36844	8267	0.26589	8162
17	400	133	7625	6999	1.88431	6999
18	400	40	16324	4909	0.64388	4809
19	400	5	92249	2845	0.18200	2845
20	400	80	8175	1801	1.13422	1789
21	500	100	9813	9178	1.77790	9138
22	500	10	49147	8579	0.45001	8579
23	500	167	4663	4619	2.82993	4619
24	500	50	16601	2961	1.00333	2961
25	500	5	51760	1938	0.35009	1828
26	600	10	55100	10091	0.61015	9917
27	600	120	8550	8307	2.76948	8307
28	600	200	4889	4578	3.15711	4498
29	600	5	39606	3033	0.41200	3033
30	600	60	18299	2077	1.47880	1989
31	700	10	75653	10190	0.66328	10086
32	700	140	9821	9497	3.52439	9297
33	700	5	43166	4700	0.53500	4700
34	700	70	17725	3113	2.09800	3013
35	800	10	79614	10610	0.91003	10400
36	800	5	35601	9934	0.71029	9934
37	800	80	20094	5057	2.85429	5057
38	900	10	86988	11160	1.27111	11060
39	900	5	31071	9513	0.96189	9423
40	900	90	18949	5228	2.53922	5128

<sup>a</sup>Coluna obtida do experimento anterior — sem *path relinking*.

# CAPÍTULO VI

## O CÓDIGO COMPUTACIONAL

### 6.1 Introdução

Os códigos computacionais foram feitos em C++, entretanto, sem usar toda a sofisticação possível de se alcançar através dos conceitos de polimorfismo, funções virtuais e virtuais puras, classes abstratas e interfaces. O código foi construído buscando um nível razoável de simplicidade. Foram usados alguns componentes da poderosa biblioteca STL — como vector, set, list e map — o que permitiu desenvolver um código estável, com alto grau de generalidade e baseado em estruturas de dados exaustivamente testadas. A implementação de C++ utilizada neste projeto está contida na versão 2010 do Visual Studio que é um ambiente de desenvolvimento bem estabelecido para fins industriais, comerciais e acadêmicos.

O código foi projetado e construído com base em uma classe de membros, dados e funções, que permitiram modular as várias operações envolvidas na solução do problema. As informações que alimentavam o código foram constituídas do número  $|N| = n$  de pontos, cidades ou vértices envolvidos e que formam o conjunto  $N$ . Além desse número, os dados continham a matriz simétrica de distâncias, de dimensão  $n \times n$  e o número máximo de medianas que a instância pretendia localizar. Um determinado número de ligações compunha algumas entradas da matriz de distâncias. Posteriormente, as demais entradas foram calculadas usando uma rotina que implementa o algoritmo — de programação dinâmica,  $\mathcal{O}(n^3)$ , detalhes em Nemhauser (1966) — de Floyd-Warshall, Algoritmo 7, para encontrar as distâncias mínimas entre todos os pares de pontos ou vértices que se associam às entradas da matriz. Um vetor de custos fixos foi estimado para teste. Entretanto, seu uso foi limitado, já que as instâncias utilizadas para teste e comparação, eram provenientes da biblioteca OR Beasley que não tem essa informação.

A matriz de distâncias ou custos variáveis tem o formato semelhante ao da Tabela 6.1, que representa os dados iniciais de uma instância — antes da aplicação do algoritmo



de Floyd-Warshall.

Os principais passos do algoritmo de Floyd-Warshall consistem de três laços, cada um interno ao anterior, a partir do segundo laço, para calcular os  $D^k(i, j)$ , de acordo com a Fórmula 6.1, em que  $D^k(i, j)$  é a distância do vértice  $i$  ao vértice  $j$ .

**Tabela 6.1:** Matriz de distâncias antes da aplicação do algoritmo de Floyd-Warshall, onde  $\infty$  representa distâncias ainda desconhecidas.

	1	2	3	4	5	6
1	0	$\infty$	8	$\infty$	6	12
2	$\infty$	0	$\infty$	3	$\infty$	$\infty$
3	8	$\infty$	0	5	7	$\infty$
4	$\infty$	3	5	0	9	1
5	6	$\infty$	7	9	0	10
6	12	$\infty$	$\infty$	1	10	0

$$D^{k+1}(i, j) = \min \left\{ D^k(i, j), D^k(i, k) + D^k(k, j) \right\}, \quad \text{para } k = 1, \dots, n - 1 \quad (6.1)$$

As operações conduzidas pelo algoritmo de Floyd-Warshall correspondem a  $n^3$  operações para calcular todos os  $D^{k+1}(i, j)$  a partir dos  $D^k(i, j)$ , portanto, um algoritmo de complexidade assintótica  $\mathcal{O}(n^3)$ .

---

**Algoritmo 7** Floyd-Warshall.

---

**executa:** *Floyd-Warshall – matriz de distâncias*

- 1 **para**  $k = 1$  **até**  $n - 1$  **faça**
  - 2     **para**  $i = 1$  **até**  $n$  **faça**
  - 3         **para**  $j = 1$  **até**  $n$  **faça**
  - 4              $D^{k+1}(i, j) = \min \left\{ D^k(i, j), D^k(i, k) + D^k(k, j) \right\};$
  - 5         **fim para**
  - 6     **fim para**
  - 7 **fim para**
  - 8 **fim** *Floyd-Warshall*
- 

Após a aplicação do Algoritmo 7, a matriz de distância torna-se de acordo com o que mostra a Tabela 6.2.

**Tabela 6.2:** Matriz de distâncias após a aplicação do algoritmo de Floyd-Warshall.

	1	2	3	4	5	6
1	0	16	8	13	6	12
2	16	0	8	3	12	4
3	8	8	0	5	7	20
4	13	3	5	0	9	1
5	6	12	7	9	0	10
6	12	4	20	1	10	0

## 6.2 A implementação da GRASP com *path relinking*

A GRASP duplamente reativa foi processada por rotinas que implementaram a GRASP em si, a fase de construção e a busca local ou melhoria. Além disso, uma rotina dedicada ao *path relinking* foi também implementada e amplamente testada. Na fase de melhoria ou busca local, os vértices eram transferidos de um *cluster* para outro quando essa transferência provocava um ganho de distância ou custo. Ou seja, a transferência se dava para o *cluster* cuja mediana era a mais próxima do vértice em foco. Além disso, podia acontecer que um vértice não era alocado a qualquer *cluster* após o procedimento normal da GRASP. Neste caso, o vértice era alocado na fase de melhoria, ao *cluster* da mediana mais próxima.

No experimento com as cidades brasileiras médias e grandes da matriz de distâncias do DNIT, os vértices que agora passam a ser as cidades, deixaram de ser representados por número e passaram a ser referidos pelos nomes reais das cidades correspondentes, que constam da matriz do DNIT. Isto permitiu visualizar melhor a coerência do procedimento de localização e a formação dos *clusters*.

Os resultados são emitidos na forma de uma descrição como a informação que consta da Tabela 2.2, ou da Tabela A.13.

## CAPÍTULO VII

# CONCLUSÃO E SUGESTÕES PARA PESQUISAS FUTURAS

### 7.1 Conclusões

Este trabalho se baseou na hipótese de que havia espaço para aprimoramento de métodos que resolvem problemas de otimização usando metaheurísticas. Em particular, a escolha para demonstrar a veracidade desta hipótese recaiu sobre a metaheurística GRASP, seus métodos reativos e procedimentos de *path relinking*, especializados na solução do modelo generalizado de localização de  $p$ -medianas. Para alcançar o objetivo foi desenvolvido razoável esforço de pesquisa que buscou explorar o processo reativo da GRASP por duas vias, ou seja, adotando o que se passou a denominar de “GRASP duplamente reativa,” já que dois parâmetros de reação passaram a atuar, cooperativamente, nesse cenário. Para entendimento próprio e para comunicar o procedimento à comunidade interessada no assunto, passou-se a explicar tal procedimento com uma metáfora que aqui foi denominada de “metáfora da peneira”. Todo o esforço feito confirmou a boa reputação da metaheurística GRASP no contexto da resolução de problemas combinatórios destinados a aplicações tecnológicas do cotidiano.

A questão básica aqui tratada se constitui de como decidir, a partir de locais elegíveis conhecidos ou a partir de um grande número de locais descritos como área, uma localização, ou suas coordenadas, onde instalar uma facilidade e como atribuir ou alocar clientes a ela. O procedimento desenvolvido para resolver este problema foi testado exaustivamente e comparado com os resultados de instâncias da biblioteca online OR Beasley. Os resultados foram estimulantes, embora ficassem num patamar inferior aos de Beasley, uma vez que lá os resultados foram produzidos por algoritmos com capacidade de alcançar a solução ótima.

O passo seguinte foi o de explorar as capacidades do *path relinking*. As experiências realizadas, também de forma exaustiva, mostraram tratar-se de procedimento com

poder de aproximar os resultados da metaheurística GRASP aos de soluções dos algoritmos exatos. A principal proposta desse método é a de intensificar a busca entre soluções elite encontradas nas fases anteriores do algoritmo. Novas comparações com os resultados de Beasley se encarregaram de confirmar o que foi progressivamente sendo descoberto durante a pesquisa. De todas as formas acredita-se que não está esgotado o espaço de aprimoramentos aqui explorado, o que constitui estímulo para prosseguir testando ideias semelhantes. Além disso, tal esforço sempre se ocupa de um aspecto particular desse problema que é amplo e variado nas diversas modalidades do problema e suas aplicações.

A pesquisa prosseguiu agora com duas vertentes. Continuou o espírito de teste de um lado, mas agora, de outro lado, ele se fez acompanhar do uso da inovação conquistada sobre a ideia de simular a formação de *clusters* com as cidades brasileiras, médias e grandes, constantes da matriz de distâncias do DNIT. Essas simulações que, a partir daí, passaram a ser conduzidas com dados reais, se enquadram no que se pode chamar de contribuição à logística de localização. A extensão dos experimentos foi relativamente grande na medida em que se procurou variar a quantidade de *clusters*, o número de iterações, assim como os valores do parâmetro de intensificação do *path relinking* que passou a ser usado em caráter definitivo.

Um subproduto interessante dessa fase de simulação de *clusters* foi a invenção de um índice que passou a ser chamado de “índice de vocação mediana” que é uma representação numérica capaz de medir a proximidade da cidade em questão com as demais cidades brasileiras, médias e grandes, constantes da matriz do DNIT. Esta é uma informação logística de valor tanto para um estudo locacional qualitativo, quanto para projetos de localização em que se explora o lado quantitativo. A pesquisa descobriu por via dos experimentos conduzidos que a cidade detentora do melhor índice de vocação mediana é RIBEIRÃO PRETO–SP, e a cidade com o pior índice é BOA VISTA–RR. Essa informação não exclui as cidades intermediárias em relação a este índice, já que participando de uma análise locacional elas podem apresentar outras características interessantes, tanto no aspecto qualitativo, quanto do lado quantitativo.

Todo o trabalho foi precedido do conhecimento e avaliação, por via da literatura existente, de outras metaheurísticas, o que foi aqui registrado na forma do APÊNDICE B. Além disso, aprendeu-se que a localização de instalações e alocação de clientes a

elas determina um padrão de distribuição com características associadas, tais como custo, tempo e eficiência do esquema de distribuição. A instalação de facilidades em locais ótimos, ou próximos a esses, juntamente com a atribuição de clientes a essas facilidades da forma mais eficiente possível, não só aprimora o fluxo de materiais e serviços ofertados, mas, além disso, usa as instalações de maneira ótima, reduzindo, a inconveniência de redundâncias no processo locacional.

A natureza do problema se encarregou de definir, ou permitiu um entendimento de termos como “instalação” ou “facilidade” e “clientes” a ela atribuídos ou alocados. Por exemplo, na determinação de locais adequados para posicionar indústrias em condições tais que lhes permitam atender a demanda proveniente das várias regiões do país, as indústrias são as facilidades e os usuários dos produtos, os clientes. Um outro exemplo se descreve como segue. No mercado de linhas aéreas a localização de um *hub* é importante para determinar como atender a várias cidades — clientes — que são conectadas por serviços de transporte rápido de curta distância ao *hub* que é a facilidade. Outros exemplos poderiam incluir hospitais, corpo de bombeiros, usinas elétricas, em que fica fácil compreender o que é a facilidade e quais são os clientes.

Outro item que adveio da preparação para este trabalho foi a observação de que a aplicação de conceitos associados a problemas de localização não está restrita a facilidades. Outros exemplos podem ser sugeridos em áreas que vão da organização de máquinas no chão de fábrica à instalação de centros de atendimento hospitalar com capacidade de responder a emergências tão rápido quanto possível. Além disso, observou-se que os enfoques de abordagem dos problemas locacionais podem ser classificados em duas grandes áreas: qualitativa e quantitativa — cada uma com seu mérito, mas, também com sua deficiência. O enfoque qualitativo padece de possível subjetividade dos critérios de escolha e da dependência do gosto e da preferência de quem escolhe ou decide. Possivelmente, o melhor enfoque de avaliação de locais para construir instalações consiste em usar um procedimento híbrido em que primeiro se usa o enfoque quantitativo e, posteriormente, adota-se o exame dos locais eleitos pelo método qualitativo, que permite observar os requisitos mínimos dos fatores subjetivos.

## 7.2 Sugestões para trabalhos futuros

Uma série de ampliações e inovações podem ser conduzidas a partir deste trabalho. Pesquisas futuras podem produzir uma sintonia fina dos procedimentos aqui inventados e estendê-los à solução de variações importantes do tipo de problema locacional aqui tratado. Inicialmente pode-se abordar o problema capacitado de  $p$ -medianas descrito neste trabalho na forma do modelo representado pelas relações (2.7) a (2.17). No problema capacitado, tanto clientes quanto fornecedores têm capacidades limitadas.

Outra questão implica em equilibrar os *clusters*, já que em algumas circunstâncias uma diferença elevada entre os números de componentes dos *clusters* pode ser inadequada. Esta questão é vizinha dos interesses do problema capacitado — pode-se limitar também o número de componentes dos *clusters* em busca do equilíbrio aqui aventado.

Uma vez estabelecidos os centros de distribuição e os *clusters* que lhes correspondem, pode-se complementar tal construção com um esquema de roteamento para aprimorar a logística do processo de distribuição. Este constitui outro aspecto importante que merece um projeto detalhado para conduzir sua pesquisa.

Outras aplicações consistem em reavaliar projetos locacionais já estabelecidos e, eventualmente, propor mudanças e aprimoramentos da estrutura logística atualmente em uso. Para ilustrar o que foi dito e lembrando que se trata de questão relevante, considera-se o problema de localização de centros de distribuição (CD's) de uma indústria. Atualmente, a empresa possui em torno de 600 clientes distribuídos nas principais regiões do País.

Trata-se de empresa que tem o foco na excelência operacional, implementando TPM (Total Productive Maintenance) nas fábricas, através de melhoras na sua cadeia de suprimentos, promovendo o desenvolvimento da função ERP (Enterprise Resource Planning). Ela visa melhorar constantemente os serviços oferecidos aos clientes e a eficiência no gerenciamento de pedidos, logística e compra. A empresa está ciente de que, de acordo com as normas da NBR ISO 9000, é conveniente que a melhoria contínua do desempenho global da organização seja seu objetivo permanente. O

objetivo da melhoria contínua de um sistema de gestão da qualidade é garantir a satisfação dos clientes e das outras partes interessadas, como, por exemplo, o público considerado estratégico para a empresa, ou seja, seus Stakeholders.

A logística da empresa, aqui hipotética, opera com dois Centros de Distribuição, que, segundo a Associação Brasileira de Logística (Aslog), são armazéns que têm por objetivo realizar a gestão dos estoques de mercadorias na distribuição física. Em geral, armazéns recebem cargas consolidadas de um ou mais fornecedores, mas no caso de algumas indústrias, os CD's recebem mercadorias de apenas uma fábrica. Essas cargas são então fracionadas e embarcadas, provendo os produtos na quantidade e variedade corretas e ao cliente determinado.

Os CD's buscam um rápido atendimento às necessidades dos clientes, melhorando assim, a qualidade dos serviços da empresa. Por funcionarem como centros consolidados de cargas e posicionarem o estoque em vários elos de uma cadeia de suprimentos, eles possibilitam a obtenção de economias de transporte e permitem um rápido atendimento às necessidades dos clientes que se encontram distantes das fábricas.

Farah (2002) considera o CD como um instrumento capaz de viabilizar, de forma competitiva, o fluxo de mercadorias entre os fabricantes e os diversos graus de capilaridade distributiva. Para o autor, a adoção de CD's se justifica pela necessidade de obtenção de uma distribuição mais eficiente, flexível e dinâmica, o que se traduz em capacidade de resposta rápida às procuras cada vez mais frequentes e específicas dos clientes, tendo como principais vantagens a redução de custos das entidades participantes da cadeia de suprimentos e a extinção de pontos de estrangulamento.

No caso da indústria hipotética aqui imaginada, um CD está localizado, diga-se, na cidade do Rio de Janeiro, encarregado de suprir os clientes do Estado. O outro, está localizado, por exemplo, na cidade de São Paulo, incumbido da distribuição para os clientes das demais regiões do País.

Para suprir os CD's e os clientes finais a empresa subcontrata transportadoras e o serviço de transbordo através dos CD's. O pedido do cliente é registrado no sistema da empresa, em seguida é colocado em uma lista de encomendas que serão enviadas ao CD mais apropriado para realizar a entrega. Caminhões de carga-lotação (FTL –

*Full Truck Load*), adequadamente dimensionados, são carregados diariamente até o limite da capacidade e saem em direção aos CD's. Quando chegam ao CD, as cargas são separadas, conforme os pedidos, e reembarcadas em direção aos clientes finais através de caminhões de carga fracionada (*LTL – Less Than Truck Load*).

De acordo com Wanke (2001), um princípio básico envolvendo a abertura de um novo CD é a consolidação do transporte. O custo de transporte cai na medida em que novos CD's são adicionados à rede logística, pois uma maior quantidade de cargas consolidadas é enviada ao novo CD, em conjunto com distâncias mais curtas que precisarão ser percorridas pela carga já fracionada até o cliente final. Porém, se o número de armazéns aumenta muito, o custo de transporte cresce, pois o volume da carga consolidada de cada instalação tende a diminuir.

O projeto que aqui se propõe como continuidade do estudo feito nesta dissertação, trataria de avaliar possíveis alterações na logística atual dessa empresa de modo a apoiar eventuais decisões quanto à instalação de um ou mais novos centros de distribuição em locais sugeridos pelo sistema locacional que seria construído de forma a minimizar os custos de transporte, mas sem prejudicar a qualidade dos serviços prestados.

Modificar a localização da fábrica não seria conveniente pois o local foi escolhido estrategicamente em função da disponibilidade de matéria-prima e mão de obra na região. A empresa pode não trabalhar com a hipótese de realocar os atuais CD's, uma vez que grande parte dos clientes se concentra próximos às cidades, Rio de Janeiro e São Paulo, dadas como exemplos nesta proposta.

Para calcular o custo do transporte, seria necessária a quilometragem percorrida nos modais, passando pelos CD's, até o cliente final. Cada modal tem o custo (frete) a ser taxado e, por isso, é importante conhecer cada distância, tanto da fábrica ao CD, quanto do CD ao cliente. Nesse ponto, entraria novamente em cena a matriz de distâncias do DNIT.

Para o cálculo dos custos de transporte é necessário conhecer o custo do frete que ocorre em cada modal. O modal em FTL percorre apenas a distância entre a fábrica e o CD. Do CD para o cliente final, a carga é fracionada e transportada em caminhão LTL com custo de frete muito maior. Consultando tabelas de fretes fornecidas por



empresas, observa-se uma diferença enorme entre carga consolidada e carga fracionada no que diz respeito a esse item. Para distâncias muito curtas, o frete chega a ser dez vezes mais caro que a mesma distância em carga consolidada — possivelmente estão presentes nesse contexto questões relacionadas a escala.

O custo fixo das instalações é uma questão de grande importância para a decisão final. Um dos fatores que ocasionam os custos fixos é a área total que o CD necessitaria, contas de água e luz, mão de obra, aluguel, limpeza, equipamentos, etc. Sabe-se ainda que os valores variam de acordo com as regiões. Por exemplo: a região sudeste tem custos operacionais mais caros, mão de obra mais qualificada, menor oferta de imóveis, do que a região nordeste.

Certamente, existem outras oportunidades de progredir com a pesquisa aqui desenvolvida. Em particular, seria interessante explorar questões ligadas ao processamento paralelo que é um recurso amplamente disponível nos processadores multi-core em uso nos dias atuais. Além disso, é um desafio estimulante desfrutar das inovações computacionais modernas que disponibilizam bibliotecas de software poderoso adequado ao uso em contextos como o dos problemas aqui tratados e sugeridos. Entretanto, e de qualquer forma, as sugestões dadas merecem ser avaliadas para, a partir delas, conduzir trabalhos futuros.

# APÊNDICE A

## EXPERIMENTO COMPUTACIONAL COM O *PATH* *RELINKING*

### A.1 Resultado de experimento locacional com as 112 cidades brasileiras médias e grandes para teste do *path relinking*

Em cada instância neste ensaio foram localizadas 5 medianas, com 2000 iterações GRASP. Na fase do *path relinking* o fator de intensificação adotado foi igual a 50, o que provocou 100000 iterações da GRASP. A exceção aqui é representada pela Tabela A.13, que corresponde à solução final com *path relinking* usando 20000 iterações para os problemas menores, fator de intensificação igual a 10, o que provocou 200000 iterações na fase de intensificação.

As Tabelas que vão de A.1 a A.13 mostram os resultados do ensaio com as 112 cidades brasileiras médias e grandes para teste do *path relinking* e demonstração de outras possibilidades que podem ser levadas a cabo com instrumentos locais.

Além desses, outros experimentos, não relatados aqui, foram feitos, como, por exemplo, utilizar para candidatas a mediana as cidades com índice<sub>vm</sub> inferior a 2,00. Utilizar as cidades com índice<sub>vm</sub> inferior a um dado índice<sub>vm</sub> e, progressivamente, aumentar o número delas com cidades de índice<sub>vm</sub> pior, aumentando com isso a diversidade do processo.

**Tabela A.1:** Resultado (1) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 108493

1) *Cluster* da mediana UBERLÂNDIA-MG:

ANÁPOLIS-GO	BARRETOS-SP	BARREIRAS-BA	BELÉM-PA
BOA VISTA-RR	BRASÍLIA-DF	CALDAS NOVAS-GO	CAMPO GRANDE-MS
CORUMBÁ-MS	CUIABÁ-MT	FRANCA-SP	GOIÂNIA-GO
IMPERATRIZ-MA	ITUMBIARA-GO	MANAUS-AM	MONTES CLAROS-MG
PALMAS-TO	PARNAÍBA-PI	PATOS DE MINAS-MG	PICOS-PI
PORTO VELHO-RO	RIO BRANCO-AC	RIO VERDE-GO	RONDONÓPOLIS-MT
S. J. DO R. PRETO-SP	SÃO LUÍS-MA	SINOP-MT	SOBRAL-CE
TERESINA-PI	TUCURUÍ-PA	UBERABA-MG	VILHENA-RO

Total de cidades neste *cluster*: 32. Distância (custo) total neste *cluster*: 448362) *Cluster* da mediana CAMPINAS-SP:

AMERICANA-SP	ARAÇATUBA-SP	ARARAQUARA-SP	BAURU-SP
GUARATINGUETÁ-SP	LIMEIRA-SP	MARÍLIA-SP	OURINHOS-SP
POÇOS DE CALDAS-MG	RIBEIRÃO PRETO-SP	SANTOS-SP	SÃO CARLOS-SP
S. J. DOS CAMPOS-SP	SÃO PAULO-SP	SOROCABA-SP	TAUBATÉ-SP

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 32943) *Cluster* da mediana FLORIANÓPOLIS-SC:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	ITAJAÍ-SC	LAGES-SC	NOVO HAMBURGO-RS
PASSO FUNDO-RS	PELOTAS-RS	PORTO ALEGRE-RS	SANTA MARIA-RS
S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS	

Total de cidades neste *cluster*: 15. Distância (custo) total neste *cluster*: 82424) *Cluster* da mediana CURITIBA-PR:

CASCAVEL-PR	DOURADOS-MS	FOZ DO IGUAÇÚ-PR	JOINVILLE-SC
LONDRINA-PR	MARINGÁ-PR	PARANAGUÁ-PR	PONTA GROSSA-PR
PONTA PORÃ-MS	PRES. PRUDENTE-SP		

Total de cidades neste *cluster*: 10. Distância (custo) total neste *cluster*: 46805) *Cluster* da mediana JUIZ DE FORA-MG:

ARACAJÚ-SE	BARBACENA-MG	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES
CAMPINA GRANDE-PB	CAMPOS DOS GOYTAC-RJ	CARUARU-PE	FEIRA DE SANTANA-BA
FORTALEZA-CE	GUARANHUNS-PE	GOV. VALADARES-MG	ILHÉUS-BA
IPATINGA-MG	ITABIRA-MG	ITABUNA-BA	ITAPARICA-BA
JOÃO PESSOA-PB	JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN
NATAL-RN	PAULO AFONSO-BA	PETROLINA-PE	PETRÓPOLIS-RJ
PORTO SEGURO-BA	RECIFE-PE	RIO DE JANEIRO-RJ	SALVADOR-BA
SÃO LOURENÇO-MG	S. MARIA DA VIT-BA	VALENÇA-BA	VITÓRIA-ES
V. DA CONQUISTA-BA	VOLTA REDONDA-RJ		

Total de cidades neste *cluster*: 34. Distância (custo) total neste *cluster*: 47441

**Tabela A.2:** Resultado (2) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 68551

1) *Cluster* da mediana FEIRA DE SANTANA-BA:

ARACAJÚ-SE	BARREIRAS-BA	BELÉM-PA	CAMPINA GRANDE-PB
CARUARU-PE	FORTALEZA-CE	GUARANHUNS-PE	ILHÉUS-BA
IMPERATRIZ-MA	ITABUNA-BA	ITAPARICA-BA	JOÃO PESSOA-PB
JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN	NATAL-RN
PARNAÍBA-PI	PAULO AFONSO-BA	PETROLINA-PE	PICOS-PI
PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA	SÃO LUÍS-MA
S. MARIA DA VIT-BA	SOBRAL-CE	TERESINA-PI	TUCURUÍ-PA
VALENÇA-BA	V. DA CONQUISTA-BA		

Total de cidades neste *cluster*: 30. Distância (custo) total neste *cluster*: 257262) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	ITAJAÍ-SC	JOINVILLE-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 63803) *Cluster* da mediana JUIZ DE FORA-MG:

BARBACENA-MG	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES	CAMPOS DOS GOYTAC-RJ
GOV. VALADARES-MG	GUARATINGUETÁ-SP	IPATINGA-MG	ITABIRA-MG
MONTES CLAROS-MG	PETRÓPOLIS-RJ	RIO DE JANEIRO-RJ	SÃO LOURENÇO-MG
TAUBATÉ-SP	VITÓRIA-ES	VOLTA REDONDA-RJ	

Total de cidades neste *cluster*: 15. Distância (custo) total neste *cluster*: 50184) *Cluster* da mediana CAMPO GRANDE-MS:

BOA VISTA-RR	CASCADEL-PR	CORUMBÁ-MS	CUIABÁ-MT
DOURADOS-MS	FOZ DO IGUAÇÚ-PR	MANAUS-AM	PONTA PORÃ-MS
PORTO VELHO-RO	PRES. PRUDENTE-SP	RIO BRANCO-AC	RONDONÓPOLIS-MT
SINOP-MT	VILHENA-RO		

Total de cidades neste *cluster*: 14. Distância (custo) total neste *cluster*: 183725) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BAURU-SP	BRASÍLIA-DF	CALDAS NOVAS-GO
CAMPINAS-SP	CURITIBA-PR	FRANCA-SP	GOIÂNIA-GO
ITUMBIARA-GO	LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP
MARINGÁ-PR	OURINHOS-SP	PALMAS-TO	PARANAGUÁ-PR
PATOS DE MINAS-MG	POÇOS DE CALDAS-MG	PONTA GROSSA-PR	RIO VERDE-GO
SANTOS-SP	SÃO CARLOS-SP	S. J. DO R. PRETO-SP	S. J. DOS CAMPOS-SP
SÃO PAULO-SP	SOROCABA-SP	UBERABA-MG	UBERLÂNDIA-MG

Total de cidades neste *cluster*: 32. Distância (custo) total neste *cluster*: 13055

**Tabela A.3:** Resultado (3) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 68613

1) *Cluster* da mediana CAXIAS DO SUL-RS:

BAGÉ-RS	BLUMENAU-SC	CASCADEL-PR	CHUÍ-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	FOZ DO IGUAÇÚ-PR	ITAJAÍ-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
PORTO ALEGRE-RS	SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC
URUGUAIANA-RS			

Total de cidades neste *cluster*: 17. Distância (custo) total neste *cluster*: 80292) *Cluster* da mediana SALVADOR-BA:

ARACAJÚ-SE	BARREIRAS-BA	CAMPINA GRANDE-PB	CARUARU-PE
FEIRA DE SANTANA-BA	GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA
ITAPARICA-BA	JOÃO PESSOA-PB	MACEIÓ-AL	NATAL-RN
PAULO AFONSO-BA	PETROLINA-PE	PORTO SEGURO-BA	RECIFE-PE
VALENÇA-BA	V. DA CONQUISTA-BA		

Total de cidades neste *cluster*: 18. Distância (custo) total neste *cluster*: 109043) *Cluster* da mediana SÃO PAULO-SP:

AMERICANA-SP	ARAÇATUBA-SP	ARARAQUARA-SP	BARRETOS-SP
BARBACENA-MG	BAURU-SP	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES
CAMPINAS-SP	CAMPOS DOS GOYTAC-RJ	CURITIBA-PR	DOURADOS-MS
FRANCA-SP	GOV. VALADARES-MG	GUARATINGUETÁ-SP	IPATINGA-MG
ITABIRA-MG	JOINVILLE-SC	JUIZ DE FORA-MG	LIMEIRA-SP
LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR	OURINHOS-SP
PARANAGUÁ-PR	PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG	PONTA GROSSA-PR
PONTA PORÃ-MS	PRES. PRUDENTE-SP	RIBEIRÃO PRETO-SP	RIO DE JANEIRO-RJ
SANTOS-SP	SÃO CARLOS-SP	S. J. DO R. PRETO-SP	S. J. DOS CAMPOS-SP
SÃO LOURENÇO-MG	SOROCABA-SP	TAUBATÉ-SP	VITÓRIA-ES
VOLTA REDONDA-RJ			

Total de cidades neste *cluster*: 41. Distância (custo) total neste *cluster*: 186774) *Cluster* da mediana TERESINA-PI:

BELÉM-PA	FORTALEZA-CE	IMPERATRIZ-MA	JUAZEIRO DO NORTE-CE
MOSSORÓ-RN	PARNAÍBA-PI	PICOS-PI	SÃO LUÍS-MA
SOBRAL-CE	TUCURUÍ-PA		

Total de cidades neste *cluster*: 10. Distância (custo) total neste *cluster*: 64495) *Cluster* da mediana GOIÂNIA-GO:

ANÁPOLIS-GO	BOA VISTA-RR	BRASÍLIA-DF	CALDAS NOVAS-GO
CAMPO GRANDE-MS	CORUMBÁ-MS	CUIABÁ-MT	ITUMBIARA-GO
MANAUS-AM	MONTES CLAROS-MG	PALMAS-TO	PATOS DE MINAS-MG
PORTO VELHO-RO	RIO BRANCO-AC	RIO VERDE-GO	RONDONÓPOLIS-MT
S. MARIA DA VIT-BA	SINOP-MT	UBERABA-MG	UBERLÂNDIA-MG
VILHENA-RO			

Total de cidades neste *cluster*: 21. Distância (custo) total neste *cluster*: 24554

**Tabela A.4:** Resultado (4) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 68899

1) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ARAÇATUBA-SP	ARARAQUARA-SP	BARRETOS-SP
BARBACENA-MG	BAURU-SP	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES
CAMPINAS-SP	CAMPOS DOS GOYTAC-RJ	CASCADEL-PR	CURITIBA-PR
DOURADOS-MS	FOZ DO IGUAÇÚ-PR	FRANCA-SP	GOV. VALADARES-MG
GUARATINGUETÁ-SP	IPATINGA-MG	ITABIRA-MG	JUIZ DE FORA-MG
LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR
OURINHOS-SP	PARANAGUÁ-PR	PATOS DE MINAS-MG	PETRÓPOLIS-RJ
POÇOS DE CALDAS-MG	PONTA GROSSA-PR	PONTA PORÃ-MS	PRES. PRUDENTE-SP
RIO DE JANEIRO-RJ	SANTOS-SP	SÃO CARLOS-SP	S. J. DO R. PRETO-SP
S. J. DOS CAMPOS-SP	SÃO LOURENÇO-MG	SÃO PAULO-SP	SOROCABA-SP
TAUBATÉ-SP	UBERABA-MG	UBERLÂNDIA-MG	VITÓRIA-ES
VOLTA REDONDA-RJ			

Total de cidades neste *cluster*: 45. Distância (custo) total neste *cluster*: 229622) *Cluster* da mediana ARACAJÚ-SE:

CAMPINA GRANDE-PB	CARUARU-PE	FEIRA DE SANTANA-BA	FORTALEZA-CE
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA
JOÃO PESSOA-PB	JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN
NATAL-RN	PARNAÍBA-PI	PAULO AFONSO-BA	PETROLINA-PE
PICOS-PI	PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA
SÃO LUÍS-MA	SOBRAL-CE	TERESINA-PI	VALENÇA-BA
V. DA CONQUISTA-BA			

Total de cidades neste *cluster*: 25. Distância (custo) total neste *cluster*: 177453) *Cluster* da mediana CUIABÁ-MT:

BOA VISTA-RR	CAMPO GRANDE-MS	CORUMBÁ-MS	MANAUS-AM
PORTO VELHO-RO	RIO BRANCO-AC	RONDONÓPOLIS-MT	SINOP-MT
VILHENA-RO			

Total de cidades neste *cluster*: 9. Distância (custo) total neste *cluster*: 121854) *Cluster* da mediana GOIÂNIA-GO:

ANÁPOLIS-GO	BARREIRAS-BA	BELÉM-PA	BRASÍLIA-DF
CALDAS NOVAS-GO	IMPERATRIZ-MA	ITUMBIARA-GO	MONTES CLAROS-MG
PALMAS-TO	RIO VERDE-GO	S. MARIA DA VIT-BA	TUCURUÍ-PA

Total de cidades neste *cluster*: 12. Distância (custo) total neste *cluster*: 96275) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	ITAJAÍ-SC	JOINVILLE-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 6380

**Tabela A.5:** Resultado (5) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 70795

1) *Cluster* da mediana TERESINA-PI:

BELÉM-PA	FORTALEZA-CE	IMPERATRIZ-MA	JUAZEIRO DO NORTE-CE
PARNAÍBA-PI	PETROLINA-PE	PICOS-PI	SÃO LUÍS-MA
SOBRAL-CE	TUCURUÍ-PA		

Total de cidades neste *cluster*: 10. Distância (custo) total neste *cluster*: 62072) *Cluster* da mediana GOIÂNIA-GO:

ANÁPOLIS-GO	BARRETOS-SP	BARREIRAS-BA	BOA VISTA-RR
BRASÍLIA-DF	CALDAS NOVAS-GO	CAMPO GRANDE-MS	CORUMBÁ-MS
CUIABÁ-MT	ITUMBIARA-GO	MANAUS-AM	MONTES CLAROS-MG
PALMAS-TO	PATOS DE MINAS-MG	PORTO VELHO-RO	RIO BRANCO-AC
RIO VERDE-GO	RONDONÓPOLIS-MT	S. J. DO R. PRETO-SP	S. MARIA DA VIT-BA
SINOP-MT	UBERABA-MG	UBERLÂNDIA-MG	VILHENA-RO

Total de cidades neste *cluster*: 24. Distância (custo) total neste *cluster*: 264043) *Cluster* da mediana CURITIBA-PR:

BAGÉ-RS	BLUMENAU-SC	CASCAVEL-PR	CAXIAS DO SUL-RS
CHUIÍ-RS	CRICIÚMA-SC	DOURADOS-MS	FLORIANÓPOLIS-SC
FOZ DO IGUAÇÚ-PR	ITAJAÍ-SC	JOINVILLE-SC	LAGES-SC
LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR	NOVO HAMBURGO-RS
OURINHOS-SP	PARANAGUÁ-PR	PASSO FUNDO-RS	PELOTAS-RS
PONTA GROSSA-PR	PONTA PORÃ-MS	PORTO ALEGRE-RS	PRES. PRUDENTE-SP
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS

Total de cidades neste *cluster*: 28. Distância (custo) total neste *cluster*: 166454) *Cluster* da mediana S. J. DOS CAMPOS-SP:

AMERICANA-SP	ARAÇATUBA-SP	ARARAQUARA-SP	BARBACENA-MG
BAURU-SP	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES	CAMPINAS-SP
CAMPOS DOS GOYTAC-RJ	FRANCA-SP	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	JUIZ DE FORA-MG	LIMEIRA-SP
PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG	RIBEIRÃO PRETO-SP	RIO DE JANEIRO-RJ
SANTOS-SP	SÃO CARLOS-SP	SÃO LOURENÇO-MG	SÃO PAULO-SP
SOROCABA-SP	TAUBATÉ-SP	VITÓRIA-ES	VOLTA REDONDA-RJ

Total de cidades neste *cluster*: 28. Distância (custo) total neste *cluster*: 111835) *Cluster* da mediana MACEIÓ-AL:

ARACAJÚ-SE	CAMPINA GRANDE-PB	CARUARU-PE	FEIRA DE SANTANA-BA
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA
JOÃO PESSOA-PB	MOSSORÓ-RN	NATAL-RN	PAULO AFONSO-BA
PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA	VALENÇA-BA
V. DA CONQUISTA-BA			

Total de cidades neste *cluster*: 17. Distância (custo) total neste *cluster*: 10356



**Tabela A.6:** Resultado (6) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 71048

1) *Cluster* da mediana LONDRINA-PR:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BAURU-SP	BLUMENAU-SC	CALDAS NOVAS-GO
CAMPINAS-SP	CAMPO GRANDE-MS	CASCADEL-PR	CORUMBÁ-MS
CURITIBA-PR	DOURADOS-MS	FLORIANÓPOLIS-SC	FOZ DO IGUAÇÚ-PR
FRANCA-SP	GOIÂNIA-GO	ITAJAÍ-SC	ITUMBIARA-GO
JOINVILLE-SC	LIMEIRA-SP	MARÍLIA-SP	MARINGÁ-PR
OURINHOS-SP	PARANAGUÁ-PR	PONTA GROSSA-PR	PONTA PORÃ-MS
PRES. PRUDENTE-SP	RIBEIRÃO PRETO-SP	SÃO CARLOS-SP	S. J. DO R. PRETO-SP
SOROCABA-SP	UBERABA-MG	UBERLÂNDIA-MG	

Total de cidades neste *cluster*: 35. Distância (custo) total neste *cluster*: 182012) *Cluster* da mediana JUIZ DE FORA-MG:

BARBACENA-MG	BELO HORIZ-MG	BRASÍLIA-DF	C. DE ITAPEMIRIM-ES
CAMPOS DOS GOYTAC-RJ	GOV. VALADARES-MG	GUARATINGUETÁ-SP	IPATINGA-MG
ITABIRA-MG	MONTES CLAROS-MG	PATOS DE MINAS-MG	PETRÓPOLIS-RJ
POÇOS DE CALDAS-MG	RIO DE JANEIRO-RJ	SANTOS-SP	S. J. DOS CAMPOS-SP
SÃO LOURENÇO-MG	SÃO PAULO-SP	TAUBATÉ-SP	VITÓRIA-ES
VOLTA REDONDA-RJ			

Total de cidades neste *cluster*: 21. Distância (custo) total neste *cluster*: 86073) *Cluster* da mediana ARACAJÚ-SE:

BARREIRAS-BA	BELÉM-PA	CAMPINA GRANDE-PB	CARUARU-PE
FEIRA DE SANTANA-BA	FORTALEZA-CE	GUARANHUNS-PE	ILHÉUS-BA
IMPERATRIZ-MA	ITABUNA-BA	ITAPARICA-BA	JOÃO PESSOA-PB
JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN	NATAL-RN
PALMAS-TO	PARNAÍBA-PI	PAULO AFONSO-BA	PETROLINA-PE
PICOS-PI	PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA
SÃO LUÍS-MA	S. MARIA DA VIT-BA	SOBRAL-CE	TERESINA-PI
TUCURUÍ-PA	VALENÇA-BA	V. DA CONQUISTA-BA	

Total de cidades neste *cluster*: 31. Distância (custo) total neste *cluster*: 278554) *Cluster* da mediana CUIABÁ-MT:

BOA VISTA-RR	MANAUS-AM	PORTO VELHO-RO	RIO BRANCO-AC
RIO VERDE-GO	RONDONÓPOLIS-MT	SINOP-MT	VILHENA-RO

Total de cidades neste *cluster*: 8. Distância (custo) total neste *cluster*: 110995) *Cluster* da mediana PELOTAS-RS:

BAGÉ-RS	CAXIAS DO SUL-RS	CHUÍ-RS	CRICIÚMA-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PORTO ALEGRE-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS

Total de cidades neste *cluster*: 12. Distância (custo) total neste *cluster*: 5286

**Tabela A.7:** Resultado (7) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 71546

1) *Cluster* da mediana ARACAJÚ-SE:

CAMPINA GRANDE-PB	CARUARU-PE	FEIRA DE SANTANA-BA	GUARANHUNS-PE
ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA	JOÃO PESSOA-PB
MACEIÓ-AL	PAULO AFONSO-BA	PETROLINA-PE	PORTO SEGURO-BA
RECIFE-PE	SALVADOR-BA	VALENÇA-BA	V. DA CONQUISTA-BA

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 79652) *Cluster* da mediana GOIÂNIA-GO:

ANÁPOLIS-GO	BARRETOS-SP	BARREIRAS-BA	BOA VISTA-RR
BRASÍLIA-DF	CALDAS NOVAS-GO	CAMPO GRANDE-MS	CORUMBÁ-MS
CUIABÁ-MT	ITUMBIARA-GO	MANAUS-AM	MONTES CLAROS-MG
PALMAS-TO	PATOS DE MINAS-MG	PORTO VELHO-RO	RIO BRANCO-AC
RIO VERDE-GO	RONDONÓPOLIS-MT	S. J. DO R. PRETO-SP	S. MARIA DA VIT-BA
SINOP-MT	UBERABA-MG	UBERLÂNDIA-MG	VILHENA-RO

Total de cidades neste *cluster*: 24. Distância (custo) total neste *cluster*: 264043) *Cluster* da mediana FORTALEZA-CE:

BELÉM-PA	IMPERATRIZ-MA	JUAZEIRO DO NORTE-CE	MOSSORÓ-RN
NATAL-RN	PARNAÍBA-PI	PICOS-PI	SÃO LUÍS-MA
SOBRAL-CE	TERESINA-PI	TUCURUÍ-PA	

Total de cidades neste *cluster*: 11. Distância (custo) total neste *cluster*: 90554) *Cluster* da mediana SANTOS-SP:

AMERICANA-SP	ARAÇATUBA-SP	ARARAQUARA-SP	BARBACENA-MG
BAURU-SP	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES	CAMPINAS-SP
CAMPOS DOS GOYTAC-RJ	CURITIBA-PR	DOURADOS-MS	FRANCA-SP
GOV. VALADARES-MG	GUARATINGUETÁ-SP	IPATINGA-MG	ITABIRA-MG
JOINVILLE-SC	JUIZ DE FORA-MG	LIMEIRA-SP	LONDRINA-PR
MARÍLIA-SP	MARINGÁ-PR	OURINHOS-SP	PARANAGUÁ-PR
PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG	PONTA GROSSA-PR	PONTA PORÃ-MS
PRES. PRUDENTE-SP	RIBEIRÃO PRETO-SP	RIO DE JANEIRO-RJ	SÃO CARLOS-SP
S. J. DOS CAMPOS-SP	SÃO LOURENÇO-MG	SÃO PAULO-SP	SOROCABA-SP
TAUBATÉ-SP	VITÓRIA-ES	VOLTA REDONDA-RJ	

Total de cidades neste *cluster*: 39. Distância (custo) total neste *cluster*: 205115) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CASCADEL-PR	CAXIAS DO SUL-RS
CHUÍ-RS	CRICIÚMA-SC	FLORIANÓPOLIS-SC	FOZ DO IGUAÇÚ-PR
ITAJAÍ-SC	LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS
PELOTAS-RS	SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC
URUGUAIANA-RS			

Total de cidades neste *cluster*: 17. Distância (custo) total neste *cluster*: 7611

**Tabela A.8:** Resultado (8) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 73940

1) *Cluster* da mediana CAXIAS DO SUL-RS:

BAGÉ-RS	BLUMENAU-SC	CASCADEL-PR	CHUI-RS
CRICIÚMA-SC	CURITIBA-PR	FLORIANÓPOLIS-SC	FOZ DO IGUAÇU-PR
ITAJAÍ-SC	JOINVILLE-SC	LAGES-SC	MARINGÁ-PR
NOVO HAMBURGO-RS	PARANAGUÁ-PR	PASSO FUNDO-RS	PELOTAS-RS
PONTA GROSSA-PR	PORTO ALEGRE-RS	SANTA MARIA-RS	S. DO LIVRAMENTO-RS
TUBARÃO-SC	URUGUAIANA-RS		

Total de cidades neste *cluster*: 22. Distância (custo) total neste *cluster*: 114832) *Cluster* da mediana JUIZ DE FORA-MG:

AMERICANA-SP	BARBACENA-MG	BELO HORIZ-MG	C. DE ITAPEMIRIM-ES
CAMPINAS-SP	CAMPOS DOS GOYTAC-RJ	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	LIMEIRA-SP	MONTES CLAROS-MG
PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG	RIO DE JANEIRO-RJ	SANTOS-SP
SÃO CARLOS-SP	S. J. DOS CAMPOS-SP	SÃO LOURENÇO-MG	SÃO PAULO-SP
SOROCABA-SP	TAUBATÉ-SP	VITÓRIA-ES	VOLTA REDONDA-RJ

Total de cidades neste *cluster*: 24. Distância (custo) total neste *cluster*: 98433) *Cluster* da mediana GOIÂNIA-GO:

ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP	BARRETOS-SP
BARREIRAS-BA	BAURU-SP	BELÉM-PA	BRASÍLIA-DF
CALDAS NOVAS-GO	CORUMBÁ-MS	DOURADOS-MS	FRANCA-SP
IMPERATRIZ-MA	ITUMBIARA-GO	LONDRINA-PR	MARÍLIA-SP
OURINHOS-SP	PALMAS-TO	PATOS DE MINAS-MG	PONTA PORÃ-MS
PRES. PRUDENTE-SP	RIBEIRÃO PRETO-SP	RIO VERDE-GO	S. J. DO R. PRETO-SP
S. MARIA DA VIT-BA	SINOP-MT	TUCURUÍ-PA	UBERABA-MG
UBERLÂNDIA-MG	VILHENA-RO		

Total de cidades neste *cluster*: 30. Distância (custo) total neste *cluster*: 243664) *Cluster* da mediana ARACAJÚ-SE:

CAMPINA GRANDE-PB	CARUARU-PE	FEIRA DE SANTANA-BA	FORTALEZA-CE
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA
JOÃO PESSOA-PB	JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN
NATAL-RN	PARNAÍBA-PI	PAULO AFONSO-BA	PETROLINA-PE
PICOS-PI	PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA
SÃO LUÍS-MA	SOBRAL-CE	TERESINA-PI	VALENÇA-BA
V. DA CONQUISTA-BA			

Total de cidades neste *cluster*: 25. Distância (custo) total neste *cluster*: 177455) *Cluster* da mediana RONDONÓPOLIS-MT:

BOA VISTA-RR	CAMPO GRANDE-MS	CUIABÁ-MT	MANAUS-AM
PORTO VELHO-RO	RIO BRANCO-AC		

Total de cidades neste *cluster*: 6. Distância (custo) total neste *cluster*: 10503

**Tabela A.9:** Resultado (9) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 72485

1) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BARBACENA-MG	BAURU-SP	BELO HORIZ-MG
BRASÍLIA-DF	C. DE ITAPEMIRIM-ES	CALDAS NOVAS-GO	CAMPINAS-SP
CAMPOS DOS GOYTAC-RJ	CASCAVEL-PR	CORUMBÁ-MS	DOURADOS-MS
FRANCA-SP	GOIÂNIA-GO	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	ITUMBIARA-GO	JUIZ DE FORA-MG
LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR
MONTES CLAROS-MG	OURINHOS-SP	PATOS DE MINAS-MG	PETRÓPOLIS-RJ
POÇOS DE CALDAS-MG	PONTA GROSSA-PR	PONTA PORÃ-MS	PRES. PRUDENTE-SP
RIO DE JANEIRO-RJ	RIO VERDE-GO	SANTOS-SP	SÃO CARLOS-SP
S. J. DO R. PRETO-SP	S. J. DOS CAMPOS-SP	SÃO LOURENÇO-MG	SÃO PAULO-SP
SINOP-MT	SOROCABA-SP	TAUBATÉ-SP	UBERABA-MG
UBERLÂNDIA-MG	VILHENA-RO	VITÓRIA-ES	VOLTA REDONDA-RJ

Total de cidades neste *cluster*: 52. Distância (custo) total neste *cluster*: 300882) *Cluster* da mediana SALVADOR-BA:

ARACAJÚ-SE	BARREIRAS-BA	BELÉM-PA	FEIRA DE SANTANA-BA
ILHÉUS-BA	IMPERATRIZ-MA	ITABUNA-BA	ITAPARICA-BA
PALMAS-TO	PAULO AFONSO-BA	PETROLINA-PE	PICOS-PI
PORTO SEGURO-BA	SÃO LUÍS-MA	S. MARIA DA VIT-BA	TERESINA-PI
TUCURUÍ-PA	VALENÇA-BA	V. DA CONQUISTA-BA	

Total de cidades neste *cluster*: 19. Distância (custo) total neste *cluster*: 173533) *Cluster* da mediana NATAL-RN:

CAMPINA GRANDE-PB	CARUARU-PE	FORTALEZA-CE	GUARANHUNS-PE
JOÃO PESSOA-PB	JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN
PARNAÍBA-PI	RECIFE-PE	SOBRAL-CE	

Total de cidades neste *cluster*: 11. Distância (custo) total neste *cluster*: 55394) *Cluster* da mediana CAXIAS DO SUL-RS:

BAGÉ-RS	BLUMENAU-SC	CHUÍ-RS	CRICIÚMA-SC
CURITIBA-PR	FLORIANÓPOLIS-SC	FOZ DO IGUAÇÚ-PR	ITAJAÍ-SC
JOINVILLE-SC	LAGES-SC	NOVO HAMBURGO-RS	PARANAGUÁ-PR
PASSO FUNDO-RS	PELOTAS-RS	PORTO ALEGRE-RS	SANTA MARIA-RS
S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS	

Total de cidades neste *cluster*: 19. Distância (custo) total neste *cluster*: 90025) *Cluster* da mediana RONDONÓPOLIS-MT:

BOA VISTA-RR	CAMPO GRANDE-MS	CUIABÁ-MT	MANAUS-AM
PORTO VELHO-RO	RIO BRANCO-AC		

Total de cidades neste *cluster*: 6. Distância (custo) total neste *cluster*: 10503

**Tabela A.10:** Resultado (10) sem *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 75445

1) *Cluster* da mediana SANTOS-SP:

AMERICANA-SP	ARARAQUARA-SP	BAGÉ-RS	BARBACENA-MG
BAURU-SP	BLUMENAU-SC	C. DE ITAPEMIRIM-ES	CAMPINAS-SP
CAMPOS DOS GOYTAC-RJ	CASCAVEL-PR	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	CURITIBA-PR	FLORIANÓPOLIS-SC	FOZ DO IGUAÇÚ-PR
GUARATINGUETÁ-SP	ITAJAÍ-SC	JOINVILLE-SC	JUIZ DE FORA-MG
LAGES-SC	LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP
MARINGÁ-PR	NOVO HAMBURGO-RS	OURINHOS-SP	PARANAGUÁ-PR
PASSO FUNDO-RS	PELOTAS-RS	PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG
PONTA GROSSA-PR	PORTO ALEGRE-RS	RIO DE JANEIRO-RJ	SANTA MARIA-RS
S. DO LIVRAMENTO-RS	SÃO CARLOS-SP	S. J. DOS CAMPOS-SP	SÃO LOURENÇO-MG
SÃO PAULO-SP	SOROCABA-SP	TAUBATÉ-SP	TUBARÃO-SC
URUGUAIANA-RS	VITÓRIA-ES	VOLTA REDONDA-RJ	

Total de cidades neste *cluster*: 47. Distância (custo) total neste *cluster*: 335912) *Cluster* da mediana SALVADOR-BA:

ARACAJÚ-SE	BARREIRAS-BA	BELÉM-PA	FEIRA DE SANTANA-BA
ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA	PAULO AFONSO-BA
PETROLINA-PE	PICOS-PI	PORTO SEGURO-BA	SÃO LUÍS-MA
S. MARIA DA VIT-BA	TERESINA-PI	VALENÇA-BA	V. DA CONQUISTA-BA

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 116973) *Cluster* da mediana JOÃO PESSOA-PB:

CAMPINA GRANDE-PB	CARUARU-PE	FORTALEZA-CE	GUARANHUNS-PE
JUAZEIRO DO NORTE-CE	MACEIÓ-AL	MOSSORÓ-RN	NATAL-RN
PARNAÍBA-PI	RECIFE-PE	SOBRAL-CE	

Total de cidades neste *cluster*: 11. Distância (custo) total neste *cluster*: 52414) *Cluster* da mediana UBERLÂNDIA-MG:

ANÁPOLIS-GO	ARAÇATUBA-SP	BARRETOS-SP	BELO HORIZ-MG
BRASÍLIA-DF	CALDAS NOVAS-GO	CAMPO GRANDE-MS	CORUMBÁ-MS
CUIABÁ-MT	DOURADOS-MS	FRANCA-SP	GOIÂNIA-GO
GOV. VALADARES-MG	IMPERATRIZ-MA	IPATINGA-MG	ITABIRA-MG
ITUMBIARA-GO	MONTES CLAROS-MG	PALMAS-TO	PATOS DE MINAS-MG
PONTA PORÃ-MS	PRES. PRUDENTE-SP	RIBEIRÃO PRETO-SP	RIO VERDE-GO
RONDONÓPOLIS-MT	S. J. DO R. PRETO-SP	SINOP-MT	TUCURUÍ-PA
UBERABA-MG			

Total de cidades neste *cluster*: 29. Distância (custo) total neste *cluster*: 210815) *Cluster* da mediana PORTO VELHO-RO:

BOA VISTA-RR	MANAUS-AM	RIO BRANCO-AC	VILHENA-RO
--------------	-----------	---------------	------------

Total de cidades neste *cluster*: 4. Distância (custo) total neste *cluster*: 3835

**Tabela A.11:** 26 cidades elite, candidatas a medianas na fase de *path relinking*.

ARACAJÚ-SE	CAMPINAS-SP	CAMPO GRANDE-MS	CAXIAS DO SUL-RS
CUIABÁ-MT	CURITIBA-PR	FEIRA DE SANTANA-BA	FLORIANÓPOLIS-SC
FORTALEZA-CE	GOIÂNIA-GO	JOÃO PESSOA-PB	JUIZ DE FORA-MG
LONDRINA-PR	MACEIÓ-AL	NATAL-RN	PELOTAS-RS
PORTO ALEGRE-RS	PORTO VELHO-RO	RIBEIRÃO PRETO-SP	RONDONÓPOLIS-MT
SALVADOR-BA	SANTOS-SP	S. J. DOS CAMPOS-SP	SÃO PAULO-SP
TERESINA-PI	UBERLÂNDIA-MG		

**Tabela A.12:** Resultado com *path relinking*.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 62393

1) *Cluster* da mediana FEIRA DE SANTANA-BA:

ARACAJÚ-SE	BARREIRAS-BA	CAMPINA GRANDE-PB	CARUARU-PE
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA
JOÃO PESSOA-PB	MACEIÓ-AL	NATAL-RN	PAULO AFONSO-BA
PETROLINA-PE	PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA
S. MARIA DA VIT-BA	VALENÇA-BA	V. DA CONQUISTA-BA	

Total de cidades neste *cluster*: 19. Distância (custo) total neste *cluster*: 105572) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BARBACENA-MG	BAURU-SP	BELO HORIZ-MG
BRASÍLIA-DF	C. DE ITAPEMIRIM-ES	CALDAS NOVAS-GO	CAMPINAS-SP
CAMPO GRANDE-MS	CAMPOS DOS GOYTAC-RJ	CASCADEL-PR	CORUMBÁ-MS
CUIABÁ-MT	CURITIBA-PR	DOURADOS-MS	FOZ DO IGUAÇÚ-PR
FRANCA-SP	GOIÂNIA-GO	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	ITUMBIARA-GO	JUIZ DE FORA-MG
LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR
MONTES CLAROS-MG	OURINHOS-SP	PARANAGUÁ-PR	PATOS DE MINAS-MG
PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG	PONTA GROSSA-PR	PONTA PORÃ-MS
PRES. PRUDENTE-SP	RIO DE JANEIRO-RJ	RIO VERDE-GO	RONDONÓPOLIS-MT
SANTOS-SP	SÃO CARLOS-SP	S. J. DO R. PRETO-SP	S. J. DOS CAMPOS-SP
SÃO LOURENÇO-MG	SÃO PAULO-SP	SINOP-MT	SOROCABA-SP
TAUBATÉ-SP	UBERABA-MG	UBERLÂNDIA-MG	VITÓRIA-ES
VOLTA REDONDA-RJ			

Total de cidades neste *cluster*: 57. Distância (custo) total neste *cluster*: 337713) *Cluster* da mediana PORTO VELHO-RO:

BOA VISTA-RR	MANAUS-AM	RIO BRANCO-AC	VILHENA-RO
--------------	-----------	---------------	------------

Total de cidades neste *cluster*: 4. Distância (custo) total neste *cluster*: 38354) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	ITAJAÍ-SC	JOINVILLE-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 63805) *Cluster* da mediana TERESINA-PI:

BELÉM-PA	FORTALEZA-CE	IMPERATRIZ-MA	JUAZEIRO DO NORTE-CE
MOSSORÓ-RN	PALMAS-TO	PARNAÍBA-PI	PICOS-PI
SÃO LUÍS-MA	SOBRAL-CE	TUCURUÍ-PA	

Total de cidades neste *cluster*: 11. Distância (custo) total neste *cluster*: 7850

**Tabela A.13:** Resultado com *path relinking* após 200000 iterações, e 20000 iterações na solução de cada um dos problemas menores.

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 62393

1) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BARBACENA-MG	BAURU-SP	BELO HORIZ-MG
BRASÍLIA-DF	C. DE ITAPEMIRIM-ES	CALDAS NOVAS-GO	CAMPINAS-SP
CAMPO GRANDE-MS	CAMPOS DOS GOYTAC-RJ	CASCADEL-PR	CORUMBÁ-MS
CUIABÁ-MT	CURITIBA-PR	DOURADOS-MS	FOZ DO IGUAÇÚ-PR
FRANCA-SP	GOIÂNIA-GO	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	ITUMBIARA-GO	JUIZ DE FORA-MG
LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR
MONTES CLAROS-MG	OURINHOS-SP	PARANAGUÁ-PR	PATOS DE MINAS-MG
PETRÓPOLIS-RJ	POÇOS DE CALDAS-MG	PONTA GROSSA-PR	PONTA PORÃ-MG
PRES. PRUDENTE-SP	RIO DE JANEIRO-RJ	RIO VERDE-GO	RONDONÓPOLIS-MT
SANTOS-SP	SÃO CARLOS-SP	S. J. DO R. PRETO-SP	S. J. DOS CAMPOS-SP
SÃO LOURENÇO-MG	SÃO PAULO-SP	SINOP-MT	SOROCABA-SP
TAUBATÉ-SP	UBERABA-MG	UBERLÂNDIA-MG	VITÓRIA-ES
VOLTA REDONDA-RJ			

Total de cidades neste *cluster*: 57. Distância (custo) total neste *cluster*: 33771

2) *Cluster* da mediana PORTO VELHO-RO:

BOA VISTA-RR	MANAUS-AM	RIO BRANCO-AC	VILHENA-RO
--------------	-----------	---------------	------------

Total de cidades neste *cluster*: 4. Distância (custo) total neste *cluster*: 3835

3) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	ITAJAÍ-SC	JOINVILLE-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS

Total de cidades neste *cluster*: 16. Distância (custo) total neste *cluster*: 6380

4) *Cluster* da mediana TERESINA-PI:

BELÉM-PA	FORTALEZA-CE	IMPERATRIZ-MA	JUAZEIRO DO NORTE-CE
MOSSORÓ-RN	PALMAS-TO	PARNAÍBA-PI	PICOS-PI
SÃO LUÍS-MA	SOBRAL-CE	TUCURUÍ-PA	

Total de cidades neste *cluster*: 11. Distância (custo) total neste *cluster*: 7850

5) *Cluster* da mediana FEIRA DE SANTANA-BA:

ARACAJÚ-SE	BARREIRAS-BA	CAMPINA GRANDE-PB	CARUARU-PE
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	ITAPARICA-BA
JOÃO PESSOA-PB	MACEIÓ-AL	NATAL-RN	PAULO AFONSO-BA
PETROLINA-PE	PORTO SEGURO-BA	RECIFE-PE	SALVADOR-BA
S. MARIA DA VIT-BA	VALENÇA-BA	V. DA CONQUISTA-BA	

Total de cidades neste *cluster*: 19. Distância (custo) total neste *cluster*: 10557



# APÊNDICE B

## METAHEURÍSTICAS

### B.1 Introdução

Com a evolução da pesquisa sobre processos que resolvem problemas quantitativos, em particular problemas de otimização global, foram surgindo dificuldades que implicavam alto custo computacional. Tal fato decorria do modo de resolver as instâncias do problema, que procurava garantir a otimalidade da solução por métodos conhecidos como exatos. A persistência dessas dificuldades que se revelavam intransponíveis, provocou uma mudança de atitude que gerou nova vertente na busca de processos de solução. Essa nova vertente ou linha de pesquisa relaxou a garantia de otimalidade das soluções em troca de métodos mais amenos de resolução dos problemas. Tal atitude trouxe ganhos na medida em que se podia obter soluções de boa qualidade, eventualmente ótimas, para problemas de engenharia, em tempo computacional aceitável. Daí surgiram as heurísticas e, com sua evolução, as metaheurísticas. Essas se enquadraram na classe dos processos inteligentes que se distanciam dos

métodos de “força bruta” que produzem buscas exaustivas em espaços enormes de soluções viáveis.

O esforço de pesquisa tem mostrado as metaheurísticas como técnicas robustas de solução, ainda em processo de amadurecimento. Apesar de recente, a existência das metaheurísticas vem permitindo uma série de aplicações tecnológicas importantes que se incumbem de demonstrar o valor desses procedimentos.

Alguns consideram o termo metaheurística pouco adequado para denominar essa subárea da otimização estocástica — parte expressiva das metaheurísticas se baseia em processos aleatórios de geração de soluções, tão boas quanto possível, para problemas de alta complexidade.

Outra característica que acompanha o desenvolvimento das metaheurísticas é a inspiração ou motivação em processos naturais — como evolução e comportamento de espécies biológicas que se organizam em sociedade, colônias, enxames e cardumes, e processos termodinâmicos.

Metaheurísticas são métodos que articulam uma interação entre procedimentos de aprimoramento iterativo e estratégias sofisticadas para escapar de ótimos locais, e produzir uma busca robusta no espaço de soluções. Para resumir o conceito, pode-se aceitar a definição segundo a qual metaheurísticas são técnicas flexíveis que imitam processos naturais e intuitivos com metáforas físicas e biológicas, que procuram alcançar, a um custo computacional razoável, soluções de boa qualidade para problemas de elevada complexidade, sem, contudo, garantir a otimalidade, ou mesmo medir a proximidade do ótimo.

As principais metaheurísticas desenvolvidas e usadas até esse momento são: algoritmos genéticos, busca tabu, GRASP, *simulated annealing*, busca em vizinhança variável, colônia de formigas e busca dispersa, Michalewicz (2010), Talbi (2009), Yang (2008).

## B.2 Espaço de busca e busca local como alicerces das metaheurísticas

Em problemas de otimização uma lista de possíveis soluções, provavelmente uma lista infinita, é pesquisada com o intuito de encontrar a solução que melhor descreve

o problema em mãos. Imagine um espaço bi-dimensional em que o melhor valor de dois parâmetros,  $x$  e  $y$ , é procurado. Os pares  $(x, y)$  de melhor desempenho certamente estarão associados aos picos da superfície que representa a função desses pontos, e os piores pares  $(x, y)$  estão associados aos vales dessa mesma superfície. Essa lista de possíveis candidatos forma o espaço de busca, e o processo de busca do melhor par  $(x, y)$ , é conhecido como busca local. A ideia é facilmente estendida para dimensões maiores onde uma visualização geométrica do processo pode ficar impossível. As técnicas de busca local já registram uma história não muito recente. Seu grau de sofisticação vai desde algoritmos construtivos e de aprimoramento iterativo, até métodos mais complexos que requerem processos significativos de sintonia fina. Alguns consideram a busca local como, possivelmente, a forma mais natural de encontrar uma solução ótima ou sub-ótima para um problema de otimização.

Voltando à ilustração bi-dimensional, a busca local parte de um ponto  $(x_n, y_n)$  e tenta alcançar o ponto  $(x_{n+1}, y_{n+1})$ , seguindo a direção de um vetor  $\vec{v}$ .

Se a função  $f$  que descreve a superfície dos pontos  $(x, y)$  é uma função diferenciável, o gradiente de  $f$  em  $(x_n, y_n)$  pode fazer o papel do vetor  $\vec{v}$ , ou seja,  $\vec{v} = \nabla f(x_n, y_n)$ , dando a  $\vec{v}$  a direção de máximo crescimento a partir daquele ponto, ou de máximo decréscimo a partir daquele ponto, caso em que  $\vec{v} = -\nabla f(x_n, y_n)$ . Em qualquer caso a sequência de pontos, na direção dos picos ou vales, pode ser gerada por:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \lambda_n \vec{v}$$

em que  $\lambda_n \in \mathbb{R}$  é o tamanho do passo a ser dado entre os pontos.

No caso de um método de descida, se  $X$  é o espaço de busca, e  $f : X \rightarrow S$ , onde  $S$  é a superfície, representa a transformação de  $X$  em  $S$ , o algoritmo de busca local pode ser estabelecido pelos cinco passos do algoritmo 8, em que  $V(x)$  representa uma vizinhança de  $x$ .

---

#### **Algoritmo 8** Busca Local.

---

- 1 Escolha  $x \in X$  para iniciar o processo de busca
  - 2 Encontre  $x' \in V(x)$  tal que  $f(x') < f(x)$
  - 3 Se nenhum  $x'$  pode ser encontrado,  $x$  é o ótimo local, vá ao passo 5
  - 4 Caso contrário, designe  $x'$  como sendo o novo  $x$  e vá ao passo 2.
  - 5 **fim** *BL*
-

As metaheurísticas, cada uma com sua forma particular de gerar soluções e escapar de ótimos locais, do início ao fim do processo, usam os conceitos de espaço de busca e de busca local para desempenhar seu papel de encontrar soluções de boa qualidade para o problema em via de resolução.

### B.3 Algoritmos genéticos

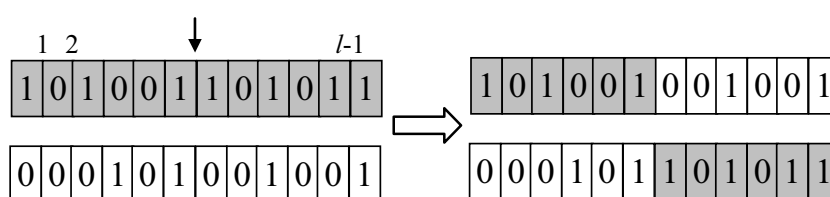
São metaheurísticas cujos fundamentos se expressam em termos de uma analogia com processos naturais de evolução. Ou ainda, são algoritmos de otimização numérica inspirados na seleção e na genética naturais. Partindo de uma população, os indivíduos com boa dotação genética têm maiores chances de sobreviver e produzir descendentes com maior grau de aptidão, enquanto que os de menor aptidão, tendem a desaparecer. Indo em direção a uma analogia, os procedimentos conhecidos como algoritmos genéticos, manipulam, a cada passo, populações de soluções. As populações (ou conjuntos) de soluções que se sucedem no processo, tendem a gerar indivíduos que são soluções progressivamente melhores para o problema ou aplicação que se tem em mãos. Aparecem aí os termos cromossomo e gene: o cromossomo — ou indivíduo da população — associa-se a uma solução do problema, e o gene é o nome que se dá a um componente da solução, Holland (1975b), Goldberg (1989), Luke (2011).

Em geral, os algoritmos genéticos são projetados e implementados com um de dois enfoques: um deles representa variáveis como strings binárias codificadas com 0's e 1's, enquanto que o outro trabalha com variáveis contínuas.

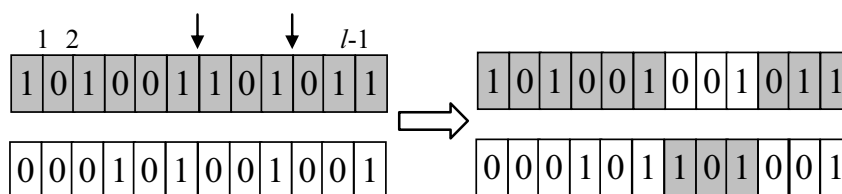
Um algoritmo genético típico ou usual é composto de:

- Uma população inicial de soluções — ou supostas soluções — para o problema
- Um método para avaliar o quanto uma solução individual é boa ou ruim
- Um processo para combinar fragmentos de soluções com vistas a formar novas — possivelmente melhores — soluções
- Um operador de mutação para garantir certo grau de diversidade por parte das soluções

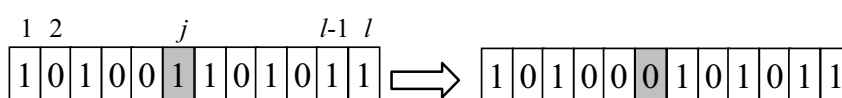
A Figura B.1 mostra uma operação de crossover em que dois cromossomos pais geram dois cromossomos filhos. Neste caso o crossover é de um único ponto. A Figura B.2 mostra uma operação de crossover com dois pontos, e a Figura B.3 mostra uma operação de mutação em que um único gene (ou bit) é trocado no cromossomo. Nas operações de crossover os cromossomos filhos herdam características dos cromossomos pais. Nas operações de mutação os cromossomos adquirem características próprias. Existe uma variedade razoável de formas já experimentadas para conduzir essas operações em contribuições diversas, como pode ser visto em Yu & Gen (2012), Sivanandam & Deepa (2007), Michalewicz (1992) e Sivanandam & Deepa (2007).



**Figura B.1:** Crossover de 1 ponto. Fonte: Yu & Gen (2012)



**Figura B.2:** Crossover de 2 pontos. Fonte: Yu & Gen (2012)



**Figura B.3:** Mutação: um único gene é trocado no cromossomo. Fonte: Yu & Gen (2012)

Um algoritmo genético inicia suas operações após a modelagem do problema. Analogamente a qualquer outro algoritmo de otimização, trabalha com a definição das variáveis de otimização, da função de custo, e das restrições associadas à aplicação para a qual ele foi projetado. Conclui suas operações, semelhante a qualquer outro algoritmo de otimização, pelo teste de convergência — número definido de iterações, tempo definido de processamento, ou pelo grau de satisfação produzido por alguma solução obtida durante o processo. O Algoritmo 9 mostra o pseudo-código para algoritmos genéticos.

---

**Algoritmo 9** Algoritmo genético.
 

---

```

executa: Populaçãotamanho, Problematamanho, Probcrossover, Probmutação
  // s* guarda a melhor solução obtida até então
  1 População ← InicializaPopulação(Populaçãotamanho, Problematamanho)
  2 AvaliaPopulação(População)
  3 s* ← ObtémMelhorSolução(População)
  4 enquanto Condição de parada não é satisfeita faça
  5   Pais ← SeleccionaPais(População, Populaçãotamanho)
  6   Filhos ← ∅
  7   para cada Pai1, Pai2 ∈ Pais faça
  8     Filho1, Filho2 ← Crossover(Pai1, Pai2, Probcrossover)
  9     Filhos ← Mutação(Filho1, Probmutação)
 10    Filhos ← Mutação(Filho2, Probmutação)
 11  fim para
 12  AvaliaPopulação(Filhos)
 13  s* ← ObtémMelhorSolução(Filhos)
 14  População ← Substitui(População, Filhos)
 15 fim enquanto
 16 retornar s*
 17 fim AG

```

---

## B.4 Busca tabu

A busca tabu é uma metaheurística equipada com a capacidade de explorar formas e estratégias de memória adaptativa<sup>1</sup>, que permitem a implementação de procedimentos capazes de executar uma busca no espaço de soluções de forma econômica e eficiente. O objetivo do algoritmo é o de restringir o retorno a áreas já visitadas da região de busca, visando escapar de ótimos locais. A estratégia do enfoque procura manter uma memória de curto prazo que retém mudanças a partir de movimentos recentes dentro do espaço de soluções, evitando movimentos futuros que, possivelmente, desfaçam essas mudanças através de movimentos reversos. O termo tabu lembra algo proibido e, no contexto do algoritmo, ele denomina uma lista (lista tabu) de movimentos proibidos durante algum tempo, no decorrer da busca, conforme Glover & Laguna (1993).

---

<sup>1</sup>“Memória adaptativa” é termo herdado da Psicologia. No caso humano, designa sistemas de memória sintonizados para lembrar informações que são processadas com a finalidade de sobrevivência ou, possivelmente, como resultado de ajustamento a vantagens que se acumulam a partir do passado ancestral, <http://pss.sagepub.com/content/19/2/176>, acessado em 20/3/2013.

As estruturas de memória na busca tabu que podem ser usadas para identificar elementos comuns a boas soluções ou a caminhos que levam a tais soluções, operam pela referência a quatro dimensões principais. Estas são formadas do “caráter recente” das soluções — permitindo tratar memórias de curto e longo prazos — da frequência com que as soluções reaparecem no processo, da qualidade e da influência na geração de futuras soluções. A busca tabu pode ser vista como meta-estratégia de solução de problemas combinatórios com meios para guiar enfoques de busca local que tenham capacidade de evitar que o procedimento fique preso a ótimos locais.

O princípio básico de qualquer iteração guiada pela busca tabu para resolver um problema de otimização pode ser declarado como segue: toma-se a solução atual que se tem em mãos e cria-se em torno dela uma vizinhança. A partir daí, tenta-se guiar a busca em uma direção adequada, usando inteligentemente o conhecimento ou memória das iterações anteriores. Isto é, o histórico da busca é usado para incorporar uma memória adaptativa e uma exploração responsiva na resolução do problema.

A transição de uma solução para outra na sua vizinhança é chamada de movimento. Dependendo da aplicação, esse termo abstrato, movimento, deve ser definido com precisão e objetividade, deixando claro como o algoritmo o executa. Por exemplo, se for considerado o vetor binário (0110110001) como um candidato a solução de determinado problema, uma vizinhança desta solução pode ser formada com todos os vetores binários viáveis que diferem deste pela troca de um único bit — de zero para 1, ou de 1 para zero. Um movimento poderia ser, portanto, partir do vetor (0110110001) e ir em direção a outro vetor de sua vizinhança — por exemplo, na direção do vetor (1110110001), se este não estiver preso na lista tabu.

O critério de escolha do melhor vizinho é utilizado para escapar de um ótimo local, de forma a superar a otimalidade local e, possivelmente, atingir um resultado ótimo ou próximo ao ótimo global. No entanto, isto pode fazer com que o algoritmo cicle<sup>2</sup>. Para evitar a ciclagem, a busca tabu utiliza estruturas flexíveis de memória para armazenar as soluções geradas (ou características destas). Existe uma lista tabu  $L_T$  que contém as  $|L_T|$  soluções decorrentes de movimentos realizados. Ela funciona como uma fila

---

<sup>2</sup>Ciclar: possibilidade do algoritmo retornar a solução já encontrada e reproduzir a mesma sequência de soluções, voltando ao ponto inicial da sequência.

de tamanho fixo. Assim, na exploração de um subconjunto  $U$  da vizinhança  $V(s)$  da solução corrente  $s$ , ficam excluídos da busca os vizinhos  $s'$  que são obtidos de  $s$  por movimentos  $m$  associados a soluções que constam na lista tabu.

Movimentos que, ao final, conduzem a uma solução previamente visitada, podem provocar o armazenamento dessa solução já visitada, na lista tabu e lá permanecer como proibida de ser revisitada por um certo número de iterações, o que é chamado de “tabu tenure,” ou seja, o tempo ou número de iterações em que o movimento fica proibido, preso à lista tabu.

As técnicas de busca tabu implementam dois conceitos que potencializam seu uso, melhorando seu desempenho e a qualidade das soluções. Tratam-se da “intensificação” e da “diversificação”. Intensificar implica em retornar a soluções já visitadas, de alguma forma promissoras, e explorar sua vizinhança de forma mais ampla, detalhada e efetiva. Além disso, o conceito se refere também à incorporação dos atributos das melhores soluções encontradas no decorrer do procedimento de busca.

A diversificação decorre do redirecionamento da pesquisa para regiões do espaço de soluções que, por algum critério ou medida, foram consideradas promissoras e ainda não foram satisfatoriamente exploradas. Com essa técnica a atenção se volta para gerar soluções com atributos diferentes dos que foram encontrados nas melhores soluções alcançadas até aqui.

Informações mais amplas, detalhadas e profundas sobre busca tabu, podem ser encontradas em Brownlee (2011), Glover (1989), Glover (1990a), Glover (1990b).

O Algoritmo 10 apresenta o pseudo-código da busca tabu.



---

**Algoritmo 10** Busca tabu.
 

---

**executa:**  $BT(f(\square), N(\square), A(\square), |V|, f_{\min}, |T|, BTmax, s)$

- 1  $s^* \leftarrow s$  // Melhor solução obtida até então
- 2  $Iter \leftarrow 0$  // Contador do número de iterações
- 3  $MelhorIter \leftarrow 0$  // Iteração mais recente que forneceu  $s^*$
- 4  $T \leftarrow \emptyset$  // Lista tabu  
// Inicializar a função de aspiração  $A$
- 5 **enquanto**  $(f_s > f_{\min}$  e  $Iter - MelhorIter < BTmax)$  **faça**
- 6  $Iter = Iter + 1$
- 7 Seja  $s' \leftarrow s \oplus m$  o melhor elemento de  $V \subset N(s)$  tal que o movimento  $m$  não seja tabu,  $(m \notin T)$  ou  $s'$  atenda a condição de aspiração  $f(s') < A(f(s))$
- 8 Atualizar a lista tabu
- 9  $s \leftarrow s'$
- 10 **se**  $(f(s) < f(s^*))$  **então**
- 11  $s^* \leftarrow s$
- 12  $MelhorIter \leftarrow Iter$
- 13 **fim se**  
// Atualizar a função de aspiração  $A$
- 14 **fim enquanto**
- 15  $s \leftarrow s^*$
- 16 retornar  $s$
- 17 **fim BT**

---

## B.5 Simulated annealing

*Simulated annealing* é um procedimento de busca local aleatória que imita o processo termodinâmico<sup>3</sup> de “*annealing*”<sup>4</sup>. No entendimento mais simples, tal processo consiste em aquecer um material, diga-se metálico, a uma elevada temperatura. Em seguida o metal é esfriado, de forma rápida ou lentamente. Os dois processos de esfriamento podem conduzir a produtos finais diferentes. Possivelmente, um mais quebradiço, enquanto que o outro se torna mais flexível. Em outras palavras, o *annealing* é o processo de submissão de um sólido a alta temperatura e, na sequência, submetido ao esfriamento, de modo a obter um produto final de alta qualidade, em estado conhecido

---

<sup>3</sup>Termodinâmica é o ramo da Física que estuda causas e efeitos de mudanças na temperatura, pressão e volume.

<sup>4</sup>O *annealing* ocorre pela difusão de átomos em um sólido usando calor. O papel do calor é o de aumentar a taxa de difusão provendo energia necessária à quebra de ligações. O processo como um todo tende a alcançar um estado de equilíbrio, ou de energia mínima. <http://en.wikipedia.org/wiki/Annealing>, acessado em 05/04/2013

como cristalino. *Simulated annealing* emula o processo físico de *annealing*. Durante o processo de esfriamento, imagina-se que as condições de equilíbrio térmico são mantidas. O processo de esfriamento termina quando o material atinge um estado de energia mínimo, que, em princípio, corresponde a um cristal perfeito, que é mais propenso a ser formado no âmbito de um esfriamento lento.

Pesquisas independentes observaram as semelhanças entre o processo físico de *annealing* e alguns problemas de otimização combinatória. Por essas observações há uma correspondência entre os estados físicos alternativos da matéria e o espaço de soluções de um problema de otimização. Além disso, foi observada uma correspondência entre a função objetivo de um problema de otimização e a energia livre do material em processo de *annealing*. O produto final com defeitos associa-se a um ótimo local, e o produto final perfeito tende a associar-se a um ótimo global.

No processo real de *annealing* o tempo com que o esfriamento ocorre é um dado importante que permite controlar a qualidade do produto final pela busca do estado cristalino, ou de mínima energia, de modo a reduzir defeitos na formação do metal. A metaheurística *simulated annealing* imita esse processo criando um parâmetro  $T$  chamado de “temperatura”. Esse parâmetro é iniciado com um valor  $T_0$  chamado de “temperatura inicial”. O uso de *simulated annealing* na otimização foi iniciado por Kirkpatrick et al. (1983).

São elementos críticos para o sucesso de implementações práticas do *simulated annealing*, a codificação de soluções, a definição de vizinhança das soluções e a função de avaliação.

Experimentos extensivos mostraram que o *simulated annealing* converge para o ótimo global se for usada aleatoriedade suficiente, combinada com um processo de esfriamento muito lento — que se dá pela queda muito vagarosa do valor do parâmetro  $T$ .

*Simulated annealing*, vista como uma metáfora, equivale a lançar algumas bolas sobre uma superfície irregular ou ambiente, com picos e vales e, enquanto as bolas quicam e perdem energia, elas se acomodam em algum mínimo local. Se for permitido às bolas quicarem um número suficiente de vezes e perderem energia de forma suficientemente lenta, algumas bolas possivelmente acabarão caindo num mínimo global,

representado por um dos vales mais profundos do ambiente, Yang (2008).

O princípio básico do algoritmo *simulated annealing* para escapar de ótimos locais, consiste em aceitar dois tipos de movimentos executados no espaço de soluções. Os movimentos associados a soluções produzidas por busca aleatória que aprimoram o processo de otimização, são aceitos com probabilidade 1. Além desses, os movimentos associados a soluções piores — aqueles que deterioram o processo de otimização — são também aceitos mas somente quando uma probabilidade gerada,  $\sigma$ , de aceitação, fica abaixo de um limiar determinado por outra probabilidade  $p$ . Essa probabilidade limiar  $p$  decresce junto com o decrescimento do parâmetro temperatura  $T$ , dificultando progressivamente a aceitação de soluções piores. Assim, uma solução pior só é aceita se  $\sigma < p$ , em que  $p$  decresce durante o processo em decorrência da queda da temperatura  $T$ .

Para descrever esse processo com um pouco mais de formalidade, representa-se por  $p$  a probabilidade limiar, também chamada de probabilidade de transição. O valor dessa probabilidade é dado por

$$p = e^{-\frac{\Delta E}{kT}}$$

onde  $k$  é conhecido como constante de Boltzman,  $T$  é a temperatura para controlar o processo de *annealing*, e  $\Delta E$  representa a mudança do nível de energia.

Considera-se um problema em que se minimiza uma função  $f$ . Para refletir uma mudança no nível de energia  $E$ , decorrente de uma mudança no valor de  $f$ , isto é,  $\Delta f$  provocar  $\Delta E$ , faz-se

$$\Delta E = \gamma \Delta f$$

onde  $\gamma \in \mathbb{R}$  é uma constante de proporcionalidade. Para simplificar, pode-se fazer  $k = 1$  e  $\gamma = 1$ , o que torna a probabilidade  $p$  igual a  $e^{-\frac{\Delta f}{T}}$ , ou seja,

$$p(\Delta f, T) = e^{-\frac{\Delta f}{T}}$$

A decisão de aceitar ou não a solução, decorrente de mudança provocada por um movimento no espaço de soluções, é tomada com base nos casos seguintes em que se considera  $\sigma$  como probabilidade de aceitação.

- Se  $\Delta f \leq 0$ , caso em que não houve piora em  $f$  — possivelmente houve melhora em  $f$  — a nova solução decorrente do movimento, é aceita
- Se  $\Delta f > 0$ , caso em que houve piora em  $f$ , a nova solução decorrente do movimento, é também aceita, desde que  $\sigma < p = e^{-\frac{\Delta f}{T}}$ .  $\sigma \in [0, 1]$  é uma probabilidade gerada

Se  $T$  é muito grande para uma dada mudança  $\Delta f$ , a maioria das soluções será aceita ( $T \rightarrow \infty$ , então  $p \rightarrow 1$ ). Se  $T$  é muito pequeno, então qualquer  $\Delta f > 0$  fará com que a solução raramente seja aceita ( $T \rightarrow 0$ , então  $p \rightarrow 0$ ).

Ao finalizar este tópico, é oportuno um rápido comentário sobre a paralelização desta metaheurística. O *simulated annealing* vem se revelando, ao longo de experimentos variados, como um procedimento guloso no que tange ao consumo de tempo computacional. Tal fato sugeriu que se adotasse esquemas paralelos<sup>5</sup> de processamento para essa metaheurística, como alternativa ao procedimento sequencial conhecido até então. Tal enfoque vem se mostrando, em algumas tentativas, como dependente do problema particular que se tem em mãos. Entretanto, algumas arquiteturas dedicadas de processadores de alto desempenho permitem antever que aplicações de grande porte poderiam ser tratadas com relativa eficiência por algoritmos de *simulated annealing*, Chandy & Banerjee (1996), Diekmann et al. (1993) e Diekmann et al. (1992)

O Algoritmo 11 apresenta o pseudo-código do *simulated annealing*.

---

<sup>5</sup> Paralelização de algoritmos constitui-se do particionamento do método através da sua divisão em tarefas independentes que podem ser atribuídas a processadores distintos.

---

**Algoritmo 11** *Simulated annealing.*


---

**executa:** SA( $f(\square)$ ,  $N(\square)$ ,  $\alpha$ ,  $SAmax$ ,  $T_0$ ,  $s$ )

- 1  $s^* \leftarrow s$  // Melhor solução obtida até então
- 2  $IterT \leftarrow 0$  // Número de iterações na temperatura T
- 3  $T \leftarrow T_0$  // Temperatura inicial
- 4 **enquanto** ( $T > 0$ ) **faça**
- 5   **enquanto** ( $IterT < SAmax$ ) **faça**
- 6      $IterT = IterT + 1$
- 7     Gerar um vizinho qualquer  $s' \in N(s)$
- 8      $\Delta f = f(s') - f(s)$
- 9     **se** ( $\Delta f \leq 0$ ) **então**
- 10        $s \leftarrow s'$
- 11       **se**  $f(s') < f(s^*)$  **então**
- 12           $s^* \leftarrow s'$
- 13       **senão**
- 14          Tome  $\sigma \in [0, 1]$
- 15          **se**  $\sigma < e^{-\frac{\Delta f}{T}}$  **então**
- 16            $s \leftarrow s'$
- 17       **fim se**
- 18     **fim se**
- 19   **fim se**
- 20 **fim enquanto**
- 21  $T \leftarrow \alpha \times T$
- 22  $IterT \leftarrow 0$
- 23 **fim enquanto**
- 24  $s \leftarrow s^*$
- 25 retornar  $s$
- 26 **fim SA**

---

## B.6 Busca em vizinhança variável (BVV)

A busca em vizinhança variável tem como estratégia a exploração de vizinhanças cada vez maiores de um dado ótimo local, até que um aprimoramento seja encontrado, após o que a busca através das vizinhanças é repetida. Trata-se de um método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança.

A busca em vizinhança variável não segue uma trajetória, mas sim explora vizinhanças mais distantes da solução atual e concentra a busca em torno de uma nova solução se, e somente se, um movimento de melhora é realizado.

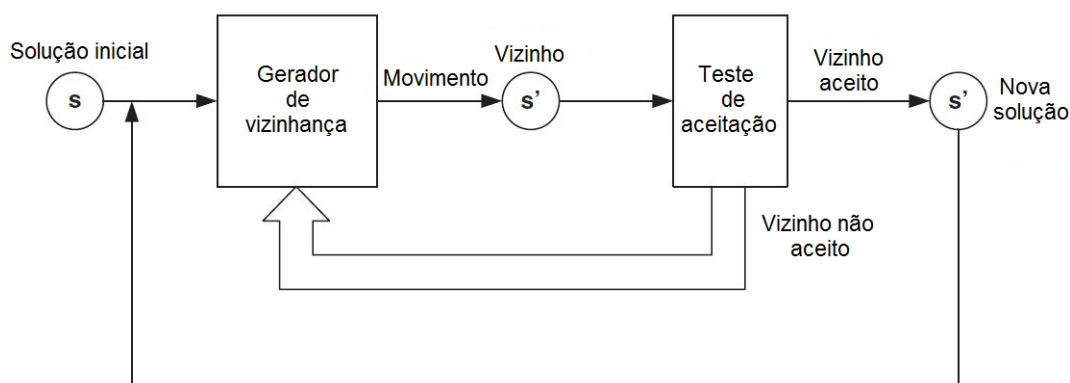
A ideia principal da VNS é baseada na observação de que um certo operador de vizinhança usado em uma busca local determinística, produz um ótimo local particular se aplicado a uma solução  $s$ . Ou seja, operadores de vizinhança<sup>6</sup> diferentes aplicados à mesma solução conduz a diferentes ótimos locais — os ótimos locais alcançados, a partir de uma solução  $s$ , são dependentes do operador de vizinhança usado. A VNS explora esta observação, empregando múltiplos operadores de vizinhança. Em outras palavras, a busca é iniciada com uma solução e um primeiro operador de vizinhança, veja o esquema da Figura B.4. Um vizinho aleatório da solução inicial é selecionado. Agora, uma busca local é aplicada à vizinhança gerada. Essa busca termina com uma solução ótima local. Esta nova solução é aceita como nova solução ou é descartada. Se a nova solução não é aceita, o próximo operador de vizinhança é usado. O processo é repetido. No caso de aceitação da nova solução o operador de vizinhança é mudado para o primeiro operador usado, e o processo é reiniciado com a nova solução alcançada.

O processo como um todo termina com a execução da regra de parada. A VNS diversifica a busca permitindo que operadores de vizinhança distintos explorem regiões distintas do espaço de busca. Portanto, a VNS estende a busca local com a habilidade de diversificá-la pela troca do operador de vizinhança. Esse procedimento tem por base três princípios:

- um mínimo local para uma estrutura de vizinhança pode não ser um mínimo local para outra estrutura de vizinhança
- um mínimo global é um mínimo local para todas as possíveis estruturas de vizinhança
- mínimos locais estão relativamente próximos ao mínimo global para muitas classes de problemas.

---

<sup>6</sup> Metaheurísticas usam operadores de vizinhança para “ligar” soluções no espaço de busca. Assim, esse operador age sobre a dinâmica da busca e impacta a eficiência da metaheurística.



**Figura B.4:** Esquema da BVV.<sup>7</sup>

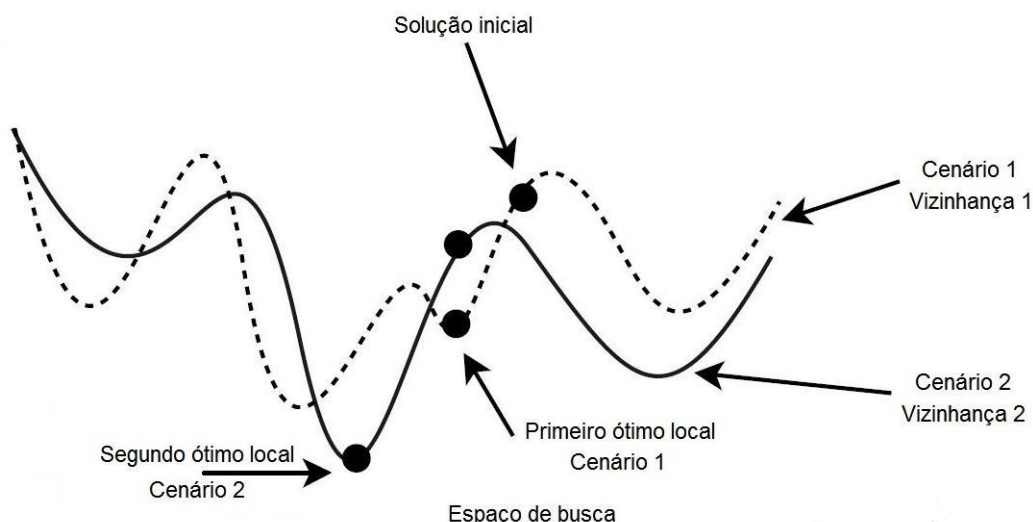
O algoritmo da VNS é baseado no procedimento de vizinhança variável de descida, que é uma versão determinística de VNS, representada pela sigla VND. A VND usa vizinhanças sucessivas de descida tentando alcançar um ótimo local. Para o algoritmo operar, primeiro define-se um conjunto de estruturas  $\mathcal{A}_k$  ( $k = 1, \dots, k_{max}$ ). Seja  $\mathcal{A}_1$  a primeira vizinhança a ser usada e  $s$  a solução inicial. Se um aprimoramento da solução  $s$  na sua vizinhança atual,  $\mathcal{A}_1(s)$ , não é possível, a estrutura de vizinhança é mudada de  $\mathcal{A}_1$  para  $\mathcal{A}_1 + 1$ . Se um aprimoramento da solução atual  $s$  é encontrado, a estrutura de vizinhança retorna à primeira,  $\mathcal{A}_1$ , para reiniciar a busca. Esta estratégia será eficiente se as diferentes vizinhanças usadas são complementares no sentido de que um ótimo local para uma vizinhança  $\mathcal{A}_i$  não é um ótimo local na vizinhança  $\mathcal{A}_j$ .

A ideia básica da VNS é explorar sucessivamente um conjunto de vizinhanças predefinidas para prover uma solução melhor. Ela pode explorar aleatoriamente ou sistematicamente um conjunto de vizinhanças para obter diferentes ótimos locais e para escapar do ótimo local. A VNS explora o fato de que o uso de várias vizinhanças na busca local pode gerar ótimos locais diferentes, e que o ótimo global é um ótimo local para uma dada vizinhança. Realmente, vizinhanças diferentes geram cenários diferentes.

A busca em vizinhança variável tem sido aplicada a um amplo arranjo de problemas de otimização combinatória e a outros como problemas de *clusterização* e problemas de otimização modelados com funções contínuas.

Um critério de parada muito utilizado para esse procedimento de busca local é definido por um número máximo de iterações sem aprimoramento.

<sup>7</sup> Esquema modificado a partir de Günther et al. (2010), lá construído para contexto semelhante.



**Figura B.5:** Ilustração do comportamento da BVV com duas estruturas de vizinhança. Fonte: Talbi (2009)

Aprofundamentos e variação da busca em vizinhança variável podem ser encontrados em Brownlee (2011), Günther et al. (2010) e Talbi (2009).

A Figura B.5 ilustra o comportamento da VNS com dois cenários e duas estruturas de vizinhança.

O Algoritmo 12 apresenta o pseudo-código da busca em vizinhança variável.

---

**Algoritmo 12** Busca em vizinhança variável.

---

**executa:** Vizinhanças

- 1  $S_{melhor} \leftarrow \text{SoluçãoAleatória}()$
  - 2 **enquanto** Condição de parada não é satisfeita **faça**
  - 3   **para cada**  $Vizinhança_i \in \text{Vizinhanças}$  **faça**
  - 4      $Vizinhança_{atual} \leftarrow \text{CalculaVizinhança}(S_{melhor}, Vizinhança_i)$
  - 5      $S_{candidata} \leftarrow \text{SoluçãoAleatóriaNaVizinhança}(Vizinhança_{atual})$
  - 6      $S_{candidata} \leftarrow \text{BuscaLocal}(S_{candidata})$
  - 7     **se**  $\text{Custo}(S_{candidata}) < \text{Custo}(S_{melhor})$  **então**
  - 8        $S_{melhor} \leftarrow S_{candidata}$
  - 9     Interrompe o **para cada**
  - 10   **fim se**
  - 11   **fim para**
  - 12 **fim enquanto**
  - 13 retorna  $S_{melhor}$
  - 14 **fim BVV**
-



## B.7 Colônia de formigas

“Colônia de formigas” é uma metáfora aqui tratada como fundamento para algoritmos ou sistemas computacionais inspirados pela inteligência coletiva que emerge da cooperação de um grande número de agentes homogêneos no ambiente. Cardumes, bandos, enxames e, em particular, colônias de formigas, são exemplos de agrupamentos desses agentes. Essa inteligência coletiva é descentralizada, distribuída e com capacidade de auto-organização no ambiente em que ela existe. Algum tipo de informação — de natureza química, de movimentos ou de proximidade — circula, ou está presente no meio frequentado por esses agentes. Por exemplo, as formigas se comunicam através de marcas químicas — feromônio — deixadas no chão. Ao passar por um local uma formiga deixa uma certa quantidade de feromônio, formando trilhas. Outras formigas podem detectar tais trilhas com feromônio, e tendem a escolher seu caminho por trilhas com mais feromônio. O elenco de procedimentos baseados nessa metáfora estabelece outra classe de memória para armazenar informação, como será visto a seguir.

A busca tabu foi o primeiro contato feito aqui com conceitos que envolvem processos de memória em otimização. A partir da busca tabu sabe-se que é possível aprimorar a eficiência da busca usando um tipo de memória que mantém registros históricos para serem usados nas iterações posteriores do método. É uma forma de combinar aleatoriedade com memória. Agora são introduzidos outros conceitos que também combinam soluções aleatórias com processos de memória e que permitem aumentar a diversidade de soluções com vistas a escapar de ótimos locais. A espécie de memória que aqui será tratada é diferente do tipo daquela que guarda registros históricos do processo de busca. Trata-se de outra forma de memória que registra mensagens químicas, tais como as informações transmitidas por substância que formigas, abelhas e outros insetos sociais depositam nas trilhas e ambientes por eles usados.

Parte da metáfora em que se baseia a metaheurística colônia de formigas está no ato de depósito do feromônio pelas formigas nas trilhas que elas percorrem entre o formigueiro e a fonte de alimento. Outra parte dessa metáfora ocorre de acordo com a descrição a seguir. Imagina-se um formigueiro e uma fonte de alimento localizada

em sua vizinhança. Entre o formigueiro e a fonte de alimento existe um obstáculo que as formigas devem ultrapassar, onde um caminho em torno do obstáculo é mais curto que outro. Experimentos foram realizados com vistas a estudar com precisão o comportamento de formigas em tal situação. De início as formigas não tinham informação sobre o ambiente e a localização da fonte de alimento. Após algum tempo foi observado que elas não só encontraram a fonte de alimento mas também passaram a utilizar o caminho mais curto para alcançá-la. Tais observações se estenderam a ambientes mais complexos, com caminhos múltiplos e, mesmo assim, as formigas ainda tiveram a habilidade de descobrir a fonte e o caminho mais curto. A pergunta natural que surge agora é: como as formigas agiram sem qualquer conhecimento prévio a respeito do seu ambiente para chegar a esse resultado?

Formigas são insetos sociais que vivem juntas em colônias organizadas em que o tamanho da população pode variar de 2 a 25 milhões de indivíduos. Colônias de formigas, são sistemas distribuídos que, a despeito da simplicidade de seus indivíduos, apresentam uma estrutura social altamente organizada. Como resultado dessa organização, colônias de formigas podem desempenhar tarefas complexas que, em alguns casos, excedem em muito as capacidades de uma formiga individual. Quando procuram ou buscam alimento, um enxame de formigas ou de agentes móveis, interage ou se comunica em seu ambiente local. Cada formiga pode depositar feromônio para se comunicar com as outras, e cada formiga está apta também a seguir a rota marcada com o feromônio deixado por outras formigas. Quando as formigas encontram a fonte de alimento elas a marcam com feromônio, assim como sua trilha a partir da fonte. Assim, quanto mais formigas passarem por uma trilha, mais feromônio esta trilha terá. Este comportamento simples de seguir trilhas, faz emergir um comportamento mais complexo de encontrar trilhas mais curtas entre dois pontos. A partir da rota aleatória de busca por alimento, a concentração de feromônio varia e as formigas seguem a rota de mais alta concentração de feromônio que continua sendo aprimorada por um número crescente de formigas. Enquanto mais e mais formigas seguem a mesma rota, ela vai se tornando a trilha favorita. Assim, alguma trilha favorita (frequentemente a mais curta ou eficiente) emerge. Trata-se de um mecanismo de retroalimentação. Embora não exista uma formiga mestre supervisionando toda a colônia e irradiando instruções para as formigas individuais, um comportamento organizado emerge automaticamente. Portanto, um tal comportamento emergente é

semelhante a outros fenômenos auto-organizados que ocorrem em muitos processos na natureza, tais como o padrão de formação da pele dos animais, dos sinais sonoros por eles emitidos e da forma de se movimentarem.

Baseado nas características de comportamento das formigas, os cientistas desenvolveram vários algoritmos poderosos baseados em colônia de formigas, com importante progresso feito nos anos recentes. De fato, usando somente alguns comportamentos de formigas e adicionando algumas novas características, foi projetada uma classe de novos algoritmos. Dois itens importantes que aparecem na construção desses algoritmos são: a probabilidade de escolher uma trilha, e a taxa de evaporação do feromônio. Trata-se de uma área atual e ativa de pesquisa, embora não sejam muito variadas as formas de resolver os problemas subjacentes. Por exemplo, para um problema de roteamento em rede, a probabilidade de formigas escolherem, em um nó  $i$  particular, a trilha a partir do nó  $i$  ao nó  $j$  é dada por

$$p_{ij} = \frac{\phi_{ij}^{\alpha} d_{ij}^{\beta}}{\sum_{i=1}^n \sum_{j=1}^n \phi_{ij}^{\alpha} d_{ij}^{\beta}} \quad (2.1)$$

onde  $\alpha > 0$  e  $\beta > 0$  são parâmetros de influência, e seus valores típicos são  $\alpha \approx \beta \approx 2$ .  $\phi_{ij}$  é a concentração de feromônio na trilha entre  $i$  e  $j$ , e  $d_{ij}$  traduz o desejo de percorrer a mesma trilha entre  $i$  e  $j$ .

Algum conhecimento prévio sobre a trilha, tal como a distância  $s_{ij}$ , é usado com frequência, de modo que  $d_{ij} \propto \frac{1}{s_{ij}}$ , implicando que trilhas mais curtas serão selecionadas preferencialmente devido ao tempo de percurso ser menor, e, assim, a concentração de feromônio é mais alta. Esta probabilidade reflete o fato de que formigas normalmente seguiriam as trilhas com maior concentração de feromônio. No caso mais simples, quando  $\alpha = \beta = 1$ , a probabilidade das formigas escolherem um caminho é proporcional à concentração de feromônio na trilha. O denominador em 2.1 normaliza a probabilidade, de modo que ela fique no intervalo de zero a 1.

A concentração de feromônio pode mudar com o tempo devido à evaporação de feromônio. Além disso, a vantagem da evaporação do feromônio se traduz no fato de que ela pode evitar que o algoritmo fique preso a um ótimo local. Se não houvesse evaporação, o caminho escolhido aleatoriamente pelas primeiras formigas se torna-

ria a trilha preferida pelas outras formigas, com base na atração provocada pelo feromônio. Para uma taxa constante  $\gamma$  de decaimento ou evaporação, a concentração de feromônio usualmente varia com o tempo de forma exponencial

$$\phi(t) = \phi_0 e^{-\gamma t}$$

onde  $\phi_0$  é a concentração inicial de feromônio e  $t$  é o tempo. Se  $\gamma t \ll 1$ , então, tem-se que  $\phi(t) \approx (1 - \gamma t)\phi_0$ . Para um acréscimo unitário do tempo,  $\Delta t = 1$ , a evaporação pode ser aproximada por  $\phi^{t+1} \leftarrow (1 - \gamma)\phi^t$ . Portanto, tem-se a fórmula simplificada de atualização do feromônio dada por

$$\phi_{ij}^{t+1} = (1 - \gamma)\phi_{ij}^t + \delta\phi_{ij}^t \quad (2.2)$$

onde  $\gamma \in [0, 1]$  é a taxa de evaporação do feromônio. O incremento  $\delta\phi_{ij}^t$  é a quantidade de feromônio depositada no tempo  $t$  ao longo da trilha de  $i$  a  $j$  quando as formigas trafegam uma distância  $L$ . Usualmente,  $\delta\phi_{ij}^t \propto \frac{1}{L}$ . Se não existem formigas em uma trilha, o depósito de feromônio nessa trilha é zero.

Um esquema possível de aceleração pode ser construído a partir de alguns limites de concentração de feromônio e, somente às formigas com a melhor solução global atual é permitido depositar feromônio. Além disso, alguma classificação da aptidão de soluções pode também ser usada.

A otimização por colônia de formigas se dá por algoritmos probabilísticos inspirados pela estigmergia<sup>8</sup> e o comportamento de busca por alimentos das formigas.

A área que desenvolve “algoritmos baseados em colônia de formigas” estuda modelos derivados da observação do comportamento real das formigas, e usa esses modelos como fonte de inspiração para o projeto de novos algoritmos visando a aplicação a problemas de otimização e de controle distribuído.

---

<sup>8</sup> Estigmergia é o conceito definido a partir do princípio de que a informação deixada no ambiente por qualquer ação estimula o desempenho de outra ação, pelo mesmo agente, ou por agentes distintos. É um método de transmissão de informações em que as partes individuais do sistema se comunicam umas com as outras pela modificação de seu ambiente local. Trata-se de um mecanismo de coordenação indireta entre agentes ou ações. Estigmergia foi inicialmente observada na natureza — formigas se comunicam, umas com as outras, pelo depósito de feromônio ao longo da trilha que elas percorrem, de modo que onde as formigas vão, dentro e em torno de sua colônia, existe um sistema estigmérgico.

Biólogos mostraram que muitos comportamentos agregados da colônia, observados em insetos sociais, podem ser explicados via modelos simples em que somente a comunicação estigmérgica está presente. Em outras palavras, os biólogos mostraram que frequentemente é suficiente considerar a comunicação indireta, estigmérgica, para explicar como insetos sociais podem alcançar a auto-organização. A ideia por detrás dos algoritmos de colônia de formigas é a de usar uma forma de “estigmergia artificial” para coordenar sociedades de agentes artificiais.

A faculdade visual perceptiva de muitas espécies de formigas foi desenvolvida somente de forma rudimentar e existem espécies que são completamente cegas. De fato, uma percepção importante da pesquisa inicial sobre o comportamento das formigas foi a de que grande parte da comunicação entre indivíduos, ou entre indivíduos e o ambiente, é baseada no uso de feromônio produzido pelas formigas. Isto é diferente, por exemplo, do que acontece com humanos e outras espécies, cujos sentidos mais importantes são a visão e a audição. Particularmente importante para a vida social de algumas espécies de formigas é o “feromônio de trilha”. O feromônio de trilha é um tipo específico de feromônio que algumas espécies de formigas, usam para fazer caminhos no chão. Por exemplo, caminhos que vão da fonte de alimento ao ninho. Seguindo a orientação do feromônio, as formigas, a procura de alimento, podem seguir o caminho para o alimento descoberto por outras formigas. Esse comportamento definido por “depositar-no-caminho” versus “seguir-pelo-caminho”, praticado pelas formigas por onde elas passam, seguindo o caminho influenciado pela trilha química deixada por outras formigas, é a fonte de inspiração para a “Otimização por colônia de formigas”.

Variações e aprofundamentos deste conteúdo podem ser encontrados em Brownlee (2011), Dorigo & Stützle (2004) e Deneubourg et al. (1990).

O Algoritmo 13 apresenta o pseudo-código da metaheurística colônia de formigas.

---

**Algoritmo 13** Colônia de formigas.
 

---

**executa:** População<sub>tamanho</sub>, Problema<sub>tamanho</sub>,  $m$ ,  $\rho$ ,  $\beta$ ,  $\sigma$ ,  $q_0$   
 1  $P_{melhor} \leftarrow \text{CriaSoluçãoHeurística}(\text{Problema}_{tamanho})$   
 2  $P_{melhor_{custo}} \leftarrow \text{Custo}(S_h)$   
 3 Feromônio<sub>inicial</sub>  $\leftarrow \frac{1.0}{\text{Problema}_{tamanho} \times P_{melhor_{custo}}}$   
 4 Feromônio  $\leftarrow \text{InicializaFeromônio}(\text{Feromônio}_{inicial})$   
 5 **enquanto** Condição de parada não é satisfeita **faça**  
 6   **para**  $i = 1$  a  $m$  **faça**  
 7      $S_i \leftarrow \text{ConstróiSolução}(\text{Feromônio}, \text{Problema}_{tamanho}, \beta, q_0)$   
 8      $S_{i_{custo}} \leftarrow \text{Custo}(S_i)$   
 9     **se**  $S_{i_{custo}} \leq P_{melhor_{custo}}$  **então**  
 10        $P_{melhor_{custo}} \leftarrow S_{i_{custo}}$   
 11        $P_{melhor} \leftarrow S_i$   
 12     **fim se**  
 13     AtualizaçãoLocal\_DecaimentoDeFeromônio(Feromônio,  $S_i$ ,  $S_{i_{custo}}$ ,  $\sigma$ )  
 14   **fim para**  
 15   AtualizaçãoGlobal\_DecaimentoDeFeromônio(Feromônio,  $P_{melhor}$ ,  
     $P_{melhor_{custo}}$ ,  $\rho$ )  
 16 **fim enquanto**  
 17 retornar  $P_{melhor}$   
 18 **fim CF**

---

## B.8 Busca dispersa

A primeira concepção de busca dispersa foi apresentada através de Glover (1974) e, inicialmente, recebeu pouca atenção. A busca dispersa ou *scatter search* é uma metaheurística de natureza evolucionária e populacional. A ideia básica da busca dispersa se concretiza a partir de um dado grupo de pontos em que o algoritmo, de alguma forma, encontra novos pontos, aceita os melhores e descarta os piores. A busca dispersa recombina soluções selecionadas a partir de um conjunto de referência — ConjRef — para gerar outras soluções, Laguna & Martí (2003). Essa metaheurística permite ver a importância de se controlar, deterministicamente, diversificação e intensificação em métodos de busca que se baseiam em combinação.

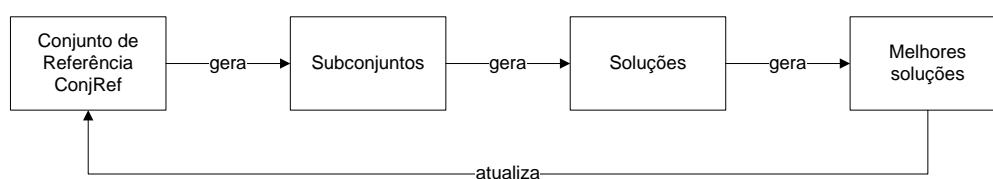
O ConjRef que é um instrumento fundamental na busca dispersa, tem a função de guardar as boas soluções que foram obtidas em passos anteriores dessa metaheurística. Uma solução é dita de boa qualidade não somente em função do valor objetivo

que ela produz, mas também por exibir características de diversidade.

A busca dispersa constitui-se de procedimentos para:

- Construir o ConjRef
- Gerar subconjuntos de soluções como forma de operar o ConjRef
- Diversificar soluções
- Executar melhoria/aprimoramento de soluções
- Combinar soluções conhecidas para obter novas soluções e atualizar o ConjRef.

Os procedimentos listados resumem o funcionamento da busca dispersa. Podem ser entendidos a partir de quatro passos básicos, conforme ilustra a Figura B.6. O ConjRef contém as  $\gamma$  melhores soluções,  $\gamma_1$  das quais são boas soluções em relação ao seu valor objetivo (ConjRef<sub>1</sub>) e  $\gamma_2$  das quais são boas soluções em relação à diversidade (ConjRef<sub>2</sub>). Portanto,  $\gamma = \gamma_1 + \gamma_2$ . As soluções iniciais da busca dispersa



**Figura B.6:** Esquema geral da busca dispersa. Fonte: Yu & Gen (2012)

mantém a máxima diversidade a partir da iniciativa de gerar soluções aleatoriamente numa dada área de tal maneira que, quanto menos oportunidade essa área tiver para gerar novos indivíduos, mais soluções são geradas nessa área. No passo seguinte à inicialização, a busca dispersa aprimora as soluções iniciais através de um procedimento de melhoria. Após este passo, são construídas duas versões do ConjRef: o ConjRef<sub>1</sub> seleciona seus elementos a partir das soluções aprimoradas, de acordo com o critério de qualidade — diga-se, o valor da função objetivo. ConjRef<sub>2</sub> é construído a partir das soluções aprimoradas restantes, de acordo com o critério de diversidade. Isto é, quanto mais “distantes” os elementos de ConjRef<sub>2</sub> estiverem de ConjRef<sub>1</sub>, melhor será. A partir daí, inicia-se o laço principal da busca dispersa. Primeiramente, o ConjRef é usado para gerar todos os subconjuntos de dois elementos contendo todas as combinações par a par das  $\gamma$  soluções do ConjRef. Em seguida, são gerados os subconjuntos:

- de três elementos derivados dos subconjuntos de dois elementos pela adição da melhor solução que não está neste subconjunto — medida pelo valor objetivo
- de quatro elementos derivados dos subconjuntos de três elementos pela adição da melhor solução que não está neste subconjunto — medida pelo valor objetivo
- contendo os  $\alpha$  melhores elementos — medidos pelo valor objetivo.

Com isso, apesar dos subconjuntos considerarem o valor objetivo como o critério mais importante, destacam ainda o fator diversidade como um item da busca dispersa. Numa contraposição aos algoritmos genéticos, que controlam indiretamente os indivíduos da população, recombinaando esses indivíduos, a busca dispersa controla explicitamente a diversidade da população através de métodos adequados para isso. A busca dispersa oferece um controle refinado de diversificação e intensificação. Adicionalmente, ela generaliza a ideia de crossover dos algoritmos genéticos, e permite a recombinação de subconjuntos que contém mais do que duas soluções. Embora a busca dispersa empregue estratégias diferentes para se tornar menos vulnerável a convergência prematura, ela não está isenta de convergir em regiões subótimas do espaço de busca.

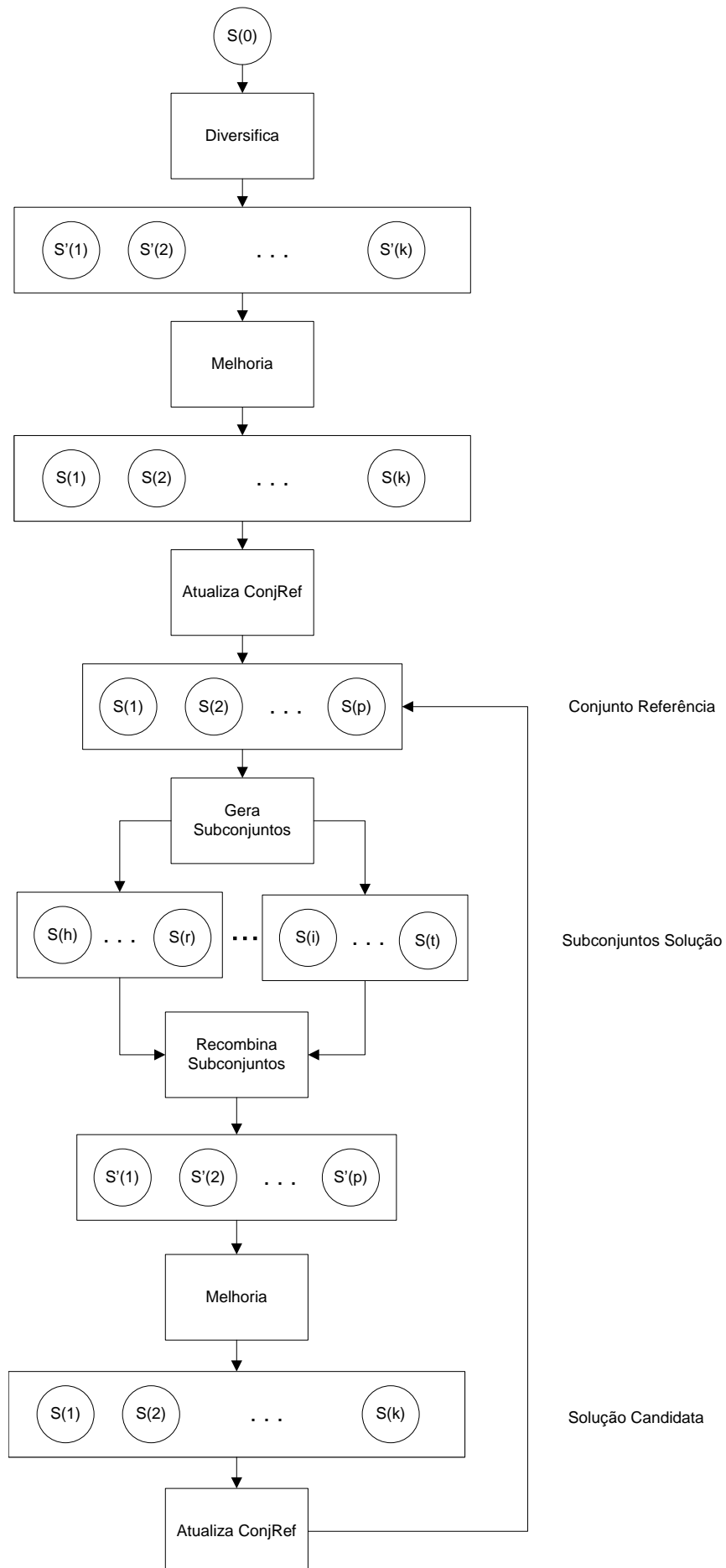
Uma implementação básica da busca dispersa pode ser conduzida usando os seguintes procedimentos:

- Diversificação
- Aprimoramento — ou melhoria
- Atualização do conjunto referência
- Geração de subconjuntos
- Recombinação de subconjuntos.

A Figura B.7 mostra o esquema geral da busca dispersa.

O algoritmo 14 formaliza a busca dispersa. É necessário um método para gerar uma solução inicial  $s_0$ . Além disso, precisa-se de um conjunto  $\mathcal{D}$  para manter as soluções diversificadas.





**Figura B.7:** Esquema da busca dispersa. Fonte: Adaptado de Günther et al. (2010)

---

**Algoritmo 14** Algoritmo de busca dispersa.

---

```
1  $\mathcal{D} \leftarrow \emptyset$ 
2  $s_0 \leftarrow$  Solução inicial
3  $\mathcal{D} \leftarrow$  Diversifica( $\mathcal{D}$ )
4  $\mathcal{D} \leftarrow$  Melhoria( $\mathcal{D}$ )
5 ConjRef  $\leftarrow$  AtualizaConjRef( $\mathcal{D}$ )
6 enquanto Condição de parada não é satisfeita faça
7    $\mathcal{D} \leftarrow \emptyset$ 
8   Subconjuntos  $\leftarrow$  Gerasubconjunto( $\mathcal{D}$ )
9    $\mathcal{D} \leftarrow$  Recombina(Subconjuntos)
10   $\mathcal{D} \leftarrow$  Melhoria( $\mathcal{D}$ )
11   $\mathcal{D} \leftarrow$  Avalia( $\mathcal{D}$ )
12  ConjRef  $\leftarrow$  AtualizaConjRef( $\mathcal{D} \cup$ ConjRef)
13 fim enquanto
14 retorna ConjRef
15 fim Busca Dispersa
```

---

## REFERÊNCIAS BIBLIOGRÁFICAS

- Amorim, F. M. S. (2011). Metaheurísticas aplicadas ao problema das  $p$ -medianas. Dissertação de Mestrado - CEFET-MG. 21
- Beasley, J. E. (1990). Or-library: distribution test problems by electronic mail. *Journal of Operational Research Society*, 41, 1069–1072. 54
- Bezerra, S. N. (2008). Algoritmos evolutivos paralelos aplicados ao problema de  $p$ -medianas. Dissertação de Mestrado - CEFET - MG. 16, 32
- Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews: the state of the art. *Computers & Operations Research*, 9, 63 – 212. 4
- Boudia, M., Louly, M. A. O., & Prins, C. (2007). A reactive grasp and path relinking for a combined production-distribution problem. *Computers & Operations*, 34, 3402 – 3419. 34, 41, 46
- Brownlee, J. (2011). *Clever Algorithms: Nature-Inspired Programming Recipes*. Lulu. 106, 114, 119
- Bueno, E. A. (2000). Redução de perdas na distribuição de energia elétrica pelo método grasp. Dissertação de Mestrado - Engenharia Elétrica - Universidade Estadual de Campinas - Campinas, SP. 30
- Caires, L. F. V., Assis, L. P., & Andrade, A. V. (2010). Aplicação do método path-relinking na resolução do problema roteamento de veículos com coleta e entrega simultâneas. XLII SBPO. 34, 41
- Cerny, V. (1985). A thermodynamical approach to the traveling salesman problem: an efficient solution algorithm. *Journal of Optimization Theory and Applications*, 45, 41–51. 33

- Chandy, J. A. & Banerjee, P. R. (1996). Parallel simulated annealing strategies for vlsi cell placement. In *In Proc. 9th International Conference on VLSI Design, Bangalore - India*. 110
- Chen, D., Batson, R., & Dang, Y. (2011). *Applied Integer Programming: Modeling and Solution*. Wiley. 25, 27
- Cheriyán, J. & Ravi, R. (1998). Approximation algorithms for network problems. *Lecture Notes*. 25, 26, 27
- Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree shortest path relaxation. *Mathematical Programming*, (20), 255 – 282. 3
- Cortes, J. M. R. (1998). Uma contribuição para a resolução do problema de alocação multiobjetivo de plataformas de produção de petróleo. Dissertação de Mestrado em Ciências de Engenharia - Universidade Estadual do Norte Fluminense. Campos dos Goytacazes - RJ. 19
- Cortes, J. M. R. (2003). Um algoritmo genético para solução do problema de localização de atividades econômicas. Tese de doutorado – UENF. 3, 11, 12, 18, 33
- Cortes, J. M. R. & Paula Junior, G. G. (2005). Uma abordagem para resolução do problema de localização de atividades econômicas. *XXV ENEGEP*. 8, 17
- Crainic, T. G. (2002). A survey of optimization models for long-haul freight transportation. Publicação CRT-2002-05. Centre de recherche sur les transports Université de Montréal, QC, Canadá. 10
- Deneubourg, J.-L., Aron, S., & Goss, S. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3, 159–169. 119
- Diekmann, R., Lüling, R., & Simon, J. (1992). A general purpose distributed implementation of simulated annealing. In *Proceedings of the Fourth IEEE Symposium on 1-4 Dec 1992* (pp. 94–101). 110
- Diekmann, R., Lüling, R., & Simon, J. (1993). Problem independent distributed simulated annealing and its applications. In *Applications, Applied Simulated Annealing*,

*Lecture Notes in Economics and Mathematical Systems, Springer LNEMS 396*  
(pp. 17–44).: Springer. 110

Dorigo, M. & Stützle, T. (2004). *Ant Colony Optimization*. Bradford Book, MIT Press.  
119

Farah, M. J. (2002). Os desafios da logística e os centros de distribuição física. *Revista FAE Business*, (2), 44 – 46. 81

Feo, T. A. & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, 109–133. 34, 36, 39

Filho, R. S. V., Gomes, M. J. N., & Xavier, A. E. (2005). Um algoritmo grasp híbrido para o problema de localização capacitado de custo fixo. *Anais do XXXVII SBPO*. 21, 34

Frinhani, R. M. D. (2011). Grasp com path-relinking para agrupamento de dados biológicos. Dissertação de Mestrado, Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, Belo Horizonte. 34, 41

García, J. A. D. (2001). Algorithmic approaches for the single source capacitated plant location problem. Tesis Doctoral, Dept. d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Barcelona. 28, 34, 41, 45, 48

Garey, M. R. & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman. 25

Ghosh, A. & Craig, S. (1983). Formulating retail location strategy in a changing environment. *Journal of Marketing*, 47, 56 – 68. 3

Glover, F. (1974). *Heuristics for Integer Programming Using Surrogate Constraints*. Report. Business Research Division, University of Colorado. 120

Glover, F. (1989). Tabu search – Part I. *ORSA Journal on Computing*, 1(3), 190–206.  
106

Glover, F. (1990a). Tabu search – Part II. *ORSA Journal on Computing*, 2(1), 4–32.  
106

Glover, F. (1990b). Tabu search: A tutorial. *Interfaces*, 4, 74–94. 106

- Glover, F. & Kochenberger, G. A., Eds. (2003). *Handbook of Metaheuristics*. International series in operations research & management science. Boston, Dordrecht, London: Kluwer Academic Publishers. 59
- Glover, F. & Laguna, M. (1993). Tabu search. In Reeves C. (ed) modern heuristic techniques for combinatorial problems. (pp. 70–150). 104
- Glover, F. & Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers. 31
- Glover, F., Laguna, M., & Martí, R. (2003). Scatter search and path relinking: Advances and applications. 33, 59
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning, Reading*. Addison Wesley. 102
- Günther, Z., Braune, R., & Bögl, M. (2010). *Metaheuristic Search Concepts: A Tutorial with Applications to Production and Logistics*. Springer-Verlag Berlin Heidelberg. xiii, 37, 113, 114, 123
- Haeser, G. & Gomes-Ruggiero, M. (2008). Aspectos teóricos de simulated annealing e um algoritmo duas fases em otimização global. *TEMA*, 9(3), 395 – 404. 34
- Hakimi, S. L. (1964). Optimal locations of switching centers and medians of a graph. *European Journal Of Operational Research*, 12(20), 255 – 282. 11, 16
- Hart, J. P. & Shogan, A. W. (1987). Semi-greedy heuristics: An empirical study. *Operations Research Letters*, 6(3), 107–114. 36
- Holland, J. (1975a). *Adaptation in natural and artificial systems*. University of Michigan Press. 31
- Holland, J. H. (1975b). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press. 102
- Junior, I. R. S. & Cechin, A. L. (2006). Comparação entre métodos exatos e heurísticos para tratar o problema de roteamento de veículos em um ambiente fabril. Anais do XXVI ENEGEP. 3
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680. 108

- Lacerda, L. (2000). Considerações sobre o estudo de localização de instalações. Disponível em <www.sargas.com.br> Acesso em 14 fev 2013. 8
- Laguna, M. & Martí, R. (1999). Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS J. on Computing*, 11(1), 44–52. 61
- Laguna, M. & Martí, R. (2003). Scatter search: Methodology and implementations in c. 120
- Lamosa, M. J. P., Marchi, M. M. D., & Lamosa, D. M. (2010). Posicionamento de sensores considerando facilidades com características distintas. Anais do XLII SOBRAPO. 35
- Leal, L. A. S. (2002). Uma fundamentação teórica para a complexidade estrutural de problemas de otimização. Tese de Doutorado - UFRS. 49, 50
- Leite, E. C. B. R., Paula Junior, G. G., Velasco, A. S., & Nogueira, R. T. (2000). Grasp para otimizar o carregamento de um contêiner. Anais do XL SOBRAPO. 35
- Longo, H. J. (2005). Técnicas para programação inteira a aplicações em problemas de roteamento de veículos. Tese de doutorado — PUC-RIO. 29
- Lopes, Y. G. & de Almeida, A. T. (2008). Enfoque multicritério para a localização de instalações de serviço: aplicação do método smarter. *Sistemas & Gestão*, 2, 114 – 128. 19
- Lorena, L. A. N., Senne, E. L. F., Paiva, J. A. C., & Pereira, M. A. (2001). Integração de modelos de localização a sistemas de informações geográficas. *Gestão & Produção*, 8(2), 180–195. 16
- Luke, S. (2011). *Essentials of Metaheuristics*. Lulu. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>. 102
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. 103
- Michalewicz, Z. (2010). *How to Solve It: Modern Heuristics*. Berlin, Heidelberg: Springer-Verlag. 100

- Nemhauser, G. & Wolsey, L. (1988). *Integer and combinatorial optimization*. Wiley-Interscience series in discrete mathematics and optimization. Wiley. 25, 27
- Nemhauser, G. L. (1966). *Introduction to Dynamic Programming*. Series in decision and control. Wiley. 74
- Nogueira, R. T. (2002). Heurísticas grasp para o problema quadrático da mochila 0-1. Tese de Doutorado - Universidade Estadual do Norte Fluminense. Campos dos Goytacazes - RJ. 49, 51, 52
- Nogueira, R. T., Paula Junior, G. G., & Póvoa, C. L. R. (2006). Uma heurística grasp para o problema do pequeno investidor. Anais do XIII SIMPEP. 35
- Owen, S. H. & Daskin, M. S. (1998). Strategic facility location: a review. *European Journal of Operation Research*, 111 - 3, 423 – 447. 17
- Papadimitriou, C. H. & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications. 25
- Pereira, M. A. (2005). Um método *Branch-and-Price* para problemas de localização de  $p$ -medianas. Tese de doutorado — INPE. 13, 29
- Porter, M. E. (1999). *On competition: competição, estratégias competitivas essenciais*, 4 ed. Editora Campus. 3
- Rangel, M. C., Abreu, N. M. M., & Netto, P. O. B. (2000). Grasp para o pqa: ufm limite de aceitações para soluções iniciais. *Pesquisa Operacional*, 20 - 1, 46–58. 35, 47
- Resende, M. G. C. & Ribeiro, C. C. (2003). Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, (pp. 219–249). 40
- Resende, M. G. C. & Werneck, R. F. (2002a). A grasp with path-relinking for the  $p$ -median problem. AT&T Labs Research Technical Report TD-5E53XL. 34, 41
- Resende, M. G. C. & Werneck, R. F. (2002b). A hibrid heuristic for the  $p$ -median problem. AT&T Labs Research Technical Report TD-5E53XL. 16, 21, 33
- Rezende, S. O. (2005). *Sistemas inteligentes: fundamentos e aplicações*. Manole Ltda. 32



- Sedgewick, R. & Flajolet, P. (2013). *An Introduction to the Analysis of algorithms*. Addison-Wesley. 49
- Sivanandam, S. N. & Deepa, S. N. (2007). *Introduction to Genetic Algorithms*. Springer Publishing Company, Incorporated, 1st edition. 103
- Souza, A. S. & Asada, E. N. (2012). Uma nova abordagem *Branch-and-Cut* aplicada ao problema de planejamento da expansão de redes de transmissão de grande porte. *Revista Controle & Automação*, 23 - 1. 29
- Sule, D. R. (2001). *Logisitcs of Facility Location and Allocation*. Industrial Engineering: A Series of Reference Books and Textboo Series. Marcel Dekker Incorporated. 8
- Talbi, E. (2009). *Metaheuristics: From Design to Implementation*. Wiley Publishing. xiii, 61, 62, 100, 114
- Thünen, J. H. V. (1866). Isolated state an english edition of der isolierte staat. Trans. C.M. Watengerg. Oxford: Pergamon. 11
- Tibes, E. (2005). Ferramenta de apoio à decisão com base em problema de p-medianas. XXVIII CNMAC. 16
- Wanke, P. (2001). Aspectos fundamentais do problema de localização de instalações em redes logísticas. Disponível em <http://www.centrodelogistica.org/new/fr-aspec.htm>. Acesso em 16 jan 2012. 82
- Weber, A. (1909). *Theory of the location of industries*. Trans. C.J. Freidrich. Chicago: The University of Chicago Press. 11
- Yang, X. S. (2008). *Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics*. Cambridge International Science Publishing, Limited. 100, 109
- Yu, X. & Gen, M. (2012). *Introduction to Evolutionary Algorithms*. Springer Publishing Company, Incorporated. xiii, 103, 121