

UNIVERSIDADE FEDERAL FLUMINENSE  
INSTITUTO DE COMPUTAÇÃO  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

**Thales Willmann Barra**

**Avaliação do Tamanho do Conjunto Elite Utilizado na  
Meta-heurística Híbrida DM-GRASP**

Niterói  
2010

**Thales Willmann Barra**

# **Avaliação do Tamanho do Conjunto Elite Utilizado na Meta-heurística Híbrida DM-GRASP**

**Monografia apresentada ao Departamento  
de Ciência da Computação da Universidade  
Federal Fluminense como parte dos  
requisitos para obtenção do Grau de  
Bacharel em Ciência da Computação.**

Orientadores: Simone de Lima Martins e Alexandre Plastino de Carvalho

Niterói

2010

**Thales Willmann Barra**

# **Avaliação do Tamanho do Conjunto Elite Utilizado na Meta-heurística Híbrida DM-GRASP**

**Monografia apresentada ao Departamento  
de Ciência da Computação da Universidade  
Federal Fluminense como parte dos  
requisitos para obtenção do Grau de  
Bacharel em Ciência da Computação.**

Aprovado em Junho de 2010

## **BANCA EXAMINADORA**

---

Simone de Lima Martins, D.Sc.  
Orientadora  
UFF

---

Alexandre Plastino de Carvalho, D.Sc.  
Co-Orientador  
UFF

---

Isabel Cristina Mello Rosseti, D.Sc.  
UFF

---

Renatha Oliva Capua, M.Sc.  
UFF

Niterói

2010

## RESUMO

Este trabalho tem como objetivo avaliar o desempenho da meta-heurística híbrida DM-GRASP considerando a variação do tamanho do conjunto elite e determinar valores adequados para este parâmetro de entrada. Para tanto foram realizadas execuções desta meta-heurística e também de sua variante DM-D5 para o problema das  $p$ -medianas.

**Palavras Chave:**

DM-GRASP, conjunto elite, meta-heurística híbrida, mineração de dados.

# ABSTRACT

This work aims to evaluate the performance of the hybrid metaheuristic DM-GRASP considering the variation of the elite set's size and to determine adequate values for this input parameter. Several computational experiments were performed using this metaheuristic and its variant DM-D5 for the  $p$ -medians problem.

**Keywords:**

DM-GRASP, elite set, hybrid metaheuristic, data mining.

# LISTA DE ACRÔNIMOS

GRASP: *Greedy Randomized Adaptive Search Procedure*

DM-GRASP: *Data Mining GRASP*

MDM-GRASP: *Multi Data Mining GRASP*

DM-D5: DM-GRASP Dinâmico

LC: Lista de Candidatos

LRC: Lista Restrita de Candidatos

FIM: *Frequent Itemset Mining*

# SUMÁRIO

<b>CAPÍTULO 1 – INTRODUÇÃO</b>	<b>5</b>
<b>CAPÍTULO 2 – AS META-HEURÍSTICAS HÍBRIDAS DM-GRASP E DM-D5</b>	<b>7</b>
2.1 – GRASP .....	7
2.2 – DM-GRASP .....	8
2.3 – DM-D5 .....	8
2.4 – IMPLEMENTAÇÃO DO DM-GRASP E DO DM-D5 PARA O PROBLEMA DAS <i>P</i> -MEDIANAS .....	9
2.4.1 – O Problema das <i>P</i> -medianas .....	9
2.4.2 – Implementação do DM-GRASP e do DM-D5 .....	9
<b>CAPÍTULO 3 – RESULTADOS EXPERIMENTAIS</b>	<b>13</b>
3.1 – RESULTADOS EXPERIMENTAIS PARA O DM-GRASP .....	13
3.2 – RESULTADOS EXPERIMENTAIS PARA O DM-D5 .....	16
<b>CAPÍTULO 4 – CONCLUSÕES</b>	<b>21</b>
<b>REFERÊNCIAS</b>	<b>22</b>

## LISTA DE FIGURAS

FIGURA 1: PSEUDOCÓDIGO DO DM-GRASP .....	10
FIGURA 2: PSEUDOCÓDIGO DA FASE DE CONSTRUÇÃO DO GRASP .....	11
FIGURA 3: PSEUDOCÓDIGO DA FASE DE CONSTRUÇÃO ADAPTADA DO DM-GRASP .....	12
FIGURA 4: NÚMERO DE VEZES QUE CADA TAMANHO DE CONJUNTO ELITE ALCANÇOU A MELHOR SOLUÇÃO (DM-GRASP) .....	14
FIGURA 5: DIFERENÇA EM RELAÇÃO À SOLUÇÃO ÓTIMA PARA CADA TAMANHO DE CONJUNTO ELITE (DM-GRASP) .....	14
FIGURA 6: EXECUÇÃO DO DM-GRASP COM CONJUNTO ELITE DE TAMANHO: (A) 2, (B) 10, (C) 35 E (D) 60 .....	15
FIGURA 7: NÚMERO DE VEZES QUE CADA TAMANHO DE CONJUNTO ELITE ALCANÇOU A MELHOR SOLUÇÃO (DM-D5) .....	17
FIGURA 8: DIFERENÇA EM RELAÇÃO À SOLUÇÃO ÓTIMA PARA CADA TAMANHO DE CONJUNTO ELITE (DM-D5) .....	17
FIGURA 9: EXECUÇÃO DO DM-D5 COM CONJUNTO ELITE DE TAMANHO: (A) 2, (B) 10, (C) 35 E (D) 60 .....	19



## **LISTA DE TABELAS**

TABELA 1: TEMPOS COMPUTACIONAIS DO DM-GRASP (EM SEGUNDOS) . . . . . 16

TABELA 2: TEMPOS COMPUTACIONAIS DO DM-D5 (EM SEGUNDOS) . . . . . 20

## CAPÍTULO 1 – INTRODUÇÃO

Meta-heurísticas representam uma classe importante de técnicas para resolver, aproximadamente, problemas de otimização combinatória para os quais o uso de métodos exatos é impraticável. Neste contexto, destaca-se a meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*), que já foi aplicada com sucesso em vários problemas [2]. Este método consiste em um processo iterativo em que cada iteração é composta de duas fases: construção e busca local, que serão descritas mais adiante.

Processos de mineração de dados são caracterizados pela extração de informações úteis, implícitas em bases de dados, sob a forma de regras e padrões. A meta-heurística híbrida DM-GRASP (*Data Mining GRASP*) utiliza um algoritmo de mineração de dados para a obtenção de padrões que auxiliam na busca por soluções melhores [10]. Esta mineração é feita em um conjunto denominado conjunto elite, que contém as melhores soluções encontradas, até o momento corrente de execução, pela heurística GRASP.

Este método foi utilizado com sucesso em quatro problemas de otimização combinatória: problema da maximização da diversidade [13], problema de empacotamento de conjuntos [9], problema de replicação de servidores em transmissão *multicast* confiável [11] e o problema das  $p$ -medianas [7].

O DM-D5 (DM-GRASP Dinâmico) é uma variação adaptativa do DM-GRASP que determina dinamicamente o momento de se realizar a mineração de dados, e também pode realizá-la várias vezes durante a execução de uma aplicação [7].

Em todos os experimentos realizados anteriormente, o tamanho do conjunto elite foi fixado arbitrariamente em dez elementos, o que não caracteriza necessariamente a melhor escolha. O objetivo deste trabalho é avaliar o desempenho do DM-GRASP e do DM-D5 variando o tamanho do conjunto elite, na tentativa de determinar tamanhos adequados para este conjunto.

O restante deste trabalho está organizado da seguinte forma. No Capítulo 2, são descritos o DM-GRASP e o DM-D5, em seguida é dada a definição do problema das  $p$ -medianas, o problema usado nos experimentos deste trabalho, e posteriormente são descritas as implementações de ambas as heurísticas para o problema das  $p$ -medianas. No Capítulo 3,

são apresentados e analisados os resultados experimentais e, no Capítulo 4, são apresentadas as conclusões.

## CAPÍTULO 2 – AS META-HEURÍSTICAS HÍBRIDAS DM-GRASP E DM-D5

Neste capítulo, será descrita a meta-heurística GRASP, que proporciona a base para as versões híbridas DM-GRASP e DM-D5, e estas últimas serão definidas em seguida. Também será abordada a implementação destas meta-heurísticas para o problema das  $p$ -medianas.

### 2.1 GRASP

O GRASP é uma meta-heurística que combina um algoritmo guloso com um componente aleatório [8]. Métodos gulosos levam a boas soluções, mas normalmente não atingem soluções ótimas. Incluindo elementos aleatoriamente à solução, é possível diversificá-la, apesar da diminuição da qualidade média das soluções.

O GRASP é um processo iterativo, sendo cada iteração composta de duas fases: a construção e a busca local. A fase de construção gera uma solução elemento por elemento. A cada passo, os elementos que ainda não foram incluídos na solução são avaliados por uma função gulosa, e os que tiverem melhor avaliação são incluídos numa lista chamada Lista Restrita de Candidatos (*LRC*).

O critério para a inclusão de um elemento na *LRC* pode ser definido usando-se um parâmetro  $\alpha \in [0,1]$ . Supondo que num problema de minimização o elemento com a melhor avaliação tenha o valor *menor\_custo*, a *LRC* pode ser composta por todos os elementos com valor de avaliação no intervalo  $[\textit{menor\_custo}, \textit{menor\_custo} + (\textit{maior\_custo} - \textit{menor\_custo}) \times \alpha]$ . Um elemento é selecionado aleatoriamente desta lista e inserido na solução.

A fase de busca local baseia-se no conceito de vizinhança de uma solução para melhorar a solução encontrada na fase de construção. A vizinhança  $N$  de uma solução  $s$  é definida a partir de uma operação  $O$  que se realiza sobre  $s$  para se gerar uma nova solução. O conjunto de todas as soluções que podem ser geradas a partir de  $s$  por  $O$  constitui a vizinhança  $N$  de  $s$ . Nessa fase, a vizinhança da solução é percorrida, e caso uma solução melhor seja encontrada, uma nova busca local é feita nesta solução encontrada. Quando soluções melhores

não forem encontradas em uma vizinhança, conclui-se que um ótimo local foi encontrado e a busca local termina.

No GRASP, as iterações são executadas independentemente, e como consequência, o conhecimento adquirido nas iterações passadas não é explorado nas seguintes.

## 2.2 DM-GRASP

O DM-GRASP é uma versão híbrida do GRASP que incorpora um algoritmo de mineração de dados para auxiliar a busca por soluções melhores [10]. Sua execução é dividida em duas etapas, como descrito a seguir.

A primeira etapa, chamada fase de geração do conjunto elite, é composta de um determinado número de iterações (*num\_it\_conj\_elite*) do GRASP original. Nesta etapa, é criado um conjunto chamado conjunto elite, que armazena as *tam\_elite* melhores soluções encontradas. Em seguida, um algoritmo de mineração de dados é executado para extrair padrões desse conjunto.

Os padrões são conjuntos de elementos que aparecem frequentemente nas soluções do conjunto elite. A quantidade de padrões a serem minerados é definida pelo parâmetro *qtd\_padroes*. O parâmetro *sup\_min* (suporte mínimo) indica o número mínimo de vezes que um trecho de solução deve aparecer no conjunto elite para ser considerado um padrão.

Na segunda etapa, chamada fase híbrida, são executadas outras *num\_it\_conj\_elite* iterações do GRASP. Nestas iterações, a fase de construção é modificada para utilizar os padrões encontrados como soluções iniciais parciais, ou seja, em cada iteração um padrão é selecionado e seus elementos são inseridos na solução parcial, e os elementos restantes são escolhidos de acordo com o algoritmo da fase de construção do GRASP. A fase de busca local permanece inalterada.

## 2.3 DM-D5

A versão adaptativa do DM-GRASP, também conhecida como MDM-GRASP (Multi Data Mining GRASP), diferencia-se por executar a mineração de dados um número variável de vezes, definido dinamicamente. Esta característica tem se mostrado útil para a obtenção de soluções de maior qualidade [7].

A ideia desta estratégia é executar a mineração de dados nas seguintes situações: (1) assim que o conjunto elite se tornar estável, isto é, ele não sofrer alterações por um determinado número de iterações; e (2) sempre que o conjunto elite sofrer alguma alteração e se tornar estável novamente. Acredita-se que deste modo a evolução gradual do conjunto elite permita a extração de padrões mais refinados ao longo das iterações.

A implementação atual do MDM-GRASP, chamada DM-D5, considera que o conjunto elite se torna estável ao não sofrer nenhuma alteração por um número de iterações igual a 5% do total das iterações a serem executadas pelo algoritmo.

## 2.4 IMPLEMENTAÇÃO DO DM-GRASP E DO DM-D5 PARA O PROBLEMA DAS $P$ -MEDIANAS

### 2.4.1 O Problema das $p$ -medianas

O problema das  $p$ -medianas é um problema NP-difícil [6] que já foi aplicado a vários problemas reais de localização [14]. Ele pode ser definido como se segue.

Define-se  $F$  como um conjunto de  $m$  facilidades em potencial e  $C$  um conjunto de  $n$  clientes. A função  $d : C \times F \rightarrow \mathfrak{R}$  avalia a distância entre um cliente e uma facilidade em potencial. Dado um inteiro positivo  $p$ ,  $p \leq m$ , o problema das  $p$ -medianas consiste em identificar um subconjunto  $R$  de  $F$  tal que  $|R| = p$  e a soma das distâncias de cada cliente em  $C$  até a facilidade mais próxima em  $R$  seja minimizada. Sem perder a generalização, consideramos  $F = C$ , isto é, em cada posição de cliente existe uma facilidade em potencial.

### 2.4.2 Implementação do DM-GRASP e do DM-D5

O pseudocódigo do DM-GRASP, utilizado para o problema das  $p$ -medianas, desenvolvido em [7], está ilustrado na Figura 1. Nas linhas 1 e 2, a melhor solução e o conjunto elite são inicializados com conjuntos vazios. O laço da linha 3 até a linha 10 corresponde à fase de geração do conjunto elite, na qual são executadas  $num\_it\_conj\_elite$  iterações do GRASP. A fase de construção é executada na linha 4, seguida da fase de busca local na linha 5. O conjunto elite, composto de  $tam\_elite$  soluções, é atualizado na linha 6, onde é verificado se a solução atual é melhor do que a pior solução desse conjunto, ou se o conjunto elite ainda não está cheio, para que a solução atual seja inserida nele.

Na linha 7, é verificado se a melhor solução deve ser atualizada, o que é feito na linha 8. O procedimento de mineração de dados extrai os padrões do conjunto elite na linha 11. O laço da linha 12 até a linha 19 corresponde à fase híbrida do DM-GRASP, que também possui  $num\_it\_conj\_elite$  iterações. Na linha 13, é selecionado o próximo padrão para servir de base para a solução. Os padrões são selecionados em ordem decrescente de tamanho, de forma circular, e caso haja mais de um padrão com o mesmo tamanho, um deles é escolhido aleatoriamente. Na linha 14, é executada a fase de construção adaptada, partindo do padrão escolhido. Na linha 15, é feita a busca local, que é a mesma do laço anterior. Nas linhas 16 e 17, novamente, é verificada a necessidade de se atualizar a melhor solução. E, finalmente, na linha 20, a melhor solução é retornada.

```

procedimento DM-GRASP(num_it_conj_elite, tam_elite, qtd_padroes, sup_min,  $\alpha$ )
01.  melhor_sol  $\leftarrow \emptyset$ ;
02.  conjunto_elite  $\leftarrow \emptyset$ ;
03.  para it  $\leftarrow 1$  até num_it_conj_elite faça
04.    sol  $\leftarrow$  Fase_de_Construção( $\alpha$ );
05.    sol  $\leftarrow$  Busca_Local(sol);
06.    Atualiza_Conjunto_Elite(conjunto_elite, sol, tam_elite);
07.    se custo(sol) < custo(melhor_sol) então
08.      melhor_sol  $\leftarrow$  sol;
09.    fim se
10.  fim para
11.  padrões  $\leftarrow$  Minerar(conjunto_elite, qtd_padroes, sup_min);
12.  para it  $\leftarrow 1$  até num_it_conj_elite faça
13.    padrão  $\leftarrow$  Próximo Padrão(padrões);
14.    sol  $\leftarrow$  Fase_de_Construção_Adaptada(padrão,  $\alpha$ );
15.    sol  $\leftarrow$  Busca_Local(sol);
16.    se custo(sol) < custo(melhor_sol) então
17.      melhor_sol  $\leftarrow$  sol;
18.    fim se
19.  fim para
20.  retorne melhor_sol;

```

Figura 1: Pseudocódigo do DM-GRASP

A Figura 2 ilustra a fase de construção. Na linha 1, a solução corrente (*sol*) é inicializada com o conjunto vazio. Na linha 2, todos os elementos do domínio são inseridos na Lista de Candidatos (*LC*). No laço da linha 3 até a linha 11, são feitas iterações até que a solução tenha  $p$  elementos. Nas linhas 4 a 6, cada elemento  $e$  da *LC* é avaliado por uma

função que calcula o custo parcial da solução ao se inserir este elemento. Em seguida, na linha 7, a  $LRC$  é criada com todos os elementos da  $LC$  cujas avaliações estejam no intervalo  $[menor\_custo, menor\_custo + (maior\_custo - menor\_custo) \times \alpha]$ . Na linha 8, um elemento  $s$  é selecionado aleatoriamente da  $LRC$ , para ser adicionado à solução  $sol$  na linha 9. Na linha 10, a  $LC$  é atualizada, sendo retirado o elemento que foi inserido na solução e, na linha 12, a solução encontrada é retornada.

<p><b>procedimento</b> Fase_de_Construção(<math>\alpha</math>)</p> <ol style="list-style-type: none"> <li>1. <math>sol \leftarrow \emptyset</math>;</li> <li>2. <math>LC \leftarrow</math> elementos do domínio;</li> <li>3. <b>repita</b></li> <li>4.     <b>para cada</b> elemento <math>e</math> na <math>LC</math></li> <li>5.         <math>custo(e) \leftarrow</math> soma das distâncias entre cada elemento em <math>C - \{sol \cup \{e\}\}</math> e seu elemento mais próximo em <math>sol \cup \{e\}</math>;</li> <li>6.     <b>fim para</b></li> <li>7.     <math>LRC \leftarrow</math> Constrói_LRC(<math>\alpha, LC</math>);</li> <li>8.     <math>s \leftarrow</math> Escolha_Aleatória(<math>LRC</math>);</li> <li>9.     <math>sol \leftarrow sol \cup \{s\}</math>;</li> <li>10.     <math>LC \leftarrow LC - \{s\}</math>;</li> <li>11.     <b>até</b> Solução_Completa(<math>sol</math>);</li> <li>12.     <b>retorne</b> <math>sol</math>;</li> </ol>
---

Figura 2: Pseudocódigo da fase de construção do GRASP

A fase de construção adaptada é ilustrada na Figura 3. Ela diferencia-se da fase de construção original apenas por inicializar a solução parcial com o padrão escolhido, fornecido como parâmetro, e excluir da  $LC$  os elementos presentes nesse padrão.

A fase de busca local começa avaliando o custo da solução atual calculando a soma das distâncias entre todos os clientes e suas facilidades mais próximas. Em seguida, para cada elemento da solução, são feitas trocas com elementos próximos em suas partições. Consideramos que um elemento  $e$  é próximo de  $r_i$  na partição  $P_i$  se a distância entre  $e$  e  $r_i$  é menor ou igual à média das distâncias entre  $r_i$  e todos os elementos em  $P_i$ . Para reduzir o esforço computacional, a solução obtida por cada troca é avaliada aproximadamente, e apenas a melhor solução é avaliada exatamente. Essa avaliação aproximada recalcula as distâncias apenas na partição  $P_i$ , desconsiderando as outras partições, que são necessárias para a avaliação exata. Se a solução encontrada for melhor do que a solução atual, a busca local recomeça, caso contrário essa solução é retornada.



A mineração de dados utilizada no DM-GRASP se baseia na tarefa de mineração de conjuntos de itens frequentes (do inglês, *Frequent Itemset Mining*, *FIM* [5]). Esta estratégia consiste em extrair de um conjunto de transações (neste caso um conjunto de soluções) esses conjuntos de itens frequentes de acordo com o suporte mínimo especificado como parâmetro. Um conjunto de itens frequentes é chamado maximal se o mesmo não estiver contido em outro conjunto que também seja frequente. Para evitar a mineração de conjuntos frequentes que são subconjuntos uns dos outros, esta implementação do DM-GRASP extrai apenas conjuntos maximais. Para tanto, é utilizado o algoritmo FPmax\* [4], disponível em <http://fimi.cs.helsinki.fi>.

O DM-D5 [7] diferencia-se do DM-GRASP por verificar ao longo das iterações se o conjunto elite se tornou estável para ativar a mineração de dados, em vez de executá-la apenas uma vez exatamente após a metade das iterações.

<p><b>procedimento</b> Fase_de_Construção Adaptada(<i>padrão</i>, <math>\alpha</math>)</p> <ol style="list-style-type: none"> <li>1. <math>sol \leftarrow padr\u00e3o</math>;</li> <li>2. <math>LC \leftarrow</math> elementos do dom\u00ednio <math>- padr\u00e3o</math>;</li> <li>3. <b>repita</b></li> <li>4.     <b>para cada</b> elemento <math>e</math> na <math>LC</math></li> <li>5.         <math>custo(e) \leftarrow</math> soma das dist\u00e2ncias entre cada elemento em <math>C - \{sol \cup \{e\}\}</math> e seu elemento mais pr\u00f3ximo em <math>sol \cup \{e\}</math>;</li> <li>6.     <b>fim para</b></li> <li>7.     <math>LRC \leftarrow</math> Constr\u00f3i_LRC(<math>\alpha</math>, <math>LC</math>);</li> <li>8.     <math>s \leftarrow</math> Escolha_Aleat\u00f3ria(<math>LRC</math>);</li> <li>9.     <math>sol \leftarrow sol \cup \{s\}</math>;</li> <li>10.     <math>LC \leftarrow LC - \{s\}</math>;</li> <li>11. <b>at\u00e9</b> Solu\u00e7\u00e3o_Completa(<math>sol</math>);</li> <li>12. <b>retorne</b> <math>sol</math>;</li> </ol>
---

Figura 3: Pseudoc\u00f3digo da fase de constru\u00e7\u00e3o adaptada do DM-GRASP

## CAPÍTULO 3 – RESULTADOS EXPERIMENTAIS

Neste capítulo, os resultados dos experimentos computacionais para as heurísticas DM-GRASP e DM-D5 são apresentados. Os algoritmos foram implementados em C++ e compilados com g++ (gcc) 4.2.3. Os testes foram realizados em um computador Intel Core 2 Quad Q6600 (2.4 GHz) com 3GB de memória RAM, executando Linux (kernel 2.6.24).

Foram utilizadas 80 instâncias do problema das  $p$ -medianas, as mesmas usadas em [3]. Existem quatro grupos, com 20 instâncias cada. As instâncias do primeiro grupo (G50) possuem 50 clientes e  $p$  variando de seis a 25, com incremento de um. No segundo, terceiro e quarto grupos (G287, G654 e G1060, respectivamente) as instâncias possuem 287, 654 e 1060 clientes, com  $p$  variando de cinco a 100 com incremento de cinco.

Para cada instância foram utilizados 15 tamanhos de conjunto elite diferentes, variando de dois a 60. Para cada tamanho, foram realizadas 10 execuções de cada heurística, com sementes aleatórias diferentes (de um a dez). O parâmetro  $\alpha$  foi definido como 0,2, o suporte mínimo definido como dois, e as execuções possuem 2000 iterações cada. O número de padrões utilizados na fase de construção adaptada foi definido como dez.

### 3.1 RESULTADOS EXPERIMENTAIS PARA O DM-GRASP

As Figuras 4 e 5 apresentam os resultados obtidos pelo DM-GRASP relacionados à qualidade das soluções obtidas por cada tamanho de conjunto elite. A Figura 4 exibe o número de vezes (eixo das ordenadas) que cada tamanho de conjunto elite (eixo das abscissas) obteve o melhor resultado, isto é, o menor desvio em relação à solução ótima, para cada grupo de instâncias, desconsiderando-se as instâncias onde os resultados foram sempre os mesmos. O conjunto elite de tamanho 10 destaca-se por obter o melhor resultado geral.

A Figura 5, que mostra a soma dos desvios médios em relação à solução ótima de cada instância, indica que, embora o tamanho do conjunto elite igual a dez tenha se destacado por alcançar mais vezes as melhores soluções, a diferença entre as soluções obtidas com os diferentes tamanhos de conjunto elite são pequenas.

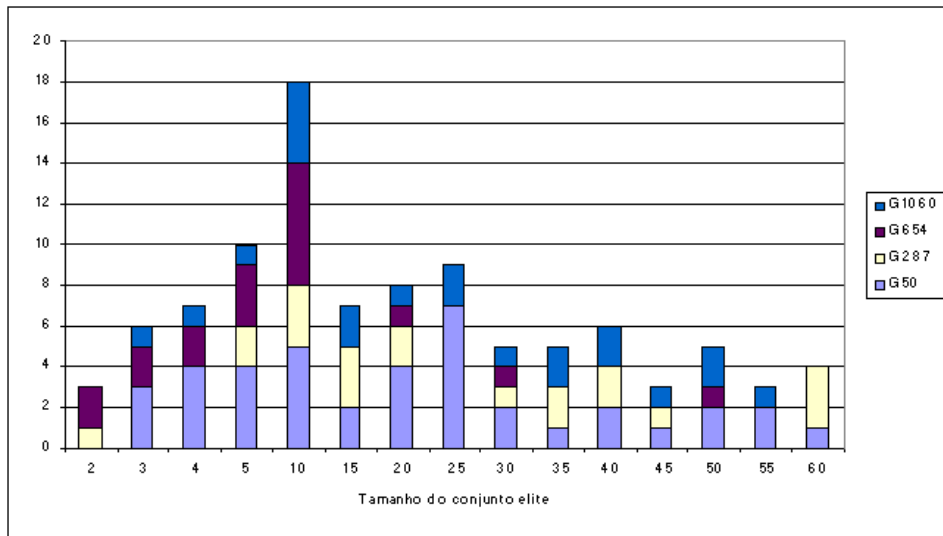


Figura 4: Número de vezes que cada tamanho de conjunto elite alcançou a melhor solução (DM-GRASP).

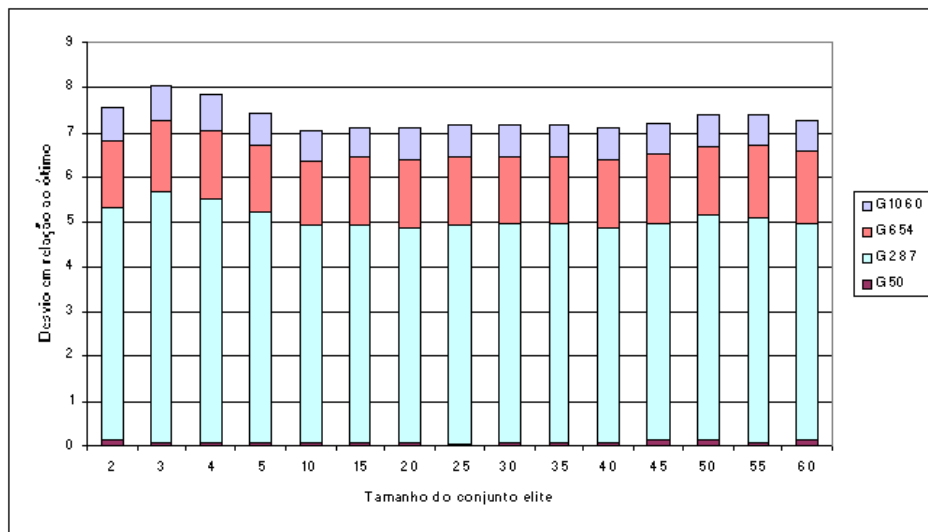


Figura 5: Diferença em relação à solução ótima para cada tamanho de conjunto elite (DM-GRASP).

A Figura 6 ilustra a variação no custo das soluções obtidas nas fases de construção e busca local, ao longo das iterações de uma execução do DM-GRASP para a instância G287,  $p = 70$  e conjunto elite de tamanho: (a) 2, (b) 10, (c) 35 e (d) 60. A queda existente na milésima iteração dos quatro gráficos coincide com o ponto onde o algoritmo de mineração de dados é acionado.

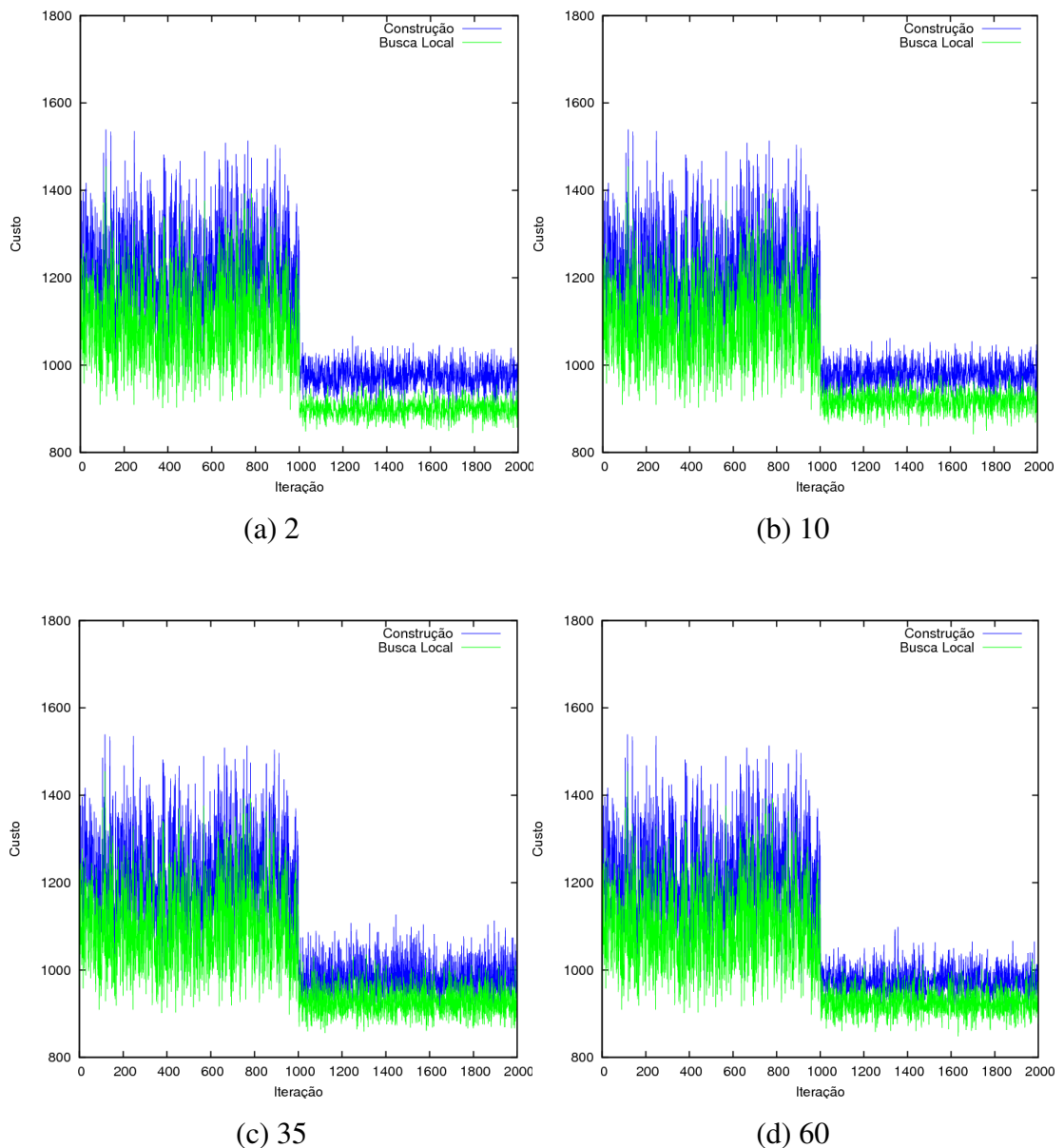


Figura 6: Execução do DM-GRASP com conjunto elite de tamanho: (a) 2, (b) 10, (c) 35 e (d) 60.

Após a extração dos padrões, a qualidade das soluções geradas tanto na fase de construção quanto na busca local aumenta significativamente, conforme observado em outros trabalhos ([7], [11], [12], [13]). Este comportamento também foi observado nas execuções para outras instâncias.

A Tabela 1 apresenta os tempos computacionais das execuções do DM-GRASP com os diferentes tamanhos de conjunto elite. A primeira coluna indica o tamanho do conjunto elite utilizado. As quatro colunas seguintes contêm os tempos computacionais de cada grupo. A última coluna indica o tempo total gasto na execução dos quatro grupos, para cada tamanho

do conjunto elite. Nota-se que os tempos de execução aumentam à medida que o tamanho do conjunto elite aumenta. Isto se deve ao aumento no tempo gasto no processo de mineração de dados, já que um conjunto elite com mais elementos torna mais complexa a identificação de padrões.

Tamanho do conj. elite	G50	G287	G654	G1060	Total
2	9,14	625,43	3138,59	8744,17	12517,33
3	9,51	640,01	3262,19	9111,37	13023,08
4	9,63	641,30	3263,53	9152,84	13067,30
5	9,57	627,84	3215,33	8845,27	12698,02
10	9,07	612,57	3102,16	8506,15	12229,94
15	9,01	614,08	3068,02	8411,34	12102,46
20	9,09	635,86	3061,34	8362,81	12069,10
25	9,36	701,12	3079,03	8369,48	12158,98
30	9,69	857,40	3135,97	8427,68	12430,74
35	10,14	1227,29	3261,15	8544,13	13042,71
40	10,72	1824,61	3459,32	8715,04	14009,69
45	11,52	2885,74	3808,08	8986,41	15691,76
50	12,47	4804,63	4323,87	9372,70	18513,67
55	13,61	7317,77	5031,32	9913,42	22276,12
60	14,91	11318,78	6129,08	10599,38	28062,14

Tabela 1: Tempos computacionais do DM-GRASP (em segundos).

### 3.2 RESULTADOS EXPERIMENTAIS PARA O DM-D5

As Figuras 7 e 8 apresentam os resultados obtidos pelo DM-D5 relacionados à qualidade das soluções obtidas por cada tamanho de conjunto elite. A Figura 7 exibe o número de vezes (eixo das ordenadas) que cada tamanho de conjunto elite (eixo das abscissas) obteve o melhor resultado, isto é, o menor desvio em relação à solução ótima, para cada grupo de instâncias, desconsiderando as instâncias onde os resultados foram sempre os mesmos. É possível observar que, de maneira geral, conjuntos elite menores tendem a obter resultados melhores e que, à medida que o tamanho do conjunto elite aumenta, os resultados tendem a piorar.

A Figura 8, que mostra a soma dos desvios médios em relação à solução ótima de cada instância, indica que o DM-D5 é mais sensível à variação no tamanho do conjunto elite, sendo significativamente mais eficiente com tamanhos de conjunto elite menores. À medida que se aumenta o tamanho do conjunto elite, é possível observar uma tendência de afastamento em relação aos valores ótimos. Isto ocorre porque o conjunto elite demora mais a se estabilizar,

fazendo com que a mineração de dados seja executada mais tarde, causando um menor aproveitamento da fase híbrida, que explora regiões mais promissoras do espaço de busca.

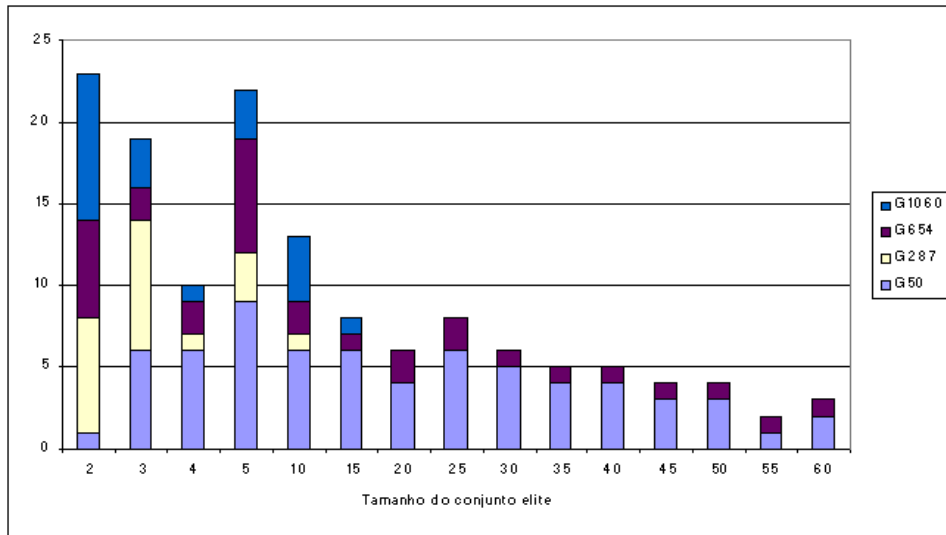


Figura 7: Número de vezes que cada tamanho de conjunto elite alcançou a melhor solução (DM-D5).

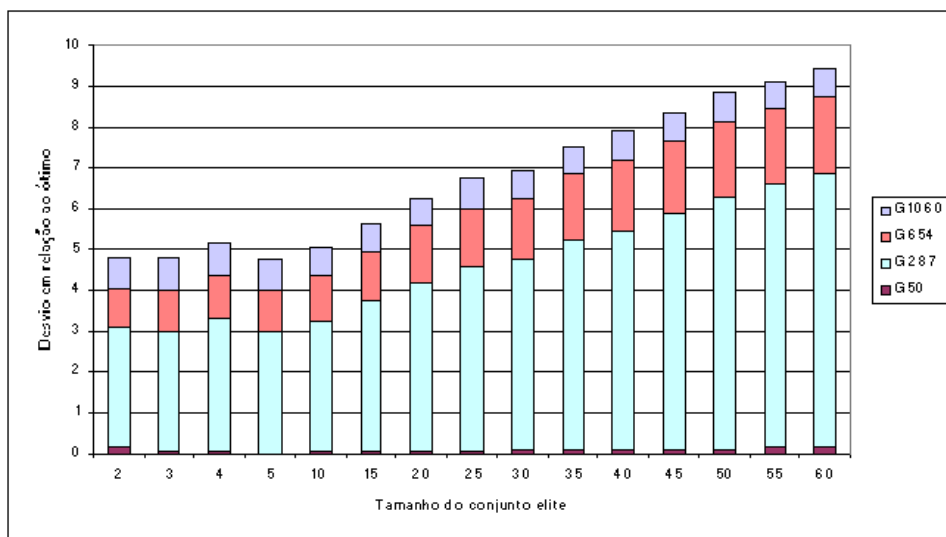
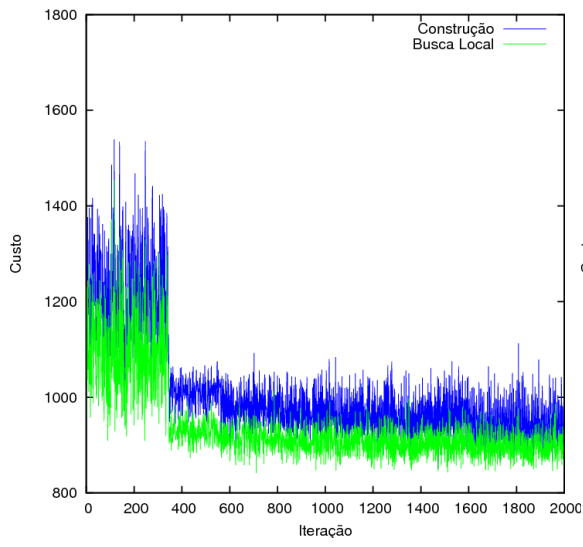


Figura 8: Diferença em relação à solução ótima para cada tamanho de conjunto elite (DM-D5).

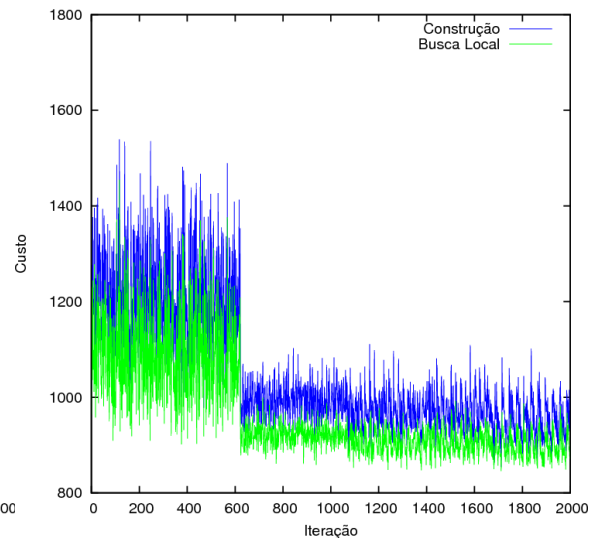
A Figura 9 ilustra a variação no custo das soluções obtidas nas fases de construção e busca local, ao longo das iterações de uma execução do DM-D5 para a instância G287,  $p = 70$  e conjunto elite de tamanho: (a) 2, (b) 10, (c) 35 e (d) 60. É possível notar que, quanto menor o conjunto elite, mais cedo este se torna estável, e como consequência, a fase híbrida é

antecipada, o que beneficia a estratégia, dado que na fase híbrida são exploradas regiões mais promissoras do espaço de busca. Além disso, a rápida estabilização dos conjuntos elite faz com que a identificação de padrões seja processada com maior frequência ao longo da execução. Na instância exibida na Figura 9, a execução com conjunto elite de tamanho 2 (a) processou cinco vezes o algoritmo de mineração, ao passo que com conjunto elite de tamanho 10 (b) foram três execuções. Já com os tamanhos 35 (c) e 60 (d), o algoritmo de mineração foi acionado apenas uma vez.

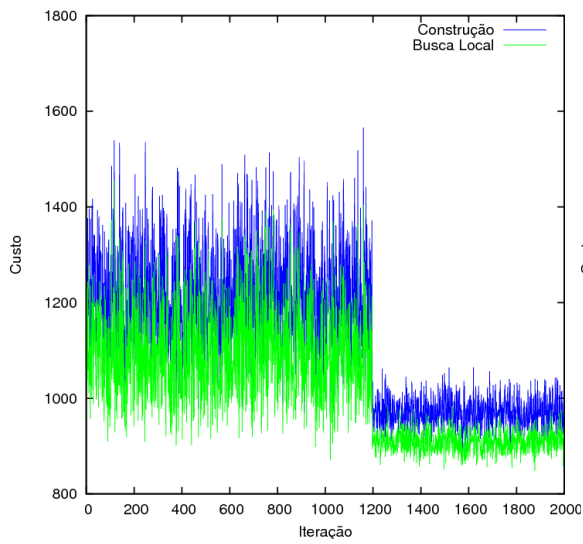
A Tabela 2 apresenta os tempos computacionais das execuções do DM-D5 com os diferentes tamanhos de conjunto elite, e está organizada da mesma forma que a Tabela 1. Nota-se um aumento mais regular nos tempos computacionais conforme o tamanho do conjunto elite aumenta, em relação ao que foi observado na Tabela 1. Tal fato ocorre porque, além do aumento no tempo gasto na execução do algoritmo de mineração de dados conforme o tamanho do conjunto elite aumenta, existe uma redução no tempo necessário para se executar cada iteração da fase híbrida, quando comparada com as iterações da fase de geração do conjunto elite. Esta redução ocorre por dois motivos: primeiramente, o tempo gasto na construção de uma solução é menor, pois esta tem como ponto de partida um padrão e, portanto, menos elementos precisam ser selecionados e adicionados à solução quando esta está sendo gerada. Segundo, as soluções geradas na fase de construção com padrões são, em geral, melhores, o que reduz o esforço necessário para a fase de busca local encontrar um ótimo local. Estes resultados foram publicados em [1].



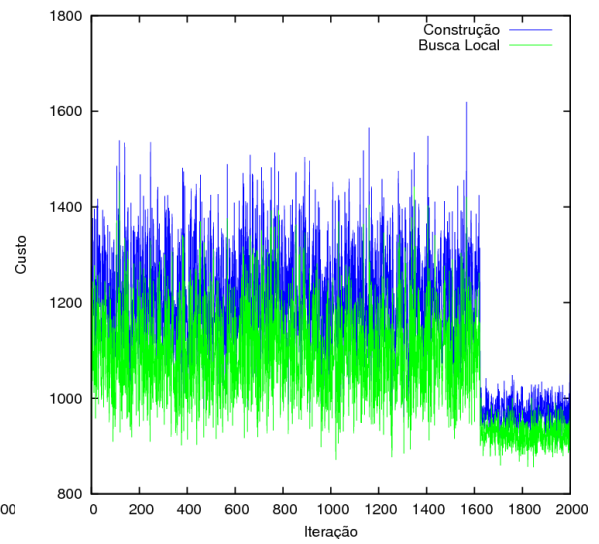
(a) 2



(b) 10



(c) 35



(d) 60

Figura 9: Execução do DM-D5 com conjunto elite de tamanho: (a) 2, (b) 10, (c) 35 e (d) 60.



Tamanho do conj. elite	G50	G287	G654	G1060	Total
2	6,26	425,09	2363,70	6483,03	9278,08
3	7,09	463,18	2571,54	7085,80	10127,61
4	7,54	489,07	2700,01	7522,73	10719,36
5	7,77	476,92	2622,34	7214,05	10321,08
10	7,70	506,97	2682,68	7319,72	10517,06
15	8,33	554,83	2837,36	7723,16	11123,67
20	8,96	619,77	2966,46	8134,38	11729,57
25	9,90	711,05	3116,47	8552,26	12389,69
30	10,76	913,17	3345,07	9074,30	13343,30
35	11,89	1154,91	3554,75	9513,86	14235,41
40	12,65	1395,04	3824,96	9881,66	15114,30
45	13,69	1837,57	4169,03	10469,37	16489,66
50	14,15	2515,24	4498,84	10883,86	17912,09
55	14,72	3248,42	4963,31	11384,96	19611,41
60	15,65	4219,59	5312,33	11781,48	21329,05

Tabela 2: Tempos computacionais do DM-D5 (em segundos).

## CAPÍTULO 4 - CONCLUSÕES

De acordo com os resultados experimentais, considerando-se a qualidade das soluções obtidas, o DM-GRASP mostrou-se sensível às variações no tamanho do conjunto elite, porque apesar dos desvios dos resultados obtidos em relação aos valores ótimos serem bastante similares para todos os tamanhos de conjunto elite avaliados, o conjunto elite de tamanho 10 proporciona melhores resultados.

O DM-D5 mostrou-se bastante sensível às variações do tamanho do conjunto elite. À medida que estes tamanhos aumentaram, a qualidade das soluções obtidas foi significativamente degradada. Surpreendentemente, conjuntos elite com apenas duas soluções levaram aos melhores resultados. Isto se deve ao fato de que, quanto menor o conjunto elite, mais rapidamente este se estabiliza, o que traz dois benefícios: a fase híbrida, que explora regiões mais promissoras do espaço de busca, é antecipada e a identificação de padrões é realizada com maior frequência, o que possibilita o uso de padrões mais refinados.

Foi observado também, tanto para o DM-GRASP quanto para o DM-D5, que conjuntos elite maiores tornam mais complexa a execução do algoritmo de mineração de dados, e que, a partir de certo ponto, o tempo gasto na identificação de padrões causa um aumento considerável para o tempo total de execução da heurística, o que evidencia que grandes conjuntos elite devem ser evitados.

Os resultados apresentados neste trabalho foram obtidos no contexto de um problema de otimização combinatória específico, o problema das  $p$ -medianas e, portanto, como trabalhos futuros, os mesmos experimentos poderiam ser realizados para problemas com características distintas de forma a verificar se os resultados observados neste trabalho, no comportamento das heurísticas, podem ser generalizados.

## REFERÊNCIAS

- [1] T. W. Barra, L. F. M. Santos, R. Fuchshuber, A. Plastino e S. L. Martins. Explorando a Heurística DM-GRASP Para o Problema das p-Medianas. *Anais do XLI SBPO*, 2009.
- [2] P. Festa e M. Resende. GRASP: An annotated bibliography. *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, 325-367, 2002.
- [3] M. D. H. Gamal e S. Salhi. A cellular heuristic for the multisource Weber problem. *Computers & Operations Research*, 30, 1609-1624, 2003.
- [4] G. Grahne e J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.
- [5] J. Han e M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishing, 2006.
- [6] O. Kariv e L. Hakimi. An algorithmic approach to network location problems, part ii: The p-medians. *SIAM Journal of Applied Mathematics*, 37, 539-560, 1979.
- [7] A. Plastino, E. R. Fonseca, R. Fuchshuber, S. L. Martins, A. A. Freitas, M. Luis e S. Salhi. A Hybrid Data Mining Metaheuristic for the p-Median Problem. *SIAM Journal on Applied Mathematics*, 305-316, 2009.
- [8] M. Resende e C. Ribeiro. Greedy Randomized Adaptive Search Procedures, em F. Glover e G. A. Kochenberger (Eds.), *Handbook of Metaheuristics*, 219-249, 2003.
- [9] M. H. Ribeiro, A. Plastino e S. L. Martins. Hybridization of GRASP Metaheuristic with Data Mining Techniques. *Journal of Mathematical Modeling and Algorithms*, 23-41, 2006.
- [10] M. H. F. Ribeiro, V. F. Trindade, A. Plastino, e S. L. Martins. Hybridization of GRASP metaheuristic with data mining techniques. *Proceedings of the Workshop on Hybrid Metaheuristics, in conjunction with the 16th European Conf. on Artificial Intelligence*, 69-78, 2004.
- [11] L. F. M. Santos, R. Milagres, C. Albuquerque, A. Plastino e S. L. Martins. A Hybrid GRASP with Data Mining for Efficient Server Replication for Reliable Multicast. *Proceedings of the IEEE GLOBECOM Conference*, 2006.

- [12] L. F. M. Santos, A. Plastino e S. L. Martins. Applications of the DM-GRASP Heuristic: A Survey. *International Transactions in Operational Research*, 15, 387-416, 2008.
- [13] L. F. M. Santos, M. H. Ribeiro, A. Plastino e S. L. Martins. A Hybrid GRASP with Data Mining for the Maximum Diversity Problem. *Proceedings of the International Workshop on Hybrid Metaheuristics*, 69-78, 2005.
- [14] B. C. Tansel, R. L. Francis e T. J. Lowe. Location on networks: A survey. *Management Science*, 29, 482-511, 1983.