

Algoritmos *Multi-Start*, GRASP e ILS Aplicados ao Problema de P-Medianas

Gustavo Zeferino¹, Flaviana M. de S. Amorim¹,
Moacir F. de F. Filho¹

¹Programa de Pós-Graduação em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
30510-000 – Belo Horizonte, MG – Brasil

zeferino@lsi.cefetmg.br, flaviana@dppg.cefetmg.br, franca@des.cefetmg.br

Abstract. *This work treats the p -Median Problem by the Metaheuristics ILS, GRASP and Multi-Start using as local search method the Fast Swap-Based algorithm proposed by [Resende (2003)]. The tests were conducted with instances Galvão and OR-Library. The results showed that ILS algorithm is the best in relation to the time of execution and the final solution quality.*

KEYWORDS. *p -median Problem. ILS. GRASP. Multi-Start.*

Resumo. *Este trabalho trata o Problema de p -Medianas por meio das Metaheurísticas ILS, GRASP e Multi-Start utilizando, como método de busca local, o algoritmo Fast Swap-based proposto por [Resende (2003)]. Foram realizados testes com as instâncias Galvão e OR-Library. Os resultados mostram que o algoritmo ILS apresenta o melhor desempenho em termos de tempo de execução e qualidade da solução.*

PALAVRAS CHAVE. *Problema de p -Medianas. ILS. GRASP. Multi-Start.*

1 Introdução

Neste trabalho será considerado o Problema de p -Medianas não-capacitado, em que o objetivo é escolher, dentre n nós de um grafo, um conjunto de p nós, denominado medianas, tal que $p \leq n$, de modo a minimizar a soma das distâncias de cada nó restante até o nó mediana mais próximo. Na versão capacitada deste Problema, a cada nó do grafo é associado um peso de demanda a ser satisfeito pela mediana escolhida. Neste caso, a soma das demandas de todos os nós cobertos por uma mediana não deve ultrapassar a capacidade de atendimento da mesma [Lorena (2001)]. Os estudos sobre o Problema de p -Medianas (PPM) têm registro na literatura a partir do início da década de 60 [Hakimi (1964)]. Exemplos de aplicações do PPM podem ser visto em diversos trabalhos, como: sistemas de distribuição [Klose (2005)], sistemas de informações geográficas [Lorena (2001)], atendimento de clientes de uma rede com demanda probabilística [Berman (2010)] e localização de unidades de atendimento [Abreu (2008)], dentre outros. Uma revisão a respeito do Problema de p -Medianas pode ser encontrada em [Mladenovic (2007)].

Pode ser observado que existem $\frac{n!}{p!(n-p)!}$ soluções para o PPM, o que o torna inviável de ser resolvido por métodos de enumeração completa para valores grandes de n , sendo n o número total de vértices do grafo considerado e p a quantidade de medianas a serem escolhidas. Segundo [Kariv (1979)], o Problema de p -Medianas é um problema da classe NP-Difícil e, portanto, heurísticas são utilizadas para resolvê-lo.

Este artigo é estruturado como segue. Na seção 2, é apresentada a definição e uma formulação do problema. Na seção 3, apresenta-se a metodologia adotada para estrutura de dados utilizada e os algoritmos implementados para solucionar o problema de interesse. Na seção 4, são mostrados os testes realizados com estes algoritmos utilizando duas classes de instâncias da literatura, sendo também realizada uma comparação de desempenho entre estes métodos. Por último, é feita a conclusão do trabalho na seção 5.

2 Definição do Problema

Seja $G = (V, A)$ um grafo não direcionado ponderado, em que V é o conjunto dos vértices do grafo e A é o conjunto das arestas, tendo associado a cada aresta o valor da distância (ou custo de viagem) entre dois vértices adjacentes. O objetivo do problema é encontrar p vértices, de forma a minimizar a distância entre os p vértices selecionados e os vértices restantes. Cada vértice restante é ligado ou associado a somente um dos p vértices escolhidos, sendo este o vértice mais próximo, denominado como mediana.

2.1 Formulação matemática

O Problema de p -Medianas pode ser formulado, segundo [Mladenovic (2007)], como:

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (1)$$

restrito a:

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i, \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i, j, \quad (3)$$

$$\sum_{j=1}^n y_j = p, \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\}. \quad (5)$$

em que:

- n representa o número de vértices;
- p representa o número de medianas que serão escolhidas;
- d_{ij} é a distância entre os vértices i e j ;
- x_{ij} assume 1 se o i -ésimo vértice é atendido pelo vértice j e 0, caso contrário.
- y_j assume 1 se o j -ésimo vértice for um dos p selecionados e 0, caso contrário.

O conjunto de restrições (2) garante que cada vértice é atendido por um dos p vértices que representam as medianas. As restrições (3) previnem que um vértice

seja atendido por outro que não seja mediana. O total de medianas é definido como p pela restrição (4). O conjunto de restrições (5) define o tipo de variáveis de decisão binária do problema.

3 Metodologia de Solução

Esta seção apresenta os procedimentos propostos para a solução do Problema de p -Medianas. Inicialmente, é mostrada a estrutura de dados utilizada para a representação de uma solução. Em seguida, são apresentadas as três Metaheurísticas implementadas para a solução do PPM, a saber, *Iterated Local Search* (ILS), *Greedy Randomized Adaptive Search Procedure* (GRASP) e *Multi-Start*. Por fim, é apresentada a técnica de busca local utilizada como heurística de refinamento das Metaheurísticas desenvolvidas.

3.1 Representação da Solução

Para representar uma solução, foi definida uma matriz com n colunas e 3 linhas. A primeira linha contém os índices dos vértices, sendo que as p primeiras posições representam os índices dos vértices que são classificados como medianas. Cada configuração destes p primeiros índices, desconsiderando ordem, representa uma possível solução para o problema. A segunda linha armazena o índice da mediana que atende o vértice com índice definido na primeira linha e a terceira linha armazena a distância entre a mediana e o vértice.

Para explorar o espaço de soluções, foi utilizado um movimento de troca, que consiste em trocar um vértice que é mediana por um que não o seja. A Figura 1 ilustra este movimento.

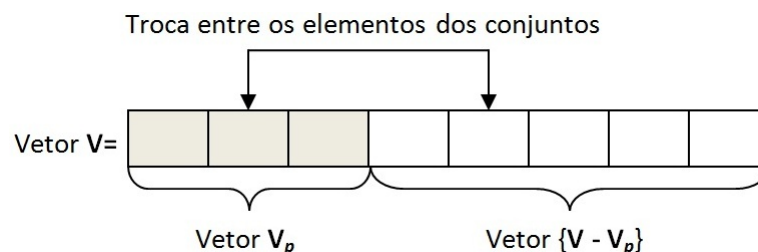


Figure 1. Movimento de troca em que um elemento do conjunto das p -medianas é trocado por um elemento do conjunto dos que não são medianas.

3.2 Metaheurísticas

Foram propostos três algoritmos baseados em Metaheurísticas, em que são detalhados nas seções seguintes.

3.2.1 ILS

A Metaheurística *Iterated Local Search* – ILS [Lourenço (2003)] consiste em partir de uma solução inicial s_0 , previamente obtida a partir da utilização de um algoritmo de construção; aplicar um procedimento de busca local a essa solução; e, para escapar do ótimo local gerado, aplicar perturbações nesse ótimo local. A Figura 2 esquematiza o

funcionamento de um ILS básico. No algoritmo ILS implementado, a solução inicial é o melhor resultado obtido dentre 20 iterações da fase de construção GRASP, descrita na seção 3.2.2.

Em seguida, foi aplicado a fase de refinamento do ILS. A busca local utilizada é a descrita na seção 3.3. Para modificar a solução s obtida, são feitas perturbações na mesma, gerando-se as soluções perturbadas s' . Sobre essas soluções, é aplicado o procedimento de refinamento, gerando um ótimo local s'' . A decisão sobre qual solução será aplicada a próxima perturbação é feita pelo critério de aceitação. O critério de aceitação adotado é o de melhora no valor da solução corrente s , isto é, uma solução s'' é aceita para ser a nova solução s corrente se $f(s) < f(s'')$.

No ILS, a intensificação é obtida em perturbações feitas na solução corrente. Já a diversificação é obtida quando se aceita qualquer solução s'' e são aplicadas perturbações maiores na solução ótima corrente. O êxito deste método está diretamente associado à definição do procedimento de busca local, do procedimento de perturbação aplicado à solução atual e do critério de aceitação das soluções. Uma perturbação no algoritmo implementado consiste em executar um movimento de troca aleatória entre uma mediana alocada (ou “aberta”) e uma mediana candidata (ou “fechada”). Além disso, as perturbações são executadas em níveis, isto é, para cada nível i de perturbação, são realizadas *Nivel_iter* iterações em relação à solução ótima local corrente. Este valor foi fixado em *Nivel_iter* = 20. As perturbações são realizadas de *Nivel_min_per* a *Nivel_max_per* níveis. Neste trabalho, estes valores foram fixados em *Nivel_min_per* = 3 e *Nivel_max_per* = 10.

Procedimento ILS

```
 $s_0 \leftarrow$  Solução Inicial;  
 $s \leftarrow$  Busca Local;  
iter  $\leftarrow$  0;  
Enquanto (iter <  $iter_{max}$ ) faça:  
    iter  $\leftarrow$  iter + 1;  
     $s' \leftarrow$  perturbação ( $s$ , histórico);  
     $s'' \leftarrow$  Busca Local ( $s'$ );  
     $s \leftarrow$  Critério de aceitação ( $s$ ,  $s'$ ,  $s''$ );  
Fim_Enquanto;  
Retorne  $s$ .
```

Figure 2. Representação do ILS.

3.2.2 GRASP

Greedy Randomized Adaptive Search Procedure – GRASP [Resende (1995)] é uma Metaheurística que consiste na aplicação iterativa de duas fases: construção e refinamento, retornando a melhor das soluções obtidas ao longo da busca.

Na fase de construção, é gerada uma solução parcialmente gulosa por meio de uma função guia. A aleatoriedade da construção é controlada por um parâmetro real $\alpha \in [0, 1]$. Para $\alpha = 1$, tem-se uma solução totalmente aleatória; para $\alpha = 0$, tem-se uma solução gulosa. Após a construção da solução, é aplicado o método de busca

local apresentado na seção 3.3, para refinar a solução construída.

A construção de uma solução no método GRASP consiste em inserir elementos, obedecendo a um valor calculado pela função guia $g(\cdot)$ e a uma regra de seleção que contém um fator aleatório. Para o PPM, este método consiste em selecionar somente p vértices para serem medianas.

Para cada inserção, é definida uma lista C com os vértices remanescentes. Para cada vértice, é calculado um valor, através da função guia $g(t)$, que consiste no somatório das distâncias do vértice $t \in C$ para todos os demais vértices $i \in C \setminus \{t\}$. Alguns destes vértices são selecionados para uma outra lista, chamada de Lista Restrita de Candidatos (LRC), pela seguinte regra:

$$LRC = \{t \in C | g(t) \leq g(t_{min}) + \alpha(g(t_{max}) - g(t_{min}))\}$$

em que $t_{min} = \arg \min_{t \in C} g(t)$, $t_{max} = \arg \max_{t \in C} g(t)$ e α é um valor real no intervalo $[0, 1]$ definido pelo usuário. Com a LRC construída, é calculado o valor de aptidão para cada um dos vértices pertencentes a ela. Os parâmetros utilizados foram definidos de forma empírica após uma bateria preliminar de testes, e foram os seguintes: $\alpha = 0,3$. A aptidão de um elemento t da lista LRC é calculada pela expressão:

$$f(t) = \frac{1/g(t)}{\sum_{t \in LRC} 1/g(t)} \quad (6)$$

A aptidão representa a probabilidade do vértice t da lista LRC ser escolhido como mediana. Desta forma, é gerado uma variável aleatória e com ela é feito um sorteio para a definição de qual vértice será escolhido como mediana. Todo o processo é repetido até que todas as p medianas sejam escolhidas.

Após a construção, a solução gerada por este procedimento é melhorada pela heurística de refinamento descrita na seção 3.3.

Esse processo de construção e refinamento é aplicado 10 vezes, retornando-se a melhor das soluções obtidas. Os parâmetros utilizados em todas as Metaheurísticas foram definidos de forma empírica, após uma bateria preliminares de testes. Para o GRASP, o valor que apresentou um melhor resultado para a função guia foi o $\alpha = 0,3$.

3.2.3 Multi-Start

Multi-Start [Martí (2003)] é uma Metaheurística simples, consistindo em gerar soluções aleatórias e melhorá-las, por meio de uma heurística de refinamento. No nosso caso, foi adotada a referenciada na seção 3.3. A solução inicial é gerada de forma aleatória com distribuição uniforme. Cada vez que uma melhor solução é encontrada, ela é armazenada.

Como critério de parada, foi adotado o número de 100 iterações sem melhora, ou seja, o método para quando a melhor solução se mantém inalterada por 100 iterações.

3.3 Busca local

Para este trabalho, foi implementada a busca local que realiza sempre a melhor troca possível e que repete este processo até que não exista mais alguma troca que melhore o valor da função objetivo (distância entre as medianas e os vértices). Esta heurística é conhecida como Método da Descida (ou *Best Improvement Method*). Neste trabalho, foi implementada uma versão deste método, denominada *Fast Swap-based* [Resende (2003)]. Esta heurística calcula o ganho, para cada vértice que não é mediana, caso esta se torne uma mediana, e calcula, para cada mediana, a perda, caso esta deixe de ser mediana. Além disto, é calculado um fator que considera o impacto total na solução caso se realize uma determinada troca. Com estes valores, é possível estimar o ganho de cada troca e efetivar aquela que possui o maior ganho. A grande vantagem deste método é que não é necessário realizar a troca em si para estimar esses valores. Um dos motivos é que as informações calculadas não são recalculadas a cada iteração e, sim, somente atualizadas.

A heurística, *Fast Swap-based*, foi implementada especificamente para o Problema das p -Medianas [Resende (2003)], ela se baseia no algoritmo proposto em 1968 por Teitz e Bart, em que são realizadas trocas entre as medianas e os clientes. Esta nova implementação é significativamente mais rápida [Resende (2003)].

4 Resultados

Os algoritmos ILS, GRASP, *Multi-Start* foram implementados na linguagem C++ e testados em um computador com processador Pentium Intel(R) Core(TM)2 Quad Q8400, com *clock* de 2.66 GHz, 3,7 GB de RAM, Kernel Linux 2.6.32-30-generic, compilador GCC versão 4.4.3 e sistema operacional Ubuntu 10.04 64-bits. Para testar os algoritmos, foram usadas dois tipos de instâncias: *Galvão*[Alp (2003)] e *OR-Library* [Beasley (1985)]. Para cada instância, cada algoritmo foi executado 50 vezes.

As tabelas em que são apresentados os resultados obtidos, têm a coluna *Instância* que é subdividida em quatro outras colunas: a subcoluna *Nome*, que expressa o nome da instância; a subcoluna n , que representa o número de vértices contido na instância; a subcoluna p , representando os números de medianas que devem ser instaladas; e a subcoluna *Ótimo*, na qual está o valor ótimo encontrado na literatura correspondente a cada instância. Nas outras três colunas, são apresentados os resultados obtidos para cada Metaheurística implementada *ILS*, *GRASP* e *Multi-Start*, com suas respectivas subcolunas *Melhor*, que lista o melhor resultado encontrado; subcoluna *Erro %*, que informa a porcentagem com que a média dos valores obtidos ficaram distantes do valor ótimo; e a subcoluna *Tempo*, relatando o tempo médio gasto pelas Metaheurísticas para cada instância, em segundos.

Pela Tabela 1, que contém os resultados da instância *Galvão*, verifica-se que o algoritmo ILS é capaz de encontrar todos os valores ótimos da classe de instâncias analisada. O algoritmo GRASP não alcançou nenhum dos resultados ótimos nas instâncias. O algoritmo *Multi-Start*, por sua vez, não alcançou o resultado ótimo apenas na instância G150-45. Na Tabela 2, em que está os resultados da instância *OR-Library*, é visto que o algoritmo ILS é capaz de encontrar todos os valores ótimos da classe de instâncias analisada. O algoritmo GRASP não alcançou o resultado ótimo nas instâncias pmed25, pmed30 e pmed40. O algoritmo *Multi-Start*, por sua vez, não alcançou o resultado ótimo apenas na instância pmed30.

Com relação aos valores de erro médio encontrados, a Metaheurística ILS apresenta valores menores ou igual a 0,01% para 9 das instâncias *Galvão* e para 25 das instâncias *OR-Library*; entre 0,01% e 0,1% é encontrado 6 e 14 para as instâncias *Galvão* e *OR-Library*, respectivamente; e maior que 0,1% para 1 em ambas instâncias. Quanto à Metaheurística GRASP, os valores de erro médio menores ou igual a 0,01% são encontrados em 32 instâncias *OR-Library*; entre 0,01% e 0,1% para 6 instâncias *OR-Library*; e maior que 0,1% para 2 instâncias *OR-Library*. Por último, na Metaheurística *Multi-Start*, os valores de erro médio menores ou igual a 0,01% são encontrados em 6 nas instâncias *Galvão* e 32 nas instâncias *OR-Library*; entre 0,01% e 0,1% para 9 e 6 nas instâncias *Galvão* e *OR-Library*, respectivamente; e maior que 0,1% para 1 e 2 nas respectivas instâncias, *Galvão* e *OR-Library*.

Com relação ao tempo, nas instâncias *Galvão*, as três Metaheurísticas alcançaram o valor rapidamente para as instâncias com $p = 5$ e para as instâncias *OR-Library*, as mesmas alcançaram o valor rapidamente para as instâncias com $n = 100$. O ILS obteve o menor tempo na instância G100-15 na instância *Galvão* e na pmed4 da *OR-Library*, com o tempo de 0,1280 e 0,0804 segundos, respectivamente. O GRASP obteve em 0,1088 segundos o valor ótimo da instância pmed1. Já a Metaheurística *Multi-Start* obteve seu menor tempo de 0,2278 e 0,1978 segundos para alcançar o valor ótimo também da instância pmed1 e da G100-5, respectivamente. O maior tempo de execução pela ILS é visto na instância G150-60 e pmed38, em que ele despende 0,4670 e 10,3790 segundos, respectivamente. O maior tempo de execução pela GRASP é visto na instância pmed30, com $n = 600$, em que demanda 105,0686 segundos. Para a Metaheurística *Multi-Start*, o maior tempo de execução é visto na instância G150-60 e pmed40, com 3,0018 e 442,2752 segundos, respectivamente.

De forma a comparar esses algoritmos com relação ao tempo necessário para encontrar um valor alvo, foram feitos experimentos, segundo a abordagem indicada em [Aiex (2002)] em que são gerados gráficos da função de distribuição acumulada empírica. O gráfico representa o tempo versus probabilidade, ou seja, o tempo gasto da instância para alcançar o valor alvo determinado, isso, em percentual. Para execução dos experimentos, utilizou-se a instância G150-45 *Galvão* e a instância pmed30 *OR-Library*, que tem seus valores ótimos, são respectivamente 4637 e 1989, salientando que, apenas uma instância não é suficiente para comparar o desempenho do método dentre todas as instâncias do conjunto, porém, foi escolhida a instância com o maior erro percentual, visto que, comparar todas as instâncias se torna inviável. Cada algoritmo foi executado 50 vezes, sendo que em cada rodada é interrompido após alcançar o valor alvo, respectivamente, 4640 e 2000. Não foram permitidos tempos de execução repetidos; assim, os tempos repetidos foram descartados e uma nova execução foi feita. Após determinados os tempos para as 50 execuções, estes foram ordenados de forma crescente e, para cada tempo t_i , foi associada uma probabilidade $p_i = \frac{t_i - 0,5}{50}$. Os resultados do gráfico $t_i \times p_i$ são apresentados nas Figuras 4 e 3.

Nestas figuras 3 e 4, verifica-se que o algoritmo ILS é o que alcança o valor alvo mais rapidamente em ambas instâncias em estudo. Na instância *Galvão*, enquanto o valor alvo é alcançado quase que instantaneamente pelo ILS, esse mesmo alvo não é alcançado pelo GRASP e pelo *Multi-Start* é alcançado em quase 9 segundos. Este resultado é corroborado pelos valores apresentados na Tabela 1 para a mesma instância, em que o ILS alcança o valor ótimo em 0,3800 segundos, tendo um erro médio de

0,0362%. O mesmo comportamento é visto na instância *pmed30 OR-Library*, em que o alvo só é alcançado pelo GRASP em cerca de 250 segundos e pelo *Multi-Start* em quase 800 segundos. Este resultado é também corroborado pelos valores apresentados na Tabela 2 para a mesma instância, uma vez que a Metaheurística ILS alcançou, em 50 execuções, o valor ótimo em 8,9566 segundos, tendo um erro médio de 0,1528%. As demais Metaheurísticas não lograram alcançar valores ótimos no mesmo número de execuções, tendo, ainda, valores de tempo significativamente superiores em ambas instâncias analisadas neste trabalho.

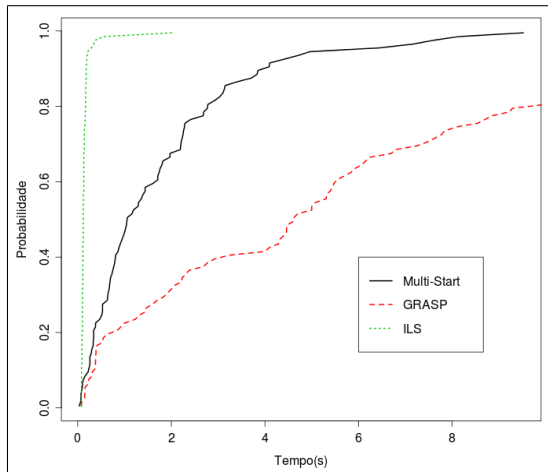


Figure 3. Time-to-target da instância G150-45 com alvo igual a 4640.

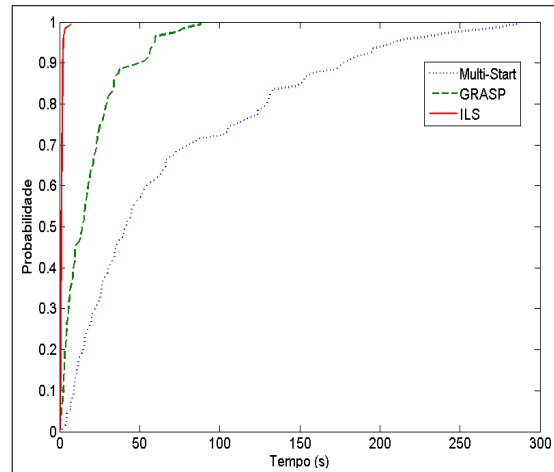


Figure 4. Time-to-target da instância pmed30 com alvo igual a 2000.

5 Conclusões

Este trabalho tratou o Problema de p -Medianas não-capacitado por meio de três Metaheurísticas: ILS, GRASP e *Multi-Start*.

A Metaheurística ILS implementada utiliza, para geração da solução inicial, a fase de construção da Metaheurística GRASP. Apresentou resultados superiores aos encontrados para as implementações de *Multi-Start* e o GRASP, alcançando valores ótimos em 100% 16 conjuntos de instâncias *Galvão* e também nos 40 conjuntos de instâncias *OR-Library*. O maior valor de erro percentual encontrado é 0,1947 para o conjunto G150-25 nas instâncias *Galvão* e, 0,1528 no conjunto *pmed30* nas instâncias *OR-Library*. O valor médio de erro médio, dentre os 16 conjuntos de instâncias *Galvão* é de 0,0357 e, nos 40 conjuntos das instâncias no *OR-Library* é de 0,0200.

A Metaheurística GRASP consiste na aplicação iterativa de duas fases, construção e refinamento, retornando a melhor das soluções obtidas ao longo da busca. Nas instâncias *Galvão* não atingiu nenhum valor ótimo, porém, nos 40 conjuntos de instâncias *OR-Library* alcançou valores ótimos em 92,5%. O maior erro percentual nas respectivas instâncias é de 18,5959 e 0,3680, na instância G150-20 e na instância *pmed30*, tendo valor médio de erro médio, dentre todas as instâncias, respectivamente, dado por 11,3065 e 0,0213.

A Metaheurística *Multi-Start*, mesmo sendo uma Metaheurística simples, em que sua solução inicial, neste trabalho, é gerada de forma aleatória, com distribuição uniforme, conseguiu alcançar, dentre os 16 conjuntos de instâncias *Galvão*, 93,75

dos valores ótimos e nos 40 conjuntos das instâncias *OR-Library*, 97,5%. Esta Meta-heurística apresenta o maior erro percentual, no valor de 0,2151 na instância G150-25 e de 0,3952 na instância pmed30, tendo valor médio de erro médio, dentre todas as instâncias *Galvão*, dado por 0,0388 e nas instâncias *OR-Library* é 0,0210.

Estes resultados mostram a superioridade da implementação realizada para a Metaheurística ILS sobre a implementação das Metaheurísticas GRASP e *Multi-Start*. Esta afirmativa se baseia no melhor desempenho no tocante ao número de valores ótimos alcançados, em que é seguida pela *Multi-Start* e, por fim, pela GRASP. Outro parâmetro de análise que mostra a superioridade da ILS é ter alcançado o menor maior valor de erro percentual, seguido pela *Multi-Start* e, após, pela GRASP para as instâncias *Galvão*. Nas instâncias *OR-Library* o ILS é seguido pelo valor alcançado pela GRASP e, após, pela *Multi-Start*. No tocante ao tempo médio de execução, a Metaheurística ILS novamente se mostra superior, alcançando o valor médio de tempo médio igual a 0,2427 segundos, das 16 instâncias *Galvão* e 3,0025 segundos, das 40 instâncias *OR-Library*. A superioridade da ILS é corroborada, por fim, pelos gráficos *time-to-target* apresentados nas Figuras 4 e 3, em que o ILS alcança o valor alvo em tempo inferior ao dos demais métodos.

A principal vantagem da ILS e da *Multi-Start* para as instâncias *Galvão* é o menor tempo de processamento e a simplicidade da implementação, respectivamente. Para as instâncias *OR-Library* a principal vantagem do GRASP reside na baixa variabilidade das soluções finais; do ILS, no menor tempo de processamento requerido. É interessante observar que, nas instâncias pmed20, pmed24, pmed37 e pmed40, o erro médio do ILS é maior que das demais Metaheurísticas.

Agradecimentos

Os autores agradecem ao CEFET-MG, à CAPES, à FAPEMIG e ao CNPq pelo apoio ao desenvolvimento deste trabalho, bem como ao Prof. Marcone Jamilson Freitas Souza, pela co-orientação no desenvolvimento do mesmo.

References

- Abreu, L. F., Rosário, R. R. L., and Coelho, A. S. (2008). Localização de unidades de atendimento ao cidadão: Comparação e proposta para a cidade de Manaus, utilizando o algoritmo de taitz e bart e algoritmo genético. In *Anais do XXVIII ENEGEP*, Rio de Janeiro. ABEPRO.
- Aiex, R. M., Resende, M. G. C., and Ribeiro, C. C. (2002). Probability distribution of solution time in GRASP: An experimental investigation. *Journal of Heuristics*, 8:343–373.
- Alp, O., Erkut, E., and Drezner, D. (2003). An efficient genetic algorithm for the p-median problem. *Annals of Operations Research*, 122:21–42.
- Beasley, J. (1985). A note on solving large p-median problems. *European Journal of Operational Research*, 31:270–273.
- Beasley, L. B. and Pullman, N. J. (1990). Linear operators that strongly preserve commuting pairs of Boolean matrices. 132:137–143.

- Berman, O. and Wang, J. (2010). The network p-median problem with discrete probabilistic demand weights. *Computer & Operation Research*, 37(8):1455–1463.
- Christofides, N. G. t. (1975). An algorithmic approach. *New York: Academic Press Inc.*
- Hakimi, S. L. (1964). Optimal locations of switching centers and the absolute centers and the medians of a graph. *Operations Research*, 12:450–459.
- Kariv, O. and Hakimi, L. (1979). An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560.
- Kariv, O. and Hakimi, S. (1969). An algorithmic approach to network location problems; part 2. the p-medians. *SIAM Journal on Applied Mathematics*, 37:539–560.
- Klose, A. and Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*, 162:4–29.
- Lorena, L. A. N., Senne, E. L. F., Paiva, J. A. C., and Pereira, M. A. (2001). Integração de modelos de localização a sistemas de informações geográficas. *Gestão e Produção*, 8(2):185–195.
- Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, pages 321–353. Kluwer Academic Publishers, Boston.
- Martí, R. (2003). Multi-start methods. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, pages 355–367. Kluwer Academic Publishers, Boston.
- Mladenovic, N., Brimberg, J., Hansen, P., and Moreno-Pérez, J. A. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927 – 939.
- Resende, M. and Werneck, R. (2003). A fast swap-based local search procedure for location problems. Technical Report TD-5R3KBH, AT&T Labs Research.
- Resende, M. and Werneck, R. (2004). A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10:59–88.
- Resende, M. and Werneck, R. (2007). A hybrid heuristic for the p-median problem. <http://www2.research.att.com/~mgcr/talks/gpr-pmedian-informs2007.pdf>. INFORMS Annual Meeting Seattle, Washington.
- Resende, M. G. C. and Feo, T. A. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 9:849–859.
- Teitz, M. B. and Bart, P. (1968). Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16.
- Whitaker, R. A. (1983). A fast algorithm for the greedy interchange of large-scale clustering and median location problems. *INFOR*, 21.

Instância				ILS			GRASP			Multi-Start		
Nome	<i>n</i>	<i>p</i>	Ótimo	Melhor	Erro %	Tempo (s)	Melhor	Erro %	Tempo (s)	Melhor	Erro %	Tempo (s)
G100-5	100	5	5703	5703	0,0000	0,1622	6374	10,7954	0,7486	5703	0,0000	0,1978
G100-10	100	10	4426	4426	0,0000	0,1280	5037	15,6905	1,4000	4426	0,0190	0,3528
G100-15	100	15	3893	3893	0,0041	0,1214	4402	12,7994	1,9722	3893	0,0005	0,4360
G100-20	100	20	3565	3565	0,0067	0,1298	3871	9,8682	2,4914	3565	0,0000	0,5106
G100-25	100	25	3291	3291	0,0043	0,1404	3530	8,1926	2,9564	3291	0,0134	0,7734
G100-30	100	30	3032	3032	0,0033	0,1542	3221	7,2856	3,3764	3032	0,0040	0,7468
G100-35	100	35	2784	2784	0,0036	0,1704	2964	7,0517	3,7252	2784	0,0000	0,7226
G100-40	100	40	2542	2542	0,0008	0,1898	2701	7,0614	4,0310	2542	0,0102	0,9492
G150-5	150	5	10839	10839	0,0000	0,3636	11615	8,5447	1,6540	10839	0,0000	0,4596
G150-15	150	15	7390	7390	0,0831	0,2382	8494	16,2089	4,5780	7390	0,0341	1,0510
G150-20	150	20	6454	6454	0,0930	0,2426	7589	18,5959	5,8902	6454	0,0834	1,3328
G150-25	150	25	5875	5875	0,1947	0,2634	6622	17,9319	7,1036	5875	0,2151	2,0072
G150-35	150	35	5192	5192	0,0597	0,3234	5842	13,2985	9,2752	5192	0,0636	2,3322
G150-45	150	45	4636	4636	0,0362	0,3800	5067	10,1091	11,1550	4637	0,0746	2,5398
G150-50	150	50	4374	4374	0,0393	0,4080	4725	9,3164	11,9912	4374	0,0498	2,8032
G150-60	150	60	3873	3873	0,0418	0,4670	4201	8,1544	13,3404	3873	0,0527	3,0018

Table 1. Resultados obtidos para cada Metaheurística, aplicadas nas instâncias Galvão. Para cada instância e método, foram realizados 50 execuções. A coluna Melhor lista o melhor resultado encontrado, sendo que os que atingiram o ótimo estão em negrito. A coluna Erro % informa a porcentagem com que a média dos valores obtidos ficou distante do valor ótimo e a coluna Tempo relata o tempo médio gasto pelas Metaheurísticas para cada instância, em segundos.

Instância				ILS			GRASP			Multi-Start		
Nome	<i>n</i>	<i>p</i>	Ótimo	Melhor	Erro %	Tempo (s)	Melhor	Erro %	Tempo (s)	Melhor	Erro %	Tempo (s)
pmed1	100	5	5819	5819	0,0000	0,1260	5819	0,0000	0,1088	5819	0,0000	0,2278
pmed2	100	10	4093	4093	0,0000	0,0884	4093	0,0000	0,1288	4093	0,0000	0,3044
pmed3	100	10	4250	4250	0,0000	0,0844	4250	0,0000	0,129	4250	0,0000	0,3176
pmed4	100	20	3034	3034	0,0000	0,0804	3034	0,0000	0,1878	3034	0,0000	0,4766
pmed5	100	33	1355	1355	0,0000	0,1048	1355	0,0000	0,2716	1355	0,0000	0,6758
pmed6	200	5	7824	7824	0,0000	0,4680	7824	0,0000	0,4472	7824	0,0000	0,8916
pmed7	200	10	5631	5631	0,0000	0,2794	5631	0,0000	0,4742	5631	0,0000	1,1858
pmed8	200	20	4445	4445	0,0000	0,2656	4445	0,0000	0,7264	4445	0,0000	1,8762
pmed9	200	40	2734	2734	0,0885	0,3304	2734	0,0000	1,3588	2734	0,0000	3,3464
pmed10	200	67	1255	1255	0,0000	0,4774	1255	0,0016	2,3238	1255	0,0000	5,2206
pmed11	300	5	7696	7696	0,0000	1,0174	7696	0,0000	0,9036	7696	0,0000	1,9640
pmed12	300	10	6634	6634	0,0000	0,6950	6634	0,0000	1,1124	6634	0,0000	2,8190
pmed13	300	30	4374	4374	0,0000	0,6026	4374	0,0000	2,1854	4374	0,0000	5,6540
pmed14	300	60	2968	2968	0,0121	0,9034	2968	0,0007	4,9838	2968	0,0013	12,4320
pmed15	300	100	1729	1729	0,0474	1,3492	1729	0,0879	9,5992	1729	0,0682	22,9796
pmed16	400	5	8162	8162	0,0000	2,0698	8162	0,0000	1,9354	8162	0,0000	4,8736
pmed17	400	10	6999	6999	0,0000	1,2032	6999	0,0000	2,004	6999	0,0000	7,3228
pmed18	400	40	4809	4809	0,0191	1,1448	4809	0,0000	5,521	4809	0,0000	23,8628
pmed19	400	80	2845	2845	0,0302	1,7824	2845	0,0295	14,8814	2845	0,0415	35,0072
pmed20	400	133	1789	1789	0,0391	2,8406	1789	0,0045	21,8842	1789	0,0056	67,1630
pmed21	500	5	9138	9138	0,0000	2,6832	9138	0,0000	2,6448	9138	0,0000	8,9816
pmed22	500	10	8579	8579	0,0000	1,9268	8579	0,0000	3,4808	8579	0,0000	14,8828
pmed23	500	50	4619	4619	0,0017	1,9748	4619	0,0000	10,1424	4619	0,0000	41,3248
pmed24	500	100	2961	2961	0,0412	3,2296	2961	0,0095	29,125	2961	0,0034	94,6284
pmed25	500	167	1828	1828	0,0996	5,3536	1829	0,1893	52,142	1828	0,1630	160,1154
pmed26	600	5	9917	9917	0,0000	4,3008	9917	0,0000	4,186	9917	0,0000	13,5858
pmed27	600	10	8307	8307	0,0000	2,9130	8307	0,0000	5,1206	8307	0,0000	24,8926
pmed28	600	60	4498	4498	0,0200	3,1048	4498	0,0080	24,139	4498	0,0089	140,2756
pmed29	600	120	3033	3033	0,0772	5,2642	3033	0,0495	58,7782	3033	0,0495	255,8950
pmed30	600	200	1989	1989	0,1528	8,9566	1993	0,3680	105,0686	1992	0,3952	324,5274
pmed31	700	5	10086	10086	0,0000	5,6248	10086	0,0000	5,619	10086	0,0000	20,6124
pmed32	700	10	9297	9297	0,0000	3,7912	9297	0,0000	6,9564	9297	0,0000	31,3434
pmed33	700	70	4700	4700	0,0264	4,5028	4700	0,0013	36,9998	4700	0,0026	201,0994
pmed34	700	140	3013	3013	0,0551	7,9398	3013	0,0504	100,6686	3013	0,0544	369,5228
pmed35	800	5	10400	10400	0,0000	7,3232	10400	0,0000	7,5938	10400	0,0000	24,6428
pmed36	800	10	9934	9934	0,0000	5,0008	9934	0,0000	9,0568	9934	0,0000	38,0824
pmed37	800	80	5057	5057	0,0336	6,2688	5057	0,0103	59,3406	5057	0,0103	335,7676
pmed38	900	5	11060	11060	0,0000	10,3790	11060	0,0000	10,8066	11060	0,0000	32,3912
pmed39	900	10	9423	9423	0,0000	5,7826	9423	0,0000	11,006	9423	0,0000	61,4738
pmed40	900	90	5128	5128	0,0558	7,8662	5129	0,0410	84,5004	5128	0,0378	442,2752

Table 2. Resultados obtidos para cada Metaheurística, aplicadas nas instâncias OR-Library. Para cada instância e método, foram realizados 50 execuções. A coluna Melhor lista o melhor resultado encontrado, sendo que os que atingiram o ótimo estão em negrito. A coluna Erro % informa a porcentagem com que a média dos valores obtidos ficou distante do valor ótimo e a coluna Tempo relata o tempo médio gasto pelas Metaheurísticas para cada instância, em segundos.