



SPOLM 2009

ISSN 2175-6295

Rio de Janeiro- Brasil, 05 e 06 de agosto de
2009.

034/2009 - HEURÍSTICA GRASP BI-OBJETIVO PARA O PROBLEMA DAS *P*-MEDIANAS NÃO CAPACITADO

Paula Mariana dos Santos

Departamento de Engenharia Produção,
Universidade Federal de Viçosa
Campus Universitário da UFV, 36570-000, Centro Viçosa, MG, BRASIL
E-mail: paula-marianna@hotmail.com

Michele dos Santos Soares, José Elias Claudio Arroyo

Departamento de Informática,
Universidade Federal de Viçosa
Campus Universitário da UFV, 36570-000, Centro Viçosa, MG, BRASIL
E-mail: myxellys@gmail.com, jarroyo@dpi.ufv.br

Resumo

Neste artigo é proposto uma heurística GRASP para o problema das p -medianas bi-objetivo, no qual são minimizadas simultaneamente duas funções objetivos: soma das distâncias de atendimento e soma dos custos fixos de abertura das facilidades. A heurística GRASP determina um conjunto de soluções dominantes, ou seja, uma aproximação das soluções *Pareto-ótimas* do problema. A fase construtiva usa uma adaptação da heurística construtiva gulosa-aleatória proposta na literatura para o problema da p -medianas mono-objetivo. A fase da busca local explora três caminhos de busca guiados pela minimização das duas funções objetivos e por uma função de utilidade na qual os pesos dos objetivos são definidos aleatoriamente. Para testar o desempenho da heurística proposta é desenvolvido um algoritmo de Programação Matemática, denominado ε -restrito, que determina soluções *Pareto-ótimas* através da resolução do modelo matemático do problema considerando restrições adicionais. São realizados testes computacionais em problemas com até $m = 402$ locais candidatos. Os resultados mostraram que a heurística GRASP apresenta um bom desempenho em termos de qualidade de soluções obtidas e tempo computacional.

Palavras-Chaves: **Otimização Multiobjetivo; p -medianas; Heurísticas; GRASP.**

Abstract

In this article is proposed a GRASP heuristic for the bi-objective p -median problem, in which two functions are simultaneously minimized: the sum of the serving distances and the total fixed costs for opening facilities. A GRASP heuristic determines a set of dominant solutions that is an approximation of the problem *Pareto-optimal* solutions. The construction phase uses a greedy random constructive heuristic proposed in the literature for the mono-objective p -median problem. The local search phase explores three search paths guided by the minimization of the two objective functions and an utility function in which the weights of the objectives are defined randomly. To test the performance of the proposed heuristic was developed a Mathematical Programming Algorithm, called ε -restrict, that finds *Pareto-*

optimal solutions by solving the mathematical model of the problem considering additional restrictions. Computational tests are performed in problems with until $m = 402$ local candidates. The results showed that the GRASP heuristic presents a good performance in terms of solutions quality and computational time.

Keywords: **Multiobjective optimization; p -medians, Heuristics, GRASP.**

INTRODUÇÃO

A logística de distribuição de produtos e/ou serviços tem recebido atenção crescente, ao longo dos anos, como parte do planejamento estratégico de empresas do setor público e privado. Decisões sobre a melhor configuração para instalação de facilidades destinadas ao atendimento da demanda de uma população são tratadas em uma ampla classe de problemas, conhecida como Problemas de Localização [4]. Problemas de localização de facilidades ocorrem em diferentes contextos e possuem numerosas aplicações. Como exemplos de facilidades podem-se citar fábricas, armazéns, depósitos, bibliotecas, corpo de bombeiros, hospitais, estações-base de serviços de telecomunicação sem fios (tais como serviço de telefone móvel, internet sem fio), etc. São problemas de grande importância econômica para planejamento estratégico de setores produtivos, indústrias, prefeituras, comércio, e outros [12]. A otimização destes modelos pode levar a grandes economias em seus investimentos.

De acordo com o objetivo desejado, duas grandes classes de problemas de localização podem ser definidas [13]. Uma classe trata da minimização de distâncias, média ou total, entre os clientes e os centros de atendimento. O modelo clássico utilizado para representação dos problemas desta classe é o do problema das p -medianas, que visa selecionar p vértices (locais) em uma rede para a instalação de facilidades de forma a minimizar a soma das distâncias entre os vértices de demanda e a facilidade mais próxima. Modelos que buscam minimizar a distância total ou média são apropriados para descrever problemas do setor privado, no qual medidas de custo estão diretamente relacionadas às distâncias envolvidas no atendimento das demandas. Também se encontram aplicações deste modelo a problemas do setor público. [8] apresenta uma forma de manipular os coeficientes da função objetivo para adequar vários problemas de localização ao modelo do problema das p -medianas.

A segunda classe de problemas de localização enfoca a distância máxima entre qualquer cliente e a facilidade designada para atendimento. Tais problemas são conhecidos como problemas de cobertura e a distância máxima de atendimento é denominada distância de cobertura ou de serviço. [18] apresenta um modelo de cobertura de conjuntos para determinar o número mínimo de centros necessários ao atendimento de todos os vértices de demanda, para uma dada distância de cobertura. Por sua simplicidade, este modelo não faz distinção da demanda em cada vértice e o número de facilidades necessárias para atendimento de todos os vértices pode ser grande, incorrendo em aumento dos custos fixos de instalação das facilidades. Uma alternativa considera que o número de facilidades a serem instaladas não é suficiente para o atendimento de toda a demanda existente. Neste caso, a restrição de que toda a demanda seja atendida – para uma dada distância de cobertura – é relaxada e procura-se localizar p facilidades de forma que a configuração de cobertura atenda a maior demanda possível [13]. Este problema é conhecido como o problema de localização de máxima cobertura [2]. Modelos de cobertura são freqüentemente utilizados por órgãos públicos para a localização de serviços emergenciais ou não-emergenciais.

O problema das p -medianas é um problema de otimização combinatória e *NP-hard* [7], o que limita o tamanho das instâncias possíveis de serem resolvidas por métodos exatos. Apesar do aumento da velocidade dos processadores, o processo de solução de problemas de

grande porte ainda baseia-se no aprimoramento de métodos heurísticos desenvolvidos para a resolução dos mesmos [17].

Neste artigo é abordado o problema das p -medianas na versão bi-objetivo, que consiste na minimização da soma das distâncias de atendimento e a soma dos custos fixos de abertura das medianas. A literatura sobre problemas de localização de facilidades com múltiplos objetivos ainda é escassa. Dentre os métodos aplicados para resolver estes problemas podem-se citar: *branch-and-bound* [9], programação dinâmica [6], heurísticas [11] e algoritmos genéticos [19].

No presente trabalho é proposto uma adaptação da heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) para o problema das p -medianas bi-objetivo. A heurística GRASP já foi aplicada com muito sucesso para resolver problemas de localização de facilidades mono-objetivo [14]. Na literatura, recentemente estão sendo propostas aplicações da heurística GRASP para resolver problemas de otimização combinatória multiobjetivo [1].

Para avaliar o desempenho da heurística proposta, é desenvolvido um algoritmo de Programação Matemática Bi-objetivo, denominado ε -restrito (ou baseado em restrições), que determina soluções *Pareto-ótimas* minimizando um dos objetivos do problema e limitando o crescimento do outro objetivo. Os algoritmos são testados utilizando um conjunto de 50 instâncias com número de locais potenciais $m = 50, 100, 200, 300$ e 402 .

FORMULAÇÃO DO PROBLEMA DAS p -MEDIANAS BI-OBJETIVO

Considere um conjunto L de m locais potenciais e um conjunto I de n clientes. Sejam, d_{ij} a distância do cliente i para o local j , f_j o custo fixo de instalação de uma facilidade no local j . O problema consiste na localização (seleção) de p locais, dentre as m , para a instalação (ou abertura) de facilidades e a designação de clientes à facilidades abertas (medianas) de modo a minimizar, simultaneamente dois objetivos, Z_1 : soma das distâncias de cada cliente à facilidade mais próxima e Z_2 : soma dos custos fixos de instalação das facilidades. Neste problema, cada cliente deve ser atendido por uma única facilidade aberta. Não são considerados capacidades dos locais potenciais, ou seja, um local potencial pode atender um número infinito de clientes.

A formulação de Programação Inteira do problema das p -medianas bi-objetivo é baseada na formulação apresentada por [10]. As variáveis de decisão do problema são y_j e x_{ij} . Se uma facilidade for aberta no local j ($j \in L$), $y_j = 1$, caso contrário $y_j = 0$. $x_{ij} = 1$ se o cliente i ($i \in I$), é atendido pela facilidade j ($j \in L$), caso contrário $x_{ij} = 0$.

O modelo do problema das p -medianas bi-objetivo é da seguinte maneira:

$$((P-Bi)) \quad \text{Minimizar } Z_1 = \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij} \quad (1)$$

$$\text{Minimizar } Z_2 = \sum_{j \in L} f_j y_j \quad (2)$$

s.a.

$$\sum_{j \in L} x_{ij} = 1, \quad i \in I \quad (3)$$

$$x_{ij} \leq y_j, \quad i \in I, j \in L \quad (4)$$

$$\sum_{j \in L} y_j = p \quad (5)$$

$$x_{ij} \in \{0,1\}, \quad i \in I, j \in L \quad (6)$$

$$y_j \in \{0,1\}, \quad j \in L \quad (7)$$

As funções objetivo a serem minimizadas são definidas em (1) e (2). A restrição (3) juntamente com (6) garantem que cada cliente seja atendido por uma única facilidade, enquanto a restrição (4) evita que algum cliente seja atendido por uma facilidade não-aberta. Pela restrição (5), conclui-se que o número total de

facilidades abertas deve ser igual exatamente p . Finalmente, (6) e (7) definem que as variáveis de decisão são binárias.

Na Figura 1, mostra-se um exemplo do problema das p -medianas no qual, a partir de $m = 6$ locais potenciais devem ser localizados $p = 3$ medianas para atender $n = 10$ clientes. Nesta Figura apresenta-se uma solução na qual é representada por um vetor binário de tamanho m , que correspondem aos valores das variáveis de decisão y_i . Nessa solução os locais 1, 3 e 5 são selecionados para instalar ou abrir facilidades.

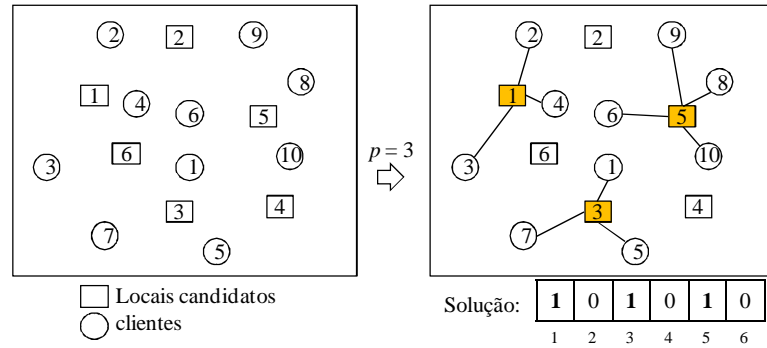


Figura 1. O problema das p -medianas e representação de uma solução.

Uma solução s do problema pertence ao *espaço de decisão factível*, denotado por Ω , se ela satisfaz as restrições (3) à (7). O *espaço objetivo factível* do problema das p -medianas bi-objetivo é definido por $Z = \{q \in \mathbb{R}^2: q = (Z_1(s), Z_2(s)), \forall s \in \Omega\}$, ou seja, para cada solução $s \in \Omega$ existe um ponto q no espaço objetivo Z , tal que $q = (Z_1(s), Z_2(s))$.

As soluções do problema bi-objetivo são caracterizadas pelas seguintes definições:

a) Definição 1 (Soluções ou pontos dominantes):

Uma solução s_1 domina a solução s_2 se as três condições seguintes forem satisfeitas:

i) $Z_1(s_1) \leq Z_1(s_2)$, ii) $Z_2(s_1) \leq Z_2(s_2)$ e iii) $Z_1(s_1) < Z_1(s_2)$ ou $Z_2(s_1) < Z_2(s_2)$.

Similarmente, um ponto $q_1 = (Z_1(s_2), Z_2(s_2))$ domina o ponto $q_2 = (Z_1(s_1), Z_2(s_1))$ se são satisfeitas as mesmas três condições.

b) Definição 2 (Soluções Pareto-ótimas):

Uma solução é *Pareto-ótima* ou *eficiente* se ela não é dominada por nenhuma solução do espaço de soluções factíveis.

O conjunto de soluções *Pareto-ótimas* aqui é denotado por E . No espaço objetivo, os pontos *Pareto-ótimos* determinam a fronteira *Pareto-ótima*. Resolver o problema das p -medianas bi-objetivo (*P-Bi*) consiste em determinar o conjunto E .

METAHEURÍSTICA GRASP PARA O PROBLEMA BI-OBJETIVO DAS P -MEDIANAS

O método GRASP (*Greedy Randomized Adaptive Search Procedure*) proposto por [5] é uma heurística bastante usada para resolver problemas de otimização combinatória mono-objetivo. A heurística GRASP é um procedimento iterativo de múltiplos reinícios, onde a cada iteração consiste de duas fases: uma fase de *construção* para determinar uma solução inicial x e uma fase de *busca local* aplicada para melhorar a solução inicial x e obtendo uma solução ótima local x' . Após executar um número *Max_Iter*, a heurística retorna a melhor solução encontrada.

Neste artigo, é proposta uma adaptação da heurística GRASP para resolver o problema *P-Bi*. A heurística é denominada de *GRASP-Bi*. Na Figura 2 apresentam-se os passos da metaheurística na forma de pseudocódigo. A heurística *GRASP-Bi*, possui duas fases: *Construtiva* e *Busca Local*. São usados dois parâmetros de entrada, o número máximo de iterações (*Max_Iter*) e o número de soluções a serem geradas na fase construtiva (*ns*). O objetivo da heurística é determinar (retornar) um conjunto *D* de *soluções dominantes* próximas das soluções *Pareto-ótimas*. Nas seguintes subseções são detalhadas as fases da heurística *GRASP-Bi*.

```

GRASP-Bi (Max_Iter, ns)
01 D = ∅; //D: armazena soluções dominantes obtidas pela heurística
02 Para iter = 1 até Max_Iter faça
    Fase Construtiva:
03     L = ∅; //L: armazena soluções dominantes obtidas da Fase construtiva.
04     Para i = 1 até ns faça
05         s = ConstruçãoGulosaAleatória();
06         L = soluções dominante de L ∪ {s};
07     Fim-Para
08     D = soluções dominantes de (D ∪ L);
    Fase da Busca Local:
09     Gere aleatoriamente um vetor de pesos λ = (λ1, λ2), tal que λ1+λ2 = 1;
10     s1 = solução de L com menor Z1;
11     s2 = solução de L com menor Z2;
12     s3 = solução de L com menor Zλ = λ1Z1 + λ2Z2;
13     s1 = BuscaLocal(s1, Z1, D);
14     s2 = BuscaLocal(s2, Z2, D);
15     s3 = BuscaLocal(s3, Zλ, D);
16     D = soluções dominantes de D ∪ {s1, s2, s3};
17 Fim-Para
18 Retorne D
Fim GRASP-Bi

```

Figura 2. Estrutura da heurística GRASP-Bi.

FASE CONSTRUTIVA

Na fase construtiva da heurística GRASP é determinado um conjunto *L* de soluções dominantes. Para tal, são construídos *ns* soluções utilizando um método similar ao método *sample greedy* proposto por [15] para o problema das *p*-medianas mono-objetivo. O conjunto *L* é formado pelas soluções dominantes dentre as *ns* soluções construídas (passos 04-07). A construção de uma solução é baseada na minimização da função ponderada $f_{\lambda} = \lambda_1 Z_1 + \lambda_2 Z_2$, onde λ_1 e λ_2 são pesos aleatórios atribuídos aos objetivos Z_1 e Z_2 , respectivamente, tal que $\lambda_1 + \lambda_2 = 1$. Iniciando com uma solução vazia (sem nenhuma mediana), passo a passo escolhe-se um local candidato para ser uma mediana. A cada passo, em vez de selecionar o melhor dentre todos os *m* locais candidatos, escolhe-se aleatoriamente *t* ($t < m$) locais candidatos. Destes *t* locais escolhe-se o melhor local para instalar a próxima mediana. Note que o melhor candidato é determinado pela função ponderada f_{λ} . Segundo [15], o valor de *t* deverá ser suficientemente pequeno para que haja uma redução no tempo computacional. O valor de *t* é definido pela seguinte equação:

$$t = 1,5 \times \log_2 \left(\frac{m}{p} \right) \quad (8)$$

FASE DE BUSCA LOCAL

Na *Fase da Busca Local* (passos 09 a 15) são exploradas três direções de busca, ou seja, a busca local é aplicada a três soluções dominantes escolhidas do conjunto L gerado na fase anterior. As duas primeiras, s_1 e s_2 , são as soluções que possuem os menores valores das funções objetivos Z_1 e Z_2 , respectivamente (passos 10 e 11). A terceira solução, s_3 , é escolhida de $L - \{s_1, s_2\}$ utilizando uma função de utilidade Z_λ (passo 12) definida de seguinte maneira:

$$Z_\lambda(s) = \max \left\{ \lambda_i \left(\frac{Z_i(s) - Z_i^{\min}}{Z_i^{\max} - Z_i^{\min}} \right); i = 1, 2 \right\} \quad (9)$$

onde, λ_1 e λ_2 são os pesos atribuídos a cada função objetivo e são gerados aleatoriamente tal que $\lambda_1 + \lambda_2 = 1$; Z_i^{\max} e Z_i^{\min} são respectivamente os valores máximo e mínimo de cada objetivo i sobre o conjunto de soluções dominantes obtidas até o momento; $Z_i(s)$ é o valor da função objetivo i para uma solução s . A função Z_λ é a função de utilidade de *Tchebycheff* que é normalizada pela faixa de variação de cada função objetivo. Cada função Z_λ possui um ótimo global no conjunto de soluções *Pareto-ótimas* e cada solução *Pareto-ótima* possui uma função Z_λ que é ótima [16].

As soluções, s_1 , s_2 e s_3 , escolhidas da lista L , são melhoradas fazendo uma busca em vizinhança até encontrar ótimos locais. s_1 e s_2 são melhoradas tentando minimizar as funções Z_1 e Z_2 , respectivamente (passos 13 e 14); s_3 é melhorada minimizando a função de utilidade Z_λ (passo 15). Note que, as três direções de busca a serem exploradas são guiadas pelas funções Z_1 , Z_2 e Z_λ , respectivamente (Veja Figura 3). Vale ressaltar que a cada iteração da metaheurística gera-se aleatoriamente um vetor de pesos $\lambda = (\lambda_1, \lambda_2)$ (passo 09), com a finalidade de explorar diferentes direções de busca.

A busca local utilizada consiste em gerar soluções vizinhas a partir das soluções s_1 , s_2 e s_3 . As soluções vizinhas são obtidas fazendo *trocadas* de facilidades na solução atual, isto é, uma facilidade aberta (ou fechada) é trocada com uma facilidade fechada (aberta). Por exemplo, se a solução atual é $s = (0 \ 1 \ 0 \ 0 \ 1 \ 1)$, os vizinhos gerados fazendo *trocadas* são: $(1 \ 0 \ 0 \ 0 \ 1 \ 1)$, $(1 \ 1 \ 0 \ 0 \ 0 \ 1)$, $(1 \ 1 \ 0 \ 0 \ 1 \ 0)$, $(0 \ 0 \ 1 \ 0 \ 1 \ 1)$, $(0 \ 0 \ 0 \ 1 \ 1 \ 1)$, $(0 \ 1 \ 1 \ 0 \ 0 \ 1)$, $(0 \ 1 \ 1 \ 0 \ 1 \ 0)$, $(0 \ 1 \ 0 \ 1 \ 0 \ 1)$, $(0 \ 1 \ 0 \ 1 \ 1 \ 0)$.

Como a busca local é um processo iterativo, deve-se selecionar a melhor solução vizinha, dentre todos os vizinhos gerados. No exemplo acima, a partir de s são gerados 9 vizinhos. Destes 9, deve-se escolher o melhor vizinho. Este melhor vizinho é determinado pelas funções que estão sendo minimizadas: Z_1 , Z_2 e Z_λ , respectivamente para s_1 , s_2 e s_3 .

A busca local finaliza quando não seja possível melhorar a solução atual (melhorar o valor da respectiva função). Vale ressaltar que é mantido um conjunto D que armazena as soluções dominantes dentre todas as soluções encontradas no processo da busca local. Para cada solução vizinha s encontrada, sempre é feito $D = \text{dominates de } (D \cup \{s\})$.

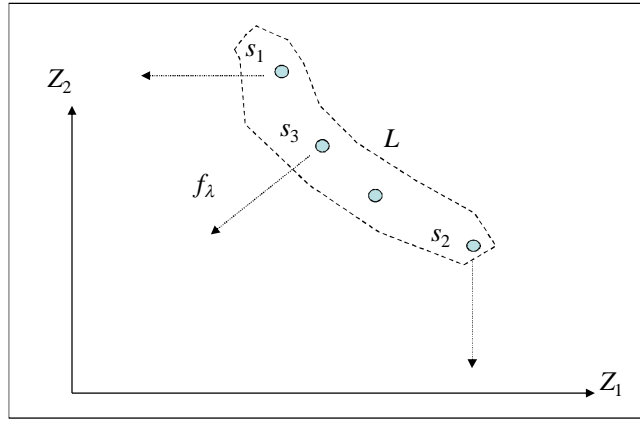


Figura 3: Direções exploradas na Busca Local.

ALGORITMO ε -RESTRITO PARA O PROBLEMA DAS P -MEDIANAS BI-OBJETIVO

Para testar o desempenho da heurística *GRASP-Bi*, é desenvolvido um algoritmo baseado em restrições, denominado ε -restrito [19] cujo objetivo é determinar um conjunto de soluções *Pareto-ótimas* do problema das p -medianas bi-objetivo. Ressalta-se que este algoritmo não determina todas as soluções *Pareto-ótimas*. Na Figura 4 é apresentado o pseudocódigo do algoritmo. Este algoritmo é baseado na resolução do modelo de Programação Inteira do problema e determina soluções *Pareto-ótimas* minimizando iterativamente a função objetivo Z_1 de tal maneira que a função objetivo Z_2 seja limitada por um valor determinado ε (ou seja, $Z_2 \leq \varepsilon$ é uma restrição adicional ao problema).

Algoritmo ε -Restrito (s) / s = número de intervalos
 $E = S = \text{vazio}$ // E : conjunto de pontos *Pareto-ótimos*, S : Conjunto de variáveis
 Determinar o ponto extremo q_2 da fronteira *Pareto-ótima*:

$$\underline{Z}_1 = \text{Min} \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij}, \text{ sujeito a: (3), (4), (5), (6), (7);}$$

$$(\overline{Z}_2, (x_{ij}, y_i)) = \text{Min} \sum_{j \in L} f_j y_j \text{ sujeito a: (3), (4), (5), (6), (7) e } \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij} \leq \underline{Z}_1;$$

$$q_2 = (\underline{Z}_1, \overline{Z}_2);$$

$$E = E \cup q_2; S = S \cup \{(x_{ij}, y_j)\}$$

$$\underline{Z}_2 = \text{Min} \sum_{j \in L} f_j y_j, \text{ sujeito a: (3), (4), (5), (6), (7);}$$

$$\Delta A = \frac{\overline{Z}_2 - \underline{Z}_2}{s}; A = \overline{Z}_2 - \Delta A; \text{ Iter} = 0;$$
Enquanto $\varepsilon > \underline{Z}_2$ **faça** // iteração do algoritmo

$$Z^*_1 = \text{Min} \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij} \text{ sujeito a: (3), (4), (5), (6), (7); e } \sum_{j \in L} f_j y_j \leq A$$

$$((x_{ij}, y_i), Z^*_2) = \text{Min} \sum_{j \in L} f_j y_j \text{ sujeito a: (3), (4), (5), (6), (7); e } \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij} \leq Z^*_1$$

$$q = (Z^*_1, Z^*_2) \text{ // ponto Pareto-ótimo encontrado;}$$
Se $Z_2 < \overline{Z}_2$ **então**

$$\overline{Z}_2 = Z_2 \text{ // procurando uma solução melhor}$$
Fim-Se

$$E = E \cup \{q\}; \text{ // } q \text{ é inserido no conjunto.}$$

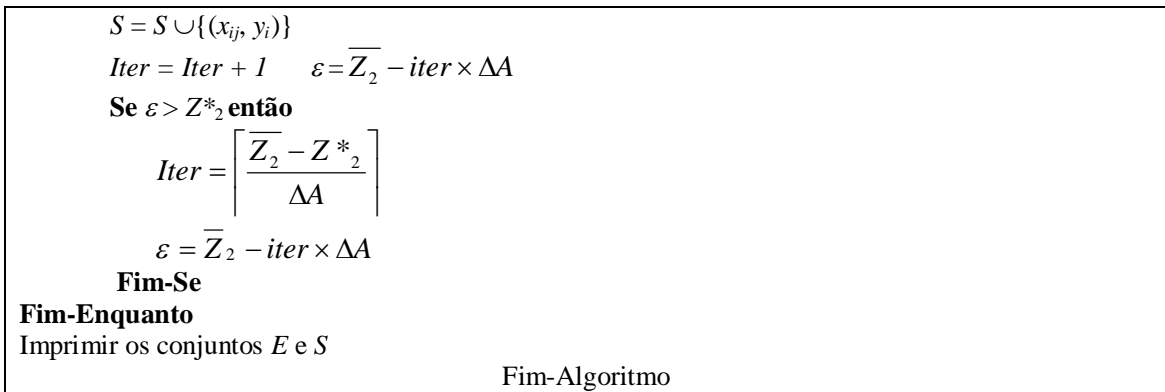


Figura 4: Algoritmo bi-objetivo ε -Restrito.

O algoritmo ε -restrito inicialmente encontra o primeiro ponto extremo $q_2 = (\underline{Z}_1, \overline{Z}_2)$ da fronteira *Pareto-ótima*, formado pelo valor máximo de Z_2 (\overline{Z}_2) e valor mínimo de Z_1 (\underline{Z}_1). \underline{Z}_1 é encontrado ao resolver o modelo (*P-Bi*) considerando apenas a função Z_1 , ou seja, $\underline{Z}_1 = \min \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij}$, sujeito a: (3), (4), (5), (6) e (7). \overline{Z}_2 é encontrado minimizando $\sum_{j \in L} f_j y_j$ sujeito a: (3), (4), (5), (6), (7) e $\sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij} \leq \underline{Z}_1$. Em seguida é determinado o valor mínimo de Z_2 : $\underline{Z}_2 = \text{Min} \sum_{j \in L} f_j y_j$ sujeito a: (3), (4), (5), (6) e (7). A faixa compreendida entre \underline{Z}_2 e \overline{Z}_2 é dividida em s intervalos tamanhos ΔA . A ideia do algoritmo é determinar um ponto *Pareto-ótimo* para cada intervalo. Ou seja, iterativamente é determinado um ponto q minimizando Z_1 sujeito a: (3), (4), (5), (6), (7) e $Z_2 \leq \varepsilon$, onde inicialmente $\varepsilon = \overline{Z}_2 - \Delta A$. A cada iteração, o valor ε é diminuído em ΔA . O algoritmo finaliza quando $\varepsilon = \underline{Z}_2$, ou seja, quando o outro ponto extremo q_1 for encontrado (veja Figura 5).

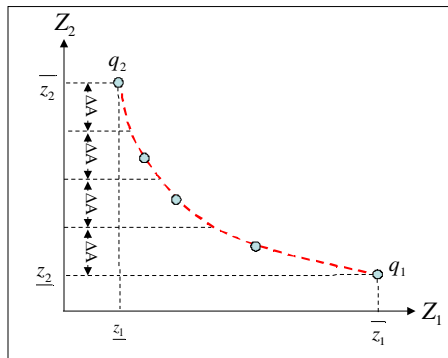


Figura 5: Comportamento dos algoritmos baseado em restrições.

O algoritmo ε -restrito determina no máximo $s+1$ pontos *Pareto-ótimos*, onde s é a quantidade de intervalos considerados. A cada iteração sempre é obtido um ponto *Pareto-ótimo*, mas este ponto pode pertencer a outro intervalo ainda não analisado, ou seja, o valor de Z_2 para o ponto encontrado pode ser menor que $\varepsilon - \Delta A$ (pois, $Z_2 \leq \varepsilon$). O primeiro ponto extremo q_2 pertence ao primeiro intervalo, o último intervalo pode conter dois pontos, o ponto extremo q_1 e mais outro ponto determinado na penúltima iteração.

TESTES COMPUTACIONAIS DOS ALGORITMOS

A heurística *GRASP-Bi* foi programada na linguagem C++ e o algoritmo ε -restrito foi codificado em *Mosel*. Os modelos de Programação Inteira foram resolvidos utilizando o Software de Programação Matemática *Xpress-MP* da *Dash Optimization*. Ambos os algoritmos foram executados em um computador com processador Intel Core 2 Quad de 2,4 Ghz com 2GB de memória.

Os parâmetros da heurística *GRASP-Bi*, *Max_Iter* (numero máximo de iterações GRASP) e *ns* (número de soluções a serem geradas na fase de construção) foram determinados experimentalmente. Os melhores resultados da heurística foram obtidos utilizando *Max_Iter* = 800 e *ns* = 20. No algoritmo ε -restrito foi utilizado *s* = 50 (número de intervalos ou iterações). Ou seja, para cada problema, o algoritmo ε -restrito encontra no máximo 51 soluções Pareto-ótimas.

Para avaliar o desempenho dos algoritmos foram gerados 50 problemas testes baseados em dados disponíveis na literatura para o problema das *p*-medianas mono-objetivo. Neste trabalho são considerados problemas no qual o número de locais candidatos para a instalação de facilidades (*m*) é igual ao número de clientes (*n*). Os tamanhos dos problemas são *n* = *m* = 50, 100, 200, 300 e 402 com número de medianas *p* = 5 e 10. Para cada combinação de *p* e *n* foram gerados 5 problemas. As distâncias *d_{ij}* (dos clientes para as facilidades) e os custos fixos de instalação das facilidades *f_j* são gerados uniformemente nos intervalos [0, 26800] e [1000, 40000], respectivamente.

A seguir são comparadas as soluções obtidas pela heurística *GRASP-Bi* com as soluções Pareto-ótimas encontradas pelo algoritmo ε -restrito. Sejam *D* e *E'* os conjuntos de pontos (soluções) determinados pelos métodos *GRASP-Bi* e ε -restrito, respectivamente. Já que *E'* não necessariamente contém todos os pontos Pareto-ótimos, determina-se o conjunto de referência, denotado por *Ref*, formado pelos pontos dominantes de $D \cup E'$. Note que o conjunto *Ref* conterá todos os pontos de *E'*, pois *E'* contém pontos Pareto-ótimos. Todos os pontos de *D* não necessariamente estarão no conjunto *Ref*. Para a heurística *GRASP-Bi* calcula-se o número de pontos que pertencem a *Ref* (ou seja, $|Ref \cap D|$). Quanto maior o número $|Ref \cap D|$, melhor será a qualidade dos pontos obtidos pela heurística.

A Tabela 1 contém os resultados (total de pontos) obtidos pelos métodos ε -restrito e *GRASP-Bi* para cada tamanho de problema com $p \in \{5, 10\}$ e $n \in \{50, 100, 200, 300, 402\}$. Esta Tabela exibe o número total de pontos nos conjuntos *Ref*, *E'*, *D*, $Ref \cap D$ e $E' \cap D$ para um total de 5 problemas para cada valor de *n*. $|E' \cap D|$ é o número total de pontos iguais obtidos pelos dois métodos. Observa-se que para os problemas com $p=5$, $n = 50, 100$, o conjunto *Ref* é formado por todos os pontos obtidos pela heurística *GRASP-Bi*. Para os outros problemas ($p=5$, $n = 200, 300, 402$), $|Ref \cap D| > |E'|$, ou seja, a heurística determina um número maior de pontos dominantes. Para os 25 problemas com $p=5$, foram determinados 863 pontos dominantes (i.e. $|Ref| = 863$) dos quais 556 e 806 pontos foram determinados pelos métodos ε -restrito e *GRASP-Bi*, respectivamente. A heurística determinou 499 pontos Pareto-ótimos encontrados pelo método ε -restrito. Para os 25 problemas com $p=10$, foram encontrados 1501 pontos dominantes ($|Ref| = 1501$) sendo que os métodos ε -restrito e *GRASP-Bi* encontraram respectivamente 823 e 1141 pontos. Para este grupo de problemas, a heurística determinou 463 pontos Pareto-ótimos encontrados pelo método ε -restrito.

<i>p=5</i>					
<i>n</i> = <i>m</i>	$ Ref $	ε -restrito	<i>GRASP-Bi</i>		$ E' \cap D $
		$ E' $	$ D $	$ Ref \cap D $	
50	151	108	151	151	108
100	144	98	144	144	98
200	157	102	157	151	96
300	173	113	178	163	103

402	238	135	240	197	94
Total	863	556	870	806	499
$p=10$					
50	285	160	284	282	157
100	275	157	287	231	113
200	322	181	341	212	71
300	297	164	354	202	69
402	322	161	379	214	53
Total	1501	823	1645	1141	463

Tabela 1. Comparação da heurística *GRASP-Bi* com o algoritmo ε -restrito para problemas com $p = 5$ e 10 medianas (800 iterações).

Os métodos também são comparados utilizando uma medida de distância proposta em [3] que mede a proximidade de um conjunto D de soluções aproximadas com relação ao conjunto de referência Ref . Assume-se que D é uma boa aproximação de Ref se as soluções de D estão bem próximas das soluções de Ref . Em outras palavras, para cada ponto $z \in Ref$ existe $z' \in D$ tal que a distância entre z' e z seja mínima. Considera-se a distância média (D_{med}) definida da seguinte maneira:

$$D_{med} = \frac{1}{|Ref|} \sum_{z \in Ref} \min_{z' \in D} d(z', z) \quad ((10))$$

onde $|R|$ é a cardinalidade do conjunto R e d é definido por:

$$d(z', z) = \max_{j=1, \dots, r} \left\{ \frac{1}{\Delta_j} (z'_j - z_j) \right\}, \quad z' = (z'_1, \dots, z'_r) \in D, \quad z = (z_1, \dots, z_r) \in Ref \quad ((11))$$

onde $\Delta_j = \max f_j - \min f_j$ (diferença entre o maior e menor valor encontrados para f_j).

$n = m$	$p=5$		$p=10$	
	ε -restrito	<i>GRASP-Bi</i>	ε -restrito	<i>GRASP-Bi</i>
50	0,154	0	0,167	0,002
100	0,173	0	0,271	0,267
200	0,275	0,020	0,308	0,767
300	0,241	0,059	0,366	0,771
402	0,288	0,187	0,425	0,749

Tabela 2. Distância média dos algoritmos ε -restrito e *GRASP-Bi*.

Na Tabela 2 são apresentadas as distâncias médias (D_{med}) dos pontos encontrados pelos dois algoritmos em relação aos pontos do conjunto de referência Ref . Para os problemas com $p = 5$, os pontos encontrados pelo algoritmo *GRASP-Bi* possuem menor distância quando comparados aos pontos encontrados pelo algoritmo de ε -restrito. Para os problemas com $p=10$, o algoritmo ε -restrito gerou pontos mais próximos ao conjunto de referência em problema com $n = 200, 300$ e 402. A heurística *GRASP-Bi*, mesmo encontrando maior número de pontos de referência, alguns pontos gerados podem estar distantes da curva de referência (veja Figura 6).

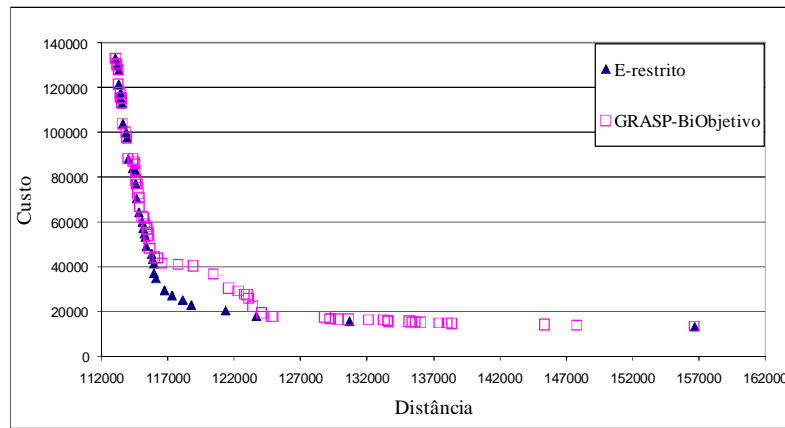


Figura 6: Comportamento dos algoritmos ε -restrito e *GRASP-Bi* para um problema com $m=402$ e 10 medianas.

A Tabela 3 apresenta os tempos computacionais médios, em segundos, gastos pelos métodos ε -restrito e *GRASP-Bi*. Nesta Tabela observa-se que os tempos da heurística são muito menores que os tempos gastos pelo algoritmo ε -restrito. Note que o método ε -restrito, para resolver um problema com $p=10$ e $n=402$, gasta em média 6 horas, já a heurística *GRASP-Bi* gasta apenas 31 minutos em média.

$n = m$	$p=5$		$p=10$	
	ε -restrito	<i>GRASP-Bi</i>	ε -restrito	<i>GRASP-Bi</i>
50	28,4	2,8	70,7	11,2
100	104,9	13,4	155,3	70,4
200	830,0	65,8	1.467,6	374,8
300	5.134,2	161,2	7.941,7	987,4
402	24.909,3	311,8	22.657,0	1.889,0

Tabela 3. Tempo computacional médio (em segundos) dos algoritmos ε -restrito e *GRASP-Bi*.

Os resultados computacionais mostram que o método *GRASP-Bi* apresentou um bom desempenho em termos de qualidade de soluções e tempo de execução quando comparado com o método iterativo ε -restrito.

CONCLUSÕES

Neste artigo foram propostos dois métodos para resolver o problema das p -medianas no qual são otimizados dois objetivos, soma das distâncias de atendimento e soma dos custos de abertura das p facilidades. O primeiro método (*GRASP-Bi*) é uma heurística baseada na técnica de busca GRASP que consiste em, repetidamente, construir soluções para serem melhoradas através de uma busca local. O segundo método (ε -restrito) é baseado na resolução do modelo de Programação Inteira do problema e determina soluções *Pareto-ótimas* minimizando um dos objetivos e limitando o crescimento do outro objetivo. A heurística *GRASP-Bi*, quando comparada com o método ε -restrito, se mostrou bastante eficiente na prática, possibilitando encontrar um número grande de soluções *Pareto-ótimas*. Dos 50 problemas resolvidos, a heurística *GRASP-Bi* e o método ε -restrito determinaram, respectivamente, 1947 e 1379 soluções dominantes, dos quais 962 soluções *Pareto-ótimas* foram iguais. Os tempos de processamento da heurística *GRASP-Bi* foram, em média, bem menores que os tempos do método ε -restrito. O método ε -restrito, mesmo não encontrando todas as soluções *Pareto-ótimas*, é capaz de encontrar um número grande de soluções

para problemas de grande porte. Este método é importante para avaliar o desempenho de técnicas heurísticas bi-objetivo.

Agradecimentos

À empresa *Dash Optimization* pela licença cedida da versão completa do *Xpress-MP*, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento parcial do trabalho e à CAIXA pela bolsa concedida.

Referências Bibliográficas

- [1] Arroyo, J. E. C.; Vieira, P. S.; Vianna, D. S.: A GRASP algorithm for the Multi-Criteria Minimum Spanning Tree Problem. *Annals of Operations Research*, v. 159, n. 1, pp.125-133, (2008).
- [2] Church, R.; Reville, C.: The Maximal Covering Location Problem. *Papers of the Regional Science Association*, v. 32, n. 1, pp. 101--118, (1974).
- [3] Czyzak, P.; Jaskiewicz, A.: Pareto simulated annealing - A metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multicriteria Decision Analysis*. v. 7, n. 1, p. 34-47, 1998.
- [4] Drezner, Z. (Editor): *Facility Location: a Survey of Splications and Methods*. New York: Springer-Verlag, pp. 571. (1995).
- [5] Feo, T. A.; Resende, M. G. C.: Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*. v. 6, pp. 109--133, (1995).
- [6] Fernandez, E.; Puerto, J.: Multiobjective Solution of Uncapacitated Plant Location Problem. *European Journal of Operational Research*, v. 145, pp. 509--529, (2003).
- [7] Garey M. R; Johnson D. S.: *Computers and Intractability, a Guide to the Theory of NP-completeness*. 1. ed. Freeman, New York, (1979).
- [8] Hillsman, E.L.: The p-median Structure as a Unified Linear Model for Location-Allocation Analysis. *Environmental and Planning A*, v. 16, pp. 305--318, (1984).
- [9] Mavrotas, G.; Diakoulaki, D.: A Branch and Bound Algorithm for Mixed Zero-One Multiple Objective Linear Programming. *European Journal of Operational Research*, v. 107, pp. 530--541, (1998).
- [10] Mladenovic, N.; Brimberg, J.; Hansen, P.; Moreno-Pérez J. A.: The p-median Problem: A Survey of Metaheuristic Approaches. *European Journal of Operational Research*, v. 179, n. 3, pp. 927--939, (2007).
- [11] Nozick, L. K.: The Fixed Charge Facility Location Problem with Coverage Restrictions. *Transportation Research Part E*, v. 37, pp. 281--296, (2001).
- [12] Owen, S. H.; Daskin, M. S.: Strategic Facility Location: A Review. *European Journal of Operational Research*, v. 111, pp. 423--447, (1998).
- [13] Pereira, M. A.: Um método Branch-and-Price para problemas de localização de p-medianas: Tese de Doutorado em Computação Aplicada, INPE, São José dos Campos, pp. 23--25, (2005).
- [14] Resende, M. G. C.; Werneck, R. F.: A hybrid heuristic for the p-median problem. *Journal of Heuristics*, v. 10, n 1, p. 59--88, (2004).
- [15] Resende, M. G. C.; Werneck, R. F. A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, v. 174, n. 1, p. 54-68, (2005).
- [16] Steuer, R. E. *Multiple criteria optimization-theory, computation and application*. 1. ed. Wiley, (1986).
- [17] Teitz, M. B.; Bart, P.: Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph. *Operations Research*, v. 16, pp. 955--961, (1968).
- [18] Toregas, C.; Swain, R.; Reville, C.; Bergaman, L.: The Location of Emergency Service Facilities. *Operations Research*, v. 19, pp. 1363--1373, (1971).

- [19] Villegas, J. G.; Palacios, F.; Medaglia, A. L.: Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. *Annals of Operations Research*, v.147, n. 1, pp. 109--141, (2006).