Um Algoritmo GRASP para o problema das *p*-medianas bi-objetivo

Michele dos Santos Soares[1], José Elias C. Arroyo[2], Paula Mariana dos Santos[3]

[1,2]Departamento de Informática
[3]Departamento de Engenharia Produção,
Universidade Federal de Viçosa
Campus Universitário da UFV, 36570-000, Centro Viçosa, MG, BRASIL
E-mails:, [1]myxellys@gmail.com, [2]jarroyo@dpi.ufv.br, [3]paula-marianna@hotmail.com,

Abstract. In this article is presented the bi-objective *p*-median problem that consists in finding *p*-locals from a set of *m* potential locals to install facilities in which two objective functions are simultaneously minimized: the sum of the distances from each customer to its nearest facility and the sum of the fixed costs incurred by the open facilities. To determine a set of dominant solutions, that is, to find an approximation of the *Pareto-Optimal* solutions, we propose a method based on GRASP (Greedy Randomized Adaptive Search Procedure) heuristic that constructs dominant solutions iteratively (constructive phase) and then, it executes a improvement procedure (local search). To test the performance of the proposed heuristic, we develop a Mathematical Programming Algorithm, called ε-restrict, that find *Pareto-optimal* solutions solving the mathematical model of the problem considering additional restrictions. Computational tests are realized on instances with up to m = 402 potential locals. The results show that the GRASP heuristic is very efficient and promising.

Keyword: Multiobjective optimization, *p*-median problem, heuristics, GRASP.

1 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 [5]. 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 [14]. A otimização destes modelos pode ocasionar grandes economias em seus investimentos.

De acordo com o objetivo desejado, duas grandes classes de problemas de localização podem ser definidas. Uma classe trata da minimização de distâncias, média ou total, entre os clientes e os centros de atendimento (facilidades). 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 à problemas do setor público. Em [9] é apresentada 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. Em [17] é apresentado 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. 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* [8], 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 [13];[15].

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* (Mavrotas e Diakoulaki, 1998), programação dinâmica [7] e algoritmos genéticos [3];[4];[12];[18].

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 [15];[16]. Na literatura, recentemente estão sendo propostas aplicações da heurística GRASP para resolver problemas de otimização combinatória multiobjetivo [1];[10].

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.

A formulação do problema das p-medianas encontra-se na Seção 2 deste trabalho. A heurística GRASP, juntamente com as explicações das fases construtivas e de busca local estão na Seção 3. A Seção 4 apresenta o algoritmo ε-restrito. Os resultados computacionais e as conclusões do trabalho são discutidos nas Seções 5 e 6, respectivamente.

2 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 os m, para a instalação (ou abertura) de facilidades e a designação de clientes às 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.

A formulação de Programação Inteira do problema das p-medianas bi-objetivo é baseada na formulação apresentada em [13]. As variáveis de decisão do problema são y_i 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*)

$$Minimizar\ Z_1 = \sum_{i\in I}\sum_{j\in L} d_{ij}x_{ij} \tag{1}$$

$$Minimizar\ Z_2 = \sum_{j\in L} f_j y_j \tag{2}$$

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

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

$$\sum_{j\in L} y_j = p \tag{5}$$

$$x_{ij} \in \{0,1\}, \quad i\in, j\in L \tag{6}$$

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

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.

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 $Z = \{q \in R^2: q = (Z_1(s), Z_2(s)), \forall s \in \Omega\}$, 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 de um 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 $q_2 = (Z_1(s_2), Z_2(s_2))$ 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^{ot}. No espaço objetivo, os pontos *Pareto-ótimos* definem a fronteira *Pareto-ótima*. Resolver o problema das p-medianas bi-objetivo (*P-Bi*) consiste em determinar o conjunto E^{ot}.

3 Heurística GRASP para o problema bi-objetivo das *p*-medianas

```
GRASP-Bi (Max_Iter, ns)
D = ∅; //D: armazena soluções dominantes obtidas pela heurística
Para iter = 1 até Max_Iter faça
        Fase Construtiva:
    L = ∅; //L: armazena soluções dominantes obtidas da Fase construtiva.
    Para i = 1 até ns faça
            s = ConstruçãoAleatória( );
            L = soluções dominantes de L ∪ {s};
    Fim-Para
    D = soluções dominantes de (D ∪ L);
        Fase da Busca Local:
    Gere aleatoriamente um vetor de pesos λ = (λ1, λ2), tal que λ1+λ2 = 1;
    s1 = solução de L com menor Z1; s2 = solução de L com menor Z2;
    s3 = solução de L com menor Zλ = λ1Z1 + λ2Z2;
    s1 = BuscaLocal(s1, Z1, D); s2 = BuscaLocal(s2, Z2, D);
    s3 = BuscaLocal(s3, Zλ, D);
    D = soluções dominantes de D ∪{s1, s2, s3};
Fim-Para
Retorne D
```

Fig. 1. Estrutura da heurística GRASP-Bi.

O método GRASP (*Greedy Randomized Adaptive Search Procedure*) proposto por Feo e Resende [6] é uma heurística bastante usada para resolver problemas de otimização combinatória mono-objetivo [15]. A heurística GRASP é um procedimento iterativo de múltiplos reinícios, onde 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 obter uma solução ótima local x'. Após executar um número máximo de iterações, 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 1 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*.

3.1 Fases Construtiva e Busca Local

A cada iteração *iter* da heurística, é construído um conjunto L de soluções dominantes (*Fase Construtiva*). Este conjunto é formado por soluções dominantes dentre *ns* soluções geradas de forma aleatória (passos 04-07). Note que, a estratégia gulosa consiste em selecionar as soluções dominantes ("as melhores" dentre as *ns* soluções geradas) e a cada iteração será construído um conjunto L diferente.

Na *Fase da Busca Local* (passos 09 a 14) 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 (passo 10). A terceira solução, s_3, é escolhida de $L - \{s_1, s_2\}$ utilizando a função de utilidade linear $Z_\lambda = \lambda_1 Z_1 + \lambda_2 Z_2$ (combinação linear dos objetivos), onde, λ_1 e λ_2 são pesos atribuídos a cada função objetivo e são gerados aleatoriamente tal que $\lambda_1+\lambda_2 = 1$.

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 (passo 12); s_3 é melhorada minimizando a função de utilidade Z_λ (passo 13). Note que, as três direções de busca a serem exploradas são guiadas pelas funções Z_1, Z_2 e Z_λ, respectivamente.

Vale ressaltar que a cada iteração da heurística, 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 *trocas* de facilidades abertas com facilidades fechadas na solução atual. Por exemplo, para $p = 3$, se a solução atual é $s = (1\ 0\ 1\ 0\ 1\ 0)$, são gerados 9 vizinhos fazendo *trocas*: (**0** **1** 1 0 1 0), (**0** 0 1 **1** 1 0), (**0** 0 1 0 1 **1**), (1 **1** **0** 0 1 0), (1 0 **0** **1** 1 0), (1 0 **0** 0 1 **1**), (1 **1** 1 0 **0** 0), (1 0 1 **1** **0** 0), (1 0 1 0 **0** **1**). A cada iteração da busca local, sempre é selecionada 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 para continuar a busca. Este melhor vizinho é determinado pela função que está sendo minimizada: Z_1, Z_2 ou Z_λ. A busca local finaliza quando não for possível melhorar a solução atual.

Vale destacar que, no processo da busca sempre é atualizado o conjunto D que armazena as soluções dominantes dentre todas as soluções encontradas. Para cada solução vizinha s encontrada, sempre é feito D = *Soluções dominantes de* $(D \cup \{s\})$.

4 Algoritmo ε-restrito para o problema das p-medianas bi-objetivo

Algoritmo ε-Restrito (s **)** / s = número de intervalos

$E = S = vazio$; //E: conjunto de pontos *Pareto-ótimos*, S: Conjunto de variáveis

$\underline{Z_2} = Min \sum_{j \in L} f_j y_j$, sujeito a: (3), (4), (5), (6), (7);

$((x_{ij}, y_j), \overline{Z_1}) = Min \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij}$ sujeito a: (3), (4), (5), (6), (7) e $\sum_{j \in L} f_j y_j \leq \underline{Z_2}$;

$q_1 = (\overline{Z_1}, \underline{Z_2})$; $E = E \cup q_1$; $S = S \cup \{(x_{ij}, y_j)\}$;

$\underline{Z_1} = Min \sum_{i \in I} \sum_{j \in L} d_{ij} x_{ij}$ sujeito a: (3), (4), (5), (6) e (7).

$\Delta A = \dfrac{\overline{Z_1} - \underline{Z_1}}{s}$; $\varepsilon = \overline{Z_1} - \Delta A$; $Iter = 0$;

Enquanto $\varepsilon > \underline{Z_1}$ **faça** *//iteração do algoritmo*

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

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

$\quad q = (Z^*_1, Z^*_2)$ *//ponto Pareto-ótimo encontrado;*

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

$\quad Iter = Iter + 1$; $\varepsilon = \overline{Z_1} - iter \times \Delta A$;

$\quad$ **Se** $\varepsilon > Z^*_1$ **então**

$\quad\quad Iter = \left\lceil \dfrac{\overline{Z_1} - Z^*_1}{\Delta A} \right\rceil$; $\varepsilon = \overline{Z_1} - iter \times \Delta A$;

$\quad$ **Fim-Se**

Fim-Enquanto

Imprimir os conjuntos E e S;

Fig. 2. Algoritmo bi-objetivo ε-restrito.

Para testar o desempenho da heurística *GRASP-Bi*, é desenvolvido um algoritmo baseado em restrições, denominado ε-restrito cujo objetivo é determinar um conjunto de soluções *Pareto-ótimas* do problema das p-medianas bi-objetivo. Em [18], o algoritmo ε-restrito é aplicado para resolver outro tipo problema de localização de facilidades bi-objetivo. Ressalta-se que este algoritmo não determina todas as soluções *Pareto-ótimas*. Na Figura 2 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_2 de tal maneira que a função objetivo Z_1 seja limitada por um valor determinado ε (ou seja, $Z_1 \leq \varepsilon$ é uma restrição adicional ao problema).

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

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_1 para o ponto encontrado pode ser menor que $\varepsilon - \Delta A$ (pois, $Z_1 \leq \varepsilon$). O primeiro ponto extremo q_1 pertence ao primeiro intervalo, o último intervalo pode conter dois pontos, o ponto extremo q_2 e mais outro ponto determinado na penúltima iteração.

5 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* = 600 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 estarão necessariamente 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.

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

p=5					
		ε-restrito	*GRASP-Bi*		
$n = m$	$\|Ref\|$	$\|E'\|$	$\|D\|$	$\|Ref \cap D\|$	$\|E' \cap D\|$
50	151	92	151	**151**	92
100	142	83	142	**142**	83
200	153	86	157	**134**	67
300	158	85	181	**124**	51
402	207	114	231	**143**	50
Total	811	460	862	**694**	343
p=10					
50	242	128	254	**207**	93
100	239	145	237	**180**	86
200	265	160	309	**176**	71
300	253	**156**	309	150	53
402	251	**150**	314	142	41
Total	1250	739	1423	**855**	344

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 e 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 811 pontos dominantes (i.e. $|Ref|$ = 811) dos quais 460 e 694 pontos foram determinados pelos métodos ε-restrito e *GRASP-Bi*, respectivamente. A heurística determinou 343 pontos *Pareto-ótimos* encontrados pelo método ε-restrito. Para os 25 problemas com p=10, foram encontrados 1250 pontos dominantes ($|Ref|$ = 1250) sendo que os métodos ε-restrito e *GRASP-Bi* encontraram respectivamente 739 e 855 pontos. Para este grupo de problemas, a heurística determinou 344 pontos *Pareto-ótimos* encontrados pelo método ε-restrito.

A Tabela 2 apresenta os tempos computacionais médios (sobre 5 problemas da mesma dimensão $n \times p$), 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 8 horas, já a heurística *GRASP-Bi* consome apenas 30 minutos em média. 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.

Tabela 2. Tempo computacional médio (em segundos) dos algoritmos ε-restrito e *GRASP-Bi*.

	p=5		p=10	
$n = m$	ε-restrito	*GRASP-Bi*	ε-restrito	*GRASP-Bi*
50	26,1	**2,6**	57,7	**10,0**
100	58,9	**15,4**	139,5	**69,8**
200	832,1	**74,4**	1.207,3	**339,8**
300	4.951,9	**187,6**	5.372,8	**834,0**
402	16.453,4	**365,4**	32.199,3	**1.832,4**

6 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 grande número de soluções *Pareto-ótimas*. Dos 50 problemas resolvidos, a heurística *GRASP-Bi* e o método ε-restrito determinaram respectivamente, 1549 e 1199 soluções dominantes, dos quais 687 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 grande número 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

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.; Revelle, C.: The Maximal Covering Location Problem. Papers of the Regional Science Association, v. 32, n. 1, pp. 101--118, (1974)
3. Dias, J.; Captivo, M. E.; Clímaco, J.: A memetic algorithm for multi-objective dynamic location problems. Journal of Global Optimization, v. 42, n. 2, pp. 221--253, (2007)
4. Doerner, K. F.; Gutjahr, W. J.; Nolz, P.C.: Multi-criteria location planning for public facilities in tsunami-prone coastal areas. OR Spectrum, paper in press, (2008)
5. Drezner, Z. (Editor): Facility Location: a Survey of Spplications and Methods. New York: Springer-Verlag, pp. 571. (1995)
6. Feo, T. A.; Resende, M. G. C.: Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization. v. 6, pp. 109--133, (1995)
7. Fernandez, E.; Puerto, J.: Multiobjetive Solution of Uncapacitated Plant Location Problem. European Journal of Operational Research, v. 145, pp. 509--529, (2003)
8. Garey M. R; Johnson D. S.: Computers and Intractability, a Guide to the Theory of NP-completeness. 1. ed. Freeman, New York, (1979)
9. 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)
10. Ishida, C. Y.; Carvalho, A. B.; Pozo, A. T. R.; Goldbarg, E. F. G.; Goldbarg, M. C.: Exploring Multi-objective PSO and GRASP-PR for Rule Induction, Eighth European Conference on Evolutionary Computation in Combinatorial Optimisation. Berlin : Springer, v. 1, pp. 73--84, (2008)
11. 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)
12. Medaglia, A. L.; Villegas, J. G.; Rodrígues-Coca, D. M.: Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. Journal of Heuristic, paper in press, (2008)
13. 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)
14. Owen, S. H.; Daskin, M. S.: Strategic Facility Location: A Review. European Journal of Operational Research, v. 111, pp. 423--447, (1998)
15. Resende, M. G. C.; Ribeiro, C. C.: Grasp with Path-relinking: Recent Advances and Applications. In: T. Ibaraki, K. Nonobe, M. Yagiura (Eds.), Metaheuristics: Progress as Real Problem Solvers, Springer, pp. 29--63, (2005)
16. 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)
17. Toregas, C.; Swain, R.; Revelle, C.; Bergaman, L.: The Location of Emergency Service Facilities. Operations Research, v. 19, pp. 1363--1373, (1971).
18. 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).