

UM ALGORITMO GENÉTICO COM PATH RELINKING PARA UM PROBLEMA DE LOCALIZAÇÃO DE FACILIDADES BI-OBJETIVO

Paula Mariana dos Santos¹, José Elias Claudio Arroyo², Michele dos Santos Soares³

¹Departamento de Engenharia de Produção e Mecânica

^{2,3}Departamento de Informática

Universidade Federal de Viçosa

Campus Universitário da UFV, 36570-000, Centro Viçosa, MG, Brasil

E-mails: ¹paula-marianna@hotmail.com, ²jarroyo@dpi.ufv.br, ³myxellys@gmail.com

RESUMO

Neste artigo é abordado o problema das p -medianas bi-objetivo que consiste em localizar p -locais a partir de um conjunto de m locais candidatos para a instalação de facilidades minimizando simultaneamente duas funções objetivos: *i*) soma das distâncias de atendimento; e *ii*) soma dos custos fixos de abertura das facilidades. Para determinar um conjunto de soluções não-dominadas para este problema, desenvolveu-se uma adaptação do algoritmo genético NSGA II proposto por (Deb *et al.*, 2002). Neste algoritmo é utilizada uma estratégia de elitismo que consiste em adicionar na população algumas soluções não-dominadas encontradas até o momento. Também foi implementado um procedimento de intensificação baseado na técnica *Path Relinking* que consiste em combinar uma solução da população atual com uma solução dominante. Neste trabalho é analisada a eficácia da técnica *Path Relinking* na obtenção de soluções não-dominadas de um problema de otimização bi-objetivo. O desempenho do algoritmo genético é testado mediante a comparação com um método de Programação Matemática denominado ε -restrito. Este método determina soluções *Pareto-ótimas* minimizando iterativamente um dos objetivos e limitando o crescimento do outro. São realizados testes computacionais em problemas com até $m = 402$ locais candidatos e $p = 20$ medianas.

PALAVRAS CHAVE: Otimização multiobjetivo, p -medianas, algoritmos genéticos, path relinking.

ABSTRACT

This paper considers the bi-objective p -median problem that consists in finding p -locals from a set of m location points to install facilities in which two objective functions are simultaneously minimized: *i*) the sum of distances of servers and *ii*) the total fixed costs for opening facilities. To determine a set of non-dominated solutions for this problem, we develop an adaptation of the genetic algorithm NSGA II proposed by (Deb *et al.*, 2002). In this algorithm is used an elitism strategy that consists in to add in the population some dominant solutions found at the moment. A intensification procedure based on the *Path Relinking* technique was proposed. This procedure consists in combining a solution from the current population with a dominant solution. In this work is analysed the performance of the *Path Relinking* technique when it finds dominant solutions of a bi-objective problem. The performance of the genetic algorithm is tested using a comparison with a Mathematical Programming Method called ε -constraints. This method finds *Pareto-optimal* solutions by minimizing iteratively one of the objectives and limiting the growth of another. Computational tests were performed using problems with up to $m=402$ local candidates and $p=20$ medians.

KEY WORDS: Combinatorial optimization, p -median problem, genetic algorithms, path relinking.

1. Introdução

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, denominados Problemas de Localização (Drezner, 1995). Esses problemas 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), usinas elétricas, etc. São problemas de grande importância econômica para planejamento estratégico de setores produtivos, indústrias, prefeituras, comércio, e outros (Owen e Daskin, 1998).

O problema das p -medianas é um clássico problema que consiste na localização de exatamente p facilidades (medianas) em um dado espaço, para satisfazer a demanda de n clientes tais que a soma total das distâncias entre cada cliente e sua facilidade mais próxima seja minimizada. Na versão não capacitada do problema, uma facilidade pode atender um número ilimitado de clientes. Este tipo de problema pertence à classe NP-difícil, o que limita a resolução de instâncias de grande porte por métodos exatos (Garey e Johnson, 1979). Na literatura, vários métodos exatos e heurísticos têm sido propostos para resolver o problema das p -medianas mono-objetivo (Reese, 2006) (Mladenovic *et al.*, 2007).

A literatura sobre problemas de localização de facilidades com múltiplos objetivos ainda é escassa. Dentre os métodos aplicados atualmente para resolver estes problemas podem-se citar: *branch-and-bound* (Mavrotas e Diakoulaki, 1998), Programação Dinâmica (Fernandez e Puerto, 2003) e heurísticas (Nozick, 2001). Dentre as metaheurísticas podem-se citar algoritmos genéticos (Villegas *et al.*, 2006, Medaglia *et al.* 2009) e GRASP (Soares *et al.*, 2009).

Pesquisas sobre otimização multiobjetivo tiveram um grande impulso na década de 90. Nas revisões bibliográficas de (Jones *et al.*, 2002) e (Gandibleux e Ehrgott, 2005) observa-se que as metaheurísticas algoritmos genéticos, busca tabu e *simulated annealing* foram os métodos mais usados para lidar com os diversos tipos de problemas de otimização multiobjetivo. Metaheurísticas constituem heurísticas de busca inteligente que possuem uma estrutura com componentes genéricos que podem ser adaptados para resolver diversos tipos de problemas de otimização.

Neste trabalho é abordado o problema das p -medianas na qual são minimização dois objetivos: a soma das distâncias de atendimento facilidade-cliente e da soma dos custos fixos de abertura das facilidades. Para resolver o problema é proposto uma adaptação do algoritmo genético NSGA II (*Non-Dominated Sorting Genetic Algorithm II*) proposto por (Deb *et al.*, 2002). O algoritmo proposto AGMO (Algoritmo Genético Multiobjetivo) utiliza uma estratégia de elitismo, similar ao proposto por (Ishibushi e Murata, 1998), que consiste na inserção de soluções não-dominadas (soluções de elite) na população. Também foi implementada uma fase de intensificação baseada da técnica *Path Relinking*.

Os resultados obtidos pelo algoritmo genético puro (AGMO) e pelo algoritmo genético com a fase intensificação (AGMO+PR) foram comparados com os resultados encontrados por um método de Programação Matemática que determina soluções *Pareto-ótimas* através da resolução do modelo matemático do problema considerando restrições adicionais. Os métodos são testados utilizando um conjunto de 50 instâncias com $m = 50, 100, 200, 300$ e 402 locais potenciais.

2. Formulação do Problema das p -Medianas Bi-objetivo

Considere um conjunto J de m locais potenciais e um conjunto I de n clientes. Sejam, d_{ij} a distância do cliente i para o local j e f_j o custo fixo de instalação de uma facilidade no local j . O problema consiste em localizar (selecionar) p locais, dentre os m , para a instalação (ou abertura) de facilidades e a designação de clientes às facilidades abertas (medianas) de modo que dois objetivos sejam minimizados: 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 consideradas as capacidades dos locais potenciais, ou seja, uma facilidade aberta 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 Revelle e Swain (1970). As variáveis de decisão do problema são y_j e x_{ij} . Se uma facilidade for aberta no local j ($j \in J$), $y_j = 1$, caso contrário $y_j = 0$. $x_{ij} = 1$ se o cliente i ($i \in I$), é atendido pela facilidade j ($j \in J$), caso contrário $x_{ij} = 0$.

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

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

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

s.a:

$$(P-Bi) \quad \sum_{j \in J} x_{ij} = 1, \quad i \in I \quad (3)$$

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

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

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

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, ao passo que 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 exatamente igual a p . Finalmente, (6) define 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 (6).

O *espaço objetivo factível* do problema das p -medianas bi-objetivo é definido por $Z(\Omega) = \{q \in R^2: q = Z(s) = (Z_1(s), Z_2(s)), \forall s \in \Omega\}$, 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 de otimização bi-objetivo são caracterizadas pelas seguintes definições:

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 $q_1 = (Z_1(s_2), Z_2(s_2)) \in Z(\Omega)$ domina o ponto $q_2 = (Z_1(s_1), Z_2(s_1))$ se são satisfeitas as mesmas três condições apresentadas acima.

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 Ω .

No espaço objetivo $Z(\Omega)$, os pontos *Pareto-ótimos* definem a fronteira *Pareto-ótima*. Resolver o problema das p -medianas bi-objetivo (*P-Bi*) consiste em determinar o conjunto de soluções *Pareto-ótimas*.

3. Algoritmo Genético Multiobjetivo (AGMO)

Os objetivos principais do algoritmo genético multiobjetivo (AGMO) consistem em gerar um conjunto D de soluções não-dominadas bem próximas do conjunto *Pareto-ótimo* E^{ot} e obter uma boa distribuição das soluções não-dominadas a serem geradas.

No AGMO são usadas as seguintes estratégias (Arroyo e Armentano, 2005):

- *Elitismo*: um subconjunto de soluções não-dominadas é incluído na população corrente (estas soluções participarão da geração de novas soluções); e
- *Classificação e diversificação da população*: as soluções da população são classificadas de

acordo ao grau de dominância. As soluções com o mesmo grau de dominância (soluções não comparáveis) são diferenciadas através de uma medida de distância (*Crowding*) no espaço objetivo (Deb *et al.*, 2002). As soluções mais dispersas possuirão maiores chances de participar na reprodução de novas soluções.

O pseudocódigo do AGMO pode ser resumido na Figura 1.

Algoritmo AGMO+PR
Entrada: parâmetros N (número de soluções na população), N_{elite} (número de soluções de elite), p_R e p_M (probabilidade de recombinação e mutação).
Saída: D , um conjunto de soluções não-dominadas.

- 1: Gere uma população inicial P_1 de N soluções.
- 2: Inicialize o conjunto de soluções não-dominadas: $D_1 = \emptyset$.
- 3: Inicialize o conjunto de soluções de elite: $E_0 = \emptyset$.
- 4: **Para** $t = 1$ até $Nmax_Iterações$ **faça**:
- 5: Classifique a população $P_t \cup E_{t-1}$ de acordo ao grau de dominação das soluções obtendo K subconjuntos F_1, F_2, \dots, F_K (F_i domina F_{i+1} , $\forall i = 1, \dots, K-1$).
- 6: Atualize o conjunto de soluções não-dominadas:
 $D_t =$ soluções não-dominadas de $(D_{t-1} \cup F_1)$.
- 7: Para cada $x \in F_i$ ($i = 1, 2, \dots, K$) calcule $DistCrowding(x)$.
- 8: Seja $SP_t = \emptyset$. Selecione N soluções de $P_t \cup E_{t-1}$ e armazená-los em SP_t .
- 9: Atualize o conjunto de soluções de elite:
 Se $|D_t| \leq N_{elite}$ **então** faça $E_t = |D_t|$
 Senão, $E_t = \{N_{elite} \text{ soluções de } D_t \text{ escolhidas aleatoriamente}\}$;
- 10: Aplique o operador de *Recombinação*:
 Seja $P_t' = \emptyset$ uma nova população.
 Para $i = 1$ até $N/2$ **faça**
 Selecione duas soluções diferentes $x_1, x_2 \in SP_t$ e faça $SP_t = SP_t - \{x_1, x_2\}$. Recombine x_1 e x_2 e gere duas soluções (filhos) y_1 e y_2 . Adicione a P_t' as novas soluções y_1 e y_2 com probabilidade p_R . Caso contrário adicionar x_1 e x_2 a P_t' .
 Fim-para
- 11: Aplique um operador de *mutação* a cada $x \in P_t'$ com probabilidade p_M .
- 12: Faça $P_t = P_t'$.
- 13: Para cada solução em P_t , calcular os valores dos objetivos.
 Intensificação:
 14: $s_o =$ Escolha uma solução da população P_t . $s_g =$ Escolha uma solução dominante de D .
 15: $L = PathRelinking(s_o, s_g)$; // $L =$ conjunto de soluções não-dominadas
 16: Faça $D_t =$ soluções não-dominadas de $(D_{t-1} \cup L)$.
 17: Faça $D = D_t$.
 Fim-Para

Fim-Algoritmo

Figura 1 - Pseudocódigo do Algoritmo Genético Multiobjetivo proposto

A seguir são detalhados os procedimentos usados no algoritmo AGMO. Estes componentes incluem geração da população inicial, estratégia de elitismo, classificação da população, diversificação e seleção e os operadores de recombinação e mutação.

3.1 Geração da População Inicial

A população inicial de tamanho N é formada por soluções não dominadas obtidas por um algoritmo construtivo e por soluções geradas aleatoriamente.

Utilizando uma adaptação heurística gulosa “*Sample Greedy*” proposta por (Resende e Werneck, 2005) são geradas $ns < N$ soluções. A partir dessas soluções é determinado um conjunto L de soluções não-dominadas. As soluções de L são inseridas na população inicial. A população é completada com soluções geradas aleatoriamente, ou seja, são geradas $N - |L|$ soluções aleatórias. A seguir detalha-se o funcionamento da heurística *Sample Greedy*.

Na heurística *Sample Greedy*, a construção de uma solução é baseada na minimização da função ponderada $Z_\lambda = \lambda_1 Z_1 + \lambda_2 Z_2$, onde λ_1 e λ_2 são pesos atribuídos aos objetivos Z_1 e Z_2 ,

respectivamente, tal que $\lambda_1 + \lambda_2 = 1$. Para a construção de uma solução, os valores de λ_1 e λ_2 são definidos aleatoriamente. Iniciando com uma solução vazia (todas as facilidades fechadas), passo a passo escolhe-se um local candidato para ser uma mediana. A cada etapa, 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 Z_j . Segundo (Resende e Werneck, 2005), 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:

$$t = 1,5 \times \log_2 \left(\frac{m}{p} \right) \tag{7}$$

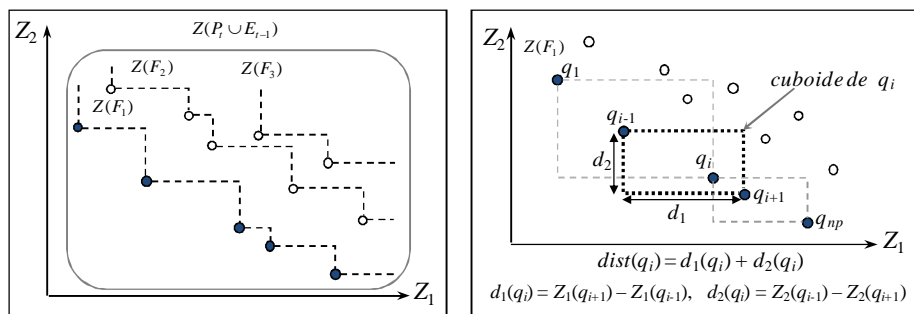
3.2 Elitismo

A estratégia de elitismo usada no algoritmo AGMO é similar à sugerida por (Ishibushi e Murata, 1998). Um conjunto E_t com N_{elite} soluções é selecionado a partir do conjunto de soluções não-dominadas D_t na iteração t . Este conjunto E_t é incluído na população P_{t+1} da próxima iteração $t+1$ formando uma população combinada $P_{t+1} \cup E_t$.

3.3 Classificação da População e Operador de Seleção

Para classificar a população, adota-se a técnica de ordenação por dominância (*Nondominated Sorting Technique*) (Deb et al., 2002) implementada em tempo computacional $O(r \times N^2)$, onde N é o tamanho da população e r é o número de objetivos. Esta técnica classifica a população de acordo com a relação de dominância das soluções, construindo K subconjuntos F_1, F_2, \dots, F_K . Em cada subconjunto $F_i, i = 1, 2, \dots, K$ todas as soluções possuem o mesmo nível de dominância (soluções incomparáveis).

Na Figura 2 (a) mostra-se um exemplo para o caso de um problema de minimização bi-objetivo, onde a população de pontos $Z(P_t \cup E_{t-1})$ é classificada em $K=3$ conjuntos $Z(F_1), Z(F_2)$ e $Z(F_3)$. O conjunto $Z(F_1)$ contém os pontos não-dominados da população.



(a) Classificação da população $P_t \cup E_{t-1}$ (b) Distância de *crowding* para dois objetivos.

Figura 2 – Classificação da população e cálculo da distancia de crowding

Os K subconjuntos gerados F_1, F_2, \dots, F_K , satisfazem a seguintes propriedades:

- As soluções de F_k dominam às soluções de $F_j, \forall k < j$; e
- As soluções no conjunto $F_k, \forall k = 1, \dots, K$, possuem o mesmo nível de dominância.

Portanto, uma solução de F_k possuirá maior “probabilidade” de seleção que uma de $F_j, \forall k < j$. Para diferenciar as soluções pertencentes a um mesmo conjunto F_k , atribui-se a cada solução uma medida denominada distância de *crowding* (*dist*) que é calculada considerando os valores objetivos (Deb et al., 2002). Para cada solução, determina-se uma estimativa do número de soluções localizadas ao redor dela. As soluções mais dispersas de um conjunto F_k possuirão maiores valores *dist* e, portanto maiores “probabilidades” de seleção. Os pontos extremos de um conjunto possuem os maiores valores de *dist*.

A seleção de soluções para reprodução é feita mediante o método de seleção por torneio (*tournament selection*). A Figura 3 ilustra este procedimento:

Tournament selection
 Selecione aleatoriamente duas soluções $x, y \in P_t \cup E_{t-1}$ ($P_t \cup E_{t-1} = F_1 \cup \dots \cup F_K$).
 Caso 1: x e y pertencem a conjuntos diferentes ($x \in F_k$ e $y \in F_j$):
 Se $k < j$ então x é selecionado, caso contrário y é selecionado.
 Caso 2: x e y pertencem a um mesmo conjunto ($x, y \in F_k$):
 Se $dist(Z(x)) > dist(Z(y))$ então x é selecionado, senão y é selecionado.

Figura 3 - Tournament Selection

3.4 Operadores Genéticos

Os operadores incorporados na implementação do AGMO+PR aplicado ao problema da p -medianas foram *recombinação* e *mutação*. Vale lembrar que uma solução do problema é representada por um vetor de p elementos, que representam as facilidades abertas (Alp *et al.* 2003).

Operador de Recombinação (*crossover*)

Sejam duas soluções diferentes s_1 e s_2 (pais). Para gerar duas soluções s_{f1} e s_{f2} (filhos), inicialmente são determinados os conjuntos diferença s_{1-s_2} e s_{2-s_1} . O conjunto s_{1-s_2} (s_{2-s_1}) é formado por facilidades pertencentes à solução s_1 (s_2) que não pertencem a s_2 (s_1). Por exemplo, para as soluções $s_1 = [1, 2, 3, 4, 5]$ e $s_2 = [2, 5, 9, 10, 12]$, $s_{1-s_2} = \{1, 3, 4\}$ e $s_{2-s_1} = \{9, 10, 12\}$. Define-se a diferença simétrica entre as soluções s_1 e s_2 , o número $\Delta = |s_{1-s_2}| = |s_{2-s_1}|$. O filho s_{f1} é gerado fazendo a inserção e remoção de nt elementos em s_1 . Ou seja, nt elementos de s_{2-s_1} são inseridos em s_1 , e nt elementos de s_{1-s_2} são removidos. O filho s_{f2} é gerado de forma similar, fazendo a inserção e remoção de nt elementos em s_2 . O número nt é gerado aleatoriamente tal que, $1 \leq nt \leq \Delta - 1$. Ou seja, nt é o número de trocas aleatórias a serem realizadas entre os elementos de s_{1-s_2} e s_{2-s_1} . Por exemplo, para $nt = 2$, a partir das soluções s_1 e s_2 definidas acima podem ser geradas as soluções $s_{f1} = [9, 2, 12, 4, 5]$ e $s_{f2} = [2, 5, 1, 10, 3]$, fazendo as trocas de 1 por 9 e 3 por 12.

É interessante ressaltar que o procedimento de recombinação garante que não haverá nenhuma facilidade duplicada nas duas soluções-filhos.

Operador de Mutação

O operador de mutação é utilizado para diversificar as soluções da nova população gerada pelo operador de recombinação. No problema abordado, é usada uma mutação que consiste em perturbar as soluções-filhos através da troca aleatória de uma facilidade aberta por uma fechada. Por exemplo, a mutação aplicada à solução $s_{f1} = [9, 2, 12, 4, 5]$ pode gerar a solução $s_{f1} = [9, 2, 8, 4, 5]$, ou seja, a facilidade aberta 12 foi trocada pela facilidade fechada 8.

3.5 Intensificação com Path Relinking

Path Relinking, originalmente proposto por Glover (1996), é uma estratégia de intensificação que consiste em explorar o espaço de busca ou caminho entre duas soluções de elite, denominadas de *origem* e *guia*, respectivamente. Esta técnica transforma gradativamente a solução de origem na solução guia realizando uma sequencia de movimentos que removem atributos da solução e introduzam atributos da solução guia. O melhor "movimento" é escolhido de modo que a distância entre as soluções seja diminuído, ou seja, a solução obtida pelo melhor movimento é escolhido. A cada etapa, são obtidas soluções menos semelhantes do que a solução a origem e mais semelhantes do que a guia.

Neste trabalho, a solução origem s_o é escolhida da população atual de soluções (obtida depois de aplicar os operadores de recombinação e mutação) e a solução guia s_g é uma solução escolhida do atual conjunto D de soluções não-dominadas.

Como no operador de cruzamento, inicialmente são determinados os conjuntos $(s_o - s_g)$ e $(s_g - s_o)$. A partir da solução s_o , a cada passo são geradas um conjunto de soluções vizinhas fazendo a

troca de facilidades $i \in (s_o - s_g)$ com facilidade $j \in (s_g - s_o)$. Ou seja, da solução s_o é removida uma facilidade $i \in (s_o - s_g)$ e inserida uma facilidade $j \in (s_g - s_o)$. Note que, no primeiro passo serão geradas $\Delta = |s_o - s_g| = |s_g - s_o|$ soluções vizinhas. Destas Δ soluções determina-se o conjunto de soluções não-dominadas C . Para continuar com a construção do caminho, no próximo passo uma solução s é escolhida aleatoriamente do conjunto C . Suponha que, no passo anterior, s foi gerada fazendo a troca das facilidades $i \in (s_o - s_g)$ e $j \in (s_g - s_o)$, então os conjuntos $(s_o - s_g)$ e $(s_g - s_o)$ devem ser atualizados fazendo: $(s_o - s_g) = (s_o - s_g) - \{i\}$ e $(s_g - s_o) = (s_g - s_o) - \{j\}$. O procedimento *Path Relinking* finaliza quando $(s_o - s_g) = (s_o - s_g) = \emptyset$ ($\Delta = 0$), ou seja quando $s = s_g$. Este procedimento retorna um conjunto L de soluções não-dominadas encontradas durante a construção do caminho de s_o para s_g . Na Figura 4 é mostrado um exemplo da técnica *Path Relinking* no qual o caminho de s_o para s_g é construída em três passos.

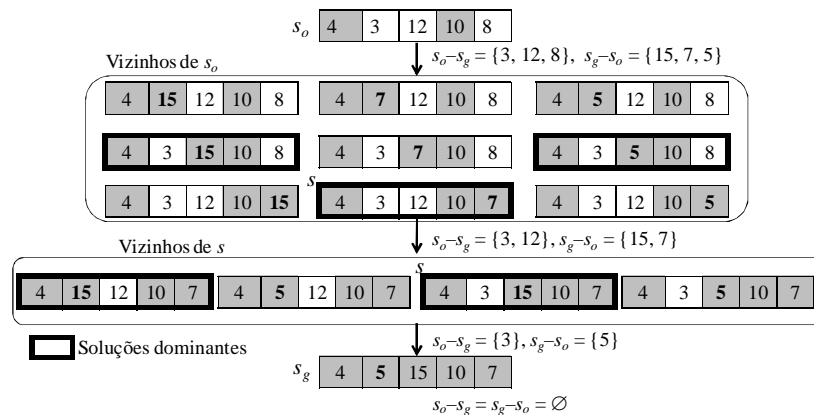


Figura 4 – Exemplo da técnica *Path Relinking* para $p = 5$

4. Algoritmo ϵ -Restrito para o Problema das p -Medianas Bi-objetivo

Para testar o desempenho do AGMO+PR implementado neste trabalho, foi implementado um algoritmo iterativo de Programação Matemática denominado ϵ -Restrito. Este algoritmo determina um subconjunto de soluções *Pareto-ótimas* para o problema das p -medianas bi-objetivo. Na literatura, o algoritmo ϵ -Restrito já foi aplicado eficientemente para resolver outro problema de localização de facilidades bi-objetivo (Villegas *et al.*, 2006).

O algoritmo ϵ -Restrito é constituído de duas fases descritas a seguir (veja Figura 5).

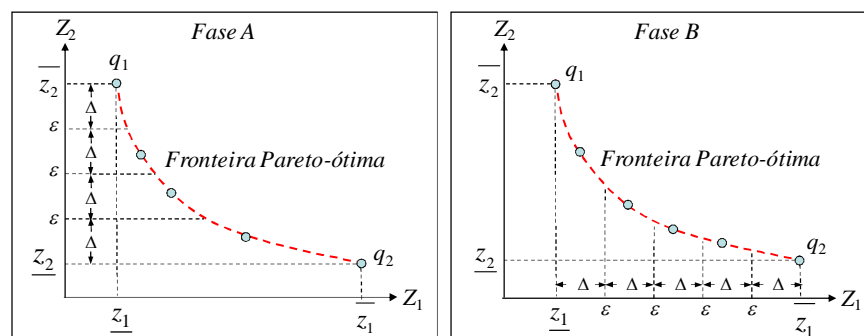


Figura 5 - Fases A e B do algoritmo ϵ -Restrito.

4.1 Algoritmo ε -Restrito: Fase A

Esta fase inicia com o cálculo do ponto extremo $q_1 = (\underline{Z}_1, \overline{Z}_2)$ da fronteira *Pareto-ótima*. Para isto são resolvidos os seguintes modelos mono-objetivo:

$$\underline{Z}_1 = \min Z_1 \text{ s.a: (3), (4), (5) e (6).} \quad (8)$$

$$\overline{Z}_2 = \min Z_2 \text{ s.a: (3), (4), (5), (6) e } Z_1 \leq \underline{Z}_1. \quad (9)$$

Em seguida, determina-se o valor mínimo de Z_2 (\underline{Z}_2), resolvendo o seguinte modelo: $\underline{Z}_2 = \min Z_2$ s.a: (3), (4), (5) e (6).

Para determinar um conjunto de pontos *Pareto-ótimos*, o algoritmo divide a faixa $\overline{Z}_2 - \underline{Z}_2$ em s intervalos de igual tamanho (veja Figura 6). O tamanho de cada intervalo é $\Delta = (\overline{Z}_2 - \underline{Z}_2)/s$. A ideia do algoritmo é determinar um ponto *Pareto-ótimo* $q = (Z^*_1, Z^*_2)$ em cada intervalo ao resolver os seguintes modelos:

$$Z^*_1 = \min Z_1 \text{ s.t.: (3), (4), (5), (6) and } Z_2 \leq \varepsilon \quad (10)$$

$$Z^*_2 = \min Z_2 \text{ s.t.: (3), (4), (5), (6) and } Z_1 \leq Z^*_1. \quad (11)$$

Na resolução do modelo (10) são considerados os seguintes limitantes superiores para a função Z_2 :

$$\varepsilon = \overline{Z}_2 - j \times \Delta, j = 1, \dots, s. \quad (12)$$

A Fase A do algoritmo finaliza quando é determinado o outro ponto extremo da fronteira *Pareto-ótima*, $q_2 = (\overline{Z}_1, \underline{Z}_2)$, isto é, quando $\varepsilon = \underline{Z}_2$.

Note que, a Fase A do algoritmo ε -restrito determina no máximo $s+1$ pontos *Pareto-ótimos*. A cada iteração j 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 $\varepsilon = \overline{Z}_2 - j\Delta$ (pois no modelo (10), $Z_2 \leq \varepsilon$). 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 considerando $\varepsilon = \overline{Z}_2 - (s-1)\Delta$.

4.2 Algoritmo ε -Restrito: Fase B

Esta fase é um algoritmo similar e complementar à Fase A, que inicia com o ponto extremo $q_2 = (\overline{Z}_1, \underline{Z}_2)$ da fronteira *Pareto-ótima* (calculado na Fase A). Novos pontos *Pareto-ótimos* $q = (Z^*_1, Z^*_2)$ são encontrados resolvendo iterativamente os seguintes modelos (13) e (14) (no modelo (13) são considerando $\varepsilon = \overline{Z}_1 - j \times \Delta, j = 1, \dots, s$):

$$Z^*_2 = \min Z_2 \text{ s.a: (3), (4), (5), (6) e } Z_1 \leq \varepsilon \quad (13)$$

$$Z^*_1 = \min Z_1 \text{ s.a: (3), (4), (5), (6) e } Z_2 \leq Z^*_2. \quad (14)$$

5. Resultados Computacionais

Nesta seção analisa-se o comportamento e o desempenho do algoritmo AGMO+PR quando aplicado ao problema das p -medianas bi-objetivo. O AGMO+PR foi programado 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*. Ambos os algoritmos foram executados em um computador com processador Intel Core 2 Quad de 2,4 Ghz com 2GB de memória.

Para avaliar o desempenho dos algoritmos foram gerados 100 problemas teste baseados em dados disponíveis na literatura para o problema das p -medianas mono-objetivo. 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 $p = 5, 10, 15$ e 20 . 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.

Neste trabalho avaliamos o desempenho do AGMO+PR com realação aos algoritmos AGMO (algoritmo genético sem a fase de intensificação *Path Relinking*) e ε -Restrito.

5.1. Parâmetros usados nos Algoritmos AGMO+PR e ε -Restrito

Diferentes valores para os parâmetros do AGMO+PR foram testados e analisados. Na Tabela 1 mostram-se as combinações de parâmetros que geraram os melhores resultados.

Tabela 1 - Parâmetros do AGMO+PR

Tamanho da População	nº máximo de soluções de elite	Probabilidade de <i>crossover</i>	Probabilidade de mutação	Critério de Parada
$N = 200$	$N_{elite} = 20$	$p_R = 0,9$	$p_M = 0,2$	4000 iterações

O único parâmetro utilizado no algoritmo ε -Restrito é o número de intervalos/iteraões (s). Em ambas as fases do algoritmo foi utilizado $s = 50$, ou seja o algoritmo retorna no máximo 102 pontos *Pareto-ótimos*. Vale ressaltar que, quanto maior o valor de s , maior é o número de pontos *Pareto-ótimos* a serem encontrados, entretanto o tempo computacional gasto pelo algoritmo também é maior. O tempo de cada fase do algoritmo ε -Restrito cresce exponencialmente em função do parâmetro s .

5.2. Avaliação das Soluções

Neste trabalho, para avaliar os conjuntos de soluções D, D' e E gerados respectivamente pelos algoritmos AGMO+PR, AGMO e ε -Restrito, primeiramente é determinado o conjunto de soluções de referência R formado por todas as soluções não-dominadas de $D \cup D' \cup E$. Para medir a qualidade dos conjuntos D, D' e E são utilizados duas mediadas:

- *Medida de cardinalidade:*

Esta medida determina o número de soluções de referência geradas por cada algoritmo. Note que todas as soluções obtidas pelo método ε -Restrito estão no conjunto R , pois este método sempre encontra soluções *Pareto-ótimas*. As soluções dos conjuntos D e D' não necessariamente estão no conjunto R . Para os algoritmos AGMO+PR e AGMO calculam-se o número de soluções geradas que pertencem ao conjunto R (ou seja, $|R \cap D|$ e $|R \cap D'|$).

- *Medida de distância:*

Esta medida mede a proximidade de um conjunto X de soluções não-dominadas com relação ao conjunto de referência R (Czyzak e Jaskiewicz, 1998):

$$d_{med}(X) = 100 \times \frac{1}{|R|} \sum_{y \in R} \min_{x \in X} d(x, y) \tag{8}$$

onde d é a distância definida por:

$$d(x, y) = \max_{j=1,2} \left\{ \frac{1}{\Delta_j} (Z_j(x) - Z_j(y)) \right\}, \quad x \in X, \quad y \in R \tag{9}$$

Com $\Delta_j = \max Z_j - \min Z_j$ com $\max Z_j = \text{Max}\{Z_j(x), x \in R\}$ e $\min Z_j = \min\{Z_j(x), x \in R\}$. Note que $d_{med}(X)$ é a média das distâncias de uma solução $y \in R$ à solução mais próxima no conjunto X .

5.3. Análise dos Resultados

A Tabela 2 contém os resultados obtidos pelos métodos ε -Restrito, AGMO e AGMO+PR para cada tamanho de problema com $p \in \{5, 10\}$ e $n \in \{50, 100, 200, 300, 402\}$. Esta tabela exhibe o número total de pontos nos conjuntos $R, E, R \cap D'$ e $R \cap D$ para um total de 5 problemas $n \times p$. Mediante análise desta tabela, nota-se que para os problemas com $p=5, n = 50, 100, 200$ e $402, |R \cap D| > |R \cap D'|$, ou seja, o AGMO+PR determina um número maior de pontos não-dominados. Para os 25 problemas com $p=5$ medianas, por exemplo, foram encontrados 904 pontos de

referência (i.e. $|R| = 904$) dos quais 755 (83,52%), 781 (86,39%) e 799 (88,38%) pontos foram determinados pelos métodos ε -Restrito, AGMO e AGMO+PR, respectivamente. Vale resaltar que o método ε -Restrito não determina todo as soluções Pareto-ótimas. Os métodos heurísticos AGMO e AGMO+PR podem determinar algumas soluções Pareto-ótimas não encontradas pelo algoritmo ε -Restrito.

Na Tabela 2 também são apresentadas as distâncias médias (d_{med}) dos pontos encontrados pelos três algoritmos em relação aos pontos do conjunto de referência R . Assume-se que E , D e D' são boas aproximações de R se as soluções de E , D e D' estão bem próximas das soluções de R . Note que as distancias para o algoritmo ε -Restrito não necessariamente são zero, pois este algoritmo não determina todas as soluções Pareto-ótimas. De um modo geral, para os problemas com $p = 5$, $n = 50, 100, 200, 300$ e 402 , os pontos localizados pelo algoritmo AGMO+PR possuem menor distância quando comparados aos pontos encontrados pelo algoritmo AGMO e ε -Restrito. O AGMO possuiu as menores distâncias apenas para o problema 300×10 , com uma diferença de apenas 0,024 quando comparado ao AGMO+PR.

Tabela 2 - Comparação entre os Algoritmos AGMO e AGMO+PR para $p = 5, 10$

Problema $n \times p$	Referência $ R $	ε -Restrito		AGMO		AGMO+PR	
		$ E $	$d_{med}(E)$	$ R \cap D' $	$d_{med}(D')$	$ R \cap D $	$d_{med}(D)$
50x5	151	136	0,037	148	0,007	148	0,007
100x5	144	123	0,031	136	0,020	137	0,016
200x5	163	138	0,045	151	0,023	155	0,023
300x5	175	153	0,027	139	0,194	139	0,162
402x5	271	205	0,071	207	0,076	220	0,070
Total	904	755	0,211	781	0,320	799	0,278
50x10	286	215	0,063	260	0,177	275	0,048
100x10	287	229	0,054	206	0,597	225	0,525
200x10	385	291	0,060	231	1,399	281	0,482
300x10	455	293	0,071	317	0,594	325	0,618
402x10	461	283	0,072	293	0,842	319	0,602
Total	1874	1311	0,320	1307	3,609	1425	2,275
TOTAL	2778	2066	0,531	2088	3,929	2224	2,553

Tabela 3 – Tempos (em segundos) dos algoritmos ε -restrito, AGMO e AGMO+PR, para $p=5,10$

Problemas	$p=5$			$p=10$		
	ε -restrito	AGMO	AGMO+PR	ε -restrito	AGMO	AGMO+PR
50	54,5	5,6	5,9	128,4	7,0	6,8
100	163,8	9,0	9,7	299,3	12,7	13,0
200	1.662,1	17,2	18,6	2.674,9	25,7	27,2
300	10.086,1	25,3	27,6	13.314,4	38,5	41,2
402	41.362,7	33,4	37,3	54.856,3	51,8	57,2

Na Tabela 3 observa-se que os tempos do algoritmo AGMO+PR apresentam valores levemente maiores que os tempos gastos pelo AGMO e muito menores quando comparados ao tempos obtidos pelo método ε -Restrito. Para resolver um problema de tamanho 300×10 , por exemplo, o método AGMO+PR gasta em média 41,2 segundos, ao passo que o AGMO retorna uma solução aceitável para este mesmo problema com apenas 2,7 segundos de diferença. O algoritmo ε -Restrito demanda aproximadamente 4 horas para resolver o mesmo problema.

A Tabela 4 evidencia o fato de que, para os problemas de tamanho $\{50, 100, 200, 402\} \times \{15, 20\}$ o AGMO+PR determina um número maior de pontos não-dominados ($|R \cap D| > |R \cap D'|$). Para os 25 problemas com $p=20$ medianas, por exemplo, foram encontrados 3071 pontos de referência (i.e. $|R| = 3071$) dos quais 2548 (82,97%) e 2927 (95,31%) pontos foram determinados pelos

métodos AGMO e AGMO+PR, respectivamente. Com relação às distâncias médias (d_{med}), a Tabela 4 demonstra que os pontos localizados pelo algoritmo AGMO+PR possuem menor distância quando comparados aos pontos encontrados pelo algoritmo AGMO. Ainda, de acordo com a Tabela 4, observa-se que os tempos do algoritmo AGMO+PR apresentam valores levemente maiores que os tempos gastos pelo AGMO.

Tabela 4 – Comparação entre os Algoritmos AGMO e AGMO+PR em problemas com $p=15$ e 20

Problema $n \times p$	Referência $ R $	AGMO			AGMO+PR		
		$ R \cap D' $	$d_{med}(D')$	Tempo	$ R \cap D $	$d_{med}(D)$	Tempo
50x15	319	308	0,013	7,7	316	0,003	7,9
100x15	399	351	0,142	15,2	378	0,121	16,4
200x15	500	445	0,091	31,8	470	0,018	35,9
300x15	486	389	0,252	47,8	454	0,014	53,6
402x15	579	388	0,808	65,4	540	0,026	81,2
Total	2283	1881	1,306	167,9	2158	0,182	195
50x20	377	363	0,019	8,1	371	0,007	8,5
100x20	611	526	0,077	17,2	603	0,008	19,3
200x20	626	470	0,220	37,3	607	0,006	44,7
300x20	702	608	0,132	56,9	639	0,072	68,8
402x20	755	581	0,343	77,6	707	0,013	106,3
Total	3071	2548	0,791	197,1	2927	0,106	247,6

6. Conclusões

Neste artigo foi abordado o problema de localização de p -medianas na versão bi-objetivo no qual são minimizados a soma das distâncias de atendimento facilidade-cliente e a soma dos custos de abertura das p facilidades. As soluções Pareto-ótimas encontradas pelo método ϵ -Restrito evidenciam que os objetivos otimizados são medidas diferentes e conflitantes. Uma solução que fornece as menores distâncias de atendimento possui os custos muito altos de abertura das facilidades, e vice versa. Para determinar soluções não-dominadas de problemas de grande porte, foi proposta uma adaptação de um Algoritmo Genético multiobjetivo bastante utilizada na literatura. Para este algoritmo foi proposto uma fase de intensificação baseada na técnica *Path Relinking* que consiste na combinação de soluções geradas pelos operadores genéticos com soluções não-dominadas encontradas até o momento. Neste trabalho mostra-se a eficiência da técnica *Path Relinking* na resolução do problema das p -medianas bi-objetivo, ou seja, na determinação de soluções não-dominadas de excelente qualidade. Dos resultados obtidos pode-se observar que o Algoritmo Genético multiobjetivo com *Path Relinking* (AGMO+PR) determina um numero maior de soluções não-dominadas de referência quando comparado com o algoritmo AGMO, que não utiliza a fase de intensificação.

Agradecimentos:

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais – FAPEMIG pelo financiamento do trabalho. À empresa *Dash Optimization* pela licença cedida da versão completa do *Xpress-MP*.

Referências

- Alp, O., Erkut, E. e Drezner, Z. (2003). An Efficient Genetic Algorithm for the p-Median problem,” *Annals of Operations Research* 122, p. 21–42.
- Arroyo, J.E.C. e Armentano, V.A. (2005). Genetic Local Search for Multi-Objective Flowshop Scheduling Problems. *European Journal of Operational Research*, 167, 3, p. 717-738.
- Czyzak, P. e Jaskiewicz, A. (1998). Pareto simulated annealing - a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, p. 734-47.

- Deb, K., Agrawal, S., Pratab, A. e Meyarivan, T.** (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6,(2) 182-197.
- Drezner, Z.** (Editor). (1995). *Facility Location: a Survey of Spplications and Methods*. Springer-Verlag, p. 571, New York.
- Fernandez, E. e Puerto, J.** (2003). Multiobjctive Solution of Uncapacitated Plant Location Problem. *European Journal of Operational Research*, v. 145, p. 509-529.
- Gandibleux X. e M. Ehrgott.** (2005). “1984-2004 - 20 Years of Multiobjective Metaheuristics. But What About the Solution of Combinatorial Problems with Multiple Objectives?.” *Third International Conference, EMO, Springer. Lecture Notes in Computer Science*. Vol. 3410, p. 33-46.
- Garey, M.R. e Johnson, D.S.** (1979). *Computers and intractability: a guide to the theory of NPcompleteness*. W. H. Freeman and Co., San Francisco.
- Glover, F.** (1996). Tabu search and adaptive memory programming - advances, applications, and challenges, In: *Interfaces in Computer Science and Operations Research*, p. 1 – 75.
- Ishibuchi, H. e Murata, T.** (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man and Cybernetics-part C: Applications and Reviews*, 28, p. 392-403.
- Jones, D.F., Mirrazavi, S.K. e Tamiz, M.** (2002). Multi-objective meta-heuristics: An overview of the current state-of-art. *European Journal of Operational Research*, 137, p. 1-19.
- Mavrotas, G. e Diakoulaki, D.** (1998). A Branch and Bound Algorithm for Mixed Zero-One Multiple Objective Linear Programming. *European Journal of Operational Research*, v. 107, p. 530-541.
- Medaglia, A. L., Villegas, J.G. e Rodríguez-Coca D.M.** (2009). Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. *Journal of Heuristics* 15(2): 153-176.
- Mladenović, N., Brimberg, J., Hansen, P. e Moreno-Pérez, J.A.** (2007). The p-median problem: A survey of metaheuristic approaches, *European Journal of Operational Research* 179: 927–939.
- Nozick, L. K.** (2001). The Fixed Charge Facility Location Problem with Coverage Restrictions. *Transportation Research Part E*, v. 37, p. 281-296.
- Owen, S. H.; Daskin, M. S.** (1998). Strategic Facility Location: A Review. *European Journal of Operational Research*, v. 111, pp. 423-447,
- Reese, J.** (2006). Solution methods for the p-median problem: An annotated bibliography, *Networks* v.48 (3), 125 – 142.
- Resende, M. G. C. e Werneck, R. F.** (2005). A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, v. 174, n. 1, p. 54-68.
- ReVelle, C.; Swain, R.** (1970). Central Facilities Location, *Geographical Analysis*, 2: 30–42..
- Soares, M. S., Arroyo, J. E. C. e Santos, P. M.** (2009). Um algoritmo GRASP para o problema das p-medianas bi-objetivo. *XXXV Conferência Latinoamericana de Informática: CLEI, Pelotas, RS*.
- Villegas, J. G., Palacios, F. e Medaglia, A. L.** (2006). Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. *Annals of Operations Research*, v.147, n. 1, p. 109-141.