



APLICAÇÃO DA RELAXAÇÃO LAGRANGEANA E DO ALGORITMO GENÉTICO CONSTRUTIVO NA SOLUÇÃO DO PROBLEMA PROBABILÍSTICO DE LOCALIZAÇÃO-ALOCAÇÃO DE MÁXIMA COBERTURA

Francisco de Assis Corrêa
Luiz Antonio Nogueira Lorena

Laboratório Associado de Computação e Matemática Aplicada,
Instituto Nacional de Pesquisas Espaciais – INPE,
Av. dos Astronautas, 1758, Jardim da Granja,
CEP 12227-010, São José dos Campos, SP, Brasil,
e-mails: fcorrea@directnet.com.br, lorena@lac.inpe.br

Recebido em 22/7/2005
Aceito em 10/4/2006

Resumo

O problema de localização de máxima cobertura (MCLP) procura localizar facilidades visando a maximizar a população atendida, considerando uma dada distância ou tempo padrão de serviço. Várias extensões desse modelo têm sido propostas para aumentar a sua aplicabilidade. Entre elas, existem modelos probabilísticos para localização-alocação de máxima cobertura com restrições no tempo de espera ou no comprimento da fila para sistemas congestionados, que levam em conta um ou vários servidores por facilidade. A proposta deste trabalho é a de resolver um modelo para um servidor por facilidade por meio da relaxação lagrangeana e do Algoritmo Genético Construtivo. Os resultados dos testes obtidos nessas abordagens são apresentados e comparados.

Palavras-chave: problemas de localização, máxima cobertura, relaxação lagrangeana, algoritmo genético construtivo.

1. Introdução

O Problema de Localização de Máxima Cobertura (MCLP) tem sido consideravelmente tratado na literatura desde a sua formulação feita por Church e ReVelle (1974). Esse problema busca obter a configuração para localizar facilidades que atenda o maior número de indivíduos de uma população, considerada uma dada distância ou um tempo padrão do ponto de demanda. Considerável revisão desse tema pode ser encontrada em Hale e Moberg (2003), Serra e Marianov (2004) e Galvão (2004). Não se busca com este modelo atender toda a população, mas oferecer o máximo de atendimento, considerando os recursos disponíveis. Vários modelos aplicados a uma grande faixa de problemas são extensões dessa formulação.

Em muitos trabalhos envolvendo problemas de localização, essa distância (ou tempo) entre pontos de demanda e facilidade à qual eles estão sendo alocados é o fator que representa a qualidade dos serviços que são prestados a usuários. Entretanto, quando se projeta uma rede de

serviços, como sistemas de saúde, bancários ou de vendas de bilhetes distribuídos, a localização dos centros tem uma forte influência no congestionamento de cada um, e, conseqüentemente, a qualidade de serviços deve ser mais bem modelada e não considerar apenas a distância ou o tempo de deslocamento. Assim, deve-se cuidar para que sua localização permita aos usuários chegar ao centro em um tempo aceitável, e que o tempo de espera para atendimento seja o menor possível ou que a quantidade de pessoas na fila seja mínima, uma vez que esses são parâmetros importantes na medida da qualidade desejada (Marianov e Serra, 1998).

O congestionamento ocorre quando um centro não é capaz de atender, simultaneamente, a todas as solicitações de serviços que lhe são feitas. Os modelos tradicionais que tratam desse problema adicionam uma restrição de capacidade que força a demanda por serviço, normalmente constante no tempo e igual a uma média, a

ser menor do que a máxima capacidade do centro. Essa abordagem não considera a natureza dinâmica do congestionamento e trata o problema de forma determinística. Isso faz com que o modelo, dependendo de como a restrição seja obtida, tenha servidores ociosos ou não tenha a capacidade de atender a todas as demandas (Marianov e Serra, 1998; 2001).

Marianov e Serra (1998) propuseram modelos baseados no fato de que o número de solicitações de serviços não é constante no tempo, mas um processo estocástico, cuja estocasticidade de demanda é explicitamente considerada no trato das restrições de capacidade, que, ao invés de serem limitadas a um máximo, têm um limite mínimo, definido pelos autores, para a qualidade dos seus serviços. Essa qualidade é refletida nos itens de tempo de espera ou quantidade de pessoas que aguardam por um atendimento.

Os modelos propostos por esses pesquisadores tratam de localizar certa quantidade de postos de saúde, com um ou vários servidores, de modo que a população, a uma distância padrão do centro, seja servida adequadamente, isto é, que ninguém fique na fila por um período maior que um dado tempo limite nem encontre um número de outros clientes acima do previsto, com uma probabilidade maior ou igual α_p , definida *a priori*. Foram considerados os sistemas de fila M/M/1 e M/M/m (Larson e Odoni, 1981), em que a taxa de chegada no centro j segue a distribuição de Poisson com taxa λ_j e o tempo de serviço no centro j é exponencialmente distribuído com taxa μ_j , para, respectivamente, um ou vários servidores.

O modelo de localização-alocação de máxima cobertura considerado neste artigo foi denominado por Marianov e Serra (1998) por “*The Queueing Maximal Covering Location-Allocation Model (QM-CLAM)*” e foi definido como:

“Localizar p centros e alocar usuários de forma a maximizar a população coberta, onde a cobertura é definida como: a população coberta é alocada a um centro dentro de um tempo ou distância padrão de sua residência (ponto de demanda) e se um usuário é coberto, na sua chegada ao centro, ele vai esperar em uma fila com no máximo b outras pessoas ou será atendido no tempo τ , com probabilidade, no mínimo, α_f ”.

O propósito deste artigo é o de examinar o Problema de Localização-Alocação de Máxima Cobertura Probabilístico proposto por Marianov e Serra (1998), para um servidor por centro, e apresentar outras duas soluções: a primeira, por meio da relaxação Lagrangeana; e a segunda, usando o Algoritmo Genético Construtivo (AGC). A Seção 2 descreve os modelos originais propostos; a Seção 3 apresenta a solução usando a relaxação lagrangeana e AGC; a Seção 4 mostra os resultados computacionais; e a Seção 5, as conclusões.

2. Modelo QM-CLAM

O modelo tradicional de *Maximum Covering Location Problem (MCLP)*, proposto por Church e ReVelle (1974), não pode ser usado para tratar as restrições de congestionamento, pois não contém variáveis de alocação, o que impede de computar as solicitações de serviços que chegam a um centro e , conseqüentemente, de determinar quando ocorre um congestionamento.

A modelagem matemática proposta para o *QM-CLAM* foi escrita como um problema tipo p -medianas (Lorena, 2003), modificado para comportar as variáveis de localização e alocação, tendo como objetivo maximizar a população coberta, considerando uma determinada quantidade de centros de atendimento. Formalmente, as alocações são representadas pelas variáveis binárias x_{ij} , tal que $i \in I$ e $j \in N_j$, em que I é o conjunto dos pontos de demanda a serem alocados e N_j é o conjunto de localizações candidatas que estão dentro de uma distância padrão do nó i . $[x_{ij}]$ é a matriz de alocações, com $x_{ij} = 1$, se o ponto de demanda i for alocado ao centro j , $x_{ij} = 0$, caso contrário. As localizações são representadas pelas variáveis binárias y_j , com $y_j = 1$, se o centro j for selecionado, $j = 0$, caso contrário. O parâmetro a_i define a população total no ponto de demanda i . Todo ponto de demanda é um potencial centro de atendimento. O *QM-CLAM* possui a formulação descrita a seguir:

Problema *QM-CLAM*

$$v(\text{QM-CLAM}) = \text{Max} \sum_{ij} a_i x_{ij} \quad (1)$$

Sujeito a

$$x_{ij} \leq y_j \quad \forall i, j \quad (2)$$

$$\sum_{j \in N_j} x_{ij} \leq 1 \quad \forall i \quad (3)$$

$$\sum_{i \in I} f_i x_{ij} \leq \mu_j \cdot b + \sqrt{1 - \alpha_f} \quad \forall j, \quad (4)$$

(para a quantidade de pessoas na fila)

ou

$$\sum_{i \in I} f_i x_{ij} \leq \mu_j + \frac{1}{\tau} \ln(1 - \alpha_f) \quad \forall j, \quad (4a)$$

(para o tempo de espera)

$$\sum_i y_i = p \quad (5)$$

$$y_j, x_{ij} \in \{0, 1\}, \quad j \in N_j \quad \forall i, j \quad (6)$$

A função objetivo (1) maximiza a população alocada a um centro. As restrições (2) definem que somente é possível alocar um ponto de demanda i a um centro j se houver um centro em j . As restrições (3) impõem que cada ponto de demanda seja alocado a, no máximo, um

centro. Usam-se as restrições (4) ou (4a) para, respectivamente, tratar de quantidade de pessoas na fila ou do tempo de espera. As primeiras, restrições (4), forçam para que cada centro tenha no máximo b pessoas na fila, com a probabilidade, no mínimo, α_r . As segundas, restrições (4a), determinam que o tempo gasto no centro j seja, no máximo, τ , com a probabilidade, no mínimo, α_r . A restrição de cardinalidade (5) define a quantidade de centros a serem abertos. As restrições (6) definem as condições de integralidade.

Para escrever as restrições (4), considera-se o sistema de filas M/M/1 e que as solicitações de serviços de cada nó de demanda i acontecem de acordo com um processo de Poisson com taxa f_i . A taxa λ_j atribuída a um centro é definida como uma superposição de processos de Poisson:

$$\lambda_j = \sum_{i \in I} f_i x_{ij}$$

o que significa que, se a variável x_{ij} for 1, o nó i será alocado ao centro j e a sua correspondente taxa f_i será incluída no cálculo de λ_j . O desenvolvimento de toda a formulação pode ser visto em Marianov e Serra (1998).

Assume-se também que o tempo de serviço tem uma distribuição exponencial, com taxa média μ_j , em que $\mu_j \geq \lambda_j$, de forma a manter o equilíbrio do sistema.

O lado direito das Equações 4 e 4a são valores constantes, calculados para μ_j , α_r , b e τ , definidos a priori, em que μ_j é a taxa média de serviço no centro j , b é a quantidade máxima de pessoas na fila, τ é o tempo máximo de espera no centro j e α_r é a probabilidade de que sejam encontradas, no máximo, b pessoas na fila ou espera de, no máximo, um tempo τ para ser atendido. Para efeito de simplificação, serão substituídos, respectivamente, por $K_{\mu b \alpha_r}$ e por $K_{\mu \tau \alpha_r}$.

$$\sum_{i \in I} f_i x_{ij} \leq K_{\mu b \alpha_r} \quad \forall j, \tag{7}$$

ou

$$\sum_{i \in I} f_i x_{ij} \leq K_{\mu \tau \alpha_r} \quad \forall j \tag{7a}$$

O problema QM-CLAM pertence à classe dos problemas NP-Completo (Pirkul e Schilling, 1991). Aplicativos comerciais podem resolver algumas pequenas instâncias desse problema. Entretanto, o tempo de computação se torna inviável para problemas de tamanhos razoáveis. Por isso, buscam-se soluções alternativas eficientes para formulações desse tipo. No próximo capítulo, duas abordagens são apresentadas: uma baseada na aplicação da relaxação lagrangeana com o algoritmo de otimização por subgradientes; e outra, por meio do Algoritmo Genético Construtivo.

3. Soluções propostas

3.1 Relaxação lagrangeana

A relaxação lagrangeana tem sido aplicada na solução de vários problemas de otimização combinatória, como: o do caixeiro viajante; os de localização, de recobrimento e particionamento de conjuntos; o generalizado de atribuição; e os de roteamento (Espejo e Galvão, 2002; Lorena e Senne, 2003). Essa relaxação é obtida multiplicando-se um conjunto de restrições por um vetor λ de multiplicadores de Lagrange, adicionando-se o produto à função objetivo. O desenvolvimento dessa relaxação para o Problema de Localização de Máxima Cobertura pode ser visto em Daskin (1995). Para o problema (QM-CLAM), relaxando as restrições de alocação (3) no sentido lagrangeano e incorporando-as à função objetivo, tem-se:

$$\sum_{i \in I} \lambda_i \left(\sum_{j \in N_i} x_{ij} - 1 \right), \lambda \geq 0$$

$$V(L_\lambda \text{QM-CLAM}) = \text{Max} \sum_i \sum_j a_i x_{ij} - \sum_i \lambda_i \left(\sum_j x_{ij} - 1 \right)$$

$$V(L_\lambda \text{QM-CLAM}) = \text{Max} \sum_i \sum_j a_i x_{ij} - \sum_i \lambda_i \sum_j x_{ij} + \sum_i \lambda_i$$

$$V(L_\lambda \text{QM-CLAM}) = \text{Max} \sum_i \sum_j (a_i - \lambda_i) x_{ij} + \sum_i \lambda_i$$

Portanto, o problema L_λ QM-CLAM pode ser escrito:

$$V(L_\lambda \text{QM-CLAM}) = \text{Max} \sum_i \sum_j (a_i - \lambda_i) x_{ij} + \sum_i \lambda_i$$

Sujeito a (2), (5), (6) e (7) ou (7a).

As restrições (8) substituem as restrições (2) e (7), pois exigem que somente seja possível alocar um ponto de demanda i a um centro j se houver um centro em j , como fazem as restrições (2), além de imporem restrições de capacidade, como fazem as (7) (Murray e Gerrard, 1997). Do mesmo modo, as restrições (8a) substituem as restrições (2) e (7a).

$$\sum_{i \in I} f_i x_{ij} \leq K_{\mu b \alpha_r} y_j \quad \forall j, \tag{8}$$

$$\sum_{i \in I} f_i x_{ij} \leq K_{\mu \tau \alpha_r} y_j \quad \forall j \tag{8a}$$

O problema L_λ QM-CLAM pode ser decomposto em $|I|$ subproblemas. Em cada um deles, a variável de localização y_j pode ser igual a zero ou a um. Se for zero, todas as variáveis de alocação x_{ij} serão também zero e seus correspondentes valores na formulação tornam-se zero. Quando y_j tem o valor um, os subproblemas tornam-se o problema da mochila 0-1 e podem ser resolvidos independentemente. Se forem tomados os p melhores resultados, a restrição de cardinalidade (5) estará sendo considerada implicitamente (Lorena e Senne, 2003; Pirkul e Schilling, 1991).

Assim, o problema original pode ser reescrito com um conjunto de subproblemas:

Para $j = 1, 2, \dots, |I|$,

$$v(\text{Knap}^j) = \text{Max} \sum_{i \in I} (a_i - \lambda_i) x_{ij} \quad (9)$$

Sujeito a:

$$\sum_{i \in I} f_i x_{ij} \leq K_{\mu b \alpha_j} \quad \forall j, \quad (8)$$

ou

$$\sum_{i \in I} f_i x_{ij} \leq K_{\mu \tau \alpha_j} \quad \forall j \quad (8a)$$

$$x_{ij} \in \{0, 1\}, \quad j \in N_i \quad \forall i, j \quad (10)$$

Cada problema é resolvido usando-se o código Horowitz e Sahni (Martello e Toth, 1990). Esse código ofereceu bons resultados, embora tenha sido desenvolvido para problemas inteiros de mochila.

Sendo J o conjunto dos p maiores $v(\text{Knap}^j)$, $j \in I$, o valor da relaxação lagrangeana é dado por:

$$v(L_\lambda \text{QM-CLAM}) = \sum_{j \in J} v(\text{Knap}^j) + \sum_{i \in I} \lambda_i \quad (11)$$

A otimização do dual Lagrangeano é obtida por meio do algoritmo de subgradientes (Lorena e Senne, 2003; Pirkul e Schilling, 1991). Dado um vetor de multiplicadores $(\lambda)^0$, a seqüência de multiplicadores é obtida por:

$$\lambda^{k+1} = \text{Max} \left(0, \lambda^k + \theta_k \left(\sum x_{ij}^\lambda - 1 \right) \right), \quad (12)$$

em que x^λ é a solução ótima do problema $L_\lambda \text{QM-CLAM}$ com multiplicadores $(\lambda)^k$ e θ_k é um escalar positivo definido por:

$$\theta = \frac{\pi (v(\text{QM-CLAM}) - lb)}{\| \sum x_{ij}^\lambda - 1 \|^2}, \quad (13)$$

em que $v(\text{QM-CLAM})$ é o melhor valor do limitante superior, lb é o melhor valor do limitante inferior (solução viável de QM-CLAM) e π é o parâmetro de controle definido por Held e Karp (1970). O valor de π inicial é 2 e é dividido à metade se o valor de $v(\text{QM-CLAM})$ não diminuir por 30 iterações sucessivas.

Os critérios de parada mais utilizados na literatura e aplicados na resolução dos problemas deste artigo são:

- $\pi \leq 0,0005$;
- $v(\text{QM-CLAM}) - lb < 1$;
- $\| \sum x_{ij}^\lambda - 1 \|^2 = 0$; e
- Quantidade de iterações.

As soluções x^λ obtidas não são necessariamente viáveis, porém o conjunto J indica as facilidades que podem ser usadas para se obter soluções viáveis. Para alocar os

pontos de demanda ao conjunto de facilidades identificadas, busca-se a solução aproximada para a seguinte variante do Problema de Atribuição Generalizado (GAP) com desigualdades nas restrições (15), sendo os operadores relacionais menor ou igual (LEGAP):

$$\text{Max} \sum_{i \in I} \sum_{j \in J} a_i x_{ij}^f \quad (14)$$

Sujeito a

$$\sum_{j \in J} x_{ij}^f \leq 1 \quad i \in I \quad (15)$$

$$\sum_{i \in I} f_i x_{ij}^f \leq K_{\mu b \alpha_j} \quad j \in J, \quad (16)$$

ou

$$\sum_{i \in I} f_i x_{ij}^f \leq K_{\mu \tau \alpha_j} \quad j \in J \quad (16a)$$

$$x_{ij}^f \in \{0, 1\}, \quad i \in I \quad j \in J \quad (17)$$

Essas soluções primais foram obtidas por meio do algoritmo MTHG de Martello e Toth (1990), adaptado para contemplar as desigualdades, pois, ao contrário do GAP, o LEGAP sempre admite solução viável, e foram melhoradas pelo algoritmo descrito em Pereira e Lorena (2001). Este último leva a obter aumentos no valor da função objetivo (v^f), correspondentes a uma solução primal x^f , substituindo-se uma facilidade por um vértice do mesmo cluster e recalculando-se o atendimento dos pontos de demanda após essa troca, pois a configuração de cobertura desses pontos pode ser modificada após essa substituição, como pode ser visto na Figura 1.

O algoritmo de melhoria de soluções primais é apresentado na Figura 2.

3.2 Algoritmo genético construtivo

O algoritmo genético foi concebido por Holland, em 1960, e aperfeiçoado nas décadas de 1960 e 1970, por ele próprio e por seus colegas da Universidade de Michigan. Vários refinamentos do método surgiram nas décadas seguintes. Alguns deles e informações básicas podem ser vistos em Goldberg (1989) e em Lacerda (1999).

O Algoritmo Genético Construtivo (AGC) (Furtado, 1998; Lorena e Furtado, 2001; Oliveira e Lorena, 2005) trabalha com uma população inicial composta por estruturas e esquemas. As estruturas referem-se a qualquer cadeia contendo ou não símbolo #; os esquemas fazem referência explícita a uma estrutura que contém o símbolo #, isto é, uma população de soluções candidatas parciais ou incompletas, que servirão de base para a construção de uma população com soluções melhores completas, ao longo do processo evolutivo. O símbolo # é chamado de curinga e representa um ponto ainda não definido para o problema. Neste artigo, esse símbolo receberá o valor 1, tornando-se uma facilidade, ou o valor 2, tornando-se um ponto

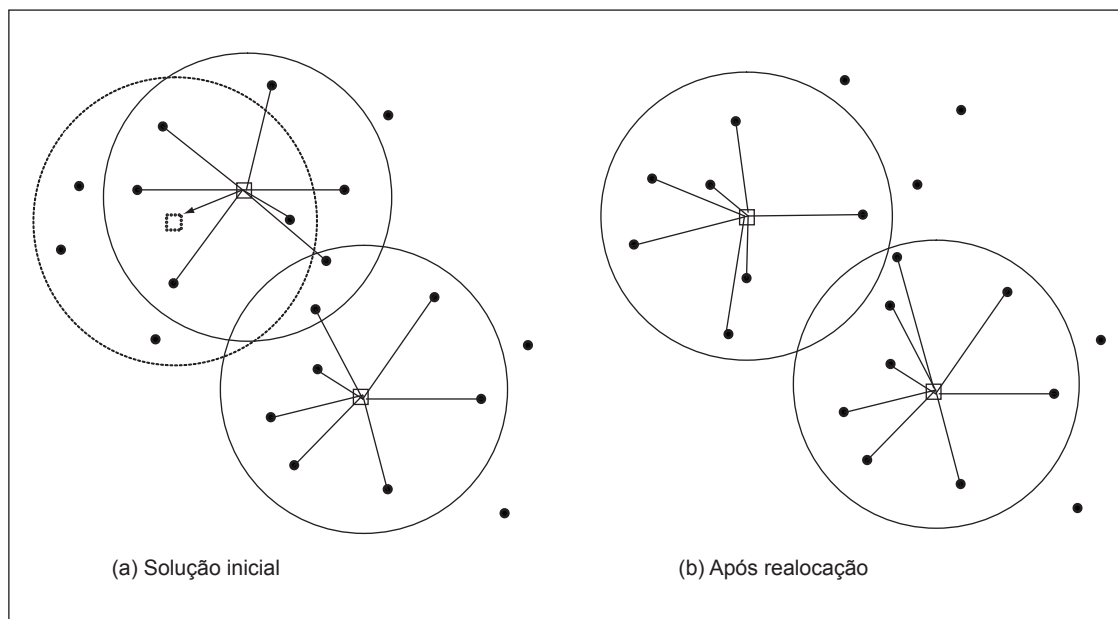


Figura 1. Realocação de facilidade por ponto de demanda (adaptado de Pereira e Lorena (2001)).

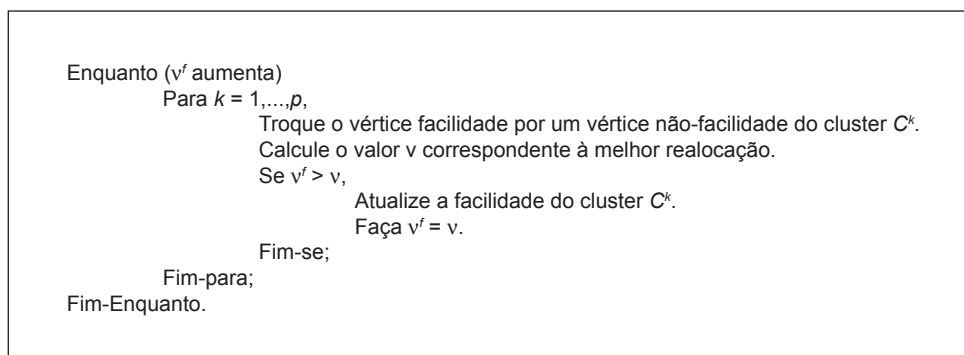


Figura 2. Algoritmo de melhoria de soluções primais (adaptado de Pereira e Lorena (2001)).

de demanda que será atribuído a uma facilidade durante a evolução do processo. Desta forma, para os problemas discutidos neste artigo, um esquema pode ser representado por uma cadeia que apresenta três tipos de informações:

- 1 : vértice que representa uma facilidade;
- 2 : vértice que representa um ponto de demanda que é atribuído a uma facilidade; e
- # : vértice curinga, que se tornará uma facilidade ou um ponto de demanda a ser atribuído a uma facilidade durante o processo evolutivo.

A Figura 3 mostra um exemplo de uma cadeia para um problema com 10 pontos e 2 centros.

O AGC possui os operadores tradicionais de seleção, cruzamento e mutação, e difere dos algoritmos tradicionais na forma de avaliar os esquemas (avaliação-fg), na

possibilidade de usar heurísticas para definir a função de avaliação da aptidão dos indivíduos e no tratamento de uma população de tamanho variável (Oliveira, 2004).

A avaliação-fg trata de uma dupla avaliação de cada indivíduo $S_k \in P_\alpha$, que é a população no instante de evolução α . O valor de f reflete a função objetivo, $f: P_\alpha \rightarrow \mathbb{R}^+$, e o valor de g é calculado a partir de uma heurística, $g: P_\alpha \rightarrow \mathbb{R}^+$, tal que $g(S_k) \geq f(S_k)$, para todo $S_k \in P_\alpha$. A primeira, função f , avalia a qualidade do indivíduo que é representada pelo valor da função objetivo, calculado pela soma das demandas dos pontos alocados aos centros. A segunda, função g , aplica uma heurística para avaliar a vizinhança do indivíduo, atribuindo a melhor solução encontrada ao valor de g . A heurística adotada usa a idéia discutida na melhoria das soluções primais, apresentada em 3.2, Figura 2. Essa melhoria é buscada realizando-se

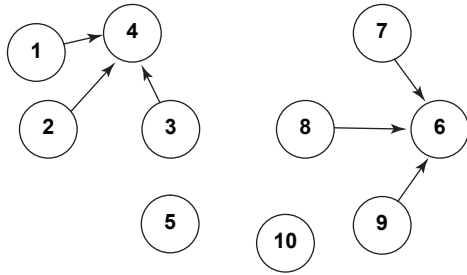


Figura 3. Representação de $S = (2221\#1222\#)$ (adaptado de Furtado (1998)).

trocias aleatórias de um vértice que é ponto de demanda por um que é centro, redefinindo as alocações e recalculando o valor da função objetivo.

O indivíduo original não estará bem adaptado se a heurística encontrar uma solução melhor. Caso contrário, o indivíduo é o melhor dentro da vizinhança estabelecida pela heurística e deve participar o máximo possível do processo evolutivo. Essa dupla avaliação é aplicada da mesma forma para qualquer tipo de indivíduo S_k , assim como para esquemas e estruturas. Deve ser definido também um limite superior comum $G_{\max} > \text{Max}_{S_k \in P_\alpha} g(S_k)$. A

Figura 4 mostra uma possível avaliação de uma população P_α .

O AGC trabalha com uma população de tamanho variável, com a população inicial gerada aleatoriamente. A cada geração do processo evolutivo, novos indivíduos são criados, um para cada cruzamento entre dois indivíduos existentes, privilegiando os mais bem adaptados. Esses novos indivíduos podem sofrer mutação, dentro de uma probabilidade definida *a priori*.

Um problema a ser modelado em AGC deve ser tratado como um problema de otimização bi-objetivo (PBO):

$$\begin{cases} \min & \{g(S_k) - f(S_k)\} \\ \max & \{f(S_k)\} \\ \text{sujeito a} & g(S_k) \geq f(S_k) \end{cases}$$

Para qualquer problema de otimização, o PBO é formulado da mesma forma: maximizar a função objetivo e minimizar o intervalo $(g-f)$. O processo evolutivo considera um limiar de rejeição adaptativo que contempla ambos os objetivos do PBO, que é calculado a partir de um *ranking* δ atribuído a cada indivíduo da população, dado pela Equação 18, que é composta de:

- Um componente referente à adaptação do indivíduo $(g-f)$. Quanto menor essa diferença, mais adaptado está o indivíduo, ou seja, o indivíduo está próximo a um ótimo local determinado pela heurística de treinamento e deve permanecer o máximo de gerações no processo evolutivo;

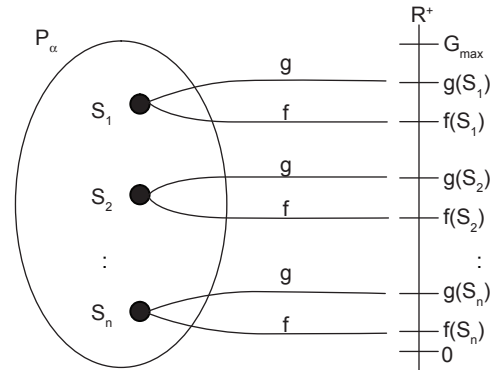


Figura 4. Possível avaliação de uma população (adaptado de Furtado (1998)).

- Um componente que privilegia a maximização da função g , calculando a distância em termos de avaliação entre o indivíduo e o limitante superior G_{\max} ($G_{\max} - g(S_k)$); e
- Uma constante $d \in [0, 1]$, que tem o papel de equilibrar os componentes da equação.

$$\delta = \frac{d \cdot G_{\max} - [g(S_k) - f(S_k)]}{d \cdot [G_{\max} - g(S_k)]} \quad (18)$$

A população inicial, denominada P_0 , é formada apenas por esquemas. Considerando os problemas tratados neste artigo, para cada indivíduo, uma porcentagem das posições recebe aleatoriamente o valor 2 e exatamente p posições recebem o valor 1, definindo, assim, as p facilidades a serem abertas. As posições restantes recebem o valor $\#$. Para os testes computacionais descritos na Seção 4, 20% das posições receberam o valor 2. A Figura 5 mostra um possível indivíduo da população inicial para um problema com quatro centros e trinta nós.

Os indivíduos são ordenados por valores decrescentes de δ . O tamanho da população é controlado dinamicamente pelo limiar de rejeição α , que está relacionado com o instante de evolução e serve para eliminar os indivíduos mal adaptados ($\delta(S_k) \leq \alpha$). O valor de α é calculado pela Equação 19, na qual δ_1 define o melhor valor de *ranking*; $\delta_{|P|}$, o pior valor de *ranking*; ε é uma constante que controla a velocidade de esvaziamento da população; $|P|$ é o tamanho da população no instante α ; RG representa o número de gerações restantes no processo evolutivo; e l é um valor que garante um passo mínimo nesse processo.

$$\alpha_i = \alpha_{i-1} + \varepsilon \cdot |P| \cdot \frac{(\delta_1 - \delta_{|P|})}{RG} + l \quad (19)$$

Considerando-se que os bons esquemas precisam ser preservados para serem recombinados, α é iniciado com o valor 0 (zero) e é lentamente incrementado, de geração em geração.

```
##2#1####2##1#2##2#2##1###1#2#
```

Figura 5. Possível indivíduo para um problema com quatro centros e trinta nós.

Desde que são criados, os esquemas e estruturas recebem os seus correspondentes valores de *ranking* (δ), que são comparados com o parâmetro α . Todo indivíduo com $\delta(S_k) \leq \alpha$ é considerado mal adaptado e é eliminado da população. Conseqüentemente, os indivíduos com maiores δ são os melhores em relação ao PBO, sobrevivem por mais gerações e se reproduzem mais. Com isso, a população no tempo de evolução α (P_α) possui tamanho dinâmico de acordo com o valor de α , conforme exemplo na Figura 6, podendo ser esvaziada durante o processo. Sempre o melhor indivíduo de cada geração é guardado e ao final do processo evolutivo a melhor solução encontrada é retornada.

O operador de seleção base-guia escolhe duas estruturas para o cruzamento. A primeira, denominada base (S_{base}), é obtida dos 20% melhores indivíduos da população. A segunda, denominada guia (S_{guia}), é selecionada da população total. O operador de cruzamento compara essas duas estruturas em posições correspondentes, gerando uma nova estrutura filha (S_{nova}), com a seguinte regra aplicada aos problemas tratados neste artigo, baseada em Furtado (1998):

- $S_{base} = \#$ e $S_{guia} = \#$ então $S_{nova} = \#$
- $S_{base} = 1$ e $S_{guia} = 1$ então $S_{nova} = 1$
- $S_{base} = 2$ e $S_{guia} = 2$ então $S_{nova} = 2$
- $S_{base} = 1$ e $S_{guia} = \#$ então $S_{nova} = 1$
- $S_{base} = 2$ e $S_{guia} = \#$ então $S_{nova} = 2$
- $S_{base} = \#$ e $S_{guia} = 2$ então $S_{nova} = 2$
- $S_{base} = \#$ ou 2 e $S_{guia} = 1$ então $S_{nova} = 1$ ou 2, escolhido aleatoriamente
- $S_{base} = 1$ e $S_{guia} = 2$ então $S_{nova} = 1$ ou 2, escolhido aleatoriamente

Após a aplicação do operador de cruzamento, pode-se ter mais do que p valores 1 em S_{nova} . Essa condição não é relaxada e uma validação da nova estrutura é feita para se garantir exatamente p facilidades, inserindo ou eliminando, aleatoriamente, valores 1 nessa nova estrutura.

O operador de mutação faz uma troca de um centro com um vizinho que não tem a sua demanda atendida por alguém. Isso aumenta a chance de todos os pontos de demanda serem centros. O algoritmo do operador de mutação é mostrado na Figura 7.

Verificar se existem pontos de demanda que não são atendidos

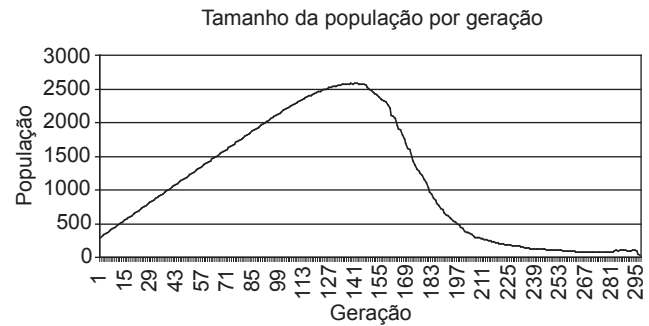


Figura 6. Tamanho da população a cada geração.

```

Verificar se existem pontos de demanda que não são atendidos
Se existirem
Então
    Escolher um deles aleatoriamente e atribuir-lhe o valor 1
    (defini-lo como facilidade)
    Escolher uma posição que contenha uma facilidade e
    atribuir-lhe o valor 2
Fim-se;

```

Figura 7. Operador de mutação.

As restrições apresentadas na formulação do problema foram tratadas na avaliação da função objetivo ou na própria definição do indivíduo.

O algoritmo para o AGC pode ser resumido no *pseudocódigo* mostrado na Figura 8.

4. Experiência computacional

A heurística lagrangiana e o AGC foram testados na rede de 30 vértices, fornecida em Marianov e Serra (1998), e em uma de 324 vértices, obtida de uma base de dados geográficos da cidade de São José dos Campos-SP, acrescida da população fictícia em cada ponto de demanda e que está disponível em <http://www.lac.inpe.br/~lorena/instancias.html>. Vários problemas foram montados, variando-se os parâmetros p , b , μ , α_i e τ . Os resultados dessas duas abordagens foram comparados aos obtidos pelo uso do *software* comercial CPLEX 7.5 (ILOG, 2001). Os valores apresentados no artigo de Marianov e Serra (1998) foram obtidos por meio de uma heurística e pelo uso do mesmo *software*, porém na versão 3.0. A versão 7.5 do CPLEX mostrou melhores resultados para a mesma formulação matemática que os apresentados naquele artigo, para as duas soluções (heurística e CPLEX 3.0). Por isso, apenas os obtidos com a versão 7.5 foram considerados para comparar com os resultados obtidos com a heurística lagrangeana e o AGC neste trabalho.

Para a implementação do problema de máxima cobertura, os centros de serviços foram considerados postos de saúde, com um médico em cada centro. Cada ponto de demanda é um potencial centro de atendimento e as distâncias são euclidianas. Para os problemas com 30 pontos foram considerados: raio de cobertura igual a 1,5 milhas, tempo médio de atendimento ($1/\mu$) igual a 20 minutos; taxa de chamada igual a $0,015 \times$ população do ponto de demanda para as restrições do comprimento da fila e $0,006 \times$ população do ponto de demanda para as restrições do tempo de espera na fila, todos definidos em Marianov e Serra (1998). Para os problemas com 324 pontos, considerou-se: raio de cobertura igual a 250 m, tempo de atendimento igual a 15 minutos; taxa de chamada igual a $0,01 \times$ população do ponto de demanda, tanto para as restrições do comprimento da fila, como para as restrições do tempo de espera.

As Tabelas 2 e 3 mostram os resultados obtidos na avaliação da função objetivo por todos os métodos. Os valores de *Gap Cplex* e *Gap Lag* iguais a zero definem que o valor ótimo foi obtido para o respectivo método. Esses valores são calculados por: $100 * (\text{limite superior} - \text{limite inferior}) / (\text{limite inferior})$, e fornece, em porcentagem, o *gap* entre esses limitantes.

Os nomes dos problemas foram codificados do seguinte modo: quantidade de pontos, quantidade de centros, tipo de restrição (0 para a quantidade de pessoas na fila e 1 para o tempo de espera), quantidade de pessoas na fila ou tempo de espera, e probabilidade. Exemplo: 324_20_

```

Defina  $G_{\max}$ ,  $d$ ,  $\epsilon$  e  $l$ ;
 $\alpha = 0$ ;
Defina a probabilidade de mutação;
Defina a população inicial ( $P\alpha$ );
Para todo  $S_k$  e  $P_\alpha$  faça
  Calcule  $f(S_k)$ ,  $g(S_k)$ ,  $\delta(S_k)$ ; // O cálculo de  $\delta(S_k)$  é feito
  segundo a fórmula (18)
Fim-Para;
Enquanto (condição de parada não for satisfeita) faça
  Enquanto (não atingir a quantidade de cruzamentos prevista)
  faça
    Seleção ( $S_{base}$ ,  $S_{guia}$ );
     $S_{nova} =$  Cruzamento ( $S_{base}$ ,  $S_{guia}$ );
    Viabilize  $S_{nova}$ ;
    Se satisfeita a probabilidade de mutação
      Então
         $S_{nova} =$  Mutação( $S_{nova}$ );
      Fim-se;
    Calcule  $f(S_{nova})$ ,  $g(S_{nova})$ ,  $\delta(S_{nova})$ ;
    Atualize ( $P\alpha$ ,  $S_{nova}$ );
  Fim-Enquanto;
  Atualize o valor de  $a$  usando a fórmula (19)
  Para todo ( $S_k \in P_\alpha$ ) e ( $\delta(S_k) < a$ ) faça
    Elimine ( $S_k$ ,  $P_\alpha$ )
  Fim-Para;
Fim-Enquanto;

```

Figura 8. Pseudocódigo para o AGC.

0_2_95, que significa 324 pontos, 20 centros, restrição para a quantidade de pessoas na fila, máximo de duas pessoas na fila com probabilidade, no mínimo, de 95%.

Os problemas com 324 pontos foram resolvidos em um computador Pentium IV 3GHz com 1GB RAM, e os de 30 pontos, em uma máquina Pentium III 800MHz com 384 MB RAM. Os programas para o acionamento do CPLEX e para a relaxação lagrangeana foram codificados em C++ e o AGC, em Object Pascal. O tempo para a obtenção das soluções do CPLEX foi limitado em 2 horas (7200s). As soluções encontradas para o AGC são uma média de 50 (cinquenta) execuções. Os valores do campo “Desvio” refletem o erro relativo da solução média do AGC em relação à melhor solução primal encontrada e é calculada pela fórmula:

$$\text{Desvio} = \frac{(\text{Max}(\text{SoluçãoCplex}, \text{Lb da relaxação Lagrangeana}) - \text{Solução média do AGC})}{\text{Max}(\text{SoluçãoCplex}, \text{Lb da relaxação Lagrangeana})} \quad (20)$$

Portanto, os valores negativos do desvio indicam que a solução média do AGC foi melhor do que as demais. Os resultados do AGC foram obtidos usando-se os valores dos parâmetros mostrados na Tabela 1, ajustados após várias execuções do programa.

A Tabela 2 compara os métodos utilizados em 24 problemas em uma rede de 324 pontos. As colunas são divididas em quatro classes: o nome do problema; a solução do CPLEX; a solução da relaxação lagrangeana; e a do AGC. Na coluna referente ao CPLEX são fornecidos: a solução inteira viável e o *gap* obtidos do CPLEX, com o tempo de todos os problemas limitado em 7200s, exceto para os problemas marcados com um *, que define uma parada na execução por falta de memória. Na coluna referente à relaxação lagrangeana, são fornecidos os limitantes inferior (Lb) e superior (Ub) com o *gap* entre eles e o tempo de execução. Os resultados do AGC são apresentados com a melhor solução, o valor médio para cinquenta execuções de cada problema e o desvio, cal-

Tabela 1. Parâmetros usados no AGC.

Parâmetro	Problemas com 30 e 324 pontos
G_{\max}	1,1* soma da população de todos os pontos de demanda
d	0,1
ϵ	0,001
l	0,0001
Cruzamentos por geração	30
Probabilidade de mutação	0,20
População inicial	300
Quantidade máxima de gerações	300

Tabela 2. Comparação entre os resultados para a rede de 324 nós.

Problema	CPLEX		Relaxação Lagrangeana				AGC			
	Solução	Gap Cplex (%)	Lb	Ub	Gap Lag (%)	Tempo (s)	Melhor solução	Solução média	Tempo (s)	Desvio (%)
324_10_0_0_95	21365	0,47	21351	21460	0,51	3466	21431	21373	8,63	- 0,04
324_10_0_1_95	35050 *	0,90	35292	35360	0,19	4671	35342	35304	7,79	- 0,03
324_10_0_2_95	45232	0,36	45183	45390	0,46	1468	45347	45245	7,81	- 0,03
324_10_0_0_85	36802	1,03	36928	37180	0,68	5554	37145	37069	8,10	- 0,38
324_10_0_1_85	50037	1,25	50562	51000	0,87	1042	50880	50711	7,69	- 0,29
324_10_0_2_85	58516	2,10	59277	59740	0,78	930	59624	59437	7,62	- 0,27
324_20_0_0_95	41676	3,01	419 98	42920	2,20	2660	42577	42318	24,30	- 0,76
324_20_0_1_95	68443	3,35	70340	70720	0,54	4391	70471	70308	23,40	0,05
324_20_0_2_95	88269	2,86	89488	90780	1,44	2156	89970	89355	24,15	0,15
324_20_0_0_85	71119	4,56	73047	74360	1,80	2143	73407	73001	23,69	0,06
324_20_0_1_85	95917	6,36	99569	102000	2,44	2088	99576	98353	24,69	1,22
324_20_0_2_85	112691 *	6,03	116905	119480	2,20	2320	116639	115235	23,33	1,43
324_10_1_40_85	27307	1,45	27560	27700	0,51	5125	27675	27602	8,54	- 0,15
324_10_1_41_85	28710 *	2,30	29137	29360	0,77	5655	29324	29260	8,27	- 0,42
324_10_1_42_85	30735	0,72	30866	30950	0,27	5665	30932	30895	8,34	- 0,09
324_10_1_48_90	26360	2,13	26799	26920	0,45	5219	26883	26835	8,35	- 0,13
324_10_1_49_90	28249	0,29	28210	28330	0,46	5159	28280	28221	8,28	0,10
324_10_1_50_90	29338	1,19	29585	29680	0,32	6508	29641	29593	8,26	- 0,03
324_20_1_40_85	53198 *	4,15	54512	55400	1,63	3216	54804	54414	23,92	0,18
324_20_1_41_85	55740 *	5,38	57841	58720	1,52	4190	58009	57571	24,70	0,47
324_20_1_42_85	60783 *	1,86	60635	61900	2,09	4726	61545	61266	23,81	- 0,79
324_20_1_48_90	52617	2,33	53067	53840	1,46	3284	53300	52958	24,32	0,21
324_20_1_49_90	54097 *	4,75	55570	56660	1,96	3095	56216	55813	24,15	- 0,44
324_20_1_50_90	57974	2,45	58804	59360	0,95	4643	58941	58577	26,03	0,39

culado em (20). Para cada problema, o valor em negrito mostra a melhor solução encontrada. De modo geral, a relaxação lagrangeana apresentou melhores resultados do que o CPLEX em um tempo computacionalmente menor. O AGC forneceu melhores resultados em 23 problemas, 96% dos testes, em tempos bastante competitivos em relação às demais soluções. Em termos de valores médios, o AGC forneceu melhores resultados em 14 problemas, 58% dos testes.

A Tabela 3 compara os métodos utilizados para uma rede de 30 pontos em 55 problemas. As colunas também são divididas nas mesmas quatro classes, porém a primeira contém um item a mais, que mostra o tempo de execução do CPLEX, ainda limitado em 7200 s. Vários valores ótimos foram encontrados em todos os métodos. Em 37 casos, 67% dos problemas considerados, todos os métodos encontraram os mesmos valores para a solução. Por serem de pequeno porte, as melhores soluções e as soluções médias do AGC foram as mesmas para 73% dos problemas. Apesar de problemas pequenos, em 27 casos,

49% dos testes, o CPLEX não encontrou o valor ótimo no tempo limite estipulado. Dentre esses, em 17 casos, 31%, a relaxação lagrangeana encontrou o valor ótimo em poucos segundos. O AGC forneceu bons resultados em tempos bastante competitivos em relação às demais soluções. Para cada problema, o valor em negrito mostra a melhor solução.

5. Conclusão

Este trabalho apresentou uma comparação entre soluções para uma formulação encontrada na literatura para o Problema Probabilístico de Localização-Alocação de Máxima Cobertura. Os resultados obtidos mostram que as abordagens da relaxação lagrangeana e do AGC são competitivas para a resolução desse problema, em tempos computacionais razoáveis. Para várias instâncias de 30 nós, os valores ótimos foram encontrados. Portanto, esses resultados validam a aplicação dessas abordagens para o QM-CLAM.

Tabela 3. Comparação entre os resultados para a rede de 30 nós.

Problema	CPLEX			Relaxação Lagrangeana				AGC			
	Solução	Gap Cplex (%)	Tempo (s)	Lb	Ub	Gap Lag (%)	Tempo (s)	Melhor Solução	Solução média	Tempo (s)	Desvio %
30_2_0_0_85	3700	0,49	7200	3700	3700	0	5	3700	3700	0,31	0,00
30_3_0_0_85	5390	0	2	5390	5517	2,36	109	5390	5390	0,51	0,00
30_4_0_0_85	5470	0	1	5470	5473	0,05	31	5470	5470	0,66	0,00
30_2_0_1_85	5100	0	38	5060	5100	0,79	23	5090	5090	0,36	0,20
30_3_0_1_85	5390	0	2	5390	5431	0,77	33	5390	5390	0,48	0,00
30_4_0_1_85	5470	0	0	5470	5470	0	14	5470	5470	0,57	0,00
30_2_0_2_85	5210	0	0	5210	5279	1,33	32	5210	5210	0,38	0,00
30_3_0_2_85	5390	0	1	5390	5430	0,77	33	5390	5390	0,47	0,00
30_4_0_2_85	5470	0	0	5470	5470	0	11	5470	5470	0,57	0,00
30_2_0_0_95	2140	0	0	2140	2140	0	0	2140	2140	0,43	0,00
30_3_0_0_95	3210	0	1	3210	3210	0	1	3210	3210	0,47	0,00
30_4_0_0_95	4280	0,31	7200	4280	4280	0	0	4280	4280	0,61	0,00
30_5_0_0_95	5330	0,69	7200	5330	5350	0,38	50	5330	5323	0,80	0,13
30_6_0_0_95	5410	0	44	5410	5468	1,07	31	5410	5392	0,84	0,33
30_2_0_1_95	3520	0,48	7200	3520	3520	0	4	3520	3520	0,30	0,00
30_3_0_1_95	5270	0	390	5260	5280	0,38	170	5240	5240	0,48	0,57
30_4_0_1_95	5390	0	0	5390	5467	1,44	31	5390	5390	0,54	0,00
30_5_0_1_95	5470	0	0	5470	5470	0	11	5470	5470	0,64	0,00
30_2_0_2_95	4520	0,43	7200	4520	4520	0	1	4520	4513	0,30	0,15
30_3_0_2_95	5390	0	0	5390	5425	0,65	32	5390	5390	0,50	0,00
30_4_0_2_95	5470	0	0	5470	5470	0	14	5470	5470	0,59	0,00
30_3_1_48_90	1440	0,93	7200	1440	1440	0	0	1440	1440	0,59	0,00
30_4_1_48_90	1920	1,47	7200	1920	1920	0	0	1920	1920	0,66	0,00
30_5_1_48_90	2400	1,47	7200	2400	2400	0	2	2390	2390	0,74	0,42
30_6_1_48_90	2880	1,47	7200	2880	2880	0	1	2860	2839	0,86	1,42
30_7_1_48_90	3350	1,78	7200	3350	3360	0,30	23	3320	3297	1,05	1,58
30_8_1_48_90	3500	0	0	3500	3693	5,53	33	3500	3500	1,05	0,00
30_3_1_49_90	2160	0	2	2150	2160	0,47	20	2160	2160	0,53	0,00
30_4_1_49_90	2880	0	0	2870	2880	0,35	21	2880	2877	0,59	0,10
30_5_1_49_90	3600	0,28	7200	3590	3600	0,28	23	3590	3590	0,70	0,28
30_6_1_49_90	4320	0,28	7200	4310	4320	0,23	24	4310	4310	0,75	0,23
30_7_1_49_90	5020	0,68	7200	5030	5035	0,10	33	5010	5010	1,03	0,40
30_3_1_50_90	2820	0,81	7200	2820	2820	0	2	2820	2820	0,52	0,00
30_4_1_50_90	3760	0,81	7200	3760	3760	0	1	3760	3753	0,58	0,19
30_5_1_50_90	4700	0,81	7200	4700	4700	0	8	4700	4700	0,73	0,00
30_6_1_50_90	5390	0	1	5390	5613	4,13	45	5390	5390	0,97	0,00
30_7_1_50_90	5470	0	1	5470	5487	0,31	32	5470	5470	1,06	0,00
30_2_1_40_85	1220	0	1122	1170	1220	4,27	18	1220	1216	0,56	0,33
30_3_1_40_85	1830	1,19	7200	1780	1830	2,81	20	1830	1819	0,59	0,60
30_4_1_40_85	2440	1,19	7200	2390	2440	2,09	23	2440	2429	0,68	0,45
30_5_1_40_85	3050	1,19	7200	3000	3050	1,67	23	3020	3001	0,74	1,61
30_6_1_40_85	3600	2,88	7200	3610	3650	1,12	33	3610	3610	0,91	0,00
30_7_1_40_85	4060	0	1	4060	4195	3,34	33	4060	4060	1,03	0,00
30_8_1_40_85	4140	0	0	4140	4158	0,45	32	4140	4140	1,09	0,00
30_2_1_41_85	1780	0	0	1780	1780	0	1	1780	1780	0,44	0,00
30_3_1_41_85	2670	0,55	7200	2670	2670	0	0	2670	2662	0,51	0,30
30_4_1_41_85	3560	0,55	7200	3560	3560	0	1	3560	3551	0,58	0,25
30_5_1_41_85	4450	0,55	7200	4450	4450	0	0	4450	4430	0,69	0,45
30_6_1_41_85	5330	0,74	7200	5330	5340	0,19	34	5300	5274	1,00	1,05
30_7_1_41_85	5410	0	27	5410	5470	1,19	31	5390	5390	0,88	0,37
30_8_1_41_85	5470	0	0	5470	5470	0	16	5470	5470	0,95	0,00
30_2_1_42_85	2300	0,81	7200	2300	2300	0	1	2300	2300	0,38	0,00
30_3_1_42_85	3450	0,81	7200	3450	3450	0	0	3450	3450	0,44	0,00
30_4_1_42_85	4600	0,81	7200	4600	4600	0	1	4600	4600	0,61	0,00
30_5_1_42_85	5390	0	1	5390	5670	5,2	65	5390	5390	0,77	0,00

Referências Bibliográficas

- CHURCH, R. L.; REVELLE, C. Maximal covering location problem. **Papers of the Regional Science Association**, v. 32, p. 101-118, 1974.
- DASKIN, M. S. **Network and discrete location: models, algorithms and applications**. New York, NY: John Wiley & Sons, Inc, 1995.
- ESPEJO, L. G. A.; GALVÃO, R. D. O uso das relaxações Lagrangeana e Surrogate em problemas de programação inteira. **Pesquisa Operacional**, v. 22, n. 3, p. 387-402, 2002.
- FURTADO, J. C. **Algoritmos genéticos construtivos na otimização de problemas combinatoriais de agrupamentos**. 1998. 112 p. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1998.
- GALVÃO, R. D. Uncapacitated facility location problems: contributions. **Pesquisa Operacional**, v. 24, n. 1, p. 7-38, 2004.
- GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning**. Reading, MA: Addison-Wesley, 1989.
- HALE, T. S.; MOBERG, C. R. Location science research: a review. **Annals of Operations Research**, v. 123, n. 1-4, p. 21-35, 2003.
- HELD, M.; KARP, R. M. The Traveling Salesman Problem and Minimum Spanning Trees. **Operations Research**, v. 18, n. 6, p. 1138-1162, 1970.
- ILOG CPLEX 7.5 **Reference Manual**, 7.5 v. 610 p. Copyright by ILOG, France, 2001.
- LACERDA, E. G. M.; DE CARVALHO, A. C. P. L. F. Introdução aos algoritmos genéticos. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19, 1999, **Anais...** Rio de Janeiro: EntreLugar, v. 2, p. 51-126.
- LARSON, R. C.; ODONI, A. R. **Urban operations research**. Englewood Cliffs, N.J: Prentice Hall, 1981.
- LORENA, L. A. N. Análise espacial de redes com aplicações em sistemas de informações geográficas. **Revista Produção (on line)**, v. 3, n. 2, 2003.
- LORENA, L. A. N.; FURTADO, J. C. Constructive genetic algorithm for clustering problems. **Evolutionary Computation**, v. 9, n. 3, p. 309-327, 2001.
- LORENA, L.A.N.; SENNE, E. L. F. Local search heuristics for capacitated p-median problems. **Networks and Spatial Economics**, v. 3, n. 4, p. 407-419, 2003.
- MARIANOV, V.; SERRA, D. Probabilistic, maximal covering location-allocation models for congested systems. **Journal of Regional Science**, v. 38, n. 3, p. 401-424, 1998.
- MARIANOV, V.; SERRA, D. Hierarchical location-allocation models for congested systems. **European Journal of Operational Research**, v. 135, n. 1, p. 195-208, 2001.
- MARTELLO, S.; TOTH, P. **Knapsack Problems – Algorithms and computer implementations**. Chichester, West Sussex: John Wiley & Sons Ltd, 1990.
- MURRAY, A. L.; GERRARD, R. A. Capacitated service and regional constraints in location-allocation modeling. **Location Science**, v. 5, n. 2, p. 103-118, 1997.
- OLIVEIRA, A. C. M. **Algoritmos evolutivos híbridos com detecção de regiões promissoras em espaços de busca contínuos e discretos**. 2004. 202 p. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2004.
- OLIVEIRA, A. C. M. ; LORENA, L. A. N. Population Training Heuristics. In: EUROPEAN CONFERENCE ON EVOLUTIONARY COMPUTATION IN COMBINATORIAL OPTIMIZATION, 2005, Lausanne. **Lecture Notes in Computer Science (LNCS)**. Berlin: Springer, 2005. v. 3448. p. 166-176
- PEREIRA, M. A.; LORENA, L. A. N. A heurística Lagrangeana/Surrogate aplicada ao problema de localização de máxima cobertura. In: XXXIII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL (SBPO), 2001, Campos do Jordão. **Anais...** p. 1326-1337.
- PIRKUL, H.; SCHILLING, D. A. The maximal covering location problem with capacities on total workload. **Management Science**, v. 37, n 2, p. 233-248, 1991.
- SERRA, D.; MARIANOV, V. **New trends in public facility location modeling**. Universitat Pompeu Fabra Economics and Business Working Paper, n° 755, 2004. Disponível em <<http://www.econ.upf.edu/docs/papers/downloads/755.pdf>>. Acesso em 16 Nov 2004.

APPLICATION OF LAGRANGIAN RELAXATION AND THE CONSTRUCTIVE GENETIC ALGORITHM TO SOLVE THE PROBABILISTIC PROBLEM OF MAXIMUM LOCATION-ALLOCATION COVERAGE

Abstract

The Maximal Covering Location Problem (MCLP) seeks to locate facilities in order to maximize the serviced population, considering a given distance or standard service time. Various extensions of this model have been proposed to enhance its applicability, e.g., probabilistic models for maximum location-allocation coverage with waiting time or queue length constraints for congested systems, taking into account one or more servers per service center. In this paper we present two procedures for solving a probabilistic model, which considers one server per center, using Lagrangian relaxation and the Constructive Genetic Algorithm. Extensive tests of these approaches are presented and their results compared.

Keywords: *location problems, maximal coverage, lagrangian relaxation, constructive genetic algorithm.*