

HEURÍSTICA HÍBRIDA COM DETECÇÃO DE REGIÕES PROMISSORAS APLICADA AO PROBLEMA PROBABILÍSTICO DE LOCALIZAÇÃO-ALOCAÇÃO DE MÁXIMA COBERTURA

Francisco de Assis Corrêa

Instituto Nacional de Pesquisas Espaciais
12.201-970 São José dos Campos, SP – Brasil
fcorrea@directnet.com.br

Antonio Augusto Chaves

Instituto Nacional de Pesquisas Espaciais
12.201-970 São José dos Campos, SP – Brasil
chaves@lac.inpe.br

Luiz Antonio Nogueira Lorena

Instituto Nacional de Pesquisas Espaciais
12.201-970 São José dos Campos, SP – Brasil
lorena@lac.inpe.br

RESUMO

O problema de localização de máxima cobertura procura localizar facilidades visando 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 de uma heurística híbrida conhecida por Clustering Search (CS), que consiste na detecção de regiões promissoras de busca, baseada em agrupamentos. São apresentados os resultados dos testes computacionais para instâncias de até 818 vértices.

PALAVRAS-CHAVE. Problemas de cobertura, sistemas congestionados, *clustering search*. Otimização Combinatória.

ABSTRACT

The Maximal Covering Location Problem (MCLP) maximizes the population that has a facility within a maximum travel distance or time. Numerous extensions have been proposed to enhance its applicability, like the probabilistic model for the maximum covering location-allocation with constraint in waiting time or queue length for congested systems, with one or more servers per service center. This paper presents one solution procedure for that probabilistic model, considering one server per center, using a Hybrid Heuristic known as Clustering Search (CS), that consists in detecting promising search areas based on clustering. The computational tests report results for network instances up to 818 vertices.

KEYWORDS. Covering problems, congested systems, clustering search. Combinatorial Optimization.

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. 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, nos setores público e privado, são extensões dessa formulação. As aplicações variam de sistemas de emergência (Eaton et al, 1986; Current e O'Kelly, 1992), serviços hierárquicos de saúde (Moore e ReVelle, 1982), controle de poluição do ar (Houglund e Stephens, 1976), e sistemas congestionados (Marianov e Serra, 1998; Marianov e Serra, 2001). Os métodos de solução para o MCLP incluem relaxação de programação linear (Church e ReVelle, 1974), heurísticas gulosas (Daskin, 1995), relaxação lagrangeana (Galvão e ReVelle, 1996), heurística lagrangeana/surrogate (Pereira e Lorena, 2002), geração de colunas (Pereira, Lorena e Senne, 2007), dentre outros. Considerável revisão deste tema pode ser encontrada em Chung (1986), Hale e Moberg (2003), Serra e Marianov (2004) e Galvão (2004).

Em muitos trabalhos envolvendo problemas de localização, essa distância (ou tempo) entre pontos de demanda e a 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. Normalmente, 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 todas as demandas (Marianov e Serra, 1998 e 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, os autores definem um limite mínimo 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 autores definiram Problema de Localização-Alocação de Máxima Cobertura Probabilístico (*The Queueing Maximal Covering Location-Allocation Model*) (QM-CLAM). Considerável revisão de modelos probabilísticos pode ser encontrada em Galvão (2004) e Brotcorne, Laporte e Semmet (2003).

Corrêa e Lorena (2006) resolveram o QM-CLAM usando o Algoritmo Genético Construtivo (AGC). Este algoritmo genético trabalha com uma população formada por esquemas (solução incompleta) e estruturas (solução completa), e possui os operadores tradicionais de seleção, cruzamento e mutação. O AGC difere do algoritmo genético tradicional na maneira de avaliar os esquemas, na possibilidade de usar heurísticas para definir a função de avaliação e no tratamento de população dinâmica. Inicialmente, a população é formada apenas por esquemas, e pode ser aumentada devido aos operadores de cruzamento ou ser diminuída durante as gerações, direcionada por parâmetros de evolução. A população é

construída geração após geração, procurando-se por estruturas bem adaptadas e por bons esquemas (Lorena e Furtado, 2001; Oliveira e Lorena, 2004; Oliveira e Lorena, 2005).

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 de serviços, e apresentar a solução usando uma heurística híbrida conhecida por *Clustering Search* (CS), proposta por Oliveira e Lorena (2004, 2007). Os resultados do CS são comparados com os obtidos com o AGC e com os obtidos pela heurística proposta por Marianov e Serra (1998).

Este artigo também explora a formulação matemática do QM-CLAM para validar os resultados computacionais do CS. O *software* comercial CPLEX (ILOG, 2006) foi usado para resolver de forma aproximada a formulação desse modelo. O restante deste artigo apresenta cinco Seções. A Seção 2 descreve a formulação matemática do QM-CLAM; a Seção 3 descreve a heurística usada neste artigo; a Seção 4 apresenta a aplicação do CS ao QM-CLAM; a Seção 5 mostra os resultados computacionais e a Seção 6, as conclusões.

2. 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 $j \in N_i$, onde I é o conjunto dos pontos de demanda a serem alocados e N_i 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, $y_j = 0$, 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:

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

Subject to

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

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

$$\sum_{i \in I} f_i x_{ij} \leq \mu_j^{b+2} \sqrt{1-\phi} \quad \forall j \quad (4)$$

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

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

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

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, φ . As segundas, restrições (4a), determinam que o tempo gasto no centro j seja, no máximo, τ , com a probabilidade, no mínimo, φ . 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 ω_j atribuída a um centro é definida como uma superposição de processos de Poisson:

$$\omega_j = \sum_{i \in I} f_i x_{ij} \quad (7)$$

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 ω_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 , onde $\mu_j \geq \omega_j$, de forma a manter o equilíbrio do sistema. O lado direito das equações 4 e 4a são valores constantes, calculados para μ_j , φ , b e τ , definidos a priori, onde 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 φ é a probabilidade de que sejam encontradas, no máximo, b pessoas na fila ou de esperar, no máximo, um tempo τ para ser atendido.

O problema QM-CLAM pertence à classe dos problemas NP-Completos (Pirkul e Schilling, 1991). Aplicativos comerciais podem resolver algumas instâncias desse problema. Entretanto, o tempo de computação se torna inviável para problemas de tamanhos razoáveis. Por isso, buscaram-se soluções alternativas eficientes para formulações desse tipo. No próximo capítulo será apresentada uma abordagem baseada no *Clustering Search*.

3. Clustering Search (CS)

A metaheurística Clustering Search (CS), proposta por Oliveira e Lorena (2004, 2007), consiste num processo de agrupamentos de soluções para detectar regiões supostamente promissoras no espaço de busca. É particularmente interessante encontrar tais regiões assim que possível para mudar a estratégia de busca sobre elas. Uma região pode ser vista como um subespaço de busca definido por uma relação de vizinhança.

No CS um processo de agrupamento iterativo é executado simultaneamente com uma metaheurística, identificando grupos de soluções que merecem especial interesse. As regiões destes grupos de soluções devem ser exploradas tão logo sejam detectadas, por meio de heurísticas de busca local específicas. Espera-se uma melhoria no processo de convergência associado a uma diminuição no esforço computacional em virtude do emprego mais racional dos métodos de busca local.

O CS procura localizar regiões promissoras através do enquadramento destas por *clusters*. Um *cluster* é definido pela tripla $\mathcal{g} = \{C; r; \beta\}$ onde C , r e β são, respectivamente, o centro e o raio de uma região de busca, e uma estratégia de busca associada ao *cluster*.

O centro C é uma solução que representa o *cluster*, identificando a sua localização dentro do espaço de busca. Inicialmente, os centros dos *clusters* são obtidos aleatoriamente e, progressivamente, tendem a deslocar-se para pontos realmente promissores no espaço de busca. O raio r estabelece a distância máxima, a partir do centro, até a qual uma solução pode ser associada ao *cluster*. Por exemplo, em otimização combinatória, r pode ser definido como o número de movimentos necessários para transformar uma solução em outra. A estratégia de

busca β é uma sistemática de intensificação de busca, na qual soluções de um *cluster* interagem entre si, ao longo do processo de agrupamento, gerando novas soluções.

O CS consiste em quatro componentes conceitualmente independentes com diferentes atribuições:

- uma metaheurística (ME);
- um agrupador iterativo (AI);
- um analisador de agrupamentos (AA);
- um algoritmo de otimização local (AO).

A Figura 1 ilustra os quatro componentes, o conjunto de soluções e os centros dos *clusters* que interagem entre si.

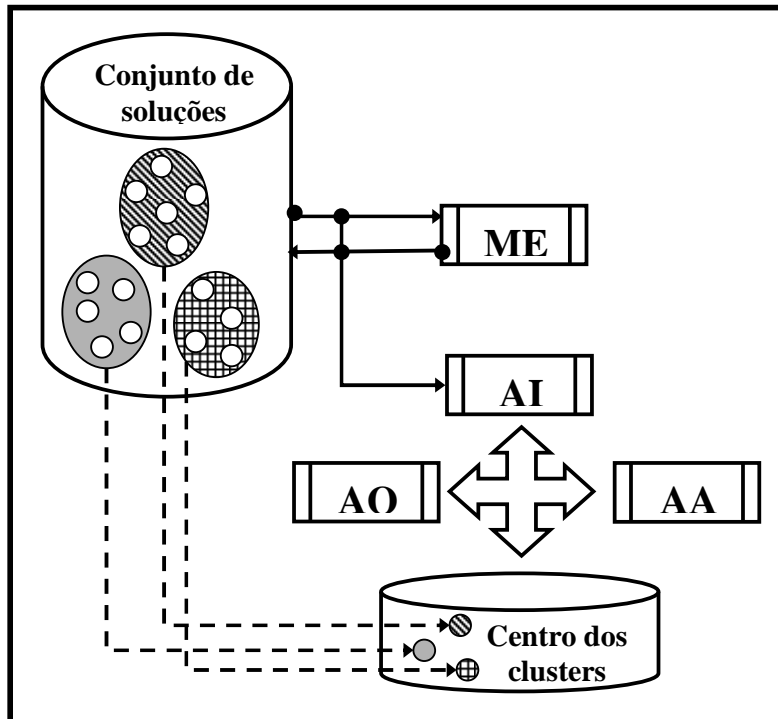


Figura 1. Diagrama Conceitual do CS

O componente ME trabalha como um gerador de soluções de tempo integral. O algoritmo é executado independentemente dos componentes restantes e precisa ser capaz de gerar soluções de forma contínua para o processo de agrupamento. Simultaneamente, *clusters* são mantidos para representar estas soluções. Este processo trabalha como um laço infinito, no qual, soluções são geradas ao longo das iterações.

O componente AI objetiva reunir soluções similares dentro de *clusters*, mantendo uma solução no centro do *cluster* que seja representativa para as demais soluções. Para evitar um esforço computacional extra, o AI é desenvolvido como um processo *online*, no qual os agrupamentos são progressivamente alimentados por soluções geradas em cada iteração do ME. Um número máximo de clusters MC é definido a priori, evitando a criação ilimitada de *clusters*. Uma métrica de distância também precisa ser definida inicialmente, permitindo uma medida de similaridade para o processo de agrupamento. Cada solução recebida pelo AI é agrupada no *cluster* que for mais similar a ela, causando uma perturbação no centro desse *cluster*. Tal perturbação é chamada de assimilação e consiste basicamente em atualizar o centro com a nova solução recebida.

O componente AA provê uma análise de cada *cluster*, em intervalos regulares, indicando um provável *cluster* promissor. A densidade do *cluster*, λ_i , é a medida que indica o

nível de atividade dentro do *cluster*. Por simplicidade, λ_i pode ser o número de soluções geradas por ME e alocadas para o *cluster* i . Sempre que λ_i atinja certo limitante, significando que algum modelo de informação foi predominantemente gerado por ME, tal *cluster* precisa ser mais bem investigado para acelerar o processo de convergência deste.

Por fim, o componente AO é um módulo de busca local que provê a exploração de uma suposta região promissora, delimitada pelo *cluster*. Este processo acontece após AA ter descoberto um *cluster* promissor e uma busca local é aplicada ao seu centro. AO pode ser considerado como a estratégia de busca β associada com o *cluster*, ou seja, uma heurística específica para o problema a ser empregada nesse *cluster*.

4. CS aplicado ao QM-CLAM

Esta Seção apresenta uma versão do CS para a resolução do QM-CLAM. Uma solução é representada por uma estrutura computacional que contém os centros de serviços localizados, a alocação dos pontos de demanda e o valor da função objetivo.

O componente ME, responsável por gerar soluções para o processo de agrupamento, foi implementado utilizando a metaheurística *Greedy Randomized Adaptive Search Procedure* – GRASP (Feo e Resende, 1995). O GRASP é basicamente composto por duas fases: uma de construção, na qual uma solução viável é gerada e uma fase de busca local, na qual a solução construída é melhorada.

A fase de construção do GRASP usa os passos 1 a 5 da heurística proposta por Marianov e Serra (1998), modificando-se a maneira de localizar um centro no passo 5. Ao invés de selecionar um vértice com a maior taxa de chamada corrente, escolhe-se um aleatoriamente com uma porcentagem (α) dentre as maiores taxas de chamada correntes.

A fase de busca local do GRASP usa a heurística de localização-alocação proposta por Lorena e Pereira (2002). As soluções obtidas na fase de construção podem ser melhoradas buscando-se um novo centro de serviços dentro de cada área de cobertura, substituindo-se o centro de serviços corrente por um vértice da mesma área. Após essa troca, o atendimento dos pontos de demanda deve ser recalculado, pois a alocação e a configuração de cobertura desses pontos podem ser modificadas.

O componente AI realiza um agrupamento iterativo de cada solução obtida pelo GRASP e trabalha como um classificador, mantendo no sistema somente informações relevantes e direcionando a busca nas áreas promissoras. Um número máximo de *clusters* é definido *a priori*. O $i^{\text{ésimo}}$ *cluster* tem o seu próprio centro C_i e um raio r que é comum aos demais *clusters*.

A métrica de distância, que calcula a similaridade entre o centro do *clusters* e a solução recebida do componente ME, é definida neste trabalho como o número de diferentes centros de serviços localizados entre a solução GRASP e a solução do centro do *cluster*. Sendo assim, quanto maior o número de centros de serviços diferentes entre duas soluções, maior será a distância entre elas e menor será a similaridade. A solução GRASP será, então, atribuída ao *cluster* com menor distância métrica.

No processo de assimilação é utilizado o método *path-relinking* (Glover, 1996), o qual realiza movimentos exploratórios na trajetória que interconecta a solução gerada pelo GRASP e a do centro do *cluster* mais similar (C_i). Assim sendo, o próprio processo de assimilação já é uma forma de busca local dentro do *cluster*, pois o centro vai ser deslocado para a melhor solução avaliada nessa trajetória, caso ela seja melhor que o centro corrente. A Figura 2 apresenta um exemplo do *path-relinking* aplicado ao QM-CLAM.

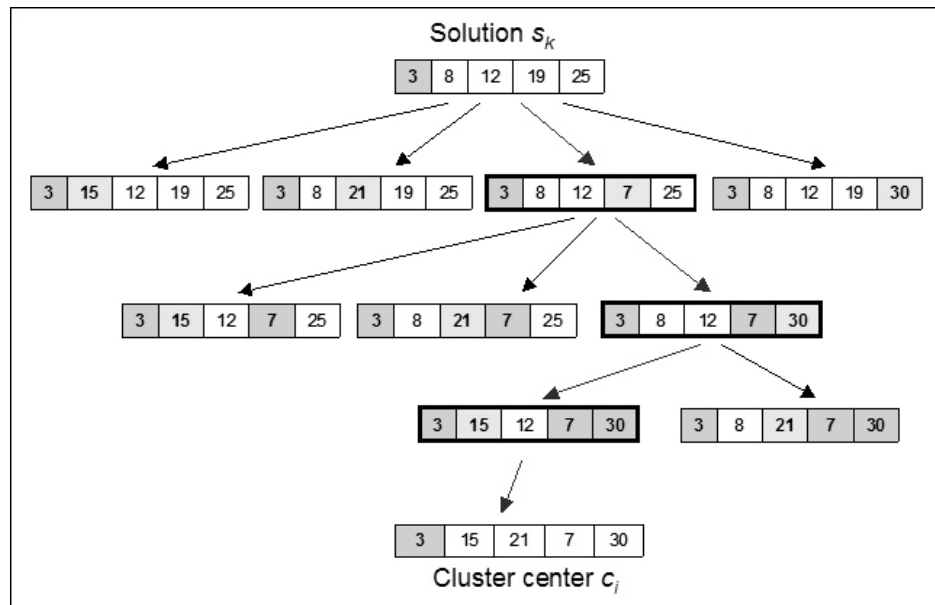


Figura 2 – Path-relinking aplicado ao CS

O componente AA é executado toda vez que uma solução for atribuída a um *cluster*. A função do AA é verificar se o *cluster* já pode ser considerado promissor. Um *cluster* torna-se promissor quando atinge certa densidade λ dada por:

$$\lambda = PD \cdot \left(\frac{NS}{|Clus|} \right) \quad (8)$$

Em que, PD é a pressão de densidade que permite controlar a sensibilidade do componente AA. Esse parâmetro indica quantas vezes a densidade deve estar acima do normal para que um *cluster* seja considerado promissor. A densidade normal é obtida se NS (número de soluções geradas em cada intervalo de análise dos *clusters*) fosse igualmente dividido pelo número de *clusters* existentes ($|Clus|$). Caso a densidade do *cluster* seja maior que esse valor significa que certo padrão de informação se tornou predominante no processo de busca, sendo o centro deste *cluster* explorado por meio do componente AO. O componente AA também tem como função executar um esfriamento de todos os *clusters* que são explorados pelo AO, ou seja, diminuir as suas densidades.

O componente AO é ativado sempre que AA descobre um *cluster* promissor. A sua implementação baseia-se também na heurística de localização-alocação proposta por Lorena e Pereira (2002), utilizada na fase de construção do GRASP, modificando-se a maneira de se selecionar um novo centro de serviços. Ao invés de trocar um centro corrente por um vértice da sua área de cobertura, troca-se o centro corrente por um vértice não alocado, escolhido aleatoriamente, modificando-se a solução de alocação e recalculando-se a cobertura. Esse processo permite buscar boas soluções em diferentes espaços de busca.

O pseudo-código do CS é apresentado na Figura 3.

```

procedimento CS
para (número de iterações não satisfeito) faça
// componente ME
     $s = \emptyset$ 
    enquanto (solução não construída) faça
        produzir Lista de Candidatos ( $C_{el}$ )
         $LCR = C_{el} * \alpha$ 
         $e = \text{selecionar elemento aleatoriamente}(LCR)$ 
         $s = s \cup \{e\}$ 
        atualizar lista de Candidatos( $C$ )
    fim enquanto
    BuscaLocal( $s$ )
    aplicar o componente AI ( $s$ )
    aplicar o componente AA (cluster ativo)
    se (cluster ativo for promissor) faça
        aplicar o componente AO
    fim para
fim procedimento CS

```

Figure 2. CS pseudo-código.

5. Resultados Computacionais

O CS foi testado em uma rede de 324 e uma de 818 vértices, obtidas de uma base de dados geográficos da cidade de São José dos Campos-SP, acrescida da população fictícia em cada ponto de demanda. Estas redes estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>. Várias instâncias foram montadas, variando-se os parâmetros p , b , μ , ϕ e τ .

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 324 vértices considerou-se: raio de cobertura igual a 250m, e para os de 818, 750m. Para as duas redes foi considerado o tempo de atendimento igual a 15 min e a taxa de chamada igual a 0,01 x população do ponto de demanda, tanto para as restrições do comprimento da fila, como para as restrições do tempo de espera.

Os nomes dos problemas foram codificados do seguinte modo: quantidade de pontos, quantidade de centros de serviços, 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_0_2_95 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%.

O CS foi codificado em Object Pascal. Os tempos nas tabelas estão em segundos e foram obtidos em um computador Pentium IV 3GHz com 1GB RAM. O tempo para a obtenção das soluções do CPLEX foi limitado em 3 horas (10800s).

A Tabela 1 compara os métodos utilizados em 24 problemas da rede de 324 vértices e a Tabela 2 compara os métodos aplicados à rede de 818 vértices. As colunas são divididas em seis classes: o nome do problema, a solução do CPLEX, a solução do AGC, a solução da heurística do Marianov e Serra, a solução do GRASP e a solução do CS. Na coluna referente ao CPLEX são fornecidos: a solução inteira viável (coluna Solução) e o *gap* obtidos do CPLEX 10.0 (coluna Gap), com o tempo de todos os problemas limitado em 10800s, exceto para os problemas marcados com um *, que define uma parada na execução por falta de memória (coluna Tempo). Para cada problema, o valor em negrito mostra a melhor solução encontrada.

Os resultados do AGC e do CS refletem cinquenta execuções de cada problema e mostram em quatro colunas: o melhor valor encontrado (coluna Melhor), o valor médio da função

objetivo (coluna Média), o tempo médio (colunas Tempo) e o desvio (coluna Desvio) que define, em porcentagem o erro relativo entre as colunas Solução e Média, calculado por $(\text{solução CPLEX} - \text{Média}) / (\text{Solução CPLEX})$. Os valores negativos de desvio mostram que a média foi melhor que a solução CPLEX.

Os resultados do GRASP refletem cinquenta execuções de cada problema e mostram em três colunas: o melhor valor encontrado (coluna Melhor), o valor médio da função objetivo (coluna Média) e o tempo médio (coluna Tempo).

Os resultados da heurística refletem as soluções de cada problema e mostram em três colunas: o valor encontrado (coluna Solução), o tempo de execução (coluna Tempo) e o desvio (coluna Desvio), calculado por $(\text{Solução CPLEX} - \text{Solução da heurística}) / (\text{Solução CPLEX})$.

Os valores para os parâmetros de desempenho da abordagem CS foram ajustados após várias execuções e também baseados em Oliveira e Lorena (2004)

- número de soluções geradas no intervalo de cada análise $NS = 15$;
- número de *clusters* $|Clus| = 5$;
- pressão de densidade $PD = 2$;

O GRASP (componente ME do CS) encontrou piores valores para a função objetivo que o CS em 75% dos problemas, considerando a rede de 324 vértices, e em cerca de 80% para a rede de 818 vértices. Isto mostra que os demais componentes do CS contribuem para melhorar os valores das soluções, e como esperado, com um custo computacional maior.

Considerando a rede de 324 vértices, o CPLEX encontrou melhores resultados em 100% dos testes. Entretanto, o CPLEX acusou erro de execução em duas instâncias, devido a falta de memória. Além disso, não encontrou o valor ótimo em três problemas em três horas de execução.

Para a rede de 818 vértices o CPLEX não encontrou o valor ótimo em 50% das instâncias testadas em três horas de execução, sendo a maioria com *gaps* acima de 200%. Para esta rede, o CS também encontrou os valores ótimos em tempos computacionais melhores.

Embora não registrados em uma tabela, os valores para a rede de 30 vértices (Marianov e Serra, 1998), o CS encontrou o valor ótimo em cerca de 65% das instâncias, considerando os valores médios, e em cerca de 81%, considerando a melhor solução, com melhor tempo médio.

6. Conclusões

Este trabalho apresentou a solução para o Problema de Localização-Alocação de Máxima Cobertura Probabilístico usando *Clustering Search* (CS). Este método foi aplicado com sucesso em alguns problemas de otimização combinatória, como problemas de seqüenciamento de padrão (Oliveira e Lorena, 2004) e caixeiro viajante com coleta de prêmios (Chaves e Lorena, 2005).

Para algumas instâncias os valores ótimos foram encontrados, o que valida os resultados do CS para o QM-CLAM. Os resultados mostram que a abordagem do CS é competitiva para a resolução deste problema, em tempos computacionais razoáveis.

A utilização de outras metaheurísticas, tais como Colônia de Formigas, Busca Tabu e Algoritmo Genético, para a geração das soluções no processo de agrupamento, e a implementação de novas heurísticas de buscas locais podem ser consideradas em trabalhos futuros para a solução do QM-CLAM. Além disto, instâncias maiores deste modelo podem ser geradas e resolvidas.

Tabela 1 - Resultados para a rede de 324 vértices

Problema	CPLEX			AGC				Heurística			GRASP			CS			
	Solução	Gap	Tempo	Melhor	Média	Tempo	Desvio	Solução	Tempo	Desvio	Melhor	Média	Tempo	Melhor	Média	Tempo	Desvio
324_10_0_0_95	21460	0	30,1	21431	21373	8.63	0.41	21386	0.14	0.34	21446	21440.1	1.12	21455	21445.6	7.89	0.067
324_10_0_1_95	35360	0	1020,6	35342	35304	7.79	0.16	35250	0.16	0.31	35339	35335.1	1.16	35355	35354.4	8.02	0.016
324_10_0_2_95	45390	0	286,1	45347	45245	7.81	0.32	45300	0.20	0.19	45337	45333.1	1.14	45366	45351.9	8.25	0.079
324_10_0_0_85	37180	0	286,0	37145	37069	8.1	0.29	37081	0.22	0.26	37157	37154.6	1.24	37168	37162.3	9.20	0.042
324_10_0_1_85	51000	0	4866,0	50880	50711	7.69	0.55	50750	0.20	0.47	50948	50945.6	1.27	50948	50945.6	8.60	0.087
324_10_0_2_85	59740	0	2506,0	59624	59437	7.62	0.49	59598	0.20	0.22	59696	59687.2	1.25	59696	59687.2	7.89	0.070
324_20_0_0_95	42902	0,042*	5454,1	42577	42318	24.3	1.37	42714	0.73	0.45	42811	42786.7	2.35	42842	42810.7	42.51	0.225
324_20_0_1_95	70720	0	3757,1	70471	70308	23.4	0.54	70368	0.74	0.46	70570	70524.3	2.45	70672	70648.1	36.16	0.059
324_20_0_2_95	90778	0,003	10800,0	89970	89355	24.15	1.52	90424	0.81	0.35	90556	90508.0	2.50	90617	90528.1	39.06	0.231
324_20_0_0_85	74321	0,053*	7348,6	73407	73001	23.69	1.77	73981	0.83	0.45	74173	74093.5	2.61	74174	74099.2	34.44	0.289
324_20_0_1_85	101962	0,037	10800,0	99576	98353	24.69	2.50	100628	1.02	0.24	101306	101127.2	2.66	101306	101127.2	31.28	-0.251
324_20_0_2_85	119440	0,034	10800,0	116639	115235	23.33	3.10	118451	0.77	0.40	118766	118562.7	2.64	118766	118563.4	29.99	0.306
324_10_1_40_85	27700	0	541,0	27675	27602	8.54	0.33	27583	0.17	0.40	27670	27664.6	1.23	27690	27686.6	10.16	0.023
324_10_1_41_85	29360	0	172,0	29324	29260	8.27	0.34	29288	0.14	0.24	29335	29328.0	1.24	29350	29348.3	9.63	0.036
324_10_1_42_85	30950	0	921,1	30932	30895	8.34	0.18	30902	0.17	0.16	30926	30920.2	1.23	30947	30940.9	9.89	0.029
324_10_1_48_90	26920	0	131,2	26883	26835	8.35	0.32	26855	0.14	0.24	26899	26895.0	1.23	26909	26906.3	9.78	0.051
324_10_1_49_90	28330	0	231,5	28280	28221	8.28	0.30	28206	0.25	0.36	28295	28282.9	1.26	28317	28311.0	9.23	-0.014
324_10_1_50_90	29680	0	94,6	29641	29593	8.26	0.28	29638	0.22	0.13	29658	29654.5	1.25	29663	29658.8	8.62	0.061
324_20_1_40_85	55398	0,004	10800,0	54804	54414	23.92	1.61	55006	0.84	0.54	55205	55133.9	2.55	55312	55241.7	33.66	0.113
324_20_1_41_85	58720	0	2194,2	58009	57571	24.7	1.94	58577	1.02	0.23	58610	58579.3	2.60	58610	58581.2	38.74	0.223
324_20_1_42_85	61900	0	6449,1	61545	61266	23.81	0.99	61637	0.88	0.39	61737	61708.0	2.57	61846	61829.8	50.19	0.081
324_20_1_48_90	53837	0,006	10800,0	53300	52958	24.32	1.58	53377	0.63	0.80	53645	53598.1	2.52	53792	53690.8	41.16	0.216
324_20_1_49_90	56655	0,009	10800,0	56216	55813	24.15	1.39	56180	1.34	0.74	56364	56239.4	2.53	56529	56477.8	36.20	0.218
324_20_1_50_90	59320	0,067	10800,0	58941	58577	26.03	1.31	59119	0.94	0.40	59263	59235.0	2.60	59263	59235.8	40.22	0.201

Tabela 2 - Resultados para a rede de 818 vértices

Problema	CPLEX			AGC				Heurística			GRASP			CS			
	Solução	Gap	Tempo	Melhor	Média	Tempo	Desvio	Solução	Tempo	Desvio	Melhor	Média	Tempo	Melhor	Média	Tempo	Desvio
818_10_0_0_95	21460	0	3016,1	21455	21449,32	43,65	0,05	21429	4,94	0,14	21459	21458,3	7,34	21460	21459,1	18,55	0,004
818_10_0_1_95	35360	0	3382,3	35356	35346,02	50,45	0,04	35339	13,05	0,06	35360	35360,0	7,70	35360	35360,0	21,93	0,000
818_10_0_2_95	45390	0	3539,3	45387	45377,4	49,37	0,03	45375	12,09	0,03	45389	45388,2	7,97	45390	45390,0	19,99	0,000
818_10_1_48_90	26920	0	3016,0	26915	26901	48,35	0,07	26907	11,38	0,05	26918	26917,9	7,11	26920	26919,3	18,33	0,003
818_10_1_49_90	28330	0	3594,6	28320	28297,98	47,78	0,11	28309	8,63	0,07	28329	28328,6	7,66	28330	28329,9	19,81	0,000
818_10_1_50_90	29680	0	3725,6	29678	29673,76	47,81	0,02	29661	18,28	0,06	29680	29679,3	7,42	29680	29680,0	16,60	0,000
818_20_0_0_95	42920	0	6024,1	42870	42817,66	76,2	0,24	42793	25,67	0,30	42905	42900,1	16,46	42920	42917,6	176,04	0,006
818_20_0_1_95	69457	677,9	10800,0	70653	70557,06	80,42	-1,58	70644	37,03	-1,71	70717	70714,8	18,74	70720	70716,6	216,32	-1,813
818_20_0_2_95	88246	512,3	10800,0	90747	90672,9	84,99	-2,75	90730	75,19	-2,81	90772	90769,4	20,99	90780	90777,6	219,06	-2,869
818_20_0_0_85	69363	678,9	10800,0	74341	74279,42	90,04	-7,09	74313	75,05	-7,14	74352	74352,0	18,12	74360	74359,4	213,70	-7,203
818_20_0_1_85	101248	433,6	10800,0	101955	101898,82	89,19	-0,64	101933	76,06	-0,68	101993	101991,6	21,17	102000	101998,2	215,08	-0,741
818_20_0_2_85	119449	352,3	10800,0	119445	119306,4	89,18	0,12	119397	105,48	0,04	119468	119465,8	23,22	119479	119474,8	258,30	-0,022
818_20_1_40_85	55400	0	7628,2	55341	55235,68	88,33	0,30	55325	66,11	0,14	55397	55395,0	17,79	55400	55398,7	206,58	0,002
818_20_1_41_85	58720	0	5982,1	58706	58677,58	87,62	0,07	58637	69,63	0,14	58712	58709,4	17,20	58720	58719,9	159,58	0,000
818_20_1_42_85	61900	0	4568,58	61839	61719,7	90,43	0,29	61814	71,81	0,14	61894	61892,8	18,03	61900	61898,0	180,37	0,003
818_20_1_48_90	53840	0	3902,0	53814	53762,18	87,87	0,14	53728	22,41	0,21	53827	53825,7	17,16	53840	53838,1	161,66	0,004
818_20_1_49_90	56660	0	9035,1	56605	56465,32	88,71	0,34	56602	34,34	0,10	56653	56651,7	17,99	56660	56659,0	194,49	0,002
818_20_1_50_90	48792	1007,3	10800,0	59336	59279,8	87,04	-21,49	59269	39,84	-21,47	59352	59350,2	16,67	59360	59358,9	136,12	-21,66
818_50_0_0_85	172692	212,9	10800,0	184428	184153,6	177,34	-6,64	185426	756,72	-7,37	185760	185753,2	55,55	185760	185753,2	2997,12	-7,563
818_50_0_1_85	248126	117,7	10800,0	253438	252884,6	171,86	-1,92	254509	730,20	-2,57	254851	254837,4	62,86	254851	254837,4	2936,70	-2,705
818_50_0_2_85	294795	83,3	10800,0	296763	296182,8	171,74	-0,47	298217	757,73	-1,16	298516	298510,6	67,99	298516	298510,6	2998,88	-1,260

818_50_1_48_90	134599	0,001	10800,0	134079	133999,4	177,67	0,45	134088	596,11	0,38	134448	134440, 0	48,64	134587	134580	2852,16 9	0,014
818_50_1_49_90	140926	283,4	10800,0	140439	140143,4	174,96	0,56	141281	571,17	-0,25	141544	141531, 4	51,09	141544	141534, 8	3069,09 1	-0,432
818_50_1_50_90	147337	266,7	10800,0	147202	147123,8	173,96	0,14	147931	667,67	-0,40	148269	148256, 0	51,15	148318	148294, 6	2889,19 1	-0,650

Referências

- Brotcorne, L., Laporte, G and Semet, F.** (2003). Ambulance Location and relocation models. *European Journal of Operational Research*, 147, 451-463.
- Chaves, A. A. and Lorena, L. A. N.** (2005). Hybrid algorithms with detection of promising areas for the prize collecting traveling salesman problem. *Fifth international conference on hybrid intelligent systems (HIS'05)*, 49-54.
- Chung, C.H.** (1986). Recent applications of the Maximal Covering Location Problem (MCLP) model. *Journal of the Operational Research Society*, 37, 735-746.
- Church, R.L. and ReVelle, C.** (1974). Maximal covering location problem. *Papers of the Regional Science Association*, 32, 101-118.
- Corrêa, F.A. and Lorena, L.A.N.** (2006). Using the Constructive Genetic Algorithm for Solving the Probabilistic Maximal Covering Location-Allocation Problem. *I Workshop on Computational Intelligence/SBRN*.
- Current, J. R. and O'Kelly, M.** (1981). Locating emergency warning sirens. *Decision Sciences*, 23, 221-234.
- Daskin, M. S.,** *Network and discrete location: models, algorithms and applications*. John Wiley & Sons, New York, 1995.
- Eaton, D., Hector, M., Sanchez, V., Latingua, R. and Morgan, J.** (1986). Determining ambulance deployment in Santo Domingo, Dominican Republic. *Journal of the Operational Research Society*, 37, 113-126.
- Feo, T. and Resende, M.** (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, 109-133.
- Galvão, R.D.** (2004). Uncapacitated facility location problems: contributions. *Pesquisa Operacional*, 24, 7-38.
- Galvão, R. D. and ReVelle, C. S.** (1996). A Lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88, 114-123.
- Glover, F.** (1996). Tabu search and adaptive memory programming: Advances, applications and challenges. *Interfaces in Computer Science and Operations Research*, 1-75.
- Hale, T.S. and Moberg, C.R.** (2003). Location science review. *Annals of Operations Research*, 123, 21-35.
- Houglund, E.S. and Stephens, N.T.** (1976). Air pollutant monitor siting by analytical techniques. *Journal of the Air Pollution Control Association*, 26, 52-53.
- ILOG CPLEX 10.0: User's Manual**, France, 2006.
- Larson, R.C. and Odoni, A. R.,** *Urban operations research*, Prentice Hall, Englewood Cliffs, N.J., 1981.
- Lorena, L.A.N. and Furtado, J.C.** (2001). Constructive genetic algorithm for clustering problems. *Evolutionary Computation*, 9(3), 309-327.
- Lorena, L.A.N. and Pereira, M.A.** (2002). A lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition, *International Journal of Industrial Engineering*, 9, 57-67.
- Marianov, V. and Serra, D.** (1998). Probabilistic maximal covering location-allocation models for congested systems. *Journal of Regional Science*, 38(3), 401-424.
- Marianov, V. and Serra, D.** (2001). Hierarchical location-allocation models for congested systems. *European Journal of Operational Research*, 135, 195-208.
- Moore, G.C. and ReVelle, C.S.** (1982). The Hierarchical Service Location Problem, *Management Science*, 28 (7), 775-780.
- Oliveira, A.C.M. and Lorena, L.A.N.** (2004). Detecting-promising areas by evolutionary clustering search. *Advances in Artificial Intelligence Series*, 385-394.
- Oliveira, A.C.M. and Lorena, L.A.N.** (2005). Population training heuristics. *Lecture Notes in Computer Science*, 3448, 166-176.
- Oliveira, A. C. M. and Lorena, L. A. N.** (2007). Hybrid Evolutionary Algorithms and Clustering Search. Editors: Crina Grosan, Ajith Abraham and Hisao Ishibuchi. *Springer SCI Series*, – to appear. (<http://www.lac.inpe.br/~lorena/alexandre/HEA-07.pdf>).
- Pirkul, H. and Schilling, D.A.** (1991). The maximal covering location problem with capacities on total workload. *Management Science*, 37(2), 233-248.
- Serra, D. and Marianov, V.** (2004). New trends in public facility location modeling. Universitat Pompeu Fabra Economics and Business. *Working Paper 755*. Available at (<http://www.econ.upf.edu/docs/papers/downloads/755.pdf>).