

Método Heurístico para Localização de Concentradores em Redes

Wesley Gomes de Almeida¹, Edson Luiz França Senne²

¹ Instituto Nacional de Pesquisas Espaciais – INPE
Caixa Postal 515 – 12.227-010 – São José dos Campos – SP – Brazil

² Universidade Estadual Paulista – UNESP Faculdade de Engenharia – Campus de Guaratinguetá Caixa Postal 205 - 12516-410 Guaratinguetá, SP, Brasil

Wesgomes2005@gmail.com, elfsenne@feg.unesp.br

Abstract. *The Uncapacitated Single Allocation Hub Location Problem (USAHLP) consists of determining in a network of n nodes, a subset of nodes where hubs must be installed and the best allocation of the other nodes (known as spokes) to these hubs. The work proposes a new approach to solve USAHLP, using a metaheuristic denominated Clustering Search.*

Resumo. *O problema de localização de concentradores não-capacitado, conhecido como USAHLP (Uncapacitated Single Allocation Hub Location Problem), consiste em determinar em uma rede de n nós, um subconjunto de nós onde devem ser instalados concentradores (conhecidos como hubs) e a melhor alocação dos demais nós da rede (conhecidos como spokes) a estes concentradores. O trabalho propõe uma nova abordagem para solucionar o USAHLP usando uma metaheurística denominada Busca por Agrupamentos (Clustering Search, CS).*

1. Introdução

Em alguns problemas definidos em redes, a comunicação entre os nós da rede não acontece de forma direta, mas por meio de nós especiais denominados concentradores (*hubs*). Isto ocorre com frequência, por exemplo, em redes de transporte e em redes de telecomunicação. Nestes casos, diz-se que a rede é do tipo *hub-and-spoke* (Aykin, 1994). Para este tipo de rede, a comunicação entre dois nós compreende as operações de coleta (do nó de origem para um concentrador de origem), transferência (do concentrador de origem para um concentrador de destino) e distribuição (do concentrador de destino para o nó de destino).

O problema de localização de concentradores em uma rede consiste em determinar o número de concentradores, a localização de cada um deles e a alocação dos demais nós da rede (conhecidos como *spokes*) aos concentradores, de forma a minimizar o custo total de operação, que podem incluir os custos variáveis de transporte e os custos fixos de localização das instalações.

Devido ao problema de localização de concentradores pertencer à classe NP-difícil, diversas propostas usando heurísticas têm sido consideradas, tais como: Busca Tabu (Klincewicz, 1991), *Simulated Annealing* (Abdnour-Helm, 1993), Redes Neurais

Artificiais (Smith *et al.*, 1996) e Algoritmos Genéticos (Abdnour-Helm, 1998; Topcuoglu *et al.*, 2005; Cunha e Silva, 2007).

O primeiro modelo matemático de Programação Inteira desenvolvido para solucionar o problema não-capacitado de localização de concentradores foi proposto por O'Kelly (1987), usando uma função-objetivo quadrática. Seja x_{ij} uma variável de decisão definida como: $x_{ij} = 1$, se o nó i está alocado ao *hub* j e $x_{ij} = 0$, caso contrário. Além disso, $x_{jj} = 1$, se o nó j é um *hub* e $x_{jj} = 0$, caso contrário. Assim, o USAHLP pode ser formulado como o seguinte problema de programação inteira 0-1:

$$f(x) = \text{Min} \sum_i \sum_j w_{ij} \sum_k \lambda c_{ik} x_{ik} + \sum_i \sum_k x_{ik} \sum_j \sum_l \alpha x_{jl} c_{kl} w_{ij} + \sum_i \sum_j w_{ji} \sum_l \delta c_{jl} x_{jl} + \sum_k f_k x_{kk} \quad (1)$$

$$\text{Sujeito a:} \quad \sum_k x_{ik} = 1, \quad \forall i \in V \quad (2)$$

$$x_{kk} - x_{ik} \geq 0, \quad \forall i, k \in V, \quad (3)$$

$$x_{ik} \in \{0,1\} \quad \forall i, k \in V. \quad (4)$$

onde: V é o conjunto dos nós de entrada em um problema, w_{ij} representa o número de unidades de fluxo entre os nós i e j , c_{ij} representa o custo de transporte por unidade de fluxo entre os nós i e j , f_k representa o custo fixo associado ao nó k , caso este nó seja um *hub*, e as variáveis λ , α e δ representam, respectivamente, os custos de coleta, transferência e distribuição.

2. Busca por Agrupamentos

O método de busca por agrupamentos (CS – *Clustering Search*), proposto por Oliveira e Lorena (2004), consiste no agrupamento de soluções (*clusters*) e na busca local dentro dos *clusters*. Um *cluster* consiste de uma tripla $G = (C, r, \beta)$, onde C é o centro, r é o raio e β uma estratégia de busca local associada ao *cluster*.

O CS é composto por 4 operadores: SM (*Search Metaheuristic*), IC (*Iterative Clustering*), AM (*Analiser Module*) e LS (*Local Search*).

O componente SM responsabiliza-se pela geração de soluções, podendo ser qualquer metaheurística. Sua execução não depende dos outros componentes, no entanto, o componente SM deve garantir que as soluções serão geradas continuamente para o processo de agrupamento. Neste trabalho, o componente SM corresponde a um Algoritmo Genético (AG) (Holland, 1975).

O componente IC tem como principal finalidade agrupar soluções similares dentro de um mesmo *cluster* e criar novos *clusters*, caso não exista um *cluster* similar a uma determinada solução. Com o objetivo de limitar o número de *clusters* a serem criados, definiu-se um limitante superior N_c para este número. Tal componente também é responsável por uma perturbação (assimilação) no centro de um *cluster* toda vez que uma nova solução é incluída neste *cluster*. Para que este componente funcione adequadamente, é necessário estabelecer uma métrica de distância entre soluções.

Em intervalos regulares, o componente AM realiza sua tarefa, que é a indicação dos *clusters* promissores. Um *cluster* torna-se promissor sempre que atingir uma densidade γ dada por:

$$\gamma = PD \frac{NS}{|C|} \quad (5)$$

onde: PD é a pressão de densidade, responsável por controlar a sensibilidade do componente AM. Esse parâmetro diz quantas vezes a densidade deve estar acima do normal para que o *cluster* seja considerado promissor. A densidade normal é obtida através da razão entre NS (número de soluções geradas em cada intervalo de análise dos *clusters*) e o número $|C|$ de *clusters* existentes até o momento.

Após a identificação de um *cluster* promissor, este é passado para o componente LS que se encarrega de fazer a busca dentro do *cluster*, objetivando encontrar a melhor solução vizinha nesta região.

3. A Abordagem Proposta

Como foi discutido na seção 2, a aplicação da busca por agrupamentos requer um algoritmo capaz de gerar continuamente soluções para um problema (o componente SM). Neste trabalho emprega-se um algoritmo genético para a geração de soluções para o problema de localização de concentradores.

No algoritmo genético utilizado, a representação dos indivíduos baseia-se na proposta de Topcuoglu *et al.* (2005) e corresponde a dois vetores de tamanho n : um para armazenar a localização dos *hubs* e outro para armazenar as alocações.

O operador de *crossover* consiste no sorteio de dois pontos de cruzamento e na troca dos materiais genéticos. Os genes que estão entre os dois pontos de corte do primeiro cromossomo, unem-se com os genes que estão antes e depois do primeiro e segundo ponto de cruzamento, no outro cromossomo. Isto é realizado tanto para o *HubArray* quanto para o *AssignArray*. Após a troca dos genes, um operador se responsabiliza em verificar se o *spoke* i , no *AssignArray*, está associado a um *hub* válido do *HubArray*, caso contrário, este *spoke* deve ser realocado ao seu *hub* mais próximo.

O método de seleção utilizado foi a Seleção por *Ranking* Linear, que ordena os indivíduos de acordo com seu grau de adaptação (*fitness*), deixando o indivíduo mais adaptado na posição 1 e o menos adaptado na posição n . A probabilidade de seleção é designada a cada indivíduo linearmente, de acordo com sua posição no *ranking* (Silva, 2004).

O operador de mutação implementado neste trabalho é composto por três fases, denominadas *shift*, *exchange* e uma busca local (*Local Search*, LS), a mesma utilizada no componente LS do CS. Os dois primeiros são uma versão estendida de parte do movimento de busca tabu usada por Abdnnour-Helm (1998), onde:

- O *shift* sorteia um *spoke* e o associa a outro *hub*; se o cromossomo possuir apenas um *hub*, essa função não se aplica;
- A função *exchange* seleciona dois *spokes* aleatoriamente e troca suas associações. Um pré-requisito para esse operador é a existência de pelo menos dois *hubs* e nós

não *hubs*. No caso de haver apenas um *hub*, ou um único *spoke*, esse método não é executado;

- O método de busca local utilizado baseia-se na exploração de vizinhanças, sendo que um movimento só é aceito se este for de melhora. Toda movimentação é feita até que um número de iterações previamente fixado seja alcançado;

As estruturas de vizinhança definidas foram o *shift* e *exchange*, citados acima, acrescidos de um movimento que troca um *spoke i* qualquer, com seu *hub j*, de forma que todos os *spokes* ligados ao nó *j* passarão a se conectar ao nó *i*, fazendo com que o *spoke i* passe a ser um *hub*, e o *hub j* torne-se um *spoke* ligado ao *hub i*.

A geração resultante da aplicação dos operadores genéticos utiliza o elitismo, mantendo os indivíduos mais adaptados a cada geração. Após esse processo, define-se uma porcentagem de corte, onde todos os cromossomos com função objetivo abaixo desse valor são eliminados. O processo de eliminação simplesmente sorteia os indivíduos com *fitness* acima do ponto de corte para sofrer *crossover* e mutação, gerando os descendentes que substituirão os indivíduos com *fitness* abaixo deste ponto.

Em relação ao componente IC, a métrica estabelecida para o USAHLP corresponde ao número de alocações diferentes para os *hubs*. Assim, pode-se medir a distância entre uma dada solução e o centro (que também corresponde a uma solução) de um *cluster*. No processo de assimilação utiliza-se o método *path-relinking* (Glover, 1996), que realiza movimentos exploratórios na trajetória que interconecta uma solução gerada pelo SM e o centro de um *cluster*.

Na Figura 1, ilustra-se o funcionamento do método *path-relinking* para o USAHLP. Neste caso, tem-se uma rede com 4 nós. Cada solução é representada por um vetor v tal que $v_i = 0$, se o nó *i* corresponde a um *spoke* e $v_i = 1$, se o nó *i* corresponde a um *hub*. Para a aplicação do método deve-se, inicialmente, gerar um conjunto de soluções vizinhas à solução inicial. Para o USAHLP, as soluções vizinhas foram obtidas trocando-se um valor de v_i da solução inicial pelo correspondente v_i da solução guia. Com isto, foram obtidas 4 novas soluções, mostradas no primeiro nível da Figura 1. O método escolhe então uma dessas novas soluções, neste caso, a de menor custo, que corresponde à solução (1, 1, 0, 1). Para esta solução escolhida, aplica-se o mesmo procedimento de troca de elementos com a solução guia, gerando novas soluções. Para este exemplo, foram geradas 3 novas soluções, mostradas no segundo nível da Figura 1. Este procedimento prossegue até que a solução guia seja encontrada.

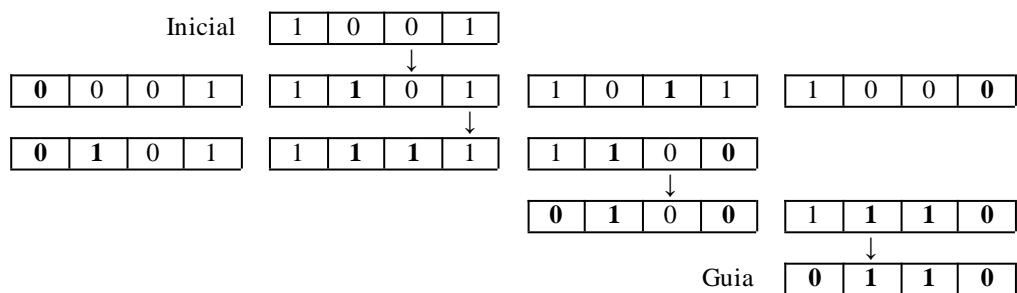


Figura 1 – Exemplo de *path-relinking* aplicado ao USAHLP

Com este método, define-se um “caminho” entre uma solução inicial e uma solução guia. A melhor solução encontrada em qualquer nível deste caminho é utilizada pelo componente IC para atualizar o centro do *cluster*.

4. Resultados Computacionais

A heurística proposta foi implementada em linguagem C. Os testes realizados executaram em um microcomputador Athlon AMD 64 2800+, com 1800Mhz e 512Mb de memória RAM, sob o sistema operacional Linux.

O conjunto de dados AP (*Australian Post*), proposto por Ernest e Krishnamoorthy (1996), derivado do fluxo de e-mails na Austrália, compõe um conjunto de testes com problemas definidos em redes de 10 a 50 nós (problemas pequenos) e de 100 a 200 nós (problemas grandes).

A Tabela 1 reporta as soluções de 7 testes realizados com o algoritmo proposto. Esta tabela contém a solução obtida (coluna **Sol**) e o tempo computacional (coluna **TC**) para 4 algoritmos: um algoritmo genético (coluna **GA**) proposto por Topcuoglu *et al.* (2005), um algoritmo híbrido que utiliza as técnicas de *Simulated Annealing* e uma lista tabu (coluna **SATL**), proposto por Chen (2007), o algoritmo genético proposto sem a utilização da busca por agrupamentos (coluna **LSGA**) e o algoritmo genético proposto com a utilização da busca por agrupamentos (coluna **CSGA**). Nesta tabela, para o método GA, a solução reportada se refere à melhor solução obtida em 100 execuções do algoritmo e incluiu-se também o número de execuções (coluna **EX**), dentre as 100, em que o algoritmo conseguiu encontrar a melhor solução. Para este algoritmo, a coluna TC corresponde ao tempo médio das 100 execuções. Para os demais algoritmos, os resultados foram obtidos em uma única execução.

Tabela 1 – Resultados para o conjunto de dados AP

N	GA			SATL		LSGA		CSGA	
	Sol	TC	EX	Sol	TC	Sol	TC	Sol	TC
10	263402,13	0,18	66	263399,95	0,24	263399,95	0,40	263399,95	0,11
20	271128,41	0,71	28	271128,18	0,50	278167,72	0,57	271128,18	0,44
25	295670,39	1,14	100	295667,84	0,23	295667,83	0,64	295667,83	0,60
40	293163,38	3,49	10	293164,83	0,70	293164,84	1,86	293164,84	0,92
50	300420,87	5,82	4	300420,98	1,29	300420,99	2,46	300420,99	1,48
100	305101,07	39,31	1	305096,76	9,79	305097,93	7,88	305097,93	2,92
200	233570,44	415,82	2	233537,33	188,54	233540,24	212,74	233540,24	85,57

Ao comparar os resultados obtidos pelos algoritmos LSGA e CSGA, é perceptível o efeito provocado pela utilização da busca por agrupamentos. Em todos os testes apresentados o método de busca por agrupamentos levou a uma solução de melhor qualidade e em todos os casos o algoritmo CSGA conseguiu obter a solução em menor tempo computacional, em relação ao algoritmo LSGA. Uma comparação perfeita dos tempos computacionais não é possível, uma vez que os algoritmos GA e SATL foram executados em um microcomputador Pentium IV, com 1.6 GHz, enquanto os

algoritmos LSGA e CSGA foram executados em um microcomputador Athlon AMD 64 2800+, com 1.8Ghz e 512Mb de memória RAM.

A Tabela 1 também mostra que os algoritmos SATL e CSGA obtêm soluções de boa qualidade, embora o SATL ainda seja ligeiramente superior em 2 dos casos testados.

4. Conclusões

O problema de localização de *hubs* é um problema de Otimização Combinatória interessante, pois ocorre em diversas situações práticas, por exemplo, no transporte de alguma entidade que precisa passar por um processo de agregação em terminais de consolidação antes de chegar ao seu destino. Boas soluções para o problema podem representar ganhos econômicos significativos para muitos setores empresariais.

A construção de algoritmos genéticos eficientes requer bons mecanismos de intensificação de busca. Sem regiões promissoras de busca, um algoritmo genético, para escapar dos ótimos locais, precisa fazer buscas locais indiscriminadas, o que aumenta o tempo computacional. A busca por agrupamentos é uma técnica de intensificação de busca que permite identificar áreas da região viável que merecem especial interesse e para as quais uma busca local deve ser intensificada.

Pelos testes realizados pode-se concluir que a busca por agrupamentos melhora a eficiência de um algoritmo genético. Uma nova proposta poderia considerar a utilização da busca por agrupamentos agregada à metaheurística SATL.

Referências

- Abdnnour-Helm, S.** (1993) *Using simulated annealing to solve the p-hub location problem*. In: IRMIS Working paper, Decision and Information Systems Department, School of Business.
- Abdnnour-Helm, S.** (1998) A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operations Research*, v. 106, p. 489-499.
- Aykin, T.** (1994) Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, v. 79, n. 33, p. 501-523.
- Chen, J.F.** (2007) A hybrid heuristic for the uncapacitated hub location problem. *Omega, The International Journal of Management Science*, v. 35, p. 211-220.
- Cunha, C.B.; Silva, M.R.** (2007) A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. *European Journal of Operational Research*, v. 179, p. 747-758.
- Ernest, A.; Krishnamoorthy, M.** (1996) Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, v. 4, n. 3, p. 139-154.
- Glover, F.** (1996) Tabu Search and adaptative memory programming: Advances, applications and challenges. *Interfaces in Computer Science and Operational Research*, Kluwer, p. 1-75.

- Holland, J.** (1975) *Adaptation in natural and artificial systems*. Ann Arbor, Michigan: University of Michigan Press.
- Klincewicz, J.** (1991) Heuristics for the p-hub median problem. *European Journal of Operations Research*, v. 79, p. 25-37.
- Oliveira, A.C.M.; Lorena, L.A.N.** (2004) Detecting promising areas by evolutionary clustering search. *Advances in Artificial Intelligence*, Springer Lecture Notes in Artificial Intelligence Series, p. 193-207.
- O'Kelly, M.** (1987) A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, v. 32, p. 393-404.
- Smith, K.; Krishnamoorthy, M.; Palaniswami, M.** (1996) Neural versus traditional approaches to the location of interacting hub facilities. *Location Science (Special Issue on Hub Location)*, v. 4, n. 3, p. 155-171.
- Silva, M.R.** (2004) *Uma contribuição ao problema de localização de terminais de consolidação no transporte de carga parcelada*. Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo.
- Topcuoglu, H.; Corut, F.; Ermis, M.; Yilmaz, G.** (2005) Solving the uncapacitated hub location problem using genetic algorithms. *Computers and Operations Research*, v. 32, p. 967-984.