

UMA APLICAÇÃO DA BUSCA POR AGRUPAMENTOS AO PROBLEMA DE LOCALIZAÇÃO DE CONCENTRADORES NÃO-CAPACITADO

Wesley Gomes de Almeida

Instituto Nacional de Pesquisas Espaciais – INPE
Av. dos Astronautas 1758, Jardim da Granja
12227-010 São José dos Campos, São Paulo, Brasil
e-mail: wesgomes2005@gmail.com

Edson Luiz França Senne

Universidade Estadual Paulista – UNESP
Faculdade de Engenharia – Campus de Guaratinguetá
Av. Ariberto Pereira da Cunha, 333, Portal das Colinas
12516-410 Guaratinguetá, SP, Brasil
e-mail: elfsenne@feg.unesp.br

RESUMO

O problema de localização de concentradores não-capacitado, conhecido como *Uncapacited Single Allocation Hub Location Problem* (USAHLP), 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. As decisões a respeito do número de concentradores, a localização de cada um deles e a alocação dos *spokes* aos *hubs* devem ser tomadas de maneira a minimizar os custos envolvidos, que correspondem a custos fixos e custos de transporte. O trabalho propõe uma nova abordagem para solucionar o USAHLP, através de uma meta-heurística denominada Busca por Agrupamentos. O trabalho apresenta resultados computacionais para os conjuntos de dados AP (*Australian Post*) e CAB (*Civil Aeronautics Board*), disponíveis na literatura, para demonstrar a efetividade do método proposto e para comparar os resultados com os obtidos por outros autores.

PALAVRAS-CHAVE. Otimização Combinatória, Problemas de Localização-Alocação, Localização de Concentradores, Busca por Agrupamentos.

ABSTRACT

The Uncapacited 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 decisions regarding the number of hubs, the location of each one and the allocation of the spokes to the hubs should be taken in order to minimize the involved costs, which correspond the fixed costs and the transport costs. The work proposes a new approach to solve USAHLP, through a metaheuristics denominated Clustering Search. The work presents computational results for the data sets AP (Australian Post) and CAB (Civil Aeronautics Board), available in the literature, to demonstrate the effectiveness of the proposed method and to compare the results with the obtained by other authors.

KEYWORDS. Combinatorial Optimization, Location Allocation Problems, Uncapacited Single Hub Location Allocation Problem, Clustering Search.

1. Introdução

Concentradores são nós especiais de uma rede, que atuam como pontos de comutação de fluxos intermodais. O problema da localização de concentradores (*hubs*) pode ser ilustrado como a seguir. Considere uma rede de n nós, onde cada nó está conectado a um dos p concentradores disponíveis. Admite-se que todos os concentradores estão interconectados. Portanto, se o nó i está conectado ao concentrador l e o nó j está conectado ao concentrador k , um fluxo de i para j será transportado, inicialmente, de i para l , depois, de l para k , e, finalmente, de k para j . Portanto, os custos de transporte em tal tipo de rede dependem criticamente de onde estão localizados os pontos de comutação. O problema da localização de concentradores consiste em determinar onde localizar os pontos de comutação de modo que o custo total de transporte seja o menor possível. Redes em que o transporte entre dois nós não acontece de forma direta, mas por meio de pontos de comutação são denominadas como do tipo *hub-and-spoke* (Aykin, 1994). Uma rede deste tipo ocorre, por exemplo, num serviço de transporte rodoviário de cargas em que o volume a ser transportado nem sempre é suficiente para lotar um veículo em uma única viagem. Por este motivo, as cargas são agregadas em terminais de consolidação e transportadas em conjunto. Neste caso, o serviço compreende as seguintes operações: coleta da carga na origem, transferência da carga de um terminal de consolidação para outro, e distribuição da carga no destino.

Diversas aplicações práticas em redes de transporte e de telecomunicações, como o transporte de pessoas, produtos, materiais ou de dados, ocorrem em redes deste tipo. A Figura 1 mostra um exemplo de rede do tipo *hub-and-spoke*, onde os concentradores são representados por quadrados e os demais nós (denominados *spokes* ou nós de demanda) são representados por círculos.

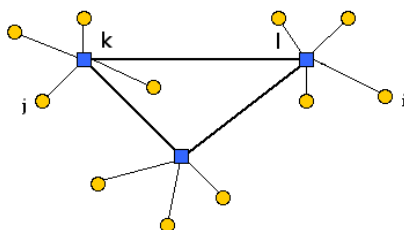


Figura 1 – Rede do tipo *hub-and-spoke*

Em muitas situações práticas, a localização dos pontos de comutação (ou dos terminais de consolidação) em redes deste tipo ocorreu a longo prazo, por meio da experiência e de acordo com a demanda dos clientes, o que nem sempre leva a uma boa solução.

Existem diferentes versões do problema de localização de concentradores. Alguns casos podem apresentar restrições de capacidade (Aykin, 1994), ou seja, um limitante no fluxo que um *hub* consegue transportar, ou ainda um custo fixo associado à localização de cada *hub*, além dos custos de alocação dos nós de demanda (*spokes*) aos *hubs*. Quando não existe restrição quanto ao fluxo que passa por um *hub* e cada *spoke* é alocado a apenas um concentrador, o problema denomina-se Problema Não-Capacitado de Localização de Concentradores (*Uncapacitated Single Allocation Hub Location Problem*, USAHLP). Neste problema, o número de concentradores é uma variável de decisão. No caso do número de concentradores ser fixo (por exemplo, igual a p), o problema é chamado de USApHMP (*Uncapacitated Single Allocation p-Hub Median Problem*) (Chen, 2008; Ebery, 2001).

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).

Este trabalho propõe uma metaheurística capaz de solucionar o USAHLP por meio da

busca por agrupamentos (*clusters*) associada a um algoritmo genético. Durante a execução das operações evolutivas de um algoritmo genético é possível observar a formação de *clusters* na região viável do problema. A idéia da busca por agrupamentos é explorar melhor estes *clusters* como regiões mais promissoras do espaço de busca.

2. Modelo Matemático

O primeiro modelo matemático de Programação Inteira desenvolvido para solucionar o problema não-capacitado de localização de p -concentradores foi proposto por O'Kelly (1987), usando uma função-objetivo quadrática. Com algumas alterações em tal modelo, pode-se obter um modelo matemático para o USAHLP, como descrito a seguir. Seja $V = \{1, 2, \dots, n\}$ o conjunto de nós de uma rede. 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, cada *hub* é alocado a si próprio, ou seja, $x_{jj} = 1$, 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:

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

(1)

sujeito a:

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

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

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

onde: 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.

Nessa formulação, a função-objetivo (1) estabelece o custo total a ser minimizado, as restrições (2) garantem que cada nó de demanda será alocado a um único concentrador, as restrições (3) asseguram que as alocações serão feitas apenas para nós que são concentradores, e as restrições (4) correspondem às condições de integridade das variáveis de decisão.

3. A Busca por Agrupamentos

O método de busca por agrupamentos (CS – *Clustering Search*), proposto por Oliveira e Lorena (2004, 2007), 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*. A princípio, o centro C de um *cluster* é gerado aleatoriamente. No entanto, após algumas iterações do algoritmo, o centro do *cluster* tende a se deslocar para regiões supostamente promissoras do espaço de busca. O raio r define a distância máxima entre o centro e qualquer solução dentro de um *cluster*. O método de busca local β responsabiliza-se pela busca de soluções melhores dentro do *cluster*.

O método CS trabalha como uma função de agrupamento iterativo e sua execução acontece simultaneamente com uma metaheurística, objetivando encontrar grupos de soluções que merecem especial atenção. Logo que estes grupos são encontrados, essas regiões do espaço de busca devem ser exploradas da melhor maneira possível, através de um método de busca local. Com a execução do método, os *clusters* formados tendem a se deslocar no espaço de busca, de maneira a convergir para regiões mais promissoras, ocasionando, ao final, a determinação de boas soluções para o problema.

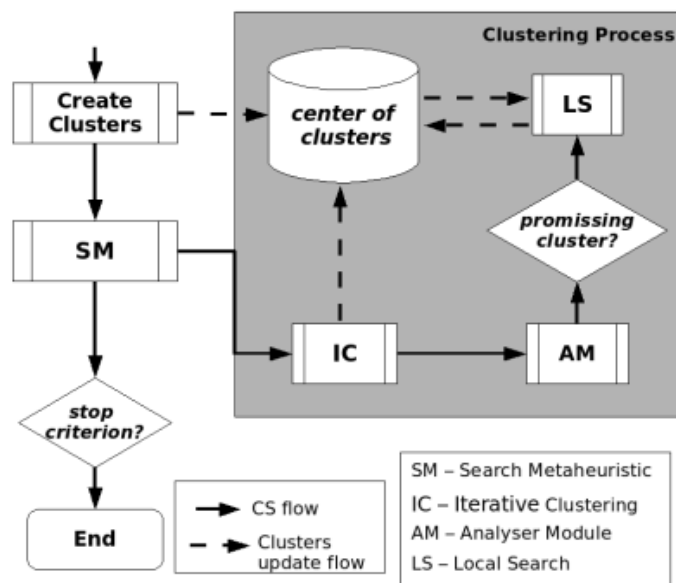


Figura 2 – Diagrama conceitual do CS (Clustering Search)

O CS é composto por 4 operadores: SM (*Search Metaheuristic*), IC (*Iterative Clustering*), AM (*Analiser Module*) e LS (*Local Search*). Na Figura 2 ilustra-se o funcionamento da metaheurística CS.

3.1. Componente SM

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. Para este trabalho, o componente corresponde a um Algoritmo Genético (AG).

Os AGs são heurísticas de busca inspiradas na teoria da evolução, capazes de encontrar boas soluções para um problema. Tal método, introduzido por Holland (1975), baseia-se em determinar em uma população de indivíduos (possíveis soluções para o problema), aqueles que, por serem mais adaptados, irão se reproduzir e gerar descendentes para novas gerações (Almeida, 2006).

Algoritmos Genéticos tradicionais podem ser descritos, resumidamente, pelos seguintes passos:

- 1) Crie uma população inicial aleatória de indivíduos (ou cromossomos) com tamanho Np .
- 2) Aplique a função de avaliação (*fitness*) a cada indivíduo, obtendo o valor da função objetivo para cada um.
- 3) Gere novos indivíduos a partir da população atual, através de operadores evolutivos como: *crossover*, mutação e elitismo.
- 4) Aplique a função de avaliação aos novos cromossomos.
- 5) Selecione os mais aptos, ou seja, os que mais melhoram a função objetivo, e substitua os indivíduos de menor aptidão na população.
- 6) Se o número de gerações alcançou um determinado limite, retorne a solução mais apta. Caso contrário, volte ao passo 3.

Para o problema considerado neste trabalho, 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 dos *spokes* aos *hubs*. Nestes vetores, denominados *HubArray* e *AssignArray*, cada posição corresponde a um nó da rede. O *HubArray* corresponde a um vetor binário em que cada posição armazena o valor 0, no caso do nó correspondente a esta posição ser um *spoke*, ou 1, no caso deste nó ser um *hub*. O *AssignArray* equivale a um vetor em que cada posição armazena o índice do *hub*, ao qual o nó

correspondente está associado.

A população inicial é gerada aleatoriamente, sendo que o número de *hubs* deve respeitar um limitante superior, como no trabalho de Chen (2007). Para calcular esse limitante, gera-se uma solução viável qualquer com apenas 1 *hub* inicialmente, calcula-se os custos fixos e variáveis separadamente, e se a soma dos custos fixos f_j for maior que a soma dos custos variáveis (coleta, transferência e distribuição) tem-se o limitante igual a 1 *hub*. Caso contrário, aumenta-se o número de *hubs* de uma unidade, fazendo com que uma nova solução seja gerada e todo o procedimento se repita até que a soma dos custos fixos seja maior que a soma dos custos variáveis, ou o número de *hubs* atingir n (número de nós da rede).

O método de cruzamento (*crossover*) escolhido para este trabalho foi o de dois pontos, pelo motivo de sua natureza de funcionamento encorajar a exploração do espaço de soluções, tornando a convergência mais promissora (Silva, 2004).

O operador de *crossover* consiste no sorteio de dois pontos de cruzamento e na troca dos materiais genéticos, como mostrado na Figura 3. 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.

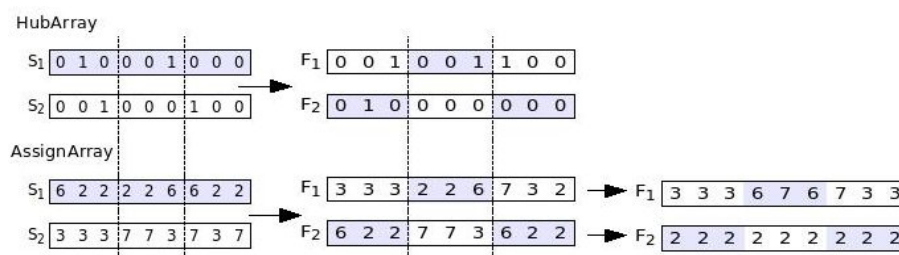


Figura 3 – Crossover de dois pontos

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, como mostra a Figura 4, 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.

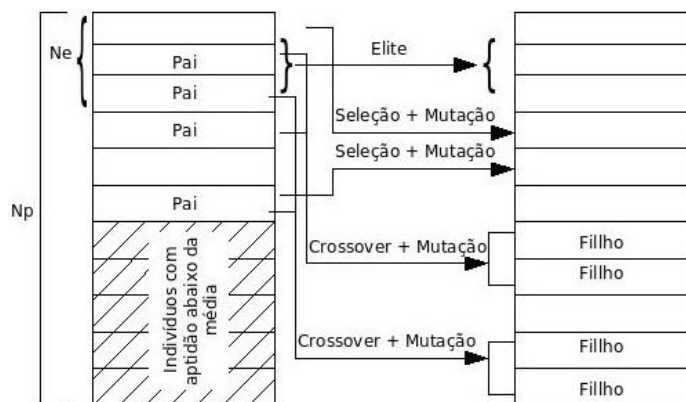


Figura 4 – Aplicações dos operadores genéticos

3.2. Componente IC

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. 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 5, 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 5. 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 5. Este procedimento prossegue até que a solução guia seja encontrada.

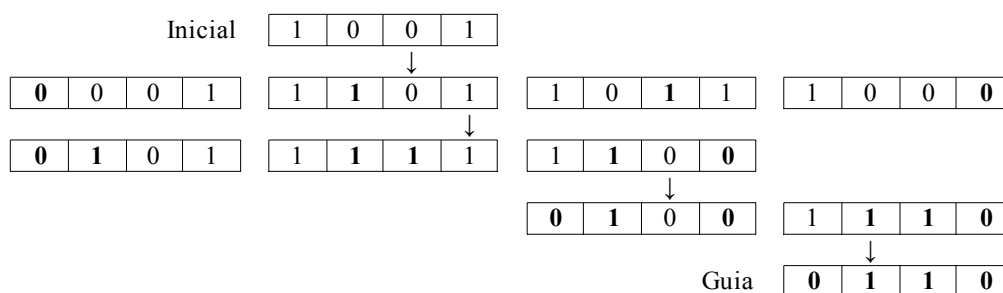


Figura 5 – Exemplo de *path-relinking* aplicado ao USAHLP

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*.

3.3. Componente AM

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.

4. Testes e Resultados

A verificação da eficiência do método heurístico implementado, baseou-se em dois conjuntos de dados: o conjunto CAB, proposto por O’Kelly (1987), e o conjunto AP (Ernest e Krishnamoorthy, 1996). Tais conjuntos de dados têm sido referenciados em diversos trabalhos, como em Sasaki e Fukushima (2003), Silva (2004), Topcuoglu *et al.* (2005), Cunha e Silva (2007) e Chen (2007).

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.

Os parâmetros do algoritmo foram definidos como: tamanho da população ($Np=50$) para problemas CAB e ($Np=100$) nos problemas AP, probabilidade de *crossover* ($Pc=0.80$), número de gerações ($Ng=200$), probabilidade de mutação ($Pm=0.10$), porcentagem de elite ($Pe=0.20$) e número de iterações da busca local ($NL=100$). Para o método *Clustering Search* utilizou-se $Nc=20$ (número de *clusters*), $PD=2.5$ (pressão de densidade).

O conjunto de dados CAB (*Civil Aeronautics Board*) são dados baseados no fluxo aéreo de passageiros entre as 25 maiores cidades dos Estados Unidos no ano de 1970. Tal conjunto tem sido muito útil na avaliação da eficiência de algoritmos em problemas de localização de concentradores (O’Kelly, 1987). Os testes realizados com o conjunto de dados CAB se agrupam em problemas de tamanho $n=\{10, 15, 20, 25\}$, sendo que para cada n o valor de α varia em $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ e os valores de λ e δ são fixos em uma unidade. Também foi definido que o custo fixo f tomaria valores no conjunto $\{100, 150, 200, 250\}$ e que o custo fixo por instalação não se alteraria de *hub* para *hub*.

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 que vão de 10 a 50 nós para problemas em pequena escala, e de 100 a 200 nós para problemas em larga escala. Os valores dos custos de coleta, transferência e distribuição são respectivamente 3.0, 0.75 e 2. Os fluxos não são simétricos, ou seja, $w_{ij} \neq w_{ji}$, além disso, um cliente (*spoke*) pode enviar um e-mail a si próprio ($w_{ii} \neq 0$).

4.1. Resultados

A seguir, nas Tabelas 1, 2 e 3, são mostrados os resultados para os diversos tipos de testes realizados. A Tabela 1 reporta os resultados de 20 testes e as Tabelas 2 e 3 reportam, cada uma, as soluções de 7 testes, resultando em um total de 34 testes diferentes. Estas tabelas contêm a solução obtida (coluna **Sol**), com “Opt.” significando que a solução ótima foi obtida, 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), e o algoritmo genético proposto neste trabalho, sem a utilização da busca por agrupamentos (coluna **LSGA**) e com a utilização da busca por agrupamentos (coluna **CSGA**). Nas Tabelas 2 e 3, 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 – CAB Data Set com n=25

α	F	Ótimo	GA		SATL		LSGA		CSGA	
			Sol	TC	Sol	TC	Sol	TC	Sol	TC
0,2	100	1029,63	Opt.	1,18	Opt.	1,79	Opt.	5,82	Opt.	2,06
	150	1217,34	Opt.	1,14	Opt.	1,10	Opt.	5,63	Opt.	1,69
	200	1367,34	Opt.	1,18	Opt.	1,10	Opt.	5,56	Opt.	1,65
	250	1500,90	Opt.	0,86	Opt.	0,70	Opt.	4,07	Opt.	1,03
0,4	100	1187,51	Opt.	0,97	Opt.	1,90	Opt.	6,00	Opt.	1,84
	150	1351,69	Opt.	1,02	Opt.	1,19	Opt.	5,87	Opt.	1,76
	200	1501,62	Opt.	1,25	Opt.	0,76	Opt.	4,15	Opt.	1,00
	250	1601,62	Opt.	0,86	Opt.	0,76	Opt.	3,92	Opt.	1,41
0,6	100	1333,56	Opt.	1,03	Opt.	1,30	Opt.	5,52	Opt.	1,79
	150	1483,56	Opt.	0,97	Opt.	1,28	Opt.	5,60	Opt.	1,66
	200	1601,20	Opt.	1,33	Opt.	0,84	Opt.	3,77	Opt.	1,02
	250	1701,20	Opt.	1,22	Opt.	0,84	Opt.	3,87	Opt.	1,07
0,8	100	1458,83	Opt.	0,92	Opt.	1,44	Opt.	5,22	Opt.	1,72
	150	1594,08	Opt.	1,14	Opt.	0,89	Opt.	3,58	Opt.	1,09
	200	1690,57	Opt.	0,74	Opt.	0,43	Opt.	1,35	Opt.	0,40
	250	1740,57	Opt.	0,70	Opt.	0,42	Opt.	1,37	Opt.	0,36
1,0	100	1556,63	1559,19	0,87	Opt.	1,52	Opt.	5,77	Opt.	1,74
	150	1640,57	Opt.	0,70	Opt.	0,24	Opt.	1,51	Opt.	0,57
	200	1690,57	Opt.	0,73	Opt.	0,24	Opt.	1,50	Opt.	0,44
	250	1740,57	Opt.	0,63	Opt.	1,24	Opt.	1,37	Opt.	0,40

Tabela 2 – AP Data Set com custos fixos “Loose”

n	GA			SATL		LSGA		CSGA	
	Sol	TC	EX	Sol	TC	Sol	TC	Sol	TC
10	224249,82	0,22	83	224250,05	0,22	224250,05	0,40	224250,05	0,12
20	234690,11	0,92	96	234690,96	0,40	234690,97	1,55	234690,97	0,55
25	236649,69	1,47	90	236650,63	0,69	236650,63	2,59	236650,63	0,65
40	240985,51	4,79	55	240986,24	2,45	241955,70	6,53	240986,23	2,50
50	237420,69	8,32	59	237421,99	4,85	237421,99	10,32	237421,99	4,05
100	238017,53	54,23	2	238015,38	25,76	238016,28	64,91	238016,28	26,50
200	228973,67	439,08	1	228944,18	179,58	248184,78	427,22	228947,08	106,98

Tabela 3 – AP Data Set com custos fixos “Tight”

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

A Tabela 1 mostra que os métodos propostos neste trabalho (algoritmos LSGA e CSGA) encontraram as soluções ótimas para todos os testes, o mesmo ocorrendo com o algoritmo SATL. A exceção ocorreu no teste em que ($n = 25$, $f = 100$, $\alpha = 1$), para o algoritmo GA.

Pelas Tabelas 2 e 3, comparando os resultados obtidos pelos algoritmos LSGA e CSGA, pode-se perceber o efeito da utilização da busca por agrupamentos. Em 3 dos 14 casos, a busca por agrupamentos levou a uma solução de melhor qualidade. Mas, em todos os casos o algoritmo CSGA conseguiu obter a solução em menor tempo computacional, em relação ao algoritmo LSGA. A Tabela 4 confirma isto, pelos tempos computacionais médios. 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.

Tabela 4 – Tempos Computacionais Médios

Tabela	GA	SATL	LSGA	CSGA
1	0.97	0.95	2.72	1.24
2	72.72	30.56	73.36	20.19
3	66.64	28.75	32.26	13.15

As Tabelas 2 e 3 também mostram que os algoritmos SATL e CSGA obtêm soluções de boa qualidade, embora o SATL ainda seja ligeiramente superior em 4 dos casos testados.

5. 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 solução do problema consiste na determinação da quantidade necessária e na localização dos concentradores, e na alocação dos demais nós de uma rede a estes concentradores, minimizando os custos envolvidos.

O método proposto neste trabalho, por ser uma heurística, não garante que a solução ótima do problema seja sempre obtida, mas, como observado pelos testes realizados, é capaz de gerar soluções de boa qualidade em um tempo computacional razoável.

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 (*clustering search*) 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

- Abdnour-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.
- Abdnour-Helm, S.** (1998) A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operations Research*, v. 106, p. 489-499.
- Almeida, W.G.** (2006) *Algoritmos genéticos para projeto de estruturas metálicas sob condições de incêndio*. Monografia, Faculdades Integradas de Caratinga, MG.
- 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.
- Chen, J.F.** (2008) A note on solution of the uncapacited single allocation p-hub median problem. *Journal of Chinese Institute of Industrial Engineers*, v. 25, p. 11-17.
- 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.
- Ebery, J.** (2001). Solving Large Single Allocation p-hub problems with two or three hubs. *European Journal of Operational Research*, v. 128, p. 447-458.
- Ernest, A.; Krishnamoorthy, M.** (1996) Efficient algorithms for the uncapacited 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.
- Oliveira, A.C.M.; Lorena, L.A.N.** (2007) *Hybrid Evolutionary Algorithms and Clustering Search*. Hybrid Evolutionary Algorithms, Studies in Computational Intelligence, Volume 75, Springer Berlin, p. 77-99.
- 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.
- Sasaki, M.; Fukushima, M.** (2003) On the hub-and-spoke model with arc capacity constraints. *Journal of the Operations Research Society of Japan*, v. 46, n. 4, p. 409-428.
- 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.