

MÉTODO HEURÍSTICO HÍBRIDO PARA RESOLUÇÃO DO PROBLEMA DE LOCALIZAÇÃO DE FACILIDADES CAPACITADAS EM DOIS NÍVEIS

Rômulo Louzada Rabello

Universidade Federal do Espírito Santo
Alegre, ES, 29500-00, Brasil
romulolouzada@gmail.com

Geraldo Regis Mauri

Universidade Federal do Espírito Santo
Alegre, ES, 29500-00, Brasil
geraldo.mauri@ufes.br

Glaydston Mattos Ribeiro

Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ, 21945-970, Brasil
glaydston@pet.coppe.ufrj.br

RESUMO

Esse trabalho propõe um método heurístico híbrido caracterizado pela utilização em conjunto da meta-heurística *Clustering Search* (CS) com um algoritmo exato para a resolução do Problema de Localização de Facilidades Capacitadas em Dois Níveis. O problema consiste em definir locais para instalação de fábricas e depósitos visando o atendimento da demanda de um conjunto de clientes. O objetivo do problema é a redução dos custos de operação (abertura de instalações e o fluxo de produtos *fábrica* → *depósito* → *cliente*) respeitando as restrições de capacidade das instalações e de demanda dos clientes. Os testes computacionais demonstram que o método proposto obteve resultados satisfatórios quando comparado com um *solver* comercial e com um Algoritmo Genético proposto em um artigo recente. Assim, o método híbrido que combina a CS com um algoritmo exato pode ser considerado uma nova alternativa para resolver o problema em questão.

PALAVRAS CHAVE. Meta-heurística. *Clustering Search*. Localização de Facilidades.

Tópicos: OC - Otimização Combinatória

ABSTRACT

This paper proposes a hybrid heuristic method which uses the metaheuristic Clustering Search (CS) combined with an exact algorithm to solve the Two-Stage Capacitated Facility Location Problem (TSCFLP). The TSCFLP consists of defining locations of factories and depots in order to supply a set of customers. The problem aims to reduce the operation cost (opening facilities and products flow: *factories* → *depots* → *customer*) ensuring the capacity constraints of facilities and the costumers' demand. The computational experiments demonstrate that our proposed method was able to present good results when comparing it with a commercial solver and a Genetic Algorithm proposed in a recent work. Thus, our hybrid method combining CS with an exact algorithm can be considered as a new alternative to solve the TSCFLP.

KEYWORDS. Metaheuristic. Clustering Search. Facility Location

Paper topics: CO - Combinatorial Optimization

1. Introdução

Problemas de localização podem ser encontrados em diversos campos de indústrias de todo o mundo. Diferentes modelos matemáticos têm sido relatados na literatura para o tratamento de casos distintos desse tipo de problema [Klose e Drexl 2005].

O Problema de Localização de Facilidades (PLF) clássico consiste em definir os locais para instalação de facilidades (depósitos, por exemplo) com o objetivo de melhor atender um conjunto de clientes (depósitos \rightarrow clientes). O Problema de Localização de Facilidades Capacitadas em Dois Níveis (*Two-Stage Capacitated Facility Location Problem* - TSCFLP) é um caso especial do PLF que considera mais um nível de instalações (fábricas e depósitos) e possui restrições de atendimento a demanda total dos clientes e a de capacidade das facilidades.

Um centro de distribuição nacional dos correios (“fábrica”) que deve satisfazer os centros regionais (“depósitos”) para atender as agências locais (clientes); ou fábricas que desejam escoar a sua produção por meio de centros de distribuição (depósitos) para atender a demanda de clientes são exemplos práticos do TSCFLP.

De acordo com [Klose 2000], o objetivo final do TSCFLP é minimizar o custo total de distribuição, respeitando as capacidades das instalações e o atendimento de toda a demanda dos clientes. O valor a ser minimizado é dado pela soma do custo de abertura de instalações (fábricas e depósitos) com o custo de transporte entre fábricas e depósitos e entre depósitos e clientes.

[Fernandes et al. 2014] propuseram um novo conjunto de problemas teste e aplicaram um Algoritmo Genético (AG) para resolver o TSCFLP. Os autores compararam os resultados obtidos pelo AG com os obtidos por um *solver* comercial. [Litvinchev e Ozuna 2012] e [Marín e Pelegrin 1999] propõem o uso de relaxação Lagrangiana para resolver o problema. [Klose 2000] usou uma Lagrangiana *cut-and-relax* como alternativa de resolução do TSCFLP, considerando que as fábricas são fixas. Uma revisão bibliográfica sobre problemas de localização de facilidades é relatada em [Klose e Drexl 2005].

Este artigo propõe um método heurístico híbrido que combina uma meta-heurística proposta recentemente, a *Clustering Search* (CS), com um algoritmo exato para resolver o TSCFLP. Para aplicação da meta-heurística CS foi considerada uma modelagem computacional não binária, diferente dos principais trabalhos reportados na literatura que utilizam uma modelagem binária. A CS é utilizada para definir quais as fábricas e os depósitos devem ser abertos, ao passo que um algoritmo exato de fluxo de redes é utilizado para definir o fluxo de produtos entre fábricas e depósitos e depósitos e clientes. Um *solver* comercial é utilizado para resolver o fluxo de produtos na rede. Usando a modelagem computacional proposta, é possível abrir ou fechar uma fábrica ou um depósito com um único movimento. A Seção 3 detalha a modelagem utilizada. Após uma busca na literatura, não foi possível encontrar relatos da aplicação dessa modelagem não binária para resolver problemas de localização de facilidades, como o TSCFLP.

Experimentos computacionais foram realizados e os resultados obtidos demonstram que o método proposto foi capaz alcançar resultados satisfatórios quando comparado com um *solver* comercial e um Algoritmo Genético (AG) proposto em um artigo recente.

O restante deste artigo apresenta uma descrição detalhada do TSCFLP na Seção 2, seguida pela modelagem computacional na Seção 3 e a descrição do método proposto na Seção 4. Os resultados computacionais são apresentados na Seção 5 e, por fim, as conclusões são apresentadas na Seção 6.

2. Descrição do Problema

O TSCFLP consiste em definir locais para instalação de fábricas e depósitos e o fluxo de produtos de fábricas para depósitos (primeiro nível) e de depósitos para clientes (segundo nível). O TSCFLP determina que os produtos transportados a partir de um conjunto de fábricas para atender às demandas dos clientes devem, obrigatoriamente, passar pelos depósitos, que faz o papel intermediário. As principais decisões a serem tomadas estão relacionadas à definição de

quais as fábricas e os depósitos deverão ser abertos a partir de um determinado conjunto de localizações disponíveis, e ao fluxo de produtos, ou seja, definir a quantidade de produtos que as fábricas abertas irão fornecer para cada depósito aberto e a quantidade que cada depósito irá fornecer para o atendimento da demanda dos clientes.

O objetivo do problema é minimizar os custos de operação, satisfazendo a demanda dos clientes e obedecendo as limitações de capacidade das instalações. Os custos de operação podem ser divididos em dois custos distintos: localização e transporte (fluxo). O custo de localização considera o custo fixo para abrir uma fábrica ou um depósito, e o custo de transporte considera o custo para transportar uma quantidade de produto a partir de uma origem até um destino (fábrica para depósito ou depósito para cliente). Cada fábrica tem uma capacidade máxima de produção e cada depósito possui uma capacidade de armazenamento. Por fim, uma demanda específica é requerida por cada cliente.

Alguns trabalhos reportados na literatura consideraram apenas a abertura de depósitos, ou seja, as fábricas são fixas. Outros artigos definem que cada cliente pode ser atendido por um único depósito. Neste trabalho, é utilizada a mesma formulação matemática considerada no trabalho de [Fernandes et al. 2014], na qual tanto fábricas quanto depósitos devem ser abertos (as fábricas não são fixas) e a demanda de um cliente pode ser atendida por mais de um depósito. Além disso, é considerado que os clientes não podem ser atendidos diretamente pelas fábricas, ou seja, as demandas dos clientes devem passar obrigatoriamente pelos depósitos.

A Figura 1 ilustra uma cadeia de suprimentos em que as fábricas (I) atendem os depósitos (J) que finalmente satisfazem a demanda do conjunto de clientes (K).

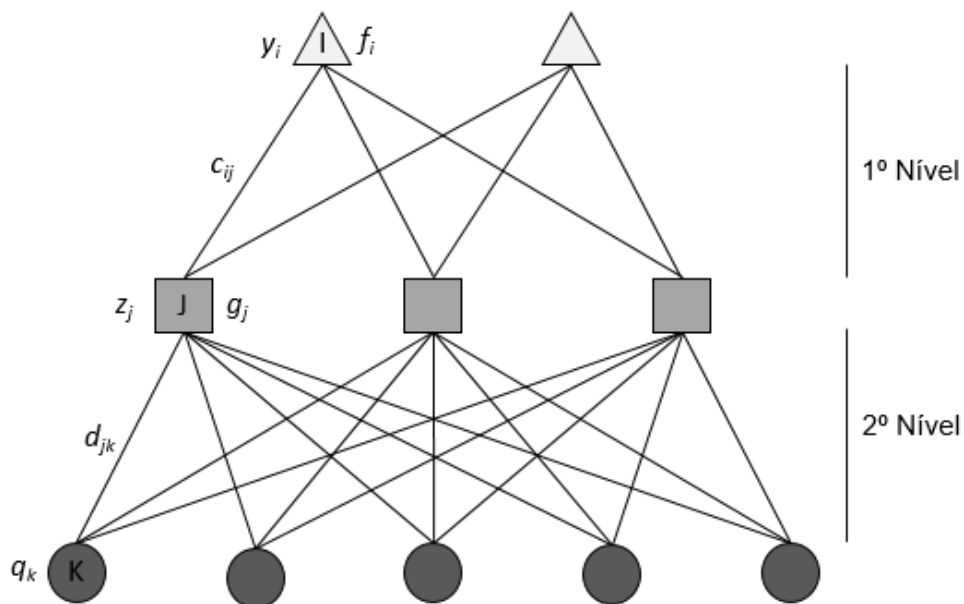


Figura 1. Rede com 2 fábricas, 3 depósitos e 5 clientes.

Na Figura 1, f_i e $g_j \forall i \in I$ e $\forall j \in J$ representam os custos fixos de abertura de fábricas e depósitos, respectivamente. b_i e p_j são as capacidades da fábrica i e do depósito j . O custo unitário de transporte no primeiro nível, ou seja, da fábrica i para o depósito j é definido por c_{ij} , e o custo unitário de transporte no segundo nível, do depósito j para o cliente k , é representado por d_{jk} . Finalmente, $q_k \forall k \in K$ define a demanda do cliente k . A modelagem matemática utilizada neste trabalho foi proposta por [Litvinchev et al. 2012] e pode ser vista a seguir:

$$\min \sum_{i \in I} f_i y_i + \sum_{j \in J} g_j z_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} d_{jk} s_{jk} \quad (1)$$

sujeito a:

$$\sum_{j \in J} s_{jk} \geq q_k \quad \forall k \in K \quad (2)$$

$$\sum_{i \in I} x_{ij} \geq \sum_{k \in K} s_{jk} \quad \forall j \in J \quad (3)$$

$$\sum_{j \in J} x_{ij} \leq b_i y_i \quad \forall i \in I \quad (4)$$

$$\sum_{k \in K} s_{jk} \leq p_j z_j \quad \forall j \in J \quad (5)$$

$$x_{ij} \in \mathbb{R}^+ \quad \forall i \in I, \forall j \in J \quad (6)$$

$$s_{jk} \in \mathbb{R}^+ \quad \forall j \in J, \forall k \in K \quad (7)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (8)$$

$$z_j \in \{0,1\} \quad \forall j \in J \quad (9)$$

As variáveis binárias y_i and z_j indicam, respectivamente, se a fábrica ou depósito está aberto ($y_i = 1$ e $z_j = 1$) ou fechado ($y_i = 0$ e $z_j = 0$) e x_{ij} e s_{jk} representam a quantidade de produto transportado da fábrica i para o depósito j e do depósito j para o cliente k , respectivamente.

A função objetivo (1) minimiza o somatório dos custos de abertura de fábricas e depósitos do custo de distribuição (fluxo). A restrição (2) assegura que as demandas de todos os clientes serão atendidas, e a restrição (3) garante que a quantidade de produtos enviados dos depósitos para os clientes não seja superior à quantidade de produtos recebida pelos depósitos. As restrições (4) e (5) asseguram que a capacidade das fábricas e depósitos não seja extrapolada e, por fim, as restrições (6) a (9) definem o domínio das variáveis de decisão.

3. Modelagem Computacional

A modelagem computacional não binária foi utilizada com o objetivo de simplificar a utilização do método proposto (descrito na Seção 4). A seguir, é detalhada como exemplo a modelagem considerando apenas para fábricas. Porém, a modelagem para os depósitos segue a mesma lógica. São considerados dois vetores de números inteiros: V1 que contém todas as fábricas selecionadas para serem abertas e V2 que possui todas as fábricas fechadas.

O número de elementos desses vetores é definido previamente e, para isso, é feita a ordenação decrescente de todas as fábricas disponíveis. A ordenação é feita utilizando a capacidade de cada fábrica e, a partir disso, uma solução viável para o problema é construída com o número mínimo de fábricas para atender a demanda total dos clientes, ou seja, utilizando as fábricas com maiores capacidades. Assim, tem-se o número mínimo de fábricas π que devem ser abertas. É possível também definir o número máximo de fábricas que devem ser abertas β da mesma forma, mas considerando as fábricas com menor capacidade.

Então, o vetor V1 (fábricas abertas) é definido com β elementos e o vetor V2 deve possuir $|I| - \beta + (\beta - \pi)$ elementos, ou seja, haverá $\beta - \pi$ posições com o valor -1, usado para indicar um local “vazio”, ou seja, uma fábrica “virtual”, que não possui custo e está inicialmente fechada. Essa modelagem evita a exploração de soluções inviáveis para o problema, pois elimina a possibilidade de criar uma solução com o um número de fábricas insuficiente para atender a demanda dos clientes.

A Figura 2 apresenta um exemplo do vetor V1 preenchido com 2 fábricas abertas e uma posição vazia, e V2 com 2 fábricas fechadas e 2 posições vazias. As posições vazias são

utilizadas para abertura ou fechamento de fábricas. Com base nessa modelagem, a partir de uma simples troca de elementos entre V1 e V2, é possível realizar três ações distintas:

1. **Figura 2 (A) – fechar uma fábrica:** seleciona-se uma posição não vazia de V1, uma posição vazia de V2 e realiza a troca dos elementos entre os dois vetores. Como pode ser visto na Figura 2 (A), após a troca o vetor V1 possui apenas uma fábrica aberta e duas posições vazias e o vetor V2 agora possui 3 fábricas fechadas e uma posição vazia.
2. **Figura 2 (B) – abrir uma fábrica:** troca-se uma posição vazia de V1 com uma posição não vazia de V2.
3. **Figura 2 (C) – abrir uma fábrica e fechar outra:** troca-se posições não vazias dos dois vetores. Na Figura 2 (C) fica claro que, após a troca, a fábrica 3 foi aberta (está em V1) e a fábrica 2 foi fechada (está em V2).

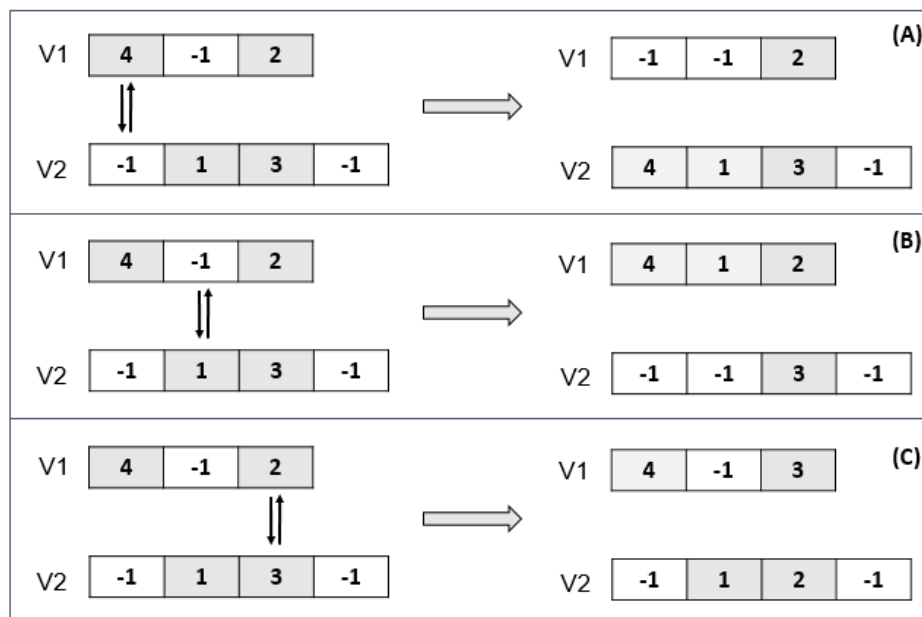


Figura 2. Exemplo da modelagem não binária proposta.

4. Método Proposto

O método proposto neste artigo combina uma meta-heurística proposta recentemente, a *Clustering Search* (CS), com um algoritmo exato para resolver o TSCFLP. Nesta seção é apresentada uma descrição detalhada sobre o método proposto.

4.1. Cálculo do custo do fluxo na rede

A modelagem computacional descrita na Seção 3 define as fábricas e os depósitos que devem ser abertos ou fechados (variáveis y_i e z_j – restrições (8) e (9), respectivamente). No entanto, o custo do fluxo de produtos (variáveis x_{ij} e s_{jk} – restrições (6) e (7)) também deve ser definido. Então, é necessário calcular o fluxo de fábricas para depósitos (x_{ij}) e de depósitos para clientes (s_{jk}). Esse fluxo deve ser definido da melhor forma possível, pois, influencia diretamente no custo final da função objetivo (1).

Para realizar o cálculo do custo do fluxo, foi usada uma estrutura de rede (um grafo), construída de modo a simplificar o processo. Todas as possíveis localizações para instalação das

fábricas e depósitos e todos os clientes são representados como nós da rede. As ligações entre todas as fábricas e depósitos e entre todos os depósitos e clientes forma os arcos da rede.

Além disso, nós artificiais (losangos) foram adicionados à rede a fim de auxiliar no cálculo do fluxo. Um nó artificial que possui arcos para todas as fábricas (triângulos) foi inserido, e cada arco saindo desse nó possui a mesma capacidade da respectiva fábrica. Para cada depósito (quadrado) também foi definido um nó artificial, com um único arco ligando o nó artificial ao respectivo depósito. A capacidade desses arcos é a mesma dos depósitos relacionados. A Figura 3 (a) mostra a rede construída em sua forma mais completa. Os losangos representam os nós artificiais adicionados.

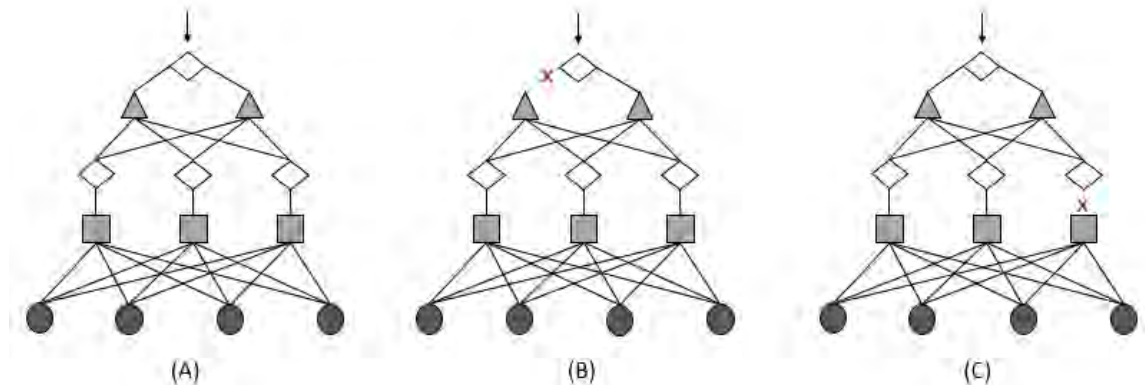


Figura 3. Exemplo de rede completa e abertura e fechamento de fábricas e depósitos.

A construção da rede é uma tarefa onerosa e sendo assim, para resolver este problema, a rede completa (Figura 3 (A) - com todos os nós e arcos) é construída apenas uma vez. Quando é necessário “remover ou adicionar” nós, isto é, abrir ou fechar fábricas ou depósitos, basta “ativar” ou “desativar” os respectivos arcos. Toda a demanda da rede (demanda total dos clientes) parte de um único ponto: o nó artificial das fábricas. Para fechar uma fábrica, a capacidade do arco que liga o nó artificial a fábrica é definida como zero, pois se o arco não tem capacidade, nenhum fluxo irá passar pelo mesmo (Figura 3 (B) – fechamento de uma fábrica). Para abrir a fábrica, basta redefinir a capacidade do arco com o valor da capacidade da fábrica. O mesmo processo é repetido para os depósitos (Figura 3 (C)).

A rede é construída no *solver* comercial CPLEX, que calcula o fluxo de menor custo. O CPLEX considera a mesma rede durante toda a execução do método proposto e, cada vez que ocorre uma alteração na estrutura da rede, isto é, abertura ou fechamento de uma fábrica ou depósito, basta atualizar a capacidade do arco relacionado e recalcular o fluxo. O valor do fluxo da rede de uma solução s será denotado a seguir por $\psi(s)$.

4.2. Clustering Search

A meta-heurística *Clustering Search* (CS) foi proposta por [Oliveira e Lorena 2007] e faz uso de agrupamento de soluções para identificar regiões promissoras no espaço de busca [Rabello et al. 2014]. A CS agrupa soluções em *clusters*, sendo que cada *cluster* pode ser definido com uma tupla (c_i, v_i, r_i) , na qual, c_i (centro) é a solução armazenada no centro do *cluster* i , v_i (volume) representa o número de soluções inseridas no *cluster* i e r_i (índice de ineficácia) é o número de vezes que o *cluster* i foi tido como promissor mas não apresentou melhores após uma exploração mais intensa.

Uma meta-heurística deve ser utilizada para gerar soluções para a CS. Assim, cada nova solução é analisada e inserida no *cluster* mais similar. Um passo-a-passo de uma iteração da CS é apresentado a seguir:

1. Uma meta-heurística fornece uma nova solução para a CS.

2. A solução é assimilada ao *cluster* i mais semelhante, de acordo com uma métrica de distância. O volume do *cluster* i é incrementado em uma unidade.
3. Se o volume do *cluster* i atingir um limite λ significa que este *cluster* é promissor. Logo, o passo 4 é executado. Caso contrário, o algoritmo começa uma nova iteração (passo 1).
4. O índice de ineficácia r_i é calculado, e apenas uma das etapas a seguir é executada:
 - a. $r_i < r_{max}$: uma busca local é aplicada na solução centro do *cluster* i . Se a busca local não melhora o valor da solução, então o índice de ineficácia r_i é incrementado.
 - b. $r_i = r_{max}$: uma perturbação é aplicada na solução centro do *cluster* i e o índice de ineficácia r_i é atualizado para 0, evitando aplicar a busca local em áreas suficientemente exploradas.

A meta-heurística *Simulated Annealing* (SA) [Kirkpatrick et al., 1983] foi escolhida para atuar como o gerador de soluções para a CS. Essa escolha se deu devido ao bom desempenho obtido em outros problemas de otimização. A Figura 4 apresenta o pseudocódigo da CS proposta para resolver o TSCFLP.

```

1. Criar  $\gamma$  clusters iniciais (centro dos clusters)
2.  $r_i \leftarrow 0$  e  $v_i \leftarrow 0 \forall i = 1 \dots \gamma$ 
3.  $s \leftarrow$  solução inicial       $s^* \leftarrow s$        $T \leftarrow T_0$ 
4. enquanto  $T > T_f$  faça
5.    $iter \leftarrow 0$ 
6.   enquanto  $iter < SA_{max}$  faca
7.      $iter \leftarrow iter + 1$ 
8.      $s' \leftarrow N(s)$  |  $N(s)$  baseada em fábricas ou depósitos
9.     se  $\delta(s') < \delta(s)$  então
10.       $s \leftarrow s'$ 
11.      se  $\delta(s') < \delta(s^*)$   $s^* \leftarrow s'$  fim-se
12.     senão
13.       Com probabilidade  $e^{-(\delta(s') - \delta(s))/T}$   $s \leftarrow s'$ 
14.     fim-se
15.   fim-enquanto
16.    $T \leftarrow \alpha T$     $i \leftarrow \underset{i \in \{1, \dots, \gamma\}}{\text{argmin}} \{H_i\}$     $v_i \leftarrow v_i + 1$     $c_i \leftarrow \text{melhor}(s, c_i)$ 
17.   se  $v_i = \lambda$  então
18.      $v_i \leftarrow 0$     $\text{busca-local}(c_i)$     $s^* \leftarrow \text{melhor}(s^*, c_i)$ 
19.     se  $c_i$  não melhorou então
20.        $r_i \leftarrow r_i + 1$ 
21.       se  $r_i = r_{max}$  então  $r_i \leftarrow 0$     $c_i \leftarrow N(c_i)$  fim-se
22.     senao
23.        $r_i \leftarrow 0$ 
24.     fim-se
25.   fim-se
26. fim-enquanto
27. retornar  $s^*$ 

```

Figura 4. Pseudocódigo da CS proposta.

No início da Figura 4, são criados os *clusters* iniciais. Nesse momento, todos os γ *clusters* não possuem solução armazenada em seu centro c_i (linha 1). Logo, o volume v_i e o índice de ineficácia r_i possuem o valor 0 (linha 2). Para preenchimento desses *clusters*, as γ primeiras soluções enviadas pela meta-heurística geradora para a CS serão assimiladas cada uma a um *cluster* ‘vazio’, ou seja, a primeira solução é inserida no primeiro *cluster*, e assim por diante, até

que a solução γ seja inserida no cluster γ . A partir de então, para assimilar uma solução a um *cluster*, o processo descrito na linha 16 é executado.

A solução inicial s (linha 3 da Figura 4) é criada de forma gulosa. Inicialmente, é calculado o custo/benefício de cada fábrica e depósito. O custo/benefício de fábricas é dado pela seguinte fórmula: $\frac{f_i + \sum_{j \in J} c_{ij}}{b_i}$ e o de depósitos: $\frac{g_j + \sum_{k \in K} d_{jk}}{p_j}$. Para compor a solução s são selecionadas as fábricas com maior custo/benefício até que essas instalações sejam capazes de atender toda a demanda dos clientes. O mesmo processo é realizado para os depósitos.

O esquema apresentado na Figura 2 é utilizado para gerar uma solução vizinha s' de uma solução s ($s' \in N(s)$ - linha 8 da Figura 4), ou seja, uma solução vizinha pode ser gerada abrindo e/ou fechando fábricas ou depósitos, e esta escolha é feita de forma aleatória. A chance de escolher se a solução vizinha será gerada a partir de mudanças nas fábricas ou nos depósitos é determinada pelo número de localizações disponíveis para instalação de fábricas ou depósitos, ou seja, se o número de depósitos for o dobro do número de fábricas, a chance de gerar uma solução vizinha mudando um depósito será o dobro da chance de mudar uma fábrica.

A solução corrente s fornecida pelo gerador de solução (SA) é recebida pela CS e imediatamente é atribuída ao *cluster* i mais similar. A distância de Hamming H_i [Hamming, 1950] é a métrica utilizada para identificar o *cluster* i mais próximo de s . Neste trabalho, H_i é calculada utilizando as fábricas e os depósitos abertos. Assim, a solução s é atribuída ao *cluster* i mais semelhante, ou seja, que minimiza a função $\arg \min_{i \in \{1, \dots, \gamma\}} \{H_i\}$.

Depois de definir o *cluster* i em que s será atribuída, a melhor solução entre s e c_i (solução do centro do *cluster* i) é definida como a nova solução do centro do *cluster* (c_i). Esse processo de substituição do *cluster* pode ser chamado de “assimilação direta”.

A função objetivo (1) de uma solução s , $\alpha(s)$, é calculada em duas etapas:

1. $\xi(s)$: a soma dos custos de abertura das fábricas e depósitos abertos na solução $s \rightarrow$ calculado pela meta-heurística CS com base na modelagem descrita na Seção 3.
 $\xi(s) = \sum_{i \in I} f_i y_i + \sum_{j \in J} g_j z_j$.
2. $\psi(s)$: o custo do fluxo na rede da solução $s \rightarrow$ calculado pelo CPLEX considerando as fábricas e depósitos abertos. $\psi(s) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} d_{jk} s_{jk}$.

Assim, o valor $\alpha(s)$ usado na Figura 4 representa o valor da função objetivo (1), que é dado pela soma dos custos de abertura e do fluxo na rede, ou seja, $\alpha(s) = \xi(s) + \psi(s)$.

Por fim, quando o volume v_i do *cluster* i atinge um limite λ , o mesmo se torna promissor, então uma busca local deve ser aplicada com o intuito de intensificar a busca por soluções melhores na vizinhança desse *cluster*. A busca local implementada neste trabalho tenta explorar todos os vizinhos da solução presente no centro do *cluster* i . Individualmente, cada fábrica aberta é trocada com todas as fábricas fechadas, sempre observando se há melhoria. Ao final, a troca que obteve a melhor melhora é aceita. O mesmo processo é realizado para os depósitos. Se a busca local for aplicada r_{max} vezes sem obter melhora, uma perturbação é aplicada no centro do *cluster* i . Neste trabalho, foi utilizada a geração de uma solução vizinha como perturbação, ou seja, $c_i = N(c_i)$.

5. Resultados Computacionais

Um conjunto de 50 instâncias apresentado em um artigo recente da literatura foi utilizado para verificar o desempenho do método proposto. Essas instâncias foram propostas por [Fernandes et al. 2014] e estão distribuídas em cinco classes. As instâncias são compostas por 50 ou 100 fábricas, 100 ou 200 depósitos e 200 ou 400 clientes. As instâncias podem ser obtidas por meio do seguinte endereço: <https://www.gerad.ca/~aloise/publications.html>.

Os parâmetros da CS foram definidos utilizando duas instâncias diferentes, escolhidas de forma aleatória: uma com 50 fábricas, 100 depósitos e 200 clientes, e outra com 100 fábricas, 200 depósitos e 400 clientes. Para definir os principais parâmetros da CS, foi seguida a metodologia utilizada por [Rabello et al. 2014]. Cada parâmetro recebe valores individualmente, enquanto os demais permanecem inalterados. O algoritmo é executado 5 vezes para cada configuração. A configuração de parâmetros que obteve o melhor resultado médio e que foi escolhida para execução final do método é apresentada na Tabela 1.

Tabela 1. Valores dos parâmetros para a CS.

Parâmetro	Descrição	Valor
γ	Número máximo de clusters	5
λ	Volume máximo do cluster	7
r_{max}	Limite para aplicação da perturbação	3
T_0	Temperatura inicial	1000
T_f	Temperatura de congelamento	0.01
α	Taxa de resfriamento	0.995
SA_{max}	Número máximo de iterações	500

Tabela 2. CS x CPLEX e CS x GA: 50 fábricas, 100 depósitos e 200 clientes.

Instância		CS x CPLEX				CS x GA				
		CPLEX		CS		GA		CS		
Classe	Id	Melhor	Tempo (s)	Melhor	Tempo (s)	Média	Tempo (s)	Melhor	Média	Tempo (s)
1	1	722178	5,8	722178	69,92	722178,0	581,41	722178	723466,0	69,92
	2	732194	46,06	732194	140,93	733350,4	564,71	732194	732463,1	140,93
	3	733473	246,45	733473	108,82	733664,2	578,28	733473	733959,0	108,82
	4	725147	58,65	730134	175,91	727325,0	533,44	730134	731123,1	175,91
	5	719431	133,39	719431	62,16	719512,6	552,90	719431	719626,0	62,16
2	1	492747	6,28	492747	44,54	492747,0	317,05	492747	492854,9	44,54
	2	494203	22,31	494203	95,77	494205,0	316,55	494203	494771,1	95,77
	3	495089	142	495089	35,55	496435,4	330,70	495089	495113,0	35,55
	4	492107	17,3	492107	115,23	492215,8	312,35	492107	492155,8	115,23
	5	489625	11,61	489625	75,96	489711,0	276,85	489625	490160,0	75,96
3	1	2689629	7323,36*	2688951	43,61	2688951,0	276,45	2688951	2688951,0	43,61
	2	2698204	9461,61*	2697537	57,57	2697803,8	285,95	2697537	2697537,0	57,57
	3	2679964	10012,95*	2678748	64,78	2679038,0	271,31	2678748	2678820,5	64,78
	4	2693384	16160,17*	2692470	77,92	2692662,0	236,55	2692470	2692470,0	77,92
	5	2646182	14523,96	2646182	37,27	2646182,0	242,34	2646182	2646182,0	37,27
4	1	541803	146,13	541803	115,50	541803,0	303,57	541803	543554,1	115,50
	2	539178	190,42	539718	107,58	539178,0	307,04	539718	540088,9	107,58
	3	544684	485,45	545624	148,90	546738,4	318,14	545624	545624,0	148,90
	4	541849	7531,17	541849	156,22	542750,0	279,54	541849	542483,0	156,22
	5	537782	129,89	537782	89,83	537806,4	264,10	537782	538640,2	89,83
5	1	2775499	517,72	2775948	193,13	2776346,8	361,04	2775948	2776171,2	193,13
	2	2781496	509,9	2781496	160,30	2781496,0	344,12	2781496	2782236,4	160,30
	3	2767634	13420,7	2767634	235,62	2767842,2	420,73	2767634	2767739,3	235,62
	4	2777307	1004,31	2777307	234,59	2777619,0	300,55	2777307	2778714,7	234,59
	5	2735567	520,35	2735567	62,26	2736077,8	318,55	2735567	2736014,2	62,26
MÉDIA		1441854,2	3305,12	1441991,8	108,39	1442145,5	355,77	1441991,8	1442436,7	108,39

* CPLEX foi abortado por falta de memória.

Utilizando os valores de parâmetros apresentados na Tabela 1, a CS foi aplicada 10 vezes para cada uma das 50 instâncias. A CS foi codificada em C++ e executada em um PC com um processador Intel Core i5 de 2.67GHz com 4 GB de memória RAM e sistema operacional Windows 7. O CPLEX 12.6 foi utilizado para resolver o algoritmo de fluxo de rede. O CPLEX e AG proposto por [Fernandes et al. 2014] também foram codificados em C++, mas os autores executaram o código em um PC com processador Pentium Intel de 2,3 GHz, com 24 GB de memória RAM.

As Tabelas 2 e 3 apresentam os resultados para as instâncias com 50 fábricas, 100 depósitos e 200 clientes e para as instâncias com 100 fábricas, 200 depósitos e 400 clientes, respectivamente.

Tabela 3. CS x CPLEX e CS x GA: 100 fábricas, 200 depósitos e 400 clientes.

Instância		CS x CPLEX				CS x GA				
		CPLEX		CS		GA		CS		
Classe	Id	Melhor	Tempo (s)	Melhor	Tempo (s)	Média	Tempo (s)	Melhor	Média	Tempo (s)
1	1	1477398	10019,98	1477524	400,20	1484057,4	2784,60	1477524	1478541,4	400,20
	2	1464441	916,43	1468081	655,95	1477503,4	2745,02	1468081	1472693,0	655,95
	3	1494399	7357,76	1502536	523,17	1497213,0	3001,83	1502536	1505733,3	523,17
	4	1462309	5175,84	1462309	740,68	1466182,2	2823,39	1462309	1463518,4	740,68
	5	1492462	967,47	1492528	409,80	1500689,2	2863,24	1492528	1497064,1	409,80
2	1	973482	9620,04	973482	884,59	979526,6	1483,25	973482	974733,4	884,59
	2	968617	3558,1	968617	611,62	973034,4	1455,77	968617	970394,6	611,62
	3	976887	2731,95	976887	236,58	989362,0	1427,72	976887	977405,7	236,58
	4	975770	28975,46	975770	715,60	978502,0	1444,07	975770	976757,1	715,60
	5	947219	121366,18	947219	427,82	952067,8	1419,02	947219	947966,9	427,82
3	1	5299973	89547,24*	5297628	1049,04	5298518,4	1355,38	5297628	5298229,0	1049,04
	2	5279599	91032,79*	5276750	1217,64	5278225,4	1320,83	5276750	5277035,3	1217,64
	3	5227517	63668,25*	5227141	989,07	5227517,0	1311,86	5227141	5227160,6	989,07
	4	5320811	58087,83*	5315009	1469,17	5316646,0	1365,90	5315009	5315837,9	1469,17
	5	5251871	69316,46*	5249852	1404,04	5251934,4	1383,04	5249852	5250452,4	1404,04
4	1	1058360	32763,62	1058471	807,73	1060799,0	1269,01	1058471	1060135,0	807,73
	2	1050553	54782,24	1050643	462,41	1053288,6	1230,08	1050643	1052238,5	462,41
	3	1057271	99944,5	1057271	734,64	1070421,0	1283,26	1057271	1059068,7	734,64
	4	1052324	128018,98	1052367	749,02	1054638,0	1301,32	1052367	1053482,7	749,02
	5	1059397	3739,27	1059423	381,18	1060621,0	1334,96	1059423	1061564,2	381,18
5	1	5506970	153249,98	5507257	873,59	5512662,0	1551,16	5507257	5507807,6	873,59
	2	5482791	70784,97	5482791	336,05	5487604,2	1499,94	5482791	5484411,4	336,05
	3	5446763	133319,11*	5452374	478,28	5458935,8	1477,01	5452374	5453488,3	478,28
	4	5517165	185827,19*	5517346	436,70	5523513,8	1513,78	5517346	5519033,4	436,70
	5	5463544	172795,94*	5464041	1080,57	5468022,8	1472,05	5464041	5464541,8	1080,57
MÉDIA		2852315,7	63902,70	2852532,7	723,01	2856859,4	1684,70	2852532,7	2853971,8	723,01

* CPLEX foi abortado por falta de memória.

Na primeira metade das tabelas é feita uma comparação entre a CS e o CPLEX (resultados relatados por [Fernandes et al. 2014]), e na segunda metade a CS é comparada ao AG proposto por [Fernandes et al. 2014]. As colunas *Melhor* e *Média* indicar a melhor solução e a solução média encontrada pelos métodos e o tempo computacional médio é relatado na coluna *Tempo*. Os melhores resultados são destacados em negrito.

Na Tabela 2, observa-se que a CS e o CPLEX apresentaram resultados semelhantes, sendo que cada método encontrou quatro melhores soluções. No entanto, a CS apresentou

melhores resultados para as instâncias da classe 3, em que o *solver* foi abortado devido à falta de memória do computador. O tempo médio de execução da CS em relação ao tempo do CPLEX foi reduzido em 96,7%. Comparando a CS com o AG, pode-se observar que as soluções médias obtidas pela CS foram piores às soluções médias obtidas pelo AG em alguns casos. A CS encontrou a solução média melhor que o AG em 11 das 25 instâncias. No entanto, quando é comparada a melhor solução obtida pela CS, o método proposto encontra soluções melhores para 24 das 25 instâncias com uma redução do tempo computacional de 69,5%. [Fernandes et al. 2014] não apresentam as melhores soluções encontradas pelo AG.

Os resultados apresentados na Tabela 3 mostram novamente um comportamento semelhante entre a CS e o CPLEX. Para essas instâncias, o tempo médio de execução da CS foi 98,9% menor que o do CPLEX. Confrontando a CS e o AG em relação à solução média, a CS encontrou resultados melhores que o AG para 23 das 25 instâncias. Já considerando as melhores soluções encontradas, a CS supera o AG em 24 das 25 instâncias, com tempo de execução 57,1% menor.

6. Conclusão

Neste trabalho foi apresentada a aplicação de um método heurístico híbrido que combina a meta-heurística *Clustering Search* (CS) com um algoritmo exato para resolver um Problema de Localização de Facilidades Capacitadas em Dois Níveis (TSCFLP). A CS foi eficiente para definir os locais de abertura de fábricas e depósitos, enquanto o fluxo foi resolvido de forma exata por um *solver* comercial. O método proposto foi capaz de encontrar boas soluções com baixo tempo computacional para um amplo conjunto de instâncias. Na comparação com um trabalho recente encontrado na literatura, o método mostrou-se eficiente obtendo resultados satisfatórios. Portanto, a aplicação do método heurístico híbrido proposto demonstrou ser uma boa alternativa para resolução do TSCFLP.

Agradecimentos

Os autores agradecem o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (processos 303052/2013-9 e 454569/2014-9) e da Fundação de Amparo à Pesquisa e Inovação do Espírito Santo - FAPES (processos 67627153/2014 e 73290475/2015).

Referências

- Fernandes, D. R. M., Rocha, C., Aloise, D., Ribeiro, G. M. e Santos, E. M.; Silva, A. (2014). A simple and effective genetic algorithm for the two-stage capacitated facility location problem. *Computers & Industrial Engineering*, 75, 200-208.
- Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, 26, 147-160.
- Marín, A. e Pelegrin, B. (1999). Applying lagrangian relaxation to the resolution of two-stage location problems. *Annals of Operations Research*, 86, 179-198.
- Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.
- Klose, A. (2000). A lagrangian relax-and-cut approach for the two-stage capacitated facility location problem. *European Journal of Operational Research*, 126, 185-198.

Klose, A. e Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*, 162, 4-29.

Litvinchev, I. e Ozuna, E. L. (2012). Lagrangian bounds and a heuristic for the two-stage capacitated facility location problem. *International Journal of Energy Optimization and Engineering*, 1, 59-71.

Oliveira, A. C. M. e Lorena. L. A. N. (2007). Hybrid evolutionary algorithms and clustering search. In Grosan. C.; Abraham, A.; Ishibuchi, H. (Eds.). *Hybrid evolutionary algorithms – Studies in computational intelligence*, 77-99.

Rabello, R. L., Mauri, G. R., Ribeiro, G. M. e Lorena, L. A. N. (2014). A clustering search metaheuristic for the point-feature cartographic label placement problem. *European Journal of Operational Research*, 234, 802-808.