Problema da Localização de Múltiplas Instalações no Contexto de Cadeias de Suprimento

Edson B. de Lima[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Belo Horizonte – MG – Brazil

1. Introdução

No contexto de logistica e cadeias de suprimento, uma das maiores preocupações do processo produtivo consiste em determinar o lugar mais eficiente para construir instalações de armazéns com a finalidade de reduzir os gastos com transporte de materiais, custos envolvidos com influência de competidores locais, armazenamento e distribuição de mercadorias, assim como a demanda dos produtos por clientes em potencial prioridade. Dados de [Ilos 2014] mostram que em 2010 empresas brasileiras gastaram cerca 8,5% de sua receita com custos logísticos. Desse percentual, 4,6% correspondem a custo com tranporte, 1,9% a custos com armazenagem e 2,0% referentes a custo com estoque. Já em 2014 (Figura 1), o valor dos gastos com custos envolvendo logística aumentou para 8,7%.

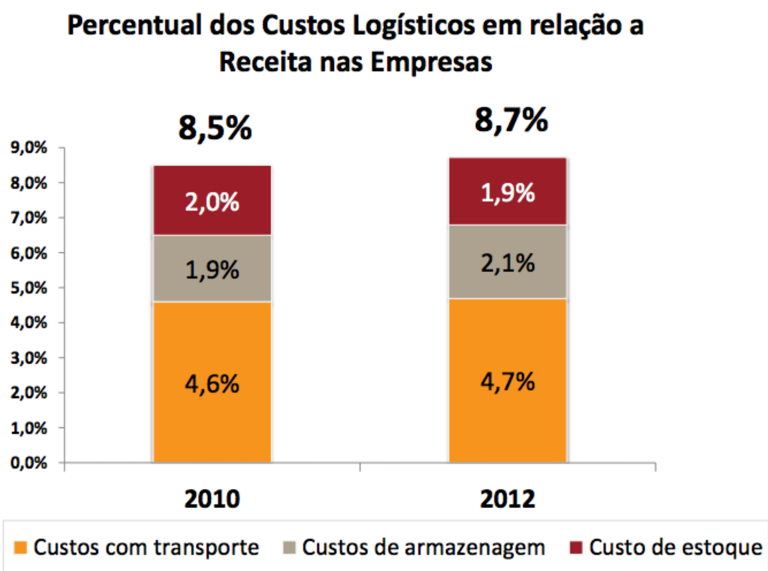

Figura 1. Custos de empresas com logística em 2010 e 2014

O problema da localização de instalações, também conhecido como Facility Location Problem (FLP) consiste em localizar certos pontos de instalações de forma otimizada com o intuito de atender um conjunto de clientes, em que as localizações e os requisitos são conhecidos [Levin 2005].

O FLP tem fundamento no problema de Fermat-Weber de localização, o qual descreve o processo de encontrar um ponto no plano que minimiza a soma dos custos de transporte de um ponto p a qualquer ponto de destino n [Brimberg 1995]. Contudo, o problema da localização das instalações envolve também outros fatores como o volume dos produtos associados, a demanda de produtos e outras relações econômicas entre instalações. Decisões sobre a localização envolvem a determinação do número, do local e da capacidade das instalações fixas que compõem os pontos de uma rede na cadeia de suprimentos onde os produtos trafegam a caminho do consumidor [Rocha 2013]. Assim, a melhor forma de localizar uma instalação de um armazem depende de uma série de fatores relacionados com uma rede de pontos estratégicos considerando todos os custos logísticos.

2. Descrição do problema

A localização de instalação pode acontecer para uma única instalação SFLP (Simple Facility Location Problem) ou ocorrer com vários pontos de suprimento MFLP (Múltiple Facility Location Problem). No SFLP a localização depende de fatores como centro de gravidade, volume dos produtos na cadeia e custo com transporte, sendo preciso calcular a distância da instalação aos pontos de suprimento próximos que dependem da instalação. Já na localização de múltiplas instalações (MFLP), algumas restrições são impostas, como o fato que as instalações não podem ser tratadas de forma independente em termos econômicos e há um grande número de configurações viáveis [Rocha 2013]. Segundo [Levin 2005], um exemplo de MFLP mais simples acontece quando o número de pontos de demanda (clientes) n e o número de instalações m são conhecidos, sendo necessário:

$a)$ Determinar a localização de m instalações

$b)$ Atribuir cada cliente a uma instalação

Para uma instalação ser ativada, deve-se anteriomente localizar um número não determinado de instalações para minimizar a soma de custos fixos e variáveis com alterações que servem a demanda de mercado. [Verter 2011].

3. Objetivos gerais

Dada a complexidade em encontrar a localização desses pontos, o presente trabalho descreve como obter a localização de múltiplas instalacões de forma otimizada. Nesse contexto, algumas perguntas devem ser respondidas, a saber:

– Quantos armazéns a rede da cadeia de suprimeiros deve ter?

– Qual deve ser sua capacidade e em que ponto devem localizar-se?

– De quais pontos de demanda cada armazém deve ficar encarregado?

– Que armazéns devem ser atribuídos a cada fábrica ou fornecedor?

4. Modelagem

Um método simples para determinar a localização de uma instalação consiste em calcular o ponto central dado um conjunto de pontos mais próximos. Esse método também chamado de Nearest Center Reclassification Algorithm (NCRA), propoe uma abordagem mais vaga, onde cada conjunto de clientes representa um cluster distinto e o ponto central pode se referir a um centro de massa, ou média desse conjunto de pontos [Levin 2005].

4.1. Modelagem em grafo

Seja o grafo $G = (V, E)$, formado por um conjunto de vértices V e um conjunto de arestas E. O conjunto V de ser particionado em $V = \{V_1, ..., V_n\}$ e o conjunto de arestas

$E = \{E_1, ..., E_n\}$, de tal forma que cada V_i representa um conjunto de vértices (pontos) do cluster com E_i consistindo no conjunto de arestas (v, c) que unem cada $v \in V_i$ ao ponto central c do cluster (centroid). Construimos o grafo $G' = (V, E')$, com $E' = E \cup (v, c_i)$, $v \in V$ e c_i como sendo o ponto central de cada cluster $G_i = (V_i, E_i)$. O problema consiste em determinar qual a menor configuração de G tal que:

$\forall G_i \subset G(d(v_i, c_k) < d(v_i, c_i) \rightarrow v_i \notin G_i$ e $v_i \in G_k)$, com $c_i \in G_i$ e $c_k \in G_k$, $i \neq k$.

Na localização de uma instalação o processo segue uma abordagem simples de encontrar o centroide de um conjunto de pontos que representam os clientes distriuídos em um espaço. Porém, quando o problema consiste em localizar mais de uma instalação considerando vários clusters cada um deles com vários clientes, diversas nuances podem acontecer, como a localização de um determinado cliente estar mais próxima do centroide outros clusters. Nessa ocasião diversas reatribuição de pontos podem acontecer com o intuito de minimizar o custo relativo a cada cliente.

De maneira formal, seja um conjunto de pontos $\mathcal{A} = \{a_1, a_2, ..., a_n\}$, e um inteiro positivo m, particionamos o conjunto $\mathcal{A}$ em m clusters não vazios $\{\Omega_1$, ..., $\Omega_m\}$, tal que cada cluster Ω_i consiste de todos os pontos mais próximos do seu centro que do centro de outros clusters Ω_k, $k \neq i$. O NCRA tem abordagem iterativa e a cada iteração o algoritmo calcula aproximação ao ponto central do cluster. A k-ésima iteração começa com uma partição Ω^k = $\{\Omega_1^k$, ..., $\Omega_n^k\}$. A partição inicial Ω^0 é escolhida aleatoriamente.

O centro x_i^k de cada Ω_i^k é computado e os pontos $a_j \in \Omega_i^k$ são reatribuídos para outros clusters se eles estão mais próximos de outros centros que do ponto x_i. O algoritmo deve parar as iterações quando nenhuma reatribuição mais for possível ou procede com nova partição Ω^{k+1} = $\{\Omega_1^{k+1}, ..., \Omega_n^{k+1}\}$ de acordo com novas atribuições. As iterações gerais são descritas pelo algoritmo (4.1).

Algorithm 1 Nearest Center Reclassification Algorithm. Iteração k

```
r ← 0
for i := 1 to m do
  calcule o centro x_i^k de Ω_i^k
end for
for j := 1 to n do
  calcule as distâncias d_ji := ||a_j − x_i^k||, i = 1, ..., m.
end for
for j := 1 to n do

  if a_j ∈ Ω_p^k and d_jl = mind_ji < d_jp, i = 1, ...m. then
    Ω_l^k := Ω_l^k ∪ {a_j}, Ω_p^k := Ω_l^k − {a_j}  (reatribui a_j)
    r ← r + 1
  end if
end for
if r = 0 then
  pare a iteração
end if
Ω^{k+1} := Ω^k
k ← k + 1
go to step one
```

Percebe-se que o algoritmo acima pode deixar um cluster vazio no caso de perder todos os clientes para as instalações vizinhas que tem ponto central mais próximo. A partição Ω^{k+1} dessa forma pode ter menos que m clusters não vazios.

5. O Método Newton Bracketing (NB)

Considerando que o objetivo de cada método consiste em minimizar o custo total em transporte, a localização ótima das instalações deve ser tal que

$$C(x) = min \sum_{j=1}^{n} w_j ||\mathrm{x} - \mathrm{a_j}|| \tag{1}$$

onde $\mathrm{x} = (x^1, x^2, ..., x^d)$ representa um vetor com as localizações das instalações e w_j o peso de transporte entre a instalacão e o cliente a_j.

O método de Weiszfeld é considerado o método padrão e o mais conhecido para resolver o problema da minimização de custos de tranporte entre pontos. A convergência do método de Weiszfeld é encontrada em [Katz 1974]. Alternativas para melhorar a otimização de (1) foram propostas por [Levin 2002b], como uma melhoria em aproximação da solução ótima. O Método NB é baseado no método Direcional de Newton [Levin 2002a], utilizando o método de Newton-Raphson como fundamento minimizar o custo através de processo iterativo. O método NB consiste em encontrar melhorias em limites sobre o valor mínimo de (1), ao invés de aproximar uma solução x*, satisfazendo o critério de otimalidade.

$$\nabla C(\mathrm{x}^*) = \sum_{j=1}^{n} w_j \frac{\mathrm{x}^* - \mathrm{a_j}}{||\mathrm{x}^* - \mathrm{a_j}||} = 0. \tag{2}$$

Uma iteração começa com um intervalo $[L^k, U^k]$, chamado $bracket$, contendo o valor mínimo C_{min}

$$L^k \leq C_{min} \leq U^k \tag{3}$$

Se o backet for pequeno o suficiente tal que

$$L^k - U^k < \varepsilon \tag{4}$$

para uma tolerância aceitável $\varepsilon > 0$ então o valor de x na iteração k é considerado ótimo, e o processo termina. Caso contrário, seleciona-se um valor M^k dentro do bracket, isto é

$$M^k := \alpha U^k + (1 - \alpha) L^k, 0 < \alpha < 1 \tag{5}$$

e resolva a iteração com a Direcional de Newton, sendo

$$C(\mathrm{x}) := M^k \tag{6}$$

para obter o valor de x^{k+1}. O Algoritmo 2 abaixo mostra o processo NB para a iteração k

Algorithm 2 O Método NB. Iteração k

Require: Dados uma iteração k e um bracket $[L^k, U^k]$
  if $L^k - U^k < \varepsilon$ **then**
    `solução := x, pare`
  end if
  $M^k := \alpha U^k + (1 - \alpha) L^k, 0 < \alpha < 1$
  $x^{k+1} := x^k - \frac{C(x^k) - M^k}{||\nabla C(x^k)||^2} \nabla C(x^k)$
  if $C(x^{k+1}) < C(x^k)$ **then**
    $U^{k+1} := C(x^{k+1}), L^{k+1} := L^k$
  else
    $L^{k+1} := M^k, U^{k+1} := U^k, x^{k+1} := x^k$
  end if
  $k := k + 1$
  `return`

Um bracket inicial é dado por:

$U^0 = C(x^0)$ onde x^0 representa a iteração inicial, e

$L^0 = ||a_i - a_j|| \min\{w_i, w_j\}$ para dois pontos quaisquer em $\mathcal{A}$

5.1. Observação

Algumas características devem ser analisadas quanto ao funcionamentos do método Newton Bracketing. Caso o ponto da instalação gerado aleatoriamente no início de cada iteração seja igual em coordenadas a um ponto que representa um cliente, então o vetor $||\mathrm{x} - \mathrm{a_j}||$ é nulo, e consequentemente o gradiente $\nabla C(x)$ é inexistente. De acordo com [Levin 2005] a alternativa consiste em substituir o vetor gradiente inexistente pelo vetor nulo **0** no subgradiente $\delta||\mathrm{x} - \mathrm{a_j}||$. Não há necessidade de recalcular $\nabla C(x)$ se $C(x^{k+1}) > C(x^k)$, e nesse caso somente o limite inferior L^k é atualizado. Portanto, o algoritmo apresenta uma redução considerável de computação, evidenciado pelo fato que na maioria das iterações apenas esse limite é atualizado.

6. Método híbrido NCRA-NB

(Heurística a ser modelada)

7. Análise de casos e testes

A análise de casos mostra que a execução dos testes no algoritmo NCRA em geral converge rápido para a aproximação do valor ótimo de x. A tabela abaixo mostra os resultados encontrados para a execução de quatro testes, com $m = 4$ instalações, no intervalo $[0, 10] \in \mathbb{R}^2$ com valores medidos:

Tabela 1. Resultados de teste. Tempo de CPU (em segundos) e número de reatrinuiçoes com n pontos em $\mathbb{R}^2$

n	seg	reatribuições
10	0,005	11
100	0.017	261
1000	0.078	3318
10000	2.149	38990
100000	161.941	390989

Os experimentos realizados com o método NB foram considerados com os mesmos parâmetros presentes no algoritmo NCRA, porém no primeiro cenário o intervalo analisado consiste em $[0, 10]$ presentes nos eixos, enquanto nos demais testes esse valor foi configurado para $[0, 20]$ tanto no eixo x quanto para o eixo y.

Tabela 2. My caption

N	Seg	Valor de x	Iterações
10	0.025	(6.051251553739743, 7.190340105512302)	10
100	0.073	(3.5666330428717608, 13.039199184190625)	44
1000	0.335	(7.113902751599934, 11.869092280908138)	60
10000	0.442	(19.32566705844289, 15.594319589920232)	79
100000	1.289	(14.27680129326317, 1.7992146231956152)	68

Referências

Brimberg, J. (1995). *The Fermat-Weber location problem revisited*.

Ilos (2014). Custos logísticos no brasil.

Katz, I. (1974). *Local convergence in Fermat's Problem*.

Levin, Y. (2002a). *Directional Newton method in n variables*.

Levin, Y. (2002b). *The Newton Bracketing method for convex minimization*.

Levin, Y. (2005). *A Heuristic Method for Large–scale Multifacility Location Problems*.

Rocha, M. (2013). Decisões de localização de instalações.

Verter, V. (2011). *Uncapacitated and Capacitated Facility Location Problems*.