UM ALGORITMO PARA O PROBLEMA DE LOCALIZAÇÃO NÃO CAPACITADO BASEADO EM TESTES DE REDUÇÃO E HEURÍSTICAS *ADD/DROP*

André Barros Pereira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS–GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Cláudio Thomas Bornstein, Dr.Rer.Nat.

Prof. Roberto Diéguez Galvão, Ph.D.

Prof. Manoel Bezerra Campêlo Neto, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
DEZEMBRO DE 2002

PEREIRA, ANDRÉ BARROS

Um algoritmo para o Problema de Localização Não Capacitado baseado em testes de redução e heurísticas *ADD/DROP* [Rio de Janeiro] 2002

X, 108 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2002)

Tese – Universidade Federal do Rio de Janeiro, COPPE

1 - Problema de Localização Não Capacitado

2 - Testes de redução

3 - Heurísticas ADD/DROP

I. COPPE/UFRJ II. Título (série)

Agradecimentos

A Deus.

Aos meus pais Amarílio e Mirtes e aos meus irmãos Adriana e Júnior, pelo amor, carinho, incentivo e compreensão sobretudo nesta época em que estive ausente.

À minha namorada Daniele Coelho que, apesar da distância, sempre se fez muito presente.

Ao professor Cláudio Bornstein, pela atenção, paciência e dedicação com as quais conduziu a orientação deste trabalho. Ao professor Manoel Campêlo pelas sugestões e revisões no texto da dissertação e ao professor Roberto Galvão pelas sugestões e por todo o material cedido que foi de grande valia para a realização deste trabalho.

A todos os meus professores de graduação, especialmente aos professores Marcos Negreiros e Wamberto Vasconcelos e aos professores do mestrado, em especial aos professores Nelson Maculan e Adilson Xavier.

Aos amigos Cláudio Prata, Tiberius Bonates e Douglas Valiati, pelos valiosos auxílios na realização deste trabalho.

Aos amigos que tive o prazer de conviver no Rio: Jorge Bergson, Sérgio Assunção, Mara Franklin, Talita Oliveira, Bruno Muniz, Marcelo Quinderé, Henrique Limaverde, Elder Macambira, Fábio Rabelo, Leonardo Holanda, Oscarina Viana e Rômulo Martins.

Aos amigos que fiz no Rio: Alexandre Henrique Porto, Herman Walenkamp, Joana Loureiro, Ana Lúcia Pimentel, Rosa Figueiredo, Ana Flávia Macambira, Rafael e Letícia Lopes, Moisés Junior, Luidi Simonetti, Carlos Henrique Sabóia, Michele Silva entre outros.

Aos meus parentes residentes no Rio: Dalva, Juarez e Juliana Sales e Margarida Costa.

A todos os parentes e amigos de Fortaleza, de uma forma muito especial. Citar os nomes significaria estender-me demais e ainda correr o risco de deixar alguém de fora por uma falha de minha memória.

A todos os funcionários do Programa. Em especial a Lourdes e Gercina.

Ao CNPq pela bolsa de estudos concedida.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM ALGORITMO PARA O PROBLEMA DE LOCALIZAÇÃO NÃO CAPACITADO BASEADO EM TESTES DE REDUÇÃO E HEURÍSTICAS *ADD/DROP*

André Barros Pereira

Dezembro/2002

Orientador: Cláudio Thomas Bornstein

Programa: Engenharia de Sistemas e Computação

Utilizamos testes de redução na solução do Problema de Localização Não Capacitado. Infelizmente, nem sempre é possível solucionar instâncias deste problema utilizando somente esses testes. Então, desenvolvemos um conjunto de heurísticas *ADD/DROP* e as combinamos aos testes, dando origem a um algoritmo heurístico. Fizemos uma implementação desse algoritmo e a submetemos a uma série de testes computacionais. Nossos resultados foram comparados aos de uma implementação do algoritmo de Galvão e Raggi [21].

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AN ALGORITHM FOR THE SIMPLE PLANT LOCATION PROBLEM BASED ON REDUCTION TESTS AND *ADD/DROP* HEURISTICS

André Barros Pereira

December/2002

Advisor: Cláudio Thomas Bornstein

Department: Computing and Systems Engineering

We begin using reduction tests to solve the Simple Plant Location Problem. Unfortunately, these tests are generally unable to fix the status of all facilities. Therefore, ADD/DROP heuristics are additionally used. The resulting algorithm was implemented and computational tests were made. The results were compared to the Galvão and Raggi's [21] algorithm.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Quando se pretende construir um *shopping center* em uma cidade, uma das principais preocupações que se tem é a escolha de sua localização. Isso acontece porque a região em que ele será instalado tem influência direta no desempenho dos negócios. Eis alguns fatores que comprovam esse fato:

1. *Facilidade de acesso de pedestres e veículos ao shopping.* Contribui para que mais clientes dêem preferência a ele para a aquisição de bens e serviços. Dessa forma, é importante que seus arredores sejam servidos de uma boa malha viária.

2. *Proximidade em relação a clientes em potencial.* Populações de baixa renda normalmente não são freqüentadoras assíduas de *shopping centers* para suprir suas necessidades de consumo. Em bairros onde predominam essas populações normalmente não se tem expectativas de se fazer bons negócios com a instalação de um *shopping* e, portanto, essa instalação passa a ser indesejada. De um modo geral, comércios de menor porte tais como mercearias, feiras-livres e pequenos merca-

dos predominam nesses locais. Por outro lado, populações de faixas salariais mais elevadas costumam adquirir não só bens em *shoppings* mas usufruem de diversos serviços. Assim, os hábitos de consumo das pessoas que residem nas proximidades de onde se pretende instalar o *shopping*, e que estão diretamente relacionados aos seus níveis salariais, também são importantes fatores.

3. *Preferencialmente ele deve ser instalado em locais em que haja pouca ou nenhuma concorrência.* Isso diz respeito à segurança de se manter uma clientela fiel. Quanto mais empreendimentos equivalentes existirem no mesmo local, mais provável será que a população local seja dividida em termos de preferência em relação a estes.

4. *Acessibilidade do preço de compra ou aluguel do terreno em que se pretende construir o shopping.* Esses preços variam bastante conforme a valorização imobiliária do local.

5. *Boas condições ambientais.* Um lugar de intenso barulho, poluição, etc, certamente prejudicaria a permanência de clientes. Esse é um importante fator a considerar pois diz respeito ao bem-estar dos clientes enquanto estão no *shopping*. Se suas permanências são comprometidas, certamente também estarão os negócios.

Esse exemplo é uma típica aplicação da localização no setor privado, onde a minimização de custos e a maximização de lucros é desejada. Nesse caso, uma má localização poderia ocasionar grandes perdas financeiras. Em aplicações de localização características do setor público, o que geralmente se almeja é distribuir serviços de forma a atender mais eficientemente a população. As conseqüências de uma má localização nessas aplicações podem ser bem piores, como no caso de localizar ambulâncias em uma cidade de forma a atender com eficiência as chamadas de emergência da população: o tempo de atendimento de uma chamada é um fator crucial e pode ser bastante comprometido por uma localização precária, o que pode levar a perda de vidas humanas. Uma discussão sobre problemas de localização nos setores público e privado pode ser vista em Revelle e outros [52].

De forma similar, inúmeros problemas práticos envolvem tomar decisões em relação a onde localizar entidades fornecedoras de bens e/ou serviços (usualmente denominadas *facilidades*), dentro de um conjunto de possíveis locais, visando atender de forma mais vantajosa a demanda por esses bens e/ou serviços nas regiões onde se pretende instalá-las. Podemos encontrar um grande número de situações reais desse tipo em Daskin [17]. Esses problemas são chamados genericamente de *problemas de localização de facilidades.*

Mais formalmente, o problema de localização de facilidades pode ser definido, baseado em Mateus e Carvalho [45], como aquele no qual facilidades devem ser alocadas entre n possíveis locais com o objetivo de minimizar o custo total em satisfazer a demanda distribuída em m locais. O custo total consiste no custo fixo de instalar as facilidades e os custos variáveis para atender a demanda (também conhecidos por custos de distribuição, custos de transporte ou custos de alocação).

Nesse trabalho, desenvolvemos um algoritmo heurístico para a solução de um modelo específico de localização (*Simple Plant Location Problem*) e fizemos uma implementação computacional desse algoritmo. Dividiremos a apresentação do trabalho em 5 capítulos: no primeiro capítulo, situaremos o problema que trabalharemos dentro do contexto geral dos modelos de localização. Os fundamentos teóricos do nosso algoritmo serão apresentados no Capítulo 2 e no terceiro capítulo exibiremos o algoritmo. No Capítulo 4 serão exibidos os testes computacionais que fizemos com nossa implementação e faremos comparações com o desempenho computacional de uma implementação do algoritmo de Galvão e Raggi [21]. Finalmente, no último capítulo, faremos algumas conclusões e apontaremos alguns tópicos do trabalho que poderão ser futuramente trabalhados.

A seguir, situaremos nosso modelo dentro de uma classificação taxionômica dos modelos de localização (seção 1.1), especificamente dentro dos modelos discretos de localização (seção 1.2). Veremos algumas técnicas de se transformar problemas reais de localização em modelos de localização (seção 1.3), bem como as principais técnicas de soluções desses modelos (seção 1.4). Apresentaremos ainda algumas características específicas do nosso problema (seção 1.5) e exibiremos seu estado-da-arte (seção 1.6).

1.1 Breve taxionomia dos problemas de localização

Os problemas de localização de facilidades podem ser classificados de diversas formas. Apresentaremos algumas classificações e situaremos os problemas que utilizaremos no nosso trabalho de acordo com elas. Taxionomias completas sobre problemas de localização de facilidades podem ser obtidas em Daskin [17] e Brandeau e Chiu [10].

1.1.1 Problemas capacitados e não-capacitados

Se para cada local candidato a localização de facilidades for estabelecida uma capacidade máxima de oferta, o problema é dito *capacitado*. Se essa capacidade for ilimitada, o problema será denominado *não-capacitado*.

A ênfase do nosso trabalho será no tratamento de problemas de localização de facilidades não-capacitados.

1.1.2 Problemas planares, em rede e discretos

Uma das principais diferenças entre os problemas de localização é a forma com que a demanda e as facilidades são representadas e como são estabelecidos os custos de transporte entre eles. Dividiremos em três tipos, segundo Chhajed e outros[11]: planares, em rede e discretos.

Nos problemas *planares*, a demanda ocorre em qualquer lugar em um plano. Normalmente ela é representada através de uma função que estabelece a demanda em cada coordenada (X, Y) do plano. Em tais problemas, facilidades podem ser localizadas em quaisquer pontos do plano. Encontramos discussões sobre esses modelos em Hurter e Martinich [28] e Love e outros [41].

Já nos modelos de localização *em rede*, a localização de facilidades se dá em um número finito de locais, objetivando atender a demanda associada a um número também finito de locais. Estes últimos são usualmente chamados de *centros de demanda* ou *clientes*. Nesses modelos, as facilidades e os centros de demanda se localizam nos nós ou arcos de uma rede específica. Os custos de transporte entre as facilidades e centros de demanda são obtidos a partir dessa rede. Modelos em rede são discutidos em Handler e Mirchandani [27] e Mirchandani e Francis [47].

Os modelos *discretos* de localização também se caracterizam por possuirem um número finito de facilidades e centros de demanda mas diferem dos modelos em rede porque permite-se atribuir custos arbitrários de transporte entre as facilidades e centros de demanda, independentes de uma rede fixa.

A ênfase do nosso trabalho será na solução de problemas discretos de localização. Entretanto, como veremos adiante, resolveremos também problemas em rede, transformando-os em modelos discretos.

1.1.3 Problemas euclideanos e não-euclideanos

Os modelos de localização também se caracterizam pela métrica utilizada. Em alguns problemas de localização, os custos de transporte entre facilidades e demanda podem ser representados simplesmente pela distância entre eles. De acordo com a métrica obedecida no cálculo dessas distâncias, esses problemas podem ser classificados como *euclideanos* ou *não-euclideanos*. Nos primeiros, para quaisquer três pontos distintos i, j e k, valem as relações:

1. $d(i,i) = 0$;

2. $d(i,j) > 0$;

3. $d(i,j) = d(j,i)$;

4. $d(i,j) < d(i,k) + d(k,j)$;

onde $d(x,y)$ representa a distância entre os pontos x e y. Se pelo menos um desses quatro itens for violado, o problema será dito não-euclidiano.

No nosso trabalho, contemplaremos a solução de ambos os tipos de problemas.

1.2 Modelos discretos de localização

Definiremos nesta seção a formulação matemática para a qual desenvolvemos nosso algoritmo. Mas, antes disso, introduziremos os principais modelos não-capacitados, a começar pelo mais abrangente deles, o *Problema Geral de Localização Não Capacitado (PGLNC)*. Baseado em Mateus e Carvalho [45] e Galvão e Raggi [21], este problema pode ser assim definido:

$(PGLNC)$:

$$\begin{aligned}
min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i && (1.1)\\
sujeito\ a: \quad & \sum_{i \in I} x_{ij} = 1, \forall j \in J && (1.2)\\
& \sum_{i \in I} y_i \leq p && (1.3)\\
& x_{ij} \leq y_i, \forall i \in I, \forall j \in J && (1.4)\\
& x_{ij} \geq 0, \forall i \in I, \forall j \in J && (1.5)\\
& y_i \in \{0,1\}, \forall i \in I && (1.6)
\end{aligned}$$

onde $I = \{1, ..., n\}$ é o conjunto de locais candidatos à localização de facilidades e $J = \{1, ..., m\}$ é o conjunto de centros de demanda. f_i representa o custo fixo de instalar uma facilidade em um local $i \in I$ e c_{ij} o custo de transporte em atender a demanda em $j \in J$ pela facilidade $i \in I$. A variável x_{ij} corresponde à parcela da demanda de $j \in J$ atendida por $i \in I$ e $y_i, i \in I$ é uma variável binária igual a 1 se a facilidade é localizada em i ou 0, caso contrário. É comum usar o adjetivo *aberta* para designar uma facilidade que será localizada na solução e *fechada* para uma facilidade não localizada.

A função objetivo (1.1) minimiza a soma dos custos variáveis e fixos. Em (1.2) garante-se que a demanda é atendida em cada centro $j \in J$. (1.3) limita o número de facilidades abertas a p. A restrição (1.4) estabelece uma relação entre as variáveis x_{ij} e y_i e assegura que se o centro de demanda $j \in J$ é servido pela facilidade $i \in I$, então necessariamente há uma facilidade aberta em i. Em (1.5) determina-se que as parcelas x_{ij} são não-negativas e em (1.6) estabelece-se que a variável y_i é binária.

Percebemos que o PGLNC possui variáveis reais e inteiras. Entretanto, criaremos um modelo de programação linear inteira binária, acrescentando a seguinte restrição ao PGLNC: um centro de demanda $j \in J$ somente pode ser atendido por uma única facilidade $i \in I$. Ou seja, basta substituir no PGLNC a restrição (1.5) por:

$$x_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J \tag{1.7}$$

Denominaremos o modelo resultante, assim como é feito em Mateus e Carvalho [45], por *Problema de Localização Não Capacitado (PLNC)*. Ele será:

$(PLNC)$:

$$\begin{aligned}
min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i & (1.8)\\
sujeito\ a: \quad & \sum_{i \in I} x_{ij} = 1, \forall j \in J, & (1.9)\\
& \sum_{i \in I} y_i \leq p & (1.10)\\
& x_{ij} \leq y_i, \forall i \in I, \forall j \in J & (1.11)\\
& x_{ij} \in \{0,1\}, \forall i \in I, \forall j \in J & (1.12)\\
& y_i \in \{0,1\}, \forall i \in I & (1.13)
\end{aligned}$$

Observamos que o PGLNC e o PLNC são equivalentes. Como as facilidades do modelo não possuem restrições de capacidade, é fácil observar que sempre haverá uma solução ótima do PGLNC em que as variáveis x_{ij} são binárias. Basta, em uma dada solução ótima do PGLNC, para cada centro de demanda atendido por mais de uma facilidade na solução, associá-lo a uma única facilidade dentre essas, o que certamente não incorrerá em acréscimo no valor da função objetivo na solução encontrada.

A partir do PLNC, outros conhecidos modelos não-capacitados podem ser obtidos. Por exemplo, seja N o conjunto de vértices de uma rede. Se fizermos, no PLNC, $N \equiv I \equiv J$, $f_i = 0, \forall i \in N$ e substituirmos a restrição (1.10) por

$$\sum_{i \in N} y_i = p \tag{1.14}$$

teremos o clássico *Problema de P-Medianas (P-MED)*:

$(P - MED)$:

$$\begin{aligned}
min \quad & \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \\
sujeito\ a: \quad & \sum_{i \in N} x_{ij} = 1, \forall j \in N \\
& \sum_{i \in N} y_i = p \\
& x_{ij} \leq y_i, \forall i \in N, \forall j \in N \\
& x_{ij}, y_i \in \{0, 1\}, \forall i \in N, \forall j \in N
\end{aligned}$$

onde os custos c_{ij} correspondem às distâncias entre os nós i e j obtidas através dos arcos da rede. Se, no P-MED, os custos fixos forem considerados, denominamos o modelo resultante por *Problema de P-medianas de Custo Fixo (P-MEDCF)*.

Uma outra modificação que pode ser feita no PLNC é estabelecermos $p = \mid I \mid$ na restrição (1.10). Esta restrição se tornará supérflua no modelo, podendo portanto ser retirada. Teremos então, usando a mesma nomenclatura de Galvão e Raggi [21], Mateus e Carvalho [45] e Körkel [37], o *Simple Plant Location Problem (SPLP)*. Nosso algoritmo foi desenvolvido para essa formulação. Eis o modelo:

$(SPLP)$:

$$\begin{aligned}
min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i && (1.15) \\
sujeito\ a: \quad & \sum_{i \in I} x_{ij} = 1, \forall j \in J && (1.16) \\
& x_{ij} \leq y_i, \forall i \in I, \forall j \in J && (1.17) \\
& x_{ij}, y_i \in \{0, 1\}, \forall i \in I, \forall j \in J && (1.18)
\end{aligned}$$

Diferentemente de outros problemas, na literatura há muita divergência a respeito da nomenclatura desse problema. Na língua inglesa, por exemplo, Krarup e Pruzan [35] apresentam até um arquétipo dos nomes usuais que ele apresenta, como observamos na Figura 1.1. Segundo eles, incluindo omissões dos adjetivos, mais de 10 nomes já foram dados na literatura a partir desse arquétipo. Por exemplo, Erlenkotter [19] e Al-Sultan e Al-Fawzan [2] o denominam *Uncapacitated Facility Location Problem* enquanto Khumawala [32] usa o termo *Uncapacitated Warehouse Location Problem* para descrever esse problema.

Adjetivo		Substantivo		Expressão
Uncapacitated Simple Optimal	+	Plant Warehouse Facility Site	+	Location Problem

Figura 1.1: Arquétipo das usuais denominações do SPLP na literatura

Os modelos de localização que apresentamos são bem gerais e são considerados clássicos na literatura. Entretanto muitos outros modelos têm sido desenvolvidos incorporando diversas características intrínsecas a problemas do mundo real. Uma boa coletânea desses modelos pode ser vista em Revelle e Laporte [53].

Faremos na seção seguinte um breve comentário sobre a relação entre um problema real de localização de facilidades e sua modelagem matemática.

1.3 Modelagem matemática de problemas de localização

O processo de se abstrair de algumas particularidades de problemas práticos, representando-as por expressões matemáticas é conhecido por *modelagem matemática.* A quantidade de elementos do problema real a serem considerados, a forma de representação desses elementos e os objetivos da representação são fatores que distinguem uma modelagem matemática de outra para o mesmo problema.

Os *modelos matemáticos de localização* estabelecem essencialmente a representação das facilidades em potencial, dos centros de demanda e dos custos inerentes ao atendimento da demanda pelas facilidades. Tais modelos objetivam minimizar esses custos, respondendo a questões do tipo:

1. Quantas facilidades devem ser localizadas?
2. Onde cada facilidade será localizada?
3. De que forma as facilidades localizadas serão alocadas aos centros de demanda?

Na prática, o que se faz normalmente é modelar o problema de uma forma que se aproxime de um dos típicos modelos de localização, como os que vimos na seção anterior. Pois, para estes, diversas técnicas de solução já foram desenvolvidas e trabalhadas ao longo do tempo, como veremos na próxima seção.

Para exemplificar o processo de modelagem matemática de problemas de localização, consideremos novamente o exemplo do *shopping center*. Suponha, entretanto, que desejamos instalar não apenas um *shopping* mas diversas unidades de uma rede de *shopping centers*. Queremos que essa rede seja instalada com o menor custo possível. Consideremos uma modelagem simplificada desse problema como SPLP:

O conjunto de locais candidatos à instalação de unidades da rede de *shoppings* seria modelado como o conjunto de facilidades em potencial (I). Os custos monetários estimados de instalação de cada unidade da rede seriam modelados como os custos fixos de instalação das facilidades (f_i).

Fazendo levantamentos sobre os hábitos de consumo das populações em torno dos locais onde se pretende instalar unidades da rede de *shoppings*, pode-se delimitar dentro de uma mesma região, agrupamentos de clientes em potencial, cada um com sua respectiva demanda associada, que pode ser uma média dos gastos com produtos e serviços em *shoppings* em um determinado intervalo de tempo. A distância entre as unidades do *shopping* e esses agrupamentos representam dificuldades no atendimento dessa demanda. Cada grupo de clientes pode ser modelado como um centro de demanda. O conjunto de agrupamentos seria, no modelo, o conjunto de centros de demanda (J). De uma forma simplificada, os custos variáveis de se atender as demandas nesses centros podem ser dados pelo produto entre o valor da demanda média nesses locais e a distância entre eles e o local candidato à instalação do *shopping* $(c_{ij}, \forall i \in I, \forall j \in J)$.

A decisão de instalar ou não uma unidade da rede de *shoppings* no modelo seria retratada pela variável y_i e o fato de uma unidade estar ou não atendendo um aglomerado de clientes em potencial seria indicado pela variável x_{ij}.

A função objetivo seria a função de custos monetários estimada para que se concretizasse as instalações dos *shoppings* e fossem atendidas as demandas.

O modelo consistiria então em minimizar a função (1.15) sujeito às restrições (1.16), (1.17) e (1.18).

Uma modelagem um pouco mais sofisticada desse problema poderia, por exemplo, contemplar de alguma forma os aspectos citados no início deste capítulo, que tornam relevantes a escolha da localização do *shopping*:

1. *Facilidade de acesso de pedestres e veículos ao shopping.* A presença de uma malha viária deficiente em torno do *shopping* seria uma qualidade bastante indesejável e poderia ser modelada através de uma variável $p_1(i), i \in I$ que penalizaria os custos fixos f_i.

2. *Proximidade em relação a clientes em potencial.* Já contemplado através da definição de c_{ij}

3. *Preferencialmente ele deve ser instalado em locais em que haja pouca ou nenhuma concorrência.* A presença de um concorrente próximo a uma unidade do *shopping* poderia ser igualmente modelada como uma penalização de f_i. Utilizaremos uma variável $p_2(i), i \in I$ para esse fim. Além disso, a própria solução tenderia a evitar que uma unidade da rede de *shoppings* fosse colocada perto de outra da mesma rede.

4. *Acessibilidade do preço de compra ou aluguel do terreno em que se pretende construir o shopping.* Essa característica já está embutida na

definição dos custos fixos de instalação das facilidades.

5. *Boas condições ambientais.* Finalmente, uma terceira variável ($p_3(i), i \in I$) indicaria a penalização que os custos fixos f_i sofreriam de acordo com os problemas ambientais da região onde se quer instalar o *shopping*.

Nesse novo modelo, o valor dos novos custos f_i' seriam dados por: $f_i' = f_i + p_1(i) + p_2(i) + p_3(i), \forall i \in I$. Os demais elementos permaneceriam inalterados.

1.4 Técnicas de soluções de problemas de localização

A principal dificuldade na solução de problemas de localização é a quantidade de mínimos locais que a função objetivo apresenta dentro do conjunto de soluções viáveis do problema e que nem sempre constituem mínimos globais ou mesmo boas soluções. Uma completa enumeração desses pontos é impraticável para a maioria dos problemas. Por isso, várias técnicas têm sido desenvolvidas visando determinar a melhor maneira de enumerar e selecionar esses pontos de forma que o mínimo global ou uma solução próxima dele possa ser obtida. Dividiremos a apresentação das técnicas de acordo com a classificação dos modelos de localização em planares, em rede ou discretos.

Os problemas planares se caracterizam por uma formulação contínua e para resolvê-los normalmente são empregadas técnicas de programação não-linear, geometria computacional e/ou otimização global. Um levantamento da aplicação dessas técnicas a problemas planares pode ser encontrado em Plastria [50].

Existem duas abordagens principais na solução dos modelos em rede:

- Procedimentos que reduzem tais problemas a modelos discretos que garantidamente reproduzem a solução ótima dos modelos em rede. Em Daskin [17] podemos encontrar procedimentos como estes.

- Algoritmos que utilizam eficientemente a estrutura especial da rede para solucionar os modelos. Podemos obter mais detalhes sobre esses algoritmos em Labbe e outros [39].

Os modelos discretos de localização são normalmente constituídos por modelos de programação linear inteira ou modelos de programação linear inteira mista e geralmente técnicas de natureza combinatória são empregadas na solução desses problemas. Em Mateus e Carvalho [45] temos um excelente levantamento das principais técnicas empregadas. Eles dividem essas técnicas em 5 áreas:

- **Decomposição**: elas seguem, em geral, o método de decomposição de Benders. Como as formulações de alguns problemas envolvem variáveis inteiras e reais, este método separa a parte inteira da parte real, resolvendo problemas menores conectados a um problema mestre.

- **Enumeração**: procuram subdividir o problema original em subproblemas menores de fácil resolução, enumerando, implicitamente ou explicitamente, todos os pontos extremos. Os algoritmos de enumeração são basicamente do tipo de Separação e Avaliação (*branch-and-bound*), que são técnicas de busca em árvore, onde cada nó da árvore representa um subconjunto de soluções viáveis. Um algoritmo de separação e avaliação fica caracterizado ao definirmos como separar o conjunto de soluções viáveis e como avaliar a função determinando limites superiores e inferiores.

- **Heurísticas**: objetivam determinar boas soluções de forma eficiente. Com essa prioridade, as heurísticas caracterizam-se pela flexibilidade e simplicidade computacional.

- **Métodos duais**: procuram resolver o dual do problema com o objetivo de obter limites inferiores justos.

- **Métodos baseados em Programação Linear**: determinam limites inferiores para os modelos, relaxando as restrições de integralidade e resolvendo o Problema de Programação Linear resultante.

Conforme mencionamos, nosso algoritmo foi desenvolvido direcionado à solução de modelos discretos de localização, mais especificamente, para o SPLP. Em nosso trabalho, fizemos uso sobretudo de heurísticas.

1.5 Simple Plant Location Problem

O SPLP é um problema que tem sido vastamente estudado na literatura. Muitos algoritmos têm aproveitado suas particularidades a fim de obter soluções de boa qualidade, em tempos razoáveis. Antes de listarmos uma coletânea de trabalhos sobre o SPLP, vamos conhecer algumas das propriedades desse problema:

1. *O SPLP é um problema NP-difícil*, como podemos observar em Garey e Johnson [23] e Lenstra e Kan [40]. Porém, em alguns casos especiais, ele é solucionável polinomialmente como vemos em Kolen [33] e Krarup e Bilde [34].

2. *O SPLP pode ser matematicamente decomposto em dois subproblemas interdependentes.*

   De acordo com Al-Sultan e Al-Fawzan [2], os subproblemas são:

   - **Localização**: determinar as facilidades a serem localizadas ($y_i's$).

   - **Alocação**: para aquelas facilidades localizadas, estabelecer como será a distribuição aos centros de demanda ($x_{ij}'s$)

   Para cada solução do subproblema de localização, uma solução ótima do subproblema de alocação é facilmente obtida. Mais especificamente, para qualquer dado vetor y, uma associação ótima aos valores de $x_{ij}'s$ pode ser obtida usando a seguinte fórmula:

$$k_j = \arg \min_{y_i=1, i \in I} c_{ij}$$

   Para todo $i \in I$ e $j \in J$:

$$x_{ij} = \begin{cases} 1, & \text{se } i = k_j \\ 0, & \text{caso contrário} \end{cases}$$

Utilizamos para simplificar a definição de k_j e utilizaremos outras vezes mais adiante a notação "arg". Para $K \subseteq I$, definimos

$$j = \arg\ \min / \max_{i \in K} f(i) \Leftrightarrow f(j) = \min / \max_{i \in K} f(i)$$

3. *O SPLP se relaciona aos problemas de empacotamento, cobertura e particionamento*, como se observa em Krarup e Pruzan [35].

1.6 Trabalhos relacionados

A primeira formulação explícita do SPLP data da década de 60 e é freqüentemente atribuída a Balinski [4]. Esse trabalho foi apresentado no simpósio *The IBM Scientific Symposium on Combinatorial Problems* em março de 1964 mas permaneceu sem ser publicado até o ano de 1966. Entretanto, alguns pesquisadores acreditam que o SPLP já teria sido abordado em trabalhos anteriores a este (ver Krarup e Pruzan [35]).

Vamos exibir o estado-da-arte do SPLP, de acordo com a classificação das técnicas de soluções vistas anteriormente, embora, em muitos trabalhos, mais de uma técnica seja empregada, como poderemos observar adiante. A literatura sobre esse problema é muito vasta e citaremos aqui alguns dos principais trabalhos.

Os algoritmos de decomposição, seguindo a linha Benders, conseguiram bons resultados através da geração de cortes no conjunto de soluções viáveis como se observa em Magnanti [42] e Wolsey e Nemhauser [60].

Na linha de Separação e Avaliação, logo em seguida ao trabalho de Balinski, Efroymson e Ray [18] propuseram um algoritmo para uma formulação modificada do SPLP. Khumawala [32] utilizou a estrutura especial do SPLP para melhorar o algoritmo de separação e avaliação de Efroymson e Ray. Guignard [25] propôs um algoritmo que utiliza desigualdades de Benders geradas durante um procedimento dual ascendente lagrangeano. Galvão e Raggi [21] propuseram um algoritmo em três fases para o PLNC, que tem o SPLP como um caso especial.

Entre as heurísticas, destacamos as gulosas ADD propostas em Kuehn e Hamburguer [38], que procuram a cada iteração selecionar a facilidade mais econômica. Uma heurística que descarta a facilidade menos econômica a cada iteração foi desenvolvida em Feldman e outros [20] e foi denominada DROP. Manne [43] desenvolveu uma heurística denominada SAOPMA que combinava as duas anteriores. Rapp [51] e Cooper [13], independentemente, também combinaram ADD e DROP e desenvolveram uma heurística conhecida por ALA (Alternate Location Allocation). Uma variação de ALA chamada SHIFT foi desenvolvida por Kuehn e Hamburguer [38]. Conforme seja usada inicialmente a heurística ADD ou DROP teremos as heurísticas SHIFT ou ALA. Destaca-se também o método de substituição de vértices desenvolvido por Teitz e Bart [58], denominado VSM (Vertex Substitution Method) desenvolvido em princípio para o P-MED mas que foi adaptado para o SPLP por Cornuéjols e outros [14]. Mais adiante, Beasley [5] propôs heurísticas lagrangeanas para vários problemas de localização que incluem o SPLP. As metaheurísticas também têm sido amplamente aplicadas ao SPLP.

Nessa linha, destacamos os trabalhos de Al-Sultan e Al-Fawzan [2] e Michel e Hentenryck [46] que apresentaram algoritmos baseados na metaheurística busca tabu; Alves e Almeida [3] utilizaram *Simulated Annealing* na solução do problema e Kratica e outros [36] fizeram uso de um algoritmo genético.

Bilde e Krarup [7] e Erlenkotter [19] introduziram os métodos duais para a solução do SPLP. Mais tarde, Tcha e outros [57] desenvolveram uma heurística dual para esse problema bastante similar ao procedimento de Erlenkotter, afirmando que ela produz soluções que são, na maioria dos casos, superiores àquelas alcançadas pelo procedimento de Erlenkotter com um leve aumento no tempo computacional. Körkel [37] mostrou como modificar uma versão primal-dual do algoritmo exato de Erlenkotter para conseguir um procedimento melhorado. Conn e Cornuéjols [12] sugeriram um novo método baseado na solução exata do dual de uma relaxação linear do SPLP via projeções ortogonais. Gao e Robinson [22] apresentaram um modelo geral e um procedimento dual de separação e avaliação de solução para encontrar soluções ótimas para vários problemas de localização, incluindo o SPLP.

Nos métodos baseados em programação linear, destacam-se os trabalhos Spielberg [56], Schrage [54] e Garfinkel e outros [24], que exploram o fato de as variáveis serem limitadas superiormente. Simão e Thizy [55] desenvolveram um algoritmo dual-simplex direcionado ao problema. Guignard e Spielberg [26] e Cornuéjols e Thizy [16] aplicaram um algoritmo primal de subgradiente. Mais recentemente, Jain and Vazirani [30] formularam algoritmos primal-dual na solução de problemas não capacitados, dentre eles, o SPLP.

Capítulo 2

Fundamentação Teórica

Veremos, a seguir, as bases teóricas que fundamentam o nosso algoritmo para a solução do SPLP. Basicamente, utilizamos testes de redução e desenvolvemos um conjunto de heurísticas ADD/DROP. Vejamos em detalhes esses fundamentos:

2.1 Testes de redução

Os testes de redução determinam, *a priori*, se algumas facilidades devem ser abertas ou fechadas na solução ótima. O termo *redução* se deve ao fato de se conseguir, desta maneira, reduzir a dimensão do problema dado originalmente.

O uso das regras de redução é encontrado originalmente nos trabalhos de Efroymson e Ray [18] e Khumawala [32]. Posteriormente, diversos trabalhos foram desenvolvidos, estendendo alguns dos resultados dos primeiros e aplicando-os ao problema de localização capacitado, como podemos observar em Akinc e Khumawala [1], que utiliza os testes dentro de um algoritmo de separação e avaliação. Jacobsen [29] e Mateus e Bornstein [44] utilizaram esses testes dentro de heurísticas ADD/DROP. Mais recentemente, temos em Bornstein e Azlan [8] a combinação desses testes à metaheurística *simulated annealing*. Em Campêlo e Bornstein [49], temos uma nova abordagem que combina os testes de redução a heurísticas ADD/DROP, que são implementadas com a ajuda de limites superiores e inferiores, usando relaxação lagrangeana.

Utilizaremos, em nosso trabalho, a abordagem unificada encontrada em Bornstein e Azlan [8] e Campêlo e Bornstein [49], que simplifica a apresentação dos testes de redução através da utilização da função $\Delta_i(.)$. Essa função nos dará os critérios para abertura/fechamento de facilidades. Mais adiante, vamos defini-la formalmente mas antes definiremos os elementos constituitivos dessa função:

Considere um subconjunto $K \subseteq I$ de facilidades. Seja $w(K)$ a função que nos dá o mínimo valor dos custos variáveis em se atender todos os centros de demanda de J pelas facilidades de K. Ou seja,

$$w(K) = \sum_{j \in J} \min_{k \in K} c_{kj}$$

Se $K = \emptyset$, definimos $w(K) = +\infty$.

Para $i \in I - K$, seja $\delta_i(K) = w(K) - w(K \cup i)$. Essa função avalia o acréscimo/decréscimo nos custos variáveis se fechamos/abrimos a facilidade i. Se $K = \emptyset$ então $\delta_i(K) = \delta_i(\emptyset) = +\infty - w(i) = +\infty - \sum_{j \in J} c_{ij} = +\infty$. Se $K \neq \emptyset$, podemos alternativamente escrever a função $\delta_i()$ em função dos custos variáveis c_{ij}, da seguinte forma:

$$\begin{aligned}
\delta_i(K) &= w(K) - w(K \cup i) \\
&= \sum_{j \in J} \min_{k \in K} c_{kj} - \sum_{j \in J} \min_{k \in K \cup i} c_{kj} \\
&= \sum_{j \in J} \min_{k \in K} c_{kj} - \sum_{j \in J} \min\{\min_{k \in K} c_{kj}, c_{ij}\} \\
&= \sum_{j \in J} \max\{0, \min_{k \in K} c_{kj} - c_{ij}\}
\end{aligned}$$

Definimos então a função $\Delta_i(.)$ da seguinte forma $\Delta_i(K) = f_i - \delta_i(K), \forall i \in I - K$. Essa função avalia o balanço entre os custos fixos e variáveis em relação a facilidade i. Se $K = \emptyset$ então $\Delta_i(\emptyset) = f_i - \delta_i(\emptyset) = f_i - \infty = -\infty$. Para $K \neq \emptyset$, podemos escrever:

$$\Delta_i(K) = f_i - \sum_{j \in J} \max\{0, \min_{k \in K} c_{kj} - c_{ij}\}$$

Podemos destacar duas importantes propriedades da função $w(K)$, de acordo com Wolsey [59] e Nemhauser e outros [48]: ela é não-crescente e supermodular. Uma função $w(K)$ é denominada não-crescente se $w(K) \leq w(K'), \forall K' \subseteq K$. E ela é dita supermodular (ou equivalentemente $-w(K)$ submodular) se $w(K) - w(K \cup i) \leq w(K') - w(K' \cup i), \forall K' \subseteq K, \forall i \notin K$. Observa-se que supermodularidade é uma espécie de concavidade. Mostremos que essas duas propriedades se verificam.

Propriedade 1: w() é não-crescente.

Prova:

Sejam $K \subseteq I$ e $K' \subseteq K$. Devemos mostrar que $w(K) \leq w(K')$. Vamos dividir em dois casos:

- Se $K' = \emptyset$ então temos imediatamente que $w(K') = w(\emptyset) = +\infty \geq$

$w(K)$.

- Se $K' \neq \emptyset$ então, por definição

$$w(K) = \sum_{j \in J} \min_{k \in K} c_{kj} \quad \text{e} \quad w(K') = \sum_{j \in J} \min_{k \in K'} c_{kj}$$

Temos, uma vez que $K' \subseteq K$, o seguinte fato

$$\min_{k \in K} c_{kj} \leq \min_{k \in K'} c_{kj}, \forall j \in J$$

Então,

$$\underbrace{\sum_{j \in J} \min_{k \in K} c_{kj}}_{w(K)} \leq \underbrace{\sum_{j \in J} \min_{k \in K'} c_{kj}}_{w(K')}$$

Logo, $w(K) \leq w(K'), \forall K' \subseteq K$. □

Como conseqüência da propriedade 1, temos que:

$$\delta_i(K) \geq 0, \forall K \subseteq I, \forall i \notin K \tag{2.1}$$

De fato, $K \subseteq K \cup i \rightarrow w(K) \geq w(K \cup i) \rightarrow w(K) - w(K \cup i) \geq 0 \rightarrow \delta_i(K) \geq 0$.

Propriedade 2: w() é supermodular.

Prova:

Sejam $K \subseteq I$, $K' \subseteq K$ e $i \notin K$. Devemos mostrar que $w(K) - w(K \cup i) \leq w(K') - w(K' \cup i)$. Mas, temos que $\delta_i(K) = w(K) - w(K \cup i)$, logo, o que se quer provar é que $\delta_i(K) \leq \delta_i(K')$. Vamos dividir em dois casos:

- Se $K' = \emptyset$ então, de forma imediata, temos que $\delta_i(K') = \delta_i(\emptyset) = \infty \geq \delta_i(K)$.

- Se $K' \neq \emptyset$ então, por definição

$$\delta_i(K) = \sum_{j \in J} \max\{0, \min_{k \in K} c_{kj} - c_{ij}\} \quad \text{e} \quad \delta_i(K') = \sum_{j \in J} \max\{0, \min_{k \in K'} c_{kj} - c_{ij}\}$$

Temos, uma vez que $K' \subseteq K$, o seguinte fato

$$\min_{k \in K} c_{kj} \leq \min_{k \in K'} c_{kj}, \forall j \in J$$

Então,

$$\min_{k \in K} c_{kj} - c_{ij} \leq \min_{k \in K'} c_{kj} - c_{ij}, \forall j \in J$$

O que implica

$$\max\{0, \min_{k \in K} c_{kj} - c_{ij}\} \leq \max\{0, \min_{k \in K'} c_{kj} - c_{ij}\}, \forall j \in J$$

Assim,

$$\underbrace{\sum_{j \in J} \max\{0, \min_{k \in K} c_{kj} - c_{ij}\}}_{\delta_i(K)} \leq \underbrace{\sum_{j \in J} \max\{0, \min_{k \in K'} c_{kj} - c_{ij}\}}_{\delta_i(K')}$$

Logo, $\delta_i(K) \leq \delta_i(K'), \forall K' \subseteq K, \forall i \notin K$. $\square$

Como conseqüência da propriedade 2, temos que:

$$\Delta_i(K) \geq \Delta_i(K'), \forall K' \subseteq K, \forall i \notin K \tag{2.2}$$

De fato, $\delta_i(K) \leq \delta_i(K') \rightarrow -\delta_i(K) \geq -\delta_i(K') \rightarrow f_i - \delta_i(K) \geq f_i - \delta_i(K') \rightarrow \Delta_i(K) \geq \Delta_i(K')$.

Antes de apresentarmos a definição dos testes de redução, faz-se necessário definir os conjuntos em que as facilidades serão inseridas de acordo com o seu status (fechadas, abertas ou indefinidas): seja K_0 o conjunto de facilidades fechadas ($K_0 = \{i \in I \mid y_i = 0\}$), K_1 o conjunto de facilidades abertas