MARIA LEONOR DE ALMEIDA GOUVEIA
OLIVEIRA ALVES

SOLUÇÕES APROXIMADAS PARA O PROBLEMA DE LOCALIZAÇÃO SIMPLES

ALGORITMO SIMULATED ANNEALING

LISBOA
1991

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Métodos Matemáticos para Economia e Gestão de Empresas do Instituto Superior de Economia da Universidade Técnica de Lisboa.

AGRADECIMENTOS

À **Professora Teresa Chaves Almeida**, orientadora desta Tese, agradeço todas as sugestões, críticas e disponibilidade que sempre mostrou, para comigo colaborar na realização deste trabalho. Quero também deixar expressa a minha admiração pela forma exigente e rigorosa como conduziu toda a orientação.

Ao Prof. José Tavares, Presidente do Conselho Científico do ISEC, agradeço os apoios e incentivo que sempre me deu durante este período.

Ao Prof. A. Tectónio, Presidente do Conselho Directivo do ISEC, agradeço o apoio e compreensão demonstrada, sem a qual não teria sido possível compatibilizar funções docentes e frequência do Curso de Mestrado.

Ao INIC agradeço o apoio financeiro que me deu, sob a forma de uma bolsa de estudo que tornou possíveis as deslocações.

Aos meus pais e a todas as pessoas que comigo colaboraram deixo uma palavra de agradecimento. À Leta quero agradecer, em particular.

Por último, mas de uma forma muito especial, agradeço ao Zé Manel pelo estímulo, encorajamento e colaboração que continuamente me deu.

À Joana e Catarina quero prometer que tudo farei para as compensar.

RESUMO

O Problema de Localização Simples (SPL) consiste na determinação da localização de equipamentos (fábricas, hospitais, bancos, armazéns, etc.) de modo a minimizar os custos (maximizar os lucros) de satisfazer procuras previamente conhecidas. Em geral são considerados custos fixos de instalação e custos de transportes entre serviços e destinatários.

O Problema de Localização Simples é um problema NP-difícil que embora tenha uma estrutura simples é bastante abrangente no sentido em que, com pequenas modificações, permite obter a formulação de muitos outros problemas.

Neste trabalho, começaremos por apresentar alguns dos principais problemas de localização e sua formalização.

Serão apresentados alguns dos métodos de resolução do Problema de Localização Simples que têm sido propostos na literatura. Uma atenção especial será dedicada a um método exacto, proposto por Bilde - Krarup e Erlenkotter, que até hoje tem vindo a ser considerado o melhor método de resolução.

Por fim será feita uma apresentação do método *"Simulated Annealing"* para a resolução de problemas de optimização combinatória e das implementações que dele realizámos para a resolução aproximada do SPL. Serão apresentados resultados computacionais comparativos, para um conjunto de problemas retirados da literatura, relativos a diversas alternativas quanto à escolha da solução inicial e dos valores a atribuir aos parâmetros do método.

LISTA DE ABREVIATURAS

CP — Problema de Cobertura

CPL — Problema de localização com restrições de capacidade

DPL — Problema de localização dinâmica

DSRS — Dual da relaxação linear forte do problema de localização simples

DWRS — Dual da relaxação linear fraca do problema de localização simples

GDPL — Problema de localização dinâmica generalizado

MPL — Problema de localização de multi-serviços

PP — Problema da partição

SCPL — Problema de localização estocástico com restrições de capacidade

SPL — Problema de localização sem restrições de capacidade

SPL-S — Problema de localização simples (sem restrições agregadas)

SPL-W — Problema de localização simples com restrições agregadas

SPLM — Problema de localização sem restrições de capacidade com vários níveis

SRS — Relaxação linear forte do problema de localização simples

WRS — Relaxação linear fraca do problema de localização simples

INDICE

CAPÍTULO 1

INTRODUÇÃO 1

CAPÍTULO 2

PROBLEMAS DE LOCALIZAÇÃO

2.1 — Modelos de Localização Planares 5
2.2 — Modelos de Localização em Redes 11
2.3 — Modelos de Localização Discretos 21

CAPÍTULO 3

PROBLEMA DE LOCALIZAÇÃO SIMPLES — SPL

3.1 — Formulação Matemática do SPL 43
3.2 — Problemas de Localização, Partição e Cobertura 47
3.3 — Complexidade Computacional 48
3.4 — Métodos de Resolução 51
3.4.1 — Heurísticas 51
3.4.2 — Relaxações Lineares 56
3.4.3 — Algoritmos Exactos 62
3.4.3.1 — Algoritmos Baseados na Resolução do Dual 63
3.4.4 — Dual Ascendente e Relaxação Lagrangeana 81
3.4.5 — Exemplos de Problemas a que tem sido Aplicado o Dual Ascendente 87

CAPÍTULO 4

SIMULATED ANNEALING

4.1	–	Introdução	88
4.2	–	Analogia Física	90
4.3	–	Algoritmo Simulated Annealing	95
4.3.1	–	Parâmetros. Arrefecimento Escalonado	101
4.3.2	–	Aplicação ao Problema de Localização Simples	104
4.3.2.1	–	Arrefecimento Escalonado. Resultados Computacionais	106
4.4	–	Conclusões	126

REFERÊNCIAS BIBLIOGRÁFICAS 127

BREVE NOTA HISTÓRICA

A necessidade de resolução de problemas de localização, tais como localização de fábricas, serviços de urgência, localização de armazéns, localização de equipamentos, postos de distribuição, redes de dados, etc , está na origem do interesse por este tema por parte dos investigadores ligados às mais diversas áreas desde economia e gestão até à engenharia, geografia e, evidentemente, matemática.

É a partir do início dos anos 60, coincidindo com o incremento da utilização do computador, que a teoria da localização começa a surgir de forma estruturada. Até então, apenas alguns trabalhos dispersos foram publicados, o primeiro dos quais se deve a FERMAT, sobre *o problema 1-mediana* que data do início do séc. XVII e que ele enunciou do seguinte modo: "*dados três pontos no plano, determinar um quarto ponto de modo a que a soma das distâncias deste aos três pontos dados seja mínima*" ; posteriormente, em 1857, SYLVESTER põe pela primeira vez *o problema 1-centro* que formulou assim: " *pretende-se determinar o menor círculo que contenha um determinado número de pontos do plano*". O centro do círculo é a localização minimax de um serviço em relação ao conjunto de pontos. ALFRED WEBER é tido como o primeiro autor a introduzir um modelo de localização num livro publicado em 1909. O modelo era resultante dum problema de localização de um armazém de modo a que a soma das distâncias a um conjunto de clientes, distribuidos no espaço, fosse mínima. A localização óptima foi obtida

por um método geométrico. Em 1937, WEISZFELD, apresenta um método iterativo para determinar: *"a localização de um ponto de forma a que a soma ponderada das distâncias Euclideanas desse ponto a n pontos dados seja mínima."* Este método publicado num jornal Japonês foi praticamente desconhecido até final dos anos 60, tendo, por isso, sido desenvolvido por outros autores nos finais dos anos 50. Entretanto, outras publicações foram surgindo, mas de uma forma dispersa.

Em 1974 Francis e Goldstein publicaram uma bibliografia sobre localização com 226 referências e em 1985 Domschke e Drexl publicaram uma bibliografia sobre este mesmo tema com 1800 referências. Sem pensarmos que qualquer delas seja exaustiva, como o dizem os próprios autores, pensamos que esta observação nos dá uma ideia sobre a forma como cresceu, durante esta década, o interesse pela resolução destes problemas, tendo como resultado este aumento explosivo de literatura sobre o assunto.

À medida que aumenta o número de investigadores a debruçar-se sobre o tema maior é o grau de especialização. Assim, embora haja problemas clássicos como, por exemplo, o problema de localização simples, que se consideram razoavelmente resolvidos, no sentido em que existem algoritmos que permitem obter boas soluções (suboptimais, em geral), continuam a ser desenvolvidos algoritmos para melhorar, modificar, ou estender outros já existentes. Para além disto, as múltiplas aplicações do problema de localização simples a situações concretas, tão diversas, têm levado ao aparecimento de uma família de formulações de modelos de localização/distribuição que não pára de crescer. Tal família cobre problemas que variam em complexidade, desde localização de um só serviço /multi-serviços, um só nível/vários níveis,

estáticos/dinâmicos, determinísticos/estocásticos. No que diz respeito a técnicas de resolução, são igualmente variadas incluindo, soluções intuitivas, heurísticas, exactas, etc.. Por estas razões, continua a ser um tema de investigação actual.

CAPÍTULO 1

INTRODUÇÃO

Duma forma genérica pode dizer-se que *os problemas de localização* consistem na determinação da localização de fábricas, armazens, equipamentos, de modo a minimizar os custos (maximizar os lucros) de satisfazer a procura. Em geral, há custos fixos de instalação e custos de transporte entre origens e destinatários, estes, em grande parte dos casos, proporcionais às distâncias.

Não existe uma estrutura unificada que permita uma classificação definitiva dos modelos de localização, vejam-se Francis (1983), Aikens (1985), Brandeau e Chiu (1989).

É frequente encontrar na literatura, uma classificação em três categorias que iremos adoptar:

— *modelos contínuos* também chamados *planares* ;

— *modelos de localização discretos*;

— *modelos de localização em redes*.

Quando o conjunto de localizações possíveis é o plano ou um subconjunto deste, não vazio, o modelo correspondente é um *modelo de localização contínuo* ou *modelo planar*. O número de localizações possíveis é infinito e não são considerados custos fixos. Estes modelos envolvem distâncias Euclideanas ou outras mais gerais, distâncias l_p, de que a Euclideana é caso particular com $p=2$.

As incógnitas são as coordenadas dos equipamentos a instalar. Por serem contínuos, estes modelos são mais tratáveis sob o ponto de vista analítico, mas são pouco realistas, como veremos a seguir. Por isso, são utilizados, na maior parte dos casos, para fornecer informações de tipo qualitativo ou em termos de vizinhança, na tomada de decisões. Os métodos de solução utilizados são, em geral, métodos de *Programação Não Linear*.

Os modelos a que chamámos *de localização discretos*, são modelos em que as localizações possíveis são previamente conhecidas, em número finito e em que as decisões são influenciadas pelos custos fixos de instalação. Estes modelos são mais flexíveis que os anteriores porque permitem alterações das localizações, provocadas por irregularidades do terreno ou características dos serviços ou clientes, sem que a estrutura do problema seja alterada. Os métodos utilizados na obtenção de soluções são métodos de *Programação Inteira* ou *Combinatória*.

Os *modelos de localização em redes* são modelos em que o conjunto de localizações possíveis é uma rede formada por um número finito de arcos. Qualquer ponto da rede, quer seja extremidade dum arco, quer seja ponto intermédio, pode ser um local de procura ou localização do serviço. Os métodos utilizados na procura de soluções são habitualmente os métodos utilizados nos *modelos discretos*. Isto porque é geralmente possível identificar um conjunto finito de localizações na rede que contenha uma solução óptima. A motivação para este tipo de modelos resulta das numerosas situações práticas em que há necessidade de alocação de serviços junto de redes de transportes ou comunicações.

Podemos dizer que a teoria dos modelos de localização em redes pode ser

vista como uma teoria intermédia entre as teorias de modelos contínuos e de modelos discretos. Se por um lado o conjunto de localizações possíveis é contínuo, por outro têm propriedades que permitem resolvê-los por métodos utilizados para modelos discretos.

Neste trabalho iremos debruçar-nos sobre um problema que é um caso particular de *modelos discretos* :

" *O Problema de Localização Simples* "

Este problema pode enunciar-se de uma forma breve do seguinte modo:

" Há n clientes para atender a partir de m serviços possíveis. Pretende-se decidir quais dos serviços devem ser abertos de modo a que cada cliente seja atendido a partir de um e um só serviço ao menor custo possível ".

A sua simplicidade, quer em termos de enunciado, quer em termos de formulação, e a sua grande importância prática levaram a que investigadores de diversas origens o procurassem resolver e aplicar a uma grande variedade de situações concretas. Em consequência disto, ele aparece na literatura sob variadíssimas designações, conforme as aplicação a que se destina ou a língua em que é feita a publicação. Praticamente podem encontrar-se títulos que resultam de todas as combinações dos adjectivos (*uncapacited, simple, optimal*) com os substantivos (*plant, wharehouse, facility, site*) seguidos da palavra *"location"*. Existe por isso também uma grande variedade de algoritmos exactos, heurísticas, relaxações, para obter soluções para este problema.

Para simplificar, passaremos a utilizar a sigla SPL para o problema de localização simples que resulta de " Simple Plant Location ".

O SPL pode classificar-se como um problema discreto, determinístico, com um só-serviço, um só-nível, e sob o ponto de vista de complexidade " *intratável* "

(NP-hard), isto é, pertence à classe dos problemas não solúveis em tempo polinomial, Garey e Johnson (1979).

No capítulo 2, faremos uma classificação e apresentação de alguns dos principais modelos de localização .

No capítulo 3 estudaremos o SPL no que se refere à complexidade e apresentação de alguns dos métodos mais utilizados para a sua resolução.

No capítulo 4, apresentaremos o *Algoritmo Simulated Annealing* aplicado a problemas de optimização combinatória. Faremos um estudo comparativo de duas versões do algoritmo *Simulated Annealing* que implementámos para o SPL, utilizando soluções iniciais pseudo-aleatórias e soluções iniciais obtidas por uma heurística primal-dual. Apresentaremos resultados computacionais obtidos pelos dois métodos, pelos correspondentes algoritmos de optimização local e ainda pelo método de Ajustamento do Dual apresentado por Erlenkotter (1978).

CAPÍTULO 2

PROBLEMAS DE LOCALIZAÇÃO

2.1 — MODELOS DE LOCALIZAÇÃO PLANARES

Nestes modelos as localizações dos equipamentos são tratadas matemáticamente como pontos do plano, os custos são proporcionais às distâncias e estas são determinadas de acordo com alguma das métricas l_p que definimos a seguir.

A formulação do problema no caso de localização de um só serviço para atender n clientes ou localização de um "novo serviço" em relação a n serviços já instalados é:

2.1.1 — PROBLEMA DE LOCALIZAÇÃO PLANAR

Formulação do Problema

$$\min_{X} W(X) = \sum_{j=1}^{n} w_j \; l_p (X, Y_j) \qquad (2.1.1)$$

em que :

n = número de clientes;

w_j = constantes de proporcionalidade dos custos em relação

às distâncias;

$X = (x_1 , x_2)$ é a localização do novo serviço;

$Y_j = (y_{j1} , y_{j2})$ é a localização dos clientes ;

$l_p (X , Y_j)$ é a distância do novo serviço ao cliente j ,

definida por:

$$l_p (X , Y_j) = (| x_1 - y_{j1} |^p + | x_2 - y_{j2} |^p)^{1/p}, \; p \geq 1.$$

Love, Morris e Wesolowsky (1988) analisam estes modelos, nos casos de distâncias rectangulares, p=1, Euclideanas, que correspondem a p=2, e distâncias l_p em geral, enunciando algumas das propriedades das funções l_p, nomeadamente, sobre a convexidade destas funções, para valores de $p \geq 1$, e, portanto, de WO; descrevem alguns algoritmos para a determinação da localização óptima do novo serviço, como, por exemplo, o algoritmo de Weiszfeld. Primeiro, para distâncias Euclideanas, e depois a sua generalização ao caso de distâncias l_p com $p \neq 2$.

Os modelos planares envolvem certas hipóteses básicas que limitam o seu realismo e consequentemente reduzem o seu interesse. Vejamos algumas dessas hipóteses:

[H1] — Um plano é uma aproximação adequada de uma superfície esférica.

[H2] — Qualquer ponto do plano é uma possível localização.

[H3] — Os serviços a localizar são considerados pontos (tendo por isso área nula).

[H4] — As distâncias a percorrer, entre os serviços a localizar e os já existentes, podem ser adequadamente representadas por uma distância l_p.

[H5] — Os custos de transporte são directamente proporcionais às distâncias l_p utilizadas, com constantes de proporcionalidade independentes dos

valores dessas distâncias.

[H6] — **Os custos fixos podem ser ignorados.**

[H7] — Não há problemas de distribuição associados.

Como se pode ver, com facilidade, estas hipóteses são muito restritivas.

Vejamos, para exemplificar, a hipótese H1: a aproximação da superfície esférica ao plano é, na maior parte dos casos, muito grosseira, servindo apenas para problemas de localização regional, o que poderá levar à rejeição dos modelos planares. No seu livro "Facilities Location" Love, Morris e Wesolowsky (1988), expõem algumas variantes do modelo planar de localização de um só serviço que procuram eliminar alguns dos inconvenientes suscitados pelas hipóteses anteriormente enunciadas. Assim, no que diz respeito a [H1], apresentam um modelo que utiliza distâncias medidas em arcos radianos sobre a superfície esférica de raio unitário. A formalização (2.1.1) passará a ser:

$$\min_{X} \; W(X) = \sum_{j=1}^{n} w_j \; A(X, Y_j) \qquad (2.1.2)$$

em que :

n = número de clientes;

w_j = constantes de proporcionalidade dos custos em relação às distâncias;

X = (x_1, x_2) são a latitude e longitude do novo serviço;

Y_j = (y_{j1}, y_{j2}) é a localização dos cliente (latitude, longitude);

$A(X, Y_j)$ é a menor distância (medida em radianos) sobre a superfície da esfera entre o novo serviço e o cliente j.

Para tal, são feitas algumas adaptações no que se refere às noções de distância, conjuntos convexos e funções convexas na superfície esférica .

A hipótese [H2] é igualmente muito forte porque pode originar soluções do tipo localização no meio dum lago ou num outro local eventualmente ainda mais absurdo! Modelos destes têm pouca probabilidade de interessar os orgãos de decisão, a não ser em casos muito específicos. Nestes casos o que pode esperar-se é que pelo menos na vizinhança haja algum local aceitável.

Quanto à hipótese [H3] ela pode ser restritiva ou não, tudo depende das características do problema; se se trata da localização de uma máquina numa fábrica, por exemplo, a área é importante, mas se se trata da localização de uma fábrica numa região já a hipótese [H3] é irrelevante.

A hipótese [H4] é mais restritiva do que parece, isto porque há uma tendência natural para utilizar a distância usual (Euclideana) que raras vezes aproxima de modo satisfatório a distância percorrida entre clientes e serviços. Na realidade o percurso é, em geral, feito através de ruas, estradas ou outras vias de comunicação cuja medida nada tem a ver com a distância Euclideana.

Mas talvez a maior crítica aos modelos planares resulte da hipótese [H6], isto porque desprezam os custos fixos de preparação e instalação dos serviços. Ora, ao desprezá-los estamos a assumir que eles são os mesmos seja qual for o local de instalação e, desse modo, não influenciam a localização óptima, o que é manifestamente um erro grave, visto que, como sabemos, os custos fixos podem influenciar de forma decisiva a escolha do local, sendo portanto, deste ponto de vista, mais aconselhável um modelo de programação inteira mista.

Quanto às hipóteses [H5] e [H7] podem ser restritivas também; a primeira, porque considera custos proporcionais às distâncias, não tendo em conta, possíveis economias de escala, a segunda, porque não considera a possibilidade de haver interacção entre o novo serviço e outros já instalados, factor que também pode influenciar a localização do novo serviço.

Existem outras variantes de modelos de localização planares: modelos dinâmicos, no caso de haver a possibilidade de o serviço vir a ser realocado no futuro, devido, por exemplo, a alterações de custos de transporte, alterações na procura do serviço, etc.; modelos multi-serviços que surgem quando há necessidade de instalar mais do que um serviço.

Nos modelos multi-serviços, se não existir fluxo entre qualquer par de novos serviços, cada um deles pode ser tratado individualmente, dando origem a tantos modelos de um só serviço quantos os serviços que se pretendem instalar; se os novos serviços são interdependentes têm de ser optimizados simultâneamente. Teremos, então a minimização de uma função de custos para os novos serviços que é soma de duas: a função de custos entre serviços e clientes com constantes de proporcionalidade w_{ij} e a função de custos entre novos serviços com constantes de proporcionalidade v_{ik} . Pretende-se determinar a localização dos m serviços que minimiza o custo total. A formulação será a seguinte:

2.1.2 — PROBLEMA DE LOCALIZAÇÃO PLANAR MULTI-SERVIÇOS

Formulação do Problema

$$\min_{X} \; WMH(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \; l_p\,(X_i\,,\,Y_j) \;+ \sum_{i=1}^{m-1} \sum_{k=i+1}^{m} v_{ik} \; l_p\,(X_i\,,\,X_k) \quad (2.1.3)$$

em que :

m = número de novos serviços a instalar;

n = número de clientes;

w_{ij} = contantes de proporcionalidade dos custos (os custos supõem-se proporcionais às distâncias entre novos serviços e os clientes);

v_{ij} = constantes de proporcionalidade de custos relativas às distâncias entre novos serviços;

$X = (X_1, X_2, \ldots, X_m)$

$X_i = (x_{i1}, x_{i2})$ é a localização do novo serviço i;

$Y_j = (y_{j1}, y_{j2})$ é a localização dos clientes;

$l_p(X_i, Y_j)$ é a distância do serviço i ao cliente j;

$l_p(X_i, X_k)$ é a distância entre o novo serviço i e o novo serviço k.

As distâncias entre novos serviços devem ser consideradas apenas uma vez, o que é traduzido pela representação do correspondente termo na função objectivo.

Se m=1, o problema reduz-se ao problema de localização de um só serviço.

As hipóteses que se aplicam aos modelos de um só serviço continuam a ser válidas neste caso.

Quando se passa de um problema de localização de um só serviço para um problema de localização multi-serviços há uma série de questões que se podem pôr tais como: *qual o número óptimo de serviços a instalar?; quais as áreas de*

serviços que estão associadas ?; qual o impacto das interacções entre serviços sobre as localizações dos novos serviços?

Para um estudo mais detalhado destes modelos, que não constituem o nosso objectivo principal, veja-se, por exemplo, Love, Morris e Wesolowski (1988), Francis e White (1983), P. Hansen, M. Labbé, D.Peeters, Thisse(1987).

2.2 — MODELOS DE LOCALIZAÇÃO EM REDES

Problemas de escolha de localização de serviços ou equipamentos junto de vias de comunicação já existentes tais como linhas férreas, estradas ou outras de modo a minimizar custos de transporte, produção, tempos de atendimento, foram a motivação para o desenvolvimento destes modelos de localização em redes.

Os modelos contínuos envolvem a minimização de uma função de custos, tempos, etc , que está dependente de uma distância l_p. Ora, estas distâncias dão valores aproximados das reais distâncias percorridas numa rede de transportes, seja estrada, linha férrea, linha fluvial, corredor aéreo ou qualquer outra via de comunicação e, sendo assim, *porque não trabalhar directamente com a própria rede*? E foi deste modo que surgiram os modelos de localização em redes alguns dos quais são análogos aos modelos planares, substituindo as *distâncias* l_p por *distâncias na rede*, entendidas estas como comprimentos dos caminhos mais curtos entre dois pontos da rede[1].

Para os modelos de localização em redes, as hipóteses H1, H2, H4, dos modelos planares, deixam de ter sentido; a hipótese H3 ainda se mantém e a hipótese H5 é

substituída pela proporcionalidade dos custos às distâncias na rede. As hipóteses H6 e H7 podem manter-se ou não, conforme a natureza do problema.

Pode dizer-se que foi a partir da publicação do artigo de Hakimi (1964) *"Optimal location of switching centers and the absolute centers and medians of a graph"* que se iniciou o estudo dos modelos de localização em redes.

Neste grupo de modelos estão incluídos os de p-mediana, p-centro e cobertura.

Na exposição que se segue designaremos por $\Re$ a rede e por V o conjunto de vértices da rede , i.e., $V = \{ v_1 , v_2 , \ldots , v_j , \ldots , v_n \}$.

2.2.1 — PROBLEMA P-MEDIANA

O problema *p-mediana* consiste na localização de *p* serviços numa rede com *n* comunidades a servir de modo a que a soma das distâncias ponderadas entre comunidades e serviços seja mínima.

Existem inúmeras situações concretas em que tais problemas se põem. Podemos referir, para exemplificar, a localização de escolas, de centrais de distribuição de tráfego telefónico, de estações de correio, etc.

Para a formulação do problema da p-mediana, comecemos por considerar o problema 1-mediana. Podemos retomar a formulação (2.1.1) do problema contínuo,

---

[1] *Para uma definição mais rigorosa de rede e alguns conceitos relacionados tais como distância na rede e propriedades, veja-se, por exemplo, Berge* (1966) *e Hansen, Labbé, Peeters, Thisse* (1987).

substituindo as distâncias l_p por distâncias na rede. As comunidades correspondem aos vértices ($v_1, v_2, \ldots, v_n$) da rede e as ligações entre elas aos arcos. A cada vértice associa-se um peso w_j , não negativo, que representa o peso relativo da respectiva comunidade (população, por exemplo). As distâncias são os comprimentos dos arcos que ligam as comunidades. O problema consiste na determinação de um ponto , s^*, da rede que minimize a soma das distâncias ponderadas às comunidades. Tem-se então:

PROBLEMA 1-MEDIANA

$$\min_{s \in \mathfrak{R}} W(s) = \sum_{j=1}^{n} w_j \; d(s, v_j) \qquad (2.2.1)$$

A solução é chamada a *mediana absoluta* e o conjunto de medianas é chamado *conjunto mediana*.

Resulta da definição que a mediana pode ser qualquer ponto da rede, i. e. um vértice ou um ponto interior de um arco. Contudo, Hakimi (1964) demonstrou que existe sempre uma solução óptima para este problema localizada num vértice da rede. Devido a esta propriedade de optimalidade nos vértices resulta que para determinar a mediana absoluta basta considerar localizações em vértices.

Hakimi demonstrou também que a solução óptima pode ser obtida em tempo polinomial, calculando a soma das distâncias ponderadas para cada vértice, v_i , e seleccionando o vértice de menor soma.

Se em vez de um serviço pretendermos instalar p para servir n comunidades de modo a que a soma das distâncias ponderadas seja mínima

teremos, naturalmente, a generalização do problema anterior que se chama *p-mediana*.

Formulação do Problema

$$\min_{S} W(S) = \sum_{j=1}^{n} w_j \; d(v_j, S) \qquad (2.2.2)$$

s.a

$$|S| = p \;, \; S \subseteq \Re \;.$$

Em que S é um conjunto de pontos da rede, vértices ou pontos interiores aos arcos e em que $d(v_j, S) = \min \{ d(v_j, s), s \in S \}$.

O conjunto S^* de p pontos que minimiza a função W(S) é chamado p-mediana da rede. O teorema de Hakimi continua a verificar-se, neste caso, i. e. , a solução óptima para este problema é formada por um conjunto de vértices da rede. A formulação combinatória do problema p-mediana pode então apresentar-se do modo seguinte:

Formulação Combinatória

$$\min_{S \subseteq V} W(S) = \sum_{j=1}^{n} \min_{i \in S} d_{ij} \qquad (2.2.3)$$

s.a

$$|S| = p$$

Em que os d_{ij} representam as distâncias ponderadas entre os vértices i e j.

Mais adiante faremos referência a uma outra formulação do problema p-mediana, em termos de programação inteira.

2.2.2 — PROBLEMA P - CENTRO

O problema *p - centro* consiste na determinação de um conjunto, S^*, de p pontos da rede, $\Re$, de modo a que o máximo das distâncias ponderadas dos vértices da rede a esse conjunto seja mínima.

Este problema põe-se em algumas situações concretas, como por exemplo, instalação de serviços de emergência, bombeiros, hospitais, etc, em que o critério de optimalidade pode ser a minimização do tempo máximo de resposta ou a minimização da distância máxima entre comunidades e serviços, i.e., a optimização do pior caso. As localizações resultantes são os centros.

Os problemas p-centro e p-mediana estão relacionados, diferem apenas na função objectivo e foram ambos introduzidos por Hakimi (1964). Os métodos para resolver os dois problemas são diferentes.

O *problema 1-centro* ou *problema minimax* consiste na determinação de um ponto s^* da rede, $\Re$, que minimiza a máxima distância às comunidades. A sua formulação é a seguinte:

PROBLEMA 1-CENTRO (MINIMAX)

$$\min_{s \in \Re} G(s) = \max_{j=1,\ldots,n} w_j \; d(s, v_j) \qquad (2.2.4)$$

Em que $w_j > 0$, representa o peso da comunidade j.

Ao ponto s^* chama-se *1-centro*.

A formulação do *problema p - centro* será a seguinte:

Formulação do Problema

$$\min_{S} G(S) = \max_{j} w_j \, d(v_j, S) \qquad (2.2.5)$$

s.a

$$|S| = p, \quad S \subseteq \Re.$$

O resultado estabelecido por Hakimi para o problema p-mediana no que se refere à localização da solução óptima num conjunto de vértices da rede não se verifica para o problema p-centro. Podem obter-se melhores soluções para o problema p-centro, se forem permitidas localizações nos arcos da rede. Este problema, em que se suprime a restrição de que os centros se localizem em vértices da rede é chamado *p-centro absoluto*. Ainda pode ser considerada uma versão mais geral do p-centro em que a procura pode também estar localizada em qualquer ponto da rede , *p-centro geral*.

Hakimi (1964) propôs um algoritmo para resolver por um processo gráfico o problema 1-centro absoluto que não é generalizável a problemas p-centro absoluto com p>1. Muitos algoritmos têm sido propostos para este problema, vejam-se, por exemplo, Singer (1968) que propôs um método heurístico para a determinação de p - centros absolutos, Minieka (1970) propôs resolvê-lo por uma

sucessão de problemas de cobertura, Christofides (1975), Kariv e Hakimi (1979). Vejam-se ainda os artigos de "survey" publicados por J.Halpern e O. Maimon (1982) e B.Tansel, L. Francis, T. Lowe (1983) .

2.2.3 — PROBLEMA DA COBERTURA

O problema de localização de serviços de emergência pode, em certos casos, ser formulado como um problema de cobertura. Isto acontece, por exemplo, quando o que se pretende é determinar o conjunto de serviços de cardinalidade mínima de modo a que a distância máxima entre comunidades e serviços mais próximos não exceda determinado valor, préviamente fixado. Se for posto deste modo, teremos um problema que pode ser formulado como um *Problema de Cobertura*.

A solução deste problema deverá ser formada por um conjunto de serviços em que toda a comunidade seja coberta por algum serviço a uma distância que não ultrapasse o valor préviamente fixado.

A sua formulação combinatória do problema de cobertura será a seguinte:

Formulação Combinatória

$$\text{Min } |S| \tag{2.2.6}$$

s.a

$$w_j \, d(v_j, S) \leq \overline{d}, \quad j = 1, \ldots, n. \tag{2.2.7}$$

em que:

$|S|$ = cardinal de S;

$\bar{d}$ = distância máxima entre comunidades e serviços;

w_j = peso da comunidade j.

Para cada valor do parâmetro $\bar{d}$ teremos um problema de cobertura diferente.

A sua formulação em termos de programação inteira seria:

$$\text{Min} \sum_{i \in I} y_i \tag{2.2.8}$$

s.a

$$\sum_{i \in I} a_{ji} y_i \geq 1, \quad j \in J, \tag{2.2.9}$$

$$y_i \in (0,1), \; i \in I. \tag{2.2.10}$$

em que:

y_i = 1, se o serviço i for aberto

= 0, caso contrário.

$\bar{d}$ = distância máxima entre comunidades e serviços;

a_{ji} = 1, se a comunidade j estiver a uma distância de i não superior a $\bar{d}$;

= 0, caso contrário.

As restrições (2.2.9) garantem a cobertura de qualquer comunidade j por algum serviço localizado a uma distância não superior a $\bar{d}$.

Este problema designado por *Problema de Cobertura Mínima* foi estudado

por vários autores de entre os quais referimos, Toregas e ReVelle (1971) e Toregas, Swain, ReVelle e Bergman (1971), Khumawala (1972), A. Neebe (1988). A maior parte dos investigadores tenta resolver o problema fixando um valor tolerável para a distância máxima; Neebe no seu artigo *"A Procedure for Locating Emergency-Service Facilities for All Possible Response Distances"* parte de um valor pequeno para a distância máxima $\overline{d}$, valor para o qual todos os serviços têm que estar abertos e vai aumentando esse valor até um limite em que basta um serviço para cobrir todas as comunidades. Procura, assim, determinar o número de serviços que seriam necessários para todos os valores possíveis de $\overline{d}$, desde o menor valor da distância até ao maior, utilizando, para cada problema uma versão modificada do algoritmo de Lemke, Salkin e Spielberg (1971).

O problema pode ainda pôr-se como um *Problema de Cobertura de Custo Mínimo*, caso em que a formulação difere da anterior apenas na função objectivo que neste caso seria substituida por:

$$\text{Min} \sum_{i \in I} f_i \, y_i \qquad (2.2.8')$$

com os f_i a representarem os custos fixos de instalação dos serviços.

Pode ainda acontecer que por alguma razão, por exemplo restrições orçamentais, não seja possível instalar todos os serviços que fazem parte da cobertura. Neste caso, pode tomar-se uma de duas opções, ou aumentar o orçamento ou relaxar as restrições de cobertura de todos os clientes. Quando esta última é escolhida, o que se procura é maximizar a população que pode ser atendida a partir de um número K de serviços, garantindo que a distância ao

serviço que a atende não é superior ao valor $\bar{d}$ previamente fixado. Tem-se, então, um problema chamado *Problema de Cobertura Máxima*, que é formalmente equivalente ao K-mediana, Church e ReVelle (1976).

Neste problema o objectivo é maximizar o número de pessoas atendidas e a sua formulação será a seguinte:

PROBLEMA DE COBERTURA MÁXIMA

Formulação do Problema

$$\underset{y\,,\,X}{\text{Max}}\; F(y\,,\,X) = \text{Max} \sum_{i\in I} \sum_{j\in J} d_{ij}\, x_{ij} \qquad (2.2.12)$$

s.a

$$\sum_{j\in J} x_{ij} = 1, \quad i\in I, \qquad (2.2.13)$$

$$y_j - x_{ij} \geq 0, \quad i\in I, \quad j\in J, \qquad (2.2.14)$$

$$x_{ij} \geq 0\,, \quad i\in I, \quad j\in J, \qquad (2.2.15)$$

$$y_j \in \{0\,,\,1\}, \quad j\in J, \qquad (2.2.16)$$

$$\sum_{j\in J} y_j = K\,. \qquad (2.2.17)$$

em que :

J = (1, 2, ..., m), conjunto de localizações possíveis para os serviços;

I = (1, 2, ..., n), conjunto de comunidades;

K = número de serviços que podem abrir-se;

y_j = 1, se o serviço j for aberto

= 0, caso contrário.

$\bar{d}$ = distância máxima de serviço;

d_i = número de clientes da comunidade i que estão a uma distância não superior a $\bar{d}$ de algum dos serviços possíveis;

d_{ij} = d_i, se algum cliente da comunidade i estiver a uma distância de j não superior a $\bar{d}$;

= 0, caso contrário.

x_{ij} = proporção de clientes da comunidade i que são atendidos a partir do serviço j.

2.3 — MODELOS DE LOCALIZAÇÃO DISCRETOS

Os modelos planares e os modelos em redes têm usualmente três limitações: qualquer ponto do plano ou da rede é possível candidato à localização do novo serviço; os parâmetros de custos estão associados aos serviços que vão ser localizados e não aos locais onde vão ser instalados; as suas funções de custos, podem não incluir custos fixos. Como estas limitações são inaceitáveis para grande número de casos surgem, então, os modelos a que chamamos "discretos" de acordo com a classificação que tem vindo a ser adoptada na literatura, vejam-se R.Francis, L.McGinnis e J.White (1983), P. Hansen,

M. Labbé, D. Peeters, J. Thisse (1987).

Nestes modelos há um número finito de locais onde podem ser instalados os serviços, os quais podem ser gerados por modelos planares , também chamados *"site-generating models"*, e/ou por soluções de compromisso de múltiplos objectivos, ou ainda por outros critérios. As questões a que se pretende que respondam são do tipo:

Quantos serviços (equipamentos) devem ser instalados? Quais? Com que dimensão? Que serviços servem quais clientes ?

Nestes modelos, os pontos não são deslocados até se encontrar a melhor localização, como acontecia nos modelos de localização em redes; há *m* possíveis locais préviamente definidos, de entre os quais vão ser seleccionados os que, satisfazendo as restrições de procura, de dimensão, de número, etc., minimizam a soma de custos de produção/distribuição e custos fixos. Por esta razão, são também chamados *" site-selecting location-allocation models"*, embora dentro desta designação caibam também os modelos de cobertura nos quais se pretende determinar o menor número de serviços que é necessário para satisfazer determinados níveis de procura.

Como já referimos, são tão variadas as aplicações em que estão envolvidos estes modelos, que têm originado uma não menos variada gama de formulações e algoritmos para resolver os problemas que lhes estão associados.

Iremos primeiramente dar uma ideia sobre algumas dessas formulações e em seguida debruçar-nos-emos sobre o *"Problema de Localização Simples"* que é simultâneamente o de formulação mais simples e mais antiga, datando do início dos anos 60. O grande interesse que este problema tem suscitado, segundo

Krarup e Pruzan (1983), deve-se, não só à transparência da sua estrutura, mas também ao contributo que tem dado para a resolução de outros problemas de planeamento mais complexos. Por outro lado, a sua estrutura é, em certo sentido, bastante abrangente; isto porque o número de serviços a abrir não é préviamente fixado, a capacidade é ilimitada, a estrutura de custos de distribuição é linear e, deste modo, permite modificações que podem levar a formulações mais realistas originando outros problemas.

Assim o Problema de Localização Simples, SPL , que é um problema:

Minimização (custos fixos + custos lineares)
discreto
estático
um só-serviço
determinístico

pode ser modificado de modo a transformar-se num problema de custos não lineares, dinâmico, múltiplos serviços, procura estocástica, etc.

2.3.1 — PROBLEMA DE LOCALIZAÇÃO SEM RESTRIÇÕES DE CAPACIDADE

O problema de localização simples consiste na determinação de um conjunto de locais em que devem ser instalados serviços para atender um conjunto de clientes de modo a que todos os clientes sejam atendidos a partir de um e um só serviço (fábrica, armazém, etc) ao menor custo possível. Como já referimos anteriormente este problema é tratado sob as mais variadas designações, consoante as aplicações. Optámos pela terminologia seguinte:

— ao ***problema***, chamaremos → ***Problema de Localização Simples*** - SPL;

— às ***origens***, chamaremos → ***serviços*** - localidade i ;

— aos ***destinos***, chamaremos → ***clientes*** - localidade j .

Supõe-se que os serviços têm capacidade ilimitada, de modo que, em princípio, qualquer serviço pode satisfazer toda a procura. Consideraremos como objectivo a minimização de custos totais, custos fixos e custos de atendimento. Diremos que um serviço i é *"aberto"* ou *"fechado"*, para significar a decisão de o *"instalar"* ou *"não instalar"* na localidade i.

A formulação do SPL em termos de programação inteira :

PROBLEMA DE LOCALIZAÇÃO SIMPLES

Formulação do Problema

$$(\text{SPL}) \qquad \text{Min} \sum_{i \in I} f_i \, y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} \, x_{ij} \qquad (2.3.1)$$

s.a

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J, \tag{2.3.2}$$

$$y_i - x_{ij} \geq 0, \quad i \in I, \ j \in J, \tag{2.3.3}$$

$$x_{ij} \in \{0, 1\}, \quad i \in I, \ j \in J, \tag{2.3.4}$$

$$y_i \in \{0, 1\}, \quad i \in I, \tag{2.3.5}$$

em que:

y_i = 1, se o serviço i for aberto

= 0, caso contrário;

x_{ij} = 1, se o cliente j for atendido no serviço i

= 0, caso contrário;

f_i = custo fixo de instalação do serviço i;

c_{ij} = custos de atender o cliente j no serviço i ;

I = (1,2, ... ,m); conjunto de locais em que potencialmente poderão vir a ser instalados os serviços;

J = (1,2, ... ,n); conjunto de clientes.

Os custos c_{ij} de satisfazer a procura do cliente j a partir do serviço i são função dos custos de produção, p_j , dos custos de transporte de i para j , t_{ij} , da procura do cliente j , d_j , assim, por exemplo: $c_{ij} = d_j (p_j + t_{ij})$.

As restrições (2.3.2) asseguram que a procura de cada cliente é

completamente satisfeita; as restrições (2.3.3) garantem que a procura dos clientes é satisfeita a partir de serviços abertos.

As restrições (2.3.4) de integralidade das variáveis x_{ij} podem ser substituidas por (2.3. 4') :

$$x_{ij} \geq 0 \ , \ i \in I, \ j \in J, \qquad (2.3.4')$$

sem perda de generalidade, visto que as *variáveis estratégicas* são as variáveis y_i. Uma vez fixado o vector de variáveis binárias (y_i) é fácil encontrar uma solução inteira para o SPL (2.3.1) - (2.3.5), bastando para tal afectar cada cliente, j, ao serviço i, "aberto", de custo mínimo para j. Pelo que acabamos de dizer é usual encontrar a formulação do SPL como problema de programação inteira mista (2.3.1) — (2.3.4') — (2.3.5).

2.3.2 — PROBLEMA DE LOCALIZAÇÃO SEM RESTRIÇÕES DE CAPACIDADE COM VÁRIOS NÍVEIS

Este problema surge quando num sistema de distribuição existe uma hierarquia de equipamentos entre a origem e o destinatário. É exemplo deste tipo de situação a necessidade de localizar simultâneamente fábricas e armazéns e/ou armazéns de diversas dimensões para estabelecer o fluxo entre fábricas e retalhistas.

Passaremos a apresentar a formulação matemática do problema de localização a dois níveis de que é exemplo a localização de fábricas e armazéns num sistema de produção, caso que foi apresentado por Kaufman, Eede e Hansen (1977) num artigo em que desenvolveram um algoritmo que utiliza um método

"branch-and-bound" para resolver o problema de distribuição a dois níveis. A generalização desta formulação a mais níveis intermédios faz-se de igual modo. Tcha e Lee em (1984) fizeram a generalização deste modelo considerando a sua formulação no caso de haver vários níveis de armazéns ou serviços intermédios entre a origem (fábrica) e o destinatário (cliente). Neste problema o objectivo é seleccionar, para cada nível, um conjunto de fábricas/armazéns a abrir dentro de um conjunto de potenciais candidatos em cada nível, de modo a que, os custos totais de distribuição e custos fixos de instalação sejam mínimos. Quando, alguma das mercadorias não precisa de utilizar todos os níveis intermédios, criam-se variáveis artificiais " dummy " para as introduzir no modelo. Neste artigo Tcha e Lee desenvolvem um algoritmo "branch-and-bound" que é baseado no procedimento dual ascendente de Erlenkotter, Bilde e Krarup com algumas modificações. Optámos pela formulação a dois níveis, apresentada por Kaufman, Eede e Hansen, porque é mais simples, sem perda de generalidade.

Formulação do Problema

$$\text{(SPLM)} \qquad \text{Min} \sum_{i\in I} f_i y_i + \sum_{j\in J} g_j z_j + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} o_{ijk}\ x_{ijk} \qquad (2.3.6)$$

s.a

$$\sum_{i\in I}\sum_{j\in J} x_{ijk} = 1, \quad k \in K, \qquad (2.3.7)$$

$$y_i - \sum_{j\in J} x_{ijk} \geq 0, \quad i\in I, \ k\in K, \qquad (2.3.8)$$

$$z_j - \sum_{i \in I} x_{ijk} \geq 0, \quad j \in J, \ k \in K, \tag{2.3.9}$$

$$z_i \geq y_i, \quad i \in I, \tag{2.3.10}$$

$$x_{ijk} \geq 0, \quad i \in I, \ j \in J, \ k \in K, \tag{2.3.11}$$

$$y_i, z_j \in (0, 1), \quad i \in I, \ j \in J. \tag{2.3.12}$$

em que:

y_i = 1, se a fábrica /serviço for aberto

= 0, caso contrário.

z_j = 1, se o armazém /serviço for aberto

= 0, caso contrário.

x_{ijk} = percentagem de procura /clientes satisfeita/ atendidos a partir da fábrica/serviço i através do armazém/ serviço j.

f_i = custo fixo de instalação da fábrica /serviço i.

g_j = custo fixo de instalação do armazém /serviço j.

c_{ijk} = custos de satisfazer a procura do cliente k a partir da fábrica /serviço i através do armazém j.

I = (1,2, ... ,l), conjunto de locais em que poderão vir a ser instaladas as fábricas /serviços.

J = (1,2, ... ,m), conjunto de locais em que poderão vir a ser instalados os armazéns/ serviços.

K = (1,2, ... ,n), conjunto de clientes.

Os custos c_{ijk} incluem custos de matérias primas e respectivo transporte para a fábrica i, custos de produção em i, custos de tranporte de i para j,

custos de armazenagem em j e custos de transporte de j para o cliente k.

As restrições (2.3.7) asseguram que a procura de cada cliente é completamente satisfeita; as restrições (2.3.8) e (2.3.9) garantem que a procura do cliente k só pode ser satisfeita a partir da fábrica i e do armazém j se estes forem abertos. As restrições (2.3.10) garantem que para cada fábrica que abre, abre um armazém ao lado, condição que foi imposta na formulação do problema feita por Kaufman e al.. No entanto, se esta condição não for exigida, as restrições (2.3.10) podem óbviamente suprimir-se. As restrições (2.3.11) garantem que os níveis de procura do cliente k a partir de i e através de j são não negativos.

Nos modelos anteriores estivemos a considerar distribuição de um só-produto (ou prestação de um só serviço); no entanto, em muitas situações podem pôr-se estes mesmos problemas, mas com vários produtos dando origem a extensões destes genéricamente designados por multi-produtos/multi-serviços e cuja apresentação passaremos a fazer no parágrafo seguinte.

2.3.3 — PROBLEMA DE LOCALIZAÇÃO DE MULTI-SERVIÇOS SEM RESTRIÇÕES DE CAPACIDADE

Warszawski e Peer (1973) introduziram o problema de localização multi-serviços. A motivação para o problema resultou da necessidade de resolver um problema de engenharia civil que envolvia um grande projecto de construção. Nesse projecto havia necessidade de definir os locais em que deviam situar-se três unidades de fabrico de produtos fundamentais: massa de cimento, blocos e aço. Cada unidade produzia apenas um destes produtos e deveria ser localizada

de tal modo que cada um dos 38 destinos candidatos pudessem ser servidos de forma eficiente. Este modelo tem um particular interesse para situações de fábricas que fazem as vendas dos seus produtos directamente ao público e em que a armazenagem é práticamente inexistente.

Formulação do problema

$$(\text{MPL}) \qquad \text{Min} \sum_{i\in I}\sum_{p\in P} f_{ip}\, y_{ip} + \sum_{i\in I}\sum_{j\in J}\sum_{p\in P} c_{ijp}\, x_{ijp} \qquad (2.3.13)$$

s.a

$$\sum_{i\in I} x_{ijp} = 1, \quad j\in J,\ p \in P, \qquad (2.3.14)$$

$$\sum_{p\in P} y_{ip} \leq 1, \quad i\in I, \qquad (2.3.15)$$

$$\sum_{j\in J} x_{ijp} \leq y_{ip}, \quad i\in I,\ p\in P, \qquad (2.3.16)$$

$$x_{ijp} \geq 0, \quad i\in I,\ j\in J,\ p\in P \qquad (2.3.17)$$

$$y_{ip} \in \{0, 1\}, \quad i\in I,\ p\in P. \qquad (2.3.18)$$

em que:

y_{ip} = 1, se a fábrica i é aberta para produzir o produto p
= 0, caso contrário;

x_{ijp} = percentagem de procura do cliente j para o produto p satisfeita a partir da fábrica i;

f_{ip} = custo fixo de instalação da fábrica i a produzir o produto p;

c_{ijp} = custos de produção e distribuição para satisfazer a procura dos clientes j, do produto p, a partir da fábrica i.

I = (1,2, ... , m) conjunto de locais em que poderão vir a ser instaladas as fábricas ou serviços.

J = (1,2, ... , n), conjunto de clientes.

P = (1,2, ... , k), conjunto de produtos ou serviços.

Se o número de produtos, k, for igual a 1 o problema reduz-se ao problema de localização simples, SPL.

Warszawski (1973), apresentou dois métodos para resolver o MLP : um método "branch-and-bound" e um método heurístico. Para o método "branch-and-bound" desenvolveu dois métodos para determinar minorantes em cada nodo da árvore de ramificação. Um deles permitia obter melhores minorantes, mas mais lento; o outro, mais rápido, mas a originar minorantes mais fracos. B. Khumawala e W. Neebe (1978) apresentaram duas sugestões que permitem melhorar os minorantes obtidos pelo segundo método. Como o método "branch-and-bound" ocupa muito tempo de computador em problemas de grande dimensão, Warszawski propôs um método heurístico, mais rápido, em que eram primeiramente resolvidos k problemas de localização simples, um para cada produto; se todas as restrições (2.3.15),

$$\sum_{p \in P} y_{ip} \leq 1 \ , \forall i,$$

fossem satisfeitas então a solução encontrada seria a solução óptima do MPL.

Se para algum r se tivesse:

$$\sum_{p \in P} y_{rp} > 1$$

então, isto significava que a fábrica localizada em r estava a produzir mais do que um produto, e procedia-se a uma realocação das fábricas que estivessem nessas condições de modo a que o acréscimo sofrido pelo minorante que se obtém relaxando as restrições de não interferência (2.3.15), fosse mínimo. Este processo repetia-se até que todas aquelas restrições fossem satisfeitas, isto é, até que deixasse de haver situações de conflito.

Karkazis e Boffey (1981) desenvolveram dois algoritmos baseados no dual para resolver o MLP. Estes algoritmos a que chamaram MultiBKE e HILLCLIMB resultaram, da possibilidade de reduzir a resolução do MLP à resolução de subproblemas SPL. O MultiBKE resultou da aplicação do algoritmo dual ascendente de Bilde , Krarup e Erlenkotter e o HILLCLIMB da aplicação de optimização subgradiente à resolução dos subproblemas SPL. São ali apresentados resultados computacionais e é feita a comparação com os resultados obtidos por Warszawski, utilizando para tal os mesmos quatro exemplos, concluindo pela maior eficiência de qualquer dos seus algoritmos.

2.3.4 — PROBLEMA DE LOCALIZAÇÃO DINÂMICA SEM RESTRIÇÕES DE CAPACIDADE

O problema de localização dinâmica surge quando as decisões sobre a localização podem depender do tempo. Warszawski no seu artigo (1973) incluiu

algumas soluções para a versão dinâmica do SPL. Esta versão foi também sugerida pelo seu problema concreto ligado à construção civil. A necessidade de mudar as localizações das unidades de produção de massa de cimento, aço, etc., de acordo com as solicitações, levaram à formulação do problema dinâmico que a seguir apresentamos:

Formulação do problema

$$\text{(DPL)} \qquad \text{Min} \sum_{i\in I}\sum_{j\in J}\sum_{t\in T} c_{ijt}\, x_{ijt} + \sum_{i\in I}\sum_{t\in T} f_{it}\, y_{it} +$$

$$+ \sum_{i\in I} g_{i1}\, y_{i1} + \sum_{i\in I}\sum_{t\in T'} g_{it}\, y_{it}\,(1 - y_{i\,t-1}) \qquad (2.3.19)$$

s.a

$$\sum_{i\in I} x_{ijt} = 1, \quad j\in J,\ t\in T, \qquad (2.3.20)$$

$$\sum_{j\in J} x_{ijt} \le m\, y_{it}, \quad i\in I,\ t\in T, \qquad (2.3.21)$$

$$x_{ijt} \ge 0, \quad i\in I,\ j\in J,\ t\in T, \qquad (2.3.22)$$

$$y_{it} \in (0, 1), \quad i\in I,\ t\in T. \qquad (2.3.23)$$

em que:

y_{it} = 1, se a fábrica i é/está instalada no instante t

= 0, caso contrário.

x_{ijt} = percentagem de procura do cliente j que no instante t é satisfeita pela fábrica i ;

f_{it} = custos fixos da fábrica *i* no instante t que são

independentes de realocações (como por exemplo, custos de capital e de manutenção);

g_{it} = custos fixos de instalação/realocação da fábrica i no instante t;

c_{ijt} = custos de produção e distribuição para satisfazer a procura dos clientes j, no instante t, a partir da fábrica i;

I = $(1,2, \ldots, n)$ conjunto de locais em que poderão vir a ser instaladas as fábricas ou serviços;

J = $(1,2, \ldots, m)$, conjunto de clientes;

T = $(1,2, \ldots, k)$, instantes em que poderão vir a ser instaladas as fábricas ou serviços;

T' = $T - (1)$.

Como pode notar-se existe uma grande semelhança entre a formulação do MPL e do DPL, podendo o último ser praticamente obtido do primeiro pela substituição do índice de produtos (mercadorias) pelo índice tempo. A principal diferença entre os dois modelos está nas componentes de custos. No modelo dinâmico, há para além dos custos fixos de instalação, os custos de realocação quando esta tem lugar de um instante para o seguinte. Warszawski, testou dois métodos para resolver este problema, um baseado na programação dinâmica e outro baseado no maior rendimento marginal.

Erlenkotter e Van Roy (1982), apresentaram uma generalização deste modelo, introduzindo um novo parâmetro para separar, no tempo, as decisões de localização das decisões de procura. A formulação do problema por eles apresentada é a seguinte:

Formulação do Problema

$$(GDPL) \quad \text{Min} \sum_{w\in W}\sum_{i\in I}\sum_{j\in J}\sum_{t\in T} c_{ijt}\ x_{ijwt} + \sum_{t\in T}\sum_{i\in I} f_{it}\ y_{it} \qquad (2.3.24)$$

s.a

$$\sum_{i\in I}\sum_{t\in T} x_{ijwt} = 1, \quad j\in J,\ w\in W, \qquad (2.3.25)$$

$$x_{ijwt} \leq y_{it}, \quad i\in I,\ j\in J,\ w\in W,\ t\in T, \qquad (2.3.26)$$

$$x_{ijwt} \geq 0, \quad i\in I,\ j\in J,\ w\in W,\ t\in T, \qquad (2.3.27)$$

$$y_{it} \in \{0, 1\}, \quad i\in I,\ t\in T. \qquad (2.3.28)$$

em que:

y_{it} = 1, se a fábrica i é instalada no instante t
= 0, caso contrário.

x_{ijwt} = percentagem de procura do cliente j no instante w que é satisfeita pela fábrica i no instante t;

f_{it} = custo fixo de instalação da fábrica i no instante t;

c_{ijwt} = custos de produção e distribuição para satisfazer a procura do cliente j, feita no instante w, a partir da fábrica i no instante t.

I = $(1,2, \ldots, n)$ conjunto de locais em que poderão vir a ser instaladas as fábricas ou serviços;

J = (1,2, ... , m) , conjunto de clientes;

W = (1,2, ... , l), conjunto de instantes em que poderão vir a ser feitas encomendas;

T = (1,2, ... , k), conjunto de instantes em que poderão vir a ser abertas ou fechadas as fábricas ou serviços.

Num determinado instante, t, pode considerar-se $I = I_O \cup I_C$, em que I_O , representa o conjunto de locais em que os serviços podem vir a ser abertos e I_C, o conjunto de locais em que os serviços podem ser fechados. Assim se $y_{it} = 1$, para $i \in I_O$, significa que o serviço foi aberto até ao instante t ; $y_{it} = 1$, para $i \in I_O$, significa que no instante t se decide abrir o serviço i .

Para assegurar que um serviço não atende clientes antes de ser aberto ou depois de ter fechado, consideram-se os custos $c_{ijwt} = M$, para $w < t$ e $i \in I_O$ e para $w > t$ e $i \in I_C$, com M um número positivo suficientemente grande.

Erlenkotter e Van Roy (1982) implementaram um algoritmo "branch-and-bound" a que chamaram DYNALOC, em que está incorporada uma versão especializada do método dual ascendente de Bilde, Krarup e Erlenkotter para a resolução do problema estático de localização simples, SPL. Neste artigo, mostra-se que, o GDPL pode ser considerado um problema estático, SPL, em que as "pseudo" localizações são todas as combinações de i com t, (i,t), em que i pode ser aberta ou fechada e os "pseudo" clientes são todas as combinações, (j,w), de clientes j com períodos w. Apesar disso, não utilizaram métodos de resolução do SPL nem mesmo o DUALOC, algoritmo "branch-and-bound" para resolver o SPL, que fora implementado por Erlenkotter (1978) e se mostrou bastante eficiente. As razões que levaram à implementação de um novo algoritmo são ali apresentadas

e têm a ver com o espaço de memória que seria necessário para este problema e com algumas particularidades deste problema que não são consideradas no DUALOC.

Neste artigo, além de apresentarem o método, DYNALOC para resolver o GDPL são ainda apresentados resultados computacionais obtidos pelo algoritmo e comparados com os que haviam sido obtidos por Roodman e Schwartz (1975) para problemas dinâmicos, com os obtidos por Erlenkotter (1978) na utilização do *ajustamento do dual* versus *ajustamento do primal-dual* aí implementado, e ainda com um conjunto de problemas dinâmicos construidos a partir dos problemas estáticos de Kuehn-Hamburger (1963). Tendo concluido que o DYNALOC é mais rápido em qualquer daqueles casos. Por último são feitas algumas extensões deste algoritmo nomeadamente a problemas dinâmicos com restrições de capacidade.

2.3.5 — PROBLEMA DE LOCALIZAÇÃO COM RESTRIÇÕES DE CAPACIDADE

O problema de localização com restrições de capacidade surge, como é evidente, quando há limitações de alguma natureza, sejam de produção, espaço ou outras. Para este problema a formulação em programação mista é a seguinte:

Formulação do Problema

$$\text{(CPL)} \qquad \text{Min} \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \qquad (2.3.29)$$

s.a

$$\sum_{i \in I} x_{ij} = d_j , \quad j \in J, \tag{2.3.30}$$

$$\sum_{j \in J} x_{ij} \leq s_i y_i , \quad i \in I, \tag{2.3.31}$$

$$x_{ij} \geq 0, \quad i \in I, \ j \in J, \tag{2.3.32}$$

$$y_i \in \{0 , 1\}, \quad i \in I. \tag{2.3.33}$$

em que:

y_i = 1, se a fábrica (serviço) i for aberto
= 0, caso contrário;

x_{ij} = quantidade de procura do cliente j satisfeita a partir da fábrica i ;

f_i = custo fixo de instalação da fábrica (serviço) i.

c_{ij} = custos unitários de produção e distribuição para satisfazer a procura do cliente j a partir da fábrica i;

s_i = um limite superior da capacidade da fábrica (serviço) i;

d_j = a procura do cliente j;

I = conjunto de locais em que poderão vir a ser instaladas as fábricas ou serviços;

J = conjunto de clientes.

As restrições (2.3.30) asseguram que a procura de cada cliente é completamente satisfeita; as restrições (2.3.31) garantem que a procura dos clientes é satisfeita a partir de fábricas abertas.

Note-se que em muitas situações pode haver necessidade de incluir no

grupo de restrições (2.3.31) outras de tipo $\sum_{j\in J} x_{ij} \geq \underline{s}_i \, y_i$, $i\in I$, isto é, incluir restrições que contemplem situações do tipo: *só interessa abrir uma fábrica ou serviço, desde que os níveis de produção (procura) sejam superiores a certos valores*. Nesse caso, bastava fazer uma ligeira modificação à formulação anterior. Transformar as restrições (2.3.31), agora alargadas às restrições $\geq$, em restrições de igualdade, introduzindo variáveis de desvio, superiormente limitadas pelas diferenças entre os níveis máximos e mínimos de oferta, s_i e $\underline{s}_i$, respectivamente; adicionar as restrições $x_{ij} \leq u_{ij}$, em que $u_{ij} = s_i - \underline{s}_i$, se j for destino em que há desvio e $u_{ij} = \infty$, se o desvio for nulo. Os métodos mais utilizados na resolução destes problemas são de tipo "branch-and-bound". Beasley (1988) apresenta uma formulação do problema que considera mais geral, por incluir restrições do tipo das que acabámos de referir e ainda restrições no número de serviços a abrir. Apresenta ainda um algoritmo para resolver problemas CPL de grande dimensão. O algoritmo permite determinar minorantes da solução óptima, utilizando uma relaxação Lagrangeana da formulação inteira mista do problema que goza da *propriedade de integralidade*, Geoffrion(1974). Numa tentativa de melhorar os minorantes obtidos pelo dual lagrangeano, Beasley desenvolveu um processo iterativo em que se procura fazer um ajustamento de multiplicadores, utilizando optimização subgradiente, e se resolve o dual lagrageano para os multiplicadores ajustados. Para completar a resolução do CPL implementou um "branch-and-bound" em que utilizou como majorante o valor de uma solução admissível, obtida de uma forma simples e como minorantes os que ali calculou via relaxação Lagrangeana e optimização subgradiente. O algoritmo foi testado para problemas com 500 armazéns e 1000 clientes. Van Roy (1986) desenvolveu um algoritmo que é mais rápido do que o apresentado por Beasley para problemas de pequena dimensão.

2.3.6 – PROBLEMA ESTOCÁSTICO DE LOCALIZAÇÃO COM RESTRIÇÕES DE CAPACIDADE

Este problema surge quando as procuras são variáveis aleatórias. A sua formulação para o caso um só produto/serviço é a seguinte:

Formulação do Problema

$$(SCPL)\quad \text{Min} \sum_{i\in I} f_i y_i + \sum_{i\in I\cup I'} \sum_{j\in J} c_{ij} x_{ij} +$$

$$+ \sum_{j\in J} \left(h_j \int_0^{z_j} (z_j - v)\, \phi_j(v)\, dv + p_j \int_{z_j}^{\infty} (v - z_j)\, \phi_j(v)\, dv \right) \qquad (2.3.34)$$

s.a

$$z_j = \sum_{i\in I} x_{ij} \qquad j\in J, \qquad (2.3.35)$$

$$\sum_{j\in J} x_{ij} \leq s_i \ , \qquad i\in I\cup I', \qquad (2.3.36)$$

$$z_j \geq 0 \ , \qquad j\in J, \qquad (2.3.37)$$

$$x_{ij} \geq 0, \qquad i\in I, \ j\in J, \qquad (2.3.38)$$

$$y_i \in (0\ ,\ 1), \quad i \in I\cup I'. \qquad (2.3.39)$$

em que:

y_i = 1, se o serviço i for aberto

= 0, caso contrário;

x_{ij} = quantidade de procura do cliente j satisfeita a partir do serviço i;

f_i = custo fixo de instalação do serviço i;

c_{ij} = custos unitários de produção e distribuição para satisfazer a procura do cliente j a partir do serviço i;

s_i = um limite superior da capacidade do serviço i;

z_j = total recebido em j de todas as origens;

$\phi_j(w)$ = função densidade de probabilidade da procura do cliente j;

h_j = custo unitário de armazenagem em j;

p_j = custo por cada unidade cuja procura ainda não foi satisfeita em j;

I = conjunto de locais em que estão instalados os serviços abertos;

I' = conjunto de locais propostos para virem a ser instalados novos serviços;

J = conjunto de localizações de clientes candidatos.

As restrições (2.3.35) são de definição dos z_j ; as restrições (2.3.36) estabelecem limites superiores para a oferta .

Estes modelos não têm merecido grande atenção por parte dos investigadores e isso deve-se em parte ao facto de os erros que são introduzidos no modelo ao utilizar uma função densidade de probabilidade

aproximada para a procura se sobreporem aos que resultam de introduzir no modelo determinístico tradicional um valor esperado dessa mesma procura.

Outras versões de problemas discretos poderiam ser formuladas como, por exemplo, problemas de localização com vários níveis e vários produtos, problemas de localização dinâmica com restrições de capacidade, etc., etc...

Procurámos descrever de forma breve alguns dos principais problemas de localização que têm vindo a ser postos. Várias contribuições têm sido dadas no sentido de sistematizar a teoria de localização das quais referimos, por exemplo: Francis e al.(1983), Tansel e al.(1983a, b), Krarup e Pruzan (1983), Aikens(1985), Domschke e Drexel (1985), Love, Morris e Wesolowsky (1988), Hansen, Labbé Peeters e Thisse (1987), Brandeau e Chiu (1989). Brandeau e Chiu, apresentam uma lista de 50 problemas de localização que consideram mais representativos e procuram dar uma ideia sobre a forma como se relacionam, formalizam, quais os principais métodos utilizados na sua resolução e bibliografia.

CAPÍTULO 3

PROBLEMA DE LOCALIZAÇÃO SIMPLES
SPL

Neste capítulo iremos tratar o problema de localização simples. Faremos referência à sua complexidade computacional. Apresentaremos alguns dos métodos mais utilizados na sua resolução: métodos aproximados(heurísticos e relaxações), métodos heurísticos baseados no dual e métodos exactos.

3.1 – FORMULAÇÃO MATEMÁTICA

Retomemos a formulação inteira do SPL já apresentada em 2.3.1 :

$$\text{(SPL)} \qquad \text{Min} \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{3.1.1}$$

s.a

$$\sum_{i\in I} x_{ij} = 1, \quad j\in J, \tag{3.1.2}$$

$$y_i - x_{ij} \geq 0, \quad i\in I, \ j\in J, \tag{3.1.3}$$

$$x_{ij} \in \{0\ ,\ 1\}, \quad i\in I, \ j\in J, \tag{3.1.4}$$

$$y_i \in (0\ ,\ 1), \quad i \in I. \tag{3.1.5}$$

em que:

y_i = 1, se o serviço i for aberto

= 0, caso contrário;

x_{ij} = 1, se o cliente j for atendido no serviço i

= 0, caso contrário;

f_i = custo fixo de instalação do serviço i;

c_{ij} = custos de atender o cliente j no serviço i (podem incluir custos de produção , transporte ou outros);

I = conjunto de locais em que potencialmente poderão vir a ser instalados os serviços; $I = (1,2, \ldots ,m)$;

J = conjunto de clientes ; $J = (1,2, \ldots ,n)$.

As restrições (3.1.2) asseguram que todo o cliente é atendido; as restrições (3.1.3) garantem que a procura dos clientes é satisfeita a partir de fábricas abertas.

As restrições(3.1.4) de integralidade das variáveis x_{ij} podem ser substituidas por :

$$x_{ij} \geq 0\ ,\ \ i \in I,\ \ j \in J, \tag{3.1.4'}$$

sem perda de generalidade, visto que as *variáveis estratégicas* são as variáveis y_i. Uma vez fixado o vector de variáveis binárias (y_i) é fácil encontrar uma solução inteira para o SPL (3.1.1) - (3.1.5), como iremos ver. Pelo que acabamos de dizer podemos formular o SPL como problema de programação inteira mista (3.1.1)– (3.1.4'), (3.1.5). Esta formulação corresponde a uma situação concreta que

ocorre com alguma frequência em que cada cliente pode ser servido parcialmente por mais do que um serviço. São exemplos do problema de programação inteira pura, a afectação dos habitantes de determinada zona a determinada central telefónica; do problema de programação mista, o abastecimento de um retalhista a partir de várias fábricas.

É frequente encontrar uma outra formalização do SPL que resulta de substituir na formalização (3.1.1)– (3.1.4'), (3.1.5) o conjunto de restrições ,

$$y_i - x_{ij} \geq 0, \quad i \in I, \ j \in J, \tag{3.1.3}$$

pelo conjunto de restrições agregadas:

$$n y_i - \sum_{j \in J} x_{ij} \geq 0, \quad i \in I, \tag{3.1.3'}$$

em que $n = |J|$.

Teremos então as duas formalizações do SPL como problema de programação mista que a seguir se apresentam no quadro 3.1.

A vantagem da formalização condensada a que chamámos SPL - W sobre a SPL - S é a redução das **mxn** restrições (3.1.3) a **m** restrições (3.1.3'). As duas formalizações são equivalentes, isto é, a solução óptima não é afectada. No entanto, quando se utilizam métodos baseados na relaxação linear em que as restrições (3.1.5) de integralidade das variáveis y_i são substituidas por $y_i \geq 0$, $i \in \setminus I$, isto deixa de ser verdadeiro, porque a região admissível do SPL-S, definida por (3.1.2), (3.1.3) e $y_i \geq 0$, $x_{ij} \geq 0$, $i \in I$, $j \in J$, está estritamente contida na região admissível do SPL-W, definida por (3.1.2), (3.1.3') e $y_i \geq 0$, $x_{ij} \geq 0$, $i \in I$, $j \in J$, razão pela qual se chama à primeira, "forte" (Strong), e à segunda, com restrições agregadas, "fraca" (Weak) .

SPL-S	SPL-W
(3.1.1) $\text{Min} \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$ (3.1.1)	
s.a	s.a
(3.1.2) $\sum_{i \in I} x_{ij} = 1, \quad j \in J,$	$\sum_{i \in I} x_{ij} = 1, \quad j \in J,$ (3.1.2)
(3.1.3) $y_i - x_{ij} \geq 0, \quad i \in I, \ j \in J$	$n y_i - \sum_{j \in J} x_{ij} \geq 0, \quad i \in I$ (3.1.3')
(3.1.4) $x_{ij} \geq 0, \quad i \in I, \ j \in J$	$x_{ij} \geq 0, \quad i \in I, \ j \in J,$ (3.1.4)
(3.1.5) $y_i \in \{0, 1\}, \quad i \in I.$	$y_i \in \{0, 1\}, \quad i \in I.$ (3.1.5)

Quadro 3.1

Como referimos anteriormente, uma vez decidido qual o conjunto de serviços que vão ser abertos, $I^+ \subseteq I$, a afectação de clientes a serviços é trivial. Em particular, afecta-se o cliente j ao serviço "aberto" $k \in I^+$ de custo mínimo, $c_{kj} = \min_{i \in I^+} c_{ij}$.

O custo total correspondente ao conjunto de serviços I^+ será:

$$z(I^+) = \sum_{j \in J} \min_{i \in I^+} c_{ij} + \sum_{i \in I^+} f_i$$

Como cada cliente é atendido num só serviço, que corresponde a dizer que " o SPL goza da propriedade de afectação simples", o SPL pode tratar-se como problema de optimização combinatória com a seguinte formulação :

Formulação Combinatória do SPL

$$\min_{P \subseteq I} \left(\sum_{j \in J} \min_{i \in P} c_{ij} + \sum_{i \in P} f_i \right)$$

Nesta formulação do SPL o problema resume-se à identificação do subconjunto P de I, para o qual é mínimo o custo total; esse conjunto indicará o conjunto de serviços a abrir. Esta forma de tratar o problema evita a formulação inteira e tem tido um papel importante na implementação de algoritmos para a sua resolução.

3.2 — PROBLEMAS DE LOCALIZAÇÃO, PARTIÇÃO E COBERTURA

Os Problemas da Partição (PP) e da Cobertura (CP) podem ser considerados como casos particulares do SPL. Se tivermos presente a formulação do problema de cobertura de custo mínimo (2.2.8') — (2.2.10) :

$$\text{Min} \sum_{i \in I} f_i \; y_i \qquad (2.2.8')$$

s.a

$$\sum_{i \in I} a_{ji} \; y_i \geq 1, \quad j \in J, \qquad (2.2.9)$$

$$y_i \in \{0,1\}, \; i \in I \; . \qquad (2.2.10)$$

É fácil de ver que este pode ser considerado um SPL particular em que:

— os custos fixos, f_i , são os mesmos;

— os $c_{ij} = 0$, ou $c_{ij} = \infty$;

$$- \text{ e os } \quad a_{ij} = \begin{cases} 1, \text{ se } c_{ij} = 0 \\ 0, \text{ se } c_{ij} = \infty \end{cases}$$

Inversamente, para qualquer problema de cobertura, podemos definir um SPL com os mesmos custos fixos e em que:

$$c_{ij} = \begin{cases} 0 \text{ , se } a_{ij} = 1 \\ \infty \text{ , se } a_{ij} = 0 \end{cases}$$

Quando nas restrições (2.2.9) se substitui o sinal $\geq$ pelo sinal de = , temos um problema de partição.

Note-se que no que diz respeito à formulação do SPL é habitual considerar as origens em linhas e os destinos em colunas. Assim, para estabelecermos esta relação entre o SPL e o CP utilizámos a transposta da matriz $A_{mxn} = [\, a_{ij} \,]$ de zeros e uns da formulação do SPL.

3.3 — COMPLEXIDADE COMPUTACIONAL

Os algoritmos para resolver problemas de optimização podem classificar-se em:

- Algoritmos polinomiais
- Algoritmos exponenciais.

Um algoritmo diz-se " *Algoritmo em tempo polinomial*" para um problema **(P)** se para todas as instâncias de **(P)** , possíveis conjuntos de dados, o seu tempo de execução, medido em número de operações elementares, for limitado

superiormente por uma função polinomial da dimensão dos dados. Se for *n* essa dimensão e *k* o grau do polinómio, diz-se que o tempo de computação do algoritmo é $O(n^k)$ o que significa que é superiormente limitado por $C\,n^k$, para alguma constante positiva C e qualquer $n \geq 0$. Sempre que um algoritmo não é polinomial diz-se " *Algoritmo exponencial*".

Como a classificação dos algoritmos em polinomiais ou "bons" e exponenciais ou "maus" é feita a partir da análise do pior caso porque é feita para todas as possíveis instâncias dos dados, pode acontecer que para uma particular instância dos dados o algoritmo exponencial dê origem a menor esforço computacional do que um polinomial.

Uma questão que se põe perante determinado problema de optimização combinatória é a de saber se haverá algum algoritmo polinomial que o resolva. Para analisar estes problemas segundo a teoria da complexidade computacional é necessário encontrar os "problemas de decisão" que lhe estão associados, isto é, problemas que têm apenas uma de duas soluções possíveis: "sim" ou "não". Assim, por exemplo, no caso do SPL o problema de decisão, **Π(SPL)** que lhe está associado é:

Para uma dada instância ***(m ; n ; C ; f) e um dado valor constante positivo, K, haverá uma solução para o SPL de valor não superior a K?***

Em que *m* é o número de origens, *n* o número de destinos, *C* é a matriz de custos de atendimento e *f* é o vector de custos fixos .

Para grande número de problemas de optimização combinatória é possível provar a equivalência, em termos de complexidade computacional, entre problemas de optimização e problemas de decisão Π que lhe estão associados, Papadimitriou

e Steiglitz (1982). No entanto isto não é válido para todos os problemas de optimização combinatória. Assim, a existência de algoritmos polinomiais para problemas de optimização permite concluir a existência de algoritmos polinomiais para os correspondentes problemas de decisão, o recíproco só em certos casos é verdadeiro.

Diz-se que um problema de decisão pertence à classe $\mathcal{P}$, se existir algum algoritmo determinístico em tempo polinomial que o resolva; diz-se que um problema de decisão pertence à classe $\mathcal{NP}$, se a sua solução puder ser verificada em tempo polinomial. As iniciais NP, resultam de "Nondeterministic Polinomial time" o que significa que um problema pertence à classe $\mathcal{NP}$, se puder ser resolvido por um algoritmo não determinístico em tempo polinomial.

A classe $\mathcal{P} \subseteq \mathcal{NP}$, como fácilmente se vê . Se $\mathcal{P} \neq \mathcal{NP}$ é uma questão que se mantém em aberto.

Cook (1971) e Karp (1972) introduziram o conceito de classe de problemas NP-completos que passamos a definir.

Diz-se que um problema de decisão Π é *NP-hard* , intratável, se qualquer problema $\Pi' \in \mathcal{NP}$ for polinomialmente transformável em Π , isto significa que Π é pelo menos tão difícil como qualquer problema de $\mathcal{NP}$.

Um problema Π diz-se NP-completo, se for *NP-hard* e pertencer à classe $\mathcal{NP}$. Podemos então, dizer que a classe de problemas NP-completos é constituida por um conjunto de problemas equivalentes no sentido em que " se algum deles for solúvel por um algoritmo polinomial todos o serão ".

Podemos concluir que para justificar que determinado problema é *NP-hard* basta encontrar um problema conhecido, NP-completo, que seja transformável em

tempo polinomial no problema dado. Garey e Johnson (1979) apresentam uma lista de problemas NP-completos da qual faz parte o problema da cobertura. Ora, vimos na secção anterior que este pode ser considerado um caso particular do SPL e daí podemos concluir que:

" O Problema de Localização Simples é NP- hard ".

3.4 — MÉTODOS DE RESOLUÇÃO

Existe uma grande variedade de algoritmos que têm vindo a ser utilizados na resolução do SPL. Como o SPL é um problema de programação linear inteira todos os algoritmos que genéricamente se aplicam à resolução daqueles podem também ser aplicados ao SPL. No entanto, a generalidade pode ter um elevado preço em termos de tempo de resolução o que leva à procura de algoritmos específicos para resolver determinado problema, em geral motivados pela estrutura particular do respectivo problema.

Assim, podemos encontrar para resolver o SPL heurísticas, relaxações, algoritmos exactos.

3.4.1 — HEURÍSTICAS

Os primeiros métodos utilizados para resolver o SPL foram heurísticos. Estes métodos permitem obter soluções admissíveis, aproximadas, só excepcionalmente óptimas, num curto espaço de tempo. Continuam a desenvolver-se

heurísticas para encontrar soluções admissíveis do primal e do dual pois estas fornecem majorantes e minorantes da solução óptima, permitindo avaliar a qualidade da solução aproximada ou reduzir o tempo de computação quando utilizados em algoritmos "branch-and-bound". A solução obtida pela heurística pode ser suficiente para dar resposta ao problema, ou pode também ser utilizada como ponto de partida para um algoritmo exacto; tudo depende da qualidade da solução que seja requerida.

Dum modo geral podemos dizer que as heurísticas desenvolvidas para o SPL resultam de três ideias fundamentais.

A primeira procura soluções admissíveis, utilizando regras que lhe permitam a máxima melhoria em cada passo, dando por isso origem às heurísticas de tipo *"greedy "*.

A segunda é a de "pesquisa local ", parte de uma solução admissível, procura fazer trocas, fechando serviços, abrindo outros na vizinhança, na tentativa de obter uma melhor solução admissível, dando origem às heurísticas ditas de *" tipo melhorativo "* ou *"por trocas "* ou ainda de *"pesquisa local "*.

A terceira ideia, mais recente, é a de construir soluções heurísticas primais e duais, aos pares, através das condições de desvio complementar para as relaxações lineares, dando origem aos algoritmos heurísticos chamados *"primais-duais "*. A heurística primal-dual que maior impacto teve para a resolução do SPL foi a *"Dual Ascendente"* que faz parte do algoritmo DUALOC de Erlenkotter que descreveremos mais adiante.

É frequente encontrar estas ideias utilizadas conjuntamente num mesmo algoritmo. Assim, porque existe uma grande variedade de heurísticas, apenas para

exemplificar, vamos referir, em particular, dois métodos heurísticos que foram dos primeiros métodos apresentados para o SPL. O primeiro, deve-se a Kuehn e Hamburger (1963) foi proposto para localizar armazéns, e pode hoje ser considerado um método heurístico *"greedy melhorativo"*. Este algoritmo é formado de duas partes: o programa principal que é chamado "rotina de adição ", parte de uma configuração inicial baseada na procura local, e prossegue abrindo os serviços um a um de modo a que o custo total de cada configuração sofra o maior decréscimo possível, até que nenhum serviço possa ser aberto sem que isso aumente o custo total; a abertura é sequencial de tal modo que um serviço uma vez aberto, mantém-se assim na solução final. A segunda parte é uma subrotina que elimina serviços economicamente sem interesse devido à proximidade de outros que foram posteriormente abertos pela heurística *greedy* e em seguida, partindo desta solução admissível, analisa as trocas entre serviços abertos e fechados, efectuando-as sempre que tal represente melhoria para a função objectivo. O processo pára quando tiver sido encontrada uma solução que não possa ser melhorada por tais trocas.

Por sua vez, Manne (1964), apresentou um método heurístico para o SPL, *"Steepest Ascent One Point Move Algorithm"*, SAOPMA, que é também considerado de tipo *"greedy melhorativo "*. Nesta heurística parte-se de uma solução inicial admissível, em termos de (y_i), que pode ser qualquer das 2^m-1 possíveis, exceptuando a solução (0,0, ... , 0) em que todos os serviços estão fechados; depois move-se para cada uma das m soluções adjacentes que resultam de substituir um y_i pelo seu complementar, o que corresponde a fazer trocas entre serviços abertos e fechados, seleccionando a que originar um maior decréscimo para a função objectivo. A SAOPMA termina quando o movimento para qualquer

ponto adjacente não introduzir qualquer melhoria em relação à solução corrente.

Os métodos heurísticos são base para muitos outros algoritmos aproximados, e podem também ser úteis para algoritmos exactos que necessitem de soluções iniciais admissíveis ou mesmo, como dissemos já, para obter soluções finais. Só que, nesta última hipótese, é importante saber se a solução encontrada se afasta muito ou pouco do valor óptimo, isto é, avaliar o erro cometido quando se adopta aquela solução.

Várias têm sido as medidas utilizadas para avaliar a precisão das soluções obtidas por métodos heurísticos.

O *desvio absoluto*:

$$\varepsilon_a = \bar{z} - z^* \qquad (3.4.1)$$

em que $\bar{z}$ representa o valor da solução aproximada obtida pela heurística e z^* o valor óptimo. Este desvio tem pouco interesse prático, porque é sensível à mudança de escala nos dados.

O *desvio relativo*:

$$\varepsilon_r = (\bar{z} - z^*) / z^* \qquad (3.4.2)$$

é mais significativo, embora seja considerado uma medida inadequada para o SPL por Cornuejols, Fisher e Nemhauser (1977), por não satisfazer à seguinte propriedade que consideram essencial:

"Uma modificação dos dados que adicione uma constante δ ao valor da função objectivo para qualquer solução admissível, deixando a execução da heurística inalterada, deve deixar inalterada a medida do erro".

Ora tal não acontece, pois se por exemplo for adicionada a constante δ a uma linha da matriz de custos $C=[c_{ij}]$ do SPL, o valor de qualquer solução vem adicionado de δ, a execução da heurística não sofre alteração e o valor do erro relativo é alterado, pois passará a ser:

$$\varepsilon_r = (\bar{z} - z^{*}) / (z^{*} + \delta) \qquad (3.4.3)$$

podendo aumentar ou diminuir conforme o δ escolhido.

Neste mesmo artigo, C.F.N., propõem para medida do erro:

$$g_r = (\bar{z} - z^{*}) / (z_r - z^{*}) \qquad (3.4.4)$$

em que z_r é um valor que resulta da análise do pior caso, i.e., ($z_r - z^{*}$) representa o máximo desvio possível entre um valor obtido pela heurística que está em análise e o valor óptimo; g_r mede o desvio da solução encontrada pela heurística em relação ao pior desvio possível.

Para problemas de grande dimensão em que z^{*} é desconhecido, utiliza-se:

$$\overline{\varepsilon_r} = (\bar{z} - \underline{z}) / \underline{z} \qquad (3.4.5)$$

em que $\underline{z}$ representa um limite inferior para o valor óptimo, obtendo-se deste modo um limite superior para o erro cometido. Nestes casos as heurísticas baseadas no dual da relaxação linear do SPL, que permitem obter soluções primais e duais, desempenham um papel importante. Os valores das soluções duais admissíveis que constituem limites inferiores, $\underline{z}$, são utilizados para obter

limites superiores dos desvios dos valores das soluções primais, obtidos pela heurística, em relação ao valor óptimo e, à posteriori, permitem avaliar a qualidade da solução.

3.4.2 — RELAXAÇÕES LINEARES

As relaxações das restrições de integralidade nas formulações do SPL que designámos SPL-S e SPL-W, deram origem às chamadas *"Relaxação linear forte"* (SRS) e *" Relaxação linear fraca "* (WRS).

Estas relaxações lineares têm tido uma grande importância prática para a resolução do SPL, não só porque servem de base para muitos algoritmos aproximados que se baseiam em técnicas de programação linear, mas também porque a sua resolução ou a dos duais permite obter minorantes que são depois utilizados em algoritmos de tipo "branch-and-bound".

O cálculo de majorantes e minorantes é de uma grande importância para resolver o SPL e, como é sabido, a sua qualidade pode reduzir muito o tempo de computação necessário para obter a solução por um método "branch-and-bound". Põe-se, então, a questão da escolha entre a SRS e a WRS, isto é, há que optar entre a qualidade dos limites e a rapidez com que podem ser obtidos. No caso do SPL esta questão nem sempre é fácil de resolver, embora grande número de autores se incline claramente para a SRS por razões que exporemos mais adiante.

No quadro 3.2 apresentam-se as formulações das duas relaxações.

Relaxações lineares do SPL

Relaxação Forte	*Relaxação Fraca*
SRS	**WRS**
(3.4.6) $\text{Min} \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$	(3.4.6)
s.a	*s.a*
(3.4.7) $\sum_{i \in I} x_{ij} = 1, \quad j \in J,$	$\sum_{i \in I} x_{ij} = 1, \quad j \in J,$ (3.4.7)
(3.4.8) $y_i - x_{ij} \geq 0, \quad i \in I, \; j \in J$	$n y_i - \sum_{j \in J} x_{ij} \geq 0, \quad i \in I$ (3.4.8')
(3.4.9) $x_{ij} \geq 0, \quad i \in I, \; j \in J$	$x_{ij} \geq 0, \quad i \in I, \; j \in J,$ (3.4.9)
(3.4.10) $y_i \geq 0, \quad i \in I.$	$y_i \geq 0, \quad i \in I.$ (3.4.10)

Quadro 3.2

Vejamos algumas das vantagens ou inconvenientes da escolha de uma ou outra ou suas duais. A relaxação forte, SRS, tem $n(m+1)$ restrições e $m(2n+1)$ variáveis, incluidas as de desvio, enquanto a relaxação fraca apresenta apenas $m+n$ restrições e $m(n+2)$ variáveis, incluidas as de desvio também. Em termos computacionais esta redução do número de variáveis e restrições representa uma clara vantagem para a WRS; no entanto, como cada restrição agregada representa a soma de n restrições da SRS, então qualquer solução admissível da SRS será admissível para a WRS, mas o recíproco não é, em geral, verdadeiro. A

região admissível da SRS está contida na região admissível da WRS, o que significa que a SRS dá melhores aproximações e isto pode representar uma grande vantagem, por exemplo, quando utilizada em algoritmos "branch-and-bound", pela economia de tempo de computação que pode representar.

A relaxação linear fraca para além de ter menos restrições pode ser resolvida por inspecção. O resultado que acabamos de referir deve-se a Efroymson e Ray (1966) e pode resumir-se na seguinte proposição:

Proposição 3.1

Se $f_i \geq 0$, *para* $i \in I$, *existe uma solução óptima* (y^*, x^*) *para a relaxação linear fraca do SPL com* $x^*_{i(j)j} = 1$ *em que* $i(j)$ *é o serviço de menor custo para o cliente* j, *i.e.,* $c_{i(j)j} = \min_{i \in I} (c_{ij} + f_i/n)$, *para todo o* $j \in J$, $x^*_{ij} = 0$, *nos restantes casos, e em que* $y^*_i = (1/n) \sum_{j \in J} x^*_{ij}$, *para todo o* $i \in I$. □

Spielberg (1969), observou que o dual do WRS é também solúvel por inspecção.

Se forem $v = (v_1, v_2, \ldots, v_n)$ e $w = (w_1, w_2, \ldots, w_m)$ as variáveis duais associadas às n restrições,

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J$$

e às m desigualdades,

$$n\, y_i - \sum_{j \in J} x_{ij} \geq 0, \quad i \in I$$

respectivamente, teremos a seguinte formalização do dual do WRS:

Dual da Relaxação Linear Fraca

$$(DWRS) \quad \text{Max} \sum_{j \in J} v_j \tag{3.4.11}$$

s.a

$$n\, w_i \leq f_i\,, \quad i \in I, \tag{3.4.12}$$

$$v_j - w_i \leq c_{ij}\,, \quad i \in I,\ j \in J, \tag{3.4.13}$$

$$w_i \geq 0, \quad i \in I, \tag{3.4.14}$$

v_j *sem restrição de sinal.*

Para maximizar o $\sum_j v_j$ basta maximizar cada v_j separadamente, isto é, basta escolher cada $v_j = \min_{i \in I} (w_i + c_{ij})$ o que por sua vez implica que a escolha óptima é $w_i = f_i/n$, como resulta imediatamente de (3.4.12) e (3.4.13).

Proposição 3.2

O valor da solução óptima do dual , DWRS, da relaxação linear fraca , WRS do SPL, é igual ao valor óptimo do WRS e igual $\sum_{j \in J} \min_i (c_{ij} + f_i / n)$. *O tempo necessário para a obter é O (mn)* . □

O facto de a relaxação linear fraca só em casos muito excepcionais ter solução inteira e a má qualidade dos limites inferiores que permite obter têm levado a que a maior parte dos autores a abandonem e optem pelo

desenvolvimento de técnicas para resolver problemas de grande dimensão com vista à resolução da SRS.

A relaxação linear forte para além de produzir minorantes de boa qualidade, no sentido em que pouca enumeração requerem, quando aplicados a algoritmos de tipo "branch-and-bound" tem solução óptima inteira em grande número de casos.

O exemplo simples que a seguir apresentamos permite ilustrar o que acabamos de afirmar.

EXEMPLO – 3.1

Consideremos o problema de localização simples, SPL(m=5,n=6) com os seguintes dados:

$$C = [c_{ij}] = \begin{bmatrix} 12 & 8 & 2 & 3 & 8 & 2 \\ 13 & 4 & 6 & 5 & 0 & 0 \\ 6 & 9 & 6 & 2 & 5 & 3 \\ 0 & 1 & 0 & 10 & 10 & 4 \\ 1 & 2 & 1 & 8 & 8 & 1 \end{bmatrix} \qquad f = (f_i) = \begin{bmatrix} 4 \\ 3 \\ 4 \\ 4 \\ 7 \end{bmatrix}$$

Utilizando a Proposição 3.1 para determinar a solução óptima da relaxação linear fraca, WRS, obtivemos os resultados seguintes:

$$x^*_{25} = x^*_{26} = x^*_{34} = x^*_{41} = x^*_{42} = x^*_{43} = 1\ ,\ y^*_2 = \frac{1}{3}\ ,\ y^*_3 = \frac{1}{6}\ ,\ y^*_4 = \frac{1}{2}\ ,\ z^* = \frac{20}{3}\ .$$

Da resolução da relaxação linear forte, SRS , obtivemos a solução :

$$x^*_{24} = x^*_{25} = x^*_{26} = x^*_{41} = x^*_{42} = x^*_{43} = 1\ ,\ y^*_2 = 1\ ,\ y^*_4 = 1\ ,\ z^* = 13\ .$$

Como se vê é inteira e portanto coincide com a solução óptima do SPL.

Note-se também a diferença entre os minorantes fornecidos pelas duas relaxações.

Balinski (1965), ReVelle e Swain (1970), Schrage (1975) entre outros observaram que : *"para todas as instâncias dum SPL a relaxação linear forte, SRS, tem solução óptima inteira em grande percentagem dos casos"*.

Morris (1978) procurou justificar que a observação anterior, conhecida como conjectura de Balinski era quase correcta. Para tal utilizou cinco conjuntos de 100 problemas gerados aleatóriamente e concluiu que em cerca de 96% dos casos as soluções óptimas do SPL e do SRS eram coincidentes. Estes resultados vieram estimular o desenvolvimento de técnicas para resolver problemas lineares de grande dimensão tendo em vista a resolução do SPL. Algoritmos de decomposição de Benders e Dantzig-Wolfe, simplex modificado, dualidade Lagrangeana, optimização subgradiente, etc., são apenas alguns exemplos dos métodos que têm sido aplicados à resolução do SRS.

Mais recentemente, têm sido desenvolvidos algoritmos que resolvem o dual da relaxação linear forte e que apresentam algumas vantagens:

– permitem obter bons minorantes para a solução óptima do SPL;

– é fácil gerar soluções inteiras do primal a partir de soluções do dual, como iremos ver.

Entre os métodos baseados no dual da relaxação linear forte do SPL, destaca-se um método que foi desenvolvido independentemente por Bilde e Krarup (1967) e por Erlenkotter (1978) e que é largamente aceite como o mais potente para resolver o SPL. Dos dois algoritmos fazem parte heurísticas designadas

"Maximização do limite inferior" no algoritmo de Bilde e Krarup e *"Dual ascendente"* no algoritmo de Erlenkotter que são essencialmente idênticas e que geram soluções quase óptimas do dual da SRS. Com base nas soluções admissíveis do dual, geram-se soluções admissíveis do primal.

3.4.3 — ALGORITMOS EXACTOS

Os algoritmos que considerámos anteriormente permitem obter soluções aproximadas. No caso das heurísticas, as soluções aproximadas são admissíveis; no caso das relaxações as soluções são aproximadas, mas não admissíveis, a não ser que a relaxação tenha solução óptima inteira que, em tal hipótese, coincidirá com a solução óptima do SPL. Iremos agora fazer referência a métodos que permitem determinar a solução óptima do SPL.

As primeiras tentativas para determinar a solução óptima do SPL datam de 1963. Stollsteimer(1963), tentou resolvê-lo, utilizando um método de enumeração completa; Balinski e Wolfe (1963), propuseram um método baseado na decomposição de Benders. No entanto, qualquer destes métodos foi abandonado em resultado da experiência computacional. Pode dizer-se que o primeiro algoritmo aceitável para determinar a solução exacta do SPL foi desenvolvido por Efroymson e Ray (1966) e, independentemente, por Zimmerman (1967).

Este algoritmo de tipo "branch-and-bound" é baseado na relaxação linear fraca, WRS, do SPL, i.e., utiliza os minorantes obtidos por aquela relaxação.

Podemos dividir os algoritmos exactos em dois grupos:

— algoritmos baseados na resolução da relaxação linear do SPL;

— algoritmos baseados na resolução do dual.

Nos algoritmos do primeiro grupo, a resolução das relaxações lineares forte, SRS, ou fraca, WRS, do SPL permite obter minorantes para o valor da solução óptima que são depois utilizados num "branch-and-bound".

Os algoritmos do segundo grupo, mais recentes, e também mais eficientes, baseiam-se na resolução do dual da relaxação linear, DSRS, ou do dual Lagrangeano para obter os minorantes que uma vez mais são utilizados num "branch-and-bound". É neste grupo que se situa o algoritmo DUALOC, desenvolvido por Erlenkotter (1978) e que descreveremos a seguir por ser considerado o mais eficiente para resolver o SPL.

Estes algoritmos exactos ou optimais são em geral preteridos em relação a algoritmos suboptimais por necessitarem de grande espaço de memória de computador e muito tempo de cáculo.

3.4.3.1 — ALGORITMOS BASEADOS NA RESOLUÇÃO DO DUAL

Como já dissemos o algoritmo que tem vindo a ser considerado mais eficiente para resolver o SPL é um algoritmo "branch-and-bound" baseado na resolução do dual da relaxação linear forte do SPL, desenvolvido por Bilde-Krarup (1967) e independentemente por Erlenkotter (1978).

O algoritmo de Bilde e Krarup foi publicado em 1967 em Dimamarquês e a sua publicação em Inglês só viria a ser feita em 1977. Em 1978, Erlenkotter, sem ter tido conhecimento prévio do algoritmo de Bilde e Krarup, publicou um artigo em que apresentou um algoritmo a que chamou, DUALOC, que vai um pouco mais

longe do que o algoritmo de B - K , na medida em que utiliza as violações das condições de desvio complementar para melhorar as soluções obtidas pelo dual ascendente, utilizando para tal um método heurístico a chamou " *Ajustamento do dual*" . Se ainda assim, a solução encontrada não for óptima para o SPL, os minorantes encontrados por estes procedimentos são utilizados num "branch-and-bound" que completa o algoritmo de Erlenkotter.

É portanto este algoritmo que começaremos por descrever. Ao longo da descrição procuraremos fazer referências às suas semelhanças com o algoritmo de Bilde e Krarup.

Consideremos em primeiro lugar a formalização (3.4.6) – (3.4.10) da relaxação linear forte, SRS, e do seu dual, DSRS.

PRIMAL – **SRS**	DUAL – **DSRS**
(3.4.6)	(3.4.15)
Min $z_P = \sum_{i\in I} f_i\, y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}\, x_{ij}$	Max $z_D = \sum_{j\in J} v_j$
s.a	*s.a*
(3.4.7) $\sum_{i\in I} x_{ij} = 1, \quad j\in J,$	$\sum_{j\in J} w_{ij} \leq f_i\,, \quad i\in I,$ (3.4.16)
(3.4.8) $y_i - x_{ij} \geq 0, \quad i\in I,\ j\in J$	$v_j - w_{ij} \leq c_{ij}\,, \ i\in I, j\in J$ (3.4.17)
(3.4.9) $x_{ij} \geq 0\,, \quad i\in I,\ j\in J$	$w_{ij} \geq 0\,, \quad i\in I,\ j\in J,$ (3.4.18)
(3.4.10) $y_i \geq 0\,, \quad i\in I.$	v_j, *sem restrição*

Quadro 3.3

Podemos eliminar restrições e variáveis desta formulação do dual, observando que:

a) Em qualquer solução óptima as restrições (3.4.17) e (3.4.18) podem ser substituidas por $w_{ij} = max\ (\ 0,\ v_j - c_{ij}\ )$;

b) Em qualquer solução óptima $v_j = \min_i\ (\ c_{ij} + w_{ij}\ )$.

Então existe uma solução óptima para o dual, DSRS, que satisfaz à seguinte igualdade:

$$max \sum_{j \in J} v_j = \sum_{j \in J} min_i\ (c_{ij} + w_{ij}\ ) . \qquad (3.4.19)$$

Sem alterar o valor da função objectivo do dual, podemos obter duas formulações, equivalentes, que envolvem na forma explícita apenas as variáveis w_{ij} ou v_j .

Assim, teremos uma formulação que envolve na forma explícita as variáveis w_{ij} , que foi a utilizada no algoritmo de Bilde e Krarup (1967), e designada por:

Maximização do Limite Inferior

$$\text{Max } z_D = \sum_{j \in J} min_i\ (\ c_{ij} + w_{ij}\ ) = z^*_{DSRS} \qquad (3.4.20)$$

s.a

$$\sum_{j \in J} w_{ij} \leq f_i\ , \quad i \in I, \qquad (3.4.21)$$

$$w_{ij} \geq 0, \quad i \in I,\ j \in J \qquad (3.4.22)$$

Ora, pelo teorema da dualidade forte em programação linear,

$$z^*_{SRS} = z^*_{DSRS}$$

e, portanto,

$$z^{*}_{DSRS} \leq z^{*}_{SPL} .$$

E se para qualquer escolha das variáveis v_j considerarmos os w_{ij} iguais ao menor valor, não negativo, possível

$$w_{ij} = max \{ 0 , v_j - c_{ij} \}, \qquad (3.4.23)$$

a admissibilidade manter-se-á e o valor da função objectivo não será alterado e teremos a seguinte formulação do dual, designada *"Dual Condensado"* que envolve apenas as variáveis v_j na forma explícita:

Dual Condensado

$$\text{Max } z_D = \sum_{j \in J} v_j \qquad (3.4.24)$$

s.a

$$\sum_{j \in J} max \{ 0, v_j - c_{ij} \} \leq f_i , \quad i \in I. \qquad (3.4.25)$$

A resolução de qualquer destas formas equivalentes permite obter um limite inferior para o valor da solução óptima do SPL. No entanto, como a ideia de utilizar métodos baseados no dual é a obtenção de minorantes para serem posteriormente utilizados em algoritmos "branch-and-bound", não há necessidade de determinar a solução óptima do dual para realizar estes objectivos. Isto porque, na maioria dos casos, soluções sub-optimais são perfeitamente

suficientes para o cancelamento dos nodos da árvore de ramificação num intervalo de tempo muito menor.

Como consequência disso, muitos autores têm procurado implementar algoritmos em que se conjuguem a qualidade das soluções do dual, sub-optimais, com tempos de computação.

Foi nesse sentido que surgiram as heurísticas *"Maximização do Limite Inferior "* de Bilde e Krarup, em que foi utilizada a versão do dual nas variáveis w_{ij} e *"Dual Ascendente"* de Erlenkotter, em que foi utilizada a versão do dual nas variáveis v_j , (Dual Condensado). Estas duas heurísticas são semelhantes e ambas procuram obter os melhores minorantes para posterior aplicação num "branch-and-bound". Descreveremos o DUALOC por ser uma versão mais refinada na medida em que inclui ainda uma subrotina que pode permitir melhorar a solução do dual.

DUALOC

O DUALOC é um algoritmo exacto para resolver o SPL e que pode resumir-se nos seguintes procedimentos:

1 - *Dual Ascendente* - heurística de tipo *"dual-greedy"* que permite obter soluções duais admissíveis e construir soluções inteiras do primal a partir daquelas;

2 - *Ajustamento do Dual* - heurística de tipo melhorativo que procura reduzir as violações das condições de desvio complementar, melhorando as soluções obtidas pela Dual Ascendente;

Se pelos procedimentos anteriores não foi encontrada a solução óptima do

SPL , completa-se o processo aplicando as soluções encontradas a um algoritmo :

3 - *Branch-and- bound* .

Em cada nodo da árvore de ramificação repetem-se os procedimentos Dual Ascendente e Ajustamento do Dual.

Antes de descrever o algoritmo vamos fazer algumas considerações prévias sobre a construção de soluções admissíveis do primal a partir das soluções do dual e definir a notação que será utilizada .

Para construir as soluções admissíveis do primal a partir de soluções do dual, utilizam-se as condições de desvio complementar para a solução óptima do SRS :

$$y_i^* \left[f_i - \sum_{j \in J} \max \left(0, v_j^* - c_{ij} \right) \right] = 0 \qquad (3.4.26)$$

$$\left[y_i^* - x_{ij}^* \right] \max \left(0 , v_j^* - c_{ij} \right) = 0. \qquad (3.4.27)$$

Adoptaremos a seguinte notação:

(y_i^* , x_{ij}^*) = solução óptima do primal, SRS;

(v_j^*) = solução óptima do dual, DSRS;

(v_j^+) = solução admissível do dual ;

$I^* = \left(i : \sum_{j \in J} \max \left(0, v_j^+ - c_{ij} \right) = f_i \right)$,

conjunto de serviços para os quais as restrições (3.4.16) do dual condensado se verificam como igualdades;

I^+ = conjunto mínimo de serviços tais que para todo o $j \in J$ se tenha $v_j^+ - c_{ij} \geq 0$ para algum $i \in I^+$, $I^+ \subseteq I^*$;

$I^+(j)$ = $\{ i \in I^+ : c_{i^+(j)\, j} = \min_{i \in I^+} c_{ij} \}$, $j \in J$,

i.e., $I^+(j)$ é o serviço do conjunto I^+ de menor custo para o cliente j ;

(y^+, x^+) = solução inteira admissível do SPL, construida a partir da solução do dual do seguinte modo:

$$y_i^+ = \begin{cases} 1 , & i \in I^+ \\ 0 , & i \notin I^+ \end{cases} \qquad (3.4.28)$$

$$x_{ij}^+ = \begin{cases} 1 , & i = i^+(j) , \; j \in J \\ 0 , & \text{nos restantes casos} \end{cases}$$

Esta solução satisfaz à condição de desvio complementar (3.4.26), mas pode violar a condição (3.4.27). Se satisfizer às duas condições, prova-se que a solução assim construida é óptima inteira e, portanto, é solução óptima do SPL. Se as condições (3.4.27) forem violadas, isso significa que a desigualdade $v_j^+ - c_{ij} > 0$ se verifica para mais do que um índice $i \in I^+$, uma vez que $y_i^+ = x_{ij}^+ = 1$, apenas para o índice i que corresponde ao serviço de custo mínimo.

Se for z_P^+ o valor da função objectivo correspondente à solução inteira do primal que acabámos de construir e z_D^+ o valor da função objectivo do dual correspondente à solução (v_j^+), podemos estabelecer a seguinte relação entre as violações das condições de desvio complementar e a diferença $z_P^+ - z_D^+$ através do lema que a seguir enunciamos e demonstramos porque a sua demonstração torna mais perceptível a forma como se desenvolve o algoritmo.

Lema

$$z_P^+ - z_D^+ = \sum_{j \in J} \sum_{i \in I^+, i \neq i^+(j)} \max (0, v_j^+ - c_{ij}).$$

Demonstração:

$$z_D^+ = z_D^+ + \sum_{i \in I^+} [f_i - \sum_{j \in J} \max (0, v_j^+ - c_{ij})] =$$

$$= \sum_{j \in J} v_j^+ + \sum_{i \in I^+} f_i + \sum_{j \in J} (c_{i^+(j)\, j} - v_j^+) -$$

$$- \sum_{j \in J} \sum_{i \in I^+, i \neq i^+(j)} \max (0, v_j^+ - c_{ij}) =$$

$$= z_P^+ - \sum_{j \in J} \sum_{i \in I^+, i \neq i^+(j)} \max (0, v_j^+ - c_{ij}) . \qquad \square$$

Se não houver violação das condições de desvio complementar, então $z_D^+ = z_P^+$ e a solução inteira é óptima do SPL. O DUALOC desenvolve-se procurando reduzir o mais possível essas violações .

1 — HEURÍSTICA DUAL ASCENDENTE

Esta heurística parte de uma solução admissível $v_j = \min_{i \in I} c_{ij}$, $j \in J$ e vai procurando aumentar cada v_j para o c_{ij} imediatamente superior até que todos os v_j estejam impossibilitados de aumentar pelo facto de algumas das restrições:

$$\sum_{j \in J} \max (0, v_j - c_{ij}) \leq f_i , \qquad i \in I \qquad (3.4.29)$$

se verificarem como igualdades. Se algum v_j puder ainda crescer, mas a passagem para o c_{ij} imediatamente superior implicar a violação de alguma das restrições (3.4.29), será acrescentado do máximo permitido pela restrição. Quando todos os v_j estiverem impossibilitados de aumentar, o processo termina.

Antes de iniciar o processo é necessário ordenar os custos c_{ij} por ordem não decrescente para cada j ; designaremos por c_j^k , para $k = 1,2, \ldots , m$, os custos ordenados e incluiremos um custo artificial $c_j^{m+1} = +\infty$. Designaremos por $J^+ \subseteq J$ o conjunto de índices j tais que v_j é candidato a aumentar. No início, $J^+ = J$.

ALGORITMO

1 – Inicialização:

$\{ v_j \}$ solução dual admissível tal que $v_j \geq \min_{i \in I} c_{ij}$, $j \in J$;

$$s_i = f_i - \sum_{j \in J} \max \{ 0, v_j - c_{ij} \} \geq 0 \ , \ i \in I \ ;$$

$k(j) = \min \{ k : v_j \leq c_j^k \}$.

Se $v_j = c_j^{k(j)}$, $k(j) = k(j) + 1$.

2 – Fazer $j = 1$ e $\delta = 0$.

3 – Se $j \notin J^+$, ir para o passo → 7 .

4 – Fazer $\Delta_j = \min_{i \in I} \{ s_i : v_j - c_{ij} \geq 0 \}$.

5 – Se $\Delta_j > c_j^{k(j)} - v_j$, fazer:

$\Delta_j = c_j^{k(j)} - v_j$, $\delta = 1$, $k(j) = k(j) + 1$.

6 – Para cada $i \in I$ tal que $v_j - c_{ij} \geq 0$ fazer :

$$s_i = s_i - \Delta_j$$

$$v_j = v_j + \Delta_j$$

7 – Se $j \neq n$, fazer $j = j + 1$ e voltar ao passo →[3] .

8 – Se $\delta = 1$, voltar ao passo →[2].

FIM .

O algoritmo Dual Ascendente permite obter uma solução dual a partir da qual se pode construir uma solução primal admissível, inteira, pelo processo descrito em (3.4.29). Se este par de soluções satisfizer as condições de desvio complementar (3.4.27), então as soluções são óptimas , $z_P^+ = z_D^+$ e a solução é óptima para o SPL .

Se houver violação das condições (3.4.27), então, $z_P^+ - z_D^+ > 0$ e o algoritmo prossegue, procurando reduzir este *gap*[2] ou mesmo eliminá-lo através de um ajustamento dos valores das variáveis duais, procedimento a que Erlenkotter deu o nome de *"Ajustamento do Dual"*. Para fazer esse ajustamento, escolhe-se algum j' para o qual a condição (3.4.27) seja violada e reduz-se $v_{j'}^+$ baixando o seu valor para o $c_{ij'}$ imediatamente inferior a $v_{j'}^+$. Com esta redução pelo menos duas restrições saturadas do tipo, $\sum_{j \in J} \max (0, v_j^+ - c_{ij}) = f_i$ deixarão de o ser, e, assim, outros v_j poderão agora ser aumentados. Haverá uma redistribuição dos recursos e, com ela, um possível aumento de z_D^+ com a consequente redução do *gap*.

Antes de procedermos à descrição do *Ajustamento do Dual* vamos definir

---

[2]Este gap não é o "gap de dualidade" no sentido rigoroso do termo, mas um limite superior daquele.

alguma notação que não foi utilizada no Dual Ascendente :

$$c_{max}^{(j)} = \max_{i \in I} \{ c_{ij} : v_j - c_{ij} > 0 \} , \; j \in J;$$

$$I_j^* = \{ i : s_i = 0 \wedge v_j - c_{ij} \geq 0 \} , \; j \in J;$$

$$I_j^+ = \{ i \in I^+ : v_j - c_{ij} > 0 \} , \; j \in J;$$

$$J_i^+ = \{ j : I_j^* = \{ i \} \} , \; i \in I .$$

Note-se que:

— J_i^+ é o conjunto de clientes que só podem ser atendidos a partir de i o que faz de i um serviço essencial.

— Se $|I_j^+| \leq 1 , \forall j$, a solução (y^+, x^+) que corresponde ao conjunto de serviços abertos, I^+, é óptima.

— Quando a condição (3.4.27) não é verificada para algum j, isso equivale a ter $|I_j^+| > 1$.

— Se para algum j, $|I_j^+| > 1$, define-se a origem $i'(j) \in I^+$, como a segunda melhor origem para atender o cliente j, isto é:

$$c_{i'(j)j} = \min_{i \in I^+, i \neq i^+(j)} c_{ij} .$$

2 — AJUSTAMENTO DO DUAL

ALGORITMO

1 — Inicializar $j = 1$

2 — Se $|I_j^+| \leq 1$, ir para o passo → 7 .

3 – Se $J^{+}_{i^{+}(j)} = \emptyset$ e $J^{+}_{i'(j)} = \emptyset$, ir para o passo [7] .

4 – Para cada $i \in I$ tal que $v_j > c_{ij}$, fazer :

$$\Delta = v_j - \underset{max}{c_{ij}} \quad ,$$

$$s_i = s_i + \Delta \quad ,$$

$$v_j = \underset{max}{c_{ij}} \quad .$$

5 – Fazer :

a) $J^{+} = J^{+}_{i^{+}(j)} \cup J^{+}_{i'(j)}$, e executar o Dual Ascendente ;

b) Ampliar J^{+}, $J^{+} = J^{+} \cup \{ j \}$ e repetir o Dual Ascendente;

c) Fazer $J^{+} = J$ e repetir o Dual Ascendente .

6 – Se $v_j \neq \underset{max}{c_{ij}} + \Delta$, (v_j diferente do valor inicial) voltar a → [2] .

7 – Se $j \neq n$, $j = j + 1$ e voltar a → [2] .

FIM .

Com o passo 5–b) pretende-se distribuir pelas restantes variáveis v_j , recursos que não tenham sido absorvidos pelos acréscimos feitos nas variáveis v_j com $j \in J^{+}_{i^{+}(j)} \cup J^{+}_{i'(j)}$ no passo 5– a) .

A execução do passo 5–c) tem por objectivo terminar o processo com uma solução admissível do dual (v^{+}_{j}).

Se o valor da função objectivo do dual melhorou em relação ao valor obtido anteriormente, repetir o Ajustamento do Dual pode ainda introduzir alguma melhoria, isto, claro, se entretanto não foi encontrada a solução óptima.

Se as execuções do Dual Ascendente e Ajustamento do Dual não permitirem obter a solução óptima inteira, completa-se o algoritmo com um "branch-and-

bound", utilizando os valores das soluções encontradas, z_D^+ e z_P^+ , para minorantes e majorantes, respectivamente. Passamos a descrever as regras deste "branch-and-bound" que são simples, dada a qualidade dos majorantes e minorantes que entretanto foram obtidos.

3 – BRANCH-AND-BOUND

1 – *Ramificação*

A ramificação é feita a partir do serviço de custo mínimo que contribua para a violação das condições de desvio complementar (3.4.27) e que primeiramente for encontrado, isto é :

$$i^+(j) \quad : \mid I_j^+ \mid > 1 \quad .$$

2 – Inicia-se a ramificação a partir de um serviço i que se fixa fechado.

3 – A pesquisa dos nodos da árvore é feita em profundidade. O retrocesso é feito pelo processo Last-In-First-Out (LIFO).

4 – Fixa-se o serviço i aberto, substituindo o custo fixo f_i , por $f_i = 0$ e de seguida restabelece-se a admissibilidade do dual. Para isso, é necessário reduzir a c_{ij} todos os v_j tais que $v_j > c_{ij}$ para o serviço i aberto e, além disso, adicionar à função objectivo do dual, z_D, o custo fixo, f_i , do serviço aberto. Deste modo, o valor da função objectivo do dual não será alterado porque, se: $\sum_{j \in J} \max (0, v_j - c_{ij}) = f_i$, $i \in I^+$, a redução dos $v_j > c_{ij}$ a c_{ij} produzirá um decréscimo em z_D igual f_i , se adicionarmos f_i a z_D o valor manter-se-á depois da redução dos v_j.

Com esta redução algumas das variáveis desvio $f_i - \sum_{j \in J} \max (0, v_j - c_{ij})$ aumentaram e, portanto, repetem-se os procedimentos Dual Ascendente e

Ajustamento do Dual.

5 – O cancelamento dos nodos faz-se ou porque se encontrou uma solução óptima inteira, ou por comparação dos valores de z_D com outros valores de soluções já encontradas anteriormente.

A simplicidade e eficiência do ponto de vista computacional da heurística dual ascendente, tem levado a que numerosos investigadores tenham procurado adaptá-la à resolução de outros problemas. Magnanti e Wong (1988) publicaram um artigo em que confirmam as potencialidades daquele método e mostram a sua eficiência quando aplicado a problemas de grande dimensão. Faremos referência a algumas dessas aplicações mais adiante na última secção, (3.4.18), deste capítulo.

Para ilustrar a aplicação do método Dual Ascendente e Ajustamento do Dual, apresentamos a seguir os resultados da resolução dum problema proposto por Khumawala (1972). No capítulo 4, apresentaremos resultados obtidos para outros problemas.

O exemplo é um SPL (m=5, n=8).

EXEMPLO – 3.4.1

Vamos considerar duas hipóteses para o vector de custos fixos :

(a) $\mathbf{f} = (f_i) = (100 , 70 , 60 , 110 , 80)$;

(b) $\mathbf{f} = (f_i) = (200 , 200 , 200 , 400 , 300)$.

A matriz de custos, designada por C é a que a seguir se apresenta.

Matriz de custos $C_{mxn} = [\, c_{ij} \,]$

	c_{ij}							
i \ j	1	2	3	4	5	6	7	8
1	120	180	100	$+\infty$	60	$+\infty$	180	$+\infty$
2	210	$+\infty$	150	240	55	210	110	165
3	180	190	110	195	50	$+\infty$	$+\infty$	195
4	210	190	150	180	65	120	160	120
5	170	150	110	150	70	195	200	$+\infty$

Para o vector de custos fixos considerados na hipótese (a), os resultados obtidos foram os que se apresentam a seguir:

iteração

$p = 1$ $\quad (\nu_j^1) = (\,170,\ 180,\ 110,\ 180,\ 55,\ 195,\ 160,\ 155)$

$(s_i^1) = (\,40,\ 20,\ 55,\ 0,\ 20\,)$

$z_D^1 = \sum_j \nu_j^1 = 1205\ ;$

$p = 2$ $\quad (\nu_j^2) = (\,180,\ 190,\ 110,\ 180,\ 60,\ 195,\ 160,\ 155)$

$(s_i^2) = (\,20,\ 15,\ 50,\ 0,\ 0\,)$

$z_D^2 = \sum_j \nu_j^2 = 1230;$

$p = 3$ $\quad (\nu_j^3) = (\ 180, 190, 110, 180, 65, 195, 160, 155)$

$(s_i^3) = (\ 15, 10, 45, 0, 0\ )$

$z_D^3 = \sum_j \nu_j^3 = 1235$.

Em que $s_i = f_i - \sum_{j \in J} \max(\ 0, \nu_j - c_{ij}\ )$ são as variáveis de desvio;

ν_j^p = variáveis duais para a iteração p e z_D^p é o valor da solução dual.

O conjunto , $I^+ = (\ 4\ ,\ 5\ )$.

A solução primal (y^+, x^+) :

$$y_4 = y_5 = 1$$

$$x_{51} = x_{52} = x_{53} = x_{54} = x_{45} = x_{46} = x_{47} = x_{48} = 1$$

O valor da solução primal: $z_P^+ = 1235 = z_D^+$. A solução é óptima.

Para o vector de custos fixos, considerado na hipótese (b) os resultados obtidos foram:

Iteração

$p = 1$ $\quad (\nu_j^1) = (\ 210, 190, 150, 240, 65, 245, 195, 235)$

$(s_i^1) = (\ 30, 0, 70, 65, 0\ )$

$z_D^1 = \sum_j \nu_j^1 = 1530$;

$I^+ = (\ 2\ ,\ 5\ )$ $\quad z_P^+ = 1605.$

Como $\nu_6 > c_{26} > c_{56}$, a solução primal viola as condições de desvio complementar e aplica-se o ajustamento do dual.

$|I_6^+| = 2$, $i^+(6) = 5$, $i'(6) = 2$, $J_{i^+(6)}^+ = (\ 2\ )$, $J_{i'(6)}^+ = (\ 5,7,8\ )$.

Se actualizarmos os valores de s_i , reduzindo ν_6 para 210 e somando $\Delta = 75$ a s_2, s_4 e s_5 teremos $(s_i) = (\ 30, 75, 70, 140, 75)$.

Aplicando o ajustamento do dual obtivemos os seguintes resultados:

$p = 2$ $\quad (\nu_j^2) = (\ 210, 220, 150, 240, 65, 245, 195, 235)$

(s_i^2) = (0, 0, 0, 35, 10)

$z_D^2 = \sum_j v_j^2 = 1560$;

I^+ = (1 , 2) $z_P^+ = 1580$.

As soluções dual e primal melhoraram , aplica-se de novo o ajustamento do dual .Como $v_5^2 > c_{15} > c_{25}$, $|I_5^+| = 2$, $i^+(5) = 2$, $J_{i^+(5)}^+$ = (2), $J_{i^+(5)}^+ = \emptyset$.

Se actualizarmos os valores de s_i , reduzindo v_5^2 para 60 e somando $\Delta = 5$, a s_1, s_2 e s_3 teremos (s_i) = (5, 5, 5, 35, 10).

Aplicando de novo o ajustamento do dual obtivemos os seguintes resultados:

p = 3 (v_j^3) = (210, 225, 150, 240, 60, 250, 195, 235)

(s_i^3) = (0, 0, 0, 25, 0)

$z_D^3 = \sum_j v_j^3 = 1565$, solução óptima dual;

I^+ = (1 , 2) $z_P^+ = 1580$.

Como $z_P^+ - z_D^3 > 0$, e já não há possibilidade de aumentar z_D há necessidade de aplicar um" branch- and- bound" para obter a solução óptima.

Como dissemos o DUALOC continua a ser considerado o método de resolução mais eficiente para o SPL, no entanto, têm sido propostas algumas modificações, ligeiras, a que vamos fazer referência já em seguida.

A primeira referência vai para uma modificação proposta pelo próprio Erlenkotter (1982), à heurística Ajustamento do Dual. Neste artigo em que é apresentado um algoritmo baseado no dual para o problema de localização dinâmico, DYNALOC, ele propõe que em vez da heurística Ajustamento do Dual se utilize uma heurística de Ajustamento Primal-Dual. A principal diferença entre

os dois procedimentos consiste no seguinte: quando se aplica o ajustamento do dual a solução do primal só é calculada após este procedimento ter terminado; quando se aplica o ajustamento primal-dual a solução do primal é actualizada de cada vez que a solução dual é alterada. No DUALOC é utilizado um contador para controlar o número de vezes que é executado o ajustamento do dual; no ajustamento primal-dual este contador faz-se igual a zero sempre que ocorra alguma melhoria para o objectivo dual.

Tcha, Ro, Yoo (1988) propuseram um método de resolução para o SPL que pode considerar-se um DUALOC modificado. A modificação proposta é a substituição da heurística Dual Ascendente pela heurística a que chamam " Heurística Aditiva Baseada no Dual" .

A diferença entre as duas heurísticas reside essencialmente na forma como são calculadas as variáveis de desvio. Essa forma diferente de definir as variáveis de desvio resultou da interpretação económica do dual da relaxação linear do SPL (Dual Condensado).

Assim no algoritmo de Erlenkotter, parte-se de um conjunto vazio de serviços abertos e vão-se ajustando as variáveis duais v_j que representam os rendimentos de atender o cliente j, até que a diferença entre custos fixos de instalação do serviço e rendimento líquido obtido a partir de i , $s_i = f_i - \sum_{j \in J} \max (0, v_j - c_{ij}) \geq 0$, para $i \in I$, se anule. Se $s_i = 0$, abre-se o serviço i.

No algoritmo proposto por Tcha e al. parte-se também dum conjunto vazio de serviços abertos, $I^* = \emptyset$, e são utilizadas variáveis de desvio, $S_i = - f_i + \sum_{j \in J_i} \max (0, v_j^+ - c_{ij})$, para $i \in \bar{I}^*$ com um papel semelhante ao das variáveis de desvio do algoritmo Dual Ascendente. Estas variáveis são diferentes porque o somatório restringe-se ao conjunto de clientes $j \in J_i$ cujos custos, c_{ij} , a partir

de serviços fechados são menores do que os custos mínimos correntes, $c_{i^*(j),j}$, a partir de serviços $i \in I^*$, abertos. À medida que os v_j aumentam os $S_i \leq 0$ vão aumentando e quando algum atinge o valor zero o correspondente serviço $i^* \in \bar{I}^*$ é adicionado ao conjunto I^*. Actualizam-se os conjuntos $I^* = I^* \cup \{i^*\}$, $\bar{I}^* = \bar{I}^* - \{i^*\}$ e $J^+ = J^+ - \{j : v_j \geq c_{i^*j}\}$. Procede-se então à actualização dos S_i para i pertencente ao novo $\bar{I}^*$. Nessa actualização pode haver um acréscimo de tempo de computação em relação ao algoritmo de Erlenkotter. Contudo, este efeito pode ser atenuado pelo facto de o conjunto J^+ ser actualizado e, consequentemente reduzido.

Segundo Tcha e al., esta substituição das variáveis s_i por S_i apresenta a vantagem de as segundas traduzirem o ganho que resulta, em termos de rendimento líquido, da abertura de um determinado serviço que se encontrava fechado. Enquanto que os s_i representam o rendimento total. São apresentados resultados computacionais da aplicação da heurística à resolução do SPL nas versões estática e dinâmica e são comparados esses resultados com os obtidos pelo Dual Ascendente. Concluem que esta heurística fornece melhores aproximações, em geral, embora com ligeiros acréscimos de tempo computacional em alguns casos.

3.4.4 — DUAL ASCENDENTE E RELAXAÇÃO LAGRANGEANA

A teoria da Relaxação Lagrangeana, Geoffrion (1974), veio permitir a unificação de vários métodos utilizados na obtenção de minorantes/majorantes dos valores óptimos de problemas de programação linear inteira, tendo em vista

a aplicação a métodos "branch-and-bound".

A Relaxação Lagrangeana consiste em identificar um determinado conjunto de restrições que ponderadas com um conjunto de multiplicadores são introduzidas na função objectivo. Com a supressão deste conjunto de restrições, *"complicantes"*, obtém-se um problema que é mais fácil de resolver e cujo valor óptimo é um minorante/majorante da solução do problema inicial, consoante se trate de um problema de minimização /maximização.

Consideremos o seguinte problema geral de programação linear inteira mista:

$$
\begin{aligned}
(P)\ \min z = {} & c\,x \\
\text{s.a} \quad & \\
& A\,x \geq b \\
& B\,x \geq d \\
& x \geq 0\,, \\
& x_j \ , \ \textit{inteiro} \ , \ j \in I.
\end{aligned}
$$

Em que b, c e d são vectores, A, B são matrizes e o conjunto de índices I designa o conjunto de variáveis que são inteiras.

A Relaxação Lagrangeana de (P) relativa ao conjunto de restrições $A\,x \geq b$ e a um vector de multiplicadores $\lambda \geq 0$ é definida por:

$$
\begin{aligned}
(PR_\lambda) \quad & z_L(\lambda) = \min [c\,x + \lambda\,(b - Ax)] \\
\text{s.a} \quad & \\
& Bx \geq d \\
& x \geq 0 \\
& x_j \ , \ \textit{inteiro}, \ j \in I.
\end{aligned}
$$

No caso do SPL podemos considerar duas relaxações Lagrangeanas. A primeira, considerando como conjunto de restrições *"complicantes"*, $y_i - x_{ij} \geq 0$, $i \in I$, $j \in J$ com um vector de multiplicadores associados, w_{ij}, não negativos; a segunda, considerando como restrições *"complicantes"*, $\sum_{i \in I} x_{ij} = 1$, $j \in J$ com o correspondente vector de multiplicadores v_j, sem restrição de sinal. Vamos apresentar a formalização das duas relaxações e em seguida analisar, em particular, a primeira.

Relaxação Lagrangeana do SPL

1 — (PR_w)

$$z_L(w) = \min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} +$$

$$+ \sum_{i \in I} \sum_{j \in J} (x_{ij} - y_i) w_{ij} =$$

$$= \min \sum_{i \in I} \sum_{j \in J} (c_{ij} + w_{ij}) x_{ij} + \sum_{i \in I} \Big(f_i - \sum_{j \in J} w_{ij} \Big) y_i \quad (3.4.30)$$

s.a

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J, \quad (3.4.31)$$

$$x_{ij} \geq 0, \quad i \in I, \quad j \in J, \quad (3.4.32)$$

$$y_i \in \{0, 1\}, \quad i \in I. \quad (3.4.33)$$

2 – (PR_v)

$$z_{\llcorner}(v) = \min \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij}$$

$$+ \sum_{j\in J} (1 - \sum_{i\in I} x_{ij}) v_j =$$

$$= \min \sum_{i\in I} (f_i y_i - \sum_{j\in J} (v_j - c_{ij})x_{ij}) + \sum_{j\in J} v_j \quad . \qquad (3.4.34)$$

s.a

$$y_i - x_{ij} \geq 0, \quad i\in I, \ j\in J, \qquad (3.4.35)$$

$$x_{ij} \geq 0, \quad i\in I, \ j\in J, \qquad (3.4.36)$$

$$y_i \in \{0 , 1\}, \quad i\in I. \qquad (3.4.37)$$

Consideremos a relaxação Lagrangeana do problema (PR_w) .

Se supusermos fixados os w_{ij} , para minimizar a função em **x** e **y** vê-se fácilmente, a partir de (3.4.30), que apenas interessa abrir os serviços, i , (y_i = 1) para os quais, $f_i - \sum_{j\in J} w_{ij} \leq 0$. Então o problema (PR_w) pode formalizar-se do seguinte modo:

$$z_{\llcorner}(w) = \min \sum_{i\in I}\sum_{j\in J} (c_{ij} + w_{ij})x_{ij} + \sum_{i\in I} \min \left\{ f_i - \sum_{j\in J} w_{ij} , 0 \right\} \qquad (3.4.38)$$

s.a

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J, \tag{3.4.39}$$

$$x_{ij} \geq 0, \quad i \in I, \ j \in J. \tag{3.4.40}$$

Se os w_{ij} foram fixados de tal modo que se tenha, $f_i - \sum_{j \in J} w_{ij} < 0$, para algum i, então, podemos reduzir os w_{ij} de modo que aquela diferença se anule. Esta redução dos w_{ij}, provoca um acréscimo no valor do mínimo, $z_L(w)$, e uma consequente melhoria do minorante do valor da solução óptima. Ficamos assim com uma região admissível para os multiplicadores reduzida a:

$$A_W = \{ w_{ij} : f_i - \sum_{j \in J} w_{ij} \geq 0 , i \in I , w_{ij} \geq 0 , i \in I, j \in J \}.$$

Para obter os melhores minorantes possíveis, a partir da relaxação Lagrangeana, há que resolver o problema dual lagrangeano, i. e., maximizar em w, a função $z_L(w)$. Mas, pelo que vimos anteriormente, só interessa considerar os $w_{ij} \in A_W$. Se chamarmos $z_L^0(w)$ à restrição de $z_L(w)$ à região A_W, esta pode ser calculada, directamente, por inspecção, determinando os mínimos das colunas da matriz (C + W), e adicionando:

$$z_L^0(w) = \sum_{j \in J} \min_i (c_{ij} + w_{ij}) . \tag{3.4.41}$$

A solução da relaxação Lagrangeana seria então:

$$y_i = \begin{cases} 1 , & se \quad \sum w_{ij} = f_i \\ 0 , & nos \ restantes \ casos \end{cases} \tag{3.4.42}$$

e se for $I^+ = (i : y_i = 1)$ e $i^+ (j)$ tal que $c_{i^+(j)} = \min_{i \in I^+} c_{ij}$,

$$x_{ij} = \begin{cases} 1 \ , \ i = i^+ (j) , j \in J \\ 0 \ , \ nos\ restantes\ casos \end{cases} \tag{3.4.43}$$

Maximizando $z_L{}^0(w)$, $w \in A_W$, teremos:

$$\max_{w} [z_L(w)] = \max z_L{}^0(w) \tag{3.4.44}$$

s.a

$$\sum_{j \in J} w_{ij} \leq f_i \ , \quad i \in I, \tag{3.4.45}$$

$$w_{ij} \geq 0, \quad i \in I, \ j \in J. \tag{3.4.46}$$

Se tivermos presentes as formulações do dual da relaxação linear do SPL, nas variáveis w_{ij} , (3.4.20) – (3.4.22) e nas variáveis v_j , (3.4.15) – (3.4.18), utilizadas por Bilde - Krarup e Erlenkotter, respectivamente, e a formulação do problema dual Lagrangeano (3.4.44) – (3.4.46) verificamos que são equivalentes, isto é, todas conduzem ao mesmo "problema de maximização do limite inferior" da solução óptima do SPL e consequentemente às mesmas soluções.

Em termos de relaxação Lagrangeana os algoritmos *"Maximização do Limite Inferior "* de Bilde - Krarup e *"Dual Ascendente"* de Erlenkotter podem ser considerados relaxação Lagrangeana *parametrizada*, na medida em que os multiplicadores são obtidos através de sucessivos ajustamentos.

3.4.5 – ALGUNS EXEMPLOS DE PROBLEMAS A QUE TEM SIDO APLICADO O DUAL ASCENDENTE

A eficiência do método Dual Ascendente tem despertado interesse por parte dos investigadores que por isso o têm procurado aplicar a diversos problemas. Referimos alguns exemplos dessas aplicações: "Árvore de Steiner", Wong (1984); " Problema de localização sem restrições de capacidade com vários níveis", Tcha e Lee, (1984); "Problema de localização dinâmica", Van Roy e Erlenkotter, (1982); "Problema de afectação generalizado", Fisher, Jaikumar, Van Wassenhove, (1986), "Problema da partição", Fisher e Kedia, (1986) e mais recentemente, Balakrishnan, Magnanti e Wong (1989), publicaram um artigo em que mostram as potencialidades do dual ascendente para resolver uma grande variedade de modelos de "*Network Design*", NDP, e fazem uma generalização do método para problemas NDP. Essa generalização abarca os algoritmos propostos para os casos particulares anteriormente referidos e ainda outros que podem ser englobados neste grupo de problemas de *Network Design*.

CAPÍTULO 4

SIMULATED ANNEALLING

4.1 – INTRODUÇÃO

As heurísticas de "tipo melhorativo" ou "por trocas" constituem uma classe importante de heurísticas para problemas de optimização combinatória, *NP-hard*. Estas heurísticas partem de uma solução inicial admissível e procuram modificá-la fazendo um número limitado de trocas, relativamente a uma vizinhança préviamente definida, com o objectivo de encontrar uma solução melhor. No entanto, as trocas só se efectuam, quando daí resultar melhoria imediata para a função objectivo. A paragem ocorre quando é encontrada uma solução óptima local, relativamente à vizinhança préviamente definida, isto é, quando nenhuma melhoria for possível na referida vizinhança. Como estes problemas podem ter vários óptimos locais é frequente correr o programa várias vezes utilizando pontos de partida arbitráriamente escolhidos, existindo sempre o risco de ficarmos presos a soluções locais de má qualidade.

Com o objectivo de reduzir esse risco, tem vindo a ser proposto um novo método aproximado para problemas de optimização combinatória, NP-hard, designado ***"Simulated Annealing"***. Este método pode ser considerado uma versão generalizada da heurística por trocas.

As heurísticas por trocas partem de uma solução admissível e produzem uma sequência de soluções, na vizinhança, que apenas são aceites quando daí resultar melhoria para o valor da função objectivo. O *simulated annealing* pode ser considerado uma heurística por trocas, mas em que estas são aleatorizadas, uma vez que, se a troca originar melhoria para a função objectivo, é aceite; se a troca piorar o valor da função objectivo, a aceitação ou rejeição é feita de acordo com um critério probabilístico, podendo assim ser aceites soluções que piorem o valor da função objectivo. A probabilidade de aceitação de trocas que piorem o valor da solução corrente é calculada através de uma função que depende da diferença entre os valores das duas soluções e dum parâmetro de controlo, a *temperatura*.

O conceito de *"annealing"* em optimização combinatória, foi introduzido no início dos anos 80 por Kirkpatrick, Gelatt & Vecchi, (1983;1984) e independentemente por Cerny (1985). A ideia resultou da forte semelhança que existe entre o processo de *annealing* dos sólidos, na Física, e o processo de resolução de problemas de optimização combinatória.

As expectativas criadas pelo *simulated annealing* na obtenção de melhores soluções e a facilidade de implementação do algoritmo têm dado origem a que vários investigadores se debrucem sobre ele. Como resultado disso, têm surgido diversas publicações sobre aplicações do *simulated annealing* a problemas variados, tais como o problema do caixeiro viajante, Kirkpatrick (1984), Cerny (1985), Johnson, Aragon, McGeoch & Schevon (1990), problema da partição dum grafo, Kirkpatrick (1984), Johnson, Aragon, McGeoch & Schevon (1989), problema de afectação quadrática, Burkard & Rendl (1984), Wilhelm & Ward (1987), problemas de escalonamento, Eglese & Rand (1987), emparelhamento máximo, Sasaki

e Hajek (1988) etc. Para uma bibliografia mais detalhada sobre as aplicações, vejam-se, por exemplo, Van Laarhoven & Aarts (1987), Collins, Eglese, Golden (1988), E. Aarts, J. Korst (1989).

Para se perceber melhor o *algoritmo simulated annealing*, começaremos por estabelecer a correspondência entre um problema de optimização combinatória e o sistema físico que motivou a utilização do *simulated annealing* em problemas de optimização. Em seguida descreveremos o algoritmo de uma forma genérica, e por fim, faremos a sua aplicação ao problema de localização simples, SPL.

4.2 — ANALOGIA FÍSICA

Na Física da matéria condensada, *annealing* é conhecido como um processo térmico que permite obter estados de baixa energia de um sólido quando submetido a um *banho quente*. O processo é formado pelos dois passos seguintes:

— Aumentar a temperatura do *banho quente* até um valor suficientemente grande de modo a que o sólido se funda.

— Baixar a temperatura cuidadosamente (*careful annealing*) até que o ordenamento das partículas se complete, dando origem ao chamado *ground state* ou *estado fundamental* do sólido.

Na fase líquida as partículas do sólido movem-se aleatóriamente e a energia do sistema é elevada. No *estado fundamental*, as partículas estão arrumadas numa malha altamente estruturada e a energia do sistema é mínima. Este estado de arrumação das partículas do sólido só é possível quando a temperatura máxima do *banho* for suficientemente alta e o abaixamento for suficientemente lento. Se assim não for, apenas serão conseguidos estados meta-

estáveis, em vez do *estado fundamental*.

O processo oposto ao *annealing* é designado por *quenching* e corresponde ao abaixamento instantâneo da temperatura dando também origem a estados meta-estáveis.

O processo que sumáriamente acabamos de descrever corresponde ao processo de formação de um cristal perfeitamente estruturado, no caso de ser conseguido o estado de energia mínima, *estado fundamental*; de formação de um cristal de má qualidade, ou mesmo um vidro[3], no caso de serem atingidos estados meta-estáveis.

O elevado número de partículas que se encontram em quaisquer amostras macroscópicas de líquidos ou sólidos, número que é da ordem de 10^{23} por centímetro cúbico, permite concluir sobre a impossibilidade de observar experimentalmente o comportamento de um qualquer sistema em equilíbrio térmico, sendo apenas possível observar o comportamento mais provável do sistema. Para isso, utiliza-se um conjunto de configurações idênticas do sistema a partir das quais se calculam o comportamento médio do sistema e desvios em torno dessa média. Cada configuração, definida por um conjunto de posições atómicas, $\{ r_i \}$, do sistema é ponderada pelo seu factor de probabilidade de Boltzmann, $exp\left(-\frac{E_i}{k_B T}\right)$ em que E_i é a energia da configuração, K_B é a constante de Boltzmann, e T é a temperatura.

A Mecânica Estatística é uma disciplina da Física da matéria condensada

---

[3]Aqui vidro é entendido no seguinte sentido: "*produto inorgânico de fusão que foi arrefecido até ficar rígido sem cristalizar*". Definição da American Society for Testing and Materials.

em que são desenvolvidos métodos para analisar propriedades agregadas dos átomos que se encontram em amostras de líquidos ou sólidos. Para tal, recorre com frequência a técnicas de simulação. Metropolis Rosenbluth, Teller&Teller (1953) propuseram um algoritmo, no âmbito da Mecânica Estatística, para simular a evolução de um sólido submetido a um *banho quente* à temperatura, T, até ao estado de equilíbrio térmico. Este algoritmo gera, por simulação, uma sucessão de estados em equilíbrio a uma dada temperatura. Em cada passo é gerado por deslocamento aleatório de um átomo um novo estado. Sejam E_i e E_j as energias dos estados corrente e seguinte, respectivamente. Se a diferença de energia for menor ou igual a zero, i. e., $\Delta E = E_j - E_i \leq 0$, o estado j é aceite como estado corrente ; se $\Delta E > 0$, estado j é aceite com uma probabilidade que é dada por :

$$P(\Delta E) = \exp\left(\frac{E_i - E_j}{k_B T}\right) , \qquad (4.2.1)$$

em que T é a *temperatura* do banho quente e k_B a *constante de Boltzmann*. Através dum gerador de números pseudo-aleatórios uniformemente distribuidos em (0 , 1), escolhe-se um número pseudo-aleatório que é comparado com a $P(\Delta E)$; se esta for superior aceita-se j como estado corrente, se for inferior, mantém-se a configuração anterior. Repetindo este procedimento muitas vezes gera-se um grande número de transições e desse modo consegue-se fazer a simulação do processo de abaixamento lento da temperatura (*careful annealing*), de modo a que seja atingido o estado de equilíbrio térmico. Daí a designação *simulated annealing*.

A regra de aceitação que acabámos de descrever é conhecida como ***critério de Metropolis*** e o algoritmo em que foi introduzida ***Algoritmo de***

Metropolis. O equilíbrio térmico de um sólido é caracterizado pela distribuição de *Boltzmann*. Essa distribuição permite obter a probabilidade de um sólido estar no estado i com energia E_i à temperatura T, e é dada por:

$$\mathbf{P}_T \{ \mathbf{X}=i \} = \frac{1}{Z(T)} \exp\left(\frac{-E_i}{k_B T} \right), \qquad (4.2.2)$$

em que $\mathbf{X}$ é a variável aleatória que designa o estado corrente do sólido e $Z(T)$ é a função de repartição. A função de repartição $Z(T)$ é definida por:

$$Z(T) = \sum_j \exp\left(\frac{-E_j}{k_B T} \right), \qquad (4.2.3)$$

em que o somatório se estende ao conjunto de todos os estados possíveis.

Estados fundamentais ou configurações próximas destas são extremamente raras entre todas as possíveis configurações de um corpo macroscópico. Determinar o estado de energia mínima dum sistema , *estado fundamental*, é um problema de optimização não menos difícil do que os encontrados em optimização combinatória, quando se pretende determinar a solução de custo mínimo.

Existe uma certa analogia entre o processo de simulação proposto por Metropolis para gerar estados em equilíbrio a uma determinada temperatura e o processo iterativo utilizado nas heurísticas de tipo melhorativo com a função custo a desempenhar o papel de energia. Nas heurísticas de tipo melhorativo, são geradas soluções que apenas são aceites quando baixam o custo. Isto equivale a apenas aceitar configurações do sistema que baixem a energia o que daria origem a um arrefecimento rápido (*quenching*). E desse modo as soluções encontradas seriam quase sempre meta-estáveis, isto é, soluções de má qualidade.

No *simulated annealing* são gerados estados que podem ser aceites, ainda que da sua aceitação resulte um aumento de energia do sistema. Esta aceitação é feita de acordo com o *critério de Metropolis*.

A analogia que acabámos de verificar entre estes dois processos levou Kirkpatrick, Gelatt e Vecchi (1983), a mostrar como era possível transpôr o processo *simulated annealing* para problemas de optimização combinatória, fazendo corresponder estados do sistema físico a soluções admissíveis do problema de optimização, energia dos estados a custos das soluções ; estado de energia mínima a solução óptima do problema de optimização. Como a temperatura do sistema físico não tinha equivalente no problema de optimização, introduziram um parâmetro que desempenha o papel de temperatura e que é utilizado no algoritmo como parâmetro de controlo.

O que se pretende ao aplicar o *simulated annealing* a problemas de optimização combinatória é tentar obter melhores soluções do que as obtidas por heurísticas de tipo melhorativo. Como dissemos já, o processo de pesquisa local corresponde ao processo de arrefecimento brusco, *quenching*. Isto porque, se a temperatura baixa bruscamente de um valor elevado até T=0°, nenhuma transição pode levar a um estado de energia superior, situação que é correspondente a só aceitar transições para soluções que baixem o valor da função objectivo, num problema de minimização. As consequências deste procedimento são, no caso do sistema físico, soluções meta-estáveis e soluções de má qualidade no problema de optimização. Se em vez do arrefecimento rápido for utilizado um processo de arrefecimento lento (*careful annealing*) ao qual corresponde o algoritmo *simulated annealing* é de esperar que, tal como acontece no sistema físico, seja possível encontrar melhores soluções. Neste sentido, o *simulated annealing* pode

considerar-se uma versão melhorada ou uma generalização dum método heurístico de tipo melhorativo.

Podemos resumir a analogia que acabámos de estabelecer entre o sistema físico e o problema de optimização no seguinte quadro:

ESTADO FÍSICO	PROBLEMA DE OPTIMIZAÇÃO
Estado	Solução Admissível
Energia	Custo
Estado fundamental	Solução Óptima
Arrefecimento Rápido	Optimização Local
Careful Annealing	Simulated Annealing

Quadro 4.1

4.3 — ALGORITMO SIMULATED ANNEALING

Como já dissemos o algoritmo *simulated annealing* pode considerar-se uma generalização do método heurístico de tipo melhorativo. Começaremos por fazer, de uma forma breve, a descrição de um algoritmo de optimização de tipo

melhorativo porque isso facilita a descrição do algoritmo *simulated annealing* e permite justificar de uma forma mais clara a afirmação anterior.

Dada uma determinada instância de um problema de optimização combinatória, procura-se por um qualquer processo uma solução admissível, S, e suponhamos que o seu custo é c(S). Para definir um algoritmo de tipo melhorativo há que definir vizinhança, N(S), de uma dada solução S. As trocas processam-se entre soluções da vizinhança. A solução final pode não ser óptima, mas é a de menor custo da vizinhança.

O *algoritmo de optimização local* pode resumir-se no seguinte:

ALGORITMO

1 — Inicialização:

1.1 — Procurar uma solução inicial S e calcular o custo (S).

2 — Repetir o seguinte:

2.1 — Se S' for uma solução não testada da vizinhança de S,

calcular custo (S').

2.2 — Se o custo (S') < custo (S) , fazer S = S'.

3 — Voltar a 2.

FIM.

Estes algoritmos apresentam, como já referimos, o inconveniente de podermos ficar presos a soluções locais, por vezes distantes da solução óptima. O *simulated annealing* oferece expectativas de que este inconveniente seja minorado por permitir que sejam aceites soluções que pioram o valor da função

objectivo. A aceitação dessas soluções ocorre sob a influência de um gerador de números pseudo-aleatórios uniformemente distribuidos em (0,1) e de um parâmetro de controlo, a *temperatura*. [4]

Tal como acontecia no algoritmo de optimização local, parte-se de uma solução admissível, define-se um processo de geração de soluções e um critério de aceitação. É neste critério de aceitação que reside a principal diferença entre os dois algoritmos.

O *algoritmo simulated annealing* pode resumir-se nos seguintes passos:

ALGORITMO

1. — Inicialização:

   1.1 — Escolher um valor inicial para a *temperatura* $To>0$.

   1.2 — Escolher um parâmetro de *arrefecimento* , $0< r <1$.

   1.3 — Procurar uma solução inicial S e calcular o custo (S).

2 — Repetir o seguinte até ser atingida a temperatura de congelação.

   2.1 — Repetir L vezes o seguinte:

      2.1.1 — Seleccionar, aleatóriamente, uma solução S'
      da vizinhança de S.

      2.1.2 — Calcular Δ = custo (S') — custo (S).

---

[4]Como o *simulated annealing* aplicado a problemas de optimização combinatória resultou da analogia com o processo de *annealing* da Física, os termos físicos ali adoptados continuaram a ser utilizados pelos autores que o têm aplicado a problemas de optimização combinatória.

2.1.3 – *Se* $\Delta \leq 0$, ir para → 2.1.6

2.1.4 – Calcular a probabilidade $P(\Delta) = exp(-\Delta / T)$.

Gerar um número pseudo-aleatório $p \in (0, 1)$.

2.1.5 – Se $P(\Delta) < p$, rejeitar a transição. Ir para → 2.1.1

2.1.6 – *Aceitar a transição.*

Fazer $S = S'$.

Custo (S) = Custo (S) + Δ .

Voltar a → 2.1.1

2.2 – Se critério de paragem for verdeiro → FIM.

2.3 – *Reduzir a temperatura.*

Fazer $T = rT$, $0 < r < 1$.

3. – Voltar a → 2.1

FIM.

Como podemos observar pela descrição do algoritmo que acabámos de fazer nele estão contidos dois ciclos. O primeiro, repete-se para valores decrescentes de T, obtidos por aplicação do factor de redução r, até que nenhuma melhoria na função objectivo pareça provável. Este processo de abaixamento do parâmetro T, corresponde ao arrefecimento lento na Física, processo que termina quando é atingido o estado de congelação (*frozen*), estado quase estável de energia localmente mínima. O segundo ciclo, contido neste, repete-se para cada valor da temperatura, T, L vezes, sendo este número L o número de transições geradas para cada valor do parâmetro T. Faremos referência mais adiante a alguns dos métodos que têm vindo a ser utilizados para definir o sistema de parâmetros de entrada, questão extremamente

importante quer para a convergência do algoritmo quer para a sua eficiência.

Para valores de $\Delta > 0$ e $T > 0$, $exp(-\Delta/T)$, é um número do intervalo (0,1) e pode interpretar-se como uma probabilidade que depende de Δ e T. Inicialmente, para valores elevados de T, a probabilidade de serem aceites grandes deteriorações é grande; mas à medida que o valor de T vai baixando, essa probabilidade diminui, sendo apenas aceites pequenas deteriorações. Finalmente, quando $T \to 0$, a probabilidade de aceitar transições que aumentem o valor da função objectivo tende para zero e praticamente deixarão de ser aceites deteriorações. Para ilustrar o que acabamos de afirmar, veja-se o gráfico da fig 4.2 em que se representa o custo médio das soluções aceites para 100 valores decrescentes da temperatura. O problema a que se refere o gráfico é o problema das 33 cidades, Karg e Thompson (1964), com custos fixos $f_i = 2000$.

Ao transportar o *algoritmo simulated annealing* para a optimização combinatória, temos estado a considerar apenas a analogia com o processo de simulação do *annealing* na Física em que este se baseia. É possível, no entanto, justificar matemáticamente o *algoritmo simulated annealing* com base na teoria das cadeias de Markov finitas, formalizando matemáticamente a noção de *equilíbrio do sistema físico* como *distribuição de equilíbrio* de uma cadeia de Markov.

Vários autores se têm dedicado à formalização matemática, justificação teórica e estudo da convergência deste algoritmo, vejam-se por exemplo, Gidas (1985), Mitra, Romeo e Sangiovanni-Vicentelli (1986), Sasaki e Hajek (1988), H.L. Aarts e J. Korst (1989).

J. Korst, H.L.Aarts (1989) apresentam alguns dos principais resultados teóricos que têm sido estabelecidos para o *algoritmo simulated annealing*, entre

os quais se encontram teoremas em que são enunciadas as condições que garantem que o algoritmo converge assintóticamente para o conjunto de soluções óptimas globais, i. e., assintóticamente, o algoritmo permite obter, uma solução óptima com probabilidade igual a um.

Da demonstração da convergência assintótica, resulta que o *simulated annealing* necessita de um número infinito de transições para atingir a solução óptima. Isto levaria a que, em qualquer implementação prática, fosse gerada uma sucessão de cadeias de Markov, infinitas, homogéneas, para valores descendentes da temperatura o que é claramente impraticavel. Korst e Aarts mostram que, pode descrever-se o *algoritmo simulated annealing* como uma sucessão de cadeias homogéneas, finitas, geradas para valores decrescentes da temperatura. Isto corresponde a considerar o processo como uma combinação de cadeias homogéneas numa só cadeia não homogénea. Deste modo, em vez de uma *sucessão de cadeias* homogéneas infinitas passaríamos a ter *uma só cadeia* infinita não homogénea. Korst e Aarts demonstram que o *algoritmo simulated annealing* assim formulado ainda converge assintoticamente para o conjunto de soluções óptimas desde que o arrefecimento seja suficientemente lento.

Em qualquer das formulações do *simulated annealing*, consideradas anteriormente, o algoritmo só permite obter a solução óptima a partir de um número infinito de transições o que é impraticável. Recorre-se então a aproximações da convergência assintótica que, como é evidente, só permitem obter soluções sub-optimais.

Vamos a seguir apresentar uma implementação do algoritmo, primeiramente de uma forma genérica e depois, em particular, para o problema de localização simples , SPL.

4.3.1 — ARREFECIMENTO ESCALONADO

Pela descrição do algoritmo que fizemos anteriormente e pelo que dissemos sobre a formalização matemática do *simulated annealing* facilmente se conclui que uma implementação do algoritmo em tempo polinomial pode ser feita, gerando cadeias homogéneas, finitas, para valores decrescentes do parâmetro de controlo, a *temperatura*. Para isso, há que especificar, previamente, um conjunto de parâmetros que combinados no chamado "***cooling schedule***" vão depois governar a convergência do algoritmo.

Um ***arrefecimento escalonado*** expressão que passaremos a adoptar para o *cooling schedule*, especifica:

1 — Uma sucessão finita de valores do parâmetro de controlo, T, através de:

- Valor inicial do parâmetro de controlo — ***Temperatura inicial*** — To;
- Factor de redução da temperatura — ***ratio de arrefecimento*** — r;
- Um valor final do parâmetro de controlo — ***Temperatura de congelação*** especificado por um critério de paragem — α .

2 — **Um *número finito de transições*** para cada valor do parâmetro T, i. e.:

- comprimento finito para cada cadeia de Markov homogénea — L.

Para além da escolha deste conjunto de parâmetros que são necessários a qualquer implementação do *simulated annealing* há ainda que definir para cada problema específico, como acontecia na optimização local:

— O que é uma solução, S ;

— Qual o custo da solução S;

— Quais as soluções que estão na vizinhança de S ;

— Como determinar uma solução inicial, So;

— Como transitar de uma solução S para uma solução vizinha, S'.

Existe uma grande variedade de possibilidades de escolha do *arrefecimento escalonado* que resulta das combinações possíveis do conjunto de parâmetros que intervêm na sua definição. Daí que a qualidade das soluções, tempo de execução, convergência do algoritmo estejam fortemente dependentes daquelas escolhas. Acresce ainda o facto de o mesmo conjunto de parâmetros poder produzir soluções diferentes devido ao carácter aleatório do algoritmo. Assim, é difícil avaliar a eficiência do algoritmo ou compará-lo com outros de optimização local, enquanto não for possível determinar um arrefecimento escalonado óptimo e que seja independente da instância dos dados ou do tipo de problema a que é aplicado.

A procura de um conjunto óptimo de parâmetros de entrada é de grande interesse científico e tem servido de tema para vários artigos, vejam-se, por exemplo, Van Laarhoven e Aarts (1987), K.H. Hoffmann e P. Salamon (1990).

Um dos arrefecimentos escalonados mais utilizados nas implementações do *simulated annealing* é o *exponencial* que utilizámos na descrição do algoritmo e que foi o originalmente proposto por Kirkpatrick, Gelatt & Vechi (1982 ; 1983). A redução da temperatura é exponencial, sendo a k-ésima temperatura:

$$T_k = To * r^k. \qquad (4.3.1)$$

Outros processos de redução têm vindo a ser propostos, como, por

exemplo, o chamado *arrefecimento linear* em que a k-ésima temperatura é :

$$T_k = \frac{(C-k)}{C} * To \qquad (4.3.2)$$

com C constante, convenientemente escolhida. Aarts e Van Laarhoven (1985) apresentaram um outro processo alternativo, em que :

$$T_{k+1} = \frac{T_k}{(1+\alpha_k T_k)} \qquad (4.3.3)$$

com α_k uma constante convenientemente determinada. Este processo sugerido pelos resultados matemáticos sobre a convergência do algoritmo, foi desenvolvido, analisado e testado por E. Aarts e J. Korst (1989), tendo concluido que conduz a um algoritmo de execução em tempo polinomial, embora não permita dar qualquer garantia sobre o desvio da solução obtida em relação ao valor da solução óptima.

D. Johnson e al.(1989) utilizaram vários tipos de arrefecimento nas extensas experimentações que fizeram e concluiram não haver razão que justificasse, de uma forma clara, a substituição do arrefecimento exponencial, proposto por Kirkpatrick, Gelatt e Vechi por qualquer dos outros. Aarts e Van Laarhoven (1988) chegaram a conclusões semelhantes, i.e., que o tipo de arrefecimento utilizado não influencia significativamente os resultados, desde que seja escolhido cuidadosamente.

K.H.Heinz e P. Salamon (1990), procuram tratar o problema da determinação do conjunto parâmetros do *simulated annealing* como um problema de optimização. Começaram por escolher a função a optimizar, de entre várias possíveis, escolheram a energia média/custo médio final. Formalizaram o problema como um *problema de minimização da energia média final* num número finito de transições,

N, feitas de acordo com o critério de Metropolis. Testaram numéricamente o método para um espaço com três estados. Compararam os resultados obtidos por este método com os resultados obtidos utilizando os arrefecimentos linear e exponencial. Concluiram que este método apresentava, para o mesmo tempo de *annealing*, vantagem em relação aos anteriores, da ordem de 10^3, em termos de *energia média final*. Justificando-se portanto a continuação da investigação no sentido de o generalizar a espaços com mais estados.

4.3.2 — APLICAÇÃO AO PROBLEMA DE LOCALIZAÇÃO SIMPLES

Vamos descrever em seguida alguns aspectos particulares da implementação que fizemos do algoritmo *simulated annealing* aplicado ao SPL. Procuraremos fazer a descrição utilizando como referência os items enunciados na secção anterior.

1 — O SPL é um problema de minimização de custos fixos mais custos de atendimento. Não tem restrições no número de serviços.

SOLUÇÃO — S é qualquer subconjunto do conjunto de localizações possíveis : $1 \leq |S| \leq m$; a afectação de clientes a serviços é trivial, afectam-se clientes a origens de custo mínimo.

2 — O CUSTO é soma de custos fixos de instalação de serviços com custos de atendimento de clientes.

3 — VIZINHANÇA de S. Se I for o conjunto de localizações possíveis para os serviços e $|I| = m$, define-se vizinhança de S :

$$N(S) = \{ S' \subseteq I : |S'-S| \leq 1 \wedge |S - S'| \leq 1 \} .$$

Isto é, uma solução é vizinha de S, se for obtida de S pelo fecho de um serviço, a abertura de um novo serviço ou abertura de um e fecho de outro.

4 — A SOLUÇÃO INICIAL — So.

As soluções iniciais que utilizámos foram heurísticas e pseudo-aleatórias.

A heurística que utilizámos para obter as soluções iniciais foi a dual ascendente, de Erlenkotter. As razões desta escolha foram : por um lado, utilizar uma boa solução de partida pode ser de grande interesse quer em termos da qualidade da solução, quer em termos de tempo de execução; por outro lado, a qualidade da solução obtida por este método heurístico que, como se sabe, resolve o problema em grande número de casos.

Considerámos ainda outra hipótese, que comparámos com a primeira e que é mais usual nas implementações de *simulated annealing*, em que a solução de partida é formada por um conjunto de serviços, escolhidos aleatóriamente. Utilizámos para isso um gerador de números pseudo-aleatórios uniformemente distribuidos em [1,m]. A cardinalidade da solução que adoptámos foi igual à da solução obtida pelo método anterior.

5 — A TRANSIÇÃO de S para S'.

As vizinhanças podem ser de três cardinalidades diferentes, |S|, |I-S| e |S| * |I-S|. Escolhe-se, aleatóriamente, para qual destes tipos vamos transitar e depois disso, escolhe-se, aleatóriamente, um serviço para fechar, $s \in S$ ou um serviço para abrir, $s' \in (I - S)$ ou um par aleatório $(s , s') \in S \times (I - S)$ para se proceder à troca. A nova solução será, respectivamente:

$S' = S - (s)$, $s \in S$; $S' = S \cup (s')$, $s' \in I - S$; $S' = S - (s) \cup (s')$, $s \in S$, $s' \in S'$.

Calcula-se o custo de S', ajustando o custo fixo e adicionando-lhe os

custos da nova afectação de clientes a serviços.

No que diz respeito aos aspectos que são comuns a qualquer implementação do *annealing* resta apenas referirmo-nos à forma como escolhemos os parâmetros de *arrefecimento escalonado*.

4.3.2.1— ARREFECIMENTO ESCALONADO . RESULTADOS COMPUTACIONAIS

Para mostrar como implementámos o *simulated annealing* aplicado ao SPL vamos utilizar dois problemas, retirados da literatura, o problema de localização da 33 cidades, SPL(33x33) e o problema de localização das 57 cidades SPL(57x57). As matrizes das distâncias que utilizámos foram retiradas de Karg e Thompson (1964). As instâncias escolhidas foram :

Problema	SPL(33x33)	SPL(57x57)
	2000 → A_1	3000 → B_1
Custos fixos	2500 → A_2	4000 → B_2
	3000 → A_3	5000 → B_3

O critério de escolha foi o de terem necessitado de mais tempo de resolução pelo algoritmo de Erlenkotter.

Pelas razões que já apontámos anteriormente, optámos pelo *arrefecimento exponencial* proposto por Kirkpatrick, Gelatt e Vechi:

$$T_k = To * r^k$$

Vamos ainda introduzir o conceito de *ratio de aceitação a temperatura t* que iremos utilizar ao longo da exposição.

Define-se *ratio de aceitação à temperatura* t, como sendo

$$\chi(t) = \frac{\textit{número de transições aceites}}{\textit{número de transições propostas}} \Bigg| T = t$$

Temperatura inicial —To

Para escolher o valor do parâmetro inicial To, ensaiámos três métodos:

— Executar o programa para diversos valores iniciais da temperatura e escolher um valor que dê um ratio de aceitação aproximado a um valor previamente fixado;

— Escolher para To um valor suficientemente elevado de modo que praticamente todas as transições sejam aceites ;

— Executar o programa, partindo de um valor baixo da temperatura e aumentar progressivamente esse valor, multiplicando por um factor $r > 1$, até se atingir um valor T, tal que $\chi(T) \simeq 1$.

Optámos pelo primeiro porque os dois últimos nos conduziam a temperaturas muito elevadas e o tempo de execução aumentava muito, sem que se tivesse grande vantagem em termos de qualidade da solução final.

Da experiência que tivemos verificámos ainda que partir de temperaturas iniciais correspondentes a ratios de aceitação superiores a 0.5, não era vantajoso, tendo em conta o aumento de tempo de execução do algoritmo versus qualidade da solução obtida.

Nos quadros 4.2 e 4.3 apresentamos o ratio de aceitação, χ(To), para

diferentes valores da temperatura inicial, To. As soluções iniciais são obtidas pela heurística dual ascendente no quadro 4.2 e pseudo-aleatórias no quadro 4.3.

To\ P	A_1	A_2	A_3	B_1	B_2	B_3
1000	12.1	14.6	16.7	10.7	9.6	5.9
2000	25.0	19.8	19.3	19.5	11.7	12.4
4000	40.4	32.7	37.3	29.2	21.8	20.5
6000	49.7	44.2	42.0	32.4	31.4	29.3
10000	53.4	52.3	55.3	42.6	43.7	37.8
15000	69.1	66.8	61.8	52.9	52.1	49.3

Quadro 4.2

To\ P	A_1	A_2	A_3	B_1	B_2	B_3
1000	29.1	22.5	19.7	15.8	14.8	11.5
2000	35.0	32.2	26.6	22.9	20.5	20.1
4000	45.5	45.9	39.5	32.8	33.9	33.1
6000	56.3	50.6	46.9	39.7	37.3	36.5
10000	63.7	59.4	60.7	52.9	50.2	47.7
15000	69.1	70.1	69.0	58.5	57.2	57.7

Quadro 4.3

Como pode observar-se, os valores obtidos partindo de soluções pseudo-aleatórias são superiores aos obtidos partindo de soluções obtidas pela heurística dual ascendente, como seria de esperar.

À medida que a temperatura vai baixando a percentagem de soluções aceitas diminui e simultâneamente o custo médio das soluções aceites vai baixando também. Nas fig. 4.1 e fig. 4.2 representam-se os gráficos de $\chi(t)$ e custo médio das soluções aceites para o problema SPL (33x33 ; f_i = 2000) e para 100 valores decrescentes da temperatura.

Apresentam-se a seguir os quadros dos resultados que obtivemos para valores diferentes de To escolhidos a partir de diferentes valores de χ(To) para este mesmo conjunto de problemas. Consideram-se duas hipóteses: soluções iniciais pseudo-aleatórias, designadas no quadro por hipótese H—(a) ; soluções iniciais obtidas pela heurística dual ascendente, designadas por hipótese H—(b). A coluna designada por Δ% representa o "melhor" desvio relativo em relação à solução óptima : $\Delta = \frac{\bar{z} - z^*}{z^*} * 100$. A "média" representa a média das soluções aceites para o mesmo número de soluções geradas.

Problema A_1:

To	pior	média	melhor	óptima	Δ%	H
10000	42423	27625	21220	20363	4.2	(a)
	48126	24259	20790	20363	2.1	(b)
6000	34049	24318	20863	20363	2.46	(a)
	40442	23723	20726	20363	1.78	(b)
4000	30946	23150	20756	20363	1.93	(a)
	34077	22992	20696	20363	1.63	(b)
2000	25786	23188	20766	20363	1.97	(a)
	26296	22952	20363	20363	0.0	(b)

Quadro 4.4

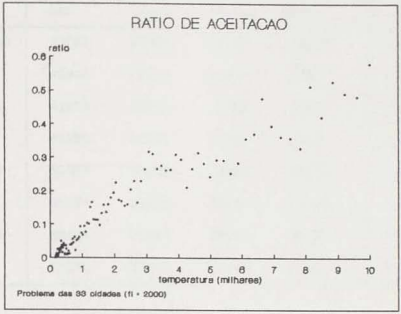
RATIO DE ACEITACAO
ratio
0.6
0.5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8 9 10
temperatura (milhares)
Problema das 88 cidades (fi = 2000)

fig. 4.1

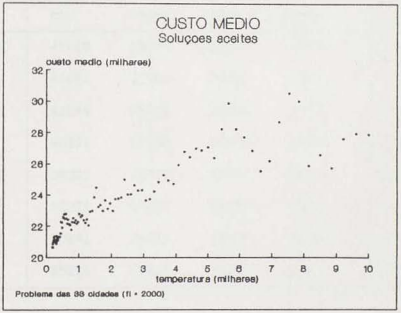
CUSTO MEDIO
Soluçoes aceites
custo medio (milhares)
32
30
28
26
24
22
20
0 1 2 3 4 5 6 7 8 9 10
temperatura (milhares)
Problema das 88 cidades (fi = 2000)

fig. 4.2

Problema A_2:

To	pior	média	melhor	óptima	Δ%	H
10000	43682	29930	22252	22127	0.5	(a)
	46946	26213	22474	22127	1.56	(b)
6000	41674	26272	22361	22127	1.06	(a)
	42651	25212	22361	22127	1.06	(b)
4000	33309	23095	22462	22127	1.52	(a)
	35077	22413	22463	22127	1.52	(b)
2000	28184	23848	22290	22127	0.74	(a)
	35077	23857	22252	22127	0.56	(b)

Quadro 4.5

Problema A_3:

To	pior	média	melhor	óptima	Δ%	H
10000	47620	29009	24316	23474	3.6	(a)
	44861	27892	23790	23474	1.35	(b)
6000	42544	27430	24294	23474	3.5	(a)
	40371	27355	23474	23474	0.0	(b)
4000	34324	26949	23474	23474	0.0	(a)
	36321	26167	23474	23474	0.0	(b)
2000	30341	26344	23474	23474	0.0	(a)
	29216	25418	23474	23474	0.0	(b)

Quadro 4.6

Problema B_1:

To	pior	média	melhor	óptima	Δ%	H
10000	60154	38945	32684	32136	1.7	(a)
	54051	37793	32452	32136	0.9	(b)
6000	51752	37830	33002	32136	2.7	(a)
	61635	38600	32452	32136	0.9	(b)
4000	45379	35928	32547	32136	1.3	(a)
	46751	35632	32452	32136	0.9	(b)
2000	41289	35455	33126	32136	3.1	(a)
	39742	33849	32394	32136	0.8	(b)

Quadro 4.7

Problema B_2

To	pior	média	melhor	óptima	Δ%	H
10000	66476	41606	35909	35547	1.02	(a)
	54070	39489	35623	35547	0.21	(b)
6000	52495	39601	35692	35547	0.41	(a)
	54445	38477	35623	35547	0.21	(b)
4000	46663	38763	36309	35547	2.14	(a)
	51923	39082	35902	35547	0.99	(b)
2000	46335	38076	35547	35547	0	(a)
	39887	38694	35907	35547	1.01	(b)

Quadro 4.8

Problema B_3:

To	pior	média	melhor	óptima	Δ%	H
10000	67440	45367	39344	38547	2.07	(a)
	60314	44063	38909	38547	0.93	(b)
6000	61166	43413	39025	38547	1.24	(a)
	56051	38969	38742	38547	0.51	(b)
4000	53808	41807	38742	38547	0.51	(a)
	53399	41905	38742	38547	0.51	(b)
2000	46218	40740	38742	38547	0.51	(a)
	48133	40764	38742	38547	0.51	(b)

Quadro 4.9

Pudémos observar que para o mesmo número de *runs* a qualidade das soluções obtidas, partindo de temperaturas muito elevadas é inferior à das soluções obtidas para temperaturas mais baixas. Para além disso, partir de temperaturas muito elevadas provoca, em geral, grandes acréscimos de tempo de execução.

É de notar também que a qualidade das soluções obtidas partindo de soluções iniciais não aleatórias, (b), foi dum modo geral bastante superior à das soluções obtidas partindo de soluções iniciais aleatórias, (a) como pode ver-se pela coluna Δ% .

Factor de redução da temperatura (*Ratio de arrefecimento*) - r

O parâmetro r controla a forma como se processa o arrefecimento. Testámos vários valores de r e verificámos não haver interesse em fazer reduções bruscas, i.e., considerar valores de $r < 0.7$, o que confirma o que vem sendo dito na literatura. Podemos dizer que para valores de $r < 0.7$ e utilizando como solução inicial a solução obtida pela heurística dual ascendente não obtivemos qualquer melhoria em 5 runs de annealing. No quadro 4.10 apresentamos resultados obtidos para diferentes valores de r .

Concluimos também das experimentações que fizemos que os valores de r devem situar-se no intervalo, $0.7 \leq r \leq 0.95$. Valores superiores a 0.95, aumentavam significativamente os tempos de execução sem que daí resultasse melhoria significativa para os valores das soluções obtidas.

Comprimento das cadeias — L

O parâmetro L define o número de iterações a uma dada temperatura, i. e. o número máximo de transições a uma dada temperatura. Tomámos para valor de L a dimensão máxima das vizinhanças, uma vez que são de dimensões diferentes. Experimentámos outros valores de L , como por exemplo, L = 3*m e ainda uma dimensão variável que vai aumentando à medida que a temperatura vai baixando. Com este procedimento, procura-se aumentar o tempo de permanência em temperaturas mais baixas devido ao facto de a probabilidade de aceitação diminuir à medida que a temperatura vai baixando, veja-se fig 4.1. Esta escolha do parâmetro L parece-nos bastante adequada porque permite obter melhores resultados, como já referimos. Pensamos que conjugar um valor de r não muito

elevado com um parâmetro L, adaptativo, que permita ajustar o comprimento da cadeia à medida que a temperatura vai baixando é uma boa forma de proceder.

L = I N I

P\ r	0.7	0.8	0.9	0.95	ncd
A_1	21285	20783	21219	21073	5
	20625	20756	21037	20825	10
	20509	20756	20933	20825	15
A_2	22989	23020	22861	23381	5
	22989	22361	22391	22418	10
	22127	22361	22391	22418	15
A_3	24599	24858	24858	24858	5
	23999	24294	23474	23474	10
	23474	23627	23474	23474	15
B_1	33724	32684	33061	33593	5
	32547	32684	32562	33066	10
	32547	32684	32562	32881	15
B_2	36354	35909	35962	36354	5
	35623	35909	35962	36354	10
	35623	35909	35692	35649	15
B_3	38968	39876	39934	39070	5
	38617	39344	38918	39070	10
	38617	38742	38881	39070	15

Quadro 4.10

Este procedimento pode levar a que se passe muito tempo a analisar

soluções, por vezes repetidas.

$L_k = (1.1)^K * |N|$

P\ r	0.7	0.8	0.9	0.95	ncd
	21285	20783	21219	21073	5
A_1	20625	20726	20813	20652	10
	20363	20393	20363	20609	15
	22989	23020	22861	23381	5
A_2	22127	22252	22474	22474	10
	22127	22252	22252	22299	15
	24599	24858	24858	24858	5
A_3	23474	23474	23474	23474	10
	23474	23474	23474	23474	15
	32814	32804	33097	32779	5
B_1	32156	32658	32290	32779	10
	32136	32547	32156	32779	15
	35909	35962	36270	36309	5
B_2	35547	35777	36270	36309	10
	35547	35617	35623	35617	15
	39344	38742	38617	39070	5
B_3	38742	38742	38617	38981	10
	38617	38547	38617	38981	15

Quadro 4.11

No quadro 4.11 são apresentados os melhores valores das soluções obtidas para diversos valores de *n*, utilizando comprimentos de cadeia adaptativos: $L_k = (1.1)^k * |N|$ em que *k* é contador do número de reduções da temperatura e *ncd* representa o número de cadeias.

Na passagem do quadro 4.10 para o quadro 4.11 foram mantidos os valores da temperatura inicial para podermos analisar o efeito produzido pela alteração na dimensão das cadeias.

No quadro 4.12 apresentam-se os desvios percentuais em relação à solução óptima das soluções obtidas nos dois quadros anteriores. São comparados os valores obtidos para $L = |N|$ e $L_k = |N| * (1.1)^k$ que serão designadas no quadro por (a) e (b), respectivamente.

Da observação do quadro 4.12 pode concluir-se que as cadeias de comprimento adaptativo produzem, em geral, soluções de melhor qualidade. Pode também verificar-se que, para cadeias de comprimento adaptativo $L_k = |N| * (1.1)^k$ e com 10 reduções de temperatura, se conseguem resultados iguais ou superiores aos que se obtêm com 15 reduções e cadeias de comprimento $L = |N|$ (75% dos casos). Isto pode significar que sem aumentar muito o tempo de execução se conseguem melhores resultados utilizando cadeias de comprimento $L_k = |N| * \beta^k$.

O critério de escolha para o factor de ampliação, β, que começámos por adoptar, foi o de utilizar um factor de ampliação igual ao inverso do factor de redução da temperatura. Verificámos que um tal critério levava a que factores de ampliação da cadeia muito elevados nem sempre produziam melhores resultados e aumentavam consideràvelmente o tempo de execução. Daí a escolha de um factor de ampliação mais moderado, β=1.1.

	Δ% (melhor desvio percentual)							
	r	0.7		0.8		0.9		0.95	
	nod	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
A_1	5	4.5	4.5	2.06	2.06	4.2	4.12	3.49	2.68
	10	1.29	1.3	1.93	1.78	3.31	2.21	2.27	1.42
	15	0.71	0	1.93	0.15	2.8	0	2.27	1.21
A_2	5	3.89	1.68	4.03	4.74	3.32	3.10	5.67	1.57
	10	3.89	0	1.06	0.56	1.19	1.57	1.32	1.57
	15	0	0	1.06	0.56	1.19	0.56	1.32	0.78
A_3	5	4.79	4.53	5.89	1.65	5.89	3.01	5.89	3.22
	10	2.24	0	3.49	0	0	0	0	0
	15	0	0	0.65	0	0	0	0	0
B_1	5	4.94	2.11	1.7	2.08	2.87	2.99	4.53	2.0
	10	1.28	0.06	1.7	1.62	1.33	0.48	2.89	2.0
	15	1.28	0	1.7	1.28	1.33	0.06	2.32	2.0
B_2	5	2.27	1.02	1.02	1.17	1.17	2.03	2.27	2.14
	10	0.2	0	1.02	0.65	1.17	2.03	2.27	2.14
	15	0.21	0	1.02	0.19	0.41	0.21	0.29	0.2
B_3	5	1.09	2.07	3.45	0.51	3.6	0.18	1.36	1.36
	10	0.18	0.51	2.07	0.51	0.96	0.18	1.36	1.13
	15	0.18	0.18	0.51	0	0.87	0.18	1.36	1.13

Quadro 4.12

Apresentamos a seguir um quadro de resultados em que se comparam os resultados obtidos para diferentes valores de L.

Como pode observar-se no quadro 4.13 os melhores resultados foram obtidos para comprimentos de cadeia adaptativos. No entanto um factor de ampliação da cadeia muito elevado nem sempre produziu melhores resultados e aumenta consideràvelmente o tempo de execução.

P\ L	3*m	|N|	|N|*$(1.1)^k$	|N|*$(1.25)^k$
A_1	20929	20756	20726	20398
A_2	22449	22989	**22127***	**22127***
A_3	23790	24294	**23474***	**23474***
B_1	32546	32547	32156	32598
B_2	36339	35909	35617	36103
B_3	38617	39344	38742	38617

Quadro 4.13

|N| – cardinalidade máxima esperada da vizinhança ;

k – é contador do número de reduções da temperatura.

z^* – valor óptimo da solução.

Para evitar que sejam geradas cadeias de grande dimensão definimos um

limite superior para $L_k = |N| * \beta^k \leq \Gamma$. Em resultado das experiências que fizemos parece-nos ser vantajosa uma solução de compromisso entre o valor do parâmetro r e um factor adaptativo para o comprimento da cadeia, podendo assim utilizar-se valores de r mais baixos, o que permite reduzir o tempo gasto sem prejudicar a qualidade da solução.

Critério de paragem — α

O critério de paragem do algoritmo corresponde à *temperatura de congelação* do processo físico. Utilizámos para critério de paragem do algoritmo um parâmetro a que chamámos α e que representa a percentagem de soluções aceites ao longo de uma cadeia. Experimentámos um critério que consistia em declarar o processo congelado quando ao longo de uma cadeia a percentagem de aceitação fosse inferior a 2% , i.e. $\alpha < 0.02$. Em resultado da experiência verificámos que por vezes acontece que para determinados valores da temperatura, ocorrem, pontualmente, valores baixos da percentagem de aceitação e depois para valores inferiores de T as percentagens aumentam, podendo ainda encontrar-se valores com algum significado. Pareceu-nos mais adequado optar por um critério de paragem, proposto por Johnson e al.(1989) e que consiste em utilizar um contador para o número de vezes que cada cadeia é percorrida com uma percentagem de aceitação ≤ 0.02 . Este contador é feito igual a zero sempre que haja alguma transição para uma solução melhor. Se o contador atingir um valor préviamente fixado, sem que haja melhoria, declara-se o processo congelado e o algoritmo pára. No nosso caso adoptámos o valor 3 para o contador anteriormente definido.

Na nossa implementação introduzimos ainda um parâmetro, *nru*, que define

à partida quantas vezes se pretende executar o *annealing* .

Em conclusão, a escolha dos parâmetros que são mais adequados para cada problema é de grande importância para a implementação do algoritmo e varia de instância para instância, ainda que a dimensão do problema seja a mesma.

Para os problemas que temos vindo a testar a escolha que nos pareceu mais adequada foi a que a seguir apresentamos no quadro 4.14:

Problema	To	r	L	nru	L	nru
A_1	4000	0.8	INI	10	INI $*\ \beta^k$	5
A_2	5000	0.7	INI	10	INI $*\ \beta^k$	5
A_3	4000	0.8	INI	10	INI $*\ \beta^k$	5
B_1	10000	0.8	INI	10	INI $*\ \beta^k$	5
B_2	10000	0.7	INI	10	INI $*\ \beta^k$	5
B_3	10000	0.8	INI	10	INI $*\ \beta^k$	5

To : $\chi(To) \simeq 0.4$ $0.7 \leq r \leq 0.95$ $L \leq \overline{L}$ $\beta = 1.1$ $0 \leq k$

Quadro 4.14

O número de *runs*, nru=10 foi adoptado para cadeias de comprimento L=INI e nru = 5 para cadeias de comprimento L = INI $*\ \beta^k$. O critério de paragem foi o que anteriormente descrevemos.

Nos quadros 4.15 e 4.16 são apresentados os resultados obtidos para este conjunto de parâmetros e para o conjunto de problemas que temos vindo a testar.

No quadro 4.15 são considerados resultados, utilizando soluções de partida obtidas pela heurística dual ascendente; no quadro 4.16 os resultados referem-se a soluções de partida pseudo-aleatórias. Foram consideradas cadeias de comprimento L = |N| e o número de *runs* do *simulated annealing* foi nru=10. Apresentam-se ainda os tempos médios de execução, t_m, e o número de vezes, f, que foi obtida a solução óptima.

P	pior	média	melhor	nru	f	t_m
A_1	20640	20529	20363*	5	1	
	20799	20571	20363*	10	3	196.72
A_2	22652	22291	22127*	5	2	
	22652	22287	22127*	10	4	260.04
A_3	23752	23529	23474*	5	4	
	23799	23550	23474*	10	7	197.12
B_1	32576	32505	32156	5	0	
	32985	32450	32136*	10	1	598.94
B_2	35962	35673	35547*	5	1	
	36644	35813	35547*	10	2	455.22
B_3	39427	38993	38617	5	0	
	39427	38847	38547*	10	2	467.31

Quadro 4.15

P	pior	média	melhor	nru	f	t_m
A_1	20914	20568	**20393**	5	0	
	20914	20574	**20363***	10	1	237.52
A_2	22652	22291	**22127***	5	3	
	22652	22287	**22127***	10	5	213.74
A_3	23752	23529	**23474***	5	4	
	23799	23550	**23474***	10	8	276.02
B_1	32881	32385	**32136***	5	1	
	35374	32818	**32136***	10	2	944.52
B_2	36070	35860	**35617**	5	0	
	36070	35777	**35547***	10	1	853.51
B_3	38742	38679	**38547***	5	1	
	39963	38822	**38547***	10	1	815.47

Quadro 4.16

Da observação dos quadros 4.15 e 4.16 podemos tirar algumas conclusões:

– Os dois algoritmos permitiram obter as soluções óptimas de todos os problemas em 10 runs;

– Os resultados obtidos partindo de soluções iniciais não aleatórias foram, em geral, superiores;

– Para quatro destes problemas foi obtida a solução óptima em 5 runs, quando foram utilizadas soluções iniciais não aleatórias;

– Os tempos de execução[5] (não incluem input, mas incluem output) foram inferiores aos que foram obtidos partindo de soluções pseudo-aleatórias.

– O pior desvio em relação à solução óptima foi, em geral, inferior ao obtido partindo de soluções pseudo-aleatórias.

Nos quadros 4.17 e 4.18 apresentam-se resultados comparativos dos valores das melhores soluções obtidas pelas duas implementações do *simulated annealing* com os valores das soluções obtidas pelos algoritmos de optimização local correspondentes. Comparam-se ainda esses resultados com os que foram obtidos pelas heurísticas Dual Ascendente e Ajustamento do Dual. Os algoritmos de optimização local (LOCAL1/LOCAL2) apenas diferem dos correspondentes algoritmos *simulated annealing* (ANNEAL1/ANNEAL2) no critério de aceitação.

No quadro 4.17 comparam-se os resultados obtidos para o mesmo número aproximado de soluções geradas. No quadro 4.18 comparam-se os resultados obtidos da utilização dos diversos algoritmos.

---

[5]Todas as implementações foram feitas em Fortran 77 num Microcomputador IBM-compatível, AT 286, tendo-se utilizado o Microsoft Fortran Compiler versão 4.0. Para gerar os números aleatórios foi utilizado um gerador de números pseudo-aleatórios uniformemente distribuidos em [0,1], apresentado em "Numerical Recipes - The Art of Scientific Computing " por William H.Press, Brian Flannery, Saul Teukolsky, Cambrige University Press 1986.

P	ANNEAL1	LOCAL1	ANNEAL2	LOCAL2	AJUSTA-DUAL
A_1	20775	20363*	20929	20609	**20503**
A_2	22651	22127*	**22270**	22290	23137
A_3	23474*	23861	23790	**23752**	23474*
B_1	**32452**	32136*	32547	32136*	32136*
B_2	35649	35547*	**35623**	35751	35547*
B_3	38623	38547*	38547*	38751	38617

Quadro 4.17

P	ANNEAL1	LOCAL1	ANNEAL2	LOCAL2	AJUSTA-DUAL
A_1	20363*	20363*	20363*	20609	**20503**
A_2	22127*	22127*	22127*	**22290**	23137
A_3	23474*	23861	23861	**23752**	23474*
B_1	**32156**	32314	32136*	32270	32136*
B_2	35547*	36008	**35617**	36142	35547*
B_3	**38617**	38547*	38547*	38751	**38617**

Quadro 4.18

ANNEAL1 / LOCAL1 solução inicial : So — primal - dual

ANNEAL2/ LOCAL2 solução inicial : So — pseudo-aleatória

AJUSTA - DUAL heurísticas Dual Ascendente e Ajustamento do Dual.

4.4 — CONCLUSÕES

O *Algoritmo Simulated Annealing* parece ser uma boa alternativa ao algoritmo Ajustamento Primal-Dual porque permitiu obter soluções óptimas em maior número de casos nos problemas que testámos.

Quando confrontado com os algoritmos de optimização local correspondentes permitiu obter melhores resultados o que nos leva a concluir que a introdução do critério de Metropolis possibilita a melhoria da qualidade das soluções obtidas, embora com acréscimos significativos de tempo.

A eficiência do algoritmo está fortemente dependente da escolha dos parâmetros iniciais. A escolha que fizemos para o conjunto de problemas que testámos e que resultou da experimentação computacional, estará longe de ser a escolha óptima.

Pensamos que a investigação nesta área deve ser dirigida no sentido de optimizar o arrefecimento escalonado.

No que se refere à escolha da solução inicial parece-nos que uma solução não aleatória apresenta vantagens em termos de tempo e qualidade das soluções.

Pensamos ainda que, à medida que a dimensão do problema aumentar, se tornará mais evidente o interesse da aplicação do *Algoritmo Simulated Annealing*.

O *Algoritmo Simulated Annealing* permitirá provavelmente obter resultados muito mais significativos em outros problemas de optimização combinatória para os quais não existam algoritmos tão "bons" como para o SPL, para o qual o DUALOC de Erlenkotter (1978) continua a ser um excelente algoritmo de resolução.

REFERÊNCIAS BIBLIOGRÁFICAS

AKINC,U. ; KHUMAWALA,M.; (1977) . "An Efficient Branch and Bound Algorithm for the Capacited Warehouse Location Problem ", *Management Science*,vol.**23**, pp. 585-594.

AIKENS,C.H. ; (1985) "Facility Location Models for Distribution Planning", *European Journal of Operations Research*, **vol.22**, pp. 263-279.

AARTS,E.; KORST,J.; (1989) "Simulated Annealing and Boltzmann Machines. A Stochastic Approach to Combinatorial Optimization and Neural Computing", *J. Wiley & Sons Inc.*.

BALINSKI, M. L.; (1965) . "Integer Programming: methods, uses, computation", *Management Science*,vol.**12**, pp. 253-313.

BEASLEY,J.E.; (1988) "An Algorithm for Solving Large Capacited Warehouse Location Problems "*European Journal of Operational Research*, **vol.33**, pp. 314-325.

BERGE,C; (1966) "Espaces Topologiques. Fonctions Multivoques", *Dunod, Paris*.

BILDE,O.;KRARUP,J. ; (1977) "Sharp Lower Bounds and Efficient Algorithms for The Simple Plant Location Problem", *Annals of Discrete Mathematics*,**vol.1**, pp. 79-97.

BRANDEAU,M.L.;CHIU,S.S.; (1989) "An Overview of Representative Problems in Location Research", *Management Science*, **vol.35**, No.6, pp. 645-674.

BURKARD, R.E.; RENDL, F.; (1984) "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems ", *European Journal of*

Operational Research, **vol.127**, pp.169-174.

CERNY,V.; (1985) "Thermodinamical Approach to the Traveling Salesman Problem: an Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, **vol.45** , pp. 41-51.

CHRISTOFIDES,N.;(1975) "Graph Theory: An Algorithmic Approach",*Academic Press*, New York.

CHURCH, R.; REVELLE, C.;(1976) "Theoretical and Computational Links Between the p-Median, Location Set-Covering and the Maximal Covering Location Problem", *Geographical Analysis*, **vol.8**, pp. 406-415.

COOLINS,N.E.;(1988)"SimulatedAnnealing-AnAnnotated Bibliography",*Computer*, **vol.21**.

COOK,S.A.;(1971a) "TheComplexityof Theorem-Proving"*Proceedings of the 3rd Annual ACM Symposium on The Theory of Computing Machinery*, (A.C.M., New York), pp.151-158.

CORNUEJOLS; FISHER; NEMHAUSER; "The Uncapacited Plant Location Problem", *Management Science Research Report*, No MSRR 493, 1-73.

DOMSCHKE, W.; DREXEL, A.; (1985) " Location and Layout Planning: An International Bibliography", Berlin : *Springer Verlag*.

EFROYMSON,M.A.; RAY,T.L. ; (1966) "A Branch-and-Bound Algorithm for Plant Location", *Operations Research*,**vol.14**, pp. 361-368.

EGLESE, R. W.; RAND, G.K.; (1987) "Conference Seminar Timetabling",*Journal of the Operational Research Society*,vol.38, pp. 591-598.

ERLENKOTTER,D. ; (1978) "A Dual-Based Procedure for Uncapacited Facility Location", *Operations Research*,**vol.26**, pp.992-1010.

FISHER, M. L.; JAIKUMAR, R. ; WASSENHOVE,V.L.(1986) "A Multiplier Ajustment Method for the Generalized Assignment Problem", *Management Science*, **vol.32**, pp. 1095 - 1103.

FRANCIS,R.L.; GOLDSTEIN, J.M.; (1974) "Location Theory: A Selective Bibliography" , *Operations Research*, **vol.22**, pp. 400-410.

FRANCIS,R.L.; McGINNIS,L.F.;WHITE,J.A.; (1983) "Locational Analysis", *European Journal of Operational Research*,vol.**12**, pp. 220-252.

GAREY,M,R.;JOHNSON,D.S.; (1979) "Computer and Intractability: A Guide to the Theory of NP-Completeness", *Freeman* , San Francisco .

GEOFFRION,A.M.; (1974)"Lagrangean Relaxation for Integer Programming" , *Mathematical Programming Study* 2,pp. 82-114.

GIDAS,B.; (1985)"Nonsationary Markov Chains and Convergence of the Annealing Algorithm" , *Journal of Statistical Physics*, **vol.39**, Nos.1/2, pp. 73-131.

GUIGNARD,M; (1988) "A Lagrangean Dual Ascent Algorithm for Simple Plant Location Problems", *European Journal of Operational Research*,vol.**35**,pp. 193-200.

HALPERN,J.;MAIMON O.; (1982) "Algorithms for m-Center Problems: A Survey", *European Journal of Operational Research*,vol.**10**,pp. 90-99.

HANSEN,P.; LABBÉ,M.; PEETERS,D.; THISSE,J.; (1987) "Single Facility Location on networks",*Annals of Discrete Mathematics*,**vol.31**,pp. 113-146.

HANSEN,P.; LABBÉ,M.; PEETERS,D.; THISSE,J.; (1987) "Facility Location Analysis", *CORE Reprint, Université Catholique de Louvain*, No. 747,pp. 1-70.

HAKIMI,S.L.; (1964) "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph",*Operations Research*,**vol.12**, pp. 450-

459.

JOHNSON, D.S. ; ARAGON,C.R. ; McGEOCH,L.A.; SCHEVON, C.,(1989) "Optimization by Simulated Annealing : An Experimental Evaluation; Part I, graph partirioning"*Operations Research*,vol.37, No. 6, pp. 865-892.

KARG,R.L.; THOMPSON,G.L.(1964) "A Heuristic Approach to Solving Travelling Salesman Problems", *Management Science*,vol.10, pp.225-248.

KARKAZIS,J.; BOFFEY,T.B.; (1981) " The Multi-commodity Facilities Location Problem"*Journal of Operational Research Society*", vol.32, pp.803-814.

KARP,R.M.; (1972),"Reductibility among Combinatorial Problems", in *Complexity of Computer Computations*,Eds: R. E. Miller e J. W. Thatcher, Plenum Press, New York, pp. 85-103.

KAUFMAN,L.,EEDE,M.V.,HANSEN,P.; (1977) "A Plant and Warehouse Location Problem", *Operational Research Quaterly* ,vol.28,No.3,pp.547-554.

KHUMAWALA,M.;(1972) "An Efficient Branch- and- Bound Algorithm for The Warehouse Location Problem", *Management Science*,vol.18, pp.718-731.

KIRKPATRICK,S. ; GELATT,C.D. Jr.; VECCHI,M.P.; (1983), "Optimization by Simulated Annealing",*Science*,vol.220, pp. 671-680.

KIRKPATRICK, S.; (1984),"Optimization by Simulated Annealing: Quantitative Studies", *Journal of Statistical Physics*, vol.34, Nos. 5/6, pp. 975-986.

KRARUP,J.,PRUZAN,P.M.;(1983) "The Simple Plant Location Problem: Survey and Syntesis",*European Journal of Operational Research*,vol.12, pp. 36-81.

KUEHN,A.A.; HAMBURGER,M.J.; (1963) "A Heuristic Program for Locating Warehouses",*Management Science*,vol.9,pp 643-666.

LEMKE,c.e.; SALKIN,H.M.; SPIELBERG,K.;(1971), "Set Covering by Single-branch Enumeration with Linear Programming Subproblems", *Operations Research*, vol.19, pp.998-1022.

LOVE,R.F.; MORRIS,J.G.; WESOLOWSKY,G.O. (1988) "Facilities Location - Models & Methods", *North - Holland* .

MANNE, A. S.;(1964) "Plant Location under Economies-of-Scale-Decentralization and Computation ", *Management Science*, vol.11, pp. 213-235.

METROPOLIS,N ; ROSENBLUTH,A.; ROSENBLUTH,M. ; TELLER,A.; TELLER,E.; (1953), "Equation of State Calculations by Fast Computing Machines", *Journal of Chemical Physics*, vol.21, pp.1087-1092.

MINIEKA,E.; (1970),"The m-CenterProblem",SIAM Review,vol.12, pp. 138-139.

MORRIS, J.G ; (1978) "On the Extent to Which Certain Fixed-Charge Depot Location Problems Can Be Solved by LP", *Journal of Operations Research Society*,vol.29 No.1, pp. 71-76.

NEEBE,A.; (1988)"A Procedure for Locating Emergency-Service Facilities for All Possible Response Distances", *Journal of Operations Research Society*, vol.39, pp. 743-748.

NEEBE,A.; KHUMAWALA,B.; (1981)"An Improved Algorithm for the Multi-commodity Location Problem", *Journal of Operations Research Society*, vol.32, pp. 143-149.

NEMHAUSER, G. ; WOLSEY, L.; (1988) " Integer and Combinatorial Optimization", *J. Wiley & Sons Inc.*.

PAPADIMITRIOU,C.H.; STEIGLITZ,K.; (1982) "Combinatorial Optimization: Algorithms and Complexity", *Prentice-Hall Inc. Englewood Cliffs*, New Jersey.

REVELLE, C.; SWAIN,R.S.;(1970) "Central Facilities Location "*Geographical Analysis*, vol.2, pp. 30-42.

RODDMAN, G.M.; SCHWARZ, L.B.; (1975)"Optimal and Heuristic Facility Phase-Out Strtegies"*AIIE Trans.*, vol.7, No.2, pp. 177-184.

SÁ,G. ; (1969) " Branch-and-Bound and Aproximate Solutions to The Capacitated Plant-Location Problem,"*Operations Research*,vol.17, pp. 1005-1016.

SASAKI, G.H.; HAJEK,B.; (1988) "The Time Complexity of Maximum Matching by Simulatad Annealing ",*Journal of the Association for Computing Machinery*, vol.35, No. 2 pp. 387-403.

SCHRAGE, L.; (1975) " Implicit Representation of Variable Upper Bounds in Linear Programming ,"*Mathematical Programming Study* ", 4, pp. 118-132.

SECHEN,C.; SANGIOVANNI-VICENTELLI; (1985) "The TimberWolfe Placement and Routing Package", *IEEE Journal of Solid-State Circuits*,vol.30,pp. 510-522.

SPIELBERG,K.; (1969) "Algorithms for The Simple Plant- Location Problem with Some Side Conditions",*Operations Research*, vol.17, pp. 85-111.

STOLLSTEIMER,J.F.; (1963)"A working Model for Plant Numbers and Locations",*Journal of Farm Economics*, vol 45, pp. 631-645.

TANSEL,B.C.; FRANCIS,R.L.; LOWE,T.J.; (1983) "Location on Networks:A Survey Part I: The p-Center and p-Median Problems", *Management Science*,vol.29,pp. 482-497.

TCHA, DONG-WAN ; RO, HYUNG-BONG ; YOO,CHUN-BEON; (1988) "A Dual-Based Add Heuristic for Uncapacited Facility Location", *Journal of Operations Research Society*, vol.39 No.9, pp. 873-878.

TCHA, DONG-WAN; LEE, BUM-IL; (1984) "A Branch-and-Bound Algorithm for

The Multi-Level Uncapacited Facility Location Problem", *European Journal of Operational Research*, **vol.18**, pp. 35- 43.

TOREGAS,C.;SWAIN,R.;REVELLE,C.;BERGMAN,L.;(1971)"The Location of Emergency Service Facilities",*Operations Research*, **vol.19**, pp.1363-1373.

VAN LAARHOVEN, P. J. M. ; AARTS, H. L. ; (1987) "Simulated Annealing: Theory and Practice", *Kluwer Academic Publishers*, Dordrecht, The Netherlands.

VAN ROY,T. J.;"A Cross Decomposition Algorithm for Capacited Facilinty Location ", *Operations Research*, **vol.34**, pp. 145-163.

VAN ROY; ERLENKOTTER,D. ; (1982) "A Dual-Based Procedure for Dynamic Facility Location", *Management Science*, **vol.28**, No.10, pp. 1091-1105.

WARSAWKI, A.; (1973) "Multi-dimensional Location Problems" *Operational Research Quaterly* ,**vol.24**, pp.165-179.

WARSAWKI, A.;PEER, S..; (1973) "Optimizing the Location of Facilities on Building Site", *Operational Research Quaterly* ,**vol.24**, pp.35-44.

WONG,R.T.;(1984) "A Dual-Ascent Approach for Steiner Tree Problems on A Directed Graph",*Mathematical Programming*,vol.**28**, pp. 271-287.

WONG,R.T.;BALAKRISSHNAN,A.;MAGNANTI,T.L.; (1989) "A Dual-Ascent Procedure for Large-Scale Uncapacited Network Design", *Operations Research*, **vol.37**, No. 5, pp.716-740.

ZIMMERMANN,R.;(1967) "A Branch and Bound Algorithm for Depot Location", *Metra* **6**, pp.661-674.