

ALGORITMO EXATO E HEURÍSTICAS PARA O  
PROBLEMA DE LOCALIZAÇÃO

Geraldo Robson Mateus

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

Aprovada por:



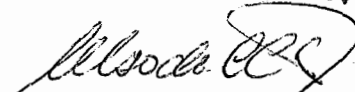
Prof. Cláudio Thomás Bornstein  
(Presidente)



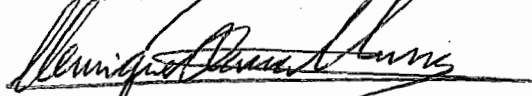
Prof. Nelson Maculan Filho



Prof. Roberto Diêguez Galvão



Prof. Celso da Cruz Carneiro Ribeiro



Prof. Henrique Pacca Loureiro Luna

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 1986

MATEUS, GERALDO ROBSON

Algoritmo Exato e Heurísticas para o Problema de Localização  
[Rio de Janeiro] 1986.

XI, 152 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas  
e Computação, 1986).

Tese - Universidade Federal do Rio de Janeiro, COPPE.

1. Algoritmos para o Problema de Localização I. COPPE/UFRJ  
II. Título (série).

A Ana Luíza e  
Giovana

AGRADECIMENTOS

A todos que de algum modo colaboraram na execução deste trabalho. Em especial ao Professor Cláudio Thomás Bornstein pelo incentivo e orientação, ao Professor Nelson Maculan Filho pela co-orientação.



Resumo da Tese Apresentada à COPPE/UFPRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.).

ALGORITMO EXATO E HEURÍSTICAS PARA O  
PROBLEMA DE LOCALIZAÇÃO

Geraldo Robson Mateus

Abril, 1986

Orientador: Cláudio Thomás Bornstein

Programa: Engenharia de Sistemas e Computação

Trataremos o Problema de Localização de Facilidades Capacitado. Inicialmente apresentamos um conjunto de Testes de Redução que permitem acelerar a convergência e reduzir o tempo computacional para resolução do problema proposto. Esses testes são básicos para o desenvolvimento das duas heurísticas gulosas apresentadas, uma para o modelo com funções custos fixos mais variáveis lineares e outra para funções custos fixos mais variáveis côncavos. Finalmente introduzimos um algoritmo exato tendo a Relaxação Lagrangeana como ferramenta básica; analisando duas relaxações possíveis.

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

EXACT ALGORITHM AND HEURISTICS FOR THE  
LOCATION PROBLEM

Geraldo Robson Mateus

April, 1986

Chairman: Cláudio Thomás Bornstein

Department: Systems and Computers Engineering

We consider the classical capacitated facility location problem. Some criteria to close or open a facility are proposed. These criteria are powerful devices for fast convergence of the algorithms discussed. We present two greedy heuristics, one for the problem with fixed costs and linear variables costs and other with concaves variables costs. Finally, we introduced an exact algorithm with two lower bounds obtained by lagrangean relaxation.

ÍNDICE

Pág.

<u>CAPÍTULO I - INTRODUÇÃO</u> .....	1
<u>CAPÍTULO II - TESTES DE REDUÇÃO</u> .....	16
1. TESTES PARA ABRIR UMA FACILIDADE .....	20
2. TESTES PARA FECHAR UMA FACILIDADE .....	29
3. TESTES COMPLEMENTARES .....	39
4. CONCLUSÕES .....	40
<u>CAPÍTULO III - UM ALGORITMO GULOSO PARA FUNÇÃO CUSTO FIXO/ LINEAR</u> .....	41
1. ALGORITMO GULOSO .....	42
2. IMPLEMENTAÇÃO .....	45
3. ANÁLISE DE RESULTADOS .....	48
<u>CAPÍTULO IV - UM ALGORITMO GULOSO PARA FUNÇÃO CUSTO FIXO/ CÔNCAVO</u> .....	60
1. ALGORITMO GULOSO .....	61
1.1. Testes para Abrir uma Facilidade .....	64
1.2. Segunda Etapa .....	69
1.3. Processo Iterativo Final .....	71
1.4. Algoritmo .....	72
2. ANÁLISE DE RESULTADOS .....	75

	<u>Pág.</u>
<u>CAPÍTULO V - RELAXAÇÃO LAGRANGEANA</u> .....	84
1. LIMITES INFERIORES .....	90
1.1. Relaxar a Demanda .....	91
1.2. Relaxar a Oferta .....	94
1.3. Comparação dos Limites Inferiores .....	97
2. LIMITE SUPERIOR .....	98
3. DETERMINAÇÃO DE UMA SOLUÇÃO VIÁVEL .....	99
4. TESTES DE REDUÇÃO - PENALIDADES .....	102
5. SEPARAÇÃO .....	106
6. MÉTODO DE SUBGRADIENTE .....	107
7. IMPLEMENTAÇÃO .....	111
8. ANÁLISE DE RESULTADOS .....	113
<u>CAPÍTULO VI - CONCLUSÕES</u> .....	119
<u>APÊNDICE A - ALGORITMO PROPOSTO POR MONTEROSSO</u> .....	121
<u>APÊNDICE B - ALGORITMO PROPOSTO POR KELLY-KHUMAWALA</u> .....	132
<u>APÊNDICE C - TESTES DE REDUÇÃO COMPLEMENTARES</u> .....	135
<u>APÊNDICE D - PROBLEMAS TESTES</u> .....	140
<u>BIBLIOGRAFIA</u> .....	144

CAPÍTULO IINTRODUÇÃO

Trataremos o Problema de Localização de Facilidades Capacitado, apresentando duas heurísticas e um algoritmo exato. Resultados significativos foram obtidos pela aplicação desses algoritmos a um grande número de problemas testes.

Inicialmente destacaremos o modelo para o Problema de Localização Não Capacitado.

Seja  $J = \{1, \dots, n\}$  um conjunto de centros de demanda que tem que ser abastecidos por um conjunto de facilidades ou fornecedores, cuja localização e dimensionamento queremos determinar. E seja  $I = \{1, \dots, m\}$  o conjunto das possíveis localizações destes fornecedores sendo  $K$  o número máximo de locais que queremos localizar.

Seja  $c_{ij}$  o custo proveniente do fato da facilidade  $i$  abastecer o centro de demanda  $j$ . Além disso, seja  $F_i$  o custo fixo da instalação da facilidade  $i$ .

O problema consiste portanto em determinar um subconjunto  $S \subseteq I$  de facilidades, tal que  $|S| \leq K$ , de maneira a minimizar os custos.

O modelo matemático para este problema é dado por:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} F_i y_i \quad (I-1)$$

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J \quad (I-2)$$

$$(M1) \quad 1 \leq \sum_{i \in I} y_i \leq K \quad (I-3)$$

$$x_{ij} \leq y_i, \quad i \in I, \quad j \in J \quad (I-4)$$

$$x_{ij}, y_i \in \{0, 1\}, \quad i \in I, \quad j \in J \quad (I-5)$$

onde  $x_{ij} = \begin{cases} 1, & \text{se a facilidade } i \text{ atende o centro demanda } j \\ 0, & \text{caso contrário} \end{cases}$

$y_i = \begin{cases} 1, & \text{se a facilidade } i \text{ for ativada ou aberta} \\ 0, & \text{caso contrário} \end{cases}$

O modelo acima permite formulações diversas e engloba uma série de problemas conhecidos como o de determinação da k-mediana, "lock-box", e em particular o problema de localização de facilidades não capacitado. Todos eles bastante estudados e com uma vasta literatura a respeito. Por exemplo: CORNUEJOLS e outros (10), FRANCIS e GOLDSTEIN (25), GEOFFRION (29), GALVÃO (26), MATEUS e BORNSTEIN (45), OLIVEIRA e MATEUS (49).

Podemos acrescentar capacidades  $a_i$  a cada facilidade  $i \in I$  e  $b_j$  a cada centro de demanda  $j \in J$ . Fazendo  $x_{ij}$  corresponder ao fluxo de  $i$  para  $j$ , o novo modelo pode ser escrito da seguinte maneira:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} F_i y_i \quad (I-6)$$

$$\sum_{j \in J} x_{ij} \leq a_i \cdot y_i, \quad i \in I \quad (I-7)$$

$$(M2) \quad \sum_{i \in I} x_{ij} = b_j, \quad j \in J \quad (I-8)$$

$$1 \leq \sum_{i \in I} y_i \leq K \quad (I-9)$$

$$x_{ij} \geq 0, \quad i \in I, \quad j \in J \quad (I-10)$$

$$y_i \in \{0, 1\}, \quad i \in I \quad (I-11)$$

Em (I-7) garantimos a restrição relativa à capacidade de cada facilidade  $i \in I$ , caso esta facilidade seja selecionada. Também as restrições (I-4), no modelo anterior, estão embutidas em (I-7). Em (I-8) explicitamos as restrições para o atendimento dos centros de demanda, anteriormente expressas em (I-2).

O modelo (M2) corresponde ao problema de localização de facilidades capacitado que será objeto de estudo neste trabalho. Em geral, a restrição (I-9) é dispensada.

De (I-6), a função a ser minimizada pode ser dividida em custos variáveis e fixos. A parcela variável pode ser acrescida dos custos operacionais das facilidades que são funções da dimensão ou dos fluxos em cada facilidade  $i \in I$ ,  $f_i(\sum_{j \in J} x_{ij})$ .

Se as funções  $f_i(\cdot)$  são lineares então podemos escrever  $f_i(\sum_{j \in J} x_{ij}) = t_i \sum_{j \in J} x_{ij}$  onde  $t_i$  é uma constante para todo  $i \in I$ . Com isto, podemos fazer  $C_{ij} = c_{ij} + t_i$ ,  $i \in I$ ,  $j \in J$ , e reescrever o modelo (M2) com os  $c_{ij}$  substituídos pelos  $C_{ij}$ ,  $i \in I$ ,

$j \in J$ , em (I-6). Este modelo será denominado de problema de localização de facilidades com função custo fixo/linear.

Se  $f_i(\cdot)$ ,  $i \in I$ , é uma função não linear, o modelo (M2) pode ser reescrito como tal a menos da função objetivo (I-6) e será assim composto:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i \left( \sum_{j \in J} x_{ij} \right) + \sum_{i \in I} F_i y_i$$

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

$$(M3) \quad \sum_{i \in I} x_{ij} = b_j, \quad j \in J$$

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

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

Explorando o conceito de economia de escala, as funções  $f_i(\cdot)$  são, em geral, côncavas. Neste caso, denominaremos de problema de localização com função custo fixo/côncavo. Cabe observar que se  $f_i(\cdot)$ ,  $\forall i \in I$ , são lineares, temos uma função custo côncavo. Mas este caso é sempre tratado a parte como função custo fixo/linear.

Além das características anteriores, os modelos de localização podem ainda ser expressos em dois níveis, ou seja, as facilidades se localizam como centros intermediários entre os



centros de produção e de demanda.

Em resumo, os modelos de localização consistem em minimizar uma função côncava sujeito a um conjunto convexo de restrições lineares. A principal dificuldade com esse problema é que um mínimo local não implica em mínimo global. Por outro lado, é conhecido que o mínimo de uma função côncava sobre um conjunto compacto e convexo, sempre ocorre em um vértice desse conjunto. Isto é consequência da própria definição de função côncava. Logo, um ponto extremo do poliedro viável é uma solução para o problema. Uma completa enumeração destes pontos extremos é impraticável para maioria dos problemas e é um aspecto importante em Combinatória. Portanto, várias técnicas tem sido desenvolvidas visando determinar a melhor maneira de enumerar e selecionar os pontos extremos de forma que o mínimo global possa ser obtido. Estas técnicas são geralmente de natureza combinatória e se fundamentam na estrutura especial de cada problema analisado.

Os algoritmos se concentram em três áreas básicas: decomposição, enumeração e heurísticas.

Os algoritmos de decomposição seguem, em geral, o método de decomposição de Benders. Como as formulações do problema de localização envolvem variáveis inteiras e reais, este método separa a parte inteira da real, resolvendo problemas menores conectados a um problema mestre.

Os algoritmos de enumeração procuram subdividir o pro

blema original em subproblemas menores de fácil resolução, enumerando, explicitamente ou implicitamente, todos os pontos extremos. Os algoritmos de enumeração são basicamente do tipo separação e avaliação (Branch-and-Bound) que são técnicas de busca em árvore onde cada nó da árvore representa um subconjunto de soluções viáveis, GEOFFRION e MARSTEN (31). Um algoritmo de separação e avaliação fica caracterizado ao definirmos como separar o conjunto de soluções viáveis e como avaliar a função, determinando os limites inferiores e superiores. Com isto, são particularizados com o uso de técnicas específicas. Assim é que podemos ter algoritmos em que a ramificação se dá pela fixação de variáveis inteiras ou pelo desdobramento de uma variável, ou arco, utilizando o conceito de envoltória convexa. Os limites inferiores são geralmente obtidos com a resolução de problemas lineares por um método primal ou utilizando um método dual como a relaxação lagrangeana, etc.

O objetivo dos algoritmos heurísticos é determinar boas soluções com um consumo mínimo de tempo e capacidade de memória do computador. Com essas prioridades, as heurísticas caracterizam-se pela flexibilidade e simplicidade computacional.

Procuramos a seguir, referenciar os principais trabalhos e algoritmos de decomposição, enumeração e heurísticas voltados para o problema de localização capacitado e suas particularidades.

Dentre os algoritmos de decomposição destaca-se, o proposto por BALINSKY (3) e GEOFFRION e GRAVES (30). Ambos uti

lizam a decomposição de Benders, sendo que o último considera vários produtos, implicando na resolução de vários problemas de transporte independentes, um para cada produto.

Recentemente, VAN ROY (61), motivado pelos métodos duais de BILDE e KRARUP (6), GALVÃO (26), ERLKOTTER (16) e GUIGNARD e SPIELBERG (34), para o problema de localização não capacitado, desenvolveu um algoritmo denominado "cross decomposition". Isto porque os algoritmos usuais se baseiam na decomposição primal ou dual, o que nos leva, respectivamente, aos algoritmos de decomposição de Benders e aos algoritmos de enumeração com limites obtidos via relaxação lagrangeana. O algoritmo de Van Roy explora simultaneamente ambas as estruturas, primal e dual. Porém, na prática, os resultados obtidos são comparáveis aos obtidos pelos métodos de enumeração com a relaxação lagrangeana.

Os algoritmos de enumeração são na sua maioria do tipo separação e avaliação (Branch-and-Bound). Para o problema de localização capacitado com custos operacionais das facilidades ( $f_i(.)$ ) lineares, destaca-se os algoritmos com fixação de facilidades, propostos por SÁ (54), DAVIS e RAY (11), ELLWEIN e GRAY (14) e AKINC e KHUMAWALA (1).

O algoritmo de SÁ (54) inicialmente considera todas as facilidades instaladas. Os subproblemas resultantes para um subconjunto de facilidades abertas são resolvidos pelo algoritmo "out-of-kilter". Da experiência computacional, sugere que o algoritmo deve ser aplicado a problemas onde o número de locais

possíveis para instalação de uma facilidade esteja em torno de 25. Para problemas maiores propõe um algoritmo aproximado baseado nas heurísticas gulosas ADD e DROP detalhadas abaixo.

ELLWEIN e GRAY (14) adicionou ao problema anterior um conjunto de restrições limitando o número de facilidades. Resolve os subproblemas pelo método primal de transporte. Acrescentou podas ou testes de viabilidade-otimalidade para redução da árvore. Estes testes advêm do acréscimo de restrições baseadas nas variáveis duais e de um limite adaptativo para o custo fixo, definido em cada nó. Consegue-se assim a exclusão de soluções não ótimas.

AKINC e KHUMAWALA (1) estenderam para o problema de localização capacitado, o algoritmo proposto por EFROYMSON e RAY (12) para resolução de um problema de localização não capacitado. Os limites são obtidos pela resolução de problemas lineares de transporte. Um conjunto eficiente de podas, que permite abrir ou fechar facilidades a priori, reduz substancialmente a árvore. Posteriormente apresentaremos maiores detalhes.

Problemas com um máximo de 25 facilidades e 50 centros de demanda foram resolvidos, na maioria dos casos, com tempos computacionais inferiores a um minuto em um IBM 370/165.

Para o problema de localização capacitado com função custo côncava destacam-se os algoritmos de separação e avaliação propostos por FALK e SOLAND (18), SOLAND (57), FRANÇA e GOMES (24) e BORNSTEIN e RUST (7).

MILLER (46) aplicou o método simplex para o caso em que a função objetivo pode ser descrita como a soma de um número de funções não-lineares de uma simples variável ou função separável. Este algoritmo gera uma solução para o problema original linearizado. FALK e SOLAND (18) baseando, em parte, no trabalho de Miller, desenvolveram um algoritmo geral para o problema de minimizar uma função côncava. Usaram o conceito de "envelope convexo" para problemas não convexos com funções separáveis e restrições não convexas separáveis. SOLAND (57) particularizou este algoritmo para o caso de uma função objetivo côncava separável, sujeito a um conjunto convexo de restrições lineares, que engloba, em particular, o problema de localização capacitado. Consiste em separar o problema proposto em subproblemas pela aproximação linear a função côncava em cada arco. São subproblemas lineares de transporte, o que facilita a obtenção dos limites inferiores.

FRANÇA e GOMES (24) estenderam o algoritmo anterior para o problema em dois níveis ou uma rede com facilidades intermediárias entre produtores e consumidores. Os limites inferiores são obtidos resolvendo os problemas lineares de fluxo mínimo pelo "out-of-kilter". De forma semelhante, BORNSTEIN e RUST (7) estenderam este algoritmo para o caso de funções escadas.

Embora bastante estudados, a experiência computacional com os algoritmos para funções côncavas tem sido limitada. A prática tem se dado a resolução de problemas com um máximo de 25 variáveis e 100 restrições.

Recentemente tem sido utilizado junto aos algoritmos de enumeração, na determinação de limites inferiores para as funções, a técnica de relaxação lagrangeana. Todos os algoritmos propostos para o problema de localização capacitado, com função custo variável linear são de separação e avaliação com fixação de variáveis. O primeiro algoritmo foi desenvolvido por GEOFFRION e McBRIDE (32). Este trabalho se destaca principalmente pelo aspecto teórico, com experiência computacional bastante limitada. Outro algoritmo utilizando a relaxação lagrangeana foi proposto por NAUSS (48). Apresenta resultados comparáveis ou mesmo superiores aos obtidos por AKINC e KHUMAWALA (1).

Para o mesmo problema anterior, com restrições adicionais, CHRISTOFIDES e BEASLEY (8), obtiveram resultados melhores. Isto porque adicionaram outros testes que permitem abrir ou fechar facilidades, reduzindo o problema. Testaram basicamente os mesmos problemas experimentados por Naus, que são problemas com 16 a 25 facilidades e 50 centros de demanda. Christofides e Beasley chegaram a testar problemas da ordem de (50 x 50), (35 x 100) e (50 x 150). Para os experimentos em comum, Naus consumiu, em média, 13 segundos de tempo de CPU (exclui tempo de entrada e saída) em um IBM 370/168, com 12 problemas de fluxo sendo resolvidos por solução ótima obtida. Enquanto Christofides e Beasley consumiram, em média, 5,1 segundos em um CDC 7600 (incluindo tempo de entrada e saída) com 8,1 problemas de fluxo resolvidos por solução ótima obtida.

Seguindo a mesma linha de ZANGWILL (62), que procurou

caracterizar pontos extremos para problemas específicos, FLORIAN e ROBILLARD (23) propuseram um algoritmo de enumeração implícita. Transformaram o problema geral de fluxos em redes em uma rede bipartida, não capacitada, de forma especial. Um ponto extremo, que seja mínimo em um problema também será no outro. A característica básica destes algoritmos é que os pontos extremos são discretos e explorados como "fluxos extremos", os quais são gerados pelos algoritmos. As aplicações se dão a problemas bem particulares e pequenos.

Uma outra linha de pesquisa, onde se inclui o problema de localização capacitado, são os algoritmos de otimização global por reconstrução de poliedros. Exploram o aspecto não linear da função objetivo, exigindo às vezes continuidade e diferenciabilidade. A idéia básica se encontra no trabalho de TUY (60) que propôs a construção de cortes (hiperplanos) para exclusão de partes da região viável. Também propôs a partição da região viável em cones. Posteriormente, aparece HORST (38) que desenvolveu um algoritmo de separação e avaliação e THOAI e TUY (59) que descreve um algoritmo de minimização global para problemas com restrições lineares baseado no trabalho de TUY (60). Estes algoritmos foram testados para problemas com um máximo de 30 variáveis. ROSEN (53) explorou a forma da função objetivo e com a eliminação de parte do interior do conjunto de soluções viáveis, conseguiu resolver problemas maiores, porém não superior a 100 variáveis. Todavia, este algoritmo exige funções diferenciáveis.

As primeiras heurísticas para o problema de localiza

ção de facilidades foram apresentadas por BAUMOL e WOLFE (5) e BALINSKI e MILLS (4).

BAUMOL e WOLFE (5) trabalharam com o problema em dois níveis onde as facilidades são intermediárias entre os centros produtores e de demanda. Os custos de transporte são lineares e os custos operacionais das facilidades côncavos, porém, sem custo fixo. Inicialmente somente os custos de transporte são considerados transformando o problema original em um problema de transporte onde os custos são dados pelos caminhos menos custosos entre os produtores e consumidores. Posteriormente, a cada iteração, resolve-se um problema de transporte considerando os custos de transporte e custos marginais de operação das facilidades calculados sobre os fluxos obtidos na iteração anterior. Prossegue iterativamente até que os fluxos se estabilizem.

BALINSKI e MILLS (4) propuseram um algoritmo para o problema em um nível com funções custo  $f_i(.)$  lineares por partes. Resolve um problema de transporte com custos de transporte mais os custos unitários de operação calculados para as facilidades em sua capacidade máxima. Obtém-se assim um limite inferior. O limite superior é obtido calculando o valor dessa solução considerando a função real. Não é um algoritmo iterativo.

MONTEROSSO (47) propôs uma heurística com custos de transporte lineares e custos operacionais côncavos com ou sem custo fixo. Aproxima a função côncava, iterativamente, com custos lineares unitários, calculados sobre os fluxos obtidos na



iteração anterior. O processo prossegue até que os fluxos não se alterem. Utiliza o "out-of-kilter" para solução dos problemas de fluxo mínimo. Visando melhorar a solução anterior, aplica-se, em uma segunda etapa, uma subrotina SHIFT, que é uma combinação das heurísticas ADD e DROP.

Apesar dos bons resultados obtidos, a não otimalidade do algoritmo pode ser mostrada por um pequeno contra-exemplo. Mas o algoritmo converge para um mínimo local, uma vez que demonstramos que a função decresce a cada iteração. (Veja Apêndice A).

KELLY e KHUMAWALA (40) apresentam um algoritmo para o problema de localização capacitado com funções côncavas e com ou sem custos fixos. É um algoritmo em duas partes. Na primeira utiliza os custos de transporte mais os custos operacionais linearizados através dos custos marginais calculados iterativamente sobre os fluxos anteriores. Na segunda parte introduz um centro de demanda artificial  $D_2$ , o qual "compra" a capacidade das facilidades que poderá ser necessária para atender aos centros de demanda reais. O custo unitário de alocar qualquer fluxo para  $D_2$  é o negativo da média do custo operacional total dividido pelo fluxo total para todos os centros de demanda reais e para o centro de demanda artificial  $D_1$ , responsável pelo equilíbrio entre ofertas e demandas totais. Portanto  $D_2$  compete com  $D_1$  e os demais centros de demanda. Os custos para  $D_2$  são um misto entre custo marginal e unitário. Para os demais arcos prossegue com os custos de transportes mais os custos marginais. Iterativamente, dentro de condições estabelecidas, procu

ra-se abrir ou fechar facilidades. Utiliza o método de transporte como subrotina básica, porém, não apresenta resultados computacionais. Mostramos a não otimalidade deste algoritmo através de um contra-exemplo. (Veja Apêndice B). De forma semelhante ao algoritmo de MONTEROSSO (47), é fácil demonstrar que este algoritmo também converge para um mínimo local.

JACOBSEN (39) generalizou para o problema de localização capacitado, heurísticas inicialmente desenvolvidas para o não capacitado. Generalizou o guloso desenvolvido por KUEHN e HAMBURGER (41), em que a cada iteração abrimos a facilidade que forneça a maior economia. Esta heurística foi denominada de ADD. A heurística inversa à anterior, desenvolvida por FELDMAN e outros (19), em que a cada iteração fechamos a facilidade que forneça a maior economia, foi denominada de DROP. Também generalizou duas outras heurísticas que consistem na utilização alternada dos gulosos anteriores. Baseou-se em MANNE (44), que combinou ADD e DROP para o problema de localização não-capacitado. Outras heurísticas foram elaboradas visando melhorar uma solução já conhecida. Estas outras heurísticas de localização alternada, denominadas de ALA, foram inicialmente desenvolvidas por COOPER (9) para o problema das p-medianas. Também KUEHN e HAMBURGER (41) utilizou esta idéia na segunda etapa do seu algoritmo, a qual denominou de SHIFT, visando obter uma solução melhor do que a obtida na primeira etapa. Conforme seja utilizada inicialmente a heurística ADD ou DROP teremos as heurísticas SHIFT ou ALA. Outra heurística generalizada, denominada VSM, foi o método de substituição de vértices, proposto por TEITZ e BART (58), originalmente desenvolvido para o problema das p-me

dianas. Levando em consideração principalmente a qualidade das soluções obtidas, em termos dos valores das funções, destacamos as heurísticas ADD e DROP seguidas da VSM.

No capítulo II que se segue, introduziremos um conjunto de testes de redução para abrir e fechar facilidades. O objetivo é procurar fixar o maior número possível de variáveis a um tempo reduzido. Para cada uma das alternativas, abrir ou fechar, apresentamos dois testes retirados da literatura e propomos um terceiro. Este capítulo é fundamental uma vez que será explorado em todos os capítulos subsequentes. No capítulo III propomos uma heurística gulosa para o problema de localização capacitado (PLC) com função custo fixo/linear. Esta heurística é estendida para o caso de funções custo fixo/côncavo, no capítulo IV. Finalmente no capítulo V apresentamos um algoritmo exato tipo separação e avaliação com a proposta de dois limites inferiores resultantes de relaxações lagrangeanas distintas. Apesar de trabalharmos apenas no nó inicial da árvore, esta implementação nos permitiu analisar o comportamento da relaxação lagrangeana e do método de subgradiente, bem como comparar os resultados obtidos para o caso exato e aqueles para a heurística gulosa.

Os três algoritmos foram testados e comparados utilizando um conjunto de problemas testes apresentados no Apêndice D em um computador Burroughs 6700.

CAPÍTULO IITESTES DE REDUÇÃO

Pelos problemas de localização apresentados anteriormente, cada facilidade está associada a uma variável inteira 0/1 e deverá ser fixada aberta (1) ou fechada (0). Independente do enfoque seguido para resolução desse problema, procura-se fixar, a priori, e sempre que possível, um conjunto de facilidades abertas ou fechadas. Esse passo inicial compõe um conjunto de cortes ou podas ao conjunto de soluções viáveis e denominaremos por TESTES DE REDUÇÃO. O objetivo desses testes é acelerar a convergência e reduzir ao máximo o tempo computacional para resolução do problema proposto. Serão apresentados em três etapas: testes para abrir uma facilidade, testes para fechar uma facilidade e testes complementares que são extensões aos testes anteriores.

Para aplicação desses testes, as funções em estudo devem satisfazer a propriedade de submodularidade, apresentada em seguida e utilizada posteriormente.

Seja o problema proposto da forma:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i \left( \sum_j x_{ij} \right) + \sum_{i \in I} F_i y_i \quad (\text{II.1})$$

$$\sum_{j \in J} x_{ij} \leq a_i y_i, \quad i \in I \quad (\text{II.2})$$

(P1)

$$\sum_{i \in I} x_{ij} = b_j, \quad j \in J \quad (\text{II.3})$$

$$x_{ij} \geq 0, y_i \in \{0, 1\}, \forall i \text{ e } \forall j \quad (\text{II.4})$$

Seja  $X$  o conjunto de soluções viáveis  $x$  dado por (II.2), (II.3) e (II.4) para um vetor  $y$  de coordenadas 0/1.

Dado um conjunto  $I = \{1, 2, \dots, m\}$  de locais candidatos à localização de facilidades, determinar um conjunto  $S \subseteq I$  tal que  $Z_r(S)$  seja mínima, onde  $Z_r(S)$  é dada por:

$$Z_r(S) = \begin{cases} \min_{\substack{S \subseteq I \\ x \in X}} \sum_{i \in S} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in S} f_i(.) + \sum_{i \in S} F_i & \text{se } S \neq \emptyset \mid \exists x \in X \text{ e } y_i = 1, \forall i \in S \\ \infty & \text{se } S \neq \emptyset \mid \nexists x \in X \\ 0 & \text{se } S = \emptyset \end{cases} \quad (\text{II.5})$$

Suponhamos  $S \neq \emptyset \mid \exists x \in X, y_i = 1, \forall i \in S$ .

Se  $f_i(.)$  é uma função linear ou  $f_i(\sum_{j \in J} x_{ij}) = t_i \sum_{j \in J} x_{ij}$ , então o correspondente valor para  $Z_r(S)$  é dado por  $Z(S)$ .

$$Z(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in S} t_i \sum_{j \in J} x_{ij} + \sum_{i \in S} F_i$$

ou

$$Z(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} (c_{ij} + t_i) x_{ij} + \sum_{i \in S} F_i$$

Fazendo  $C_{ij} = c_{ij} + t_i$  para todo  $(i, j)$  resulta:

$$Z(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in S} F_i \quad (\text{II.6})$$

que pode ser decomposta em:

$$W(S) = \begin{cases} \min_{x \in X} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} & \text{se } S \neq \emptyset \mid \exists x \in X \text{ e } y_i = 1, \forall i \in S \\ \infty & \text{se } S \neq \emptyset \mid \nexists x \in X \\ 0 & \text{se } S = \emptyset \end{cases}$$

$$F(S) = \begin{cases} \sum_{i \in S} F_i & \text{se } S \neq \emptyset \\ 0 & \text{se } S = \emptyset \end{cases}$$

A função  $Z(S)$  é submodular, pois é uma combinação linear de funções submodulares  $W(S)$  e  $F(S)$ . (FISHER e outros (21). Portanto,  $Z(S)$ ,  $W(S)$  e  $F(S)$ , satisfazem as seguintes propriedades:

$$(a) Z(R \cup \{k\}) - Z(R) \leq Z(S \cup \{k\}) - Z(S), \forall R \subseteq S, k \notin S \quad (II.7)$$

$$(b) Z(S - \{k\}) - Z(S) \leq Z(R - \{k\}) - Z(R), \forall R \subseteq S, k \in R \quad (II.8)$$

Cabem aqui três observações:

(i) É fácil verificar que as propriedades (II.7) e (II.8) se equivalem.

(ii) Conforme definição acima,  $W(S)$  é o valor da função correspondente a solução ótima do problema de transporte com um conjunto de centros de oferta dados por  $S \subseteq I$

(iii) Para simplificar a notação, sempre que possível, escreve

remos  $Z(S \cup \{k\}) = Z(S \cup k)$ .

Se  $f_i(\cdot)$  é uma função côncava, as propriedades (II.7) e (II.8) não são atendidas pela função  $Z_r(S)$ . A propriedade de submodularidade está associada à idéia de convexidade. Se  $Z_r(S)$  é uma função côncava então não é submodular (BABAYEV (2)).

Seja o conjunto  $I$  de facilidades dividido entre os seguintes subconjuntos:

$K_0$  = conjunto das facilidades fixadas fechadas,  $y_i = 0$ .  
Inicialmente é um conjunto vazio.

$K_1$  = conjunto das facilidades fixadas abertas,  $y_i = 1$ .  
Inicialmente é também um conjunto vazio.

$K_2$  = conjunto das facilidades não fixadas ou livres.  
Inicialmente contém todas as facilidades candidatas a localização.

Para os subconjuntos acima podemos estender as definições efetuadas anteriormente. Seja

$$W(K_1 \cup K_2) = \min_{\substack{S \subseteq (K_1 \cup K_2) \\ x \in X}} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} \quad (\text{II.9})$$

Portanto, como  $X$  é o conjunto de soluções viáveis de (P1),  $W(K_1 \cup K_2)$  é o valor da função objetivo do problema de transpor

te onde os centros de oferta pertencem aos conjuntos  $K_1$  e  $K_2$ . Este problema de transporte será referenciado como  $T(K_1 \cup K_2)$ . Conseqüentemente,

$$Z(K_1 \cup K_2) = \min_{\substack{S \subseteq (K_1 \cup K_2) \\ x \in X}} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in S} F_i \quad (\text{II.10})$$

O mínimo da função ocorre para um subconjunto  $S \subseteq (K_1 \cup K_2)$  e deve ser tal que  $S \supseteq K_1$ .

Inicialmente serão apresentados os testes de redução para o PLC com funções custo variável de armazenamento linear e custo fixo. Posteriormente, estenderemos a aplicação desses testes para o PLC com funções custo variável côncavo.

## 1. TESTES PARA ABRIR UMA FACILIDADE

Para todo  $i \in K_2$  faça

$$\Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2) \quad (\text{II.11})$$

$\Delta_i$  é a variação nos custos variáveis do problema de transporte  $T(K_1 \cup K_2)$  ao desativar a facilidade  $i$ . Pela definição de  $K_0$ ,  $K_1$ ,  $K_2$ , podemos afirmar que  $(K_1 \cup K_2)$  é um conjunto não crescente em número de elementos. E como  $W(S)$  é submodular, então pela propriedade (II.8),  $\Delta_i$  é não decrescente. Com isto,  $\Delta_i$  é a economia mínima obtida, com relação aos custos variáveis, ao abrir a facilidade  $i$ . Ainda mais, se esta economia mínima for



superior ao custo fixo dessa facilidade então podemos fixá-la aberta.

TEOREMA 1.1 Se  $\Delta_i \geq F_i$  então a facilidade  $i$  é fixada aberta, isto é,  
 $y_i = 1$ .

Demonstração: seja  $\Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2)$  e

$$\Delta_i^* = Z(K_1 \cup K_2 - i) - Z(K_1 \cup K_2)$$

onde os valores de  $W(\cdot)$  e  $Z(\cdot)$  são dados por (II.9) e (II.10) e  $\Delta_i^*$  é a variação real da função objetivo ao desativar a facilidade  $i$ . Mostraremos que se  $\Delta_i \geq F_i$  então  $\Delta_i^* \geq 0$  e não decrescente. Portanto podemos fixar  $i$  aberta, pois, caso contrário poderemos ter um acréscimo no custo total do sistema.

É fácil mostrar que a função  $W(\cdot)$  é não crescente, logo  $\Delta_i \geq 0$ .

Suponhamos  $\Delta_i > 0$ . Nesse caso podemos afirmar que a facilidade  $i$  está em atividade ao resolver o problema de transporte  $T(K_1 \cup K_2)$ , ou  $i \in S \subseteq (K_1 \cup K_2)$  tal que  $W(S)$  é mínimo. Ao retirar  $i$  de  $(K_1 \cup K_2)$  e resolver  $T(K_1 \cup K_2 - i)$  duas coisas podem ocorrer:

(i) Toda demanda atendida por  $i$  em  $T(K_1 \cup K_2)$  será absorvida pelos elementos ativos em  $T(K_1 \cup K_2)$ . Logo, a variação na função real é dada por:

$$Z(K_1 \cup K_2) + \Delta_i - F_i = Z(K_1 \cup K_2 - i)$$

Se  $\Delta_i \geq F_i$  então  $Z(K_1 \cup K_2) \leq Z(K_1 \cup K_2 - i)$  ou  $\Delta_i^* \geq 0$ .

(ii) A demanda atendida por  $i$  em  $T(K_1 \cup K_2)$  será absorvida pelos elementos não ativos em  $T(K_1 \cup K_2)$ . Seja  $\{s_1, s_2, \dots, s_s\}$  este conjunto de facilidades não ativas em  $T(K_1 \cup K_2)$  e ativadas em  $T(K_1 \cup K_2 - i)$ . Logo, a variação na função real é dada por:

$$\begin{aligned} Z(K_1 \cup K_2) + \Delta_i + F_{s_1} + F_{s_2} + \dots + F_{s_s} - F_i &= \\ &= Z(K_1 \cup K_2 - i) \end{aligned}$$

Se  $\Delta_i \geq F_i$  e uma vez que  $F_{s_1}, \dots, F_{s_s} \geq 0$  então  $Z(K_1 \cup K_2) \leq Z(K_1 \cup K_2 - i)$  ou  $\Delta_i^* \geq 0$ . Para completar esta segunda parte é necessário mostrar que com a retirada de  $i$  e com a ativação de novos elementos não implica na desativação de qualquer facilidade em  $T(K_1 \cup K_2)$ .

Seja  $r \in (K_1 \cup K_2)$  ativo em  $T(K_1 \cup K_2)$  então  $\Delta_r = W(K_1 \cup K_2 - r) - W(K_1 \cup K_2) \geq 0$ . Pela submodularidade de  $W(\cdot)$  temos que  $\Delta_r' = W(K_1 \cup K_2 - i - r) - W(K_1 \cup K_2 - i) \geq \Delta_r \geq 0 \Rightarrow W(K_1 \cup K_2 - i - r) \geq W(K_1 \cup K_2 - i)$ . Então  $r$  não será desativado.

Se  $\Delta_i = 0$  então  $\Delta_i \geq F_i$  implica  $F_i = \Delta_i = 0$ . Logo, de forma semelhante teremos  $Z(K_1 \cup K_2) \leq Z(K_1 \cup K_2 - i)$  ou  $\Delta_i^* \geq 0$ . Portanto se  $\Delta_i \geq F_i$  então  $\Delta_i^* \geq 0$ . Pela submodularidade de  $Z(\cdot)$  e como  $(K_1 \cup K_2)$  é um conjunto não crescente então podemos afirmar que  $\Delta_i^*$  é não decrescente. A economia mínima em abrir a facilidade  $i$  é não negativa,  $\Delta_i^* \geq 0$ , podemos abrir  $i$ . ●

Executar este teste para todo  $i \in K_2$  exige a resolução de  $|K_2| + 1$  problemas de transporte podendo consumir um tempo computacional expressivo. Daí ser necessário restringir a sua utilização repetidamente. ELLWEIN (13) utiliza este teste apenas no primeiro nó do seu algoritmo de enumeração. Por outro lado, uma vez que  $\Delta_i$  é não decrescente, as chances de fixar a facilidade  $i$  aberta são crescentes.

Um processo alternativo para o teste anterior é determinar um limite inferior para  $\Delta_i$ , facilmente calculado. Acharia em uma perda no poder de poda desse teste, porém ganharia em simplicidade computacional. AKINC e KHUMAWALA (1) propuseram um limite inferior para  $\Delta_i$  onde nenhum problema de transporte é resolvido. Denominaremos-lo de  $\bar{\Delta}_i$ , de forma que:

$$\text{se } \Delta_i \geq \bar{\Delta}_i \geq F_i, i \in K_2 \text{ então } y_i = 1.$$

Posteriormente, proporemos um outro limite inferior,  $\Delta_i^D$ , que, apesar de superior ao anterior, exige a resolução de um problema de transporte.

O limite proposto por AKINC e KHUMAWALA (1),  $\bar{\Delta}_i$ , é calculado de forma semelhante a  $\Delta_i$  relaxando as restrições de capacidade das facilidades em  $(K_1 \cup K_2)$  a menos de  $i$ . Corresponde em adicionar para cada facilidade em  $(K_1 \cup K_2 - i)$  uma outra artificial com capacidade infinita. Chamando de  $R$  este conjunto de facilidades artificiais resulta:

$$\bar{\Delta}_i = W(K_1 \cup K_2 - i \cup R) - W(K_1 \cup K_2 \cup R) \quad (\text{II.12})$$

Pela propriedade (II.8) podemos afirmar que  $\bar{\Delta}_i$  é um limite inferior:

$$\bar{\Delta}_i \leq \Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2)$$

Calcular  $\bar{\Delta}_i$  em (II.12) consiste em resolver os problemas de transporte  $T(K_1 \cup K_2 - i \cup R)$  e  $T(K_1 \cup K_2 \cup R)$ . O primeiro é semelhante a um problema não capacitado, e solução dada por:

$$x_{ij} = \begin{cases} b_j, & \text{se } C_{ij} = \min_{\ell \in (K_1 \cup K_2 - i)} \{C_{\ell j}\} \\ 0, & \text{caso contrário} \end{cases}$$

Para o segundo,  $T(K_1 \cup K_2 \cup R)$ , entra a facilidade  $i$ , juntamente com a restrição  $\sum_{j \in J} x_{ij} \leq a_i$ , no problema anterior. Com isso, os decrêscimos de custos, por centro de demanda  $j$ , são dados por:

$$\Delta_{ij} = \min_{\ell \in (K_1 \cup K_2 - i)} \{\max(C_{\ell j} - C_{ij}, 0)\}$$

Consequentemente,

$$\bar{\Delta}_i = \max_{0 \leq x'_{ij} \leq b_j} \left\{ \sum_{j \in J} \Delta_{ij} x'_{ij} \mid \sum_{j \in J} x'_{ij} \leq a_i \right\}$$

que é um problema da mochila resolvido por inspeção. Ainda mais, não é necessário resolver qualquer problema de transporte.

Este teste de redução apesar de sua simplicidade com

putacional perde em poder de poda. Nosso próximo passo é apresentar um novo limite inferior a  $\Delta_i$ ,  $\Delta_i^D$ , que não seja inferior a  $\bar{\Delta}_i$  e resolvendo apenas um problema de transporte.

De (II.11),  $\Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2)$ . Resolvendo o problema  $T(K_1 \cup K_2)$  obtemos o valor mínimo  $W(K_1 \cup K_2)$ . Aproximando inferiormente o valor de  $W(K_1 \cup K_2 - i)$ , podemos caracterizar um limite inferior para  $\Delta_i$ . Uma solução dual viável para o problema  $T(K_1 \cup K_2 - i)$  é um limite inferior para  $W(K_1 \cup K_2 - i)$  e portanto implica em um limite inferior para  $\Delta_i$ .

Seja o problema  $T(K_1 \cup K_2)$ . Associando as variáveis duais  $u_\ell$ ,  $\ell \in (K_1 \cup K_2)$  e  $v_j$ ,  $j \in J$ , respectivamente às restrições de oferta e demanda, temos os problemas primal e dual:

<u>Primal</u>	<u>Dual</u>
$\min \sum_{\ell \in (K_1 \cup K_2)} \sum_{j \in J} C_{\ell j} x_{\ell j}$	$\max \sum_{\ell \in (K_1 \cup K_2)} a_\ell u_\ell + \sum_{j \in J} b_j v_j$
$\sum_{j \in J} x_{\ell j} \leq a_\ell, \ell \in (K_1 \cup K_2)$	$u_\ell + v_j \leq C_{\ell j}, \ell \in (K_1 \cup K_2), j \in J$
$\sum_{\ell \in (K_1 \cup K_2)} x_{\ell j} \geq b_j, j \in J$	$u_\ell \leq 0, \ell \in (K_1 \cup K_2)$
$x_{\ell j} \geq 0, \ell \in (K_1 \cup K_2), j \in J$	$v_j \geq 0, j \in J$

Seja  $x^* = (x_{\ell j}^*)$ ,  $\ell \in (K_1 \cup K_2)$  e  $j \in J$ , a solução ótima primal e  $u_\ell^* \leq 0$ ,  $\ell \in (K_1 \cup K_2)$  e  $v_j^* \geq 0$ ,  $j \in J$ , a correspondente solução ótima dual. Logo,

$$W(K_1 \cup K_2) = \sum_{\ell \in (K_1 \cup K_2)} a_{\ell} u_{\ell}^* + \sum_{j \in J} b_j v_j^* \quad (\text{II.13})$$

Retirando a facilidade  $i \in K_2$ , o problema dual de  $T(K_1 \cup K_2 - i)$  corresponde ao dual de  $T(K_1 \cup K_2)$  a menos das restrições:

$$\begin{cases} u_i + v_j \leq C_{ij}, \quad \forall (i, j), j \in J \\ u_i \leq 0 \end{cases}$$

Logo, uma solução dual viável para  $T(K_1 \cup K_2 - i)$  é obtida fazendo:

$$u'_{\ell} = u_{\ell}^*, \quad \ell \in (K_1 \cup K_2 - i)$$

$$v'_j \leq C_{\ell j} - u'_{\ell} = C_{\ell j} - u_{\ell}^*, \quad \forall (\ell, j), \ell \in (K_1 \cup K_2 - i), j \in J$$

Ou, mais especificamente

$$v'_j = \min_{\ell \in (K_1 \cup K_2 - i)} \{C_{\ell j} - u_{\ell}^*\}, \quad j \in J$$

Ainda mais,  $v'_j \geq v_j^*$  para todo  $j \in J$  tal que  $x_{ij}^* \neq 0$

(II.14)

$$v'_j = v_j^* \quad \text{para todo } j \in J \text{ tal que } x_{ij}^* = 0$$

que pode ser mostrado pela aplicação do teorema de complementaridade das folgas.

Esta solução fornece um limite inferior para  $W(K_1 \cup K_2 - i)$  igual

a:

$$W'(K_1 \cup K_2 - i) = \sum_{\ell \in (K_1 \cup K_2 - i)} a_\ell u'_\ell + \sum_{j \in J} b_j v'_j$$

ou,

$$W'(K_1 \cup K_2 - i) = \sum_{\ell \in (K_1 \cup K_2 - i)} a_\ell u_\ell^* + \sum_{j \in J} b_j v'_j \quad (\text{II.15})$$

Definindo o novo limite inferior para  $\Delta_i$  como  $\Delta_i^D$ , resulta de (II.13) e (II.15):

$$\begin{aligned} \Delta_i^D &= W'(K_1 \cup K_2 - i) - W(K_1 \cup K_2) \\ &= \sum_{\ell \in (K_1 \cup K_2 - i)} a_\ell u_\ell^* + \sum_{j \in J} b_j v'_j - \sum_{\ell \in (K_1 \cup K_2)} a_\ell u_\ell^* + \sum_{j \in J} b_j v_j^* \\ &= -a_i u_i^* + \sum_{j \in J} b_j (v'_j - v_j^*) \end{aligned}$$

De (II.14),  $\Delta_i^D = -a_i u_i^* + \sum_{j \in J | x_{ij}^* \neq 0} b_j (v'_j - v_j^*)$

Como  $x_{ij}^* \leq b_j$  e  $\Delta_i^D$  um limite inferior, podemos substituir  $b_j$  por  $x_{ij}^*$

$$\Delta_i^D = -a_i u_i^* + \sum_{j \in J | x_{ij}^* \neq 0} x_{ij}^* (v'_j - v_j^*) \quad (\text{II.16})$$

$x_{ij}^* \neq 0$ , uma variável básica, então pelo teorema de complementaridade das folgas  $u_i^* + v_j^* - C_{ij} = 0$  ou  $v_j^* = C_{ij} - u_i^*$ . Levando esse valor em (II.16):

$$\Delta_i^D = -a_i u_i^* + \sum_{j \in J | x_{ij}^* \neq 0} x_{ij}^* (v'_j - C_{ij} + u_i^*)$$

$$\Delta_i^D = \sum_{j \in J | x_{ij}^* \neq 0} (v_j^i - c_{ij}) x_{ij}^* + u_i^* \left( \sum_{j \in J | x_{ij}^* \neq 0} x_{ij}^* - a_i \right)$$

Novamente, pelo teorema de complementaridade das folgas, a segunda parcela do lado direito é nula. Resulta:

$$\Delta_i^D = \sum_{j \in J | x_{ij}^* \neq 0} (v_j^i - c_{ij}) x_{ij}^*$$

onde

$$v_j^i = \min_{\ell \in (K_1 \cup K_2 - i)} \{c_{\ell j} - u_\ell^*\} \quad (\text{II.17})$$

Podemos mostrar que as variáveis duais  $u_i$ , além de não positivas, são não crescentes à medida que retiramos as facilidades em  $(K_1 \cup K_2)$ , (ERLENKOTTER (15)). Isto faz com que o valor  $\Delta_i^D$  torne cada vez maior.

A superioridade de  $\Delta_i^D$  sobre  $\bar{\Delta}_i$  é mostrada pelos testes computacionais apresentados posteriormente. Em todos os resultados temos  $\Delta_i^D \geq \bar{\Delta}_i$  e aproximando bastante a  $\Delta_i$ . Porém, caberia demonstrar se isto sempre acontece.

Uma outra maneira para caracterizar  $\Delta_i^D$  em (II.17) é utilizar a interpretação econômica do problema dual. Seja  $(-u_i)$  = o preço unitário do produto na facilidade  $i$   
 $(v_j)$  = o preço unitário do produto após entrega em  $j$ .

Reescrevendo as restrições duais como  $v_j \leq c_{ij} - u_i$ , podemos interpretá-las da seguinte forma: o custo de fornecer mais uma unidade a  $j$  deve ser inferior ao custo da unidade em  $i$  mais o



custo de transporte de  $i$  para  $j$ . Na solução ótima atingimos o equilíbrio, de forma que, para toda variável básica  $x_{ij}$  temos  $v_j = C_{ij} - u_i$ .

Ao retirar-se a facilidade  $i$  em  $T(K_1 \cup K_2)$ , o fluxo que sai de  $i$  para  $j$  deve ser transferido para as outras facilidades remanescentes. Supondo suas capacidades relaxadas, e como  $u_i \leq 0$  é não crescente, o fluxo de  $i$  para  $j$ ,  $x_{ij}^*$ , será transferido para uma facilidade  $k \in (K_1 \cup K_2 - i)$  tal que o novo custo da unidade em  $j$ ,  $v_j'$ , seja o menor possível, ou:

$$v_j' = \min_{\ell \in (K_1 \cup K_2 - i)} \{C_{\ell j} - u_\ell^*\}$$

para os demais  $j \in J$  tais que  $x_{ij}^* = 0$  temos  $v_j' = v_j^*$ . Repetindo o processo para todo  $j$  tal que  $x_{ij}^* \neq 0$ , o limite inferior para  $W(K_1 \cup K_2 - i)$  é dado por:

$$W'(K_1 \cup K_2 - i) = \sum_{\ell \in (K_1 \cup K_2 - i)} a_\ell u_\ell^* + \sum_{j \in J} b_j v_j'$$

que corresponde ao limite anterior, (II.15).

## 2. TESTES PARA FECHAR UMA FACILIDADE

Para todo  $i \in K_2$  faça

$$\Omega_i = W(K_1) - W(K_1 \cup i) \tag{II.18}$$

$\Omega_i$  é a variação nos custos variáveis do problema de transporte  $T(K_1)$  ao ativar a facilidade  $i$ . Pela definição, o conjunto  $K_1$ , inicialmente vazio, é crescente em número de elementos. E como  $W(\cdot)$  é uma função submodular, então pela propriedade (II.8),  $\Omega_i$  é não crescente. Com isto,  $\Omega_i$  é a economia máxima obtida, com relação aos custos variáveis, ao abrir a facilidade  $i$ . Ainda mais, se esta economia máxima for inferior ao custo fixo dessa facilidade então podemos fixá-la fechada.

Cabe observar que este teste somente é aplicável após atingirmos a viabilidade primal,  $\sum_{i \in K_1} a_i \geq \sum_{j \in J} b_j$ .

TEOREMA 2.1. Se  $\Omega_i \leq F_i$  então a facilidade  $i$  é fixada fechada, isto é,  $y_i = 0$ .

Demonstração: Seja  $\Omega_i = W(K_1) - W(K_1 \cup i)$  e  
 $\Omega_i^* = Z(K_1) - Z(K_1 \cup i)$

onde as funções  $W(\cdot)$  e  $Z(\cdot)$  são dadas, de forma semelhante, por (II.9) e (II.10), respectivamente, e  $\Omega_i^*$  é a variação real na função objetivo ao ativar a facilidade  $i$ . Mostraremos que se  $\Omega_i \leq F_i$  então  $\Omega_i^* \leq 0$  e não crescente. Logo, podemos fechar a facilidade  $i$ , pois, caso contrário, poderemos ter um acréscimo no custo total do sistema.

Como  $W(\cdot)$  é não crescente logo  $\Omega_i \geq 0$ .

Seja  $\Omega_i > 0$ . Então a facilidade  $i$  está em atividade ao resolver o problema de transporte  $T(K_1 \cup i)$ , ou  $i \in S \subseteq (K_1 \cup i)$  tal que

$W(S)$  é mínimo. Portanto, ao adicionar  $i$  a  $K_1$  temos que  $Z(K_1) - \Omega_i + F_i = Z(K_1 \cup i)$ . Se  $\Omega_i \leq F_i$  então  $Z(K_1) \leq Z(K_1 \cup i)$  ou  $\Omega_i^* \leq 0$ .

Seja  $\Omega_i = 0$ . Então independente do estado da facilidade  $i$ , se  $\Omega_i = 0 \leq F_i$  resulta  $Z(K_1) \leq Z(K_1 \cup i)$  ou  $\Omega_i^* \leq 0$ .

Mostraremos a seguir que ao ativar  $i$  nenhum elemento  $k \in K_1$  será desativado, garantindo o resultado acima,  $\Omega_i^* \leq 0$ . Por absurdo, suponha que  $k \in K_1$  é desativado ao ativar  $i$ . Então,  $\Omega_k = W(K_1 \cup i - k) - W(K_1 \cup i) = 0$ .

Pela submodularidade de  $W(\cdot)$ , (II.8),

$$\begin{aligned} \Delta_k &= W(K_1 \cup K_2 - k) - W(K_1 \cup K_2) \leq W(K_1 \cup i - k) - W(K_1 \cup i) = \\ &= \Omega_k = 0. \end{aligned}$$

Portanto  $\Delta_k \leq 0 \leq F_i$ , pelo teorema (1.1), a facilidade  $i$  não foi aberta anteriormente. Logo  $k$  não será desativado.

Se  $\Omega_i \leq F_i$  então  $\Omega_i^* \leq 0$ . Pela submodularidade de  $Z(\cdot)$  e como  $K_1$  é um conjunto crescente em número de elementos então podemos afirmar que  $\Omega_i^*$  é não crescente. A economia máxima em abrir uma facilidade  $i$  é não positiva,  $\Omega_i^* \leq 0$ , podemos fechar  $i$ . ●

Este teste como o anterior exige a resolução de  $|K_2| + 1$  problemas de transporte com um consumo elevado de tempo computacional.

Também aqui, o processo alternativo é determinar um limite superior para  $\Omega_i$ , diminuindo a capacidade de poda, porém, ganhando em simplicidade computacional. AKINC e KHUMAWALA (1) propuseram um limite superior para  $\Omega_i$  onde apenas um problema de transporte é resolvido. Denominaremos-lo  $\bar{\Omega}_i$ , de forma que:

$$\text{se } \Omega_i \leq \bar{\Omega}_i \leq F_i, \quad i \in K_2, \quad \text{então } y_i = 0.$$

Posteriormente proporemos uma pequena alteração no cálculo do limite anterior visando obter um limite mais justo,  $\Omega_i^D$ .

De (II.18),  $\Omega_i = W(K_1) - W(K_1 \cup i)$ . Resolvendo o problema  $T(K_1)$  e relaxando  $T(K_1 \cup i)$  obtemos um limite superior  $\bar{\Omega}_i$  para  $\Omega_i$ .

Seja o problema de transporte  $T(K_1)$ :

$$W(K_1) = \min \sum_{\ell \in K_1} \sum_{j \in J} C_{\ell j} x_{\ell j} \quad (\text{II.19})$$

$$\sum_{j \in J} x_{\ell j} \leq a_{\ell}, \quad \ell \in K_1 \quad (u_{\ell}) \quad (\text{II.20})$$

$$\sum_{\ell \in K_1} x_{\ell j} \geq b_j, \quad j \in J \quad (v_j) \quad (\text{II.21})$$

$$x_{\ell j} \geq 0, \quad \ell \in K_1, \quad j \in J \quad (\text{II.22})$$

Seja a solução ótima primal  $x^* = (x_{\ell j}^*)$ ,  $\ell \in K_1, j \in J$  e corresponde solução dual  $u_{\ell}^*$ ,  $\ell \in K_1$  e  $v_j^*$ ,  $j \in J$ . Portanto o va

lor m̃nimo para funçãõ objetivo em  $T(K_1)$ :

$$W(K_1) = \sum_{\ell \in K_1} a_\ell u_\ell^* + \sum_{j \in J} b_j v_j^* \quad (\text{II.23})$$

Seja o problema  $T(K_1 \cup i)$ :

$$W(K_1 \cup i) = \min_{\ell \in (K_1 \cup i)} \sum_{j \in J} C_{\ell j} x_{\ell j} \quad (\text{II.24})$$

$$\sum_{j \in J} x_{ij} \leq a_i \quad (\text{II.25})$$

$$\sum_{j \in J} x_{\ell j} \leq a_\ell, \ell \in K_1 (u_\ell) \quad (\text{II.26})$$

$$\sum_{\ell \in (K_1 \cup i)} x_{\ell j} \geq b_j, j \in J (v_j) \quad (\text{II.27})$$

$$x_{\ell j} \geq 0, \ell \in (K_1 \cup i), j \in J \quad (\text{II.28})$$

Um limite inferior para  $W(K_1 \cup i)$  ẽ dado pelo Lagrangeano  $L(K_1 \cup i)$ , relaxando as restrições (II.26) e (II.27) e utilizando a soluçãõ dual para  $T(K_1)$  como multiplicadores:

$$L(K_1 \cup i) = \min_{0 \leq x_{\ell j} \leq b_j} \left\{ \sum_{\ell \in (K_1 \cup i)} \sum_{j \in J} C_{\ell j} x_{\ell j} + \sum_{\ell \in K_1} u_\ell^* (a_\ell - \sum_{j \in J} x_{\ell j}) + \sum_{j \in J} v_j^* (b_j - \sum_{\ell \in (K_1 \cup i)} x_{\ell j}) \right\}$$

$$\text{sujeito a } \sum_{j \in J} x_{ij} \leq a_i \quad (\text{II.29})$$

Levando (II.23) em (II.29) resulta:

$$L(K_1 U i) = W(K_1) + \min_{\substack{0 \leq x_{\ell j} \leq b_j \\ \ell \in (K_1 U i)}} \left\{ \sum_{\ell \in K_1} \sum_{j \in J} (C_{\ell j} - u_{\ell}^* - v_j^*) x_{\ell j} + \sum_{j \in J} (C_{ij} - v_j^*) x_{ij} \right\}$$

$$\text{sujeito a } \sum_{j \in J} x_{ij} \leq a_i .$$

Pela otimalidade da solução dual em  $T(K_1)$  temos que  $(C_{\ell j} - u_{\ell}^* - v_j^*) \geq 0$  para todo  $(\ell, j)$ ,  $\ell \in K_1$ ,  $j \in J$ . Portanto podemos fazer  $x_{\ell j} = 0$  para  $\ell \in K_1$  e  $j \in J$ , na expressão acima:

$$L(K_1 U i) = W(K_1) + \min_{0 \leq x_{ij} \leq b_j} \left\{ \sum_{j \in J} (C_{ij} - v_j^*) x_{ij} \mid \sum_{j \in J} x_{ij} \leq a_i \right\}$$

ou

$$L(K_1 U i) = W(K_1) - \max_{0 \leq x_{ij} \leq b_j} \left\{ \sum_{j \in J} (v_j^* - C_{ij}) x_{ij} \mid \sum_{j \in J} x_{ij} \leq a_i \right\}$$

Substituindo  $W(K_1 U i)$  por  $L(K_1 U i)$  em  $\Omega_i$  resulta  $\bar{\Omega}_i$ :

$$\bar{\Omega}_i = W(K_1) - L(K_1 U i)$$

$$\bar{\Omega}_i = \max_{0 \leq x_{ij} \leq b_j} \left\{ \sum_{j \in J} (v_j^* - C_{ij}) x_{ij} \mid \sum_{j \in J} x_{ij} \leq a_i \right\}$$

Problema semelhante ao anterior para o cálculo de  $\bar{\Delta}_i$ , portanto de fácil resolução e exigindo o cálculo de apenas um problema de transporte.

Nosso próximo passo é apresentar um limite superior a  $\Omega_i$ , porém, que não seja superior a  $\bar{\Omega}_i$ . Isso pode ser obtido através de uma interpretação semelhante à apresentada para o cálculo de  $\Delta_i^D$ . Este novo limite, que denominaremos de  $\Omega_i^D$ , utiliza além das variáveis duais do problema  $T(K_1)$  a solução ótima primal  $x^* = (x_{\ell j}^*)$ ,  $\ell \in K_1$ ,  $j \in J$ .

De forma semelhante, uma solução dual viável para o problema  $T(K_1 \cup i)$  fornece um limite inferior para  $W(K_1 \cup i)$ . E da resolução do problema  $T(K_1)$  resulta  $W(K_1)$ . Levando em (II.18) temos um limite superior para  $\Omega_i$ .

Da resolução de  $T(K_1)$  temos  $x^*$ ,  $u^*$ ,  $v^*$  e  $W(K_1)$  dado por (II.23).

Adicionar a facilidade  $i$  corresponde em acrescentar ao problema dual de  $T(K_1)$  as restrições  $u_i + v_j \leq C_{ij}$ , para todo  $j \in J$ .

Uma solução dual viável para  $T(K_1 \cup i)$  é obtida fazendo:

$$u_i' = 0$$

$$u_{\ell}' = u_{\ell}^*, \quad \ell \in K_1 \tag{II.30}$$

$$v_j' = \min \{C_{ij}, v_j^*\}, \quad j \in J$$

Esta solução fornece um limite inferior para  $W(K_1 U i)$  dado por:

$$W'(K_1 U i) = \sum_{\ell \in K_1} a_\ell u'_\ell + a_i u'_i + \sum_{j \in J} b_j v'_j$$

$$W'(K_1 U i) = \sum_{\ell \in K_1} a_\ell u_\ell^* + \sum_{j \in J} b_j v'_j \quad (\text{II.31})$$

Levando (II.23) e (II.31) em (II.18), resulta  $\Omega_i^D$  igual a:

$$\begin{aligned} \Omega_i^D &= W(K_1) - W'(K_1 U i) \\ &= \sum_{j \in J} (v_j^* - v'_j) b_j \end{aligned}$$

Como  $\Omega_i^D$  é um limite superior e  $\sum_{\ell \in K_1} x_{\ell j}^* \geq b_j$ ,  $j \in J$  então podemos fazer:

$$\begin{aligned} \Omega_i^D &= \sum_{j \in J} (v_j^* - v'_j) \sum_{\ell \in K_1} x_{\ell j}^* \\ &= \sum_{\ell \in K_1} \sum_{j \in J} x_{\ell j}^* (v_j^* - v'_j) \end{aligned}$$

Substituindo  $v'_j$ :

$$\begin{aligned} \Omega_i^D &= \sum_{\ell \in K_1} \sum_{j \in J} x_{\ell j}^* (v_j^* - \min \{C_{ij}, v_j^*\}) \\ &= \sum_{\ell \in K_1} \sum_{j \in J} x_{\ell j}^* \max (0, v_j^* - C_{ij}) \quad (\text{II.32}) \end{aligned}$$



Seja  $\bar{x}$  uma solução primal ótima em  $T(K_1 \cup i)$  e  $\bar{u}$  e  $\bar{v}$  solução dual correspondente. Então pelo teorema de complementaridade das folgas temos  $(C_{ij} - \bar{u}_i - \bar{v}_j) \bar{x}_{ij} = 0$  para todo par  $(i, j)$ ,  $j \in J$ . Fixando  $\bar{u}_i = 0$  então  $(C_{ij} - \bar{v}_j) \bar{x}_{ij} = 0$ . Com isto, podemos retirar as seguintes conclusões:

$$\bar{x}_{ij} \geq 0 \text{ se } (C_{ij} - \bar{v}_j) = 0 \text{ e portanto } \bar{v}_j = C_{ij} \quad (\text{II.33})$$

$$\bar{x}_{ij} = 0 \text{ se } (C_{ij} - \bar{v}_j) \geq 0 \text{ e portanto } \bar{v}_j \leq C_{ij}$$

Pela mesma razão em (II.30), resulta de (II.33):

$$\bar{x}_{ij} \geq 0 \text{ então } \bar{v}_j = C_{ij} \leq v_j^* \Rightarrow \max(0, v_j^* - C_{ij}) \geq 0$$

$$\bar{x}_{ij} = 0 \text{ então } \bar{v}_j \leq C_{ij} \geq v_j^* \Rightarrow \max(0, v_j^* - C_{ij}) = 0$$

A parcela  $\max(0, v_j^* - C_{ij})$  poderá ser positiva somente se  $\bar{x}_{ij} \geq 0$ , caso contrário será nula. Ainda mais, como  $\bar{x}$  é uma solução primal então deve satisfazer as restrições de oferta,  $\sum_{j \in J} \bar{x}_{ij} \leq a_i$ . E este somatório é a quantidade de fluxo que sai de  $i$  para todo  $j \in J$  e que anteriormente, em  $T(K_1)$ , saía dos demais elementos  $l \in K_1$ .

Considerando que  $\Omega_i^D$  é um limite superior, de (II.32) e das conclusões acima, podemos determinar um fluxo  $\bar{x}_{lj} \leq x_{lj}^*$ ,  $l \in K_1$ ,  $j \in J$ , tal que  $\Omega_i^D$  seja máximo. Essa nova variável  $\bar{x}_{lj}$  apresentará o fluxo que saía de  $l$  para  $j$  em  $T(K_1)$  e que poderá sair de  $i$  para  $j$  em  $T(K_1 \cup i)$ . Uma vez que a capacidade da facilidade  $i$  é limitada, faz-se necessário estabelecer  $\sum_{l \in K_1} \sum_{j \in J} \bar{x}_{lj} \leq a_i$ .

Levando esses resultados em (II.32) resulta:

$$\begin{aligned} \Omega_i^D &= \max \sum_{\ell \in K_1} \sum_{j \in J} \hat{x}_{\ell j} \cdot \max(0, v_j^* - C_{ij}) \\ &\text{s.a. } \sum_{\ell \in K_1} \sum_{j \in J} \hat{x}_{\ell j} \leq a_i \\ &\hat{x}_{\ell j} \leq x_{\ell j}^*, \ell \in K_1 \text{ e } j \in J \end{aligned} \quad (\text{II.34})$$

Como  $\hat{x}_{\ell j} \leq x_{\ell j}^*$ , somente interessa os arcos básicos, tais que  $x_{\ell j}^* \neq 0$ , em  $T(K_1)$ . É um problema da mochila semelhante ao anterior, onde, além das variáveis duais do problema de transporte  $T(K_1)$ , também utilizamos as suas variáveis básicas primais.

De forma semelhante as conclusões para  $\Delta_i^D$ , podemos mostrar que as variáveis duais  $v_j$ , além de não negativas, são também não crescentes à medida que adicionamos facilidades em  $K_1$  (ERLENKOTTER (15)). Isto faz com que  $\Omega_i^D$  torne cada vez menor, aumentando a possibilidade de vir a fechar a facilidade  $i$  a cada passo.

O melhor desempenho de  $\Omega_i^D$  com relação a  $\bar{\Omega}_i$  também será mostrado pelos testes computacionais. Em todos os resultados temos  $\Omega_i^D \leq \bar{\Omega}_i$  e, em muitos casos, igual a  $\bar{\Omega}_i$ . Mas, caberá demonstrar se isto sempre acontece.

Da mesma maneira ao elaborado anteriormente,  $\Omega_i^D$  pode ser caracterizado utilizando a interpretação econômica do problema dual. Como  $v_j$  é não crescente ao acrescentar uma facilidade  $i$ , o preço do produto no destino  $j$  é não crescente. Conse

quentemente, uma estimativa, por cima, do custo unitário de transferir o fluxo  $x_{\ell j}^* \neq 0$ , de uma facilidade  $\ell \in K_1$ , para a nova facilidade  $i$ , é dado por  $v_j^i = v_j^*$ . E portanto, a variação no custo, correspondente a  $x_{\ell j}^*$  é dado por:

$$\max (0, v_j^* - c_{ij}) x_{\ell j}^*$$

Lembrando que a capacidade da facilidade  $i$  é limitada em  $a_i$ , a soma máxima de todas as variações no custo, nos fornece um limite superior para  $\Omega_i$ ,  $\Omega_i^D$ , que equivale a (II.32).

### 3. TESTES COMPLEMENTARES

Esta etapa consiste em explorar ainda mais os testes anteriores, aproveitando ao máximo as informações e cálculos já efetuados. Esta análise poderá implicar em abrir ou fechar uma facilidade, mesmo em casos em que os testes anteriores falharam. Ou, ainda, forçar a definição de algumas delas.

Para os algoritmos de enumeração podemos utilizar as informações dos testes anteriores com o objetivo de definir penalidades que implicarão em fechar ou abrir facilidades. Estas penalidades estão amarradas as definições dos limites inferiores e superiores e restrições tais como o número máximo e mínimo de facilidades a serem abertas. Serão analisadas para o algoritmo de enumeração apresentado no capítulo (V).

Outros testes foram desenvolvidos, porém, devido ao baixo desempenho, observado na prática e apresentado nos resultados dos capítulos seguintes, dispensamos um tratamento formal e passamos a descrevê-los no Apêndice C.

#### 4. CONCLUSÕES

A eficiência desses testes está intimamente relacionada às capacidades das facilidades,  $a_i$ ,  $i \in I$ , e aos custos fixos,  $F_i$ ,  $i \in I$ . Assim é que, para facilidades com capacidades elevadas e custos fixos pequenos, os testes de redução aproximados fornecem valores, na maioria das vezes, iguais aos obtidos pelos testes exatos. Eleva-se com isso o poder de poda dos testes aproximados. Ao passo que, se as capacidades das facilidades são pequenas e os custos fixos elevados, temos um distanciamento entre os testes exatos e os aproximados, diminuindo o poder de poda dos aproximados.

Todos estes testes são dependentes de  $K_1$  e  $K_2$ . À medida que estes conjuntos sofrem alterações, consequência da aplicação de algum cálculo, todos os demais testes devem ser executados novamente. São utilizados de maneira cíclica até que nenhuma alteração ocorra nos conjuntos  $K_1$  e  $K_2$ .

Finalmente, cabe ressaltar que, quase todos estes testes podem ser expandidos para o problema em dois níveis de distribuição, ou seja, centros de produção-facilidades-centros de demanda.

CAPÍTULO IIIUM ALGORITMO GULOSO PARA FUNÇÃO CUSTO FIXO/LINEAR

O problema proposto é da forma:

$$\begin{aligned} \min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i \left( \sum_{j \in J} x_{ij} \right) + \\ & + \sum_{i \in I} F_i y_i \end{aligned} \quad (\text{III.1})$$

(P1)

$$\sum_{j \in J} x_{ij} \leq a_i y_i, \quad i \in I \quad (\text{III.2})$$

$$\sum_{i \in I} x_{ij} = b_j, \quad j \in J \quad (\text{III.3})$$

$$x_{ij} \geq 0, \quad y_i \in \{0, 1\}, \quad \forall_i \text{ e } \forall_j \quad (\text{III.4})$$

Seja  $f_i(\cdot)$  uma função linear ou  $f_i \left( \sum_{j \in J} x_{ij} \right) = t_i \sum_{j \in J} x_{ij}$ , onde  $t_i, i \in I$ , é uma constante. Fazendo  $C_{ij} = c_{ij} + t_i$ , para todo  $(i, j)$ , resulta o problema a ser estudado neste capítulo:

$$\min \quad \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in I} F_i y_i$$

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

(P2)

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

$$x_{ij} \geq 0, \quad y_i \in \{0, 1\}, \quad \forall_i \text{ e } \forall_j$$

Para cada arco  $(i, j)$  temos uma função custo fixo mais variável linear. Veja figura (III.1).

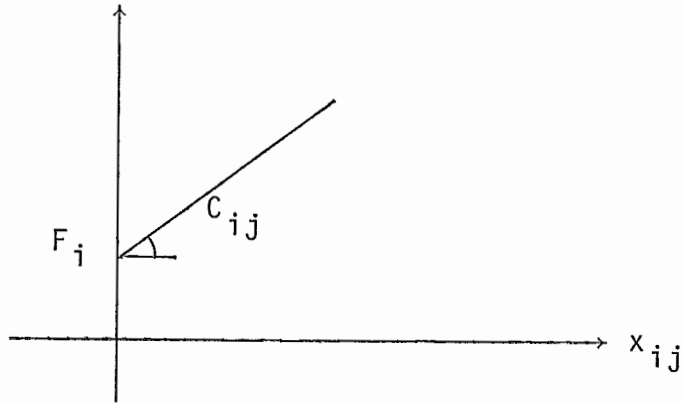


Figura (III.1) - Função custo fixo/linear.

Alguns dos algoritmos para tratamento do modelo (P2) foram discutidos no capítulo I. Aqui apresentaremos uma heurística gulosa. Sua característica básica é procurar abrir a facilidade que proporcione o maior decréscimo da função a cada iteração. Porém, antes desse processo iterativo, aplicaremos os testes de redução para abrir uma facilidade, apresentados no capítulo anterior, visando abrir o maior número possível de facilidades. Faremos uma análise do algoritmo com e sem essa etapa inicial além de comparar o desempenho dos testes de redução.

## 1. ALGORITMO GULOSO

Seja inicialmente os conjuntos:

$K_0 = \emptyset$  - conjunto das facilidades fechadas;

$K_1 = \emptyset$  - conjunto das facilidades abertas;

$K_2 = I = \{1, \dots, m\}$  - conjunto das facilidades livres.

O algoritmo guloso proposto para o modelo (P2) é composto por duas etapas:

- (1) Aplicando um dos testes de redução para abrir facilidades, apresentados no capítulo II, transferir os elementos abertos de  $K_2$  para  $K_1$ .
- (2) Executar um processo iterativo, transferindo para  $K_1$ , a cada iteração, o elemento  $i \in K_2$  que forneça o maior retorno:  $\Omega_i - F_i = W(K_1) - W(K_1 \cup i) - F_i$ , enquanto  $\Omega_i - F_i > 0$ , onde  $\Omega_i$  é dado pelo teste de redução para fechar uma facilidade.

Na primeira etapa calculamos  $\Delta_i$  (ou  $\bar{\Delta}_i$  ou  $\Delta_i^D$ ) para todo  $i \in K_2$ . Se  $\Delta_i \geq F_i$  então  $i$  será retirado de  $K_2$  e lançado em  $K_1$ . Com este conjunto passamos a segunda etapa.

A determinação de  $\Omega_i - F_i$  será feita utilizando apenas  $\bar{\Omega}_i$  e  $\Omega_i^D$  uma vez que o cálculo de  $\Omega_i$ , para todo  $i \in K_2$ , iterativamente é bastante custoso. Seja  $\bar{\Omega}_i - F_i$  (ou  $\Omega_i^D - F_i$ ) para todo  $i \in K_2$ . A heurística gulosa seleciona em cada iteração aquele elemento  $i$  tal que  $\bar{\Omega}_i - F_i > 0$  e que proporcione o maior retorno ou o maior decréscimo na função. A facilidade  $i_m$ , tal que  $\bar{\Omega}_{i_m} - F_{i_m}$  seja o maior dos  $\bar{\Omega}_i - F_i$ , será retirada de  $K_2$  e lançada em  $K_1$ . Voltamos a calcular novos valores para  $\bar{\Omega}_i - F_i$ , para todo  $i \in K_2$ , selecionamos nova facilidade  $i_m$  e sucessivamen

te até que  $\bar{\Omega}_i - F_i \leq 0$ , para todo  $i \in K_2$ , ou  $K_2 = \emptyset$ .

Cabe observar que:

(a) O algoritmo pode ser implementado apenas com a segunda etapa, porém, o acréscimo da primeira aumenta significativamente o seu desempenho.

(b) Ao aplicar a segunda etapa trabalhamos com um conjunto  $K_1$  que não necessariamente atende a condição de viabilidade,  $\sum_{i \in K_1} a_i \geq \sum_{j \in J} b_j$ . Este problema será discutido na seção de Implementação, a seguir.

ALGORITMO GULOSO:

Passo 0. Seja  $K_1 = \emptyset$  e  $K_2 = I = \{1, 2, \dots, m\}$ .

Passo 1. Execute um dos testes de redução para abrir facilidades ( $\Delta_i$ ,  $\bar{\Delta}_i$  ou  $\Delta_i^D$ ) para todo  $i \in K_2$ . Se  $\Delta_i \geq F_i$ ,  $i \in K_2$  então faça  $K_1 = K_1 \cup \{i\}$ ,  $K_2 = K_2 - \{i\}$ .

Passo 2. Calcule  $W(K_1)$ .

Passo 3. Se  $K_2 = \emptyset$  então PARE.  $K_1$  é o conjunto de facilidades abertas e  $Z(K_1) = W(K_1) + F(K_1)$ .

Passo 4. Para todo  $i \in K_2$  calcule  $\bar{\Omega}_i - F_i$  (ou  $\Omega_i^D - F_i$ ).

Passo 5. Determine  $i_m \in K_2$  tal que  $\bar{\Omega}_{i_m} - F_{i_m} = \max_{i \in K_2} (\bar{\Omega}_i - F_i)$ .



Passo 6. Se  $(\bar{\Omega}_{i_m} - F_{i_m}) \leq 0$  então PARE.  $K_1$  é o conjunto de facilidades abertas.

Passo 7. Calcule  $W(K_1 \cup i_m)$  e  $\Omega_{i_m} - F_{i_m}$ .

Passo 8. Se  $(\Omega_{i_m} - F_{i_m}) \leq 0$  então PARE.  $K_1$  é o conjunto de facilidades abertas.

Passo 9. Faça  $W(K_1) = W(K_1 \cup i_m)$ ,  $K_1 = K_1 \cup i_m$ ,  $K_2 = K_2 - i_m$ . Retorne ao Passo 3.

## 2. IMPLEMENTAÇÃO

Discutiremos alguns detalhes para implementação seguindo as necessidades do algoritmo guloso passo a passo.

Trabalhamos com os conjuntos  $K_1$  e  $K_2$  inicialmente definidos como  $K_1 = \emptyset$  e  $K_2 = \{1, \dots, m\}$ . Eles serão utilizados na composição de problemas de transporte com o objetivo de calcular a função  $W(S)$ ,  $S \subseteq I$ . A viabilidade de um problema de transporte  $T(S)$  é garantida pela condição  $\sum_{i \in S} a_i \geq \sum_{j \in J} b_j$ ,  $S \subseteq I$ . Ainda mais, essa condição deve ser atendida na igualdade para aplicação do algoritmo para resolução de um problema de transporte. Viável ou não, essa igualdade será garantida adicionando um centro de oferta  $i^a$  e um centro de demanda  $j^a$ , artificiais, com capacidades dadas por:

$$a_{ia} = \sum_{j \in J} b_j, \quad b_{ja} = \sum_{i \in S} a_i$$

E custos de transporte iguais a:

$$C_{iaj} = \infty, \text{ para todo } j \in J$$

$$C_{ija} = 0, \text{ para todo } i \in S$$

$$C_{ija} = 0$$

Assim, dado um conjunto  $S$  de facilidades e  $J$  de centros de de manda, o problema de transporte  $T(S)$ , será resolvido com os con juntos  $(S \cup i^a)$  e  $(J \cup j^a)$ . Valores  $C_{iaj} = \infty, \forall j \in J$ , faz com que a capacidade do centro de oferta artificial somente seja utilizada quando o problema for inviável e portanto  $W(S) = \infty$  e  $Z(S) = \infty$ . Caso contrário, toda capacidade de  $i^a$  será descarregada em  $j^a$  a custo nulo, não alterando o valor real da função.

Portanto, ao iniciar o algoritmo fazemos  $K_1 = \{m + 1\}$  e  $J = J \cup \{n + 1\}$ , onde  $(m + 1)$  e  $(n + 1)$  são respectivamente os elementos  $i^a$  e  $j^a$ , com capacidades e custos definidos acima.

O acréscimo dos dois centros artificiais também permite um maior desempenho e flexibilidade na resolução de um problema de transporte, seja  $T(S \cup i)$ , utilizando como solução inicial a solução final de um outro,  $T(S)$ . Acrescentar um elemento  $i$  a  $S$  corresponde em adicionar uma linha ao tableau do problema de transporte  $T(S)$ . Essa linha terá as seguintes características:

$$(a) \ x_{ij} = 0, \ \forall j \in J \quad e \quad x_{ij^a} = a_i$$

(b) Os custos reduzidos da linha são dados por:

$C_{ij}^! - u_i^! - v_j^! = C_{ij} - v_j^*$ ,  $\forall j \in (J \cup j^a)$ , onde  $v_j^*$  é a solução dual correspondente a  $T(S)$ . Se  $C_{ij}^! - u_i^! - v_j^! \geq 0$ ,  $\forall j \in (J \cup j^a)$ , então a solução é ótima para  $T(S \cup i)$ , caso contrário, iniciamos o processo iterativo do algoritmo de transporte, para resolução de  $T(S \cup i)$ , a partir dessa solução inicial bastante próxima da ótima.

De forma semelhante, podemos resolver o problema de transporte  $T(S - i)$ , tomando como solução inicial a solução final para  $T(S)$ . Basta fazer os custos  $C_{ij} = \infty$ ,  $\forall j \in J$ . Com isso forçamos as variáveis básicas  $x_{ij}$ , para algum (ns)  $j \in J$ , saírem da base. Toda a oferta de  $i$  será descarregada em  $j^a$  a custo nulo.

No Passo 1, correspondente a primeira etapa do algoritmo, aplicamos um dos testes de redução para abrir facilidades. A comparação entre  $\Delta_i$ ,  $\bar{\Delta}_i$  e  $\Delta_i^D$  será feita na seção seguinte.

Também no Passo 4, escolhemos um dos testes de redução para fechar facilidades. Comparamos  $\bar{\Omega}_i$  e  $\Omega_i^D$  e os resultados serão apresentados na próxima seção. Porém, cabe observar que estamos utilizando um teste para fechar com o objetivo de escolher a facilidade a ser aberta. Escolhendo o máximo  $\bar{\Omega}_i - F_i$  estamos fazendo opção por abrir a facilidade com a menor chance de ser fechada naquela iteração.

$\bar{\Omega}_i$  e  $\Omega_i^D$  são superiores ou iguais a  $\Omega_i$ . Logo, temos  $\bar{\Omega}_i - F_i \geq \Omega_i - F_i$  (ou  $\Omega_i^D - F_i \geq \Omega_i - F_i$ ) e pelo Passo 6 do algoritmo, a tendência é abrir um número maior de facilidades. Tentando contornar esse problema, acrescentamos os Passos 7 e 8. Caracterizado  $i_m$ , calculamos  $W(K_1 U i_m)$  e a diferença real  $\Omega_{i_m} - F_{i_m} = W(K_1) - W(K_1 U i_m) - F_i$  no Passo 7. Se essa diferença é não positiva então paramos o processo iterativo, caso contrário as informações relativas ao problema de transporte  $T(K_1 U i_m)$  serão utilizadas na próxima iteração com o novo conjunto  $K_1 = K_1 U i_m$ .

Utilizamos um algoritmo primal, com estruturas otimizadas e unidimensionais, para o problema de transporte.

### 3. ANÁLISE DE RESULTADOS

O algoritmo foi testado utilizando um conjunto de cinco problemas básicos: I, III, IV, V e VI, descritos no Apêndice D. Os custos fixos são iguais para todas as facilidades. Porém, a influência desses custos é verificada resolvendo cada um dos problemas com os custos fixos: \$ 7.500, \$ 12.500, \$ 17.500 e \$ 25.000, implicando respectivamente nos problemas I.1, I.2, I.3, I.4 ou III.1, III.2, III.3 e III.4, etc., resultando em 20 problemas testes.

Visando cobrir todo o trabalho teórico apresentado, possibilitando retirar maiores conclusões, foi testado para ca

da um dos 20 problemas um conjunto básico de combinações. Primeiramente, aplicar o guloso em sua forma padrão, composto apenas pela segunda etapa (retirando o Passo 1 do algoritmo), o qual denominamos GULOSO PURO. Ou, executar as duas etapas, GULOSO COM PODA, que consiste em aplicar no Passo 1 do algoritmo, para todo  $i \in K_2$ , uma das podas:  $\Delta_i$  (PODA EXATA) ou  $\bar{\Delta}_i$  ou  $\Delta_i^D$  (PODAS APROXIMADAS). Na segunda etapa ou processo iterativo, utilizamos no Passo 4, para todo  $i \in K_2$ , os valores de  $\bar{\Omega}_i$  ou  $\Omega_i^D$ . Ainda paralelo a utilização de  $\bar{\Delta}_i$  com  $\bar{\Omega}_i$  ou  $\bar{\Delta}_i$  com  $\Omega_i^D$ , podemos ou não verificar a influência dos testes complementares (seção II.3) os quais denominaremos de EXTENSÃO aos testes anteriores.

Com isso resultou um conjunto de sete combinações para cada um dos 20 problemas:

1. Guloso Puro com  $\bar{\Omega}_i$
2. Guloso Puro com  $\Omega_i^D$
3. Poda  $\bar{\Delta}_i$  sem Extensão mais Guloso com  $\Omega_i^D$
4. Poda  $\bar{\Delta}_i$  com Extensão mais Guloso com  $\bar{\Omega}_i$
5. Poda  $\bar{\Delta}_i$  com Extensão mais Guloso com  $\Omega_i^D$
6. Poda  $\Delta_i^D$  mais Guloso com  $\Omega_i^D$
7. Poda  $\Delta_i$  mais Guloso com  $\Omega_i^D$

Reunimos nas tabelas (III.1), (III.2) e (III.3) os valores médios para tempos computacionais, número médio de problemas de transporte resolvidos e de iterações executadas na reso

lução desses problemas e em (III.4) o número médio de facilidades abertas com as combinações acima. Posteriormente, faremos uma análise mais detalhada e comparativa.

A tabela (III.1) apresenta os valores médios para as combinações (1) e (2). A tabela (III.2) os valores médios para a combinação (7). Os tempos médios resultantes da aplicação da primeira etapa, poda exata  $\Delta_i$ , e da segunda estão em colunas distintas. Conforme dito no capítulo anterior, a poda exata eleva substancialmente os valores computacionais o que não ocorre com as podas aproximadas. Isto pode ser verificado na tabela (III.3) que reúne os resultados relativos as demais combinações.

Na tabela (III.4) o número médio de facilidades abertas para o Guloso Puro é o mesmo quer aplicando  $\bar{\Omega}_i$  ou  $\Omega_i^D$ . O mesmo ocorre para o Guloso com a Poda  $\bar{\Delta}_i$  com ou sem Extensão. Também podemos verificar que o número de facilidades abertas pelas podas é bastante elevado comparado ao número total de facilidades abertas (última coluna) com destaque para  $\Delta_i^D$  que além de superar  $\bar{\Delta}_i$ , aproxima bastante de  $\Delta_i$ .

De posse dos resultados apresentados pelas tabelas (III.1), (III.2), (III.3) e (III.4) faremos algumas comparações de interesse: guloso puro versus com poda, guloso puro usando  $\bar{\Omega}_i$  ou  $\Omega_i^D$ , guloso com poda usando  $\Delta_i$ ,  $\bar{\Delta}_i$  ou  $\Delta_i^D$ , guloso com a poda  $\bar{\Delta}_i$  usando  $\bar{\Omega}_i$  ou  $\Omega_i^D$  e a introdução da Extensão ou testes complementares.

PROBLEMA	GULOSO PURO					
	$\bar{\Omega}_i$			$\Omega_i^D$		
	TEMPO (SEG.)	Nº MÈDIO PT	Nº MÈDIO ITERAÇÕES	TEMPO (SEG.)	Nº MÈDIO PT	Nº MÈDIO ITERAÇÕES
I.1	49.130	17	174	44.812	17	174
I.2	42.498	14	163	39.874	14	163
I.3	40.272	10	143	40.045	10	143
I.4	30.311	7	130	32.112	7	130
III.1	41.634	15	172	40.469	15	172
III.2	39.270	12	154	39.682	12	154
III.3	36.129	10	146	40.574	10	146
III.4	30.789	8	138	35.094	8	138
IV.1	41.480	16	182	42.687	16	182
IV.2	42.413	15	180	40.814	15	180
IV.3	40.559	15	180	42.116	15	180
IV.4	34.951	15	180	39.612	15	180
V.1	37.649	14	192	40.979	14	192
V.2	36.063	12	183	37.013	12	183
V.3	35.076	11	181	35.626	11	181
V.4	30.433	9	163	29.683	9	163
VI.1	75.580	20	238	77.622	20	238
VI.2	71.754	19	235	78.040	19	235
VI.3	71.474	17	221	74.244	17	221
VI.4	69.775	15	210	70.808	15	210

PT - Problema de Transporte.

TABELA (III.1) - VALORES MÈDIOS GULOSO PURO.

PROBLEMA	PODA EXATA ( $\Delta_i$ ) + GULOSO				
	TEMPO $\Delta_i$ (SEG.)	TEMPO GULOSO ( $\Omega_i^D$ )	TEMPO TOTAL (SEG.)	NÚMERO MÉDIO PT	NÚMERO MÉDIO ITERAÇÕES PT
I.1	45.713	43.004	88.717	32	244
I.2	45.908	39.578	85.486	33	244
I.3	45.155	32.747	77.902	31	209
I.4	42.091	26.097	68.188	30	200
III.1	46.563	40.813	87.376	33	244
III.2	42.117	35.899	78.016	31	228
III.3	43.649	29.447	73.096	31	213
III.4	40.618	26.694	67.312	31	209
IV.1	60.560	31.776	92.336	21	323
IV.2	58.530	28.986	87.516	20	321
IV.3	60.304	29.530	89.834	20	321
IV.4	61.980	30.705	92.685	20	321
V.1	24.247	23.117	47.364	21	218
V.2	26.674	22.821	49.495	21	212
V.3	24.197	23.531	47.728	22	235
V.4	23.400	24.769	48.169	22	255
VI.1	62.948	75.899	138.847	35	382
VI.2	54.040	66.502	120.542	35	355
VI.3	58.936	70.968	129.904	35	352
VI.4	58.971	69.716	128.687	36	348

PT - Problema de Transporte.

TABELA (III.2) - VALORES MÉDIOS GULOSO COM PODA EXATA.



PROBLEMA	PODA APROXIMADA + GULOSO							
	TEMPO (SEG.) $\bar{\Delta}_i$		TEMPO $\Delta_i^D$ (SEG.)	T. GULOSO (SEG.)		TEMPO TOTAL (SEG.)	NÚMERO MÉDIO PT	NÚMERO MÉDIO ITERAÇ. PT
	SEM EXT.	COM EXT.		$\bar{\Omega}_i$	$\Omega_i^D$			
I.1	0.562	0.696	0.140	45.491	42.065	43.445	7	128
I.2	0.548	0.581	0.122	34.537	37.232	37.016	8	126
I.3	0.554	0.653	0.102	34.042	30.694	32.022	6	107
I.4	0.518	0.541	0.125	26.043	26.270	26.644	5	78
III.1	0.542	0.692	0.134	36.488	38.990	38.880	7	125
III.2	0.638	0.683	0.143	34.375	35.802	35.982	5	109
III.3	0.626	0.659	0.124	30.754	31.589	31.897	5	103
III.4	0.555	0.618	0.112	25.712	27.305	27.382	5	84
IV.1	0.337	0.364	0.095	34.482	34.095	34.482	6.5	120.5
IV.2	0.279	0.328	0.078	36.296	33.102	34.154	9.25	132.75
IV.3	0.344	0.323	0.089	40.192	37.762	38.639	11.25	158
IV.4	0.311	0.318	0.083	41.034	39.563	40.188	11.5	164.25
V.1	0.284	0.343	0.077	24.1	26.141	25.892	5	97
V.2	0.332	0.336	0.074	23.803	23.824	24.088	5	91
V.3	0.318	0.304	0.069	29.422	30.403	30.406	8	152
V.4	0.293	0.290	0.069	25.537	25.883	26.032	6	134
VI.1	0.614	0.692	0.167	70.749	80.630	77.701	10.75	211.25
VI.2	0.523	0.597	0.132	76.915	76.071	76.744	14	220
VI.3	0.605	0.560	0.136	67.373	72.059	71.353	12.5	209.75
VI.4	0.564	0.562	0.147	66.023	68.379	68.248	12.6	200.5

PT - Problema de Transporte.

TABELA (III.3) - VALORES MÉDIOS PARA O GULOSO COM PODAS APROXIMADAS.

PROBLEMA		GULOSO PURO		PODA APROXIMADAS OU EXATA + GULOSO				TOTAL FACILIDADES ABERTAS
		$\bar{\Omega}_i$	$\Omega_i^D$	$\bar{\Delta}_i$		$\Delta_i^D$	$\Delta_i$	
GRUPO	CUSTO FIXO			SEM EXT.	COM EXT.			
I.1	7.500	14	14	10	10	10	10	14 (25)
I.2	12.500	11	11	6	6	6	6	11 (25)
I.3	17.500	7	7	4	4	4	4	7 (25)
I.4	25.000	4	4	2	2	2	2	4 (25)
III.1	7.500	12	12	8	8	8	8	12 (26)
III.2	12.500	9	9	7	7	7	7	9 (26)
III.3	17.500	7	7	5	5	5	5	7 (26)
III.4	25.000	5	5	3	3	3	3	5 (26)
IV.1	7.500	13	13	9	9	11	11	13 (16)
IV.2	12.500	12	12	4	4	11	11	12 (16)
IV.3	17.500	12	12	2	2	9	11	12 (16)
IV.4	25.000	12	12	2	2	8	11	12 (16)
V.1	7.500	11	11	9	9	9	9	11 (16)
V.2	12.500	9	9	7	7	7	7	9 (16)
V.3	17.500	8	8	3	3	3	4 *	7/8* (16)
V.4	25.000	6	6	3	3	3	3	6 (16)
VI.1	7.500	17	17	9	9	11 *	11 *	17/18* (25)
VI.2	12.500	16	16	4	4	8	9	16 (25)
VI.3	17.500	14	14	4	4	6	7	14 (25)
VI.4	25.000	12	12	2	2	3	4	12 (25)

(.) Número de facilidades candidatas a localização.

\* Valores correspondentes.

TABELA (III.4) - NÚMERO MÉDIO DE FACILIDADES ABERTAS.

Para comparar o guloso puro e com poda montamos a tabela (III.5) com os tempos totais médios para resolver cada problema. É fácil verificar a redução significativa no tempo computacional conseguida com o acréscimo dos testes de redução para abrir facilidades. A diferença média  $(t^{SP} - t^P) / t^P$ , onde  $t^{SP}$  é o tempo médio para o guloso sem poda e  $t^P$  para o guloso com poda aproximada ( $\bar{\Delta}_i$  ou  $\Delta_i^D$ ), é da ordem de 14%.

Também o número de facilidades abertas para os problemas (V.3) e (VI.1), tabela (III.4), é maior para o guloso com poda o que acarretou um valor inferior para a função objetivo.

	I	III	IV	V	VI	MÉDIA
GULOSO PURO	40.381	37.954	40.578	35.315	73.661	45.577
PODA APROXIMADA + GULOSO	34.781	33.535	36.865	26.604	73.511	41.109
$\frac{t^{SP} - t^P}{t^P}$ (%)	16.1	13.177	10.071	32.743	0.204	14.459

TABELA (III.5) - TEMPOS TOTAIS MÉDIOS E COMPARAÇÕES.

Os testes de redução  $\bar{\Omega}_i$  e  $\Omega_i^D$ , para os algoritmos guloso puro ou com poda, apresentaram comportamentos semelhantes, quer em valores, quer em tempos computacionais.

Na tabela (III.6) lançamos os tempos computacionais médios consumidos pelas podas  $\Delta_i$ ,  $\bar{\Delta}_i$  e  $\Delta_i^D$  para cada problema.

A superioridade de  $\Delta_i^D$  é bastante evidente bem como o alto tempo computacional consumido pela poda exata  $\Delta_i$ . Com relação aos custos fixos, essas podas tem um comportamento padrão, quanto maior o custo menor o número de facilidades.

Em termos do número de facilidades abertas, essas podas se equivalem para os problemas I e III que são não capacitados. Os problemas IV e V são semelhantes a menos das ofertas que são maiores em V. Nesse caso, também os valores tendem a se equilibrar em V, enquanto distanciam em IV. O mesmo pode ser observado para os problemas I e VI. Esses resultados vem confirmar as conclusões apresentadas no capítulo II.

	I	III	IV	V	VI	MÉDIA
$\Delta_i$	44.716	43.236	60.343	24.629	58.723	46.329
$\bar{\Delta}_i$ SEM EXTENSÃO	0.545	0.590	0.317	0.306	0.576	0.467
$\bar{\Delta}_i$ COM EXTENSÃO	0.617	0.663	0.333	0.318	0.602	0.506
$\Delta_i^D$	0.122	0.128	0.086	0.072	0.145	0.110

TABELA (III.6) - TEMPOS COMPUTACIONAIS MÉDIOS PARA AS PODAS.

Veja na tabela (III.7) o número médio de facilidades abertas em cada problema pelas três podas. Também nesse caso, a superioridade de  $\Delta_i^D$  sobre  $\bar{\Delta}_i$  é evidente. E os resultados

apresentados por  $\Delta_i^D$  aproximam bastante a  $\Delta_i$ .

	IV	V	VI	MÉDIA
$\Delta_i$	11	5.75	7.75	8.16
$\bar{\Delta}_i$ SEM EXTENSÃO	4.25	5.5	4.75	4.83
$\bar{\Delta}_i$ COM EXTENSÃO	4.25	5.5	4.75	4.83
$\Delta_i^D$	9.75	5.5	7.0	7.42

TABELA (III.7) - NÚMERO MÉDIO DE FACILIDADES ABERTAS.

A introdução dos testes complementares, ou extensão, em nada acrescentou ao algoritmo. Nenhuma melhora foi observada. Se os problemas são não capacitados, I e III, o tempo computacional tende a aumentar com o acréscimo da extensão. Isto porque, alterando os custos de transporte na extensão, aumentam as iterações, sem alterar a solução final. Para os modelos capacitados existe uma leve tendência a reduzir os tempos computacionais ao anexar a extensão ao guloso com poda. (Veja coluna de  $\bar{\Delta}_i$  na tabela (III.4) e as tabelas (III.6) e (III.7).

Um outro teste de redução foi implementado com o algoritmo guloso. Ao iniciar, Passo 0, verificamos se algum  $i \in K_2$  atende a condição:

$$a_i > \sum_{i \in I} a_i - \sum_{j \in J} b_j$$

Caso afirmativo, necessariamente a facilidade  $i$  deve ser aberta, retirada de  $K_2$  e lançada em  $K_1$ . Para o caso específico dos problemas testes utilizados, nenhuma facilidade foi aberta por esta condição. Apesar de não implementada, uma verificação semelhante pode ser feita durante o processo iterativo:

$$a_i > \sum_{i \in (K_1 \cup K_2)} a_i - \sum_{j \in J} b_j, \quad i \in K_2$$

Se verdadeira o elemento  $i$  deve ser aberto.

Finalmente a tabela (III.8) compara os valores para função objetivo obtidos por esse e por outros algoritmos e o valor ótimo. Apresentamos em uma coluna a melhor solução obtida pelo guloso puro e em outra o obtido pelo guloso com poda. Em apenas dois problemas as soluções diferem em valor, com destaque para o guloso com poda.

Comparando o guloso com poda ( $Z^G$ ) aos valores ótimos ( $Z^*$ ), o erro médio  $(Z^G - Z^*)/Z^*$  é de 0.211%. Apesar de muito pequeno, outros algoritmos também apresentaram um comportamento semelhante para os mesmos problemas. Porém, o grande destaque para o algoritmo guloso está na sua simplicidade com um tempo computacional bastante reduzido. Na tabela (III.8) os dois primeiros algoritmos são para problemas não capacitados. O algoritmo de Sã é também uma heurística gulosa que após atingir um ótimo local executa um processo de tentativa de substituir as facilidades selecionadas por outras que estão fora da solução. Este processo eleva substancialmente o tempo computacional associado a essa heurística.

GRUPO	SOLUÇÃO ÓTIMA	KUEHN HAMBURGER (*)	KHUMAWALA (*)	SĀ	GULOSO PURO	PODA + GULOSO	$\frac{Z^G - Z^*}{Z^*}$ (%)
I.1	796.4	800.1	796.6	796.4	797.5	797.5	0.138
I.2	854.7	854.6	854.7	854.4	854.7	854.7	0.0
I.3	893.8	896.8	893.8	896.8	894.0	894.0	0.022
I.4	928.9	928.9	928.9	928.9	934.6	934.6	0.613
III.1	614.5	615.5	614.5	615.4	615.4	615.4	0.146
III.2	659.0	660.2	659.9	660.2	660.0	660.0	0.151
III.3	690.7	690.9	690.7	691.0	691.0	691.0	0.043
III.4	724.9	725.1	724.9	725.2	724.9	724.9	0.0
IV.1	1040.0			1040.5	1040.4	1040.4	0.038
IV.2	1097.6			1097.6	1098.0	1098.0	0.036
IV.3	1152.6			1152.6	1153.0	1153.0	0.034
IV.4	1235.1			1235.1	1235.5	1235.5	0.032
V.1	932.4			932.4	932.6	932.6	0.021
V.2	977.6			977.6	977.8	977.8	0.020
V.3	1013.9			1013.9	1014.1	1014.0	0.020
V.4					1060.7	1060.7	
VI.1					850.4	847.3	
VI.2	910.6			910.6	929.4	929.4	2.064
VI.3					998.5	998.5	
VI.4					1097.2	1097.2	

(\*) Heurísticas para problemas não capacitados.

TABELA (III.8) - VALORES PARA FUNÇÃO OBJETIVO.

CAPÍTULO IVUM ALGORITMO GULOSO PARA FUNÇÃO CUSTO FIXO/CÔNCAVO

Seja o problema proposto:

$$\min \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in I} f_i \left( \sum_j x_{ij} \right) + \sum_{i \in I} F_i y_i$$

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

(P1)

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

$$x_{ij} \geq 0, \quad y_i \in \{0, 1\}, \quad \forall_i \text{ e } \forall_j$$

E seja  $f_i(\cdot)$  uma função côncava. O caso específico em que  $f_i(\cdot)$  é linear foi estudado no capítulo anterior (Figura VI.1(a)). Analisaremos o problema (P1) em que para cada  $i \in I$  temos uma função custo fixo,  $F_i$ , mais uma função custo variável,  $f_i(\cdot)$ , estritamente côncava. Veja figura (IV.1(b)). Em geral essas funções são da forma  $f_i \left( \sum_{j \in J} x_{ij} \right) = K_i \left( \sum_{j \in J} x_{ij} \right)^q$ ,  $0 < q \leq 1$ , onde  $K_i$  é uma constante.

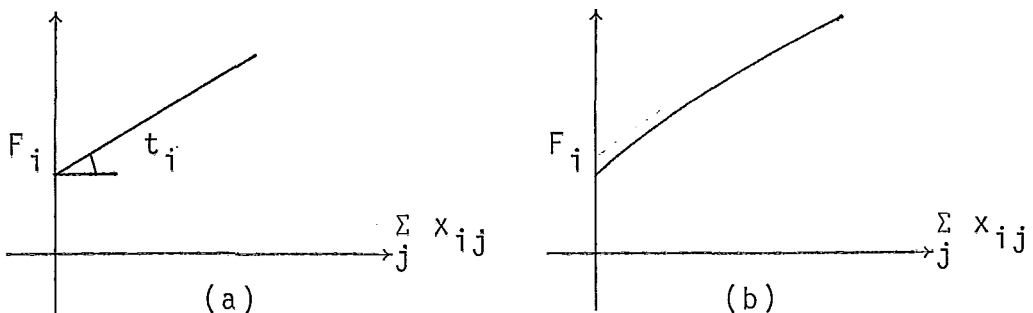


FIGURA (IV.1) - FUNÇÕES CUSTO FIXO/CÔNCAVO.



A concavidade das funções dificulta ainda mais a resolução de (P1). Heurísticas e algoritmos exatos para esse problema foram introduzidos no capítulo I. Utilizando as definições, conceitos e idéias dos capítulos anteriores, apresentaremos uma heurística gulosa para funções côncavas, também composta por duas etapas. Inicialmente aplicamos os testes de redução para abrir uma facilidade, capítulo II, extendidos para funções côncavas e em uma segunda etapa, um processo iterativo que procura abrir a facilidade que proporcione o maior decréscimo da função a cada iteração.

As extensões dos testes de redução para funções côncavas são elaboradas por aproximações lineares de  $f_i(.)$  e sendo estudado dois tipos de linearização: por arco e por nó. Também estudaremos o acréscimo de uma terceira etapa, ou processo iterativo final, com o objetivo de melhorar a solução encontrada.

Inicialmente introduziremos os testes de redução para funções côncavas, seguindo o processo iterativo final e o algoritmo guloso. Posteriormente faremos uma análise dos resultados.

## 1. ALGORITMO GULOSO

Seja o conjunto  $I$  de facilidades dividido entre os subconjuntos  $K_0$ ,  $K_1$  e  $K_2$ , definidos no capítulo II. Inicialmente  $K_0 = K_1 = \emptyset$  e  $K_2 = I = \{1, \dots, m\}$ .

Com aproximações lineares das funções côncavas, faremos uma extensão as definições do capítulo II, introduzindo  $\Delta_i^a$  (ou  $\bar{\Delta}_i^a$  ou  $\Delta_i^{aD}$ ) e  $\Omega_i^a$  (ou  $\bar{\Omega}_i^a$  ou  $\Omega_i^{aD}$ ), respectivamente para os testes de redução para abrir e fechar uma facilidade. Com isso, de forma semelhante ao caso linear, podemos propor uma heurística gulosa para o modelo (P1) composta por três etapas:

- (1) Aplicando um dos testes de redução para abrir facilidades, transferir os elementos abertos de  $K_2$  para  $K_1$ .
- (2) Executar um processo iterativo, transferindo para  $K_1$ , a cada iteração, o elemento  $i \in K_2$  que forneça o maior retorno:  $\Omega_i^a - F_i > 0$ , onde  $\Omega_i^a$  é dado pelo teste de redução para fechar uma facilidade.
- (3) Com o conjunto  $K_1$  de facilidades, obtido nas etapas (1) e (2), executar um processo iterativo final variando as aproximações lineares das funções côncavas.

Na primeira etapa calculamos  $\bar{\Delta}_i^a$  e  $\Delta_i^{aD}$ , para todo  $i \in K_2$ , como extensões naturais de  $\bar{\Delta}_i$  e  $\Delta_i^D$  para funções custo fixo/linear. Se  $\bar{\Delta}_i^a \geq F_i$  (ou  $\Delta_i^{aD} \geq F_i$ ) então  $i$  será retirado de  $K_2$  e lançado em  $K_1$ .

Na segunda etapa, também de forma semelhante, podemos caracterizar  $\bar{\Omega}_i^a$  ou  $\Omega_i^{aD}$ . Trabalharemos apenas com  $\Omega_i^{aD}$ . Seja  $\Omega_i^{aD} - F_i$  para todo  $i \in K_2$ . O algoritmo guloso seleciona a cada iteração o elemento  $i_m \in K_2$  que proporcione o maior decréscimo na função ou tal que  $\Omega_{i_m}^{aD} - F_{i_m} = \max_{i \in K_2} \Omega_i^{aD} - F_i > 0$ . A facilidade  $i_m$

será aberta, retirada de  $K_2$  e lançada em  $K_1$ . Voltamos a calcular novos valores para  $\Omega_i^{aD} - F_i$ ,  $\forall i \in K_2$ , selecionamos nova facilidade  $i_m$  e sucessivamente até que  $\Omega_i^{aD} - F_i \leq 0$ ,  $\forall i \in K_2$  ou  $K_2 = \emptyset$ .

Com o conjunto de facilidades abertas  $K_1$ , resultante das etapas anteriores, podemos, de forma iterativa, calcular novas aproximações das funções côncavas até que as diferenças entre as inclinações, ou custos, em cada arco sejam inferiores a uma precisão estabelecida.

Ressaltamos que:

- (a) Como no caso linear, o algoritmo pode ser implementado apenas com a segunda etapa, guloso padrão. Mas, o acréscimo da primeira e terceira etapas aumentam significativamente o seu desempenho em alguns casos.
- (b) Na segunda etapa, trabalhamos com um conjunto  $K_1$  que não necessariamente atende a condição de viabilidade  $\sum_{i \in K_1} a_i \geq \sum_{j \in J} b_j$ . Este problema será contornado acrescentando os elementos artificiais  $i^a$  e  $j^a$  apresentados na seção (III.2). Ao fim da segunda etapa, se a condição de viabilidade ainda não foi atendida, a terceira etapa obviamente será evitada.
- (c) Nas três etapas trabalhamos com aproximações lineares das funções côncavas. Discutiremos os casos em que essas linearizações são obtidas utilizando os fluxos em cada arco, apro

ximação por arco, ou em cada centro de oferta, aproximação por nó.

### 1.1. Testes Para Abrir uma Facilidade

A grande limitação para utilização dos testes de redução, apresentados anteriormente, para problemas com funções custo côncavo é o não atendimento da propriedade de submodularidade, (II.7) e (II.8). Utilizando aproximações lineares dessas funções procuraremos caracterizar os testes para abrir facilidades.

Se  $f_i(.)$  é uma função linear então podemos escrever  $f_i(\sum_{j \in J} x_{ij}) = t_i \sum_{j \in J} x_{ij}$  onde  $t_i$ ,  $i \in I$ , é uma constante. Substituindo essa função e fazendo  $C_{ij} = c_{ij} + t_i$  no problema  $(P_1)$  resulta (II.6):

$$Z(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in S} F_i$$

que pode ser decomposta em:

$$W(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} \quad \text{e} \quad F(S) = \sum_{i \in S} F_i$$

funções submodulares.

De (II.1) temos que:  $\Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2)$ ,  $i \in K_2$ . E

pelo teorema (1.1) se  $\Delta_i \geq F_i$  então a facilidade  $i$  pode ser aberta. Para obtenção desse resultado utilizamos a propriedade de submodularidade da função  $W(\cdot)$  para concluir que  $\Delta_i$  torna cada vez maior, ou seja, é não decrescente uma vez que o conjunto  $(K_1 \cup K_2)$  é não crescente em número de elementos.

Seja  $f_i(\cdot)$  uma função côncava e

$$\Delta_i^C = W^C(K_1 \cup K_2 - i) - W^C(K_1 \cup K_2), \quad i \in K_2$$

onde  $W^C(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in S} f_i \left( \sum_{j \in J} x_{ij} \right)$ .

Se  $\Delta_i^C \geq F_i$  não podemos concluir em abrir a facilidade  $i$ . Isto porque a função côncava  $W^C(i)$  não satisfaz a propriedade de submodularidade e portanto  $\Delta_i^C$  não é necessariamente não decrescente com a redução de  $(K_1 \cup K_2)$ . Nosso objetivo é caracterizar

$$\Delta_i^a = W^a(K_1 \cup K_2 - i) - W^a(K_1 \cup K_2), \quad i \in K_2$$

tal que  $\Delta_i^a \leq \Delta_i^C$ ,  $\forall i \in K_2$ , e não decrescente. Logo, se  $\Delta_i^a \geq F_i$  então a facilidade  $i$  pode ser aberta para o problema côncavo.

Se determinarmos  $W^a(K_1 \cup K_2 - i) \leq W^C(K_1 \cup K_2 - i)$

$$W^a(K_1 \cup K_2) \geq W^C(K_1 \cup K_2)$$

então teremos  $\Delta_i^a \leq \Delta_i^C$  e restando apenas mostrar que  $\Delta_i^a$  seja não decrescente. Procuraremos a seguir caracterizar a função  $W^a(\cdot)$ .

Seja  $t_i = f_i(a_i)/a_i$  uma aproximação linear da função  $f_i(\cdot)$ ,  $i \in I$ . De forma semelhante, podemos fazer  $C_{ij} = c_{ij} + t_i$

obtendo a função  $W(S) = \min_{x \in X} \sum_{i \in S} \sum_{j \in J} C_{ij} x_{ij}$  que corresponde ao valor ótimo para o problema de transporte com o conjunto  $S$  de centros de oferta,  $T(S)$ , e, devido as linearizações das funções, é também um limite inferior para  $W^C(S)$ ,  $S \subseteq I$ . Logo, podemos fazer

$$W^a(K_1 \cup K_2 - i) = W(K_1 \cup K_2 - i) \quad (IV.1)$$

Portanto,  $W^a(K_1 \cup K_2 - i)$  é o valor ótimo para o problema de transporte com o conjunto  $(K_1 \cup K_2 - i)$  de centros de oferta,  $T(K_1 \cup K_2 - i)$ , e custos  $C_{ij} = c_{ij} + t_i$  para todo  $(i, j)$ ,  $i \in (K_1 \cup K_2 - i)$ ,  $j \in J$ , resultantes da linearização das funções.

Por sua vez, resolvendo o problema de transporte  $T(K_1 \cup K_2)$  com os custos aproximados  $C_{ij} = c_{ij} + t_i$ ,  $i \in (K_1 \cup K_2)$ ,  $j \in J$ , definidos acima, resulta em uma solução primal viável. Levando essa solução as funções côncavas caracterizamos um limite superior para  $W^C(K_1 \cup K_2)$ . Logo, podemos fazer

$$\begin{aligned} W^a(K_1 \cup K_2) &= W(K_1 \cup K_2) + \\ &+ \sum_{i \in (K_1 \cup K_2)} [f_i \left( \sum_{j \in J} x_{ij} \right) - \\ &- t_i \sum_{j \in J} x_{ij}] \end{aligned} \quad (IV.2)$$

onde  $x$  é uma solução primal viável para  $T(K_1 \cup K_2)$  e  $W(K_1 \cup K_2)$  o correspondente valor para função linearizada.

Subtraindo (IV.2) de (IV.1) resulta:

$$\Delta_i^a = \Delta_i - \sum_{i \in (K_1 \cup K_2)} [f_i \left( \sum_{j \in J} x_{ij} \right) - t_i \sum_{j \in J} x_{ij}] \quad (\text{IV.3})$$

onde  $\Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2)$

Por definição,  $\Delta_i$  é não decrescente (capítulo II). Para que  $\Delta_i^a$  também seja não decrescente, é necessário que a segunda parcela, a direita em (IV.3), seja não crescente. Esta condição pode ser garantida no processo iterativo, mantendo o menor valor para a parcela  $\sum_{i \in (K_1 \cup K_2)} [f_i \left( \sum_{j \in J} x_{ij} \right) - t_i \sum_{j \in J} x_{ij}]$ , ou tomando como referência a solução primal que forneça o menor limite superior para  $W^C(K_1 \cup K_2)$  até a iteração atual. Assim,  $\Delta_i^a$  é um limite inferior para  $\Delta_i^C$  e não decrescente e se  $\Delta_i^a \geq F_i$  então  $y_i = 1$ .

Inicialmente, para calcular  $\Delta_i^a$  resolvemos o problema de transporte  $T(K_1 \cup K_2)$ , com custos linearizados, obtendo um valor para a segunda parcela. Calculamos  $\Delta_i$ , conforme desenvolvimento anterior, resolvendo um novo problema de transporte para cada  $i \in K_2$ , e temos  $\Delta_i^a$  para todo  $i \in K_2$ .

Pelas mesmas razões apresentadas anteriormente, podemos substituir  $\Delta_i$  por  $\bar{\Delta}_i$  ou  $\Delta_i^D$  obtendo limites ainda menores para  $\Delta_i^a$ , porém, de cálculo rápido. Resulta:

$$\bar{\Delta}_i^a = \bar{\Delta}_i - \sum_{i \in (K_1 \cup K_2)} [f_i \left( \sum_{j \in J} x_{ij} \right) - t_i \sum_{j \in J} x_{ij}] , \text{ ou} \quad (\text{IV.4})$$

$$\Delta_i^{aD} = \Delta_i^D - \sum_{i \in (K_1 \cup K_2)} [f_i \left( \sum_{j \in J} x_{ij} \right) - t_i \sum_{j \in J} x_{ij}] \quad (IV.5)$$

Em ambos os casos, apenas um problema de transporte  $\bar{e}$  resolvido,  $T(K_1 \cup K_2)$ .

Estes testes serão utilizados apenas na primeira etapa do algoritmo e não iterativamente. Portanto, a solução primal que determina o limite superior para  $W^C(K_1 \cup K_2)$  em (IV.2), e conseqüentemente utilizada em (IV.4) e (IV.5),  $\bar{e}$  a solução ótima do problema de transporte  $T(K_1 \cup K_2)$ .

Os resultados acima foram obtidos a partir da aproximação linear de  $f_i(\cdot)$  dada por  $t_i = f(a_i)/a_i$ ,  $\forall i \in I$ , o que permitiu identificar um limite inferior para  $W^C(K_1 \cup K_2 - i)$  dado por  $W(K_1 \cup K_2 - i)$ . Todavia, esses testes de redução possuem um poder de poda muito baixo, não cumprindo o objetivo de abrir um número elevado de facilidades na etapa inicial. Isto porque, a aproximação linear utilizada,  $t_i$ , distancia demasiadamente da função real, aumentando a segunda parcela a direita em (IV.4) e (IV.5) e conseqüentemente reduzindo  $\bar{\Delta}_i^a$  e  $\Delta_i^{aD}$ . O poder de poda torna-se significativamente maior, ou eleva-se  $\bar{\Delta}_i^a$  e  $\Delta_i^{aD}$ , se a aproximação linear for feita por arco e não mais por centro de oferta, ou  $\bar{n}$ , como acima. Portanto, teremos para cada arco  $(i, j)$  uma aproximação de  $f_i(\cdot)$  dada por:

$$t_{ij} = f_i(d_{ij})/d_{ij},$$

$$d_{ij} = \min \{a_i, b_j\}, \forall (i, j), i \in (K_1 \cup K_2), j \in J$$



Neste caso,  $W(K_1 \cup K_2)$  corresponderá ao valor ótimo da função objetivo do problema de transporte  $T(K_1 \cup K_2)$  com os custos de transporte  $C_{ij} = c_{ij} + t_{ij}$ ,  $\forall i \in (K_1 \cup K_2)$  e  $\forall j \in J$ . Perdemos a otimalidade teórica dos testes de redução, uma vez que  $W(K_1 \cup K_2 - i)$  deixa de ser um limite inferior para  $W^C(K_1 \cup K_2 - i)$ , porém com essa heurística ganhamos em poder de poda, e em termos de práticos, as facilidades abertas são, em geral, as ideais. Essas duas aproximações serão comparadas posteriormente na análise de resultados.

Quanto aos aspectos de implementação desses testes, seguem as mesmas rotinas adotadas para o caso linear.

## 1.2. Segunda Etapa

A segunda etapa é composta pelo processo iterativo do guloso. A cada iteração seleciona um elemento  $i_m \in K_2$  que proporcione o maior retorno, retirando  $i_m$  de  $K_2$  e lançando em  $K_1$ . O processo prossegue até que  $\Omega_i^C - F_i \leq 0$ ,  $\forall i \in K_2$  ou  $K_2 = \emptyset$ . Portanto, é semelhante ao algoritmo para funções lineares, sendo necessário apenas caracterizar  $\Omega_i^C = W^C(K_1) - W^C(K_1 \cup i)$ .

Para o problema linear utilizamos os testes de redução para fechar facilidades ( $\Omega_i$ ,  $\bar{\Omega}_i$  ou  $\Omega_i^D$ ) com o objetivo de abrir. Assim, o elemento  $i_m$  com a menor chance de ser fechado na iteração, ou com o maior valor para  $\Omega_i - F_i$ , seria aberto, retirado de  $K_2$  e lançado em  $K_1$ . Aqui a idéia é a mesma. Apro

ximando linearmente as funções  $f_i(\cdot)$ ,  $\forall i \in I$ , definimos  $\Omega_i^a$ ,  $\bar{\Omega}_i^a$  e  $\Omega_i^{aD}$  em substituição a  $\Omega_i^C$ ,  $\bar{\Omega}_i^C$  e  $\Omega_i^{CD}$  e calculados de forma semelhante a  $\Omega_i$ ,  $\bar{\Omega}_i$  e  $\Omega_i^D$ . Assim temos:

$$\Omega_i^a = W^a(K_1) - W^a(K_1 \cup i), \forall i \in K_2$$

onde  $W^a(K_1)$  é o valor ótimo para o problema de transporte  $T(K_1)$  com os custos de transporte definidos conforme linearizações propostas para a primeira etapa:

$$(i) \quad t_{ij} = f_i(a_i)/a_i \Rightarrow C_{ij} = c_{ij} + t_{ij}, \forall i \in I, j \in J \quad (IV.6)$$

$$(ii) \quad t_{ij} = f_i(d_{ij})/d_{ij} \Rightarrow C_{ij} = c_{ij} + t_{ij}, \forall i \in I, j \in J \quad (IV.7)$$

$$d_{ij} = \min \{a_i, b_j\}$$

Testamos apenas  $\Omega_i^{aD}$  calculado tal qual  $\Omega_i^D$ . Ainda mais, as aproximações lineares adotadas na etapa inicial, seguem para a segunda. Apesar de não testado, poderíamos também definir  $\Omega_i^a$  ou  $\bar{\Omega}_i^a$ .

Quer para o caso linear quer para o côncavo, os critérios de escolha de facilidades a serem abertas nessa etapa não garantem otimalidade, mas justificam uma heurística eficiente como a aqui adotada.

### 1.3. Processo Iterativo Final (PIF)

Definido um conjunto  $K_1$  de facilidades abertas nas duas primeiras etapas, o PIF tem o objetivo de procurar uma solução melhor variando as aproximações lineares adotadas anteriormente e utilizando os fluxos ótimos para o problema de transporte  $T(K_1)$ . Essa procura poderá levar ao fechamento de alguma facilidade até então aberta.

Também analisamos duas alternativas nessa etapa:

(i) Aproximação de  $f_i(\cdot)$  por  $n\bar{o}$ :

$$t_i = \left\{ \begin{array}{l} f_i(\sum_{j \in J} x_{ij}) / \sum_{j \in J} x_{ij}, \text{ se } \sum_{j \in J} x_{ij} \neq 0 \\ \infty, \text{ se } \sum_{j \in J} x_{ij} = 0 \end{array} \right\} \Rightarrow C_{ij} = c_{ij} + t_i, \quad \forall i \in K_1, j \in J \quad (\text{IV.8})$$

(ii) Aproximação de  $f_i(\cdot)$  por arco:

$$t_{ij} = \left\{ \begin{array}{l} f_i(x_{ij}) / x_{ij}, \text{ se } x_{ij} \neq 0 \\ \infty, \text{ se } x_{ij} = 0 \end{array} \right\} \Rightarrow C_{ij} = c_{ij} + t_{ij}, \quad \forall i \in K_1, j \in J \quad (\text{IV.9})$$

onde  $x$  é o fluxo ótimo para o problema de transporte  $T(K_1)$ .

Com as linearizações (i) ou (ii), resolvemos a cada iteração o problema de transporte  $T(K_1)$  com os custos calculados sobre os fluxos ótimos do problema anterior. O processo

prossegue até que os custos  $C_{ij}$ ,  $\forall i \in K_1, j \in J$ , não variem. Mais especificamente, chamando de  $C'_{ij}$ ,  $\forall i \in K_1, j \in J$ , os novos custos e  $C_{ij}$ ,  $\forall i \in K_1, j \in J$ , os custos para o problema anterior, verificamos se o erro:

$$\varepsilon = \frac{C'_{ij} - C_{ij}}{C'_{ij}}$$

atende a uma precisão estabelecida,  $|\varepsilon| \leq 0.01$ , para todo arco  $(i, j)$ ,  $i \in K_1, j \in J$ .

Essa etapa, com linearizações por  $\bar{n}_0$ , (IV.8), proporcionou saltos significativos nos valores das funções, conforme mostraremos com os testes computacionais.

Apresentada as três etapas, passamos ao algoritmo geral e posteriormente a análise de resultados.

#### 1.4. Algoritmo

Passo 0. Seja  $K_1 = \emptyset$  e  $K_2 = I = \{1, 2, \dots, m\}$

Passo 1. Calcule as aproximações lineares a  $f_i(\cdot)$  para todo  $i \in I$  e os custos:

$$(i) \quad t_i = f(a_i)/a_i, \quad \forall i \in I \Rightarrow C_{ij} = c_{ij} + t_i, \quad \forall i \in I, j \in J, \text{ ou}$$

$$(ii) \quad t_{ij} = f(d_{ij})/d_{ij}, \quad \forall i \in I, j \in J \Rightarrow C_{ij} = c_{ij} + t_{ij}, \quad \forall i \in I, j \in J$$

$$d_{ij} = \min \{a_i, b_j\}$$

Passo 2. Execute um dos testes para abrir facilidades ( $\bar{\Delta}_i^a$  ou  $\Delta_i^{aD}$ ) para todo  $i \in K_2$ .

Se  $\bar{\Delta}_i^a \geq F_i$  (ou  $\Delta_i^{aD} \geq F_i$ ),  $i \in K_2$ , então faça  $K_1 = K_1 \cup \{i\}$

$$K_2 = K_2 - \{i\}$$

Passo 3. Calcule  $W(K_1)$  com os custos  $C_{ij}$ ,  $\forall i \in K_1, j \in J$ .

Passo 4. Se  $K_2 = \emptyset$  então vá para o Passo 11.

Passo 5. Para todo  $i \in K_2$  calcule  $\Omega_i^{aD} - F_i$ .

Passo 6. Determine  $i_m \in K_2$  tal que  $\Omega_{i_m}^{aD} - F_{i_m} = \max_{i \in K_2} (\Omega_i^{aD} - F_i)$ .

Passo 7. Se  $(\Omega_{i_m}^{aD} - F_{i_m}) \leq 0$  então vá para Passo 11.

Passo 8. Calcule  $W(K_1 \cup i_m)$  e  $\Omega_{i_m} - F_{i_m} = W(K_1) - W(K_1 \cup i_m) - F_{i_m}$ .

Passo 9. Se  $(\Omega_{i_m} - F_{i_m}) \leq 0$  então vá para Passo 11.

Passo 10. Faça  $W(K_1) = W(K_1 \cup i_m)$ ,  $K_1 = K_1 \cup i_m$ ,  $K_2 = K_2 - i_m$ .

Retorne ao Passo 4.

Passo 11. Se "não for executar PIF" então PARE.  $K_1$  é o conjunto de facilidades abertas.

Passo 12. Calcule as aproximações lineares a  $f_i(\cdot)$  para todo  $i \in K_1$  e os custos:

(i)  $C'_{ij} = c_{ij} + t_i$ ,  $\forall i \in K_1$ ,  $j \in J$  onde

$$t_i = \begin{cases} f_i(x_{ij})/x_{ij}, & \text{se } x_{ij} \neq 0 \\ \infty & , \text{ se } x_{ij} = 0 \end{cases}$$

ou

(ii)  $C'_{ij} = c_{ij} + t_{ij}$ ,  $\forall i \in K_1$ ,  $j \in J$  onde

$$t_{ij} = \begin{cases} f_i(\sum_{j \in J} x_{ij}) / \sum_{j \in J} x_{ij}, & \text{se } \sum_{j \in J} x_{ij} \neq 0 \\ \infty & , \text{ se } \sum_{j \in J} x_{ij} = 0 \end{cases}$$

Passo 13. Calcule  $\epsilon_{ij} = (C'_{ij} - C_{ij})/C'_{ij}$ ,  $\forall i \in K_1$ ,  $j \in J$ .

Se  $|\epsilon_{ij}| \leq 0.01$ ,  $\forall i \in K_1$ ,  $j \in J$  então PARE.  $K_1$  é o conjunto de facilidades abertas.

senão faça  $C_{ij} = C'_{ij}$ ,  
 $\forall i \in K_1$ ,  $j \in J$ .

Passo 14. Calcule  $W(K_1)$  com os novos custos  $C_{ij}$ ,  $\forall i \in K_1$ ,  $j \in J$ .

Passo 15. Atualize  $Z_r(K_1)$  caso a nova solução seja melhor.

Passo 16. Se "excedem nº iterações PIF" então PARE.  $K_1$  é o conjunto de facilidades abertas;

senão volte ao Passo 12.

Está implícito no algoritmo a não consideração em  $K_1$  da facilidade  $i$  tal que  $\sum_{j \in J} x_{ij} = 0$ .

Associado ao conjunto final  $K_1$ , de facilidades abertas, a função:

$$Z_r(K_1) = \sum_{i \in K_1} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in K_1} f_i \left( \sum_{j \in J} x_{ij} \right) + \sum_{i \in K_1} F_i$$

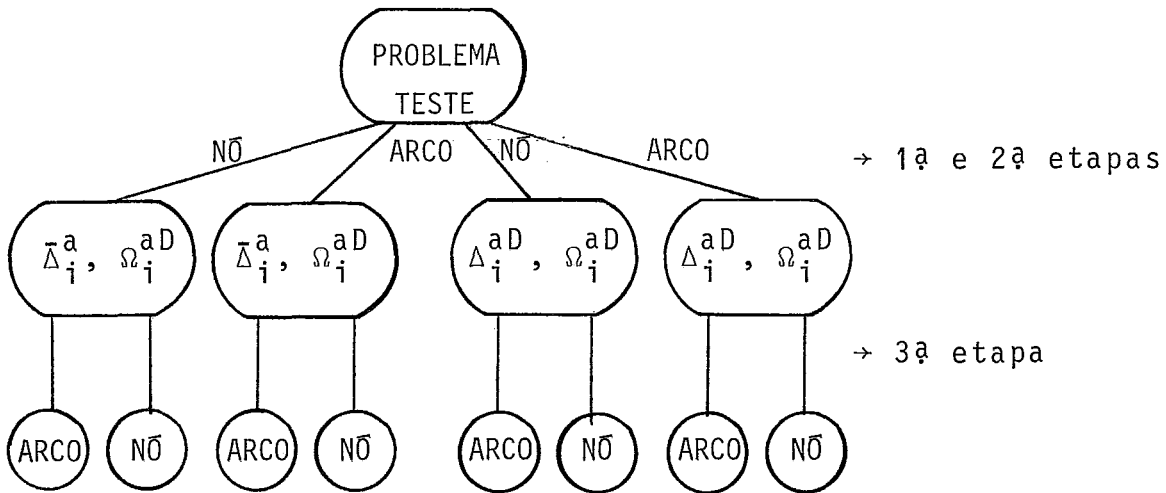
onde  $x_{ij}$  é o fluxo ótimo do problema de transporte  $T(K_1)$  com os custos  $C_{ij} = c_{ij} + t_i$  ou  $C_{ij} = c_{ij} + t_{ij}$ ,  $\forall i \in K_1$ ,  $j \in J$ .

## 2. ANÁLISE DE RESULTADOS

Utilizamos para teste do algoritmo proposto os problemas I, IV, V e VI, Apêndice D. Em cada um, todas as facilidades apresentam os mesmos custos fixos os quais podem assumir os valores \$ 7500, \$ 12500 e \$ 25000, implicando respectivamente nos problemas I.1, I.2 e I.4, IV.1, IV.2 e IV.4, etc. Não consideramos, nesses testes, os custos fixos iguais a \$ 17500, utilizados nos testes anteriores. Ainda mais, as funções côncavas  $f_i(\cdot)$ ,  $\forall i \in I$ , são da forma  $f_i(\cdot) = K_i \sqrt{\sum_{j \in J} x_{ij}}$ . Em um dado problema todas as facilidades também receberam o mesmo valor para a constante  $K_i$ . Porém, esse valor pode ser: 200, 400 ou 800. Com isso, temos um conjunto de 36 problemas testes.

Mais uma vez, visando cobrir todo o trabalho teórico

e as possibilidades apresentadas, foi testado para cada um dos problemas um conjunto básico de 8 combinações. Inicialmente utilizar  $\bar{\Delta}_i^a$  ou  $\Delta_i^{aD}$  e  $\Omega_i^{aD}$  na segunda etapa. Cada uma das opções acima pode ser empregada com aproximações lineares por n̄o ou por arco. Finalmente, uma última ramificação correspondente ao PIF utilizando aproximações por n̄o ou por arco.



As tabelas (IV.1), (IV.2) e (IV.3) apresentam os resultados gerais para os testes. A primeira apresenta os valores médios para a primeira e segunda etapas do algoritmo, aproximando linearmente as funções  $f_i(\cdot)$ ,  $\forall i \in I$ , por n̄o, (IV.6). Na primeira metade da tabela temos  $\bar{\Delta}_i^a$  para primeira etapa do guloso e  $\Omega_i^{aD}$  para segunda. A outra metade temos  $\Delta_i^{aD}$  e  $\Omega_i^{aD}$ . O mesmo ocorre para tabela (IV.2), para aproximações por arco, (IV.7). A tabela (IV.3) resume a terceira etapa ou o PIF, com as aproximações por arco, (IV.8) e por n̄o, (IV.9).



PROBLEMA		$\bar{\Delta}_i^a$ (1)		$\Omega_i^{aD}$ (2)		CUSTO	Nº	$\Delta_i^{aD}$ (1)		$\Omega_i^{aD}$ (2)		CUSTO	Nº
Nº	$K_i$	FAC. ABER.	TEMPO (1)	FAC. ABER.	TEMPO (1)+(2)	(\$ 1000)	P. T.	FAC. ABER.	TEMPO (1)	FAC. ABER.	TEMPO (1)+(2)	(\$ 1000)	P. T.
I.1	200	0	0.503	14	41.328	970.9	17	0	0.096	14	40.809	970.9	17
	400	0	0.532	14	42.724	1144.3	17	0	0.109	14	40.947	1144.3	17
	800	0	0.576	14	45.108	1491.1	17	0	0.104	14	40.035	1491.1	17
I.2	200	0	0.556	11	41.098	1006.4	14	0	0.128	11	41.623	1006.4	14
	400	0	0.515	11	41.058	1158.1	14	0	0.105	11	39.617	1158.1	14
	800	0	0.559	11	44.298	1461.4	14	0	0.117	11	42.998	1461.4	14
I.4	200	0	0.500	4	32.892	1022.2	7	0	0.121	4	34.647	1022.2	7
	400	0	0.545	4	32.499	1109.7	7	0	0.114	4	33.872	1109.7	7
	800	0	0.574	4	34.151	1284.8	7	0	0.126	4	34.470	1284.8	7
IV.1	200	1	0.296	12	40.658	1212.8	15	8	0.078	5	36.233	1212.8	8
	400	1	0.292	12	41.442	1385.2	15	7	0.089	6	40.573	1385.2	9
	800	1	0.338	12	43.910	1729.9	15	1	0.089	12	42.452	1729.9	15
IV.2	200	1	0.311	11	40.978	1265.1	14	8	0.089	4	37.593	1265.1	7
	400	1	0.295	11	40.695	1432.3	14	7	0.088	5	37.693	1432.3	8
	800	1	0.307	11	43.088	1766.5	14	1	0.083	11	40.734	1766.5	14
IV.4	200	1	0.288	11	40.415	1402.6	14	7	0.080	5	36.574	1402.6	8
	400	1	0.376	11	45.205	1569.8	14	4	0.091	8	42.425	1569.8	11
	800	0	0.318	12	44.679	1904.0	15	0	0.087	12	42.392	1904.0	15
V.1	200	1	0.325	10	36.363	1083.9	13	1	0.066	10	36.159	1083.9	13
	400	0	0.309	11	37.330	1235.3	14	0	0.066	11	36.287	1235.3	14
	800	0	0.304	11	38.119	1538.0	14	0	0.083	11	38.362	1538.0	14
V.2	200	1	0.327	8	35.229	1115.8	11	1	0.069	8	33.865	1.115.8	11
	400	0	0.307	9	34.946	1253.9	12	0	0.068	9	34.583	1.253.9	12
	800	0	0.284	9	33.857	1530.0	12	0	0.073	9	35.845	1.530.0	12
V.4	200	1	0.328	5	32.045	1175.3	8	1	0.080	5	33.101	1175.3	8
	400	0	0.324	6	33.656	1267.1	9	0	0.073	6	34.718	1267.1	9
	800	0	0.334	6	35.808	1480.9	9	0	0.073	6	33.272	1480.9	9
VI.1	200	0	0.537	17	79.844	1045.8	20	0	0.153	17	80.739	1045.8	20
	800	0	0.494	17	69.398	1631.7	20	0	0.116	17	72.091	1631.7	20
VI.4	200	0	0.545	12	68.058	1264.3	15	0	0.123	12	63.173	1264.3	15
	800	0	0.596	12	66.813	1765.6	15	0	0.132	12	64.518	1765.6	15
MÉDIA GERAL		0.27	0.417	10.73	44.840		13.73	1.27	0.097	9.73	43.716		12.73

PT - Problema de Transporte.

TABELA (IV.1) - VALORES MÉDIOS 1ª E 2ª ETAPAS GULOSO APLICADO A FUNÇÕES CÔNCAVAS - APROXIMAÇÃO POR NÚ

PROBLEMA		$\bar{\delta}_i^a$ (1)		$\Omega_i^{aD}$ (2)		CUSTO	Nº	$\Delta_i^{aD}$ (1)		$\Omega_i^{aD}$ (2)		CUSTO	Nº
Nº	$K_i$	FAC. ABER.	TEMPO (1)	FAC. ABER.	TEMPO (1)+(2)	(\$ 1000)	P. T.	FAC. ABER.	TEMPO (1)	FAC. ABER.	TEMPO (1)+(2)	(\$ 1000)	P. T.
I.1	200	10	0.519	4	38.569	970.9	7	10	0.113	4	40.879	970.9	7
	400	10	0.532	4	37.733	1144.3	7	10	0.113	4	39.276	1144.3	7
	800	10	0.504	4	39.108	1491.1	7	10	0.101	4	38.169	1491.1	7
I.2	200	6	0.501	5	34.482	1006.4	8	6	0.139	5	38.856	1006.4	8
	400	6	0.557	5	36.692	1158.1	8	6	0.110	5	34.553	1158.1	8
	800	6	0.547	5	39.923	1461.4	8	6	0.114	5	38.008	1461.4	8
I.4	200	2	0.563	2	27.645	1022.2	5	2	0.127	2	26.538	1022.2	5
	400	2	0.494	2	25.162	1109.7	5	2	0.102	2	26.250	1109.7	5
	800	2	0.592	2	29.227	1284.8	5	2	0.123	2	26.610	1284.8	5
IV.1	200	1	0.291	12	38.012	1212.8	15	8	0.085	5	32.321	1212.8	8
	400	1	0.318	12	38.770	1385.2	15	2	0.094	11	38.955	1385.2	14
	800	0	0.354	13	34.955	1729.9	16	0	0.081	13	33.706	1729.9	16
IV.2	200	1	0.296	11	38.101	1265.1	14	8	0.085	4	33.186	1265.1	7
	400	1	0.306	11	35.970	1432.3	14	2	0.083	10	34.152	1432.3	13
	800	0	0.325	12	34.057	1766.5	15	0	0.093	12	35.627	1766.5	15
IV.4	200	1	0.325	11	38.752	1402.6	14	5	0.103	7	38.175	1402.6	10
	400	1	0.322	11	37.284	1569.8	14	1	0.097	11	40.337	1569.8	14
	800	0	0.324	12	34.775	1904.0	15	0	0.082	12	33.250	1904.0	15
V.1	200	9	0.325	2	24.856	1083.9	5	9	0.064	2	22.573	1083.9	5
	400	9	0.336	2	30.069	1235.3	5	9	0.076	2	28.081	1235.3	5
	800	9	0.343	2	31.771	1538.0	5	9	0.066	2	30.631	1538.0	5
V.2	200	7	0.319	2	23.048	1115.8	5	7	0.068	2	21.987	1115.8	5
	400	7	0.295	2	24.540	1253.9	5	7	0.072	2	26.054	1253.9	5
	800	7	0.304	2	27.869	1530.0	5	7	0.067	2	28.053	1530.0	5
V.4	200	3	0.308	3	29.824	1175.3	6	3	0.075	3	28.912	1175.3	6
	400	3	0.318	2	30.884	1267.1	5	3	0.072	2	30.998	1267.1	5
	800	3	0.333	2	34.027	1480.9	5	3	0.073	2	34.976	1480.9	5
VI.1	200	1	0.563	15	80.109	1039.8	18	2	0.131	14	78.960	1039.8	17
	800	0	0.524	18	70.820	1643.8	21	0	0.156	18	77.990	1643.8	21
VI.4	200	1	0.512	13	72.221	1268.4	16	1	0.136	13	72.969	1268.4	16
	800	0	0.525	12	64.031	1745.1	15	0	0.137	12	69.645	1745.1	15
MÉDIA GERAL		3.73	0.416	7.27	40.208		10.18	4.27	0.100	6.73	40.316		9.73

PT - Problema de Transporte.

TABELA (IV.2) - VALORES MÉDIOS 1ª E 2ª ETAPAS DO GULOSO APLICADO A FUNÇÕES CÔNCAVAS - APROXIMAÇÃO POR ARCO.

PROBLEMA		APROXIMAÇÃO POR ARCO						APROXIMAÇÃO POR NÓ					
Nº	K <sub>i</sub>	FAC. FEC.	TEMPO (3)	CUSTO TOTAL	REDUÇÃO F.O.(%)	Nº P.T.	Nº ITER.	FAC. FEC.	TEMPO (3)	CUSTO TOTAL	REDUÇÃO F.O.(%)	Nº P.T.	Nº ITER.
I.1	200	0	0.621	970.9	0	1	1	0	0.512	970.9	0	1	1
	400	0	0.578	1144.3	0	1	1	1	2.721	1130.5	1.22	4	11
	800	0	0.606	1491.1	0	1	1	9	10.660	1244.9	19.77	6	61
I.2	200	0	0.511	1006.4	0	1	1	0	0.878	1006.4	0	2	3
	400	0	0.514	1158.1	0	1	1	0	1.978	1156.0	0.18	3	10
	800	0	0.550	1461.4	0	1	1	6	8.086	1264.9	15.53	5	50
I.4	200	0	0.284	1022.2	0	1	1	0	0.202	1022.2	0	1	1
	400	0	0.293	1109.7	0	1	1	0	0.170	1109.7	0	1	1
	800	0	0.305	1284.9	0	1	1	0	0.490	1284.9	0	2	5
IV.1	200	0	0.564	1212.8	0	1	1	0	1.635	1212.7	0	3	5
	400	0	0.627	1385.2	0	1	1	0	1.399	1385.0	0.01	2	4
	800	0	1.563	1729.9	0	2	4	0	1.698	1729.9	0	2	6
IV.2	200	0	0.601	1265.1	0	1	1	0	0.534	1265.1	0	1	1
	400	0	0.596	1432.3	0	1	1	0	0.683	1432.3	0	1	2
	800	0	1.468	1766.5	0	2	4	0	0.697	1766.5	0	1	2
V.1	200	0	0.492	1083.9	0	1	1	0	0.420	1083.9	0	1	1
	400	0	0.514	1235.3	0	1	1	1	3.312	1225.9	0.76	4	17
	800	0	0.485	1538.0	0	1	1	6	7.850	1410.9	9.0	7	41
V.2	200	0	0.420	1115.8	0	1	1	0	0.995	1115.8	0	2	5
	400	0	0.441	1253.9	0	1	1	0	2.066	1253.0	0.07	3	13
	800	0	0.426	1530.0	0	1	1	4	5.869	1430.9	6.92	8	36
V.4	200	0	0.367	1175.3	0	1	1	0	0.689	1175.3	0	2	4
	400	0	0.354	1267.1	0	1	1	0	0.617	1267.0	0	2	4
	800	0	0.336	1480.9	0	1	1	0.5	0.996	1480.7	0.01	2	6
VI.1	200	0	0.799	1039.8	0	1	1	0.5	2.165	1034.7	0.49	3	5
	800	0	0.762	1631.7	0	1	1	1	4.740	1601.3	1.89	3	14
VI.4	200	0	0.654	1264.3	0	1	1	0	0.675	1264.3	0	1	1,5
	800	0	1.091	1745.1	0	1.5	2.5	0	0.655	1745.1	0	1	1,5
MÉDIA GERAL		0	0.640		0	1	1.36	0.82	2.105		4.65	2.36	10.04

PT - Problema de Transporte.

TABELA (IV.3) - PROCESSO ITERATIVO FINAL (3ª ETAPA) GULOSO APLICADO A FUNÇÕES CÔNCAVAS.

Faremos a seguir algumas comparações: o comportamento do algoritmo em cada uma das etapas conforme as aproximações lineares utilizadas, o desempenho de  $\Delta_i^a$  e  $\Delta_i^{aD}$  para a primeira etapa, as implicações relativas as variações nas ofertas, nos custos fixos e nas constantes  $K_i$  nas funções  $f_i(.)$  e finalmente os valores obtidos para as funções objetivas.

Pelas médias gerais das tabelas (IV.1) e (IV.2) compomos a tabela (IV.4) onde podemos analisar a primeira etapa do algoritmo.

APROXIMAÇÃO	$\Delta_i^a$		$\Delta_i^{aD}$	
	FACILIDADES ABERTAS	TEMPO	FACILIDADES ABERTAS	TEMPO
POR NÔ	0.27	0.417	1.27	0.097
POR ARCO	3.73	0.416	4.27	0.100

TABELA (IV.4) - MÉDIAS GERAIS PRIMEIRA ETAPA.

Conforme enunciado anteriormente, a aproximação por arco na primeira etapa, torna os testes de redução para abrir facilidades mais poderosos, aumentando o número de elementos abertos e consequentemente o desempenho do algoritmo, porém, é um critério heurístico. Enquanto isso, os testes com aproximação por nô são excessivamente fracos.

Para a segunda etapa, o algoritmo se comporta independente da aproximação linear utilizada, porém, dependente da primeira etapa, onde será aberto ou fechado um conjunto maior ou menor de facilidades. Veja tabela (IV.5). Todavia, cabe observar que os valores encontrados para as funções objetivos, quer utilizando  $\Delta_i^a$  ou  $\Delta_i^{aD}$  inicialmente, são sempre iguais ao fim da segunda etapa. De um modo geral, esses valores são também independentes das aproximações lineares. (Ver colunas de custo nas tabelas (IV.1) e (IV.2)).

APROXIMAÇÃO	$\Omega_i^{aD}(\Delta_i^a)$		$\Omega_i^{aD}(\Delta_i^{aD})$	
	FACILIDADES ABERTAS	TEMPO	FACILIDADES ABERTAS	TEMPO
POR NÔ	10.73	44.840	9.73	43.716
POR ARCO	7.27	40.208	6.73	40.316

TABELA (IV.5) - MÉDIAS GERAIS SEGUNDA ETAPA.

A eficiência do PIF está significativamente vinculada as aproximações lineares empregadas, tabela (IV.3). O tempo computacional é superior para aproximação por nÔ, porém, altamente justificável se comparado os custos totais, e também, o número de facilidades fechadas para uma e outra aproximação. A diferença de custos em alguns problemas é significativa, consequência de um baixo desempenho da heurística na primeira e segunda etapas.

A aproximação por arco no PIF, garantidamente convergente, não conseguiu nenhuma melhora nas soluções obtidas nas duas primeiras etapas. Ao passo que, a aproximação por  $n\bar{o}$ , apesar de não propiciar um decrescimento da função a cada iteração, algoritmo não convergente, gerou soluções a custos bem inferiores para maioria dos problemas. As funções objetivas que sofreram alterações reduziram em média 4,65% para um tempo médio adicional de 2,1 segundos e um número médio de 2,36 iterações ou problemas de transporte resolvidos.

Pela tabela (IV.4) podemos também comparar  $\Delta_i^a$  e  $\Delta_i^{aD}$ . Independente da linearização, mais uma vez o teste de redução para abrir facilidades, utilizando informações do problema dual,  $\Delta_i^{aD}$ , tem um comportamento superior quer em número de elementos abertos, quer em tempo computacional.

Vimos nos capítulos anteriores que os testes de redução para abrir são sensíveis as variações de capacidade e custos fixos das facilidades. Assim é que, para altos valores de capacidade e baixos custos fixos o número de facilidades abertas na primeira etapa tende a elevar-se. Analisando as tabelas (IV.1) e (IV.2) podemos verificar um comportamento semelhante se as aproximações lineares são feitas por arco. O contrário está ocorrendo para linearizações por  $n\bar{o}$ . Isto porque os altos valores de capacidade ( $a_i$ ) acarretam inclinações  $t_i$  baixas, distanciando da função real e reduzindo o poder de poda. Porém, a elevação de custos fixos sempre implica em possível redução no número de facilidades abertas, independente da linearização.

A elevação das constantes  $K_i$  acarreta em perda de capacidade de poda dos testes de redução para abrir, também devido ao distanciamento da função real da aproximação linear.

Finalmente, destacamos mais uma vez os valores obtidos, para função objetivo, os tempos computacionais e também a simplicidade do algoritmo. Na tabela (IV.6) apresentamos uma comparação entre o desempenho do guloso e um algoritmo separação e avaliação para funções côncavas, proposto por SOLAND (57). A parada deste algoritmo está associada a uma tolerância, que é a diferença máxima entre o limite inferior e superior. Em alguns casos essa tolerância teve de ser bastante grande devido aos elevados tempos computacionais.

PROBLEMA		SOLAND				GULOSO	
Nº	$K_i$	LI (\$1000)	LS (\$1000)	TOLERAN CIA (%)	TEMPO (s)	VALOR F. O (\$1000)	TEMPO (s)
IV.1	200	1200.5	1212.8	1,0	74.26	1212.7	37.858
	400	1367.4	1377.6	1,0	72.96	1385.0	41.972
	800	1699.6	1711.9	1,0	73.98	1729.9	44.150
IV.2	200	1253.1	1265.4	1,0	62.41	1265.1	38.127
	400	1418.4	1432.6	1,0	58,75	1432.3	38.376
	800	1752.1	1766.9	1,0	62.71	1766.5	41.431
IV.4	200	1391.9	1402.9	1,0	65.91	1402.6	37.091
	400	1556.7	1570.1	1,0	61.73	1569.8	43.174
	800	1886.4	1904.4	1,0	63.18	1904.0	43.121
V.1	200	991.1	1108.6	10	101.58	1083.9	36.579
	400	1117.7	1241.6	10	208.21	1225.9	39.599
	800	1294.7	1532.6	15	113.53	1410.9	46.212
V.2	200	1019.8	1135.9	10	153.3	1115.8	34.860
	400	1126.1	1249.9	10	129.85	1253.0	36.649
	800	1347.7	1501.1	10	287.06	1430.9	41.714
V.4	200	1036.4	1221.7	15	76.41	1175.3	33.790
	400	1154.6	1364.0	15	109.5	1267.0	35.335
	800	1352.5	1592.5	15	92.36	1480.7	34.268

Tabela (IV.6) - Comparação entre Guloso e Soland.

CAPÍTULO VRELAXAÇÃO LAGRANGEANA

Até o momento analisamos algoritmos heurísticos abordando o problema geral proposto com funções custo fixo-linear (capítulo III) e fixo-côncavo (capítulo IV). Nesta seção voltamos a tratar o problema de localização com custo fixo-linear da forma:

$$\begin{aligned} \min \quad & \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij} + \sum_{i \in I} F_i y_i \\ \text{(P2)} \quad & \\ \text{s.a.} \quad & \sum_{j \in J} x_{ij} \leq a_i y_i, \quad i \in I \end{aligned}$$

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

$$x_{ij} \geq 0, \quad y_i \in \{0, 1\}, \quad i \in I, \quad j \in J$$

onde  $C_{ij} = c_{ij} + t_i$ , para todo  $i \in I$ ,  $j \in J$  e  $t_i$  uma constante,  $i \in I$ .

Objetivamos um algoritmo exato tipo separação e avaliação, (ou Branch-and-Bound) seguindo a estrutura geral proposta por GEOFFRION e MARSTEN (31) e utilizando a relaxação lagrangeana (RL) como ferramenta básica (NAUSS (48), CHRISTOFIDES e BEASLEY (8)). Estes algoritmos se apresentam sobre a forma de árvore onde os conceitos de limite inferior, superior e separa



ção são fundamentais. Nas seções seguintes trataremos cada um desses elementos. Adiantamos que os limites inferiores serão obtidos pelas relaxações Lagrangeanas da função objetivo e ajustados pelo método de subgradiente. O maior dos limites inferiores será denominado de ZINF. Os limites superiores correspondem a soluções viáveis obtidas durante o processo iterativo. De forma semelhante, o menor limite superior será denominado ZSUP. A separação divide um  $n$  em dois outros, uma vez que trabalhamos com as variáveis binárias  $y_i$ . A solução ótima será obtida quando  $ZINF = ZSUP$ . Também será apresentado uma seção onde enfocamos os testes de redução, capítulo (II), e penalidades que implicam em um menor número de nós na árvore ou redução no tempo computacional. Finalmente, trataremos os aspectos gerais para implementação e a análise de resultados que permitirão uma comparação ao comportamento da heurística gulosa. Esse é um dos objetivos do acréscimo desse capítulo.

Antes porém, apresentaremos a RL dentro do contexto de programação linear inteira mista. O termo RL foi proposto por GEOFFRION (28), onde se encontram as demonstrações dos resultados aqui apresentados. Também destacamos os trabalhos de EVERETT (17), FISHER e outros (22), SHAPIRO (55) e (56), MACULAN e outros (42), etc. FISHER (20) apresenta uma visão geral da teoria e das aplicações de RL na última década. A aplicação inicial da RL ao problema do caixeiro viajante por HELD e KARP (35) motivou o seu largo emprego em problemas de Combinatória. Isto se justifica na medida em que problemas complexos são decompostos em subproblemas de fácil resolução. Ainda mais, é uma ferramenta que fornece bons limites inferiores e que po

dem ser utilizados em algoritmos de separação e avaliação.

Denominaremos  $v(P)$  o valor ótimo para função objetivo do problema (P).

Seja o problema genérico:

$$\min f(x) = c^T x \quad (V.1)$$

$$(P) \quad \text{s.a } g(x) = b - Ax \leq 0 \quad (V.2)$$

$$x \in X \subseteq \mathbb{R}^n \quad (V.3)$$

onde as funções  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  e  $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $X$  um conjunto não vazio, finito e discreto. Implícito nesta formulação está o fato de que as restrições  $g(x)$  dificultam a solução do problema. Podemos assim relaxá-las associando a cada restrição em (V.2) um multiplicador de Lagrange, não negativo, e levando-as a função objetivo. Define-se a RL de (P) relativo a (V.2) e ao vetor  $\lambda \geq 0$ , ao problema:

$$(P_\lambda) \quad L(\lambda) = \min_{x \in X} c^T x + \lambda (b - Ax) \quad (V.4)$$

O problema  $(P_\lambda)$  é uma relaxação de (P) podendo mostrar que para qualquer  $\lambda \geq 0$ ,  $v(P_\lambda) \leq v(P)$ . Ainda mais, o problema  $(P_\lambda)$  deve ser mais simples de se resolver que (P), ou, a estrutura do conjunto  $X$  deve ser especial.

Uma outra alternativa seria relaxar sobre o conjunto

X. A relaxação usual de integralidade das variáveis  $x \in X$  é uma delas. O problema resultante denominamos  $(\bar{P})$ .

No entanto podemos mostrar que:

$$v(\bar{P}) \leq v(P_{\bar{\lambda}}) \leq \max_{\lambda \geq 0} v(P_{\lambda}) \leq v(P) \quad (V.5)$$

onde  $\bar{\lambda}$  são os multiplicadores duais correspondentes a solução ótima de  $(\bar{P})$  (GEOFFRION (28)).

O uso de qualquer relaxação está vinculado a qualidade do limite inferior de aproximar ao valor ótimo de  $(P)$ . Isto justifica a utilização de RL visto que, de (V.5), o limite resultante dessa relaxação é superior ou no mínimo igual ao obtido pela relaxação usual  $(\bar{P})$ , para valores de  $\bar{\lambda}$ . Todavia a escolha ideal para  $\lambda$  é dada pela solução ótima do problema:

$$(P_D) \quad \max_{\lambda \geq 0} L(\lambda) \quad \text{ou} \quad \max_{\lambda \geq 0} v(P_{\lambda})$$

que coincide com o problema dual de  $(P)$ , relativo as restrições (V.2). (GEOFFRION (27)).

Se o valor ótimo de  $(P_{\lambda})$  não altera ao relaxar as restrições de integralidade sobre as variáveis  $x \in X$  então temos a propriedade de integralidade (GEOFFRION (28)), ou:

$$v(\bar{P}) = v(P_{\bar{\lambda}}) = v(P_D) \leq v(P)$$

Desta forma, a relaxação usual é tão boa quanto a lagrangeana. Portanto é mais interessante usar a RL em problemas em que a propriedade de integralidade não é satisfeita, ou para casos em que  $(P_\lambda)$  é mais simples de ser resolvido.

Cabe observar que esta propriedade não está vinculada a uma classe de problemas, mas a formulação de um problema dessa classe. Assim é que, para exemplificar, o problema da p-mediana, dependendo da formulação, após o relaxamento atenderá ou não a propriedade de integralidade (CORNUEJOLS e outros (10)).

Desde que  $X$  é um conjunto finito então  $L(\lambda)$  é um valor real para todo  $\lambda$ . Ainda mais, a função lagrangeana  $L(\lambda)$  é côncava, contínua e não diferenciável. Uma vez que  $v(P_\lambda) = L(\lambda) \leq v(P)$ , poderá existir um  $\hat{\lambda}$  tal que exista um  $\hat{x} \in X$  e  $L(\hat{\lambda}) = f(\hat{x}) = v(P)$ , determinando assim uma solução ótima para  $(P)$ . As condições suficientes de otimalidade para o problema  $(P)$  podem ser assim enunciadas:

$$(i) \quad f(\hat{x}) + \hat{\lambda}g(\hat{x}) = \min_{x \in X} \{f(x) + \hat{\lambda}g(x)\}$$

$$(ii) \quad \hat{\lambda} \cdot g(\hat{x}) = 0 \quad (\text{complementaridade de folga})$$

$$(iii) \quad g(\hat{x}) \leq 0 \quad (\text{viabilidade primal}).$$

"Se  $(\hat{x}, \hat{\lambda})$  satisfazem as condições de otimalidade para o problema  $(P)$  então  $\hat{x}$  é ótimo em  $(P)$ ".

De (V.5) concluímos que a RL é uma boa ferramenta na

determinação de limites inferiores para (P). E o primeiro passo é determinar um valor  $\lambda$  que forneça o maior limite inferior. Este  $\lambda$  corresponde a solução do problema dual. Assim podemos acrescentar ao enunciado acima:

"Se  $(\hat{x}, \hat{\lambda})$  satisfazem as condições de otimalidade para o problema (P) então  $\hat{x}$  é ótimo em (P) e  $\hat{\lambda}$  é ótimo no problema dual  $(P_D)$ ".

Resumindo, uma estratégia para aplicação da RL na resolução de (P) consiste em encontrar uma solução ótima  $\hat{\lambda}$  para o problema dual  $(P_D)$ . Feito isso, tentamos encontrar uma solução complementar  $\hat{x} \in X$  tal que as condições de otimalidade sejam atendidas.

Na resolução do problema dual  $(P_D)$  vários métodos podem ser utilizados. Porém, a maioria dos resultados tem sido obtidos através da aplicação do Método de Subgradiente que descrevemos posteriormente. Outra opção é a resolução de um problema linear generalizado, dual do problema  $(P_D)$ , utilizando métodos de decomposição e geração de colunas (SHAPIRO (56), MACULAN e outros (42)).

No entanto, independente dos métodos, não temos garantia que esta estratégia será bem sucedida. Ou seja, não temos garantia que uma solução do dual  $(P_D)$  possa gerar uma solução para o problema primal (P). Neste caso, temos um "gap" de dualidade ou  $v(P_D) < v(P)$ . Este problema é contornável permitindo a obtenção de uma solução ótima para (P). Uma alternativa é

fortalecer o problema dual ( $P_D$ ), restringindo as soluções possíveis na minimização da função lagrangeana para um conjunto discreto. Isto é feito pela teoria dos grupos. Outra alternativa é utilizar um algoritmo tipo separação e avaliação que trataremos neste trabalho.

## 1. LIMITES INFERIORES

Dado o problema (P2) podemos definir dois limites inferiores pela RL. Relaxando as restrições de demanda ou de oferta e lançando-as na função objetivo. A primeira, já bastante explorada foi utilizada por NAUSS (48), CHRISTOFIDES e BEASLEY (8) para o problema (P2). A segunda é uma proposta alternativa e analisada posteriormente, comparada a primeira sobre os pontos de vista de complexidade computacional e obtenção de limites inferiores mais justos.

Independente da relaxação efetuada, os subproblemas resultantes são de fácil resolução. Ainda mais, se trabalhamos sobre o modelo (P2), definindo-o em função de variáveis  $x'_{ij} = x_{ij}/b_j$ ,  $i \in I$ ,  $j \in J$  ou  $x'_{ij} = x_{ij}/a_i$ ,  $i \in I$ ,  $j \in J$ , eles se tornam mais simples e resolvidos por inspeção. Este tratamento será abordado nas duas seções seguintes.

### 1.1. Relaxar a Demanda

Seja o problema (P2) acima. Fazendo  $x'_{ij} = x_{ij} / b_j$ ,  $i \in I$ ,  $j \in J$ , temos o novo modelo:

$$\min \sum_{i \in I} \sum_{j \in J} C'_{ij} x'_{ij} + \sum_{i \in I} F_i y_i \quad (V.6)$$

$$\sum_{j \in J} b_j x'_{ij} \leq a_i y_i, \quad i \in I \quad (V.7)$$

(P3)

$$\sum_{i \in I} x'_{ij} \geq 1, \quad j \in J \quad (V.8)$$

$$0 \leq x'_{ij} \leq 1, \quad i \in I, \quad j \in J \quad (V.9)$$

$$y_i \in \{0, 1\}, \quad i \in I \quad (V.10)$$

onde  $C'_{ij} = C_{ij} \times b_j$ ,  $i \in I$ ,  $j \in J$ .

Por simplicidade de notação trataremos  $x'_{ij}$  por  $x_{ij}$  e  $C'_{ij}$  por  $C_{ij}$ .

A viabilidade do problema (P2) está garantida pela condição  $\sum_{i \in S} a_i \geq \sum_{j \in J} b_j$ ,  $S \subseteq I$ . De outra maneira, esta viabilidade pode ser dada por:

$$NFI \leq \sum_{i \in I} y_i \leq NFS \quad (V.11)$$

onde NFI = número mínimo de facilidades

e NFS = número máximo de facilidades,  
são constantes fixadas inicialmente.

Estas condições serão incorporadas ao problema (P3).

Introduzindo um conjunto de multiplicadores de Lagrange  $\lambda_j$ ,  $j \in J$  para as restrições de demanda, (V.8), e levando-as a função objetivo em (P3), resulta:

$$L(\lambda) = \min_{\lambda \geq 0} \sum_{i \in I} \sum_{j \in J} (C_{ij} - \lambda_j) x_{ij} + \sum_{i \in I} F_i y_i + \sum_j \lambda_j$$

$$\sum_{j \in J} b_j x_{ij} \leq a_i y_i, \quad i \in I$$

$$(P3_{\lambda}) \quad NFI \leq \sum_{i \in I} y_i \leq NFS$$

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

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

Uma outra relaxação é obtida pela substituição das restrições (V.10) pelas novas condições  $0 \leq y_i \leq 1$ ,  $i \in I$ . O problema resultante desta relaxação de integralidade é denominado de ( $\bar{P}3$ ).

Também podemos mostrar que:

$$v(\bar{P}3) \leq v(P3_{\lambda}) \leq v(P) \quad (V.12)$$



onde  $\bar{\lambda}$  são os multiplicadores duais correspondentes a solução ótima de (P3), (GEOFFRION (28)).

O melhor limite inferior será obtido pela resolução do problema dual:

$$(P3_D) \quad \max_{\lambda \geq 0} L(\lambda) \quad \text{ou} \quad \max_{\lambda \geq 0} v(P3_\lambda)$$

Para um dado  $\lambda \geq 0$  o problema  $(P3_\lambda)$  pode ser facilmente resolvido. Por projeção sobre o espaço das variáveis  $y$ , o problema se reduz a  $m$  subproblemas independentes, um para cada  $y_i$ , função das variáveis  $x$  (GEOFFRION (28)).

Seja  $y_i = 1$ , então os fluxos ótimos  $x_{ij}$ ,  $j \in J$  são dados por:

$$(P3_\lambda^i) \quad \min \sum_{j \in J} (C_{ij} - \lambda_j) x_{ij} + F_i$$

$$\sum_{j \in J} b_j x_{ij} \leq a_i \quad (V.13)$$

$$0 \leq x_{ij} \leq 1, \quad j \in J$$

Portanto, temos um conjunto de  $m$  subproblemas da mochila contínuo, de solução imediata.

Basta ordenar os  $x_{ij}$ , em ordem crescente das razões  $(C_{ij} - \lambda_j)/b_j$  e alocar valores as variáveis  $x_{ij}$ , obedecendo as capacidades

des. As variáveis não explicitadas são nulas.

Caso  $y_i = 0$  então  $x_{ij} = 0$  para todo  $j \in J$ .

Com os valores  $v(P3_\lambda^i)$  para todo  $i \in I$ , podemos facilmente calcular  $v(P3_\lambda)$  por:

$$v(P3_\lambda) = \min \left[ \sum_{i \in I} v(P3_\lambda^i) y_i + \sum_{j \in J} \lambda_j \right]$$

$$NFI \leq \sum_{i \in I} y_i \leq NFS \quad (V.14)$$

$$y_i \in \{0, 1\}$$

Um problema 0/1 resolvido por inspeção. Ordenamos os  $v(P3_\lambda^i)$  em ordem crescente. Fazemos  $y_i = 1$  para todo  $i$  correspondente aos primeiros NFI menores valores de  $v(P3_\lambda^i)$ . Se  $NFI = NFS$  paramos, caso contrário, continuamos fazendo  $y_i = 1$  enquanto  $v(P3_\lambda^i) < 0$  ou até que tenhamos selecionado NFS facilidades abertas. As variáveis não escolhidas são nulas.

Em resumo, a solução de  $(P3_\lambda)$  consiste de  $(m+1)$  subproblemas de baixa complexidade, gerando o conjunto  $I_\lambda = \{i | y_i = 1\}$  e a solução relaxada  $x^r$ .

## 1.2. Relaxar a Oferta

Seja o problema (P3) com a condição de viabilidade

(V.11):

$$NFI \leq \sum_{i \in I} y_i \leq NFS.$$

Introduzindo um conjunto de multiplicadores de Lagrange,  $u_i$ ,  $i \in I$ , para as restrições de oferta, (V.7), e levando-as a função objetivo em (P3), resulta:

$$L(u) = \min_{u \geq 0} \sum_{i \in I} \sum_{j \in J} (C_{ij} + u_i b_j) x_{ij} +$$

$$+ \sum_{i \in I} (F_i - a_i u_i) y_i$$

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

$$(P3_u) \quad NFI \leq \sum_{i \in I} y_i \leq NFS$$

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

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

De forma semelhante, podemos mostrar que:

$$v(\bar{P}3) \leq v(P3_{\bar{u}}) \leq v(P) \quad (V.15)$$

onde  $\bar{u}$  são os multiplicadores duais correspondentes a solução ótima de ( $\bar{P}3$ ).

Na verdade, se analisamos o conjunto de soluções viáveis de  $(P3_u)$  podemos verificar que se relaxarmos as variáveis  $y_i$ ,  $0 \leq y_i \leq 1$ ,  $i \in I$ , não implicará em alteração na solução final, uma vez que a matriz de restrições é unimodular. Logo, a propriedade de integralidade é satisfeita e (V.15) pode ser substituído por:

$$v(\bar{P}3) = v(P3_{\bar{u}}) \leq v(P)$$

A RL é semelhante a relaxação usual. No entanto, a simplicidade de cálculo do problema relaxado  $(P3_u)$  é significativa, como veremos abaixo, justificando seu uso. Restará comparar os limites inferiores em ambas as relaxações lagrangeanas, que também será feito posteriormente.

O problema dual de  $(P3)$  relativo as restrições (V.7), é da forma:

$$(P3_D) \quad \max_{u \geq 0} L(u) \quad \text{ou} \quad \max_{u \geq 0} v(P3_u)$$

Para um dado vetor  $u$ , o problema  $(P3_u)$  pode ser facilmente resolvido. Isto porque, as variáveis  $x_{ij}$  e  $y_i$  são independentes. Podemos dividir  $(P3_u)$  em dois subproblemas, um com as variáveis  $x$ , outro com as variáveis  $y$ :

$$\begin{array}{ll}
 \min \sum_{i \in I} \sum_{j \in J} (C_{ij} + u_i b_j) x_{ij} & \min \sum_{i \in I} (F_i - a_i u_i) y_i \\
 (P3_u^1) \quad \sum_{i \in I} x_{ij} \geq 1, j \in J & (P3_u^2) \quad NFI \leq \sum_{i \in I} y_i \leq NFS \\
 0 \leq x_{ij} \leq 1, i \in I, j \in J & y_i \in \{0,1\}, i \in I
 \end{array}$$

O problema  $(P3_u^1)$  é resolvido fazendo  $x_{ij} = 1$  para cada  $j$  correspondente ao elemento  $i$  com o menor valor de  $(C_{ij} + u_i b_j)$ .  $(P3_u^2)$  é resolvido por inspeção. Ordena crescentemente os valores de  $(F_i - u_i a_i)$ . Faça  $y_i = 1$  para todo  $i$  correspondente aos primeiros NFI elementos. Se  $NFI = NFS$  então pare. Caso contrário, prossegue a seleção enquanto  $(F_i - u_i a_i) \leq 0$  ou atinja NFS elementos.

Portanto a solução de  $(P3_u)$  consiste em resolver dois subproblemas de baixa complexidade.

### 1.3. Comparação dos Limites Inferiores

Essa comparação pode ser feita a dois níveis. Primeiro analisando a complexidade dos subproblemas resolvidos. Segundo, a capacidade de gerar limites mais justos.

Resolvemos  $(m+1)$  subproblemas em  $(P3_\lambda)$  enquanto em  $(P3_u)$  apenas dois. Porém, o conjunto de operações é basicamente o mesmo em ambos os casos. Isto acarreta um equilíbrio em termos de complexidade computacional.

Quanto aos limites gerados, também não temos uma superioridade definida. Pela montagem de contra-exemplos (RIBEIRO (52)) podemos verificar casos em que o primeiro limite é mais justo que o segundo e vice-versa.

O desempenho de uma e outra relaxação só ficará bem caracterizado pela execução de testes computacionais, analisando e identificando condições em que haja uma superioridade bem definida.

Nas seções seguintes caracterizaremos as demais etapas para um algoritmo separação e avaliação, voltadas para a relaxação de demanda, restrições (V.8), conforme CHRISTOFIDES e BEASLEY (8). Porém, poderemos refazê-las para a segunda relaxação.

## 2. LIMITE SUPERIOR

Os algoritmos separação e avaliação padrões utilizam como limite superior inicial  $ZSUP = +\infty$ . A medida que soluções viáveis para o problema proposto são obtidas, este limite vai sendo atualizado, igual ao menor valor encontrado para  $f(x)$ .

Ao utilizar a RL juntamente com o método de subgradiente, este exige um limite superior. O bom desempenho do método está vinculado, entre outros fatores, a qualidade do limite. Qualidade que também influi diretamente no tamanho da árvore gerada pelo algoritmo de separação e avaliação. Neste caso, o limite superior inicial pode ser definido pela utilização de uma heurística imediata. Seleciona um conjunto mínimo de facilidades capaz de atender o total de demanda e resolve o problema de transporte correspondente.

Como nosso objetivo é obter resultados que permitam analisar o desempenho da heurística gulosa, utilizamos a solução final obtida por essa heurística como limite superior inicial para o algoritmo.

Durante o processo iterativo, novas soluções viáveis são geradas, veja seção seguinte, e sempre que possível atualizamos o limite superior.

### 3. DETERMINAÇÃO DE UMA SOLUÇÃO VIÁVEL

Seja  $x^r$  e  $y^r$  ( $y_i^r = 1$  se  $i \in I_\lambda$ ) a solução obtida ao resolver o problema relaxado  $(P3_\lambda)$ . Uma vez que as restrições de demanda foram relaxadas, essa solução não é necessariamente viável em  $(P3)$ . Todavia, uma solução viável pode ser facilmente obtida, utilizando as variáveis  $x^r$  e  $y^r$ . Isto implicará na possibilidade de limites superiores melhores e conseqüentemente na

redução da árvore pesquisada.

Seja  $x^V$  e  $y^V$  a solução viável correspondente. Esta será obtida com as seguintes etapas:

- (i) Inicialmente ordenamos os elementos  $j \in J$  decrescentemente por demandas ( $b_j$ ).
- (ii) Para cada elemento  $j \in J$ , ordenadamente, procuramos atender a sua demanda com as facilidades  $i \in I_\lambda$  tal que  $x_{ij}^r \neq 0$  segundo o critério de menor custo.
- (iii) Caso não seja possível atender toda a demanda de  $j$  no item (ii) então passamos a procurar entre as facilidades  $i \in I_\lambda$  e  $x_{ij}^r = 0$  e com capacidade disponível, também pelo critério de menor custo.
- (iv) Caso não seja possível atender a demanda em todos os  $j \in J$  então não existe solução viável.

Algoritmo:

Passo 0. Ordene decrescentemente por demandas ( $b_j$ ) os elementos  $j \in J$ . Seja  $\bar{J} = \{\bar{j}_1, \dots, \bar{j}_n\}$  esse conjunto.

Passo 1. Faça  $K = 1$ ,  $j = \bar{j}_K$ ,  $a_i^j = a_i, \forall i \in I$ ,  $b_j^j = b_j, \forall j \in J$

Passo 2. Se  $\exists i \in I_\lambda$  tal que  $a_i^j \neq 0$  e  $x_{ij}^r \neq 0$  então vá para Passo 6.



Passo 3. Selecione  $i_s \in I_\lambda$  tal que  $C_{i_s j} = \min_{\ell | x_{\ell j}^r \neq 0} \{C_{\ell j}\}$ .

Passo 4. Se  $a_{i_s}^1 \geq b_j^1$  então faça  $x_{i_s j}^V = b_j^1 / b_j$ ,  $a_{i_s}^1 = a_{i_s}^1 - b_j^1$   
vã para Passo 10.

Passo 5. Faça  $x_{i_s j}^V = a_{i_s}^1 / b_j$ ,  $b_j^1 = b_j^1 - a_{i_s}^1$ ,  $a_{i_s}^1 = 0$

Retorne ao Passo 2.

Passo 6. Se  $\nexists i \in I_\lambda$  tal que  $a_i^1 \neq 0$  e  $x_{ij}^r = 0$  então Pare. Não  
existe solução viável.

Passo 7. Selecione  $i_s \in I_\lambda$  tal que  $C_{i_s j} = \min_{\ell | x_{\ell j}^r = 0} \{C_{\ell j}\}$

Passo 8. Se  $a_{i_s}^1 \geq b_j^1$  então faça  $x_{i_s j}^V = b_j^1 / b_j$ ,  $a_{i_s}^1 = a_{i_s}^1 - b_j^1$

vã para o Passo 10.

Passo 9. Faça  $x_{i_s j}^V = a_{i_s}^1 / b_j$ ,  $b_j^1 = b_j^1 - a_{i_s}^1$ ,  $a_{i_s}^1 = 0$

Retorne ao Passo 6.

Passo 10. Faça  $K = K + 1$ .

Passo 11. Se  $K \leq N$  então faça  $j = \bar{j}_k$

retorne ao Passo 2.

Passo 12. Pare.  $x^V$  é a solução viável em (P3).

#### 4. TESTES DE REDUÇÃO - PENALIDADES

Sejam os conjuntos definidos anteriormente:

$K_0$  = conjunto das facilidades fixadas fechadas.  
 $y_i = 0$ . Inicialmente  $\bar{K}_0$  é um conjunto vazio.

$K_1$  = conjunto das facilidades fixadas abertas,  
 $y_i = 1$ . Inicialmente  $\bar{K}_1$  é também um conjunto vazio.

$K_2$  = conjunto das facilidades não fixadas ou livres.  
 Inicialmente  $K_2 = I = \{1, \dots, m\}$ .

Para cada nó da árvore do algoritmo de separação e avaliação temos os três conjuntos básicos acima. E todos os cálculos estão voltados para esses conjuntos. Assim é que, ao resolver  $(P3_\lambda)$  em (V.14), o conjunto de todas as facilidades selecionadas,  $I_\lambda$ , deve automaticamente conter o conjunto  $K_1$ . Consequentemente ao caracterizar uma solução viável também estaremos levando  $K_1$  em consideração.

Esses conjuntos também são importantes para aplicação dos testes de redução para abrir e fechar facilidades que são aplicados no início e durante o processo iterativo. Os testes são utilizados tal qual definidos no capítulo (II):

- (i) para todo  $i \in K_2$  faça  $\Delta_i = W(K_1 \cup K_2 - i) - W(K_1 \cup K_2)$   
 se  $\Delta_i \geq F_i$  então faça  $y_i = 1$  ou  $K_1 = K_1 \cup \{i\}$   
 $K_2 = K_2 - i$

(ii) para todo  $i \in K_2$  faça  $\Omega_i = W(K_1) - W(K_1 \cup i)$

se  $\Omega_i \leq F_i$  então faça  $y_i = 0$  ou  $K_2 = K_2 - i$

$K_0 = K_0 \cup \{i\}$

Também aqui os cálculos podem ser exatos ( $\Delta_i$  e  $\Omega_i$ ) ou aproximados ( $\bar{\Delta}_i$  ou  $\Delta_i^D$  e  $\bar{\Omega}_i$  ou  $\Omega_i^D$ ). Porém, devido as características do algoritmo, apenas os testes aproximados se justificam para aplicações. Mais especificamente, sugerimos  $\Delta_i^D$  e  $\Omega_i^D$ .

Tal qual os testes de redução, podemos definir penalidades (CHRISTOFIDES e BEASLEY (8)) que poderão forçar a abertura ou fechamento de facilidades, possibilitando uma redução na árvore pesquisada. Também divididas em penalidades para abrir e fechar facilidades.

Se ao abrir uma facilidade  $i \in (K_2 - I_\lambda)$ , o custo mínimo ou penalidade mínima resultante, adicionada ao limite inferior ZINF, superar o limite superior ZSUP, então esta facilidade deve ser fechada. Da mesma forma, se ao fechar uma facilidade  $i \in (I_\lambda - K_1)$ , a penalidade mínima resultante, adicionada a ZINF, superar o limite superior ZSUP então essa facilidade deve ser aberta.

#### PENALIDADE PARA ABRIR $i \in (K_2 - I_\lambda)$

(i)  $v(P3_\lambda^i)$  se  $|I_\lambda| \neq \text{NFI}$  e  $|I_\lambda| \neq \text{NFS}$

$$(ii) \quad v(P3_{\lambda}^i) - \max_{j \in (I_{\lambda} - K_1)} v(P3_{\lambda}^j) \text{ se } |I_{\lambda}| = \text{NFS}$$

$$(iii) \quad v(P3_{\lambda}^i) - \max \{0, \max_{j \in (I_{\lambda} - K_1)} v(P3_{\lambda}^j)\} \text{ se } |I_{\lambda}| \neq \text{NFS e} \\ |I_{\lambda}| = \text{NFI}$$

PENALIDADE PARA FECHAR  $i \in (I_{\lambda} - K_1)$

$$(iv) - v(P3_{\lambda}^i) + \min_{j \in (K_2 - I_{\lambda})} v(P3_{\lambda}^j) \text{ se } |I_{\lambda}| = \text{NFI}$$

$$(v) - v(P3_{\lambda}^i) + \min \{0, \min_{j \in (K_2 - I_{\lambda})} v(P3_{\lambda}^j)\} \text{ se } |I_{\lambda}| \neq \text{NFI}$$

Essas penalidades procuram preservar os limites NFI e NFS para o número de facilidades abertas na solução ótima. Assim, no caso (i) estamos no interior do intervalo [NFI, NFS] e o custo mínimo associado a possibilidade de abrir a facilidade  $i$  é dado por  $v(P3_{\lambda}^i)$ . Lembramos que de (V.13),  $v(P3_{\lambda}^i)$  corresponde a soma do mínimo custo variável de distribuição da capacidade  $a_i$  entre os centros de demanda  $j \in J$ , mais o custo fixo  $F_i$ . Em (ii), estamos com o número máximo, NFS. Logo, para abrir  $i$  faz-se necessário retirar alguma outra facilidade  $j \in (I_{\lambda} - K_1)$ . Em (iii) temos o inverso, estamos com o número mínimo. Neste caso, se o custo mínimo da facilidade  $j \in (I_{\lambda} - K_1)$  for positivo, ela deve ser retirada com a entrada de  $i$ , permanecendo NFI elementos. Caso contrário, podemos entrar com  $i$  sem que nenhuma facilidade seja retirada.

As penalidades para fechar se comportam da mesma forma. A função decrescerá no mínimo de  $v(P3_{\lambda}^i)$  com a possibilidade de fechar  $i$ . Porém, crescerá obrigatoriamente, pelo acréscimo do elemento  $j \in (K_2 - I_{\lambda})$ , com o menor custo, se o número de facilidades abertas está no mínimo,  $|I_{\lambda}| = \text{NFI}$ , em (iv). Caso contrário poderemos retirar  $i$  sem nenhum acréscimo a função ou até mesmo um decréscimo correspondente a uma facilidade  $j \in (K_2 - I_{\lambda})$  que poderá substituir  $i$  com uma variação de custo negativa  $v(P3_{\lambda}^j)$ , item (v).

#### PENALIDADE PARA FECHAR INTERVALO [NFI, NFS]

Uma outra penalidade pode ser definida com o objetivo de fechar o intervalo [NFI, NFS] que contém o número de facilidades abertas. Isto reforça as penalidades acima e implica em limites inferiores mais justos. Se impomos a condição que na solução ótima ocorra  $\ell$  facilidades abertas então de (V.14), o valor mínimo para  $(P3_{\lambda})$  é dado por:

$$Z = \sum_{j \in J} \lambda_j + \sum_{i \in K_1} v(P3_{\lambda}^i) + (\ell - |K_1|) \min_{j \in K_2} \{v(P3_{\lambda}^j)\} \quad (V.16)$$

Se  $\ell = \text{NFI}$  e  $Z > \text{ZSUP}$  então podemos fazer  $\text{NFI} = \text{NFI} + 1$ ;  
se  $\ell = \text{NFS}$  e  $Z > \text{ZSUP}$  então podemos fazer  $\text{NFS} = \text{NFS} - 1$ ,  
sem que a solução ótima seja alterada.

Todos esses resultados são consequências diretas do significado do valor de  $v(P3_{\lambda}^i)$ , ou, o custo mínimo associado em

abrir a facilidade  $i$ . Será positivo se o ganho com os custos variáveis decorrente do acréscimo de  $i$  for inferior (em termos absolutos) ao custo fixo  $F_i$ . Será negativo, caso contrário.

## 5. SEPARAÇÃO

O processo de separação em um algoritmo de separação e avaliação consiste em dividir um problema, ou não, em um conjunto de subproblemas, ou não, de tal forma que a união dos conjuntos de soluções viáveis dos subproblemas corresponda ao conjunto de soluções viáveis origem. Para o caso específico, um problema de programação inteira 0/1, separamos um problema em dois outros fixando uma variável  $y_i$  em 0 em um subproblema e em 1 em outro.

A escolha da variável a ser fixada na divisão do problema poderá seguir vários critérios. A facilidade  $i \in K_2$  tal que a penalidade associada seja máxima ou aquela que forneça o mínimo  $v(P3_\lambda^i)$  (a facilidade com a maior possibilidade de ser aberta). Não existe o melhor critério, tudo depende de testes.

Os critérios para escolha do problema ou não a ser dividido são dois:

- (i) LIFO - o último subproblema gerado é o primeiro a ser dividido. Este processo procura uma solução rápida que permita podar um número maior de nós.

(ii) Selecionar o nó com o menor limite inferior, possível candidato a uma solução a baixo custo.

Nestes critérios existe um compromisso em reduzir o espaço de memória ocupada, ampliando a árvore na vertical, em (i), enquanto em (ii) procura-se reduzir o tempo computacional, ampliando a árvore na horizontal. Podemos trabalhar com um ou outro ou os dois intercalados.

## 6. MÉTODO DE SUBGRADIENTE

É o método mais popular para resolução de problemas duais em RL. Isto porque é fácil de programar, os subgradientes são facilmente obtidos e é superior aos demais para alguns problemas em combinatória em que o número de variáveis é muito elevado ou quando ocorre o "gap" de dualidade, sendo necessário um método simples. Porém não existe garantia que os limites inferiores serão crescentes para os multiplicadores duais gerados por este método. Embora na prática bons resultados tem sido obtidos após alguns ajustes.

O método de subgradiente é uma adaptação do método de gradiente em que os gradientes são substituídos por subgradientes. Dado um vetor  $\lambda^K \geq 0$ , interação K, solucionamos o problema  $(P_{\lambda^K})$ , (V.4), que nos fornece os subgradientes  $g(x^K) = b - Ax^K$ , correspondentes a solução ótima  $x^K$ , permitindo obter um novo  $\lambda^{K+1}$  dado por:

$$\lambda^{K+1} = \lambda^K + \alpha_K \frac{g(x^K)}{\|g(x^K)\|} = \lambda^K + \alpha_K (b - Ax^K) \quad (V.17)$$

onde  $\alpha_K$  é um escalar positivo, o tamanho do passo.

Aspectos computacionais e propriedades de convergência teórica do método são discutidos em HELD, WOLFE e CROWDER (37), GOFFIN (33), RIBEIRO (51), MAHEY (43).

Uma das formas de caracterizar o tamanho do passo é dada por:

$$\alpha_K = \rho_K \frac{(ZSUP - L(\lambda^K))}{\|g(x^K)\|^2} \quad (V.18)$$

onde  $\rho_K$  é um escalar satisfazendo  $0 < \rho_K \leq 2$ , ZSUP um limite superior para (P) e  $L(\lambda^K) \leq \max L(\lambda)$ , em  $(P_\lambda)$ . E  $\|g(x^K)\|$  é uma norma qualquer do subgradiente, em geral, a euclidiana.

Teoricamente, para sucessivos K, fixa-se  $\rho_K$ , e  $L(\lambda^K)$  converge a ZSUP ou é obtido um ponto em que  $L(\lambda^K) \geq ZSUP$ . POLYAK (50) mostrou que  $L(\lambda^K)$  converge ao  $\max L(\lambda)$  se  $\alpha_K \rightarrow 0$  e  $\sum_{K=1}^{\infty} \alpha_K = +\infty$ . Seguindo a proposta de HELD e outros (37) fazemos  $\rho_K = 2$  durante  $2n$  iterações, então dividimos  $\rho_K$  por dois após as  $2n$  iterações, após  $n$  iterações, e  $n/2$ ,  $n/4$ , ..., até atingirmos um número mínimo de iterações, no nosso caso igual a 5. Se que dividindo  $\rho_K$  por dois, após cada 5 iterações, até que  $\rho_K$  se torne suficientemente pequeno.



O valor  $\lambda^{K+1}$  definido em (V.17) não é necessariamente não negativo. Para que essa condição seja satisfeita fazemos:

$$\lambda_i^{K+1} = \max \{0, \lambda_i^K + \alpha_K g_i(x^K)\}, \quad i = 1, \dots, m \quad (V.19)$$

Apresentamos a seguir a estrutura do algoritmo de subgradiente, utilizando os resultados acima aplicados a resolução de (P3). Nesse algoritmo será introduzido as informações apresentadas nas seções anteriores entre elas as penalidades e os testes de redução. Apesar de iterativo omitiremos os índices K acima.

ALGORITMO:

Passo 0. Faça  $\rho = 2$ ,  $ZINF = -\infty$

Escolha um valor inicial para  $\lambda$

$$\lambda_j = \min_{i \in (K_1 \cup K_2)} \{C_{ij}\}, \quad j = 1, \dots, n \quad (V.20)$$

Passo 1. Determine um limite superior para (P3), ZSUP. (Seção V.2).

Passo 2. Para o conjunto de multiplicadores  $\lambda$  determine  $v(P3_\lambda)$ , (V.14), caracterizando a solução ótima  $x^r$  e o conjunto de facilidades abertas  $I_\lambda$ .

Passo 3. Atualiza, se necessário, o limite inferior ZINF.

Passo 4. De  $x^r$  e  $I_\lambda$  determine a solução viável em (P3) (Seção (V.3)).

Passo 5. Atualiza, se necessário, o limite superior ZSUP.

Passo 6. Se ZSUP = ZINF então PARE. SAÍDA1.

Passo 7. Execute os testes com as penalidades para abrir e fechar facilidades (Seção V.4). Atualiza  $K_1$  e  $K_2$ . Execute o teste para fechar o intervalo  $[NFI, NFS]$ , (V.16).

Passo 8. Se ocorreu alguma alteração em  $K_1, K_2$  no passo anterior, execute ciclicamente os testes de redução para abrir e fechar facilidades. Atualiza  $K_1$  e  $K_2$ .

Passo 9. Defina o vetor de subgradientes SUBG:

$$\text{SUBG}(j) = 1 - \sum_{i \in I_\lambda} x_{ij}^r, \quad j = 1, \dots, n$$

Passo 10. Se  $\text{SUBG}(j) = 0, \forall j \in J$ , então PARE. SAÍDA2.

Passo 11. Calcule o tamanho do salto  $\alpha = \rho \frac{[ZSUP - v(P3_\lambda)]}{\|\text{SUBG}\|^2}$ , (V.18)

Passo 12. Atualize os multiplicadores, (V.19):

$$\lambda_j = \max \left\{ \min_{i \in (K_1 \cup K_2)} (C_{ij}), \lambda_j + \alpha \cdot \text{SUBG}(j) \right\}, \quad j=1, \dots, n$$

Passo 13. Atualize  $\rho$ , se necessário, e retorne ao Passo 2.

Peelo algoritmo temos três pontos de parada. Essas saídas correspondem a:

- (i) O limite superior concide com o inferior;
- (ii) O vetor subgradiente se anula;
- (iii)  $\rho < 0.01$

No caso (i) temos uma solução ótima para (P3). Em (ii), a solução  $x^r$  é viável em (P3) e pode ser ótima. Finalmente, a saída 3, temos um excesso de iterações ou  $\rho < 0.01$ .

## 7. IMPLEMENTAÇÃO

Para implementação de um algoritmo de separação e avaliação destacamos três tipos básicos de nós: inicial, intermediários e finais ou terminais.

### NÓ INICIAL:

Inicialmente aplicamos de maneira cíclica os testes de redução para abrir e fechar facilidades, caracterizando os conjuntos  $K_0$ ,  $K_1$  e  $K_2$ .

Por uma heurística qualquer encontramos um limite superior para (P3). Nesse passo, podemos entrar com a solução obtida pela heurística gulosa.

O método de subgradiente é aplicado conforme seção anterior. Teremos três saídas possíveis. Se ocorrer a primeira, limites superior e inferior iguais, teremos achado a solução ótima para o problema. Caso contrário, passaremos a separação do  $n\bar{o}$  em dois outros. Antes porém, resolvemos um problema de transporte, se viável, com as facilidades em  $K_1$ , com o objetivo de obter um melhor limite superior ZSUP. Aplica-se novamente o método de subgradiente a partir dos valores de  $\lambda$  correspondentes ao melhor limite inferior, para um número fixo de iterações (no caso 30) e um valor fixo para  $\rho (= 0,1)$ . Voltando a ocorrer as saídas (ii) e (iii) dividimos o  $n\bar{o}$  inicial em dois outros. Fazemos  $y_j = 0$  e  $y_j = 1$ , seguindo as orientações dadas na seção (V.5).

#### NÃO INTERMEDIÁRIO:

Em cada  $n\bar{o}$  necessitamos das seguintes informações: os multiplicadores  $\lambda$  e o respectivo melhor limite inferior obtido no  $n\bar{o}$  pai, os valores de NFI e NFS e os conjuntos  $K_1$  e  $K_2$ .

Aplicamos os testes de redução para abrir e fechar facilidades de modo cíclico.

O método de subgradiente é aplicado para um número fixo de iterações (30) e também  $\rho$  fixo ( $= 0,1$ ). As penalidades são calculadas a cada iteração do algoritmo e sempre que atualizamos  $K_1$  e  $K_2$ , também reaplicamos os testes de redução.

NÃO TERMINAL:

Aquele em que  $K_2 = \emptyset$ . Resolvemos o problema de transporte com todas as facilidades em  $K_1$  e atualizamos, se necessário, o limite superior.

Apesar de apresentarmos o algoritmo geral de separação e avaliação, nossos testes computacionais se prenderão apenas ao  $n\tilde{o}$  inicial, com os resultados na seção seguinte.

8. ANÁLISE DE RESULTADOS

Os testes computacionais foram realizados utilizando os problemas IV.1, IV.4, V.1, V.4 e VI.2, Apêndice D. IV.1 e V.1 correspondem aos problemas IV e V com os custos fixos de todas as facilidades iguais a \$ 7.500. Idem para IV.4 e V.4 com custos fixos iguais a \$ 25.000. E VI.2 é o problema VI com todos os custos fixos iguais a \$ 12.500.

Ao aplicar o algoritmo apresentado, necessitamos de um limite superior inicial. Em alguns testes utilizamos a solução encontrada pela heurística gulosa do capítulo III. Em outros fornecemos o limite superior igual ao valor ótimo do problema de transporte  $T(K_1 \cup K_2)$  ou  $T(I)$ , em que todas as facilidades candidatas estão disponíveis. É portanto um limite superior maior que o primeiro.

A constante NFI é feita igual ao menor número de facilidades necessárias para garantir a viabilidade. E  $NFS = |I|$ , número de facilidades candidatas.

Também aqui testamos para cada problema um conjunto de 3 combinações que diferem pelos testes de redução:

1. Testes de redução  $\bar{\Delta}_i$  sem Extensão e  $\bar{\Omega}_i^D$
2. Testes de redução  $\bar{\Delta}_i$  e  $\bar{\Omega}_i$  com Extensão
3. Testes de redução  $\Delta_i^D$  e  $\Omega_i^D$

Os parâmetros utilizados pelo método de subgradiente e especificados nas seções anteriores, não sofreram qualquer variação para possível verificação do desempenho do método.

As tabelas (V.1) e (V.2) resumem os resultados obtidos utilizando um computador Burroughs B6700. Em (V.1) temos os valores dos limites e número de facilidades. E em (V.2) os tempos computacionais e informações adicionais. Foi fixado, na maioria dos testes, um tempo máximo de processamento de 180 segundos. De forma que, em alguns problemas, esse tempo foi excedido não fornecendo todas as informações referentes aos cálculos efetuados para o nó inicial.

Devido ao pequeno número de testes realizados e, na maioria dos casos, o tempo computacional elevado, podemos reunir poucas conclusões.

A utilização de limites superiores iniciais diferentes para um mesmo problema, em nada mudou o desempenho do algoritmo.

Comparando, para cada problema, o número de facilidades abertas e fechadas, pelos testes de redução, tabela (V.1), podemos verificar a superioridade dos testes duais  $\Delta_i^D$  e  $\Omega_i^D$ . Porém, uma vez que cada um desses testes exigem a resolução de pelo menos um problema de transporte, o tempo computacional é maior para a combinação 3, onde são utilizados (Tabela V.2).

Em nenhum desses problemas atingimos o valor ótimo para a função objetivo no nó inicial. Nos problemas V.1 e V.4 conseguimos um melhor desempenho.

Finalmente, comparemos esses resultados com aqueles obtidos pela heurística gulosa. Pelos tempos de processamento apresentados aqui (última coluna tabela (V.2) e pela tabela (III.3), e também pela qualidade das soluções encontradas, podemos afirmar que o algoritmo guloso é totalmente justificável.

Cabe observar que o algoritmo de separação e avaliação com a RL é eficiente, porém, o método de subgradiente, apesar do baixo tempo computacional consumido, precisa ser bem trabalhado. O mesmo pode ser dito para a subrotina problema de transporte que em todos os algoritmos consome a maior parte do tempo total de processamento.

PROBLEMA		VALOR SOL. INICIAL	TESTES REDUÇÃO		MÉTODO SUBGRADIENTE		P. T. EM $K_1$ $Z(K_1)$	MÉTODO SUBGRADIENTE	
Nº	COMB.		FAC. ABER.	FAC. FEC.	LIMITE INFERIOR	LIMITE SUPERIOR		LIMITE INFERIOR	LIMITE SUPERIOR
IV.1	1	1040.4	9	0	1007.9	1040.4			
	2	1040.4	9	0	1013.8	1040.4			
	3	1040.4	11	0	1011.3	1040.4			
IV.4	1	1235.5	2	0	1200.7	1235.5			
	2	1235.5	2	0	1194.4	1235.5			
	3	1235.5	8	0	1194.4	1235.5			
V.1	1	950.4	10	2	866.8	950.4	940.0	866.8	940.0
	2	950.4	10	2	866.8	950.4	940.0	866.8	940.0
V.1	1	932.6	10	2	866.8	932.6	940.0	866.8	932.6
	2	932.6	10	2	866.8	932.6	940.0	866.8	932.6
	3	932.6	10	2	866.8	932.6	940.0	866.8	932.6
V.4	1	1212.9	3	0	912.9	1212.9		912.9	1212.9
	2	1212.9	3	0	912.9	1212.9		912.9	1212.9
	3	1212.9	3	0	912.9	1212.9		912.9	1212.9
V.4	1	1060.7	3	0	912.9	1060.7		912.9	1060.7
	2	1060.7	3	0	912.9	1060.7		912.9	1060.7
	3	1060.7	3	0	912.9	1060.7		912.9	1060.7
VI.2	1	987.8	3	0	839.8	961.8	929.4		
	2	987.8	3	0	839.8	961.8			
	3	987.8	7	0	839.8	961.8			

OBS.: Soluções ótimas para esses problemas na tabela (III.8)

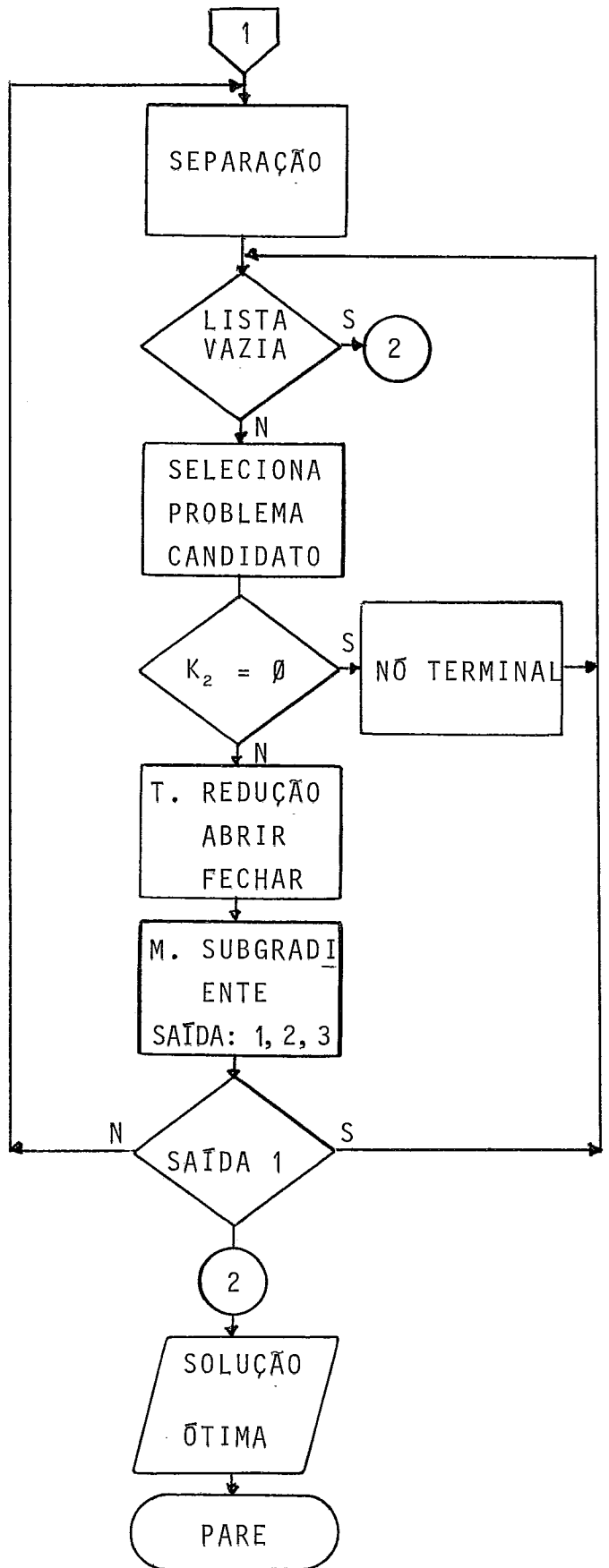
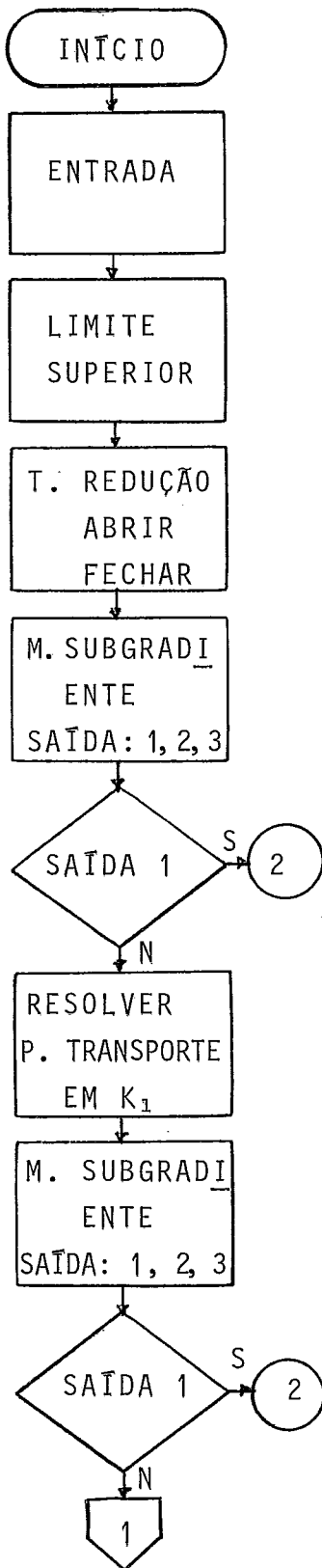
TABELA (V.1) - VALORES DOS LIMITES E FACILIDADES.



PROBLEMA		TESTES REDUÇÃO	MÉTODO SUBGRADIANTE (1ª FASE)		MÉTODO SUBGRADIANTE (2ª FASE)		FINAL			
Nº	COMB.	TEMPO PROC.	TEMPO PROC.	$\rho$ FINAL	TEMPO PROC.	$\rho$	Nº P.T.	Nº I. P. T.	Nº IT. SUB.	TEMPO PROC.
IV.1	1	0.396		1.0					112	140
	2	0.394		1.0					147	140
	3	16.964		1.0					117	40
IV.4	1	0.377		1.0					102	180
	2	0.330		2.0					84	180
	3	15.824		2.0					75	180
V.1	1	9.918	0.905	2.0	0.944	2.0	3	77	2	23
	2	10.073	0.962	2.0	0.936	2.0	3	77	2	23
V.1	1	9.090	0.960	2.0	0.908	2.0	3	77	2	21
	2	9.770	1.113	2.0	1.053	2.0	3	77	2	24
	3	24.003	1.101	2.0	1.054	2.0	4	113	2	36
V.4	1	0.320	0.833	2.0	0.813	2.0	0	0	2	4
	2	0.397	0.973	2.0	0.935	2.0	0	0	2	4
	3	13.233	0.854	2.0	0.897	2.0	1	48	2	17
V.4	1	0.378	1.155	2.0	1.075	2.0	0	0	2	5
	2	0.450	1.156	2.0	1.027	2.0	0	0	2	5
	3	15.310	1.054	2.0	1.054	2.0	1	48	2	20
VI.2	1	0.527		2.0					89	180
	2	0.514		2.0					89	180
	3	24.909		2.0					78	180

TABELA (V.2) - TEMPOS COMPUTACIONAIS E CONTADORES.

ALGORITMO GERAL:



CAPÍTULO VICONCLUSÕES

Os testes de redução utilizando informações do problema dual,  $\Delta_i^D$  e  $\Omega_i^D$ , apresentaram um bom desempenho o que justifica o emprego acentuado nos algoritmos propostos. Destacamos as demonstrações dos teoremas (1.1) e (2.1), até então apresentados de forma intuitiva. Como proposta ficaria a extensão desses resultados para o problema em dois níveis.

No capítulo III propomos um novo algoritmo guloso para o problema localização capacitado com custo fixo/linear. É bastante simples, rápido e confiável à medida que que aproximamos significativamente da solução ótima. Além disso, poderia ser estudado o acréscimo de uma outra etapa que procuraria elaborar trocas, incluindo e retirando facilidades a solução final.

Propomos a extensão do algoritmo guloso para funções custo fixo/côncavo devido a simplicidade e desempenho para o caso anterior. Pela análise de resultados, o algoritmo se comportou muito bem devido a grande dificuldade de se resolver esses problemas com um algoritmo exato, consumindo, as vezes, um tempo computacional bastante elevado.

No capítulo V, além da proposta de um novo limite inferior, resultou a pequena experiência com a técnica de RL e o método de subgradiente. Apesar da boa performance apresentada pela literatura, essas técnicas exigem um grande número de tes

tes computacionais para que possamos ajustar o algoritmo para resolução do problema específico. Nossos resultados são ainda escassos para maiores conclusões.

Finalmente ressaltaria o número de programas computacionais desenvolvidos, o uso de problemas já estudados pela literatura e a grande massa de resultados.

APÊNDICE AALGORITMO PROPOSTO POR MONTEROSSO

O problema tratado por MONTEROSSO (47) foi definido da seguinte maneira: dado um conjunto finito de  $\ell$  centros de produção,  $n$  centros de demanda, queremos escolher alguns dentre os  $m$  locais candidatos dados, localizando e dimensionando os armazéns correspondentes, de modo a minimizar custos de transporte e armazenagem.

O modelo matemático é da forma:

$$\min \sum_{i=1}^{\ell} \sum_{j=1}^m r_{ij} x_{ij} + \sum_{j=1}^m f_j (t_j) + \sum_{j=1}^m \sum_{k=1}^n s_{jk} z_{jk}$$

$$\text{s.a. } \sum_{j=1}^m x_{ij} = a_i, \quad i=1, \dots, \ell$$

$$(M1) \quad \sum_{i=1}^{\ell} x_{ij} - \sum_{k=1}^n z_{jk} = 0, \quad j=1, \dots, m$$

$$\sum_{j=1}^m z_{jk} = b_k, \quad k=1, \dots, n$$

$$x_{ij} \geq 0, \quad i=1, \dots, \ell \quad \text{e} \quad j=1, \dots, m$$

$$z_{jk} \geq 0, \quad j=1, \dots, m \quad \text{e} \quad k=1, \dots, n$$

onde  $x_{ij}$  = fluxo do centro de produção  $i$  ao armazém  $j$

$z_{jk}$  = fluxo do armazém  $j$  ao centro demanda  $k$

$a_i$  = produção no centro  $i$

$b_k$  = demanda no centro  $k$

$r_{ij}$  = custo unitário de transporte do centro produção  $i$  para o armazém  $j$

$s_{jk}$  = custo unitário de transporte do armazém  $j$  ao centro de demanda  $k$

$t_j = \sum_{i=1}^{\ell} x_{ij}$  = fluxo total passando pelo armazém  $j$

$f_j(t_j)$  = função côncava de custo de armazenagem em  $j$ .  
Inclui custo fixo caso exista.

Para este modelo apresenta um método iterativo em duas partes. Na primeira procura resolver o problema por sucessivas aproximações lineares das funções côncavas. Na segunda parte procura retirar, adicionar ou substituir armazéns selecionados por outros não selecionados na primeira etapa. Portanto, a convergência do algoritmo está associada a primeira parte e sobre a qual nos concentramos.

Em cada iteração  $k$  fixa os valores dos custos unitários de armazenagem  $c_j^k$  considerando-os constantes. A função de custo de armazenagem pode ser portanto representada por  $c_j^k t_j^k$ . A seguir, calcula os fluxos utilizando o algoritmo out-of-kilter de modo a minimizar o custo total. Na verdade, os custos unitários de armazenagem são uma função  $g_j(t_j)$  dos fluxos  $t_j$ . Diz que, numa iteração  $k$ , atingimos um estado de equilíbrio pa

ra o armazém  $j$  quando  $c_j^k = g_j(t_j^k)$ .

Como no início de uma iteração  $k$  ainda não se sabe os valores dos fluxos  $t_j^k$ , utiliza os fluxos  $t_j^{k-1}$  calculados na iteração  $(k-1)$  para determinação dos custos unitários. Isto é, faz-se  $c_j^k = g_j(t_j^{k-1})$ . Procede-se desta maneira até que seja atingido um estado de equilíbrio para todo armazém  $j$ , terminando assim a primeira parte do algoritmo.

Na verdade, considerando os custos unitários de armazenagem constantes, reduz-se o modelo (M1) a um modelo de rede de fluxo capacitada o qual permite suficiente flexibilidade. O novo modelo (M2) é dado por:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$(M2) \quad \sum_{j \in S(i)} x_{ij} - \sum_{j \in A(i)} x_{ji} = 0, \quad \forall i \in N$$

$$a_{ij} \leq x_{ij} \leq b_{ij}, \quad \forall (i,j) \in A$$

onde  $N = P \cup W \cup D \cup \{o\} \cup \{z\}$  = conjunto de nós, respectivamente, centros de produção, armazéns, centros de demanda, nó fonte e nó poço.

$A$  = conjunto de arcos da rede.

$c_{ij}$  = custo unitário no arco  $(i,j)$ .

$S(i)$  = conjunto de nós sucessores do nó  $i$ .

$A(i)$  = conjunto de nós antecessores do nó  $i$

$a_{ij}, b_{ij}$  = limites inferior e superior do fluxo.

Cada armazém  $j \in W$  é representado por um arco  $(W_j - W'_j)$ .

O algoritmo proposto consiste nos seguintes passos:

#### ALGORITMO A:

Passo 0. Faça  $k = 0$ . Atribua o mínimo valor possível a  $c_{ij}^k$  para todo  $(i, j)$  correspondente a  $j \in W$ .

Passo 1. Faça  $K = K+1$ .

Se  $K >$  "máximo número de iterações" então PARE.

Se não, utilize o algoritmo out-of-kilter para determinar os fluxos  $x_{ij}^k$  para todos os arcos  $(i, j)$ .

Considere os custos unitários  $c_{ij}^k$  para todo  $(i, j)$ .

Passo 2. Calcule novos valores  $c_{ij}^{K+1} = g_{ij}(x_{ij}^k)$  para todo  $(i, j), j \in W$ , correspondentes aos fluxos  $x_{ij}^k$  calculados no Passo 1.

Passo 3. Para todo  $(i, j)$  tal que  $j \in W$  calcule

$$\text{ERRO}_{ij}^k = \left| \frac{c_{ij}^{k+1} - c_{ij}^k}{c_{ij}^{k+1}} \right|$$

Se  $\text{ERRO}_{ij}^k \leq \text{TOLERÂNCIA}$  para todo  $(i, j), j \in W$  então PARE.

Se não volte ao Passo 1.



Mostraremos que este algoritmo converge para um ótimo local.

TEOREMA A.1. O algoritmo A é convergente.

Demonstração: será dividida em duas partes:

$$(i) f(x, z) = \sum_{i=1}^{\ell} \sum_{j=1}^m r_{ij} x_{ij} + \sum_{j=1}^m f_j(t_j) + \sum_{j=1}^m \sum_{k=1}^n s_{jk} z_{jk}$$

é não crescente ou  $f(x^{K+1}, z^{K+1}) \leq f(x^k, z^k)$

(ii) Se  $f(x^{K+1}, z^{K+1}) \leq f(x^k, z^k)$  então o algoritmo converge.

1ª PARTE: mostramos inicialmente que  $f(x, z)$  é uma função côncava. Com esta propriedade e a aplicação do algoritmo mostraremos que é não crescente.

A primeira e terceira parcelas de  $f(x, z)$  são lineares e portanto côncavas. As funções custo  $f_j(t_j)$ , associadas aos armazéns, são côncavas por definição. Logo, a soma de funções côncavas é uma função côncava. Por simplicidade de notação, utilizando o modelo correspondente a uma rede capacitada, seja  $f(x)$  a função côncava igual ao somatório das funções côncavas em cada arco:

$$f(x) = \sum_{(i,j) \in A} c_{ij}(x_{ij}) \quad (A.1)$$

Pela concavidade de  $f(x)$  podemos escrever que

$$f(x^{K+1}) \leq f(x^K) + \nabla^T f(x^K) (x^{K+1} - x^K) \quad (\text{A.2})$$

onde  $\nabla f(x)$  é o gradiente da função  $f(x)$ .

Logo, de (A.1) e (A.2) resulta

$$\begin{aligned} \sum_{(i,j) \in A} C_{ij}(x_{ij}^{K+1}) &\leq \sum_{(i,j) \in A} C_{ij}(x_{ij}^K) + \\ &+ \sum_{(i,j) \in A} \frac{dC_{ij}(x_{ij}^k)}{dx_{ij}} (x_{ij}^{K+1} - x_{ij}^k) \end{aligned} \quad (\text{A.3})$$

Ainda mais, como as funções  $C_{ij}(x_{ij})$  são crescentes temos:

$$\frac{C_{ij}(x_{ij}^k)}{x_{ij}^k} \geq \frac{dC_{ij}(x_{ij}^k)}{dx_{ij}} \geq 0, \quad \forall (i, j) \in A \quad (\text{A.4})$$

Em um passo  $k$  onde temos os fluxos  $x_{ij}^k$  em cada arco, o algoritmo determina  $x^{K+1}$  resolvendo o problema linear com a seguinte função objetivo:

$$\min \sum_{(i,j) \in A} \frac{C_{ij}(x_{ij}^k)}{x_{ij}^k} x_{ij}$$

Com isto, o ponto  $x^{K+1}$  é tal que

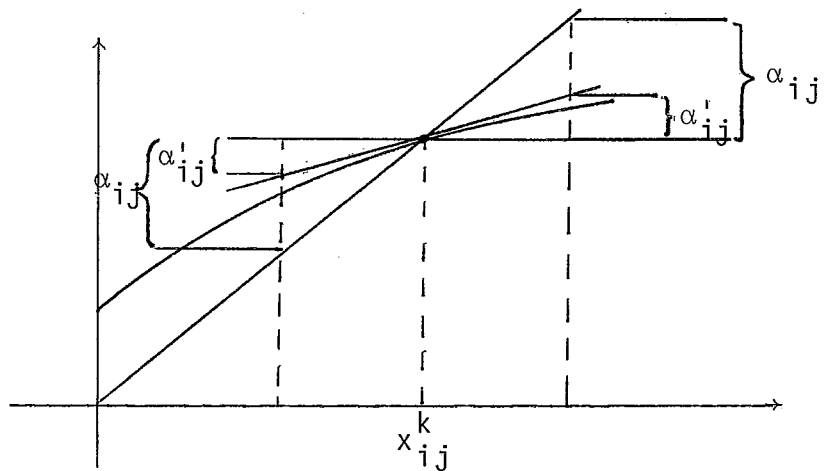
$$\sum_{(i,j) \in A} \frac{C_{ij}(x_{ij}^k)}{x_{ij}^k} (x_{ij}^{K+1} - x_{ij}^k) \leq 0 \quad (\text{A.5})$$

Utilizando o resultado (A.5) mostraremos também que

$$\sum_{(i,j) \in A} \frac{dC_{ij}(x_{ij}^k)}{dx_{ij}} (x_{ij}^{k+1} - x_{ij}^k) \leq 0$$

e portanto em (A.3) e (A.2) poderemos concluir que  $f(x^{k+1}) \leq f(x^k)$

Seja a figura abaixo:



Ao aplicar o algoritmo a partir de  $x^k$  temos que o fluxo em cada arco  $(i, j)$ ,  $x_{ij}^{k+1}$ , pode aumentar, diminuir ou não variar. Conforme figura acima, podemos caracterizar os elementos:

$$\alpha_{ij} = \frac{C_{ij}(x_{ij}^k)}{x_{ij}^k} (x_{ij}^{k+1} - x_{ij}^k) \quad (A.6)$$

$$\alpha'_{ij} = \frac{dC_{ij}(x_{ij}^k)}{dx_{ij}} (x_{ij}^{k+1} - x_{ij}^k) \quad (A.7)$$

De (A.6) e (A.7) resulta  $(x_{ij}^{k+1} - x_{ij}^k) = \alpha_{ij} / (C_{ij}(x_{ij}^k) / x_{ij}^k)$

$$(x_{ij}^{k+1} - x_{ij}^k) = \alpha'_{ij} / dC_{ij}(x_{ij}^k) / dx_{ij}$$

ou ainda

$$\alpha_{ij} = \frac{C_{ij}(x_{ij}^k)/x_{ij}^k}{dC_{ij}(x_{ij}^k)/dx_{ij}} \alpha'_{ij} \quad (\text{A.8})$$

Tomando o somatório para todos os arcos  $(i, j) \in A$  temos de (A.6) e (A.8) que:

$$\begin{aligned} \sum_{(i,j) \in A} \alpha_{ij} &= \sum_{(i,j) \in A} \frac{C_{ij}(x_{ij}^k)}{x_{ij}^k} (x_{ij}^{k+1} - x_{ij}^k) \\ &= \sum_{(i,j) \in A} \frac{C_{ij}(x_{ij}^k)/x_{ij}^k}{dC_{ij}(x_{ij}^k)/dx_{ij}} \alpha'_{ij} \end{aligned} \quad (\text{A.9})$$

Logo, de (A.5) e (A.9) podemos escrever:

$$\sum_{(i,j) \in A} \frac{C_{ij}(x_{ij}^k)/x_{ij}^k}{dC_{ij}(x_{ij}^k)/dx_{ij}} \alpha'_{ij} \leq 0 \quad (\text{A.10})$$

E, de (A.4) e (A.10) podemos concluir

$$\sum_{(i,j) \in A} \alpha'_{ij} \leq 0$$

ou em (A.7), 
$$\sum_{(i,j) \in A} \frac{dC_{ij}(x_{ij}^k)}{dx_{ij}} (x_{ij}^{k+1} - x_{ij}^k) \leq 0$$

que levado a (A.3), e conseqüentemente a (A.2) resulta

$$f(x^{k+1}) \leq f(x^k)$$

ou  $f(x)$  é uma função não crescente.

2ª PARTE: se  $f(x^{k+1}) \leq f(x^k)$  então o algoritmo converge.

Se  $f(x^{k+1}) < f(x^k)$  então temos  $x^{k+1} \neq x^k$ . Como  $x^{k+1}$  é resultado de um problema de programação linear,  $x^{k+1}$  é um novo vértice do conjunto de soluções viáveis. Uma vez que o número de vértices é limitado o algoritmo converge.

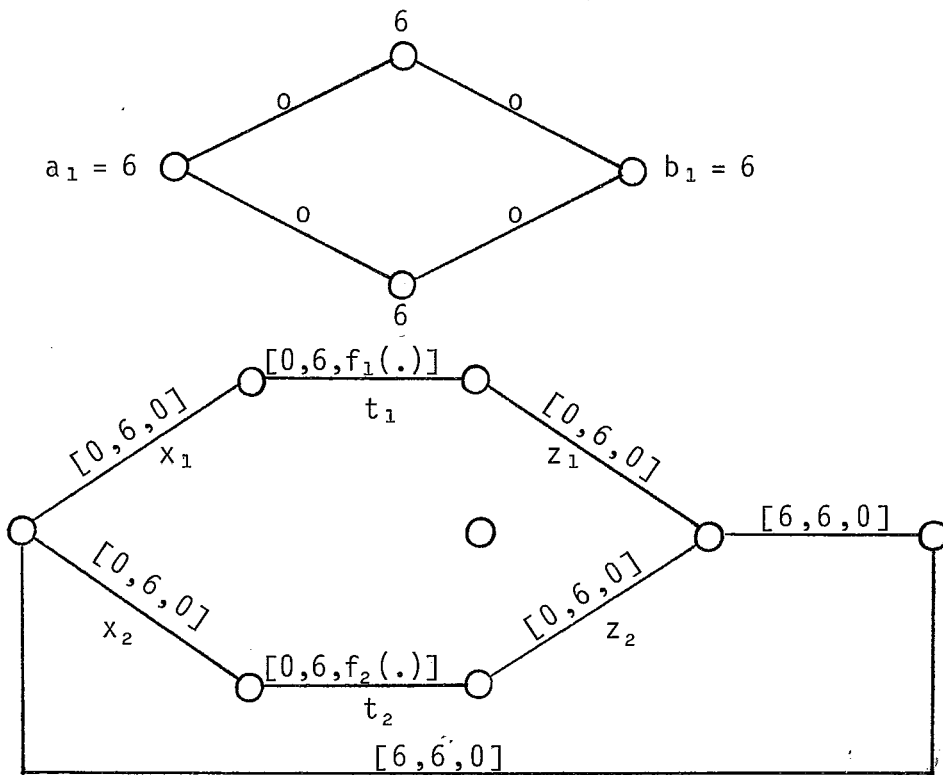
Se  $f(x^{k+1}) = f(x^k)$  então temos duas possibilidades:

- (i)  $x^{k+1} \neq x^k$ . Temos a geração de um novo vértice e é semelhante ao passo anterior. Porém, neste caso, o algoritmo pode ciclar. Isto pode ser contornado acrescentando a regra de parada um limite de iterações ou verificar se  $f(x^{k+1}) = f(x^k)$  ocorre repetidamente.
- (ii)  $x^{k+1} = x^k$ . Repetindo a mesma solução a regra de parada é atendida interrompendo o algoritmo. ●

A seguir mostraremos, através de um contra-exemplo, que o algoritmo converge para um ótimo local porém não global.

Seja o problema com um centro de produção e capacidade de 6 unidades e um centro de demanda com capacidade também de 6 unidades. Entre esses elementos estão dois locais candida

tos a localização de facilidades. Ambos com capacidade de 6 unidades sendo o custo de armazenamento dado pela função  $f_1(t_1) = 2 + 1/2 t_1$  para o primeiro e  $f_2(t_2) = 2 + t_2$  para o segundo. Os custos de transporte são todos nulos. Este problema é representado pelo grafo abaixo onde  $[a, b, c]$  são respectivamente as capacidades mínima e máxima e o custo associado a cada arco:



Denominando os fluxos de  $x_1$ ,  $x_2$ ,  $z_1$  e  $z_2$  e portanto  $t_1 = x_1$  e  $t_2 = x_2$ , podemos escrever o seguinte modelo matemático para o problema proposto:

$$\begin{aligned} \min Z &= 4 + 1/2 x_1 + x_2 \\ x_1 + x_2 &= 6 \\ x_1 - z_1 &= 0 \\ x_2 - z_2 &= 0 \\ z_1 + z_2 &= 6 \\ x_1, x_2, z_1, z_2 &\geq 0 \end{aligned}$$

Partindo com a solução inicial  $x^0 = (2, 4) = z^0$  e aplicando o algoritmo temos:

$$x_1^0 = 2 \Rightarrow f_1(t_1 = 2) = 3 \Rightarrow g_1(t_1) = 3/2$$

$$x_2^0 = 4 \Rightarrow f_2(t_2 = 4) = 6 \Rightarrow g_2(t_2) = 6/4$$

Logo, determinando o fluxo de custo mínimo para o modelo acima e função linear aproximada:

$$\min Z = g_1(t_1) x_1 + g_2(t_2) x_2 = 3/2 x_1 + 3/2 x_2 \quad (\text{A.11})$$

podemos atingir uma das soluções finais do algoritmo:

$$x = (6,0) \Rightarrow Z = 5 \quad \text{ou} \quad x = (0,6) \Rightarrow Z = 8$$

A segunda é uma solução de equilíbrio (ou final) do algoritmo e no entanto não é um ponto de mínimo global.

Para toda solução inicial tal que  $g_1(t_1) \geq g_2(t_2)$  em (A.11), o algoritmo caminha para a solução não ótima  $x = (0,6)$ .

APÊNDICE BALGORITMO PROPOSTO POR KELLY-KHUMAWALA

KELLY e KHUMAWALA (40) propuseram um algoritmo para o problema de localização de facilidades capacitado, em um nível e custos côncavos, seguindo o modelo (M3) apresentado no capítulo I:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i \left( \sum_{j \in J} x_{ij} \right) + \sum_{i \in I} F_i y_i$$

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

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

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

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

Em cada arco utiliza a aproximação linear da função, tal qual SOLAND (57), juntamente com os custos marginais obtidos pelas derivadas das funções  $f_i(\cdot)$ ,  $i \in I$ .

O algoritmo iterativo, descrito informalmente no capítulo I, resolve a cada iteração um problema de transporte e é dividido em duas etapas. Na primeira utiliza os custos de transporte mais os marginais. Na segunda introduz um centro de demanda artificial ( $D_2$ ) que compete com os centros de demanda reais. Em cada iteração pode alterar os custos em mais de um

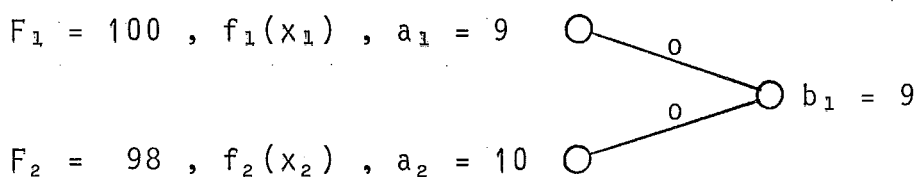


arco e não apenas em um como em SOLAND (57) e armazena somente a última solução.

Utilizando uma função especial definida sobre a função objetivo podemos mostrar a convergência do algoritmo seguindo as idéias da demonstração do algoritmo de MONTEROSSO (47), Apêndice A. O algoritmo é finito a medida que essa função especial decresce passo a passo. Porém, apesar de finito, a função objetivo pode crescer ao passar de uma etapa para outra.

É apenas uma heurística e não foi apresentado resultados computacionais. Nosso objetivo é mostrar um contra-exemplo em que o algoritmo converge para um ótimo local porém não global.

Seja o problema de localização com dois centros de oferta e capacidades  $a_1 = 9$  e  $a_2 = 10$ , e um centro de demanda com capacidade  $b_1 = 9$ . Os custos de transporte são nulos e os custos operacionais dados respectivamente pelas funções  $f_1(x_1) = \sqrt{x_1}$  e  $f_2(x_2) = 2\sqrt{x_2}$ . Os custos fixos são  $F_1 = 100$  e  $F_2 = 98$ .



A solução básica viável inicial  $x_{11} = 0$  ,  $x_{1D_1} = 9$  ,  $x_{21} = 9$  e  $x_{2D_1} = 1$  , e  $Z = 104$  ( $D_1$  é o centro de demanda artificial para equilibrar ofertas e demandas), é também a solução final para a

primeira etapa.

Entrando com o centro de demanda artificial  $D_2$ , iniciamos a segunda etapa. Fazendo a solução inicial para essa etapa semelhante a anterior:  $x_{11} = 0$ ,  $x_{1D_2} = 9$ ,  $x_{21} = 9$ ,  $x_{2D_2} = 1$ , o processo iterativo termina com a solução final:  $x_{11} = 0$ ,  $x_{1D_1} = 0$ ,  $x_{1D_2} = 9$ ,  $x_{21} = 9$ ,  $x_{2D_1} = 1$ ,  $x_{2D_2} = 0$  e  $Z = 104$ , correspondente a abrir a facilidade 2. No entanto, podemos verificar que se a facilidade 1 fosse aberta para atender a demanda  $b_1$  teríamos um custo  $Z = 103$ , inferior ao anterior.

APÊNDICE CTESTES DE REDUÇÃO COMPLEMENTARES

Estes testes consistem em explorar ainda mais os testes de redução apresentados no capítulo (II), aproveitando ao máximo as informações e cálculos já efetuados. Esta análise procurará forçar a abertura ou fechamento de alguma facilidade ainda indefinida.

Apresentamos uma extensão aos testes para abrir uma facilidade e uma outra para os testes para fechar.

1. EXTENSÃO AOS TESTES PARA ABRIR UMA FACILIDADE

Dos testes para abrir uma facilidade temos do capítulo (II):

$$\bar{\Delta}_i = \max_{0 \leq x'_{ij} \leq b_j} \left\{ \sum_{j \in J} \Delta_{ij} x'_{ij} \mid \sum_{j \in J} x'_{ij} \leq a_i \right\}$$

$$\text{onde } \Delta_{ij} = \min_{\ell \in (K_1 \cup K_2 - i)} \{ \max (C_{\ell j} - C_{ij}, 0) \}$$

$$\Delta_i^D = \sum_{j \in J \mid x_{ij}^* \neq 0} (v_j^j - C_{ij}) x_{ij}^*$$

$$\text{onde } v_j^j = \min_{\ell \in (K_1 \cup K_2 - i)} \{ C_{\ell j} - u_\ell^* \}$$

Conforme foi dito anteriormente, os testes a seguir apresentaram um baixo desempenho. Isto nos levou a fazer uma análise informal considerando apenas  $\bar{\Delta}_i$ , porém, os resultados são extensivos a  $\Delta_i^D$ .

Para  $i \in K_2$  seja  $\bar{\Delta}_i \geq F_i$ . Então pelo teorema (1.1) mostramos que a facilidade  $i$  será aberta. Como extensão procuraremos fixar quais os elementos  $j \in J$  que necessariamente serão atendidos por  $i$ . Isto implicaria em fazer  $C_{\ell j} = \infty$  para os demais elementos  $\ell \in (K_1 \cup K_2 - i)$ , contribuindo para o fechamento dessas facilidades em passos posteriores dentro de um processo iterativo em que trabalhamos com subconjuntos de  $(K_1 \cup K_2)$ . Todo desenvolvimento será feito comparando os custos de atender  $j$  por  $i$  ou por uma outra facilidade  $\ell \in (K_1 \cup K_2 - i)$ .

Para todo  $j \in J$ ,  $\Delta_{ij}$ , conforme definição acima, é a menor diferença positiva entre os custos variáveis para atender  $j$  via  $i$  ou por um outro elemento  $\ell \in (K_1 \cup K_2 - i)$ . Portanto se  $\Delta_{ij} > 0$  então  $C_{ij}$  é o menor custo variável para atender  $j$  e um  $C_{\ell j}$ ,  $\ell \in (K_1 \cup K_2 - i)$ , é o segundo menor.

Nossa análise está dividida em duas partes, considerando  $a_i \geq \sum_{j \in J} b_j$  para  $i \in K_2$  e a condição contrária.

(i) Seja  $a_i \geq \sum_{j \in J} b_j$  para  $i \in K_2$

Tudo se passa como se o problema fosse não capacitado. Uma vez que a facilidade  $i$  será aberta se  $\Delta_{ij} > 0$  então a demanda em  $j$  será atendida por  $i$  pelo menor custo. Na solução

ótima teremos  $x_{ij} = b_j$  e  $x_{\ell j} = 0$  para todo  $\ell \in (K_1 \cup K_2 - i)$ . Assim podemos fazer  $C_{\ell j} = \infty$ ,  $\ell \in (K_1 \cup K_2 - i)$ . Elevando os custos estaremos favorecendo os testes para fixar essas facilidades em fechado.

(ii) Seja  $a_i < \sum_{j \in J} b_j$  para  $i \in K_2$

Nesse caso para fixar o fluxo de  $i$  para algum(s)  $j \in J$ , é necessário estabelecer novas condições que serão descritas por etapas:

(a) Pela definição, se  $\Delta_{ij} > 0$  a demanda em  $j$  será atendida por  $i$  pelo menor custo. Ainda mais,  $\Delta_{ij}$  será constante se o segundo menor custo para atender  $j$ ,  $C_{\ell j}$ , corresponde a uma facilidade  $\ell \in K_1 \subseteq (K_1 \cup K_2 - i)$ . Pois, caso contrário, se  $\ell \in K_2$  então  $\Delta_{ij}$  poderá crescer desde que  $\ell$  venha a ser fechado e um novo  $\ell' \in (K_1 \cup K_2 - i)$  que atenderá  $j$  com o segundo menor custo implicará em  $C_{\ell' j} \geq C_{\ell j}$ .

(b) Ao calcular  $\bar{\Delta}_i$  somamos as mínimas economias obtidas ao atender totalmente ou parcialmente os elementos  $j \in J$  pela facilidade  $i$ . Se a soma desses ganhos superar o custo fixo  $F_i$  então  $i$  deve ser aberto. Isto está garantido pela condição  $\bar{\Delta}_i \geq F_i$ . Seja  $\Delta_{ik} = \max_{j \in J} \Delta_{ij}$ . Logo a primeira parcela do somatório para cálculo de  $\bar{\Delta}_i$  é dada por  $\Delta_{ik} x'_{ik}$ , onde  $x_{ik} = \min \{a_i, b_k\}$ . Se  $\Delta_{ik}$  for sempre o maior dos  $\Delta_{ij}$ , ou,  $\Delta_{ik} \geq \Delta_{ij}$  para todo  $j \in J$ ,  $j \neq k$ , em todos os passos subsequentes, ou para subconjuntos de  $(K_1 \cup K_2 - i)$ , essa será sem

pre a primeira parcela do somatório.

(c) De (a), se a facilidade  $\ell \in (K_1 \cup K_2 - i)$ , que atende um  $j \in J$  com o segundo menor custo, pertence a  $K_1$ ,  $\ell \in K_1$ , então  $\Delta_{ij}$  será sempre constante nos passos seguintes. Se isso ocorre para todos os  $\Delta_{ij}$ ,  $j \in J$ , então  $\Delta_{ik}$  será sempre o maior dos  $\Delta_{ij}$ ,  $j \in J$ . Caso ocorra  $\ell \notin K_1$ , para um  $\Delta_{ij}$ ,  $j \in J$ , podemos determinar um  $\ell' \in K_1$  tal que  $j$  seja atendido pelo segundo menor custo por um elemento de  $K_1$ , fornecendo um novo  $\Delta'_{ij} \geq \Delta_{ij}$ , ou seja, um limite superior para  $\Delta_{ij}$  nos passos subsequentes. Ainda assim, se  $\Delta_{ik}$  for o maior dos  $\Delta_{ij}$  ou  $\Delta'_{ij}$ , para todo  $j \in J$ ,  $j \neq k$ , então o será nos passos seguintes.

De (a), (b) e (c) se  $\Delta_{ik}$  for sempre o maior dos  $\Delta_{ij}$ ,  $j \in J$ , então a primeira parcela para o cálculo de  $\bar{\Delta}_i$  será sempre  $\Delta_{ik} \cdot x'_{ik}$ ,  $x'_{ik} = \min \{a_i, b_k\}$ .

Suponhamos que além de  $\bar{\Delta}_i \geq F_i$ , mais especificamente, também  $\Delta_{ik} \cdot x'_{ik} \geq F_i$ . Portanto é sempre justificável abrir a facilidade  $i$  mesmo que seja para atender apenas o elemento  $k \in J$ . Ainda mais, de (a), (b) e (c) podemos fixar o fluxo de  $i$  para  $k$ . Se  $a_i \geq b_k$  então  $x_{ik} = b_k$  e  $x_{\ell k} = 0$  ou  $C_{\ell k} = \infty$  para todo  $\ell \in (K_1 \cup K_2 - i)$ . Se  $a_i < b_k$  então  $x_{ik} = a_i$  e o complemento da demanda em  $k$  será atendido por alguma facilidade  $\ell \in (K_1 \cup K_2 - i)$ .

O que fizemos para o maior  $\Delta_{ij}$ ,  $j \in J$ , pode ser feito para o segundo maior, o terceiro, etc., até o menor  $\Delta_{ij} > 0$ .

## 2. EXTENSÃO AOS TESTES PARA FECHAR UMA FACILIDADE

Dos testes para fechar uma facilidade temos do capítulo II:

$$\bar{\Omega}_i = \max_{0 \leq x_{\ell j} \leq b_j} \left\{ \sum_{j \in J} (v_j^* - C_{ij}) x_{ij} \mid \sum_{j \in J} x_{ij} \leq a_i \right\}$$

$$\Omega_i^D = \max_{\ell \in K_1} \sum_{j \in J} \hat{x}_{\ell j} - \max(0, v_j^* - C_{ij})$$

$$\text{sujeito a } \sum_{\ell \in K_1} \sum_{j \in J} \hat{x}_{\ell j} \leq a_i, \hat{x}_{\ell j} \leq x_{\ell j}^*, \ell \in K_1, j \in J$$

De forma semelhante, consideramos a seguir apenas  $\bar{\Omega}_i$ , porém, os resultados são também extensivos a  $\Omega_i^D$ .

Anteriormente verificamos que  $v_j^*$  é não crescente em quanto o conjunto  $K_1$  cresce dentro de um processo iterativo. Logo,  $(v_j^* - C_{ij})$  é também não crescente com  $K_1$  crescente. Seja  $(v_j^* - C_{ij}) \leq 0$  para um certo  $K_1$ . Portanto ao calcular  $\bar{\Omega}_i$  na expressão acima teremos  $x_{ij} = 0$  em todos os passos subsequentes a medida que anexamos novos elementos a  $K_1$ . Assim podemos fazer  $C_{ij} = \infty$  para todo  $j \in J$  tal que  $(v_j^* - C_{ij}) \leq 0$ .

APÊNDICE DPROBLEMAS TESTES

O conjunto de seis problemas testes utilizados foram formalizados por KUEHN e HAMBURGER (41) e outros. São problemas bastante explorados pela comunidade científica para comparação do desempenho dos algoritmos estudados.

Conforme referência, são dados três conjuntos de fábricas (Indianapolis, Jacksonville, Indianapolis e Baltimore), 24 centros de armazenamento e 50 centros de demanda. As distâncias entre esses centros estão em "Rand McNally Cosmopolitan World Atlas" (Chicago, Rand McNally & Co., 1951), p. 193. Os custos de transporte foram obtidos multiplicando as distâncias acima por um custo unitário por unidade de distância. Essa razão é de \$ 0.0125/milha para as distâncias entre fábricas - centro de armazenamento, e de \$ 0.0250/milha para centros de armazenamento-centros de demanda.

São problemas em dois níveis, porém testados com apenas um, dos centros de armazenamento para os centros de demanda. Os custos do primeiro nível são adicionados aos do segundo. Com isso, o problema consiste em localizar um conjunto de armazéns, podendo, ou não, considerar as fábricas como centros de armazenamento de custo fixo nulo.

As ofertas são iguais para todas os armazéns, veja tabela (D.1). As demandas iguais a:



1-146	2-87	3-672	4-1337
5-31	6-559	7-2370	8-1089
9-33	10-32	11-5495	12-904
13-1466	14-143	15-615	16-564
17-226	18-3016	19-253	20-195
21-38	22-807	23-551	24-304
25-814	26-337	27-4368	28-577
29-482	30-495	31-231	32-322
33-685	34-12912	35-325	36-366
37-3671	38-2213	39-705	40-328
41-1681	42-1117	43-275	44-500
45-2241	46-733	47-222	48-49
49-1464	50-222		

cuja soma de 58.268 unidades.

Os custos fixos dos centros de armazenamento são to dos iguais, por<sup>ẽ</sup>m, para cada problema testamos 4 custos diferentes: \$ 7500, \$ 12500, \$ 17500 e \$ 25000.

O problema I tem 25 facilidades, sendo os 24 centros de armazenamento mais a f<sup>ã</sup>brica em Indianapolis, com custo fixo nulo.

O problema II tamb<sup>ẽ</sup>m tem 25 facilidades, considerando a f<sup>ã</sup>brica em Jacksonville. E para o III, com 26 facilidades, s<sup>ã</sup>o anexadas as f<sup>ã</sup>bricas em Indianapolis e Baltimore como locais candidatos a localiza<sup>ç</sup>o de um centro de armazenamento de custo fixo nulo.

Os problemas IV e V, de dimensões menores, (16 x 50), são obtidos do I, onde aparece apenas os 15 primeiros centros de armazenamento mais a fábrica em Indianapolis. E o VI é idêntico ao I a menos das ofertas.

Todos os testes computacionais foram executados em um computador Burroughs 6700.

GRUPO	NÚMERO PROBLEMAS	NÚMERO FORNECEDORES		NÚMERO CENTROS DEMANDA	OFERTAS	SOMA DAS DEMANDAS	CUSTOS FIXOS (\$ 1.000)
		ESTÁGIO I	ESTÁGIO II				
I	4	1	24 (25)	50	58.268	58.268	7.5/12.5/17.5/25.
II	4	1	25 (25)	50	58.268	58.268	7.5/12.5/17.5/25.
III	4	2	24 (26)	50	58.268	58.268	7.5/12.5/17.5/25.
IV	4	1	15 (16)	50	5.000	58.268	7.5/12.5/17.5/25.
V	4	1	15 (16)	50	15.000	58.268	7.5/12.5/17.5/25.
VI	4	1	24 (25)	50	5.000	58.268	7.5/12.5/17.5/25.

TABELA (D.1) - PROBLEMAS TESTES.

BIBLIOGRAFIA

- (1) AKINC, U. e KHUMAWALA, B. M., "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem", Man. Science, vol. 23, nº 6; pp. 585-594, (1977).
- (2) BABAYEV, D. A., "Comments on the Note of Frieze", Mathematical Programming, vol. 7, pp. 249-252, (1974).
- (3) BALINSKY, M. L., "On Finding Integer Solutions to Linear Programs", Mathematica, Princeton, N. J., (1964).
- (4) BALINSKY, M. L. e MILLS, H., "A Warehouse Problem", prepared for: Veterans Administration, Mathematica, Princeton, N. J., (1960).
- (5) BAUMOL, W. J. e WOLFE, P., "A Warehouse Location Problem", Operations Research, vol. 6, pp. 252-263, (1958).
- (6) BILDE, O. e KRARUP, J., "Sharp Lower Bounds and Efficient Algorithms for the Simple Plant Location Problem", Ann. Discrete Mathematics, vol. 1, pp. 79-97, (1977).
- (7) BORNSTEIN, C. T. e RUST, R., "Um Algoritmo Branch and Bound para Problemas de Localização com Custos Expressos por Funções Escada", XV Simpósio Brasileiro P. Operacional, Rio de Janeiro, (1982).

- (8) CHRISTOFIDES, N. e BEASLEY, J. E., "An Algorithm for the Capacitated Warehouse Location Problem", European J. O. Research, vol. 12, nº 1, (1983).
- (9) COOPER, L., "Heuristic Methods for Location-Allocation Problems", SIAM Review, vol. 6, nº 1, pp. 37-53, (1964).
- (10) CORNUEJOLS, G.; FISHER, M. L. e NEMHAUSER, G. L., "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms", Man. Science, vol. 23, pp. 789-810, (1977).
- (11) DAVIS, P. S. e RAY, T. L., "A Branch and Bound Algorithm for the Capacitated Facilities Location Problem", Naval Res. Logist. Quartely, vol. 16, nº 3, pp. 331-334, (1969).
- (12) EFROYMSON, M. A. e RAY, T. L. "A Branch and Bound Algorithm for Plant Location", Op. Research, vol. 14, pp. 361-368, (1966).
- (13) ELLWEIN, L. B., "Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints", Technical Report Nº 70.2, Dep. Industrial Engineering Stanford University, (1970).
- (14) ELLWEIN, L. B. e GRAY, P., "Solving Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints", AIIE Trans., vol. 3, nº 4, pp. 290-298, (1971).

- (15) ERLKOTTER, D., "Preinvestment Planning for Capacity Expansion: A Multi-Location Dynamic Model", Ph. D. Dissertation Stanford University, (1969).
- (16) ERLKOTTER, D., "A Dual-based Procedure for Uncapacitated Facility Location", Operations Research, vol. 26, n° 6, pp. 992-1009, (1978).
- (17) EVERETT, H. M., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", Operations Research, vol. 11, pp. 399-417, (1963).
- (18) FALK, J. E. e SOLAND, R. M., "An Algorithm for Separable Nonconvex Programming Problems", Man. Science, vol. 15, n° 9, pp. 550-569, (1969).
- (19) FELDMAN, E.; LEHRER, F. A. e RAY, T. L., "Warehouse Location Under Continuous Economies of Scale", Man. Science, vol. 12, n° 9, pp. 670-684, (1966).
- (20) FISHER, M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems", Man. Science, vol. 27, pp. 1-18, (1981).
- (21) FISHER, M. L.; NEMHAUSER, G. L. e WOLSEY, L. A., "An Analysis of Approximations for Maximizing Submodular Set Functions I", Math. Programming, vol. 14, pp. 265-294, (1978).

- (22) FISHER, M. L.; NORTHUP, W. D. e SHAPIRO, J. F., "Using Duality to Solve Discrete Optimization Problems: Theory and Computational Experience", Math. Programming Study, vol. 3, pp. 56-94, (1975).
- (23) FLORIAN, M. e ROBILLARD, P., "An Implicit Enumeration Algorithm for the Concave Cost Network Flow Problem", Man. Science, vol. 18, pp. 184-193, (1971).
- (24) FRANÇA, P. M. e GOMES, M. A., "Fluxo de Custo Mínimo em Redes de Custo Côncavo", XIV Simpósio Brasileiro de Pesquisa Operacional, Vitória, pp. 185-200, (1981).
- (25) FRANCIS, R. M. e GOLDSTEIN, J. M., "Location Theory: A Selective Bibliography", Operations Research, vol. 22, pp. 400-409, (1974).
- (26) GALVÃO, R. D., "The Optimal Location of Facilities on a Network", Ph. D. Thesis, Imperial College, London, (1977).
- (27) GEOFFRION, A. M., "Duality in Nonlinear Programming", SIAM Review, vol. 13, pp. 1-37, (1971).
- (28) GEOFFRION, A. M., "Lagrangian Relaxation for Integer Programming", Math. Programming Study 2: Approaches to Integer Programming, (1974).

- (29) GEOFFRION, A. M., "A Guide to Computer-Assisted Methods for Distribution Systems Planning", Sloan Man. Review, vol. 16, pp. 17-41, (1975).
- (30) GEOFFRION, A. M. e GRAVES, G. W., "Multicommodity Distribution System Design by Benders Decomposition", Man. Science, vol. 20, pp. 822-844, (1974).
- (31) GEOFFRION, A. M. e MARSTEN, R. E., "Integer Programming Algorithms: A Framework and State-of-the-Art Survey", Man. Science, vol. 18, n° 9, pp. 465-491, (1972).
- (32) GEOFFRION, A. M. e McBRIDE, R. "Lagrangean Relaxation Applied to Capacitated Facility Location Problems", AIIE Trans., vol. 10, pp. 40-47, (1978).
- (33) GOFFIN, J. L., "On Convergence Rates of Subgradient Optimization Methods", Math. Programming, vol. 13, pp. 329-347, (1977).
- (34) GUIGNARD, M. e SPIELBERG, K., "A Direct Dual Method for the Mixed Plant Location Problem with some Side Constraints", Math. Programming, vol. 17, pp. 198-228, (1979).
- (35) HELD, M. e KARP, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research, vol. 18, pp. 1138-1162, (1970).



- (36) HELD, M. e KARP, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", Math. Programming, vol. 1, pp. 6-25, (1971).
- (37) HELD, M.; WOLFE, P. e CROWDER, H. D., "Validation of Subgradient Optimization", Math. Programming, vol. 6, pp. 62-88, (1974).
- (38) HORST, R., "An Algorithm for Nonconvex Programming Problems", Math. Programming, vol. 10, pp. 312-321, (1976).
- (39) JACOBSEN, S. K., "Heuristics for the Capacitated Plant Location Model", European Jr. Op1. Research, vol. 12, pp. 253-261, (1983).
- (40) KELLY, D. L. e KHUMAWALA, B. M. "Capacitated Warehouse Location with Concave Costs", J. Op1. Res. Society, vol. 33, pp. 817-826, (1982).
- (41) KUEHN, A. A. e HAMBURGER, M. J., "A Heuristic Program for Locating Warehoures", Management Science, vol. 9, nº 4, pp. 643-666, (1963).
- (42) MACULAN, N.; CAMPELLO, R. E. e LOPES, L. B. R., "Relaxação Lagrangeana em Programação Inteira", Relatório Técnico COPPE/UFRJ, ES40-84, (1984).

- (43) MAHEY, P., "Subgradient Techniques and Combinatorial Optimization", School on Combinatorial Optimization, (1985).
- (44) MANNE, A. S., "Plant Location Under Economies of Scale", Management Science, vol. 11, nº 2, pp. 213-235, (1964).
- (45) MATEUS, G. R. e BORNSTEIN, C. T., "Um Algoritmo Guloso para o Problema de Localização Não Capacitado", Relatório Técnico do Programa de Sistemas e Computação, COPPE/UFRJ, ES06-81, (1981).
- (46) MILLER, C. E., "The Simplex Method for Local Separable Programming", In Recent Advances in Mathematical Programming, McGraw-Hill, New York, pp. 89-100, (1963).
- (47) MONTEROSSO, C. D. B., "Um Algoritmo Heurístico para a Localização e o Dimensionamento de Armazéns em Sistemas de Grande Porte e Considerando Economias de Escala", Tese de Mestrado, COPPE/UFRJ, (1977).
- (48) NAUSS, R. M., "An Improved Algorithm for the Capacitated Facility Location Problems", J. Operational Res. Society, vol. 29, pp. 1195-1202, (1978).
- (49) OLIVEIRA, A. A. F. e MATEUS, G. R., "Análise de Pior Caso do Algoritmo Guloso para Determinados Problemas de Localização Não-Capacitados", Relatório Técnico do Programa S. Computação, COPPE/UFRJ, ES66-85, (1985).

- (50) POLYAK, B. T., "A General Method for Solving Extremum Problems", Soviet Math. Dokl., vol. 8, pp. 593-597, (1967).
- (51) RIBEIRO, C. C., "Algorithms de Recherche de Plus Courts Chemins avec Contraintes: Etude Théorique, Implémentation et Parallélisation", Doct. Ing. Thesis, Paris, (1983).
- (52) RIBEIRO, C. C., Comunicação Pessoal, (1985).
- (53) ROSEN, J. B., "Global Minimization of a Linearly Constrained Concave Function by Partition of Feasible Domain", Math. Oper. Research, vol. 8, nº 2, pp. 215-230, (1983).
- (54) SÁ, G., "Branch-and-Bound and Aproximate Solutions to the Capacitated Plant Location Problem", Operations Research, vol. 17, pp. 1005-1016, (1969).
- (55) SHAPIRO, J. F., Mathematical Programming-Structures and Algorithms, John Wiley, (1979(a)).
- (56) SHAPIRO, J. F., "A Survey of Lagrangian Techniques for Discrete Optimization", Annals of Discrete Mathematics, vol. 5, pp. 133-138, (1979(b)).

- (57) SOLAND, R. M., "Optimal Facility Location with Concave Costs", Operations Research, vol. 22, n<sup>o</sup> 2, pp. 373-382, (1974).
- (58) TEITZ, M. B. e BART, P., "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph", Operations Research, vol. 16, n<sup>o</sup> 5, pp. 955-961, (1968).
- (59) THOAI, N. V. e TUY, H., "Convergent Algorithm for Minimizing a Concave Function", Math. Oper. Research, vol. 5, pp. 556-566, (1980).
- (60) TUY, H., "Concave Programming under Linear Constraints", Dokl. Akad. Nauk SSSR 159, pp. 32-35, English Translation in Soviet Math. Dokl., vol. 5, pp. 1437-1440, (1964).
- (61) VAN ROY, T. J., "A Cross Decomposition Algorithm for Capacitated Facility Location", Operations Research, to appear, (1983).
- (62) ZANGWILL, W. I., "Minimum Concave Cost Flows in Certain Networks", Management Science, vol. 14, n<sup>o</sup> 7, pp. 429-450, (1969).