

UNIVERSIDADE ESTADUAL DE CAMPINAS.  
FACULDADE DE ENGENHARIA ELÉTRICA.

UMA GENERALIZAÇÃO DO PROBLEMA  
DE SELEÇÃO DE VÉRTICES EM  
DIGRAFOS.

Autora : Arlene Fortunato Machado.

DCC. - IMECC. - UNICAMP.

Orientador : Prof. Dr. Clovis Perin Filho. *ñ*

DMA. - IMECC. - UNICAMP.

Trabalho apresentado à Comissão de Pós-Graduação da  
Faculdade de Engenharia Elétrica, como parte dos requisitos  
para a obtenção do título de Doutor em Engenharia Elétrica.

80/9110-4630

Este exemplar corresponde à redação final da tese  
defendida por Arlene Fortunato  
Machado e aprovada pela Comissão  
Julgadora em 29 / 4 / 91.

*[Handwritten Signature]*  
Orientador

**Ao meu marido Nelson e**

**a meus pais Armando e Deolinda.**

## AGRADECIMENTOS:

- ao Prof. Dr. Clovis Perin Filho, minha gratidão pela orientação, dedicação e esforço constante durante o desenvolvimento deste trabalho;
- ao Prof. Dr. Hermano de Medeiros Ferreira Tavares, pela amizade, orientação de programa e em especial pelo decisivo apoio na fase inicial de meu programa de doutorado;
- ao Prof. Dr. Miguel Taube Netto, pela amizade, sugestões e irrestrito apoio na fase inicial deste trabalho;
- ao Nelson, meu marido, pelo auxílio na parte gráfica e confecção dos desenhos constantes neste trabalho, além da paciência, compreensão e carinho, que ajudaram a superar os momentos difíceis;
- aos colegas do DCC.- IMECC. , pelo apoio e incentivo;
- e a todos que direta ou indiretamente, contribuíram na elaboração deste trabalho.

## RESUMO.

Este trabalho apresenta um estudo de uma generalização de um problema de seleção de vértices em digrafos e propõe a resolução deste problema através de métodos iterativos, em que em cada iteração, um problema de seleção é resolvido. A importância deste problema é devida ao seu relacionamento com alguns problemas clássicos de otimização (designação, b-emparelhamento bipartido de custo máximo, fluxo de custo mínimo).

É também, apresentado um estudo para um problema de seleção de vértices de um digrafo, estabelecendo relações entre este problema e o de b-emparelhamento máximo bipartido.

Algoritmos para os problemas de b-emparelhamento máximo bipartido, seleção de vértices e seleção de vértices generalizada são desenvolvidos. Os algoritmos apresentados para um mesmo problema são comparados entre si.

## ABSTRACT.

A generalization of a vertex selection problem is presented and a resolution of this problem using an iterative method is proposed. A vertex selection problem is solved in each iteration of this method. The importance of this problem lies on its relationship to some classical optimization problems (assignment, maximum cost bipartite b-matching, minimum cost flow).

A study of a vertex selection problem in a digraph is also presented. A relationship between this problem and the maximum bipartite b-matching problem is established.

Algorithms to solve the maximum bipartite b-matching, the vertex selection and the generalized vertex selection problems are developed and the algorithms for each of the problems are compared.

# ÍNDICE.

CAPÍTULO 1 : INTRODUÇÃO.....	1.1
1.1- Definições fundamentais.....	1.2
1.1.1- Definições básicas da teoria de programação linear.....	1.2
1.1.2- Noções básicas de teoria de algoritmos.....	1.3
1.1.3- Definições básicas de teoria de grafos.....	1.4
1.2- Propriedades básicas utilizadas neste trabalho.....	1.6
1.2.1- Propriedades fundamentais de programação linear.....	1.6
1.2.2- Propriedades básicas de programação linear inteira.....	1.7
1.3- Problema de seleção.....	1.8
1.4- Problema de seleção generalizada.....	1.10
1.5- Problema de b-emparelhamento bipartido.....	1.12
1.5.1- Problema de b-emparelhamento máximo.....	1.12
1.5.2- Problema de b-emparelhamento de custo máximo.....	1.14
1.6- Problemas de fluxo em um digrafo.....	1.15
1.6.1- Problema de fluxo máximo.....	1.15
1.6.2- Problema de caminho mínimo.....	1.17
1.6.3- Problema de fluxo de custo mínimo.....	1.18
CAPÍTULO 2 : UM PROBLEMA DE SELEÇÃO EM UM DIGRAFO.....	2.1
2.1- Problema de seleção.....	2.1
2.2- Transformação em um problema de seleção bipartido.....	2.4
2.3- Algoritmo I de resolução do problema de seleção.....	2.5
2.3.1- Algoritmo I de resolução do problema de seleção em $G = (V,E)$ .....	2.6
2.3.2- Exemplo de resolução do problema de seleção utilizando o Algoritmo I.....	2.6
2.3.3- Complexidade do Algoritmo I do problema de seleção.....	2.8
2.4- Transformação em um problema de b-emparelhamento bipartido.....	2.8
2.5- O problema de b-emparelhamento máximo bipartido.....	2.10

2.6- Algoritmo A de b-emparelhamento máximo bipartido. ....	2.12
2.6.1- Algoritmo A de b-emparelhamento máximo bipartido.....	2.12
2.6.2- Exemplo de resolução do b-emparelhamento máximo bipartido utilizando o Algoritmo A. ....	2.13
2.6.3- Validade e complexidade do Algoritmo A de b-emparelhamento máximo bipartido. ....	2.14
2.7- Algoritmo B de b-emparelhamento máximo bipartido.....	2.21
2.7.1- Introdução. ....	2.21
2.7.2- Algoritmo B de b-emparelhamento máximo bipartido. ....	2.23
2.7.3- Exemplo de resolução do problema de b-emparelhamento máximo bipartido utilizando o Algoritmo B. ....	2.23
2.7.4- Validade e complexidade do Algoritmo B de b-emparelhamento máximo bipartido. ....	2.25
2.8- Algoritmo II de resolução do problema de seleção. ....	2.31
2.8.1- Algoritmo II para determinar uma solução ótima do problema de seleção. ....	2.31
2.8.2- Exemplo de resolução do problema de seleção utilizando o Algoritmo II. ....	2.31
2.8.3- Complexidade do Algoritmo II do problema de seleção. ....	2.32
 CAPÍTULO 3 : PROBLEMA DE SELEÇÃO GENERALIZADA EM UM DIGRAFO. ....	 3.1
3.1- Uma generalização do problema de seleção. ....	3.1
3.2- O problema de b-emparelhamento bipartido de custo máximo.....	3.4
3.2.1- Definição e formulação do problema. ....	3.4
3.2.2- Obtenção de um b-emparelhamento bipartido de custo máximo a partir de uma seleção generalizada de custo máximo. ....	3.5
3.2.3- Obtenção de uma seleção generalizada bipartida de custo máximo a partir de um b-emparelhamento bipartido de custo máximo. ....	3.7
3.3- O problema de fluxo de custo mínimo em um digrafo. ....	3.10
3.3.1- Definição e formulação do problema. ....	3.10
3.3.2- Obtenção de um fluxo de custo mínimo a partir de uma seleção generalizada de custo máximo. ....	3.13
3.3.3- Obtenção de uma seleção generalizada de custo máximo a partir de um fluxo de custo mínimo. ....	3.15

3.4- Problema de determinação do cronograma de execução de um projeto composto de diversas tarefas. ....	3.17
3.4.1- Formulação do Problema. ....	3.18
3.4.2- Transformação em um problema de seleção generalizada. ....	3.19
3.4.3- Exemplo de determinação do cronograma de execução de um projeto através do problema de seleção generalizada. ....	3.20
 <b>CAPÍTULO 4 : RESOLUÇÃO DO PROBLEMA DE SELEÇÃO GENERALIZADA. ....</b>	<b>4.1</b>
4.1- Problema de seleção generalizada. ....	4.1
4.2- Validade do procedimento iterativo. ....	4.3
4.3- Algoritmo I de resolução do problema de seleção generalizada. ....	4.8
4.3.1- Algoritmo I de resolução do problema de seleção generalizada em um digrafo $G = (V,E)$ . ....	4.8
4.3.2- Exemplo de resolução do problema de seleção generalizada pelo Algoritmo I. ....	4.9
4.3.3- Complexidade do Algoritmo I. ....	4.11
4.4- Transformação em um problema de seleção generalizada bipartido. ....	4.16
4.5- Algoritmo II de resolução do problema de seleção generalizada. ....	4.22
4.5.1- Algoritmo II de resolução do problema de seleção generalizada. ....	4.22
4.5.2- Exemplo de resolução do problema de seleção generalizada pelo Algoritmo II. ....	4.23
4.5.3- Complexidade do Algoritmo II. ....	4.25
4.6- Técnica de escalonamento de um dos parâmetros do problema. ....	4.28
4.6.1- Introdução. ....	4.28
4.6.2- Algoritmo III para resolução do problema P de seleção generalizada. ....	4.29
4.6.3- Exemplo de resolução do problema de seleção generalizada pelo Algoritmo III. ...	4.30
4.6.4- Validade do Algoritmo III. ....	4.32
4.6.5- Complexidade do Algoritmo III. ....	4.34
 <b>CAPÍTULO 5 : COMPARAÇÕES ENTRE OS ALGORITMOS E CONCLUSÕES. ....</b>	<b>5.1</b>
5.1- Introdução. ....	5.1
5.2- Algumas comparações entre as complexidades dos algoritmos. ....	5.1
5.2.1- Problema de b-emparelhamento máximo bipartido. ....	5.1
5.2.2- Problema de seleção. ....	5.2
5.2.3- Problema de seleção generalizada. ....	5.3

5.3- Algumas comparações práticas entre os algoritmos.....	5.5
5.3.1- Problema de b-emparelhamento máximo bipartido.....	5.6
5.3.2- Problema de seleção.....	5.8
5.3.3- Problema de seleção generalizada. ....	5.12
5.4- Conclusões. ....	5.18
CAPÍTULO 6 : REFERÊNCIAS BIBLIOGRÁFICAS.....	6.1

## CAPÍTULO 1.

### INTRODUÇÃO.

Neste trabalho é apresentado um estudo de um problema de seleção de vértices em um digrafo (capítulo 2) e de uma generalização deste problema (capítulos 3 e 4).

Os problemas de seleção estudados, são problemas de programação inteira, onde as técnicas de programação combinatória são bastante utilizadas. Os modelos utilizados têm por finalidade auxiliar no processo de tomada de decisão em problemas com recursos limitados.

Nas secções 1.1 e 1.2 deste capítulo são apresentadas definições e propriedades básicas necessárias ao desenvolvimento deste trabalho.

Na secção 1.3 é feita uma apresentação preliminar do problema de seleção estudado, com algumas aplicações práticas do problema, e uma breve exposição histórica do desenvolvimento do problema é realizada.

Na secção 1.4 o mesmo é feito para a generalização do problema de seleção estudada.

Nas secções 1.5 e 1.6 são tratados, respectivamente os problemas de b-emparelhamento bipartido e os problemas de fluxo, sendo apresentadas breves exposições históricas do desenvolvimento de algoritmos para cada um dos problemas abordados.

Nas secções 1.5.1 e 1.6.1 são apresentados, respectivamente o problema de b-emparelhamento máximo e o problema fluxo máximo, devido aos algoritmos desenvolvidos para a resolução do problema de seleção estudado, utilizarem-se de algoritmos de resolução destes problemas.

Nas secções 1.5.2 e 1.6.2 são apresentados, respectivamente o problema de b-emparelhamento de custo máximo e o problema fluxo de custo mínimo, que foram aqui, também, incluídos, devido ao forte relacionamento existente entre estes problemas e a generalização do problema de seleção estudada (mostrado no capítulo 3).

Na secção 1.6.3 é apresentado o problema de caminho mínimo, devido aos algoritmos II e III desenvolvidos para a resolução do problema de seleção generalizada estudado utilizarem algoritmos de resolução do problema de caminho mínimo para efetuar uma modificação no digrafo do problema.

## 1.1- Definições fundamentais.

Nesta secção são encontradas as definições básicas utilizadas e não colocadas nos demais capítulos deste trabalho. Estas definições foram retiradas dos livros de Bondy e Murty [BdM76], Harary [Hrr72], Berge [Bge76], Hu[Hu-70], Murty [Mrt83], Lawler [Lwl76] e Brassard e Bratley [BsB88].

Ao leitor familiarizado com a literatura básica de programação linear, de complexidade de algoritmos, e da teoria de grafos é desnecessária a leitura desta secção.

### 1.1.1- Definições básicas da teoria de programação linear.

- Um problema  $P$  de programação linear pode sempre ser escrito na forma:

$$P: \quad \text{máx } f(x) = c^T \cdot x$$

$$\text{s.a. } A \cdot x \geq b$$

$$x_i \geq 0 \quad \text{para } i = 1, 2, \dots, n$$

onde  $x = (x_1, x_2, \dots, x_n)^T$ .

$A$  é uma matriz  $m$  por  $n$  e  $b$  e  $c$  são matrizes  $n$  por  $1$ .

A função  $f(x)$  é chamada **função objetivo** de  $P$ . O conjunto de inequações é chamado de **conjunto das restrições** de  $P$ .

- **Solução viável** de  $P$  é qualquer vetor  $x^0$  que satisfaz o conjunto das restrições de  $P$

- **Solução ótima** do problema  $P$  é qualquer solução viável  $x^*$  de  $P$ , tal que  $f(x^*) \geq f(x^0)$ , para toda solução viável  $x^0$  de  $P$ .

- Uma matriz  $A$  é **unimodular** quando o determinante de toda submatriz quadrada de ordem  $k$  é igual a  $0$ ,  $+1$ , ou  $-1$ ; onde  $k$  é o posto da matriz  $A$ .

- Uma matriz  $A$  é **totalmente unimodular** quando todo subdeterminante de  $A$  é igual a  $0$ ,  $+1$  ou  $-1$ .

(Como conseqüência, todo elemento de uma matriz totalmente unimodular é portanto  $0$ ,  $+1$  ou  $-1$ ).

### 1.1.2- Noções básicas de teoria de algoritmos.

**Complexidade de um algoritmo** é um processo analítico de medir a eficiência do algoritmo.

De início, até meados da década de 60, os critérios de avaliação de um algoritmo eram em geral empíricos, principalmente no que se refere a sua eficiência com relação a tempo. O algoritmo era descrito, um programa era implementado e executado em um computador para alguns conjuntos de dados de entrada diferentes, o tempo de execução para cada conjunto de dados era medido e através das curvas obtidas com os tempos de execução a avaliação era feita.

Embora, as informações obtidas por este critério sejam úteis, não permitem aferir o comportamento do algoritmo, são dependentes do ambiente computacional utilizado. Os tempos obtidos dependem dos dados particulares testados, da qualidade do programa, do compilador e do computador utilizados. É desejável fazer comparações entre algoritmos, para determinar qual o melhor para um certo conjunto de dados, independentemente do ambiente computacional utilizado.

Uma medida de um algoritmo de otimização combinatória é fornecida pela análise do pior caso, onde são contados o número de passos (operações) do algoritmo, em função dos dados da entrada de maneira a estabelecer um limite superior. Esta medida indica o número máximo de operações necessárias e é denominada complexidade do pior caso ou simplesmente complexidade.

**Notação O** é uma notação para indicar a complexidade de um algoritmo, com ênfase no seu comportamento assintótico e onde os termos não predominantes são desprezados. Desta forma, esta notação indica como cresce o número de passos, quando o conjunto de dados cresce para infinito (problemas grandes).

Dadas duas funções  $f(n)$  e  $g(n)$  do conjunto dos inteiros no conjunto dos reais positivos define-se que :

$f(n) = O(g(n))$ , quando existe uma constante positiva  $c$ , tal que para  $n$  suficientemente grande  $f(n) \leq c \cdot g(n)$ .

Por exemplo, se um algoritmo executa  $100 \cdot n^2 + 1000 \cdot n + 10000$  passos, sendo  $n$  um dado da entrada do problema, é dito que a complexidade deste algoritmo é  $O(n^2)$ . Um outro algoritmo, que execute  $n^2 + n \cdot \log_2 n$  passos, também possui complexidade  $O(n^2)$ ; indicando que ambos apresentam um comportamento semelhante, quando  $n$  tende para infinito.

### 1.1.3- Definições básicas de teoria de grafos.

- Um **grafo**  $G = (V, E)$  é um conjunto finito de vértices  $V$  juntamente com um conjunto finito  $E$  de arestas (pares não ordenados de vértices distintos de  $V$ ).

- **Grafo bipartido**  $G = (V, E)$  é um grafo em que o conjunto de vértices pode ser subdividido em dois conjuntos  $V_1$  e  $V_2$ , com  $V = V_1 \cup V_2$  e  $V_1 \cap V_2 = \emptyset$ ; tais que toda aresta de  $E$  é formada por um vértice de  $V_1$  e um vértice de  $V_2$ .

- Um grafo bipartido é **completo**, quando todo par  $\{v_1, v_2\}$  com  $v_1 \in V_1$  e  $v_2 \in V_2$  é uma aresta do grafo.

- **Grau de um vértice**  $v$  em um grafo  $G = (V, E)$  é o número de arestas de  $E$  que contêm  $v$ .

- Um **digrafo**  $G = (V, E)$ , também chamado de grafo dirigido, é um conjunto finito de vértices  $V$ , juntamente com um conjunto finito  $E$  de arcos (pares ordenados de vértices distintos de  $V$ ).

- Um digrafo (grafo)  $G$  com  $n$  vértices e  $m$  arcos (arestas) é chamado **digrafo denso** (grafo denso) quando  $m = O(n^2)$  e  $n = O(m^{1/2})$ ; e é chamado **digrafo esparso** (grafo esparso) quando  $m = O(n)$ .

- **Multigrafo** é um conjunto finito de vértices  $V$  juntamente com uma coleção finita  $E$  de arcos (pares ordenados de vértices distintos de  $V$ ).

(Em um multigrafo é possível haver dois ou mais pares iguais de vértices.)

- Um grafo (digrafo)  $G_1 = (V_1, E_1)$  é um **subgrafo** de um grafo (digrafo)  $G = (V, E)$ , quando  $V_1$  está contido em  $V$  e  $E_1$  está contido em  $E$ .

- Um **subgrafo maximal** de um grafo  $G$  com relação a uma propriedade  $p$  é um subgrafo de  $G$  que satisfaz  $p$  e não está contido propriamente em nenhum subgrafo de  $G$  que satisfaz  $p$ .

- **Caminho entre dois vértices**  $v_0$  e  $v_k$  de um grafo (digrafo)  $G = (V, E)$  é uma seqüência finita não vazia  $v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k$  cujos termos são alternadamente vértices e arestas (arcos) de  $G$ , com todos os elementos da seqüência diferentes entre si; e tais que  $v_i \in V$  para  $i=0, 1, \dots, k$ ,  $e_i = (v_{i-1}, v_i) \in E$  ( $e_i = (v_i, v_{i-1}) \in E$ ) para  $i=1, 2, \dots, k$ .

Um caminho é, em geral indicado pela seqüência de seus vértices.

- **Ciclo** em um grafo (digrafo)  $G = (V, E)$  é uma seqüência finita não vazia  $v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_0$  cujos termos são alternadamente vértices e arestas (arcos) de  $G$ , com todos os vértices da seqüência diferentes entre si, com exceção do primeiro e último vértices que são iguais; e tais que  $v_i \in V$  para  $i=0, 1, \dots, k-1$ ,  $e_i = (v_{i-1}, v_i) \in E$  ( $e_i = (v_{i-1}, v_i) \in E$ ) para  $i=1, 2, \dots, k-1$ , e  $e_k = (v_{k-1}, v_0) \in E$  ( $e_k = (v_{k-1}, v_0) \in E$ ).

- **Comprimento de um caminho** é o número de arestas (arcos) do caminho.

- **Peso de uma aresta** (arco) é um número real associado a cada aresta (arco) de  $G$ .

A definição de comprimento de um caminho pode, eventualmente ser estendida para a soma dos pesos de suas arestas (arcos).

- Um vértice  $v_i$  é dito **sucessor de um vértice**  $v_j$  em  $G$ , quando existe em  $G$  um caminho de  $v_j$  para  $v_i$ .

- Um vértice  $v_i$  é dito **antecessor de um vértice**  $v_j$  em  $G$ , quando existe em  $G$  um caminho de  $v_i$  para  $v_j$ .

- **Busca em um grafo** (digrafo) é um processo de determinação de um subgrafo deste, a partir de um vértice denominado raiz.

- A busca é chamada **busca em profundidade** ("depth-first search") quando o subgrafo é determinado da seguinte maneira :

I) A raiz é colocada no subgrafo.

II) Seja  $P_1$  uma pilha (último a entrar é o primeiro a sair) de vértices contendo, inicialmente, só a raiz.

III) Enquanto  $P_1$  não está vazia executar : se existe uma aresta (arco) entre o vértice  $v$  (topo de  $P_1$ ) e um vértice  $v_1$  não pertencente ao subgrafo, então esta aresta (arco) e este vértice  $v_1$  são introduzidos no subgrafo, além disto, o vértice  $v_1$  é introduzido no topo de  $P_1$ ; caso contrário, se não existe uma aresta (arco) entre o vértice  $v$  (topo de  $P_1$ ) e um vértice  $v_1$  não pertencente ao subgrafo, então  $v$  (topo de  $P_1$ ) é retirado de  $P_1$ .

IV) Parar, a busca está completa.

- A busca é chamada **busca em largura** ("breadth-first search") quando o subgrafo é

determinado da seguinte maneira :

I) A raiz é colocada no subgrafo.

II) Seja  $L_1$  uma fila (primeiro a entrar é o primeiro a sair) de vértices contendo, inicialmente, só a raiz.

III) Enquanto  $L_1$  não está vazia executar : retirar de  $L_1$  o primeiro vértice  $v$ ; se existem arestas (arcos) entre o vértice  $v$  e vértices não pertencentes ao subgrafo (seja  $V_1$  o conjunto destes vértices), então estas arestas (arcos) e o conjunto  $V_1$  de vértices são introduzidos no subgrafo, além disto, o conjunto  $V_1$  de vértices é introduzido no fim de  $L_1$ .

IV) Parar, a busca está completa.

## 1.2- Propriedades básicas utilizadas neste trabalho.

Nesta secção são encontradas cinco propriedades básicas utilizadas e não colocadas nos demais capítulos deste trabalho. Estas propriedades foram retiradas dos livros de Lawler [Lwl76], Lovasz e Plummer [LvP86], Hu [Hu-70] e Murty [Mrt83].

Ao leitor familiarizado com as propriedades básicas de programação linear, programação inteira e de teoria de otimização é desnecessária a leitura desta secção.

### 1.2.1- Propriedades fundamentais de programação linear.

As três propriedades colocadas a seguir encontram-se demonstradas, por exemplo em Lawler [Lwl76]. Sejam  $P$  e  $P_1$ , um par de problemas duais, apenas para facilitar a notação, com a forma :

$$\begin{array}{ll}
 P : \min c^T \cdot x & P_1 : \max y^T \cdot b \\
 \text{s.a} : A \cdot x \geq b & \text{s.a} : y^T \cdot A \leq c \\
 x \geq 0 & y \geq 0
 \end{array}$$

- **Dualidade fraca** : Se  $x^0$  é uma solução viável de um problema linear primal  $P$  e  $y^0$  é uma solução viável do problema dual  $P_1$  então  $c^T \cdot x^0 \geq y^{0T} \cdot b$ .

- **Dualidade forte** : Se um de um par de problemas duais tem uma solução ótima finita então o outro problema também tem e as funções objetivos têm o mesmo o valor ótimo. Se um destes problemas tem um ótimo não limitado, o outro não tem solução viável.

- **Ortogonalidade de soluções de problemas duais** : Se  $x^0$  é uma solução viável de um problema linear primal  $P$  e  $y^0$  é uma solução viável do problema dual  $P_1$  então  $x^0$  e  $y^0$  são soluções ótimas de  $P$  e  $P_1$  se e somente se  $(y^{0T} \cdot A - c)^T \cdot x^0 = 0$  e  $y^{0T} \cdot (A \cdot x^0 - b) = 0$ .

**Condição suficiente para a equivalência do conjunto de soluções ótimas de dois problemas de programação linear  $P$  e  $P_0$  com mesma função objetivo :**

- Uma condição suficiente para que dois problemas de programação linear  $P$  e  $P_0$  com mesma função objetivo tenham mesmo conjunto de soluções ótimas é :

se  $x^*$  é solução viável de  $P$  então  $x^*$  é solução viável de  $P_0$  e

se  $x^*$  é solução ótima de  $P_0$  então  $x^*$  é solução viável de  $P$ .

Com a primeira das afirmações é demonstrado que o conjunto das soluções viáveis de  $P$  está contido no conjunto das soluções viáveis de  $P_0$ . Com a segunda, que o conjunto de soluções ótimas de  $P_0$  está contido no conjunto de soluções viáveis de  $P$ .

Logo, qualquer  $x^*$  solução ótima de  $P_0$  é solução viável de  $P$  e como os dois problemas têm mesma função objetivo e toda solução viável de  $P$  é solução viável de  $P_0$  conclue-se que  $x^*$  é solução ótima de  $P$ .

Portanto, qualquer  $x^*$  solução ótima de  $P$  é solução viável de  $P$  e, portanto solução viável de  $P_0$ , como os dois problemas têm mesma função objetivo e toda solução ótima de  $P_0$  é solução viável de  $P$  conclue-se que  $x^*$  é solução ótima de  $P_0$ .

### 1.2.2- Propriedades básicas de programação linear inteira.

**Relação entre a matriz dos coeficientes das restrições de um problema de programação linear ser totalmente unimodular e a sua solução ótima ser inteira :**

A primeira das duas propriedades colocadas a seguir, estabelece uma condição suficiente para que uma matriz seja totalmente unimodular. A segunda estabelece uma condição suficiente para que as soluções básicas de um problema de programação linear sejam inteiras. Estas propriedades são encontradas, por exemplo em Lovász e Plummer [LvP86].

- Se a matriz dos coeficientes das restrições só contém elementos pertencentes ao conjunto

de números  $\{0, +1, -1\}$ , em nenhuma linha da matriz existem mais que dois elementos diferentes de zero e as colunas da matriz podem ser divididas em duas componentes, tais que elementos diferentes de zero de uma mesma componente são diferentes e de componentes diferentes são iguais então a matriz de restrições é totalmente unimodular (Teorema de Heller, Tompkins e Gale).

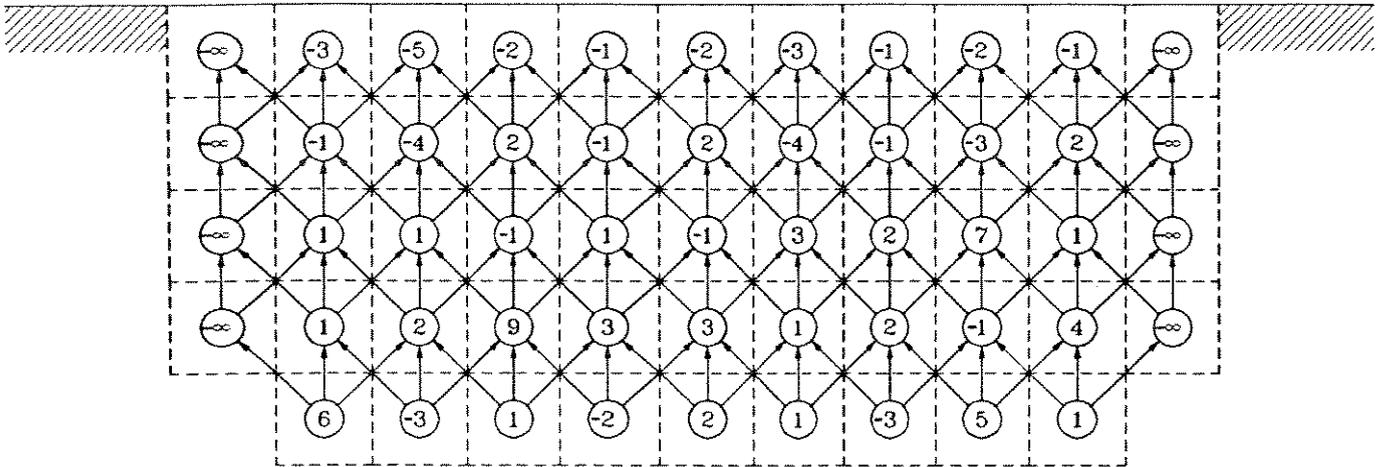
- Se a matriz dos coeficientes das restrições de um problema de programação linear  $P$  é totalmente unimodular e os dados constantes de  $P$  são inteiros, então qualquer solução básica de  $P$  é inteira (Teorema de Hoffman e Kruskal).

### 1.3- Problema de seleção.

O problema de seleção de vértices estudado é um problema de otimização combinatória (programação 0,1). Dado um digrafo em que a cada vértice está associado um valor inteiro, o problema consiste em determinar um subconjunto do conjunto de vértices do digrafo de valor máximo, com a propriedade que se um vértice pertence ao subconjunto selecionado todos os seus sucessores no digrafo também pertencem.

O problema de seleção de vértices foi inicialmente discutido por Lerchs e Grossmann [LcG65] em seu trabalho sobre exploração de minas a céu aberto. O planejamento de um programa de mineração envolve o projeto da forma final da escavação a ser feita. As reservas de minério e sua distribuição espacial são consideradas estimadas através de interpretações geológicas do terreno. Neste trabalho foram considerados conhecidos, para cada ponto do solo: o tipo do material, seu valor mineral, e o custo de extração daquele tipo de minério. Foram também, consideradas restrições quanto à inclinação máxima das paredes do buraco. A forma final do buraco a ser escavado é determinada de maneira que o lucro (valor do material - custo da extração) seja máximo.

Exemplo de um corte vertical de um terreno discretizado com indicação do valor líquido de cada ponto.



Foi considerado que cada ponto do solo só pode ser extraído se todos os pontos diretamente acima dele, na vertical ou nas diagonais, também o são. Os pontos com valor igual a  $-\infty$  servem para delimitar a área que pode ser retirada e para forçar uma dada inclinação das paredes do buraco a ser excavado.

Este problema foi representado em uma série de digrafos, um para cada corte vertical do terreno. Foi apresentado um algoritmo que resolvia, separadamente cada corte vertical do terreno, e a seguir combinava estas soluções, de maneira a suavizar as curvas entre dois cortes. Esta última etapa apresentava diversas dificuldades.

Rhys [Rhy70] discutiu o problema para o caso de custos fixos compartilhados e como motivação apresentou o problema de construção de terminais (bancários ou de transporte de carga ou passageiros): o custo para operação de cada terminal é fixo e conhecido, sendo que a existência de um par de terminais permite o estabelecimento de um serviço entre eles e este serviço dá um lucro conhecido. O problema é determinar uma seleção ótima de terminais a construir de forma a maximizar o lucro líquido. Este problema foi formulado e resolvido como um problema de fluxo máximo em um digrafo bipartido (de um lado da bipartição de vértices os serviços disponíveis e do outro lado da bipartição os terminais necessários).

Balinski [Blk70] fez uma generalização teórica do problema apresentado por Rhys, considerando o mesmo problema, mas com a condição, que para a existência de um serviço, é necessária a construção de um número finito de terminais. Balinsky estabeleceu também, para problemas de seleção em digrafos bipartidos, a relação entre a seleção de valor máximo e o problema de corte mínimo.

Picard [Pic76] generalizou o conceito de problema de seleção, definindo o problema de fecho ótimo em um digrafo qualquer. O problema de seleção foi apresentado, então como o problema do fecho ótimo em um digrafo bipartido. Apresentou o problema do contorno ótimo de uma mina a céu aberto formulado, como um problema de fecho ótimo. Demonstrou ainda, que o problema pode ser resolvido, através do problema de corte mínimo (fluxo máximo) em um digrafo. O digrafo utilizado por Picard é descrito no capítulo 2 secção 2.1.

No capítulo 2, deste trabalho, são propostos dois novos algoritmos para determinar uma seleção ótima de vértices. A base destes algoritmos é uma transformação do problema de seleção de vértices de um digrafo qualquer em um problema de seleção de vértices de um digrafo bipartido.

O primeiro, após a transformação do digrafo em um digrafo bipartido, utiliza o resultado de Picard, resolvendo o problema de seleção bipartido, através de um problema de fluxo máximo em um digrafo bipartido.

O segundo algoritmo, após a transformação do digrafo em um digrafo bipartido, resolve o problema de seleção bipartido através da resolução do problema de b-emparelhamento máximo em um grafo bipartido. Este problema é também chamado problema de determinação do subgrafo máximo restrito em grau de um multigrafo.

Ainda no capítulo 2, é apresentado um estudo da complexidade dos dois algoritmos de resolução do problema de seleção. No cálculo da complexidade do primeiro algoritmo é utilizado o resultado do trabalho de Gushfield, Martel e Fernandez-Bacca [GMF87] sobre a complexidade do algoritmo de Karzanov [Kzn74] para o caso particular do problema de fluxo máximo em digrafo bipartido.

No capítulo 5, é realizada uma comparação teórica dos algoritmos apresentados no capítulo 2, através de comparações entre suas complexidades. É também, feita uma comparação prática dos tempos de execução destes algoritmos, para problemas gerados aleatoriamente.

#### **1.4- Problema de seleção generalizada.**

É uma generalização do problema de seleção de vértices estudado no capítulo 2. Neste caso, em vez de programação 0,1, para determinar se um vértice deve ou não ser selecionado, foi considerada a programação inteira, para determinar em que quantidade cada vértice deve ser selecionado, para que o valor da seleção seja máximo; sendo que a cada vértice está associado

um valor inteiro e a cada arco está associado um valor inteiro não negativo, e que a seleção de uma certa quantidade de um vértice implica em um limitante superior (valor do arco + quantidade selecionada deste vértice), para a quantidade, em que cada um de seus antecessores pode ser selecionado.

No capítulo 3 são apresentadas, além da definição do problema de seleção generalizada, relações entre as soluções ótimas deste problema e as do problema de b-emparelhamento de custo máximo ou as do problema de fluxo de custo mínimo. Para o estabelecimento destas relações é utilizado o Teorema de Hoffman e Kruskal e a teoria de dualidade de programação linear.

Uma aplicação prática do problema de seleção generalizada, também, colocada no capítulo 3, é o problema de determinação de um cronograma ótimo de execução de um projeto composto de tarefas, quando entre estas tarefas existe uma relação de dependência.

No capítulo 4 são propostos três novos algoritmos para a resolução do problema de seleção generalizada. Um estudo da complexidade destes algoritmos é realizado.

Os três algoritmos apresentados são algoritmos iterativos, em que a cada iteração um problema de seleção é resolvido.

O primeiro algoritmo utiliza o digrafo do problema, e por isto, é teoricamente mais simples.

O segundo algoritmo, modifica o problema, transformando-o em um problema de seleção generalizada em um digrafo bipartido completo, para a seguir resolvê-lo através do primeiro algoritmo.

O terceiro algoritmo, além da transformação do digrafo utilizada pelo segundo algoritmo, utiliza, o conceito de escalonamento de um dos parâmetros do problema introduzido por Edmonds e Karp [EdK72] (para o problema de fluxo de custo mínimo).

Os três algoritmos, resolvendo em cada iteração um problema de seleção, têm suas complexidades dependentes do algoritmo utilizado na resolução dos problemas de seleção em um digrafo (capítulo 2).

Os três algoritmos são algoritmos do tipo primal-dual. No primeiro a solução ótima de  $P$  é aproximada por soluções viáveis do primal, isto é, a viabilidade do primal é mantida em todas as iterações. No segundo e no terceiro, a viabilidade do primal não é mantida em todas as iterações, porém é mantida a viabilidade do dual.

No capítulo 5, é realizada uma comparação teórica dos algoritmos apresentados no capítulo 4, através de comparações entre suas complexidades. É também, feita uma comparação prática dos tempos de execução destes algoritmos, para problemas gerados aleatoriamente.

### 1.5- Problema de b-emparelhamento bipartido.

Esta secção é composta pelos problemas de b-emparelhamento máximo e pelo problema de b-emparelhamento de custo máximo. O primeiro problema é um caso particular do segundo (custos dos arcos iguais a um), e está relacionado com o problema de seleção de vértices estudado, enquanto o segundo problema está relacionado com a generalização do problema de seleção de vértices estudada.

#### 1.5.1- Problema de b-emparelhamento máximo.

É um problema de programação inteira (estudado no capítulo 2, secção 2.5), sendo às vezes denominado de problema de determinação de um subgrafo máximo restrito em grau de um multigrafo, e é uma generalização do problema de emparelhamento máximo (problema de programação 0,1).

O trabalho de Edmonds [Edm65] foi o primeiro a ser publicado com um estudo do problema de emparelhamento máximo em grafos. Este trabalho apresentava um algoritmo para a resolução do problema em um grafo qualquer, e introduzia o conceito de contração (“shrink”) de subgrafos do grafo dado em um pseudo-vértice, em alguns passos do algoritmo.

Berge [Bge76] estudou o problema de b-emparelhamento máximo como um problema de programação 0,1 em grafos ou multigrafos. É de sua autoria um resultado, citado em [Bge76], mas publicado inicialmente em um trabalho de 1957, que caracteriza o b-emparelhamento máximo. Este resultado é o corolário 2.1 do capítulo 2 de nosso trabalho. Para resolução do problema, Berge propôs uma transformação do multigrafo em um grafo, aumentando consideravelmente, o número de vértices e o número de arcos envolvidos no problema.

Edmonds e Johnson [EdJ73] estudaram a relação entre este problema e o problema do carteiro chinês (caminho mínimo para percorrer as ruas de uma cidade), que é mais simples que o problema do caixeiro viajante.

Lovász e Plummer [LvP86] apresentaram o caso particular do 2-emparelhamento, estabelecendo uma relação entre este problema e o problema do carteiro chinês; este trabalho foi inicialmente publicado por Lovász em 1975 e a transformação utilizada difere da apresentada por Edmonds e Johnson. Em seu estudo sobre o caso geral, consideraram o problema de  $b$ -emparelhamento definido em um grafo e apresentaram uma modificação do grafo, diferente da apresentada por Berge, para estabelecer uma correspondência entre o problema de  $b$ -emparelhamento perfeito e o problema de emparelhamento perfeito. Esta modificação também aumenta muito o número de vértices e o número de arestas do problema.

Pulleybank [Plb80] caracterizou para que tipos de grafos a solução do dual do problema do  $b$ -emparelhamento é garantida ser inteira. Seu estudo inclui, também o problema de  $b$ -emparelhamento de custo mínimo.

O trabalho de Hoffman e Kruskal (citado, por exemplo em Lovász e Plummer [LvP86]) estabeleceu em que condições da matriz dos coeficientes do conjunto de restrições de um problema de programação linear é garantido que suas soluções básicas são inteiras.

Gabow [Gbw85] apresentou um algoritmo para o problema de  $b$ -emparelhamento máximo em um multigrafo bipartido (apresentado como o problema de determinação do subgrafo máximo restrito em grau). Em seu algoritmo o problema foi primeiro transformado em um problema de emparelhamento e a técnica de escalonamento do maior parâmetro (número de vértices, número de arestas) do problema foi utilizada.

Gabow e Tarjan [GbT88] apresentaram um algoritmo para o problema de  $b$ -emparelhamento máximo em um multigrafo, com escalonamento do maior parâmetro do problema. Foi feito um estudo da complexidade deste algoritmo utilizando processamento paralelo.

Hopcroft e Karp [HpK73] apresentaram um estudo muito completo do problema de emparelhamento bipartido, com um algoritmo de resolução, que utiliza o conceito de referente introduzido por Dinic [Dnc70] em seu algoritmo para o problema de fluxo máximo. A complexidade do algoritmo de Hopcroft e Karp é considerada a melhor até o momento.

Even e Tarjan [EvT75] discutiram a complexidade do algoritmo de fluxo máximo de Dinic [Dnc70] para um tipo especial de rede (redes com vértices com capacidade um ou redes com arcos com capacidade um) e o resultado por eles obtido tem sido amplamente utilizado para o cálculo da complexidade de algoritmos para o problema do emparelhamento bipartido.

No capítulo 2, são estabelecidas relações entre o problema de seleção em um digrafo qualquer e o problema de seleção em um digrafo bipartido e entre o problema de seleção em um digrafo bipartido e o problema de b-emparelhamento máximo bipartido. Para o estabelecimento desta última relação é utilizado o Teorema de Hoffman e Kruskal e a teoria de dualidade de programação linear. Devido a esta relação entre os dois problemas, é realizado um estudo sobre o problema de b-emparelhamento máximo bipartido, e são propostos dois novos algoritmos para a resolução deste problema.

O primeiro algoritmo apresentado é uma generalização do algoritmo de Hopcroft e Karp para o problema de emparelhamento máximo bipartido, e utiliza, também a noção de referente introduzida por Dinic [Dnc70]. Na prova da complexidade deste algoritmo é feita uma generalização de resultados obtidos por Hopcroft e Karp e também de resultados obtidos por Even e Tarjan.

O segundo algoritmo apresentado utiliza o conceito de escalonamento de um dos parâmetros do problema, que foi introduzido por Edmonds e Karp [EdK72] para o problema de fluxo de custo mínimo. O parâmetro utilizado para o escalonamento foi o máximo dos valores de  $b$ .

Alguns autores, Lawler [Lwl76] entre eles, consideram que a técnica de escalonamento, embora de fácil implementação, é provavelmente, de importância limitada. Sua relevância, ainda segundo estes autores, situa-se na área de Teoria de Computação, área em que fornece resultados muito satisfatórios.

### 1.5.2- Problema de b-emparelhamento de custo máximo.

É um problema de programação inteira (formulado no capítulo 3, secção 3.2), sendo às vezes denominado de problema de determinação de um subgrafo de custo máximo restrito em grau de um multigrafo, e é uma generalização do problema de emparelhamento de custo máximo, também conhecido como problema de designação (problema de programação 0,1).

Berge [Bge76] estudou o problema de b-emparelhamento de custo máximo como um problema de programação 0,1 em grafos ou multigrafos. Para resolução do problema foi proposta a transformação do multigrafo em um grafo, aumentando consideravelmente, o número de vértices e o número de de arcos envolvidos no problema, e a resolução do problema de designação de custo máximo resultante.

Gabow [Gbw85] apresentou um algoritmo para o problema de b-emparelhamento de custo máximo em um multigrafo bipartido (apresentado como o problema de determinação do subgrafo de custo máximo restrito em grau). Em seu algoritmo, o problema foi primeiro

transformado em um problema de designação e, a seguir, foi utilizada a técnica de escalonamento do maior parâmetro (número de vértices, número de arestas) do problema.

Gabow e Tarjan [GbT88] apresentaram um algoritmo para o problema de b-emparelhamento de custo máximo em um multigrafo, com escalonamento do maior parâmetro do problema. Foi feito um estudo da complexidade deste algoritmo utilizando processamento paralelo.

## **1.6- Problemas de fluxo em um digrafo.**

Nesta secção estão colocados os problemas de fluxo máximo, de caminho mínimo e de fluxo de custo mínimo. O problema de fluxo máximo está relacionado com o problema de seleção estudado, enquanto os outros dois problemas estão relacionados com a generalização do problema de seleção estudada.

### **1.6.1- Problema de fluxo máximo.**

É um problema de programação linear e seu dual é conhecido como o problema de determinação do corte mínimo em um digrafo. O problema de fluxo máximo é um dos problemas mais conhecidos, mais fundamentais e mais estudados da teoria de redes. Sua resolução é fundamental para a resolução de diversos problemas de áreas diferentes, notadamente os, de otimização combinatória, uma vez que, alguns destes problemas podem ser resolvidos, através de sua transformação em um problema de fluxo máximo em redes.

Diversos algoritmos cada vez mais rápidos (com menor complexidade) para a sua resolução têm sido apresentados; a seguir, são apresentados os trabalhos mais importantes ou mais recentes sobre este problema.

Ford e Fulkerson [FdF56] apresentaram o primeiro algoritmo para a resolução do problema de fluxo máximo. Este algoritmo determinava no digrafo, enquanto possível, caminhos aumentantes (sequência de arcos que admitem um aumento do fluxo) do vértice  $s$  para o vértice  $t$ .

Edmonds e Karp [EdK68] demonstraram que se fosse utilizado o caminho aumentante de menor comprimento para aumento do fluxo, a complexidade do algoritmo diminuiria, tornando-se polinomial; isto é, poderia ser escrita como uma função polinomial dos dados de entrada do problema.

Dinic [Dnc70] propôs, utilizando o processo de busca em largura em um digrafo, um método para determinação do conjunto maximal dos menores caminhos aumentantes (definido como referente).

Karzanov [Kzn74] apresentou um algoritmo que utilizava o conceito de referente introduzido por Dinic e introduzia o conceito de pré-fluxo (para cada vértice, a quantidade de “fluxo”, que chega é maior ou igual à quantidade que sai, exceto no vértice de fonte do fluxo). Este algoritmo é, no momento, um dos de menor complexidade para digrafos densos.

Malhotra, Kumar e Maheshwari [MKM78] apresentaram um algoritmo, ainda utilizando o conceito de referente, com a mesma complexidade, que era mais simples que o de Karzanov.

Sleator e Tarjan [SIT83] apresentaram um algoritmo que utilizava para manipulação dos dados do digrafo, o conceito de árvores dinâmicas da teoria de estrutura de dados. Este algoritmo é, no momento, um dos de menor complexidade para digrafos esparsos.

Galil [Gll80] apresentou um algoritmo que é uma pequena modificação do de Karzanov, mas que é o de menor complexidade para alguns poucos tipos de digrafos.

Goldberg e Tarjan [GdT86] utilizando a estrutura de dados de árvores dinâmicas e o conceito de pré-fluxo, escreveram um novo algoritmo, com complexidade para grafos esparsos, igual à dos melhores. Neste trabalho foi, também, demonstrado que o algoritmo permite uma eficiente implementação paralela e distribuída.

Existem alguns trabalhos recentes utilizando, também o conceito de escalonamento do máximo da capacidade dos arcos do digrafo.

Gushfield, Martel e Fernandez-Bacca [GMF87] calcularam a complexidade do algoritmo de Karzanov e a do algoritmo de Malhotra, Kumar e Maheshwari, para o caso particular de digrafos bipartidos. Foi demonstrado, que para digrafos bipartidos, com bipartição  $V_1, V_2$  do conjunto de vértices e  $V_1 = O(n^{1-\epsilon})$  para algum  $\epsilon > 0$ , a complexidade destes algoritmos é menor do que a demonstrada para digrafos quaisquer.

No capítulo 2 é apresentado um algoritmo (I) para a resolução do problema de seleção, que calcula o fluxo máximo em um digrafo bipartido, e no cálculo da complexidade deste algoritmo, os resultados deste último trabalho, são utilizados. No mesmo capítulo, no algoritmo A de b-emparelhamento, foi utilizada a busca em largura no grafo, para determinação do conjunto

maximal de caminhos aumentantes de comprimento mínimo (referente introduzido no algoritmo de Dinic).

### 1.6.2- Problema de caminho mínimo.

É um problema de programação inteira, definido em um grafo (digrafo) em que a cada aresta (arco) está associado um inteiro não negativo (peso) e consiste em dados dois vértices, determinar o caminho de menor comprimento (soma dos pesos das arestas ou arcos do caminho) entre estes dois vértices.

Algoritmos para a resolução deste problema são de grande interesse prático, além de serem muito utilizados, na resolução de outros problemas de otimização combinatória como, por exemplo, em grande parte dos algoritmos de fluxo de custo mínimo. Os algoritmos para determinação do caminho mínimo entre dois vértices  $v_i$  e  $v_j$ , na verdade determinam o caminho mínimo de  $v_i$  para todos os outros vértices.

O algoritmo de Dijkstra [Djk59] é o mais antigo e, praticamente, a base de todos os algoritmos atualmente utilizados.

Gabow [GbW85] apresentou um algoritmo utilizando o algoritmo de Dijkstra e a técnica de escalonamento de Edmonds e Karp, melhorando a complexidade do algoritmo.

Fredman e Tarjan [FmT87] apresentaram uma nova versão do algoritmo de Dijkstra com a utilização de uma estrutura de dados especial ("Fibonacci heaps"), conseguindo diminuir a complexidade do algoritmo.

Ahuja, Mehlhorn, Orlin e Tarjan [AMO90] implementaram o algoritmo de Dijkstra utilizando uma nova estrutura de dados ("radix heaps") e também o implementaram utilizando uma combinação desta nova estrutura com a utilizada por Fredman e Tarjan ("Fibonacci heaps"). O resultado obtido para a complexidade deste último é o melhor até o momento.

No capítulo 4, são apresentados dois algoritmos (II e III) para o problema de seleção generalizada estudado, que modificam o problema, utilizando o cálculo de caminhos mínimos entre vértices do digrafo. No cálculo das complexidades destes dois algoritmos, este último resultado é utilizado.

### 1.6.3- Problema de fluxo de custo mínimo.

É um dos problemas mais fundamentais e mais estudados da teoria de redes (formulado no capítulo 3, secção 3.5). Este problema é, também apresentado na literatura como problema de transbordo. Este problema pode ser transformado em tempo polinomial, em outros problemas tais como : o problema de transporte (definido em um digrafo bipartido), o problema de circulação (definido em um digrafo com suprimento ou demanda nos vértices iguais a zero), o mesmo se verificando para as transformações inversas.

Diversos algoritmos para a resolução deste problema têm sido apresentados recentemente. Estes algoritmos estão sempre relacionados à técnica de escalonamento de um dos parâmetros do problema introduzida por Edmonds e Karp [EdK72]. Esta técnica é utilizada nos capítulos 2 e 4 deste trabalho.

Os algoritmos com escalonamento de um dos parâmetros são iterativos e podem ser divididos em dois grandes grupos, de acordo com o problema que é resolvido em cada iteração. O primeiro grupo utiliza em cada iteração o algoritmo de Dijkstra (contido por exemplo em Bondy e Murty [BdM76]) de resolução do problema de caminhos mínimos e é composto pelos trabalhos de Orlin [Orl88], Galil e Tardos [GIT86], Fujishige [Fjs85], e Edmonds e Karp [EdK72]. O segundo grupo utiliza em cada iteração um dos algoritmo de fluxo máximo e fazem parte deste grupo os trabalhos de Goldberg e Tarjan [GdT88], Goldberg e Tarjan [GdT87], Tardos [Trd84] e Rock [Roc80]. O algoritmo apresentado por Goldberg e Tarjan [GdT88] utiliza o problema de fluxo máximo de uma maneira diferente, uma vez que o utiliza para o cancelamento de ciclos com custo negativo.

Os algoritmos com escalonamento de um dos parâmetros podem, também, ser classificados em três grupos quanto ao parâmetro que está sendo escalonado. O primeiro grupo utiliza como parâmetro de escalonamento o máximo da capacidade dos arcos do digrafo; neste grupo estão os trabalhos de Galil e Tardos [GIT86], Fujishige [Fjs85], Rock [Roc80] e Edmonds e Karp [EdK72]. O segundo grupo utiliza como parâmetro de escalonamento o máximo custo dos arcos do digrafo; neste grupo estão os trabalhos de Gabow e Tarjan [GbT89], Goldberg e Tarjan [GdT88], Goldberg e Tarjan [GdT87], Tardos [Trd84], e Rock [Roc80]. O terceiro grupo é formado pelo algoritmo de Orlin [Orl88], que inicialmente transforma o problema de fluxo de custo mínimo em um problema de fluxo de custo mínimo não capacitado, e utiliza como parâmetro de escalonamento o máximo da demanda ou suprimento de seus vértices.

Uma terceira subdivisão destes algoritmos pode, ainda ser feita, se considerarmos qual o problema, que é considerado para introdução teórica do algoritmo. Apresentam o problema sob a forma de circulação os trabalhos de Gabow e Tarjan [GbT89], Goldberg e Tarjan [GdT88],

Goldberg e Tarjan [GdT87], Galil e Tardos [GIT86], Fujishige [Fjs85], e Tardos [Trd84]. Apresentam o problema como problema em uma rede balanceada (soma das ofertas é igual à soma das demandas) com várias entradas e várias saídas os trabalhos de Rock [Roc80], Orlin [Orl88], e Edmonds e Karp [EdK72].

## CAPÍTULO 2.

### UM PROBLEMA DE SELEÇÃO EM UM DIGRAFO.

#### 2.1- Problema de seleção.

Seja  $G = (V, E)$  um digrafo com um conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$  e um conjunto de arcos  $E = \{e_1, e_2, \dots, e_m\}$ , onde cada arco é um par ordenado de vértices distintos e a cada vértice  $v_i$  é associado um valor inteiro  $b_i$ .

#### Definições :

- Um subconjunto  $V^*$  do conjunto de vértices  $V$  é um fecho ("closure") se para todo arco  $(v_i, v_j)$  de  $E$ ,  $v_i \in V^*$  implica  $v_j \in V^*$ .

- Valor de um fecho  $V^*$  é  $\sum [b_i : i=1, 2, \dots, n \text{ e } v_i \in V^*]$

- O problema do fecho ótimo consiste em determinar um fecho de valor máximo.

O problema do fecho ótimo é um problema de seleção de vértices e foi estudado por Rhys [Rhy70] e Balinski [Blk70] (ambos considerando apenas digrafos bipartidos) e por Picard [Pic76].

O problema do fecho ótimo pode ser formulado como o problema de determinar um vetor binário  $X = (x_1, x_2, \dots, x_n)$  solução ótima de :

$$P: \quad \text{máx } f(x) = \sum_{i=1}^n b_i \cdot x_i$$

$$\text{s. a. } x_i \leq x_j$$

$$x_i = 0, 1$$

para todo arco  $(v_i, v_j) \in E$

para todo vértice  $v_i \in V$

Assim,  $x_i = 1$  ( $x_i = 0$ ) indica que o vértice  $v_i$  deve (não deve) ser selecionado. Desta forma,  $x_i = 1$  e  $(v_i, v_j) \in E$  implicam em  $x_j = 1$ .

O interesse por este problema de seleção se deve ao estudo dos modelos para planejamento de exploração de minas do tipo céu aberto. Deseja-se determinar a forma final da geometria do buraco a ser aberto, a fim de maximizar o lucro, dado o corpo do minério decomposto em blocos de diferentes tipos de minérios e conhecendo, para cada bloco, o valor do minério e o custo de extração daquele tipo de minério, além das limitações geométricas quanto à inclinação das paredes do buraco. O problema de determinação do contorno ótimo do buraco a ser feito é modelado como um problema de determinação do fecho ótimo em um digrafo, onde cada vértice  $v_i$  com valor  $b_i$ , corresponde a um bloco  $i$  com valor líquido (valor do minério - custo de extração)  $b_i$ , e onde existe um arco  $(v_i, v_j)$  do vértice  $v_i$  para o vértice  $v_j$ , se a remoção do bloco  $j$  é necessária para a remoção do bloco  $i$ . Este problema foi descrito e modelado em um digrafo por Lerchs e Grossman [LcG65].

Outros exemplos de problemas que podem ser formulados como um problema de fecho ótimo são:

- Planejamento da localização ótima de terminais, de modo a maximizar o lucro, sendo dados o custo fixo de instalação de cada terminal e o lucro associado a cada serviço, que se torna disponível; a existência de um par de terminais permite o estabelecimento de um serviço entre eles. Este problema foi modelado por Rhys [Rhy70], em um digrafo bipartido, com os vértices de um lado da bipartição, representando as possíveis localizações de terminais, com valores iguais ao custo de cada terminal, e os vértices do outro lado da bipartição representando os possíveis serviços com valores iguais ao lucro de cada serviço. Os arcos  $(v_i, v_j)$  indicam que o terminal correspondente ao vértice  $v_i$  é necessário para a existência do serviço correspondente ao vértice  $v_j$ .

- Determinação dos projetos a serem executados para maximizar o lucro, quando existe um lucro líquido (receita - custo) associado a cada projeto e existe uma relação de dependência entre os projetos. Este problema pode ser modelado em um grafo misto; com cada projeto sendo representado por um vértice e as relações de dependência entre projetos, por arcos ou arestas entre os vértices correspondentes a estes projetos, dependendo da relação de dependência existente entre estes.

O conceito de fecho pode ser estendido para um grafo misto  $G'$  definido por um conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$ , um conjunto de arcos  $E = \{e_1, e_2, \dots, e_{m_1}\}$  e um conjunto de arestas  $E' = \{e'_1, e'_2, \dots, e'_{m_2}\}$ ; cada aresta é um par não ordenado de

vértices distintos. Um subconjunto  $V^*$  do conjunto de vértices é um fecho de  $G'$ , se para todo arco  $(v_i, v_j) \in E$ ,  $v_i \in V^*$  implica  $v_j \in V^*$  e se para toda aresta  $\{v_i, v_j\} \in E'$ ,  $v_i \in V^*$  se e somente se  $v_j \in V^*$ . Este problema pode ser transformado no problema de fecho ótimo em um digrafo, substituindo-se cada aresta  $\{v_i, v_j\}$  do grafo misto  $G'$  pelo par de arcos  $(v_i, v_j)$  e  $(v_j, v_i)$ . Uma outra transformação mais eficiente consiste em reduzir (“shrink”) cada aresta  $\{v_i, v_j\}$  e os vértices  $v_i, v_j$  a um pseudo-vértice. Isto é, cada ocorrência de  $x_j$  na formulação do problema é substituída por  $x_i$  e a restrição  $x_i = x_j$  é eliminada.

Através de transformações algébricas, Picard [Pic76] mostrou que o problema de fecho ótimo no digrafo  $G$  pode ser transformado em um problema de corte mínimo, separando  $s$  de  $t$ , no digrafo  $G_1 = (V_1, E_1)$ , onde  $V_1 = V \cup \{s, t\}$ ,  $V^+ = \{v_i : b_i \geq 0\}$ ,  $V^- = \{v_i : b_i < 0\}$  e  $E_1 = E \cup \{(s, v_i) : v_i \in V^+\} \cup \{(v_i, t) : v_i \in V^-\}$ ; aos arcos dos tipos  $(s, v_i)$  e  $(v_i, t)$  são associadas capacidades  $|b_i|$  e a cada arco de  $E$  é associada capacidade infinita. Utilizando os mais eficientes algoritmos de fluxo máximo, o problema de seleção pode ser resolvido em  $O(n_1^3)$  pelos métodos de Karzanov [Kzn74] e de Malhotra, Kumar e Maheshwari [MKM78], ou em  $O(m_1 \cdot n_1 \cdot \log_2(n_1^2/m_1))$  pelo método de Goldberg e Tarjan [GdT88], ou ainda em  $O(n_1^{5/3} \cdot m_1^{2/3})$  pelo método de Galil [Gal80], onde  $m_1 = (m+n)$  é o número de arcos de  $G_1$  e  $n_1 = n+2$  é o número de vértices de  $G_1$ .

Neste capítulo, é proposta a resolução do problema de fecho ótimo através de sua transformação em um problema de fecho ótimo em um digrafo bipartido. A resolução do fecho ótimo bipartido é proposta através do problema de fluxo máximo em um digrafo bipartido ou através de sua transformação em um problema de b-emparelhamento bipartido. Dois algoritmos são apresentados para a determinação do b-emparelhamento ótimo em um grafo bipartido.

O problema de seleção resolvido através do cálculo do fluxo máximo, não tira proveito, no cálculo do fluxo dos arcos do digrafo  $G$  não terem limitante superior. Na realidade nenhum corte mínimo pode apresentar um arco de  $E$  (com capacidade infinita) direcionado de  $S$  para  $T$ , onde  $(S, T)$  é a bipartição de vértices determinado pelo corte mínimo com  $s \in S$  e  $t \in T$ . No problema de b-emparelhamento máximo os arcos não têm limitante superior, logo o problema de seleção se enquadra melhor neste problema, isto é, o cálculo da complexidade do algoritmo que utiliza o problema de b-emparelhamento fica mais próximo do esforço computacional real.

## 2.2- Transformação em um problema de seleção bipartido.

Sejam  $P$  o problema do fecho ótimo no digrafo  $G = (V, E)$  e  $P_B$  o problema do fecho ótimo no digrafo  $G_B = (V, E_B)$ , onde  $E_B = \{ (v_i, v_j) : v_i \in V^+, v_j \in V^- \text{ e existe caminho orientado de } v_i \text{ para } v_j \text{ em } G \}$ . Ou seja,  $G_B$  é um subgrafo bipartido maximal do fecho transitivo de  $G$ , com partição de vértices  $(V^+, V^-)$ .

$$P_B: \quad \text{máx } f(x) = \sum_{i=1}^n b_i \cdot x_i$$

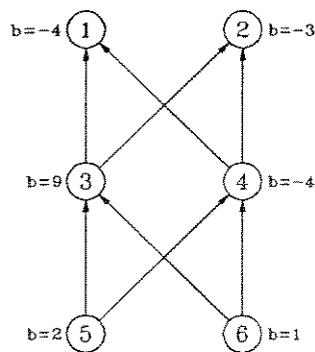
$$\text{s.a. } x_i \leq x_j$$

$$x_i = 0, 1$$

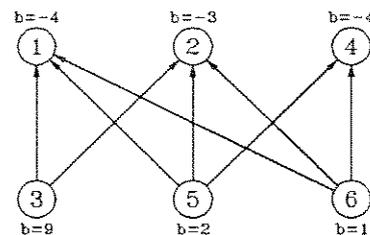
$$\text{para todo arco } (v_i, v_j) \in E_B$$

$$\text{para todo vértice } v_i \in V$$

Exemplo :



$G = (V, E)$



$G_B = (V, E_B)$

**Teorema 2.1 :**  $x^*$  é solução ótima de  $P$  se e somente se  $x^*$  é solução ótima de  $P_B$ .

**Demonstração :**  $P$  e  $P_B$  possuem a mesma função objetivo; desta forma, é suficiente provar que :

I:- se  $x^*$  é solução viável de  $P$  então  $x^*$  é solução viável de  $P_B$ .

II:- se  $x^*$  é solução ótima de  $P_B$  então  $x^*$  é solução viável de  $P$ .

I:- Seja  $x^*$  uma solução viável de  $P$ . Se  $(v_{i_1}, v_{i_r}) \in E_B$  então existe uma cadeia  $v_{i_1}, v_{i_2}, \dots, v_{i_r}$  em  $G$  e  $x^*$  satisfaz  $x_{i_1}^* \leq x_{i_2}^* \leq \dots \leq x_{i_r}^*$ , de forma que  $x_{i_1}^* \leq x_{i_r}^*$ . Ou seja,  $x^*$  é uma solução viável de  $P_B$ .

II:- Seja  $x^*$  uma solução ótima de  $P_B$ . Devemos verificar que a restrição  $x_i \leq x_j$  correspondente a cada arco  $(v_i, v_j) \in E$  é satisfeita por  $x^*$ . Assim, devemos considerar três casos:

1:-  $(v_i, v_j) \in E_B$  :

Neste caso,  $x^*$  satisfaz à restrição  $x_i \leq x_j$ , porque é uma solução ótima de  $P_B$ .

2:-  $(v_i, v_j)$  não pertence a  $E_B$  com  $v_j \in V^+$  e  $v_i \in V^+$ .

Se  $x_j^* = 1$  então a restrição  $x_i \leq x_j$  é satisfeita; assim, suponha que  $x_i^* = 0$ .

Caso não exista  $(v_j, v_{j_r}) \in E_B$  com  $x_{j_r}^* = 0$ , então é possível construir uma solução melhor do que  $x^*$  fazendo  $x_j = 1$ . Desta forma, existe  $(v_j, v_{j_r}) \in E_B$  com  $x_{j_r}^* = 0$ ; ou seja, existe uma cadeia  $v_j, v_{j_1}, \dots, v_{j_r}$  em  $G$  que define a restrição  $x_j \leq x_{j_r}$  em  $P_B$ , da mesma forma, que existe a cadeia  $v_i, v_j, v_{j_1}, \dots, v_{j_r}$  em  $G$ , que define a restrição  $x_i \leq x_{j_r}$  em  $P_B$ . Portanto,  $x_i^* \leq x_{j_r}^* = 0$  e  $x^*$  satisfaz à restrição  $x_i \leq x_j$ .

3:-  $(v_i, v_j)$  não pertence a  $E_B$  com  $v_i \in V^-$ .

Se  $x_i^* = 0$  então a restrição  $x_i \leq x_j$  é satisfeita; assim, suponha que  $x_i^* = 1$ .

Caso não exista  $(v_{i_1}, v_i) \in E_B$  com  $x_{i_1}^* = 1$ , então é possível construir uma solução melhor do que  $x^*$  fazendo  $x_i = 0$ . Desta forma, existe  $(v_{i_1}, v_i) \in E_B$  com  $x_{i_1}^* = 1$ ; ou seja, existe uma cadeia  $v_{i_1}, v_{i_2}, \dots, v_{i_r}, v_i$  em  $G$  que define a restrição  $x_{i_1} \leq x_i$  em  $P_B$ , da mesma forma, que existe a cadeia  $v_{i_1}, v_{i_2}, \dots, v_{i_r}, v_i, v_j$  em  $G$ , que se  $v_j \in V^-$  define a restrição  $x_{i_1} \leq x_j$  em  $P_B$ . Portanto,  $1 = x_{i_1}^* \leq x_j^*$  e  $x^*$  satisfaz à restrição  $x_i \leq x_j$ . Se  $v_j \in V^+$  e  $x_j^* = 0$  existe a cadeia  $v_{i_1}, \dots, v_{i_r}, v_i, v_j, v_{j_1}, \dots, v_{j_s}$  com  $v_{i_1} \in V^+$  e  $v_{j_s} \in V^-$  que define as restrições  $x_{i_1} \leq x_i$ ,  $x_j \leq x_{j_s}$  e  $x_{i_1} \leq x_{j_s}$  em  $P_B$ , logo  $x_j^*$  não pode ser igual à zero e se  $x_j^* = 1$  então  $x^*$  satisfaz  $x_i \leq x_j$ . ■

### 2.3- Algoritmo I de resolução do problema de seleção.

$P_B$  é um problema de seleção definido no digrafo bipartido  $G_B$ , então, aplicando Picard [Pic76], pode ser resolvido através do problema de fluxo máximo em um digrafo bipartido. Logo  $P$  pode, também, ser resolvido pelo mesmo problema de fluxo máximo.

### 2.3.1- Algoritmo I de resolução do problema de seleção em $G = (V, E)$ .

I) Construir  $G_B = (V, E_B)$ , subgrafo bipartido do fecho transitivo de  $G$  com  $n = |V^+| + |V^-|$  vértices e no máximo  $|V^+| \cdot |V^-|$  arestas.

II) Construir  $G_1 = (V_1, E_1)$ , onde  $V_1 = V \cup \{s, t\}$  e  $E_1 = E_B \cup \{(s, v_i) : v_i \in V^+\} \cup \{(v_i, t) : v_i \in V^-\}$ ; os arcos dos tipos  $(s, v_i)$  e  $(v_i, t)$  têm capacidades  $|b_i|$  e os arcos de  $E_B$  têm capacidade infinita.

III) Resolver o problema de fluxo máximo em  $G_1$ .

IV) Transformar a solução ótima do problema de fluxo máximo obtida na solução ótima do problema de corte mínimo em  $G_1$  (= solução ótima do problema de seleção  $P$ ).

### 2.3.2- Exemplo de resolução do problema de seleção utilizando o Algoritmo I.

$$P: \quad \text{máx } f(x) = -4 \cdot x_1 - 3 \cdot x_2 + 9 \cdot x_3 - 4 \cdot x_4 + 2 \cdot x_5 + x_6$$

$$\text{s. a. } x_3 \leq x_1$$

$$x_4 \leq x_1$$

$$x_3 \leq x_2$$

$$x_4 \leq x_2$$

$$x_5 \leq x_3$$

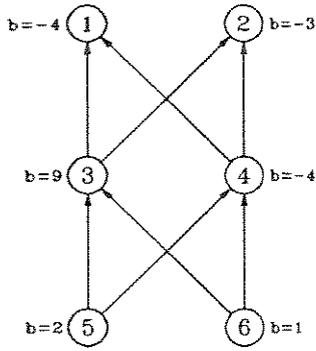
$$x_6 \leq x_3$$

$$x_5 \leq x_4$$

$$x_6 \leq x_4$$

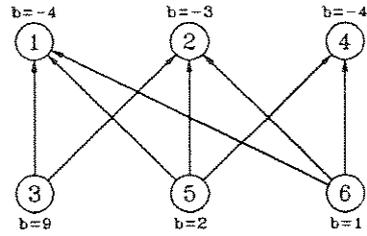
$$x_i = 0, 1 \quad i \in \{1, 2, 3, 4, 5, 6\}$$

Este problema de seleção é representado no digrafo  $G$  e, como demonstrado, uma solução ótima de  $P$  pode ser determinada a partir de uma solução ótima do problema representado pelo digrafo  $G_B$ ; os dois digrafos são indicados a seguir.



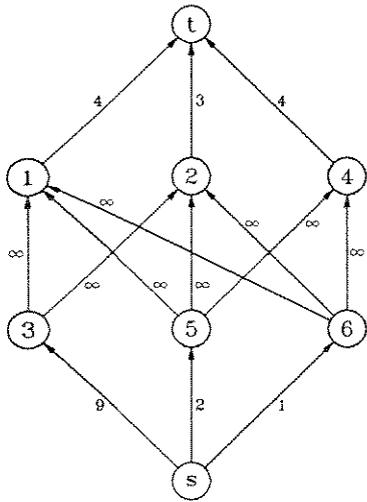
$G = (V, E)$

I)



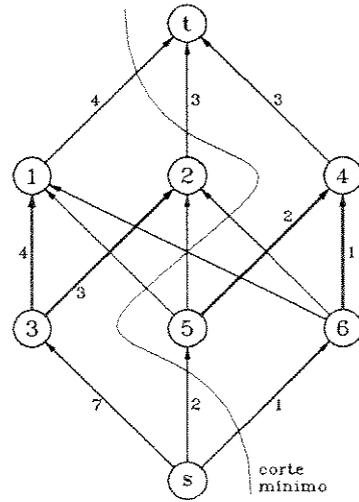
$G_B = (V, E_B)$

II)



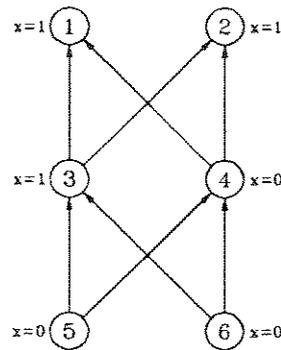
$G_1 = (V_1, E_1)$   
com a indicação da  
capacidade de seus arcos.

III)



$G_1 = (V_1, E_1)$   
com a indicação do corte mínimo  
e de um fluxo máximo de  $G_1$ .

IV) Os vértices  $v_1, v_2$  e  $v_3$  estão do mesmo lado do vértice  $s$  no corte mínimo de  $G_1$ , isto é,  $v_1, v_2$  e  $v_3$  têm  $x_1 = x_2 = x_3 = 1$ . Os vértices  $v_4, v_5$  e  $v_6$  estão do mesmo lado do vértice  $t$  no corte mínimo de  $G_1$ , ou seja,  $v_4, v_5$  e  $v_6$  têm  $x_4 = x_5 = x_6 = 0$ . Indicando que os vértices  $v_1, v_2$  e  $v_3$  são os vértices a serem selecionados, para maximizar o valor da seleção.



$$G = (V, E)$$

com a indicação da seleção ótima.

### 2.3.3- Complexidade do Algoritmo I do problema de seleção.

Complexidade do Algoritmo I do problema de seleção em um digrafo qualquer é dada pela soma de :

$$\text{número de arestas examinadas na determinação de } G_B = o(|V^+|^2 \cdot |V^-|)$$

$$\text{número de operações no cálculo do fluxo máximo de } G_I = o(|V^+|^2 \cdot |V^-|)$$

número de operações para transformar a solução

$$\text{do fluxo máximo na solução do fecho ótimo} = o(|V^+| + |V^-| + |V^+| \cdot |V^-|)$$

Complexidade do Algoritmo I:	$o( V^+ ^2 \cdot  V^- )$ .
------------------------------	----------------------------

Para o número de operações no cálculo do fluxo máximo de  $G_I$  foi considerada a complexidade para o algoritmo de fluxo máximo em um digrafo bipartido calculada por Gusfield, Martel e Fernandez-Bacca [GMF87].

### 2.4- Transformação em um problema de b-emparelhamento bipartido.

Seja  $D$  o problema de b-emparelhamento máximo no grafo  $G^* = (V, E^*)$ , onde  $E^* = \{ (v_i, v_j) : i, j = 1, 2, \dots, n \text{ e } (v_i, v_j) \in E_B \}$  e a cada vértice  $v_i$  é associado o inteiro positivo  $|b_i|$ , isto é:

$$\begin{aligned}
D: \quad & \text{máx } g(z) = \sum [z_{ij} : (v_i, v_j) \in E^*] \\
& \text{s.a. } \sum [z_{ij} : (v_i, v_j) \in E^*] \leq |b_i| && \text{para todo } v_i \in V^+ \\
& \sum [z_{ij} : (v_i, v_j) \in E^*] \leq |b_j| && \text{para todo } v_j \in V^- \\
& z_{ij} \geq 0 && \text{para todo } (v_i, v_j) \in E^* \\
& z_{ij} \text{ inteiro} && \text{para todo } (v_i, v_j) \in E^*
\end{aligned}$$

Seja  $D^*$ , definido em  $G^*$ , o problema D sem a restrição  $z_{ij}$  inteiro; como  $G^*$  é um grafo bipartido, a matriz dos coeficientes das restrições de  $D^*$  é totalmente unimodular, então qualquer solução básica do programa linear  $D^*$  é inteira e satisfaz D.

Seja  $P^*$  o dual do problema  $D^*$ , isto é:

$$\begin{aligned}
P^*: \quad & \text{mín } h(y) = \sum_{i=1}^n |b_i| \cdot y_i \\
& \text{s.a. } y_i + y_j \geq 1 && \text{para toda aresta } (v_i, v_j) \in E^* \\
& y_i \geq 0 && \text{para todo vértice } v_i \in V
\end{aligned}$$

**Teorema 2.2:**  $x^*$  é solução ótima de  $P_B$  se e somente se  $y^*$  é solução ótima inteira de  $P^*$ , onde

$$y_i^* = \begin{cases} 1 - x_i^* & \text{se } v_i \in V^+ \text{ e} \\ x_i^* & \text{se } v_i \in V^- \end{cases}$$

**Demonstração:** Como  $G^*$  é um grafo bipartido, a matriz dos coeficientes das restrições de  $P^*$  é totalmente unimodular, então qualquer solução básica do programa linear  $P^*$  é inteira.  $P^*$  é um problema de minimização com função objetivo linear com coeficientes de custo não negativos, então qualquer solução básica satisfaz  $0 \leq y_i \leq 1$ ; ou seja é solução de  $P_1$ , onde:

$$\begin{aligned}
P_1: \quad & \text{mín } h(y) = \sum_{i=1}^n |b_i| \cdot y_i \\
& \text{s.a. } y_i + y_j \geq 1 && \text{para toda aresta } (v_i, v_j) \in E^* \\
& y_i = 0, 1 && \text{para todo vértice } v_i \in V
\end{aligned}$$

Para verificar que os problemas  $P_A$  e  $P_B$  são equivalentes, basta fazer a mudança de variáveis:

$$y^*_i = \begin{cases} 1-x^*_i & \text{se } v_i \in V^+ \text{ e} \\ x^*_i & \text{se } v_i \in V^- \end{cases},$$

multiplicar a função objetivo por (-1), transformando o problema de minimização em um problema de maximização e desprezar, na função objetivo obtida, a constante:  $-\sum [b_i : v_i \in V^+]$ .

■

## 2.5- O problema de b-emparelhamento máximo bipartido.

Seja  $G^* = (V, E^*)$  um grafo bipartido com  $|V| = n$  vértices e  $|E^*| = m$  arestas, onde a cada vértice  $v_i \in V$  está associado um número inteiro não negativo  $b_i$ .

### Definições:

-  $R=(r_1, r_2, \dots, r_m)$  é um **b-emparelhamento** de  $G^*$ , quando cada  $r_k$  é um inteiro não negativo, associado à aresta  $e_k$ , satisfazendo:

$$\sum [r_k : k=1, 2, \dots, m \text{ e } v_i \in e_k] \leq b_i \quad \text{para todo } v_i \in V.$$

(Como conseqüência,  $R=(r_1, r_2, \dots, r_m)$  representa uma solução viável do problema D (secção 2.4) de b-emparelhamento máximo bipartido; bastando considerar  $z_{ij} = r_k$  para toda aresta  $e_k = \{v_i, v_j\}$ .)

- Um **b-emparelhamento** de  $G^*$  é **perfeito**, quando satisfaz:

$$\sum [r_k : k=1, 2, \dots, m \text{ e } v_i \in e_k] = b_i \quad \text{para todo } v_i \in V.$$

- Valor de um b-emparelhamento  $R$  é  $\|R\| = \sum [r_i : i=1, 2, \dots, m]$ .

- O problema de b-emparelhamento máximo consiste em determinar o b-emparelhamento de valor máximo.

- Se  $R$  é um  $b$ -emparelhamento de  $G^*$  então o vértice  $v_i$  é dito **R-saturado**, quando  $\sum [r_j : j=1,2,\dots,m \text{ e } v_i \in e_j] = b_i$ , caso contrário é dito **não R-saturado**.

- Se  $R$  é um  $b$ -emparelhamento de  $G^*$  então um caminho é **R-alternante**, quando é um caminho entre dois vértices de  $V$ , com arestas  $e_i$  satisfazendo alternadamente uma das duas relações:  $r_i > 0$  ou  $r_i < \min\{b_j : v_j \in e_i \text{ e } j=1,2,\dots,n\}$ .

- Seja  $R$  um  $b$ -emparelhamento de  $G^*$ . Um caminho **R-aumentante**  $P$  é um caminho R-alternante entre dois vértices não R-saturados, com os demais vértices do caminho  $P$  R-saturados, com um número ímpar de arestas e toda aresta par  $e_i$  do caminho  $P$  tem  $r_i > 0$ . Indicamos  $P$  por  $(p_1, p_2, \dots, p_m)$ , onde:

$$p_i = \begin{cases} 0 & \text{se } e_i \text{ não pertence a } P, \\ 1 & \text{se } e_i \text{ é aresta ímpar de } P, \text{ e} \\ -1 & \text{se } e_i \text{ é aresta par de } P. \end{cases}$$

- O comprimento de um caminho  $P$  é o número de arestas do caminho e é indicado por  $|P|$  ( $= \sum [ |p_i| : i=1,2,\dots,m ]$ ).

- Seja  $R$  um  $b$ -emparelhamento de  $G^*$ . O peso de um caminho R-aumentante  $P$  é o máximo que  $R$  pode ser aumentado com o uso de  $P$ :

$\alpha = \min(\{r_i : p_i = -1\}, \{b_i - \sum [r_j : j=1,2,\dots,m \text{ e } v_i \in e_j] : v_i \text{ é vértice extremo de } P\})$ .

- Seja  $R$  um  $b$ -emparelhamento de  $G^*$ . Referente de  $G^*$  relativo a  $R$  é o conjunto de todos os caminhos R-aumentantes de comprimento mínimo.

O conceito de referente é uma adaptação daquele introduzido por Dinic [Dnc70] em seu algoritmo para o problema de fluxo máximo.

Quando  $b_1 = b_2 = \dots = b_n = 1$ , o problema de  $b$ -emparelhamento máximo é chamado de emparelhamento máximo. O algoritmo para a solução do problema de emparelhamento máximo em um grafo bipartido de menor complexidade ( $O(n^b \cdot m)$ ) é o de Hopcroft e Karp [HpK73]. O algoritmo para o problema de  $b$ -emparelhamento máximo bipartido,

apresentado a seguir, é uma generalização do de Hopcroft e Karp para o problema de emparelhamento máximo bipartido.

## 2.6- Algoritmo A de b-emparelhamento máximo bipartido.

### 2.6.1- Algoritmo A de b-emparelhamento máximo bipartido.

I) Inicializar  $R$  com  $(0, 0, \dots, 0)^T$ ;

II) Repetir os passos (1,2) até o passo (1) ordenar parada.

1) Pesquisa : Construir o referente, fazendo uma pesquisa em largura ("breadth-first search") no grafo, partindo das raízes, que são os vértices não saturados de  $V^+$ . Se o referente for vazio, então parar : o b-emparelhamento é máximo.

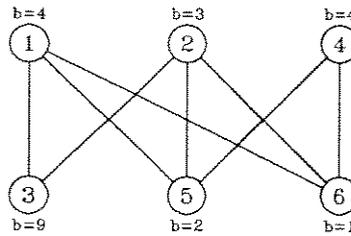
2) Ampliação : Para cada caminho  $P$  do referente, determinar o seu peso  $\alpha$ , fazer a ampliação do b-emparelhamento e atualizar o referente.

A validade do algoritmo e sua complexidade são demonstradas nos teoremas da seção 2.7, que são uma generalização dos de Hopcroft e Karp [HpK73] para emparelhamento bipartido ou uma generalização dos de Even e Tarjan [EvT75] para fluxo máximo em um tipo particular de grafo. Hopcroft e Karp demonstram que o número de referentes utilizados por seu algoritmo é no máximo igual a  $n^{\frac{1}{2}}$ . O número de referentes utilizados no algoritmo acima é demonstrado, no Teorema 2.8, ser no máximo  $2 \cdot b^{\frac{1}{2}}$ , onde  $b \leq \min\{\sum[b_i : v_i \in V^+], \sum[|b_j| : v_j \in V^-]\}$ .

### 2.6.2- Exemplo de resolução do b-emparelhamento máximo bipartido utilizando o Algoritmo A.

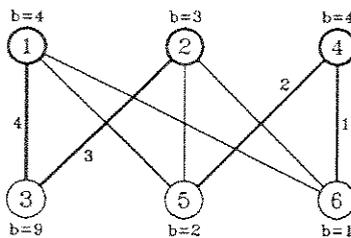
$$\begin{aligned}
 \text{D:} \quad & \text{máx } g(z) = z_{13} + z_{15} + z_{16} + z_{23} + z_{25} + z_{26} + z_{45} + z_{46} \\
 \text{s.a.} \quad & z_{13} + z_{15} + z_{16} \leq 4 \\
 & z_{23} + z_{25} + z_{26} \leq 3 \\
 & z_{45} + z_{46} \leq 4 \\
 & z_{13} + z_{23} \leq 9 \\
 & z_{15} + z_{25} + z_{45} \leq 2 \\
 & z_{16} + z_{26} + z_{46} \leq 1 \\
 & z_{ij} \geq 0 \qquad \text{para todo } \{v_i, v_j\} \in E^* \\
 & z_{ij} \text{ inteiro} \qquad \text{para todo } \{v_i, v_j\} \in E^*
 \end{aligned}$$

Este problema pode ser representado no grafo abaixo.



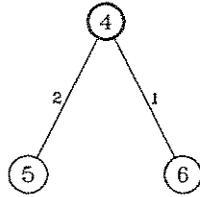
$$\text{I) } z_{13} = z_{15} = z_{16} = z_{23} = z_{25} = z_{26} = z_{45} = z_{46} = 0$$

II) As raízes são:  $v_1$ ,  $v_2$  e  $v_4$ . O referente é, portanto, o grafo dado a seguir.



$$z_{13} = 4, \quad z_{23} = 3, \quad z_{45} = 2, \quad z_{46} = 1 \text{ e } z_{15} = z_{16} = z_{25} = z_{26} = 0$$

II) O conjunto das raízes é :  $\{v_4\}$ . Na tentativa de construção do referente é formado o grafo dado a seguir, que não contém nenhum caminho aumentante; isto é, o referente é vazio, logo o b-emparelhamento obtido no passo anterior é ótimo.



$z_{13} = 4$  ,  $z_{23} = 3$  ,  $z_{45} = 2$  ,  $z_{46} = 1$  e  $z_{15} = z_{16} = z_{25} = z_{26} = 0$  é uma solução ótima do problema D de b-emparelhamento máximo bipartido.

**2.6.3- Validade e complexidade do Algoritmo A de b-emparelhamento máximo bipartido.**

Seja  $G = (V, E)$  um grafo simples e bipartido com  $|V| = n$  vértices e  $|E| = m$  arestas.

**Teorema 2.3 :** Se  $R$  é um b-emparelhamento de  $G$ ,  $P$  é um caminho  $R$ -aumentante em  $G$  com peso  $\alpha$  e  $R' := R + \alpha \cdot P$ , então  $R'$  é um b-emparelhamento e  $\|R'\| = \|R\| + \alpha$ .

**Demonstração :** Do algoritmo temos:

$$r_i' = r_i + \alpha \cdot p_i = \begin{cases} r_i \geq 0 , & \text{se } p_i = 0 \\ r_i + \alpha \geq 0 , & \text{se } p_i = 1 \\ r_i - \alpha \geq 0 , & \text{se } p_i = -1 \end{cases}$$

logo,  $r_i' \geq 0$

$$\sum [r_i' : i=1, 2 \dots m \text{ e } v_j \in e_i] = \begin{cases} \sum [r_i : i=1, 2 \dots m \text{ e } v_j \in e_i] & \text{se } v_j \text{ não é} \\ & \text{vértice extremo de } P \\ \sum [r_i + \alpha : i=1, 2 \dots m \text{ e } v_j \in e_i] & \text{se } v_j \text{ é} \\ & \text{vértice extremo de } P \end{cases}$$

então  $\sum [r_i' : i=1,2,\dots,m \text{ e } v_j \in e_i] \leq b_j$  para todo  $v_j \in V$ .

$P$  tem comprimento ímpar, portanto :

$$\sum_{i=1}^m r_i' = \sum_{i=1}^m r_i + \alpha, \quad \text{ou} \quad \|R'\| = \|R\| + \alpha$$

■

**Teorema 2.4 :** Se  $R$  é um b-emparelhamento de  $G$ , então :  
 $0 \leq r_i \leq \min \{b_j : v_j \in e_i \text{ e } j=1,2,\dots,n\}$ , para todo  $e_i \in E$ .

Conseqüência imediata da definição de peso de caminho aumentante  $\alpha$ .

■

**Teorema 2.5 :** Sejam  $R$  e  $S$  dois b-emparelhamentos de  $G$  com  $\|R\| > \|S\|$  e seja  $R \oplus S$  o b-emparelhamento de  $G$  representado por  $(|r_1-s_1|, |r_2-s_2|, \dots, |r_m-s_m|)$ , então o grafo  $G_{R+S} = (V, \{e_i : e_i \in E \text{ e } |r_i-s_i| > 0\})$  contém caminhos  $S$ -aumentantes com pesos tais que a soma destes pesos é maior ou igual a  $\|R\| - \|S\|$ .

**Demonstração :** Seja  $G'_{R+S}$  o multigrafo em que a cada aresta  $e_i \in E$  correspondem  $|r_i-s_i|$  arestas com os mesmos vértices extremos que  $e_i$ . Em  $G'_{R+S}$  as arestas correspondentes a  $e_i \in E$  podem ser consideradas como arestas de  $R$  ( $S$ ) quando  $r_i > s_i$  ( $r_i < s_i$ ). O grafo  $G'_{R+S}$  tem  $\|R\| - \|S\|$  arestas de  $R$  a mais que de  $S$ .

Em  $G'_{R+S}$  qualquer ciclo formado alternadamente com arestas de  $R$  e de  $S$  contém um número igual de arestas de  $R$  e de  $S$  ( $G$  é bipartido). Retirando de  $G'_{R+S}$  todos os ciclos alternantes, o grafo resultante continua com  $\|R\| - \|S\|$  arestas de  $R$  a mais que de  $S$ . O grafo  $G'_{R+S}$  pode, então ser considerado sem ciclos alternantes.

Seja  $P$  um caminho  $S$ -alternante maximal em  $G'_{R+S}$ , se  $P$  não é  $S$ -aumentante e as arestas de  $P$  são retiradas de  $G'_{R+S}$ , o multigrafo continuará com pelo menos  $\|R\| - \|S\|$  arestas de  $R$  a mais que de  $S$ . Entretanto, se  $P$  é  $S$ -aumentante, a diferença no número de arestas diminui de exatamente um. O processo de determinação e retirada de caminhos  $S$ -alternantes maximais de  $G'_{R+S}$  pode ser repetido até terminarem as arestas; pelo menos  $\|R\| - \|S\|$  destes caminhos  $S$ -alternantes são caminhos  $S$ -aumentantes com peso um.

A soma dos pesos destes caminhos  $S$ -aumentantes é maior ou igual a  $\|R\| - \|S\|$ .

Os caminhos S-aumentantes com uma mesma seqüência de vértices de  $G'_{R+S}$  podem ser considerados como um único caminho S-aumentante, tendo como peso o número destes caminhos em  $G'_{R+S}$ . Como conseqüência tem-se determinados em  $G'_{R+S}$  caminhos S-aumentantes com soma dos pesos maior ou igual a  $\|R\| - \|S\|$ .

■

**Corolário 2.1:** R é um b-emparelhamento máximo de G, se e só se não existe caminho R-aumentante em G (Berge-58).

**Demonstração:** Se existe um caminho R-aumentante P com peso  $\alpha$  então  $R' = R + \alpha \cdot P$  e  $\|R'\| > \|R\|$ , logo R não é máximo.

Se R não é máximo, então existe S que é máximo e em  $G_{R+S}$  ( $G_{R+S}$  contido em G) existe um caminho R-aumentante P, logo P é caminho R-aumentante em G.

**Teorema 2.6:** Se S é um b-emparelhamento e R é um b-emparelhamento máximo de G, então existe em G um caminho S-aumentante com comprimento menor ou igual a  $2 \cdot \lfloor \|R\| / (\|R\| - \|S\|) \rfloor - 1$

**Demonstração:** Em  $G'_{R+S}$  existem pelo menos  $\|R\| - \|S\|$  caminhos S-aumentantes com peso um.

$G'_{R+S}$  tem  $\sum [ |r_i - s_i| : e_i \in E \text{ e } s_i < r_i ]$  arestas de R, e

$$\sum [ |r_i - s_i| : e_i \in E \text{ e } s_i < r_i ] \leq \sum_{i=1}^m r_i = \|R\|$$

Se todos os caminhos S-aumentantes tiverem mesmo comprimento, existem no máximo  $\lfloor \|R\| / (\|R\| - \|S\|) \rfloor$  arestas de R por caminho S-aumentante de  $G'_{R+S}$ . Se algum caminho tiver um número maior destas arestas, outro terá um número menor, logo existe pelo menos um caminho S-aumentante em  $G'_{R+S}$ , com no máximo  $2 \cdot \lfloor \|R\| / (\|R\| - \|S\|) \rfloor - 1$  arestas. Da existência deste caminho com peso um em  $G'_{R+S}$ , temos a existência de um caminho em G com igual comprimento e peso maior ou igual a um.

■

**Corolário 2.2:** Sejam  $G = (V, E)$  um grafo finito, simples e bipartido,  $R$  um b-emparelhamento máximo e  $S$  um b-emparelhamento não máximo de  $G$ , então o comprimento do menor caminho S-aumentante é no máximo:  $(2 \cdot b / (\|R\| - \|S\|)) - 1$ , onde  $b = \min \left( \sum [b_i : v_i \in V^+], \sum [b_i : v_i \in V^-] \right)$ .

**Demonstração:**  $\|S\| < \|R\| \leq b$

Como, demonstrado no Teorema 2.6, existe em  $G$  um caminho S-aumentante com comprimento no máximo igual a:

$$2 \cdot \lfloor \|R\| / (\|R\| - \|S\|) \rfloor - 1 \leq (2 \cdot b / (\|R\| - \|S\|)) - 1.$$

■

**Teorema 2.7:** Sejam  $R$  um b-emparelhamento de  $G$ ,  $P$  um menor caminho R-aumentante com peso  $\alpha$  em  $G$  e  $P'$  um caminho R'-aumentante, sendo  $R' = R + \alpha * P$ , então  $|P'| \geq |P|$ .

**Demonstração:** Trivial se  $P$  e  $P'$  não têm arestas comuns ou as arestas comuns têm mesma paridade.

Se pelo menos uma das arestas comuns tem paridade diferente em  $P$  e em  $P'$ , seja  $P \cap P' = C_1 \cup C_2 \cup \dots \cup C_k$ , onde os  $C_i$ 's são caminhos R-alternantes e R'-alternantes maximais. A demonstração é feita por indução finita sobre  $k$ .

Se  $k=1$ ,  $P = P_1 \cup C_1 \cup P_2$  e  $P' = P'_1 \cup C_1 \cup P'_2$ , então  $P_1 \cup P'_1$  e  $P_2 \cup P'_2$  são caminhos R-aumentantes, então  $|P'_1| \geq |C_1 \cup P_2|$  e  $|P'_2| \geq |C_1 \cup P_1|$  logo:  $|P'| \geq |P| + 2 \cdot |C_1|$ .

Hipótese de indução: se  $P \cap P'$  tem no máximo  $k-1$  caminhos R-alternantes e R'-alternantes maximais então  $|P'| \geq |P|$

A seguir, a demonstração é dividida em duas partes: na primeira é verificada a validade do teorema quando  $C_k$  possui um número par de arestas, na segunda, quando  $C_k$  possui um número ímpar de arestas.

- Se  $C_k$  possui um número par de arestas, seja  $G^* = (V^*, E^*)$  uma contração de  $G$  tal que  $C_k$  se torna apenas um vértice e são retiradas de  $G$ , todas as arestas que contêm um vértice interno de  $C_k$ . Contraíndo  $C_k$  em  $P$  e em  $P'$  temos  $P^*$  e  $P'^*$ , que são caminhos respectivamente,  $(R - C_k)$ -aumentante e  $(R' - C_k)$ -aumentante em

$G^*$  e  $P^* \cap P^{*'} = C_1 \cup C_2 \dots C_{k-2} \cup C_{k-1}$ , logo pela hipótese de indução  $|P^{*'}| \geq |P^*|$ , portanto:  $|P^{*'} \cup C_k| \geq |P^* \cup C_k|$  ou  $|P'| \geq |P|$ .

- Se  $C_k$  tem um número ímpar de arestas, e seus vértices extremos são  $v_r$  e  $v_s$ .

Se  $P \equiv C_k$  é trivial, consideremos então  $P \neq C_k$ , isto é em  $P$  existe um vértice  $v_l$  adjacente a  $v_r$  (ou a  $v_s$ ) e em  $P'$ , também existe um vértice  $v_k$  adjacente ao mesmo vértice ( $P$  é um menor caminho R-aumentante de  $G$ ).

Seja  $G^* = (V^*, E^*)$  uma contração de  $G$  tal que  $C = C_k \cup \{v_r, v_l\} \cup \{v_r, v_k\}$  torna-se um único vértice e são retiradas todas as arestas que contêm um vértice interno de  $C$ . Contraindo  $C_k \cup \{v_r, v_l\}$  em  $P$  e  $C_k \cup \{v_r, v_k\}$  em  $P'$  obtemos  $P^*$  e  $P^{*'}$ .  $P^*$  é o menor caminho R-aumentante em  $G^*$ ,  $P^{*'}$  é um caminho R'-aumentante em  $G^*$  e  $P^* \cap P^{*' } = C_1 \cup C_2 \dots \cup C_{k-2} \cup C_{k-1}$ , então, pela hipótese de indução,  $|P^{*' }| \geq |P^*|$ , logo

$$|P^{*' } \cup C_k \cup \{v_r, v_l\}| \geq |P^* \cup C_k \cup \{v_r, v_k\}|, \text{ isto é: } |P'| \geq |P|$$

■

**Teorema 2.8:** Sejam  $R$  um b-emparelhamento máximo de  $G$  e  $P_0, P_1, \dots, P_i$  uma seqüência de menores caminhos aumentantes que determinam  $R$ , então o número de inteiros distintos na seqüência  $|P_0|, |P_1|, \dots, |P_i|$  é no máximo  $2 \cdot \lfloor (\|R\|)^{\frac{1}{2}} \rfloor - 1$ .

**Demonstração:** Seja  $S$  o último b-emparelhamento que satisfaz  $\|S\| \leq \lfloor \|R\| - \|R\|^{\frac{1}{2}} \rfloor$ , obtido na construção de  $R$ , utilizando a seqüência  $P_0, P_1, \dots, P_j$  de menores caminhos aumentantes. Existe um caminho S-aumentante com comprimento menor ou igual a:

$$2 \cdot \lfloor \|R\| / (\|R\| - \|S\|) \rfloor - 1 \leq 2 \cdot \lfloor \|R\| / (\|R\| - \lfloor \|R\| - (\|R\|)^{\frac{1}{2}} \rfloor) \rfloor - 1 \leq 2 \cdot \lfloor \|R\| / (\|R\|)^{\frac{1}{2}} \rfloor - 1 = 2 \cdot \lfloor (\|R\|)^{\frac{1}{2}} \rfloor - 1$$

Então  $|P_0|, |P_1|, \dots, |P_j|, |P_{j+1}|$  são menores ou iguais a  $2 \cdot \lfloor (\|R\|)^{\frac{1}{2}} \rfloor - 1$  e são todos inteiros e ímpares, formando uma seqüência de no máximo  $\lfloor (\|R\|)^{\frac{1}{2}} \rfloor$  inteiros distintos entre si.

Se  $P_{j+2}, P_{j+3}, \dots, P_i$  são caminhos aumentantes com peso um temos  $\|R\| - \lfloor \|R\| - \|R\|^{\frac{1}{2}} \rfloor - 1$  caminhos aumentantes, e se estes caminhos têm comprimentos todos diferentes, temos o mesmo número de inteiros distintos na seqüência  $|P_{j+2}|, |P_{j+3}|, \dots, |P_i|$ .

A seqüência  $|P_0|, |P_1|, \dots, |P_k|$  tem um número de inteiros distintos menor ou igual a :  
 $\lfloor \|R\|^{\frac{1}{2}} \rfloor + \|R\| - \lfloor \|R\| - \|R\|^{\frac{1}{2}} \rfloor - 1 \leq 2 \cdot \lfloor (\|R\|)^{\frac{1}{2}} \rfloor - 1$

■

**Teorema 2.9 :** Sejam  $R$  um b-emparelhamento máximo de  $G$  e  $S$  um b-emparelhamento de  $G$ , e  $P_1, P_2, \dots, P_k$  uma seqüência de caminhos aumentantes, com pesos respectivamente  $\alpha_1, \alpha_2, \dots, \alpha_k$ , que podem ser utilizados para determinar  $R$  a partir de  $S$  então :  $\|R\| = \|S\| + \sum [\alpha_i : i=1, 2, \dots, k]$

**Demonstração :** Por indução finita em  $k$  ( número de caminhos ).

Para  $k = 1$  foi demonstrado no Teorema 3.

Hipótese de indução : Se  $R$  é um b-emparelhamento máximo obtido a partir de um b-emparelhamento  $S'$  com o uso de  $(k-1)$  caminhos aumentantes com pesos, respectivamente,  $\alpha_1', \alpha_2', \dots, \alpha_{k-1}'$ , então :

$$\|R\| = \|S'\| + \sum_{i=1}^{k-1} \alpha_i'$$

$S^* = S + \alpha_1 * P_1$  é um b-emparelhamento e  $\|S^*\| = \|S\| + \alpha_1$ .

$R$  pode ser obtido a partir de  $S^*$  com o uso dos caminhos aumentantes  $P_2, P_3, \dots, P_k$  com pesos, respectivamente,  $\alpha_2, \alpha_3, \dots, \alpha_k$ , então, pela hipótese de indução :

$$\|R\| = \|S^*\| + \sum_{i=2}^k \alpha_i, \quad \text{logo} \quad \|R\| = \|S\| + \sum_{i=1}^k \alpha_i.$$

■

**Teorema 2.10 :** A soma dos comprimentos de todos os caminhos aumentantes minimais utilizados na determinação do b-emparelhamento máximo é no máximo  $b + 2 \cdot b \cdot \log_2 b$ , onde  $b = \min \{ \sum [b_i : v_i \in V^+], \sum [ |b_j| : v_j \in V^- ] \}$ .

**Demonstração :** Se  $R$  é o b-emparelhamento máximo de  $G$ , e  $P_1, P_2, \dots, P_k$  é uma seqüência de menores caminhos aumentantes que podem ser utilizados para a determinação de  $R$  e  $S_i$  é o b-emparelhamento obtido após a utilização de  $P_1, P_2, \dots, P_{i-1}$ , então pelo Corolário 2.2 tem-se :  $|P_i| \leq (2 \cdot b / (\|R\| - \|S_i\|)) - 1$ .

De  $k \leq \|R\| \leq b$ , tem-se:

$$\begin{aligned} \sum_{i=1}^k |P_i| &\leq 2 \cdot b/b^{-1} + 2 \cdot b/(b-1)^{-1} + \dots + 2 \cdot b/2^{-1} + 2 \cdot b/1^{-1} = \\ &= -b + 2 \cdot b \cdot [1/1 + 1/2 + 1/3 + \dots + 1/b] \leq -b + 2 \cdot b \cdot \left( 1 + \int_1^b dx/x \right) = \\ &= b + 2 \cdot b \cdot \log_2 b \end{aligned}$$

■

**Teorema 2.11 :** O Algoritmo A proposto para o problema de b-emparelhamento máximo em um grafo bipartido tem complexidade:  $O((\min(|V^+|, |V^-|, b^{1/2}) \cdot m + b \cdot \log_2 b)$ , onde  $b = \min\{\sum [b_i : v_i \in V^+], \sum [b_j : v_j \in V^-]\}$ .

**Demonstração :**

- O número máximo de referentes calculados pelo algoritmo está limitado pelo número máximo de caminhos aumentantes de comprimentos diferentes. Este número é certamente no máximo igual a  $|V^+| + |V^-|$ , uma vez que um caminho aumentante pode ter  $1, 2, \dots$  ou  $|V^+| + |V^-|$  vértices de  $V^+ \cup V^-$ .

Sendo  $R$  um b-emparelhamento máximo, então  $\|R\| \leq b$ , onde  $b = \min\{\sum [b_i : v_i \in V^+], \sum [b_j : v_j \in V^-]\}$ , pelo Teorema 2.8, o número de caminhos aumentantes com comprimentos diferentes no algoritmo é no máximo igual a  $2 \cdot b^{1/2} - 1$ .

O número de vezes que o referente é calculado é limitado por  $\min(|V^+|, |V^-|, 2 \cdot b^{1/2} - 1)$ .

- O cálculo de cada referente é feito examinando, para cada vértice  $v_i \in V$ , todas as arestas que contêm  $v_i$ . Toda aresta é examinada no máximo duas vezes, uma devido a cada um de seus vértices. O número de arestas examinadas na construção de um referente é, portanto, no máximo igual a  $2 \cdot m$ .

- Na determinação de um caminho aumentante  $P$  no referente e seu peso, toda aresta do referente é examinada como candidata a entrar no caminho ou entra no caminho ou sai do referente. O número de arestas examinadas na construção dos caminhos de um referente e que não entram nos caminhos é limitado superiormente por  $m$ .

- A soma dos comprimentos de todos os caminhos aumentantes minimais determinados é, como demonstrado no Teorema 2.10, no máximo igual a  $b + 2 \cdot b \cdot \log_2 b$ . Este número é então,

um limite superior do número total de vezes, que arestas ao serem examinadas entram em caminhos aumentantes e, também, do número total de aumentos no b-emparelhamento de arestas.

A complexidade do Algoritmo A de b-emparelhamento máximo em um grafo bipartido é dada por :

Número de referentes calculados \* [número de arestas examinadas na construção de cada referente + número de arestas examinadas e não utilizadas na construção dos caminhos de um referente] + número de arestas examinadas e utilizadas na construção dos caminhos aumentantes + número de atualizações de b-emparelhamento das arestas

$$\leq \min(|V^+|, |V^-|, 2 \cdot b^{\frac{1}{2}} + 1) \cdot 3 \cdot m + 2 \cdot (b + 2 \cdot b \cdot \log_2 b) =$$

$$O((\min(|V^+|, |V^-|, b^{\frac{1}{2}}) \cdot m + b \cdot \log_2 b) \cdot b)$$

■

Complexidade do Algoritmo A :  $O((\min(|V^+|, |V^-|, b^{\frac{1}{2}}) \cdot m + b \cdot \log_2 b) \cdot b)$

Onde  $b = \min(\sum [b_i : v_i \in V^+], \sum [|b_j| : v_j \in V^-])$ .

A título de comparação, Gabow e Tarjan [GbT88] apresentam um algoritmo para o cálculo do subgrafo com restrição dos graus dos vértices de um multigrafo bipartido, a partir do qual pode-se, também obter o b-emparelhamento máximo, mas a complexidade do algoritmo seria:

$$O(b'^{\frac{1}{2}} \cdot m' + b' \cdot \log_2 b')$$

onde  $b' = \sum [b_i : i=1, 2, \dots, n]$  e  $m' = \sum [\min\{b_j : v_j \in e_i\} : i=1, 2, \dots, m]$ .

## 2.7- Algoritmo B de b-emparelhamento máximo bipartido.

### 2.7.1- Introdução.

A base do algoritmo a ser apresentado são as propriedades de problemas de programação linear. (Como visto na secção 2.4,  $G^*$  é um grafo bipartido, logo a matriz dos coeficientes de  $D^*$  é totalmente unimodular, então qualquer solução básica de  $D^*$  é inteira e satisfaz  $D$ .)

É um algoritmo iterativo em que um problema de b-emparelhamento máximo bipartido é

resolvido em cada iteração. O problema de b-emparelhamento bipartido a ser resolvido em cada iteração é obtido a partir de escalonamento do valor de  $B_i = \max \{ |b_i| : v_i \in V \}$ .

Métodos iterativos para solução de problemas de redes, utilizando escalonamento de um dos parâmetros do problema, foram inicialmente apresentados por Edmonds e Karp [EdK72]. Recentemente foram apresentados alguns trabalhos utilizando este tipo de procedimento para, principalmente, o problema de fluxo de custo mínimo.

A validade dos algoritmos, com escalonamento de um dos parâmetros do problema, é fortemente baseada no Teorema de ortogonalidade de soluções ótimas de problemas duais e na linearidade da função objetivo e do conjunto de restrições.

A condição de folgas complementares, para o problema de b-emparelhamento definido em  $G^*$ , como consequência do Teorema de ortogonalidade de soluções ótimas de problemas duais, é:

“Se  $z^0$  é um b-emparelhamento viável de  $G^*$ , então  $z^0$  é solução ótima de  $D$  se e somente se existe  $x^0$  solução viável de  $P$  tal que:

$$\begin{aligned} (b_i - \sum_j [z^0_{ij} : (v_i, v_j) \in E^*]) \cdot (1 - x^0_i) &= 0, \text{ para todo } v_i \in V^+, \\ (|b_j| - \sum_i [z^0_{ij} : (v_i, v_j) \in E^*]) \cdot x^0_j &= 0, \text{ para todo } v_j \in V^-, \text{ e} \\ z^0_{ij} \cdot (x^0_i - x^0_j) &= 0, \text{ para todo } (v_i, v_j) \in E^*. \end{aligned}$$

Uma relaxação da condição de folgas complementares, dita condição de  $\beta$ -otimalidade, para o problema de b-emparelhamento bipartido é:

**Definição:** Dado  $\beta \geq 0$ , um b-emparelhamento viável  $z^0$  de  $G^*$  é dito  $\beta$ -ótimo, se e somente se existe uma solução viável  $x^0$  de  $P_\beta$ , tal que:

$$\begin{aligned} (b_i - \sum_j [z^0_{ij} : (v_i, v_j) \in E^*]) \cdot (1 - x^0_i) &= 0, \text{ para todo } v_i \in V^+ \text{ com } b_i \geq \beta, \\ (|b_j| - \sum_i [z^0_{ij} : (v_i, v_j) \in E^*]) \cdot x^0_j &= 0, \text{ para todo } v_j \in V^- \text{ com } |b_j| \geq \beta, \text{ e} \\ z^0_{ij} \cdot (x^0_i - x^0_j) &= 0, \text{ para todo } (v_i, v_j) \in E^*. \end{aligned}$$

Dado um número inteiro positivo  $\beta$ , seja  $D^*(\beta)$  o seguinte problema de programação linear, que é associado com o problema de b-emparelhamento bipartido definido no grafo  $G^*$ :

$$\begin{aligned}
D^*(\beta) : \quad & \text{máx } g(z) = \sum [z_{ij} : (v_i, v_j) \in E^*] \\
& \text{s.a. } \sum_j [z_{ij} : (v_i, v_j) \in E^*] \leq \lfloor |b_i|/\beta \rfloor \quad \text{para todo } v_i \in V^+ \\
& \quad \quad \quad \sum_i [z_{ij} : (v_i, v_j) \in E^*] \leq \lfloor |b_j|/\beta \rfloor \quad \text{para todo } v_j \in V^- \\
& \quad \quad \quad z_{ij} \geq 0 \quad \quad \quad \text{para todo } (v_i, v_j) \in E^*
\end{aligned}$$

### 2.7.2- Algoritmo B de b-emparelhamento máximo bipartido.

Algoritmo B para determinação da solução ótima do problema de b-emparelhamento bipartido com escalonamento de  $B_1 = \text{máx}\{|b_i| : v_i \in V\}$ .

$$\begin{aligned}
\text{I) Determinar: } \quad & B_1 := \text{máx}\{|b_i| : v_i \in V\} ; \quad \beta := 2^{\lfloor \log_2 B_1 \rfloor} ; \\
& z^0 := (0, 0, \dots, 0)^T ;
\end{aligned}$$

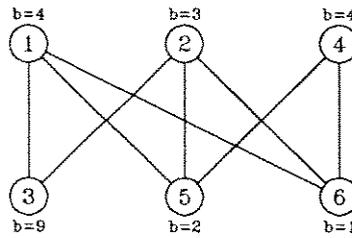
II) Enquanto  $\beta \geq 1$  fazer:

- 1) Determinar  $D^*(\beta)$  e utilizando  $z^0/\beta$  como solução inicial determinar  $z^*$ , solução ótima de  $D^*(\beta)$ ;
- 2)  $z^0 := \beta \cdot z^*$  ;
- 3)  $\beta := \beta/2$  .

### 2.7.3- Exemplo de resolução do problema de b-emparelhamento máximo bipartido utilizando o Algoritmo B.

$$\begin{aligned}
D: \quad & \text{máx } g(z) = z_{13} + z_{15} + z_{16} + z_{23} + z_{25} + z_{26} + z_{45} + z_{46} \\
& \text{s.a. } z_{13} + z_{15} + z_{16} \leq 4 \\
& \quad \quad \quad z_{23} + z_{25} + z_{26} \leq 3 \\
& \quad \quad \quad z_{45} + z_{46} \leq 4 \\
& \quad \quad \quad z_{13} + z_{23} \leq 9 \\
& \quad \quad \quad z_{15} + z_{25} + z_{45} \leq 2 \\
& \quad \quad \quad z_{16} + z_{26} + z_{46} \leq 1 \\
& \quad \quad \quad z_{ij} \geq 0 \quad \quad \quad \text{para todo } (v_i, v_j) \in E^* \\
& \quad \quad \quad z_{ij} \text{ inteiro} \quad \quad \quad \text{para todo } (v_i, v_j) \in E^*
\end{aligned}$$

Este problema pode ser representado no grafo abaixo.



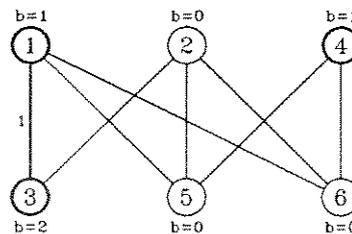
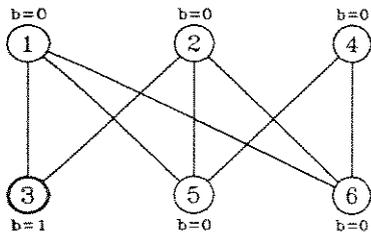
I)  $B_1 = 9$ ,  $\beta = 8$  e

$$z^0_{13} = z^0_{15} = z^0_{16} = z^0_{23} = z^0_{25} = z^0_{26} = z^0_{45} = z^0_{46} = 0$$

II)

1)  $\beta = 8$  e o grafo correspondente a  $D^*(8)$ .

2)  $\beta = 4$  e o grafo correspondente a  $D^*(4)$ .



A solução ótima de  $D^*(8)$  é:

$$z^*_{13} = z^*_{23} = z^*_{45} = z^*_{46} = z^*_{15} = z^*_{16} = z^*_{25} = z^*_{26} = 0$$

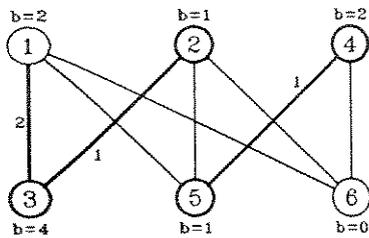
$$z^0_{13} = z^0_{23} = z^0_{45} = z^0_{46} = z^0_{15} = z^0_{16} = z^0_{25} = z^0_{26} = 0$$

A solução ótima de  $D^*(4)$  é:

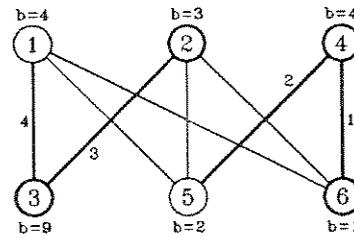
$$z^*_{13} = 1 \text{ e } z^*_{23} = z^*_{45} = z^*_{46} = z^*_{15} = z^*_{16} = z^*_{25} = z^*_{26} = 0, \text{ logo}$$

$$z^0_{13} = 4 \text{ e } z^0_{23} = z^0_{45} = z^0_{46} = z^0_{15} = z^0_{16} = z^0_{25} = z^0_{26} = 0$$

3)  $\beta = 2$  e grafo correspondente a  $D^*(2)$ .



4)  $\beta = 1$  e grafo correspondente a  $D^*(1)$ .



A solução ótima de  $D^*(2)$  é:

$$z^*_{13}=2, z^*_{23}=1, z^*_{45}=1 \quad \text{e} \quad z^*_{46}=z^*_{15}=z^*_{16}=z^*_{25}=z^*_{26}=0, \text{ logo}$$

$$z^0_{13}=4, z^0_{23}=2, z^0_{45}=2 \quad \text{e} \quad z^0_{46}=z^0_{15}=z^0_{16}=z^0_{25}=z^0_{26}=0$$

A solução ótima de  $D^*(1)$  é:

$$z^*_{13}=4, z^*_{23}=3, z^*_{45}=2, z^*_{46}=1 \quad \text{e} \quad z^*_{15}=z^*_{16}=z^*_{25}=z^*_{26}=0, \text{ e}$$

$D^*(1) = D^*$  logo  $z^0_{13} = 4, z^0_{23} = 3, z^0_{45} = 2, z^0_{46} = 1$  e  $z^0_{15} = z^0_{16} = z^0_{25} = z^0_{26} = 0$  é uma solução ótima do problema  $D$  de b-emparelhamento máximo bipartido.

#### 2.7.4- Validade e complexidade do Algoritmo B de b-emparelhamento máximo bipartido.

A validade do algoritmo é demonstrada no Teorema 2.14. O Teorema 2.13 demonstra que o algoritmo apresentado preserva a  $\beta$ -otimalidade.

**Lema 2.1:** A solução viável  $z^0$  de  $D^*$  obtida no final de uma iteração tem componentes  $z^0_{ij}$  múltiplos do  $\beta$  desta iteração, para todo  $(v_i, v_j) \in E^*$ .

Conseqüência imediata do algoritmo. ■

**Teorema 2.12:** Sejam  $\beta$  um número inteiro positivo e  $\beta \cdot z^0$  uma solução viável de  $D^*$ . Se utilizando  $z^0$  como solução inicial, uma solução ótima  $(z^*)$  de  $D^*(\beta)$  é obtida então  $\beta \cdot z^*$  é solução viável de  $D^*$  e  $g(\beta \cdot z^0) \leq g(\beta \cdot z^*)$ .

**Demonstração:** A viabilidade de  $\beta \cdot z^*$  é consequência imediata de  $z^*$  ser viável de  $D^*(\beta)$ , ou seja:

$$\begin{aligned} \sum_j [z^*_{ij} : (v_i, v_j) \in E^*] &\leq \lfloor |b_i|/\beta \rfloor \quad \text{para todo } v_i \in V^+, \text{ implica em} \\ \sum_j [\beta \cdot z^*_{ij} : (v_i, v_j) \in E^*] &\leq \beta \cdot \lfloor |b_i|/\beta \rfloor \leq |b_i| \quad \text{para todo } v_i \in V^+ \text{ e} \\ \sum_i [z^*_{ij} : (v_i, v_j) \in E^*] &\leq \lfloor |b_j|/\beta \rfloor \quad \text{para todo } v_j \in V^-, \text{ implica em} \\ \sum_i [\beta \cdot z^*_{ij} : (v_i, v_j) \in E^*] &\leq \beta \cdot \lfloor |b_j|/\beta \rfloor \leq |b_j| \quad \text{para todo } v_j \in V^- \\ \text{e } z^*_{ij} \geq 0 \text{ com } \beta > 0 &\text{ implicam em } \beta \cdot z^*_{ij} \geq 0. \end{aligned}$$

O acréscimo do valor da função objetivo  $g$  é consequência de  $z^0$  ser solução viável de  $D^*(\beta)$  e da linearidade de  $g$ , ou seja:

$$g(\beta \cdot z^*) = \beta \cdot g(z^*) \geq \beta \cdot g(z^0) = g(\beta \cdot z^0)$$

■

**Teorema 2.13:** Sejam  $z^0$  uma solução viável de  $D^*$ ,  $\beta$  um número inteiro positivo e  $z^*$  a solução ótima obtida de  $D^*(\beta)$ . Se  $x^*$  é uma solução ótima do problema  $P^*(\beta)$ , problema de seleção bipartida associado ao dual de  $D^*(\beta)$ , determinada a partir de  $z^*$  então  $\beta \cdot z^*$  é  $\beta$ -ótimo com relação a  $x^*$ .

**Demonstração:**  $z^*$  é solução ótima obtida de  $D^*(\beta)$  e  $x^*$  é a solução ótima do problema  $P^*(\beta)$  dual de  $D^*(\beta)$ , portanto:

$$(\lfloor |b_i|/\beta \rfloor - \sum_j [z^*_{ij} : (v_i, v_j) \in E^*]) \cdot (1 - x^*_i) = 0, \text{ para todo } v_i \in V^+,$$

$$(\lfloor |b_j|/\beta \rfloor - \sum_i [z^*_{ij} : (v_i, v_j) \in E^*]) \cdot x^*_j = 0, \text{ para todo } v_j \in V^-,$$

$$\text{e } z^*_{ij} \cdot (x^*_i - x^*_j) = 0, \text{ para todo } (v_i, v_j) \in E^*$$

logo  $v_i \in V^+$  e  $(\lfloor |b_i|/\beta \rfloor - \sum_j [z^*_{ij} : (v_i, v_j) \in E^*]) > 0$  implicam em  $x^*_i = 1$ ,

mas se  $v_i \in V^+$  e  $\beta \leq (|b_i| - \sum_j [\beta \cdot z^*_{ij} : (v_i, v_j) \in E^*]) =$

$$\beta \cdot (\lfloor |b_i|/\beta \rfloor - \sum_j [z^*_{ij} : (v_i, v_j) \in E^*]) + \text{resto da divisão } (|b_i|/\beta)$$

com  $0 \leq \text{resto da divisão } (|b_i|/\beta) < \beta$ , então

$$(\lfloor |b_i|/\beta \rfloor - \sum_j [z^*_{ij} : (v_i, v_j) \in E^*]) > 0, \text{ o que implica que } x^*_i = 1$$

Da mesma forma, demonstra-se que se  $v_j \in V^-$  e

$$\beta \leq (|b_j| - \sum_i [\beta \cdot z^*_{ij} : (v_i, v_j) \in E^*]), \text{ então } x^*_j = 0.$$

A terceira condição  $\beta \cdot z^*_{i,j} \cdot (x^*_i - x^*_j) = 0$ , para todo  $(v_i, v_j) \in E^*$  para  $\beta \cdot z^*$  ser  $\beta$ -ótimo com relação a  $x^*$  é obviamente satisfeita. ■

**Teorema 2.14 :** Se  $z^0$  é o b-emparelhamento ótimo de  $D^*(1)$  determinado pelo algoritmo então,  $z^0$  é um b-emparelhamento máximo de  $D^*$ .

**Demonstração :** Por definição  $D^*(1) = D^*$ . ■

### Definições :

- A iteração do algoritmo em que a solução ótima de  $D^*(\beta)$  é determinada, é chamada iteração  $\beta$ .

- Sendo  $z^0$  a solução ótima de  $D^*(2 \cdot \beta)$ , um vértice  $v_i \in V^+$  com  $|b_i| - \sum_j [z^0_{i,j} : (v_i, v_j) \in E^*] \geq 2 \cdot \beta$  é chamado vértice velho da iteração  $\beta$ , caso contrário  $v_i$  é chamado vértice novo da iteração  $\beta$ .

- Sendo  $z^0$  a solução ótima de  $D^*(2 \cdot \beta)$ , um vértice  $v_j \in V^-$  com  $|b_j| - \sum_i [z^0_{i,j} : (v_i, v_j) \in E^*] \geq 2 \cdot \beta$  é chamado vértice velho da iteração  $\beta$ , caso contrário  $v_j$  é chamado vértice novo da iteração  $\beta$ .

**Teorema 2.15 :** Se  $\beta_k = 2^{\lfloor \log_2 B_1 \rfloor} / 2^{(k-1)}$  com  $1 \leq 2^{(k-1)} \leq 2^{\lfloor \log_2 B_1 \rfloor}$  e  $B_1 := \max\{|b_i| : v_i \in V\}$ , e  $R_k (R_{k-1})$  é a aproximação do b-emparelhamento de  $G$  obtido após o cálculo do b-emparelhamento máximo de  $D^*(\beta_k) (D^*(\beta_{k-1}))$ , então  $\|R_k\| \leq \|R_{k-1}\| + n \cdot \beta_k$ .

**Demonstração :** A ampliação do b-emparelhamento  $R_{k-1} / \beta_k$ , utilizado como inicial na iteração  $\beta_k$  pode ser feita através de caminhos aumentantes entre :

- um vértice velho e um vértice novo da iteração  $\beta_k$ ;
- dois vértices novos da iteração  $\beta_k$ ;
- dois vértices velhos da iteração  $\beta_k$ .

Nos dois primeiros tipos de caminho foram utilizados vértices novos e a ampliação do b-emparelhamento de  $D^*(\beta_k)$  é de 1.

Entre dois vértices velhos da iteração  $\beta_k$ , não havia no final da iteração  $\beta_{k-1}$  um caminho aumentante; se existe um caminho aumentante antes do final da iteração  $\beta_k$  é devido a uma aresta par deste caminho ter seu b-emparelhamento aumentado durante esta iteração. Este aumento pode ter sido obtido após a utilização da aresta como aresta ímpar em um caminho aumentante de um dos dois primeiros tipos acima descritos. Mas, se o caminho que alterou o b-emparelhamento da aresta é do primeiro tipo, existe a necessidade de outro caminho aumentante antes destes, ainda na iteração  $\beta_k$ , pois uma de suas arestas pares tinha b-emparelhamento nulo ao fim da iteração  $\beta_{k-1}$ . Esta necessidade de outro caminho aumentante ser utilizado antes pode eventualmente formar uma cadeia de exigências, que só termina após a utilização de um caminho do segundo tipo (entre dois vértices novos).

O número de caminhos utilizados é portanto menor ou igual ao número de vértices novos que é menor que o número de vértices. O aumento do b-emparelhamento de  $D^*(\beta_k)$ , durante a iteração  $\beta_k$  é no total menor que o número de vértices. Ao sair da iteração  $\beta_k$  o valor do b-emparelhamento obtido é multiplicado por  $\beta_k$ , e portanto

$$\|R_k\| \leq (\|R_{k-1}\| / \beta_k + n) \cdot \beta_k, \text{ ou seja } \|R_k\| \leq \|R_{k-1}\| + n \cdot \beta_k.$$

■

**Lema 2.2 :** Sejam  $\beta_k = 2^{\lfloor \log_2 B_1 \rfloor} / 2^{(k-1)}$  com  $1 \leq 2^{(k-1)} \leq 2^{\lfloor \log_2 B_1 \rfloor}$  e  $B_1 := \max\{|b_i| : v_i \in V\}$ , e  $R_k$  é a aproximação do b-emparelhamento de  $G$  obtido após o cálculo do b-emparelhamento máximo de  $D^*(\beta_k)$ . Se o b-emparelhamento inicial  $R_0$  do método é igual a zero então  $\|R_k\| \leq n \cdot \log_2 B_1$ .

**Demonstração :** Pelo teorema 2.15, segue-se que:

$$\|R_i\| \leq \|R_{i-1}\| + n \cdot \beta_i \quad \text{para } i=1, 2, \dots, k, \text{ logo}$$

$$\sum_{i=1}^k (\|R_i\| - \|R_{i-1}\|) \leq n \cdot \left( \sum_{i=1}^k \beta_i \right), \quad \text{mas}$$

$$\sum_{i=1}^k (\|R_i\| - \|R_{i-1}\|) = \|R_k\| - \|R_0\| = \|R_k\| \quad \text{e}$$

$$n \cdot \left( \sum_{i=1}^k \beta_i \right) = n \cdot 2^{\lfloor \log_2 B_1 \rfloor + 1} \cdot \left( \sum_{i=1}^k 2^{-i} \right) \leq n \cdot 2^{\lfloor \log_2 B_1 \rfloor} \cdot 2^{\log_2 2} \leq n \cdot \lfloor \log_2 B_1 \rfloor$$

logo  $\|R_k\| \leq n \cdot \log_2 B_1$

■

**Teorema 2.16:** Sejam  $\beta_k = 2^{\lfloor \log_2 B_1 \rfloor / 2^{(k-1)}}$  com  $1 \leq 2^{(k-1)} \leq 2^{\lfloor \log_2 B_1 \rfloor}$  e  $B_1 := \max\{|b_i| : v_i \in V\}$ , e  $R_k (R_{k-1})$  é a aproximação do b-emparelhamento de  $G$  obtido após o cálculo do b-emparelhamento máximo de  $D^*(\beta_k) (D^*(\beta_{k-1}))$  e  $P_0, P_1, \dots, P_i$  uma seqüência de menores caminhos aumentantes que determinam  $R_k / \beta_k$  a partir de  $R_{k-1} / \beta_k$ , então o número de inteiros distintos na seqüência  $|P_0|, |P_1|, \dots, |P_i|$  é no máximo  $\lfloor \lfloor n^{1/2} \rfloor \cdot (\log_2 B_1 + 1) \rfloor$ .

**Demonstração:** Pelo Lema 2.2, tem-se  $\|R_k\| \leq n \cdot \log_2 B_1$

Seja  $S$  o último b-emparelhamento que satisfaz  $\|S\| \leq \|R_k / \beta_k\| - \lfloor n^{1/2} \rfloor$ , na seqüência de b-emparelhamentos obtidos na construção de  $R_k / \beta_k$  a partir de  $R_{k-1} / \beta_k$  utilizando a seqüência  $P_0, P_1, \dots, P_j$  de menores caminhos aumentantes.

Pelo Teorema 2.6 existe em  $G_B(\beta)$  um caminho S-aumentante com comprimento menor ou igual a:

$$\begin{aligned} &\leq 2 \cdot \lfloor \|R_k / \beta_k\| / (\|R_k / \beta_k\| - \|S\|) \rfloor - 1 \leq \\ &\leq 2 \cdot \lfloor (\|R_k / \beta_k\| / (\|R_k / \beta_k\| - \|R_k / \beta_k\| + \lfloor n^{1/2} \rfloor)) \rfloor - 1 = \\ &= 2 \cdot \lfloor \|R_k\| / (\beta_k \cdot \lfloor n^{1/2} \rfloor) \rfloor - 1 \leq \\ &\leq 2 \cdot \lfloor n \cdot \log_2 B_1 / \beta_k \cdot \lfloor n^{1/2} \rfloor \rfloor - 1 = \\ &= 2 \cdot \lfloor n \cdot (\log_2 B_1) \cdot 2^{(k-1)} / 2^{\lfloor \log_2 B_1 \rfloor} \cdot \lfloor n^{1/2} \rfloor \rfloor - 1 \leq \\ &\leq 2 \cdot \lfloor n \cdot (\log_2 B_1) \cdot (\log_2 B_1) / \log_2 B_1 \cdot \lfloor n^{1/2} \rfloor \rfloor - 1 = \\ &= 2 \cdot \lfloor \lfloor n^{1/2} \rfloor \cdot \log_2 B_1 \rfloor - 1. \end{aligned}$$

Então  $|P_0|, |P_1|, \dots, |P_j|, |P_{j+1}|$  são menores ou iguais a  $2 \cdot \lfloor \lfloor n^{1/2} \rfloor \cdot \log_2 B_1 \rfloor - 1$  e são todos inteiros e ímpares, formando uma seqüência de no máximo  $\lfloor \lfloor n^{1/2} \rfloor \cdot \log_2 B_1 \rfloor$  inteiros distintos entre si.

Se  $P_{j+2}, P_{j+3}, \dots, P_i$  são caminhos aumentantes com peso um temos  $\|R(\beta) / \beta\| - \|R(\beta) / \beta\| + \lfloor n^{1/2} \rfloor - 1 = \lfloor n^{1/2} \rfloor - 1$  caminhos aumentantes, e se estes

caminhos têm comprimentos todos diferentes, temos o mesmo número de inteiros distintos na seqüência  $|P_{j+2}|, |P_{j+3}|, \dots, |P_i|$ .

A seqüência  $|P_0|, |P_1|, \dots, |P_i|$  tem um número de inteiros distintos menor ou igual a:

$$\lfloor \lfloor n^{1/2} \rfloor \cdot \log_2 B_1 + \lfloor n^{1/2} \rfloor \rfloor = \lfloor \lfloor n^{1/2} \rfloor \cdot (\log_2 B_1 + 1) \rfloor.$$

■

**Teorema 2.17 :** O Algoritmo B para o problema de b-emparelhamento máximo em um grafo bipartido (que utiliza escalonamento de  $B_1 = \max\{|b_i| : v_i \in V\}$ ) tem complexidade :  $O((\log_2 B_1) \cdot ((\log_2 B_1) \cdot n^{1/2} \cdot m + n^2))$ .

### Demonstração :

- O número de iterações do algoritmo (número de b-emparelhamentos calculados) é  $(\log_2 B_1 + 1)$ .

- Na iteração  $\beta$ , o valor do b-emparelhamento calculado é limitado superiormente por  $n \cdot \beta$ , então o número de caminhos aumentantes, utilizados nesta iteração, é limitado por  $n$ . Como cada caminho é formado por no máximo  $n-1$  arestas temos no máximo  $n \cdot (n-1)$  ajustes no b-emparelhamento de arestas por iteração.

- Pelo Teorema 2.16, o número máximo de referentes a calcular na iteração  $\beta$  é  $\lfloor \lfloor n^{1/2} \rfloor \cdot (\log_2 B_1 + 1) \rfloor$ . Como o cálculo de cada referente é feito através do exame de todas as arestas incidentes em cada vértice, no máximo  $2 \cdot m$  arestas são examinadas na construção de um referente. O total de arestas examinadas na construção dos referentes da iteração  $\beta$  é limitado superiormente por

$$2 \cdot \lfloor \lfloor n^{1/2} \rfloor \cdot (\log_2 B_1 + 1) \rfloor \cdot m.$$

A complexidade do algoritmo é portanto :

$$O((\log_2 B_1) \cdot ((\log_2 B_1) \cdot n^{1/2} \cdot m + n^2)).$$

■

Complexidade do Algoritmo B : $O((\log_2 B_1) \cdot ((\log_2 B_1) \cdot n^{1/2} \cdot m + n^2))$
--

onde  $B_1 = \max\{|b_i| : v_i \in V\}$ .

## 2.8- Algoritmo II de resolução do problema de seleção.

Seja  $G = (V, E)$  um digrafo qualquer com  $|V| = n$  vértices e  $|E| = m$  arcos, em que a cada vértice  $v_i$  é associado um valor inteiro  $b_i$ .

### 2.8.1- Algoritmo II para determinar uma solução ótima do problema de seleção.

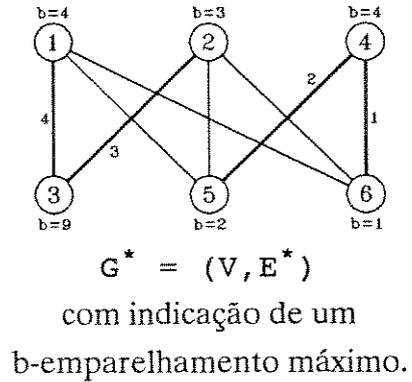
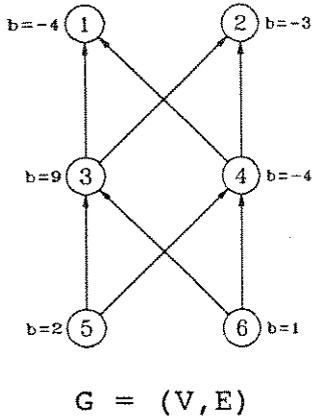
- I) Construir  $G^*$ , subgrafo bipartido do fecho transitivo de  $G$  com  $n = |V^+| + |V^-|$  vértices e no máximo  $|V^+| \cdot |V^-|$  arestas.
- II) Calcular um b-emparelhamento máximo de  $G^*$ .
- III) Determinar uma solução ótima do problema de seleção  $P$  a partir da solução ótima de  $D^*$ .

### 2.8.2- Exemplo de resolução do problema de seleção utilizando o Algoritmo II.

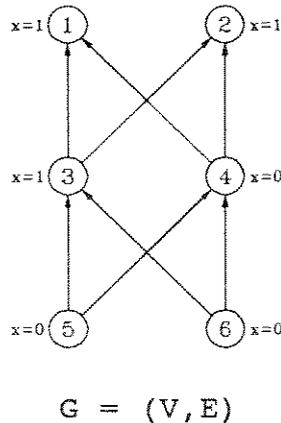
$$\begin{aligned}
 P: \quad \text{máx } f(x) &= -4 \cdot x_1 - 3 \cdot x_2 + 9 \cdot x_3 - 4 \cdot x_4 + 2 \cdot x_5 + x_6 \\
 \text{s.a. } x_3 &\leq x_1 \\
 x_4 &\leq x_1 \\
 x_3 &\leq x_2 \\
 x_4 &\leq x_2 \\
 x_5 &\leq x_3 \\
 x_6 &\leq x_3 \\
 x_5 &\leq x_4 \\
 x_6 &\leq x_4 \\
 x_i &= 0, 1 \quad i \in \{1, 2, 3, 4, 5, 6\}
 \end{aligned}$$

Este problema de seleção é representado no digrafo abaixo.

I e II)



III) Não estão saturados os vértices  $v_3$  e  $v_4$ , então  $x_3 = 1$  e  $x_4 = 0$ . De  $x_3 = 1$  tem-se que  $x_1 = x_2 = 1$  e de  $x_4 = 0$  tem-se que  $x_5 = x_6 = 0$ . Uma solução ótima do problema de seleção  $P$  é, portanto,  $x = (1, 1, 1, 0, 0, 0)^T$ , ou seja os vértices a selecionar são  $v_1, v_2$  e  $v_3$ .



com indicação da seleção ótima.

### 2.8.3- Complexidade do Algoritmo II do problema de seleção.

A complexidade do Algoritmo II de seleção em um digrafo qualquer depende do algoritmo utilizado na resolução do problema de b-emparelhamento máximo bipartido, e é dada por:

número de arestas examinadas na determinação de  $G^*$  + número de operações no cálculo do b-emparelhamento máximo de  $G^*$  + número de operações para transformar a solução do b-emparelhamento na solução do problema de seleção =

I) (utilizando o Algoritmo A para cálculo do b-emparelhamento máximo de  $G^*$ )

$$O(\min(|V^+|, |V^-|) \cdot m) + O(\min(|V^+|, |V^-|, b^{\frac{1}{2}}) \cdot |V^+| \cdot |V^-| + b \cdot \log_2 b) + O(n+m) = \\ O(\min(|V^+|, |V^-|) \cdot m + \min(|V^+|, |V^-|, b^{\frac{1}{2}}) \cdot |V^+| \cdot |V^-| + b \cdot \log_2 b)$$

II) (utilizando o Algoritmo B para cálculo do b-emparelhamento máximo de  $G^*$ )

$$O(\min(|V^+|, |V^-|) \cdot m) + O((\log_2 B_1) \cdot ((\log_2 B_1) \cdot n^{\frac{1}{2}} \cdot |V^+| \cdot |V^-| + n^2)) + O(n+m) = \\ = O(\min(|V^+|, |V^-|) \cdot m + (\log_2 B_1)^2 \cdot (n^{\frac{1}{2}} \cdot |V^+| \cdot |V^-|) + (\log_2 B_1) \cdot n^2)$$

A complexidade do Algoritmo II para o problema de seleção é :

Algoritmo II de seleção com Algoritmo A de b-emparelhamento

$$O(\min(|V^+|, |V^-|) \cdot m + \min(|V^+|, |V^-|, b^{\frac{1}{2}}) \cdot |V^+| \cdot |V^-| + b \cdot \log_2 b)$$

Algoritmo II de seleção com Algoritmo B de b-emparelhamento

$$O(\min(|V^+|, |V^-|) \cdot m + (\log_2 B_1)^2 \cdot (n^{\frac{1}{2}} \cdot |V^+| \cdot |V^-|) + (\log_2 B_1) \cdot n^2)$$

onde  $b = \min(\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-])$  e  $B_1 = \max\{|b_i| : v_i \in V\}$

De acordo com o método proposto por Picard [Pic76], e aplicando o algoritmo de fluxo máximo mais apropriado para o grafo do problema, a complexidade é :  $O(\min(n^3, n \cdot m \cdot \log_2(n^2/m), n^{5/3} \cdot m^{2/3}))$ .

## CAPÍTULO 3.

### PROBLEMA DE SELEÇÃO GENERALIZADA EM UM DIGRAFO.

#### 3.1- Uma generalização do problema de seleção.

Seja  $G = (V, E)$ , um digrafo com um conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$  e um conjunto de arcos  $E = \{e_1, e_2, \dots, e_m\}$ , onde cada arco é um par ordenado de vértices distintos. A cada arco  $e_k = (v_i, v_j)$  é associado um valor inteiro não negativo  $c_{ij}$ , e a cada vértice  $v_i$  são associados um valor inteiro  $b_i$  e dois limitantes inteiros  $l_i$  (limitante inferior) e  $u_i$  (limitante superior).

Neste digrafo  $G$ , as definições de fecho de um digrafo, dadas no capítulo 2, admitem a generalização apresentada a seguir.

#### Definições :

- Um vetor de inteiros não negativos  $X = (x_1, x_2, \dots, x_n)$ , onde cada componente  $x_i$  está associada a um vértice  $v_i$  do digrafo  $G$ , representa um **fecho generalizado** de  $G$  se para todo arco  $(v_i, v_j)$  de  $E$  :  $x_i - x_j \leq c_{ij}$ , e para todo vértice  $v_i$  de  $V$  :  $l_i \leq x_i \leq u_i$ .

- Valor de um fecho generalizado é :  $\sum [b_i \cdot x_i : i=1, 2, \dots, n]$ .

- O **problema do fecho generalizado ótimo** consiste em determinar um fecho generalizado de valor máximo.

Associando a cada vértice  $v_i$  uma variável  $x_i$ , o problema de fecho generalizado ótimo pode ser formulado como o problema de determinar um vetor de números inteiros não negativos  $X = (x_1, x_2, \dots, x_n)$  solução ótima de :

$$\begin{array}{ll}
 P: & \max \quad f(x) = \sum_{i=1}^n b_i \cdot x_i \\
 & \text{s.a. } x_i - x_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E \\
 & \quad 1_i \leq x_i \leq u_i \quad \text{para todo vértice } v_i \in V \\
 & \quad x_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V
 \end{array}$$

Assim,  $x_j$  é igual a um número inteiro e indica que o vértice  $v_j$  deve ser selecionado este número inteiro de vezes. Sendo que  $x_j$  igual a um número inteiro  $a_j$  e  $(v_i, v_j) \in E$  implicam em  $x_i \leq c_{ij} + a_j$ .

O problema  $P$ , quando  $u_i = 1$  e  $1_i = 0$  para todo vértice  $v_i \in V$  e  $c_{ij} = 0$  para todo arco  $(v_i, v_j) \in E$  é chamado problema de seleção e é apresentado no capítulo 2.

O problema de seleção consiste na determinação de um subconjunto de vértices  $V^*$  de soma  $\sum [b_i : v_i \in V^*]$  máxima, tal que se  $v_i \in V^*$  então todos os antecessores de  $v_i$  também pertencem a  $V^*$ .

O problema  $P$  é, portanto, uma generalização do problema de seleção, e é chamado, neste trabalho, de **problema de seleção generalizada**.

Se o problema  $P$  admite uma solução viável  $x^0$ , é possível transformá-lo em um problema de seleção generalizada com  $c_{ij} \geq 0$ ,  $1_i \leq 0$ , e  $u_i \geq 0$ , através da substituição da variável  $x$  por  $x + x^0$ .

O problema  $P$  pode, também, ser transformado no problema  $P_1$ , dado abaixo, após a mudança de variáveis  $y_i = x_i - 1_i$ , a substituição da constante  $u_i - 1_i$  por  $a_i$  ( $a_i \geq 0$ ) e o desprezo, na função objetivo da constante  $\sum [b_i \cdot 1_i : i=1, 2, \dots, n]$ .

$$\begin{array}{ll}
 P_1: & \text{máx } f_0(y) = \sum_{i=1}^n b_i \cdot y_i \\
 & \text{s.a. } y_i - y_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E \\
 & \quad 0 \leq y_i \leq a_i \quad \text{para todo vértice } v_i \in V \\
 & \quad y_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V
 \end{array}$$

Nas restrições de  $P_1$ , podem acontecer os seguintes três casos :

$$1) \quad c_{ij} \geq a_i$$

Neste caso, uma vez que  $y_i - y_j \leq c_{ij}$ ,  $y_i \leq a_i$  e  $0 \leq y_j$  a restrição  $y_i - y_j \leq c_{ij}$  é redundante.

$$2) \quad c_{ij} < -a_j$$

Neste caso, uma vez que  $y_i - y_j \leq c_{ij}$ ,  $y_j \leq a_j$  e  $0 \leq y_i$ , o problema  $P$  é inviável, mas este caso não ocorre, porque se  $P$  é viável, podemos considerar  $c_{ij} \geq 0$  e  $a_j \geq 0$ .

$$3) \quad c_{ij} + a_j \leq a_i$$

Neste caso, uma vez que,  $y_i - y_j \leq c_{ij}$ ,  $y_j \leq a_j$  e  $y_i \leq a_i$ , a restrição  $y_i \leq a_i$  é redundante.

Sejam  $V^+ = \{v_i : b_i \geq 0\}$  e  $V^- = \{v_j : b_j < 0\}$ . Se em  $P_1$ ,  $|V^+| > |V^-|$  então, através da mudança de variáveis  $y_i = a_i - z_i$ , o problema  $P_1$  é transformado em um problema do mesmo tipo com  $|V^+| \leq |V^-|$ .

Como consequência, o estudo do problema  $P$ , sem perda de generalidade, pode ser restrito a:

$$\left[ \begin{array}{ll} l_i = 0 & \text{para } i=1, 2, \dots, n, \\ 0 \leq c_{ij} \leq u_i \leq c_{ij} + u_j & \text{para todo arco } (v_i, v_j) \in E \text{ e} \\ |V^+| \leq |V^-| & \text{onde } V^+ = \{v_i : b_i \geq 0\} \text{ e } V^- = \{v_i : b_i < 0\} \end{array} \right.$$

E com estas restrições,  $P$  tem  $x^0 = (0, 0, \dots, 0)^T$  como solução viável.

Um dos motivos da importância deste problema de seleção generalizada é a relação existente entre este problema e o dual dos seguintes problemas clássicos: fluxo de custo mínimo em um digrafo, b-emparelhamento máximo de custo máximo em um grafo bipartido (subgrafo máximo restrito em grau de custo máximo de um multigrafo bipartido) e como caso particular deste último problema, o emparelhamento de custo máximo bipartido (designação de custo máximo).

O problema de determinação do cronograma de execução de um projeto composto de diversas tarefas, de maneira a minimizar o custo total do projeto pode ser formulado como um problema

de seleção generalizada, onde cada tarefa é representada por dois vértices, um para o horário de início e outro para o horário de término da tarefa.

Neste capítulo, são demonstradas relações existentes, entre as soluções ótimas de um problema de seleção generalizada e as dos seguintes problemas clássicos de programação inteira :

- problema de b-emparelhamento máximo de custo máximo em um grafo bipartido (subgrafo máximo restrito em grau de custo máximo de um multigrafo bipartido);
- problema de emparelhamento bipartido de custo máximo (designação de custo máximo);
- problema de fluxo de custo mínimo em um digrafo.

Neste capítulo são também formulados, como problema de seleção generalizada o problema de determinação de cronograma de execução de um projeto composto de diversas tarefas, de maneira a minimizar o custo total do projeto.

### 3.2- O problema de b-emparelhamento bipartido de custo máximo.

Esta seção está subdividida em três partes. Na primeira parte, seção 3.2.1, é apresentada a definição e formulação do problema. Na seção 3.2.2 é mostrado como determinar uma solução ótima do problema de b-emparelhamento de custo máximo bipartido através da resolução de um problema de seleção generalizada bipartido. Na seção 3.2.3 é mostrado como determinar uma solução ótima do problema de seleção generalizada bipartido através da resolução de um problema de b-emparelhamento de custo máximo bipartido.

#### 3.2.1- Definição e formulação do problema.

Seja  $G^* = (V, E^*)$  um grafo bipartido, onde  $V = (V^+, V^-)$ ;  $\{v_i, v_j\} \in E^*$  implica  $(v_i \in V^+ \text{ e } v_j \in V^-)$  ou  $(v_i \in V^- \text{ e } v_j \in V^+)$ ; a cada vértice  $v_i \in V$  é associado um inteiro não negativo  $|b_i|$ , e a cada aresta  $\{v_i, v_j\} \in E^*$  é associado um inteiro não negativo  $h_{ij}$ .

## Definições :

- $R=(r_1, r_2, \dots, r_m)$  é um b-emparelhamento de  $G^*$ , quando cada  $r_j$  é um inteiro não negativo, associado a aresta  $e_j$ , satisfazendo :  

$$\sum [r_j : j=1, 2, \dots, m \text{ e } v_i \in e_j] \leq |b_i| \quad \text{para todo } v_i \in V$$
- Valor de um b-emparelhamento  $R$  é:  $\|R\| = \sum [r_i : i=1, 2, \dots, m]$ .
- O problema de b-emparelhamento máximo consiste em determinar um b-emparelhamento de valor máximo.
- Custo de um b-emparelhamento  $R$  é :  

$$|R| = \sum [h_{j_k} \cdot r_i : e_i = (v_j, v_k) \text{ e } i=1, 2, \dots, m].$$
- O problema de b-emparelhamento de custo máximo consiste em determinar um b-emparelhamento de custo máximo.

Associando a cada aresta  $(v_i, v_j)$  de  $E^*$  uma variável  $z_{ij}$ , o problema de b-emparelhamento de custo máximo no grafo  $G^*$ , tem a seguinte formulação:

$$\begin{aligned}
 D: \quad & \text{máx } g(z) = \sum [h_{ij} \cdot z_{ij} : v_i \in V^+, v_j \in V^- \text{ e } (v_i, v_j) \in E^*] \\
 & \text{s.a. } \sum [z_{ij} : v_j \in V^- \text{ e } (v_i, v_j) \in E^*] \leq |b_i| \quad \text{para todo } v_i \in V^+ \\
 & \quad \sum [z_{ij} : v_i \in V^+ \text{ e } (v_i, v_j) \in E^*] \leq |b_j| \quad \text{para todo } v_j \in V^- \\
 & \quad z_{ij} \geq 0 \quad \text{para todo } (v_i, v_j) \in E^* \\
 & \quad z_{ij} \text{ inteiro} \quad \text{para todo } (v_i, v_j) \in E^*
 \end{aligned}$$

### 3.2.2- Obtenção de um b-emparelhamento bipartido de custo máximo a partir de uma seleção generalizada de custo máximo.

Seja  $D^*$ , definido em  $G^*$ , um programa linear igual ao problema  $D$  sem a restrição  $z_{ij}$  inteiro.

Como  $G^*$  é um grafo bipartido, a matriz dos coeficientes das restrições de  $D^*$  é totalmente unimodular, e como as constantes do problema  $D$  são inteiras, então qualquer solução básica do programa linear  $D^*$  é inteira e é uma solução viável de  $D$ .

Seja  $P^*$  o dual do problema  $D^*$ , isto é:

$$\begin{aligned}
 P^*: \quad \min f_1(y) &= \sum_{i=1}^n |b_i| \cdot y_i \\
 \text{s.a. } y_i + y_j &\geq h_{ij} && \text{para toda aresta } (v_i, v_j) \in E^* \\
 y_i &\geq 0 && \text{para todo vértice } v_i \in V
 \end{aligned}$$

As restrições do tipo  $y_i + y_j \geq h_{ij}$  podem ser escritas sempre de tal forma, que a primeira variável ( $y_i$ ) corresponda a um vértice ( $v_i$ ) de  $V^+$  e a segunda ( $y_j$ ), a um vértice de  $V^-$ ; isto é sempre possível uma vez que,  $G^*$  é bipartido com bipartição  $V^+, V^-$ .

Seja  $u_i = \max \{h_{ij} : v_j \in V^-\}$  para todo  $v_i \in V^+$  e  $c_{ij} = u_i - h_{ij}$ , logo  $c_{ij}$  é por definição, inteiro não negativo.

Através da mudança de variáveis:

$$y_i = \begin{cases} u_i - x_i & \text{se } v_i \in V^+ \text{ e} \\ x_i & \text{se } v_i \in V^- \end{cases}$$

multiplicação da função objetivo por  $(-1)$ , transformando o problema de minimização em um problema de maximização, e desprezo, na função objetivo obtida, da constante:  $-\sum [ |b_i| \cdot u_i : v_i \in V^+ ]$ ;  $P^*$  pode ser reescrito como o problema  $P_1$ .

$$\begin{aligned}
 P_1: \quad \max f(x) &= \sum [ |b_i| \cdot x_i : v_i \in V^+ ] - \sum [ |b_i| \cdot x_i : v_i \in V^- ] \\
 \text{s.a. } x_i - x_j &\leq c_{ij} && \text{para toda } (v_i, v_j) \in E^* \\
 x_i &\geq 0 && \text{para todo } v_i \in V^- \\
 x_i &\leq u_i && \text{para todo } v_i \in V^+
 \end{aligned}$$

Como  $G^*$  é um grafo bipartido, a matriz dos coeficientes das restrições de  $P_1$  é totalmente unimodular. Além disto, as constantes do problema  $P_1$  são inteiras então qualquer solução básica do programa linear  $P_1$  é inteira.

Seja  $u_j = \max \{u_i : (v_i, v_j) \in E^*\}$  para todo  $v_j \in V^-$ .

O problema  $P_1$  é um problema de maximização com função objetivo linear com coeficientes  $|b_i|$  não negativos, para  $v_i \in V^+$ , então, as condições  $x_i - x_j \leq c_{ij}$  para toda  $(v_i, v_j) \in E^*$ ,  $c_{ij} \geq 0$  e  $x_j \geq 0$  para todo  $v_j \in V^-$ , implicam que  $x_i \leq x_j + c_{ij}$

para todo  $v_j \in V^-$  e o maior possível, logo  $x_i \geq 0$  para todo  $v_i \in V^+$ .

Da mesma forma, a função objetivo é linear com coeficientes  $-|b_j|$  não positivos, para  $v_j \in V^-$ , então, as condições  $x_i - x_j \leq c_{ij}$  para toda aresta  $(v_i, v_j) \in E^*$  e  $x_i \leq u_i$  para todo  $v_i \in V^+$  implicam que  $x_j \geq x_i - c_{ij}$  para todo  $v_i \in V^+$  e o menor possível, logo  $x_j \leq u_j$  para todo  $v_j \in V^-$ .

Seja  $P$  o problema de seleção generalizada definido no digrafo  $G_B = (V^+ \cup V^-, E_B)$ , onde: a todo arco  $(v_i, v_j) \in E_B$  é associado um inteiro positivo  $c_{ij} = u_i - h_{ij}$ ; a cada vértice  $v_i \in V^+$  são associados um inteiro positivo  $|b_i|$  e o limitante superior  $u_i = \max\{h_{ij} : v_j \in V^-\}$ ; a cada vértice  $v_i \in V^-$  são associados um inteiro negativo  $-|b_i|$  e o limitante superior  $u_j = \max\{u_i : (v_i, v_j) \in E^*\}$ ; e  $E_B = \{(v_i, v_j) : (v_i, v_j) \in E^*, v_i \in V^+ \text{ e } v_j \in V^-\}$ .

Portanto, uma solução ótima de  $P$  é uma solução ótima básica do problema  $P_1$ , obtido a partir do problema dual de  $D^*$ .

Ou seja, a partir da determinação de uma solução ótima de  $P$  é possível determinar uma solução ótima de  $D$ .

### 3.2.3- Obtenção de uma seleção generalizada bipartida de custo máximo a partir de um b-emparelhamento bipartido de custo máximo.

Sejam  $V^+ = \{v_i : b_i \geq 0\}$ ,  $V^- = \{v_i : b_i < 0\}$ ,  $G_B = (V, E_B)$  um digrafo bipartido, onde  $(v_i, v_j) \in E_B$  implica  $v_i \in V^+$  e  $v_j \in V^-$ , a cada vértice  $v_i$  são associados um inteiro não negativo  $u_i$  e um inteiro  $b_i$ , e a cada arco  $(v_i, v_j)$  é associado um inteiro não negativo  $c_{ij}$ .

Seja  $P$  o problema de seleção generalizada no digrafo  $G_B$ , isto é:

$$P: \quad \max \sum_{i=1}^n b_i \cdot x_i$$

$$\text{s. a. } x_i - x_j \leq c_{ij}$$

$$0 \leq x_i \leq u_i$$

$$x_i \text{ inteiro}$$

$$\text{para todo arco } (v_i, v_j) \in E_B$$

$$\text{para todo vértice } v_i \in V$$

$$\text{para todo vértice } v_i \in V$$

Seja  $P_1$ , o problema obtido, a partir de  $P$ , com a mudança de variáveis :

$$y_i = \begin{cases} x_i & \text{se } v_i \in V^- \text{ e} \\ u_i - x_i & \text{se } v_i \in V^+, \end{cases}$$

isto é :

$$P_1: \quad \min \sum_{i=1}^n |b_i| \cdot y_i - \sum [ |b_i| \cdot u_i : v_i \in V^+ ]$$

$$\text{s.a. } y_i + y_j \geq u_i - c_{ij} \quad \text{para todo } (v_i, v_j) \in E_B$$

$$0 \leq y_i \leq u_i \quad \text{para todo } v_i \in V$$

$$y_i \text{ inteiro} \quad \text{para todo } v_i \in V$$

Na secção 3.1, para evitar inviabilidade e redundância, o estudo do problema de seleção generalizada foi restrito a  $0 \leq c_{ij} \leq u_i$ , portanto,  $u_i - c_{ij} \geq 0$ , isto é  $P_1$  é viável.

A matriz dos coeficientes das restrições de  $P_1$  é totalmente unimodular e as constantes de  $P_1$  são inteiras, então qualquer solução básica do problema  $P^*$  de programação linear, dado a seguir, é inteira e satisfaz  $P_1$ .

$$P^*: \quad \min \sum_{i=1}^n |b_i| \cdot y_i$$

$$\text{s.a. } y_i + y_j \geq u_i - c_{ij} \quad \text{para todo } (v_i, v_j) \in E_B$$

$$-y_i \geq -u_i \quad \text{para todo } v_i \in V$$

$$y_i \geq 0 \quad \text{para todo } v_i \in V$$

Seja  $D^*$  o dual do problema  $P^*$ , isto é :

$$D^*: \quad \max \sum [ (u_i - c_{ij}) \cdot z_{ij} : (v_i, v_j) \in E_B ] - \sum_{i=1}^n u_i t_i$$

$$\text{s.a. } \sum [ z_{ij} : v_j \in V^- \text{ e } (v_i, v_j) \in E_B ] - t_i \leq |b_i| \quad \text{para todo } v_i \in V^+$$

$$\sum [ z_{ij} : v_i \in V^+ \text{ e } (v_i, v_j) \in E_B ] - t_j \leq |b_j| \quad \text{para todo } v_j \in V^-$$

$$z_{ij} \geq 0 \quad \text{para todo } (v_i, v_j) \in E_B$$

$$t_i \geq 0 \quad \text{para todo } v_i \in V$$

A matriz dos coeficientes das restrições de  $D^*$  é uma matriz totalmente unimodular. Os dados constantes de  $D^*$  são inteiros, então qualquer solução básica de  $D^*$  é inteira, e satisfaz o problema linear  $D_1$ , onde:

$$\begin{aligned}
 D_1: \quad & \max \sum [(u_i - c_{ij}) \cdot z_{ij} : (v_i, v_j) \in E_B] - \sum_{i=1}^n u_i t_i \\
 \text{s.a.} \quad & \sum [z_{ij} : v_j \in V^- \text{ e } (v_i, v_j) \in E_B] - t_i \leq |b_i| \quad \text{para todo } v_i \in V^+ \\
 & \sum [z_{ij} : v_i \in V^+ \text{ e } (v_i, v_j) \in E_B] - t_j \leq |b_j| \quad \text{para todo } v_j \in V^- \\
 & z_{ij} \geq 0 \quad \text{para todo } (v_i, v_j) \in E_B \\
 & t_i \geq 0 \quad \text{para todo } v_i \in V \\
 & z_{ij} \text{ inteiro} \quad \text{para todo } (v_i, v_j) \in E_B \\
 & t_i \text{ inteiro} \quad \text{para todo } v_i \in V
 \end{aligned}$$

Uma solução viável de  $D_1$  é  $z_{ij} = 0$  para todo  $(v_i, v_j) \in E_B$  e  $t_i = 0$  para todo  $v_i \in V$ .

Dada uma solução viável de  $D_1$ , um aumento de uma unidade no valor de um  $t_i$ , permite um aumento igual em  $z_{ij}$  (ou em  $z_{ji}$ ), podendo haver ou não necessidade de igual aumento em  $t_j$ , para manter viabilidade da solução. A função objetivo tem seu valor alterado de  $u_i - c_{ij} - u_i$ ,  $u_i - c_{ij} - u_j$ , ou ainda,  $u_i - c_{ij} - u_i - u_j$ .

Na secção 3.1, para evitar inviabilidade e redundância, o estudo do problema de seleção generalizada foi restrito à  $u_i - c_{ij} - u_j \leq 0$  então as variações nos valores das variáveis, acima mencionados, correspondem a uma diminuição no valor da função objetivo. Com esta condição, a solução ótima de  $D^*$  tem então:  $t_i = 0$  para todo  $v_i \in V$ , ou seja o problema  $D_1$  fica equivalente ao problema  $D$  de b-emparelhamento bipartido de custo máximo:

$$\begin{aligned}
 D: \quad & \max \sum [c^*_{ij} \cdot z_{ij} : (v_i, v_j) \in E_B] \\
 \text{s.a.} \quad & \sum [z_{ij} : v_j \in V^- \text{ e } (v_i, v_j) \in E_B] \leq |b_i| \quad \text{para todo } v_i \in V^+ \\
 & \sum [z_{ij} : v_i \in V^+ \text{ e } (v_i, v_j) \in E_B] \leq |b_j| \quad \text{para todo } v_j \in V^- \\
 & z_{ij} \geq 0 \quad \text{para todo } (v_i, v_j) \in E_B \\
 & z_{ij} \text{ inteiro} \quad \text{para todo } (v_i, v_j) \in E_B
 \end{aligned}$$

onde  $c^*_{ij} = u_i - c_{ij}$

O problema  $D$  de  $b$ -emparelhamento de custo máximo está definido no grafo bipartido  $G^* = (V^* \cup V^-, E^*)$ , onde a cada vértice  $v_i$  é associado o inteiro positivo  $|b_i|$ , a cada aresta  $(v_i, v_j)$  é associado o inteiro positivo  $c_{ij}^*$  e  $E^* = \{(v_i, v_j) : 1 \leq i, j \leq n \text{ e } (v_i, v_j) \in E_b\}$ .

Portanto, a partir da determinação de uma solução ótima de  $D$  é possível determinar uma solução ótima de  $P$ .

Quando  $b_1 = b_2 = \dots = b_n = 1$ , o problema de  $b$ -emparelhamento bipartido  $D$  é o problema de emparelhamento bipartido de custo máximo (designação de custo máximo) e pode ser transformado (caso particular da secção 3.2.2) em um problema de seleção generalizada bipartido com  $|b_1| = |b_2| = \dots = |b_n| = 1$ .

O problema de seleção generalizada bipartido com  $|b_1| = |b_2| = \dots = |b_n| = 1$  pode, também ser transformado (caso particular da secção 3.2.3) em um problema de emparelhamento bipartido de custo máximo.

### 3.3- O problema de fluxo de custo mínimo em um digrafo.

Esta secção está subdividida em três partes. Na primeira parte, secção 3.3.1, é apresentada a definição e formulação do problema, além de outros problemas relacionados, constantemente a este, na literatura. Na secção 3.3.2 é mostrado como determinar uma solução ótima do problema de fluxo de custo mínimo através da resolução de um problema de seleção generalizada. Na secção 3.2.3 é mostrado como determinar uma solução ótima do problema de seleção generalizada através da determinação de uma solução básica ótima do problema de fluxo de custo mínimo.

#### 3.3.1- Definição e formulação do problema.

O problema de fluxo de custo mínimo é um dos problemas mais fundamentais da teoria de fluxo em rede e tem sido exaustivamente estudado.

Dado um digrafo  $G = (V, E)$ , onde a cada vértice  $v_i$  é associado um número real  $b_i$  (demanda ou suprimento); a cada arco  $(v_i, v_j)$  são associados um número real não negativo  $c_{ij}$  (custo) e dois limitantes reais não negativos  $l_{ij}$  (inferior) e  $u_{ij}$  (superior).

## Definições :

-  $f = (f_1, f_2, \dots, f_m)^T$  é um **fluxo** de  $G$ , quando  $f_k$  é um número real não negativo associado a cada arco  $e_k = (v_i, v_j)$  de  $E$  e satisfaz:

$$\sum_j [f_k : e_k = (v_i, v_j) \in E] - \sum_j [f_k : e_k = (v_j, v_i) \in E] \leq b_i \text{ para todo } v_i \in V$$

-  $f = (f_1, f_2, \dots, f_m)^T$  é um **fluxo canalizado** de  $G$ , quando é um fluxo que satisfaz a condição:

$$l_{ij} \leq f_k \leq u_{ij} \quad \text{para todo arco } e_k = (v_i, v_j) \in E$$

- Os vértices  $v_i$  com  $b_i > 0$  são ditos **vértices de suprimento**; os vértices com  $b_i < 0$  são ditos **vértices de demanda** e os vértices com  $b_i = 0$  são ditos **vértices de transbordo** ou **intermediários**. Sejam  $V^+$  o conjunto dos vértices de suprimento e vértices transbordo, e  $V^-$  o conjunto dos vértices de demanda.

- **Valor** de um fluxo  $f$  é :  $\sum_i [ ( \sum_k [f_k : e_k = (v_i, v_j) \in E] - \sum_k [f_k : e_k = (v_j, v_i) \in E] ) : v_i \text{ é vértice de suprimento} ]$

- **Custo** de um fluxo  $f$  é :  $\sum [c_{ij} \cdot f_k : k=1, 2, \dots, m \text{ e } (v_i, v_j) = e_k]$ .

- O **problema de fluxo máximo** é o problema de determinar um fluxo com valor máximo.

- O **problema de fluxo de custo mínimo** é o problema de determinar um fluxo com custo mínimo.

- **Problema de fluxo de custo mínimo não capacitado** é o problema de fluxo de custo mínimo com limitante inferior  $(l_{ij})$  igual a zero e limitante superior  $(u_{ij})$  igual a infinito para todo arco  $(v_i, v_j)$ .

Associando a cada arco  $(v_i, v_j)$  de  $E$  uma variável  $z_{ij}$ , o problema de fluxo de custo mínimo tem a seguinte formulação :

$$D_f: \quad \begin{aligned} \text{mín } F(z) &= \sum [c_{ij} \cdot z_{ij} : (v_i, v_j) \in E] \\ \text{s.a. } \sum_j [z_{ij} : (v_i, v_j) \in E] - \sum_j [z_{ji} : (v_j, v_i) \in E] &\leq b_i \quad \text{para todo } v_i \in V \\ l_{ij} &\leq z_{ij} \leq u_{ij} \quad \text{para todo } (v_i, v_j) \in E \end{aligned}$$

A formulação acima é encontrada, por exemplo, em Ford e Fulkerson [FdF62].

O problema de fluxo de custo mínimo é definido para parâmetros reais  $(c_{ij}, b_i, l_{ij}, u_{ij})$ ; entretanto, o tipo de expressões utilizadas em sua formulação permite que se estes parâmetros forem racionais, sejam transformados em inteiros.

Muitos autores apresentam  $D^*$ , dada a seguir, como a formulação do problema de fluxo de custo mínimo.

$$D^*: \quad \min F(z) = \sum [c_{ij} \cdot z_{ij} : (v_i, v_j) \in E]$$

$$\text{s.a.} \quad \sum_j [z_{ij} : (v_i, v_j) \in E] - \sum_j [z_{ji} : (v_j, v_i) \in E] = b_i \quad \text{para todo } v_i \in V$$

$$l_{ij} \leq z_{ij} \leq u_{ij} \quad \text{para todo } (v_i, v_j) \in E$$

O problema de fluxo de custo mínimo na formulação  $D^*$  exige para viabilidade que :

- $\sum [b_i : i=0, 1, 2, \dots, n] = 0$ .
- Exista um caminho de todo vértice de  $V^+$  para todo vértice de  $V^-$ .

O problema  $D_f$  pode ser transformado em um problema  $D^*$  definido no digrafo  $G_1 = (V_1, E_1)$ , onde  $V_1 = V \cup \{v_0\}$  com  $b_0 = -\sum [b_i : i=1, 2, \dots, n]$ ;  $E_1 = E \cup \{(v_0, v_i) : v_i \in V^-\} \cup \{(v_j, v_0) : v_j \in V^+\}$ ; aos arcos  $(v_0, v_i)$  com  $v_i \in V^-$  é associado um custo  $c_{0i} = n \cdot \max\{c_{jk} : (v_j, v_k) \in E\}$  e capacidade ilimitada; e aos arcos  $(v_j, v_0)$  com  $v_j \in V^+$  é associado um custo zero e capacidade ilimitada.

O problema  $D^*$  definido sobre o digrafo  $G_1$  satisfaz, pela forma como  $G_1$  foi construído, as duas condições necessárias para viabilidade de  $D^*$ . É fácil determinar a solução ótima de  $D_f$  a partir de qualquer solução ótima de  $D^*$  definido sobre o digrafo  $G_1$ .

Outro problema fortemente relacionado na literatura com o problema de fluxo de custo mínimo é o problema de circulação de custo mínimo. O problema de circulação é o problema de fluxo de custo mínimo  $D^*$ , com  $b_1 = b_2 = \dots = b_n = 0$ .

O problema  $D^*$  pode ser facilmente transformado em um problema de circulação de custo mínimo  $D_c$  definido no digrafo  $G_c = (V_c, E_c)$ , onde  $V_c = V \cup \{v_{n+1}\}$ ;  $E_c = E \cup \{(v_i, v_{n+1}) : v_i \in V^-\} \cup \{(v_{n+1}, v_j) : v_j \in V^+\}$ ; aos arcos  $(v_{n+1}, v_j)$  com  $v_j \in V^+$  é associado um custo zero e capacidade  $b_j$ ; e aos arcos  $(v_i, v_{n+1})$  com  $v_i \in V^-$  é associado um custo zero e capacidade  $-b_i$ .

O problema de circulação de custo mínimo  $D_c$ , definido em  $G_c$ , pode ser escrito como :

$$D_c: \min F_1(z) = \sum [c_{ij} \cdot z_{ij} : (v_i, v_j) \in E_c]$$

$$\text{s.a. } \sum_j [z_{ij} : (v_i, v_j) \in E_c] - \sum_j [z_{ji} : (v_j, v_i) \in E_c] = 0 \quad \text{para todo } v_i \in V_c$$

$$l_{ij} \leq z_{ij} \leq u_{ij} \quad \text{para todo } (v_i, v_j) \in E_c$$

Existe uma transformação bem conhecida, citada por exemplo em Orlin [Orl88], para converter um problema de fluxo de custo mínimo em um problema de fluxo de custo mínimo não capacitado. Esta transformação consiste, como indicado na figura abaixo, em introduzir um vértice  $v_k$  para cada arco  $(v_i, v_j)$  com capacidade  $(u_{ij})$  finita; em substituir cada arco  $(v_i, v_j)$  com capacidade finita por dois arcos:  $(v_i, v_k)$  com custo  $c_{ij}$  e  $(v_j, v_k)$  com custo zero, ambos com capacidade infinita; e em associar ao vértice  $v_i$ , o número  $b_i$ ; ao vértice  $v_j$ , o número  $b_j + u_{ij}$ ; e ao vértice  $v_k$ , o número  $-u_{ij}$ .



Nas secções 3.3.2 e 3.3.3, é considerado o problema de fluxo de custo mínimo não capacitado, com parâmetros inteiros e utilizada a formulação  $D^*$ .

### 3.3.2- Obtenção de um fluxo de custo mínimo a partir de uma seleção generalizada de custo máximo.

Sejam  $G_1 = (V_1, E_1)$  um digrafo e  $D^*$  o problema de fluxo de custo mínimo não capacitado definido em  $G_1$  com parâmetros inteiros e  $\sum [b_i : i=0, 1, 2, \dots, n] = 0$ .

$$D^*: \min F(z) = \sum [c_{ij} \cdot z_{ij} : (v_i, v_j) \in E_1]$$

$$\text{s.a. } \sum_j [z_{ij} : (v_i, v_j) \in E_1] - \sum_j [z_{ji} : (v_j, v_i) \in E_1] = b_i \quad \text{para todo } v_i \in V_1$$

$$z_{ij} \geq 0 \quad \text{para todo } (v_i, v_j) \in E_1$$

Seja  $P^*$  o dual do problema  $D^*$  :

$$P^*: \quad \text{máx} \quad f(x) = \sum_{i=1}^n b_i \cdot x_i$$

$$\text{s.a.} \quad x_i - x_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E_1$$

A matriz das restrições é totalmente unimodular e os dados constantes de  $P^*$  são inteiros, então qualquer solução básica do problema  $P^*$ , de programação linear, é inteira.

Se  $x^*$  é uma solução ótima de  $P^*$  e  $h$  é um número real qualquer então  $x^* + (h, h, h, \dots, h)^T$  é solução ótima de  $P^*$ ; conseqüência imediata de  $\sum [b_i : i=0, 1, 2, \dots, n] = 0$ .

Retirando o vértice  $v_0$  de  $G_1$ ; isto é, fazendo  $x_0 = 0$  temos os vértices  $v_i \in V^+$  com limitante superior igual a  $u = n \cdot \text{máx}\{c_{jk} : (v_j, v_k) \in E_1\}$ ; e os vértices  $v_i \in V^-$  com limitante inferior zero.

O problema  $P^*$  é um problema de maximização com função objetivo linear com coeficientes  $|b_i|$  não negativos, para  $v_i \in V^+$ , então, as condições  $x_i - x_j \leq c_{ij}$  para todo  $(v_i, v_j) \in E_1$ ,  $c_{ij} \geq 0$  e  $x_j \geq 0$  para todo  $v_j \in V^-$ , implicam que  $x_i \leq x_j + c_{ij}$  para todo  $v_j \in V^-$  e o maior possível, logo  $x_i \geq 0$  para todo  $v_i \in V^+$ .

Da mesma forma, a função objetivo é linear com coeficientes  $-|b_j|$  não positivos, para  $v_j \in V^-$ , então, as condições  $x_i - x_j \leq c_{ij}$  para todo  $(v_i, v_j) \in E_1$  e  $x_i \leq u$  para todo  $v_i \in V^+$ , implicam que  $x_j \geq x_i - c_{ij}$  para todo  $v_i \in V^+$  e o menor possível, logo  $x_j \leq u$  para todo  $v_j \in V^-$ .

O problema  $P^*$ , problema dual de  $D^*$ , é, portanto, equivalente ao problema  $P$  de seleção generalizada, definido no digrafo  $G_1 = (V^+ \cup V^-, E_1)$ , onde : a cada vértice  $v_i \in V^+$  são associados um inteiro positivo  $|b_i|$  e o limitante superior  $u = n \cdot \text{máx}\{c_{jk} : (v_j, v_k) \in E_1\}$ ; a cada vértice  $v_i \in V^-$  são associados um inteiro não positivo  $-|b_i|$  e o limitante superior  $u$ ; e a todo arco  $(v_i, v_j) \in E_1$  é associado um inteiro positivo  $c_{ij}$ .

$$P: \quad \text{máx} \quad f(x) = \sum_{i=1}^n b_i \cdot x_i$$

$$\text{s.a.} \quad x_i - x_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E_1$$

$$0 \leq x_i \leq u \quad \text{para todo vértice } v_i \in V_1$$

$$x_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V$$

Portanto, a partir da determinação de uma solução ótima de  $P$  é possível determinar uma solução ótima de  $D^*$ .

### 3.3.3- Obtenção de uma seleção generalizada de custo máximo a partir de um fluxo de custo mínimo.

Seja  $P$  o problema de seleção generalizada no digrafo  $G = (V, E)$ , isto é:

$$\begin{array}{ll}
 P: & \text{máx } f(x) = \sum_{i=1}^n b_i \cdot x_i \\
 & \text{s.a. } x_i - x_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E \\
 & \quad 0 \leq x_i \leq u_i \quad \text{para todo vértice } v_i \in V \\
 & \quad x_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V
 \end{array}$$

Seja  $P_1$  o problema obtido a partir de  $P$  através da mudança de variáveis  $x_i = y_i - y_0$  para  $i = 1, 2, \dots, n$ , onde  $y_0$  é uma variável com valor inteiro não negativo.

$$\begin{array}{ll}
 P_1: & \text{máx } f_1(y) = \sum_{i=1}^n b_i \cdot y_i - \left( \sum_{i=1}^n b_i \right) \cdot y_0 \\
 & \text{s.a. } y_i - y_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E \\
 & \quad y_i - y_0 \leq u_i \quad \text{para todo vértice } v_i \in V \\
 & \quad y_0 - y_i \leq 0 \quad \text{para todo vértice } v_i \in V \\
 & \quad y_0 \geq 0 \\
 & \quad y_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V \text{ ou para } i = 0
 \end{array}$$

Qualquer solução viável de  $P_1$  satisfaz à restrição  $y_i \geq 0$  para  $i = 1, 2, \dots, n$  de  $P$ , visto que satisfaz às restrições  $y_0 - y_i \leq 0$  e  $y_0 \geq 0$ .

Seja  $G_1 = (V_1, E_1)$ , onde  $E_1 = E \cup \{(v_i, v_0) : v_i \in V\} \cup \{(v_0, v_i) : v_i \in V\}$ ,  $V_1 = V \cup \{v_0\}$ , a cada vértice  $v_i$  é associado um valor  $b_i$  (com  $b_0 = - \sum [b_i : i=1, 2, \dots, n]$ ) e a cada arco  $(v_i, v_j)$  é associado um inteiro não negativo  $h_{ij}$  com

$$h_{ij} = \begin{cases} c_{ij} & \text{para todo } (v_i, v_j) \in E \\ u_i & \text{para } v_j = v_0 \\ 0 & \text{para } v_i = v_0 \end{cases}$$

O problema  $P_1$ , é então transformado no problema  $P_2$  de seleção generalizada, definido em  $G_1$ .

$$P_2: \quad \begin{aligned} \text{máx} \quad f_1(y) &= \sum_{i=0}^n b_i \cdot y_i \\ \text{s. a.} \quad y_i - y_j &\leq h_{ij} && \text{para todo arco } (v_i, v_j) \in E_1 \\ y_i &\geq 0 && \text{para todo vértice } v_i \in V_1 \\ y_i &\text{ inteiro} && \text{para todo vértice } v_i \in V_1 \end{aligned}$$

A matriz das restrições de  $P_2$  é totalmente unimodular e os dados constantes de  $P_2$  são inteiros, então qualquer solução básica do problema  $P^*$ , de programação linear, dado a seguir, é inteira e satisfaz o problema  $P_2$  e como consequência  $P_1$ .

$$P^*: \quad \begin{aligned} \text{máx} \quad f_1(y) &= \sum_{i=0}^n b_i \cdot y_i \\ \text{s. a.} \quad y_i - y_j &\leq h_{ij} && \text{para todo arco } (v_i, v_j) \in E_1 \\ y_i &\geq 0 && \text{para todo vértice } v_i \in V_1 \end{aligned}$$

Seja  $D_1$  o dual do problema  $P^*$ , isto é:

$$D_1: \quad \begin{aligned} \text{mín} \quad g(z) &= \sum [h_{ij} \cdot z_{ij} : (v_i, v_j) \in E_1] \\ \text{s. a.} \quad \sum_j [z_{ij} : (v_i, v_j) \in E_1] - \sum_j [z_{ji} : (v_j, v_i) \in E_1] &\geq b_i && \text{para todo } v_i \in V_1 \\ z_{ij} &\geq 0 && \text{para todo } (v_i, v_j) \in E_1 \end{aligned}$$

No problema  $D_1$ , o primeiro conjunto de restrições somado termo a termo exceto por uma das desigualdades, resulta em:

$$\begin{aligned} & \sum [\sum_j [z_{ij} : (v_i, v_j) \in E_1] - \sum_j [z_{ji} : (v_j, v_i) \in E_1] : i=1, 2, \dots, k-1, k+1, \dots, n] \geq \\ & \geq \sum [b_i : i=0, 1, \dots, k-1, k+1, \dots, n] , \text{ mas } \sum [b_i : i=0, 1, \dots, n] = 0 \text{ logo:} \\ & \sum_j [z_{kj} : (v_k, v_j) \in E_1] - \sum_j [z_{jk} : (v_j, v_k) \in E_1] \leq b_k \text{ para todo } v_k \in V_1 , \text{ e} \\ & \text{portanto, } \sum_j [z_{kj} : (v_k, v_j) \in E_1] - \sum_j [z_{jk} : (v_j, v_k) \in E_1] = b_k \text{ para todo } v_k \in V_1 . \end{aligned}$$

O problema  $D_1$  é então equivalente ao problema  $D^*$ , escrito abaixo.

$$\begin{aligned} D^* : \min g(z) &= \sum [h_{ij} \cdot z_{ij} : (v_i, v_j) \in E_1] \\ \text{s. a. } & \sum_j [z_{ij} : (v_i, v_j) \in E_1] - \sum_j [z_{ji} : (v_j, v_i) \in E_1] = b_i \text{ para todo } v_i \in V_1 \\ & z_{ij} \geq 0 \text{ para todo } (v_i, v_j) \in E_1 \end{aligned}$$

O problema  $D^*$ , satisfaz a condição:  $\sum [b_i : i=0, 1, \dots, n] = 0$ .

Com esta condição, o problema  $D^*$  fica equivalente ao problema de fluxo de de custo mínimo não capacitado no digrafo  $G_1 = (V_1, E_1)$ , onde  $E_1 = E \cup \{(v_0, v_i) : v_i \in V\} \cup \{(v_i, v_0) : v_i \in V\}$ ; aos arcos do tipo  $(v_0, v_i)$  com  $v_i \in V$  é associado custo  $u_i$ , aos arcos do tipo  $(v_i, v_0)$  com  $v_i \in V$  é associado custo zero e aos arcos do tipo  $(v_i, v_j) \in E$  é associado custo  $c_{ij}$ .

Portanto, a partir da determinação de uma solução básica ótima de  $D^*$  é possível determinar uma solução ótima de  $P$ .

### 3.4- Problema de determinação do cronograma de execução de um projeto composto de diversas tarefas.

O problema de determinação de cronograma de execução de um projeto composto de diversas tarefas, de maneira a minimizar o custo total do projeto está relacionado com o problema denominado problema de curva de custo de projetos.

Este último problema foi formulado por Kelley e Walker (citado por exemplo em [Fks61]), e é um problema de programação linear para calcular a curva de custo para um projeto composto de algumas tarefas individuais.

Este problema foi estudado por Fulkerson [Fks61], e considera um projeto como um conjunto parcialmente ordenado de tarefas, sendo a ordenação consequência de restrições técnicas, que determinam que certas tarefas devem ser terminadas antes de outras começarem a ser executadas. Para cada tarefa são conhecidos o tempo mínimo necessário para sua execução e o tempo normal (ou máximo) para sua execução. O custo de execução é suposto variar linearmente entre estes dois extremos. Dado que o projeto precisa ser terminado em um prazo pré-estabelecido, deseja-se determinar o tempo de execução de cada tarefa de modo que o custo total (soma dos custos de execução das tarefas) seja o menor possível.

Fulkerson utilizando uma formulação diferente da apresentada a seguir, propõe um algoritmo paramétrico que constroi a curva de custos do projeto; isto é, o custo mínimo para cada prazo viável de execução do projeto. Pode-se provar que esta curva é não decrescente, linear por partes sempre que os custos de execução de cada tarefa forem não decrescentes.

### 3.4.1- Formulação do Problema.

Seja  $G = (V, E)$  um digrafo. Cada tarefa  $i$  do projeto é representada em  $G$  por dois vértices, um indicando a data de início ( $v_i$ ) e outro a data de término ( $v_i'$ ) de execução da tarefa, e pelo arco  $(v_i, v_i')$ . A estes arcos são associados dois limitantes inteiros não negativos  $l_i$  (inferior) e  $u_i$  (superior), que são os tempos mínimo e normal para execução de  $i$ , e um custo  $b_i$ , que é o custo para reduzir de uma unidade o tempo de execução de  $i$ . Para cada relação de ordem indicando que uma tarefa  $i$  deve ser executada antes de uma tarefa  $j$ , existe um arco  $(v_i', v_j)$  em  $E$ , tendo o número zero como limitante inferior.

Se a cada vértice  $v_i$  de  $V$  é associada a variável  $s_i$ , que indica a data de início de  $i$ , a cada vértice  $v_i'$  de  $V$  é associada a variável  $t_i$ , que indica a data de término de  $i$ , e a constante  $T$  representa a duração desejada para execução do projeto, então o problema pode ter a seguinte formulação:

$$\begin{aligned}
 P_1: \quad \min \quad f(t, s) &= \sum_{i=1}^n b_i \cdot (u_i - (t_i - s_i)) \\
 \text{s.a.} \quad l_i &\leq t_i - s_i \leq u_i && \text{para todo } (v_i, v_i') \in E \\
 0 &\leq s_j - t_i && \text{para todo } (v_i', v_j) \in E \\
 t_n - s_1 &\leq T \\
 t_i &\geq 0 && \text{para todo } v_i' \in V \\
 s_i &\geq 0 && \text{para todo } v_i \in V
 \end{aligned}$$

A solução ótima de  $P_1$  deve ter  $t_i - s_i$  o mais próximo possível de  $u_i$  e  $s_j - t_j$  o menor possível, ou seja o mais próximo de zero e  $t_n - s_1$  o mais próximo possível de  $T$ .

Para a viabilidade de  $P_1$  é necessário que qualquer caminho  $v_1 = v_{11}, v'_{i1}, v_{i2}, v'_{i2}, v_{i3}, v'_{i3}, \dots, v_{ir}, v'_{ir} = v'_n$  em  $G$  entre  $v_1$  e  $v'_n$  satisfaça a condição:  $\sum [l_{ij} : j=1, 2, \dots, r] \leq t_n - s_1 \leq T$

### 3.4.2- Transformação em um problema de seleção generalizada.

Sejam  $s^0$  e  $t^0$  uma solução qualquer viável de  $P_1$ ;  $a_i$  uma constante não negativa, definida por  $a_i = (t^0_i - s^0_i)$  e  $P_2$  o problema  $P_1$ , após a mudança de variáveis:  $s = x + s^0$  e  $t = y + t^0$ ; isto é:

$$\begin{aligned}
 P_2: \quad \text{máx} \quad f(t, s) &= \sum_{i=1}^n b_i \cdot y_i - \sum_{i=1}^n b_i \cdot x_i + \sum_{i=1}^n b_i \cdot (u_i - a_i) \\
 \text{s.a.} \quad y_i - x_i &\leq u_i - a_i && \text{para todo } (v_i, v'_i) \in E \\
 x_i - y_i &\leq a_i - l_i && \text{para todo } (v_i, v'_i) \in E \\
 y_i - x_j &\leq s^0_j - t^0_i && \text{para todo } (v'_i, v_j) \in E \\
 y_n - x_1 &\leq T - (t^0_n - s^0_1) \\
 y_i &\geq -t^0_i && \text{para todo } v'_i \in V \\
 x_i &\geq -s^0_i && \text{para todo } v_i \in V
 \end{aligned}$$

**Teorema 3.1:** Uma solução ótima de  $P_2$  pode ser determinada a partir da solução de um problema de seleção generalizada.

**Demonstração:** Da viabilidade de  $s^0$  e  $t^0$  tem-se que:  $l_i \leq t^0_i - s^0_i \leq u_i$ ; logo  $u_i - a_i \geq 0$  e  $a_i - l_i \geq 0$ ; para todo  $(v_i, v'_i) \in E$ ;  $s^0_j - t^0_i \geq 0$  para todo  $(v'_i, v_j) \in E$ ; e  $T - (t^0_n - s^0_1) \geq 0$ .

Por construção,  $P_2$  tem  $x^0 = (0, 0, \dots, 0)^T$  e  $y^0 = (0, 0, \dots, 0)^T$  como solução viável.

Se  $x^*$  e  $y^*$  é uma solução ótima de  $P_2$  e  $\alpha$  é uma constante qualquer positiva, então  $x^* + (\alpha, \alpha, \dots, \alpha)^T$  e  $y^* + (\alpha, \alpha, \dots, \alpha)^T$  é também solução ótima de  $P_2$ ; uma vez que, o conjunto de restrições continua satisfeito e o valor da função objetivo permanece o

mesmo. Portanto, uma solução ótima de  $P_2$  pode ser determinada substituindo em suas duas últimas restrições  $-t^0_i$  e  $-s^0_i$  por zero.

Na função objetivo o termo  $\sum [b_i \cdot (u_i - a_i) : i=1, 2, \dots, n]$  é constante e pode ser retirado sem alterar a solução ótima.

Seja  $G_1 = (V, E_1)$  um digrafo, com  $E_1 = E \cup \{(v_i', v_i) : i=1, 2, \dots, n\}$ , onde a cada arco  $(v_i, v_i') \in E$  está associado um inteiro não negativo  $u_i - a_i$ , a cada arco  $(v_i', v_i) \in E_1$  está associado um inteiro não negativo  $a_i - l_i$ , e a cada arco  $(v_i', v_j) \in E$  está associado um inteiro não negativo  $s^0_j - t^0_i$ .

Existe, então um problema de seleção generalizada  $P_3$ , definido em  $G_1$ , em que não existe limitante superior para as variáveis, e qualquer solução ótima deste problema é uma solução ótima de  $P_2$ . ■

**Corolário 3.1:** Dada uma solução viável de  $P_1$  é possível determinar uma solução ótima de  $P_1$  resolvendo um problema de seleção generalizada.

**Demonstração:** Conseqüência imediata da construção de  $P_2$  e do Teorema 3.1. Dada a linearidade do problema, sem perda de generalidade,  $s_1$  pode ser feito igual a zero; neste caso, todas as variáveis do problema passam a ter como limitante superior  $T$  e a solução ótima de  $P$  pode ser obtida a partir da solução ótima de um problema de seleção generalizada. ■

### 3.4.3- Exemplo de determinação do cronograma de execução de um projeto através do problema de seleção generalizada.

Seja o problema de determinação do cronograma de execução de um projeto composto de um conjunto bem definido de tarefas  $\{1, 2, 3, 4, 5\}$ , com limite de 13 unidades de tempo para a sua completa execução, de modo a minimizar o custo total de execução.

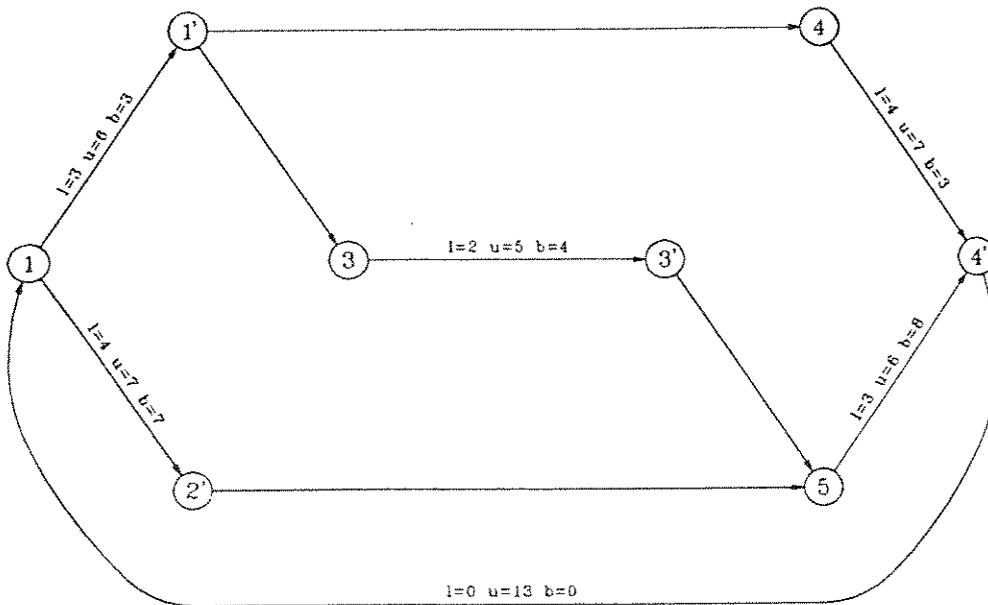
São dadas as seguintes relações de dependência entre as tarefas :

- tarefas 3 e 4 só podem iniciar após o término da tarefa 1;
- tarefa 5 só pode iniciar após o término das tarefas 2 e 3.

São dados, também, para cada tarefa o tempo mínimo, o tempo normal e o custo para diminuir de uma unidade o tempo de execução de cada tarefa :

- tarefa 1 tem tempo mínimo = 3, tempo normal = 6, custo = 3;
- tarefa 2 tem tempo mínimo = 4, tempo normal = 7, custo = 7;
- tarefa 3 tem tempo mínimo = 2, tempo normal = 5, custo = 4;
- tarefa 4 tem tempo mínimo = 4, tempo normal = 7, custo = 3;
- tarefa 5 tem tempo mínimo = 3, tempo normal = 6, custo = 8.

A representação em um digrafo  $G = (V,E)$  e a formulação deste problema são apresentados a seguir.



$$G = (V, E)$$

$$P_1: \min 3 \cdot (6 - t_1 + s_1) + 7 \cdot (7 - t_2 + s_1) + 4 \cdot (5 - t_3 + s_3) + 3 \cdot (7 - t_4 + s_4) + 8 \cdot (6 - t_4 + s_5)$$

$$\text{s.a. } 3 \leq t_1 - s_1 \leq 6$$

$$4 \leq t_2 - s_1 \leq 7$$

$$2 \leq t_3 - s_3 \leq 5$$

$$4 \leq t_4 - s_4 \leq 7$$

$$3 \leq t_4 - s_5 \leq 6$$

$$s_4 - t_1 \geq 0$$

$$s_3 - t_1 \geq 0$$

$$s_5 - t_2 \geq 0$$

$$s_5 - t_3 \geq 0$$

$$t_4 - s_1 \leq 13$$

$$s_i \geq 0$$

para  $i=1, 3, 4, 5$

$$t_i \geq 0$$

para  $i=1, 2, 3, 4$

$$s_i, t_i \text{ inteiros}$$

Uma solução viável deste problema pode ser determinada da seguinte maneira :

$$s^0_1 = 0$$

$$t^0_i = s^0_i + l_i \quad \text{para } i = 1, 2, 3$$

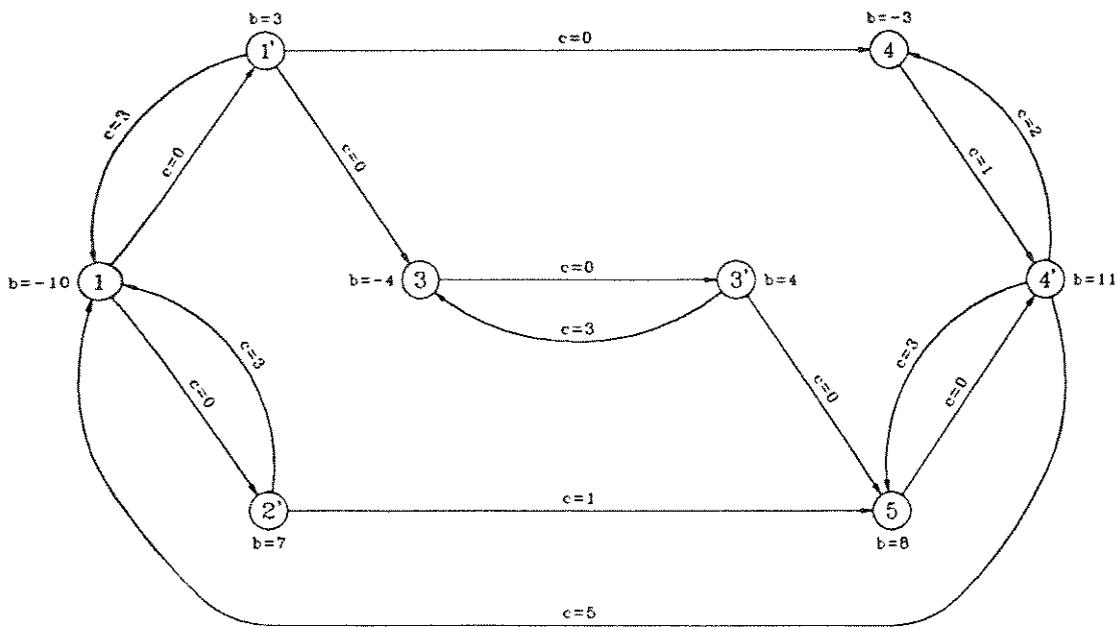
$$t^0_4 = \min (s^0_4 + l_4, s^0_5 + l_5)$$

$$s^0_{i+1} = \max\{t^0_j : (t^0_j, s^0_{i+1}) \in E\}$$

Uma solução viável de  $P_1$  é, portanto:  $s^0_1 = 0$ ;  $t^0_1 = 3$ ;  $t^0_2 = 4$ ;  $s^0_3 = 3$ ;  $s^0_4 = 3$ ;  $t^0_3 = 5$ ;  $s^0_5 = 5$ ;  $t^0_4 = 8$

Após a mudança de variáveis:  $s = x + s^0$  e  $t = y + t^0$ , o problema  $P_1$  se transforma num problema ( $P_2$ ) de seleção generalizada. Este problema  $P_2$  e sua representação em um digrafo  $G_1(V, E_1)$  são dados a seguir.

$$\begin{aligned}
 P_2: \quad & \min 3y_1 + 7y_2 + 4y_3 + 11y_4 - 3x_1 - 7x_2 - 4x_3 - 3x_4 - 8x_5 \\
 & \text{s.a.} \quad y_1 - x_1 \leq 3 \\
 & \quad y_2 - x_1 \leq 3 \\
 & \quad y_3 - x_3 \leq 3 \\
 & \quad y_4 - x_4 \leq 2 \\
 & \quad y_4 - x_5 \leq 3 \\
 & \quad x_1 - y_1 \leq 0 \\
 & \quad x_1 - y_2 \leq 0 \\
 & \quad x_3 - y_3 \leq 0 \\
 & \quad x_4 - y_4 \leq 1 \\
 & \quad x_5 - y_4 \leq 0 \\
 & \quad y_1 - x_4 \leq 0 \\
 & \quad y_1 - x_3 \leq 0 \\
 & \quad y_2 - x_5 \leq 1 \\
 & \quad y_3 - x_5 \leq 0 \\
 & \quad y_4 - x_1 \leq 5 \\
 & \quad x_i \geq 0 \quad \text{para } i=1, 3, 4, 5 \\
 & \quad y_i \geq 0 \quad \text{para } i=1, 2, 3, 4 \\
 & \quad x_i, y_i \text{ inteiros}
 \end{aligned}$$

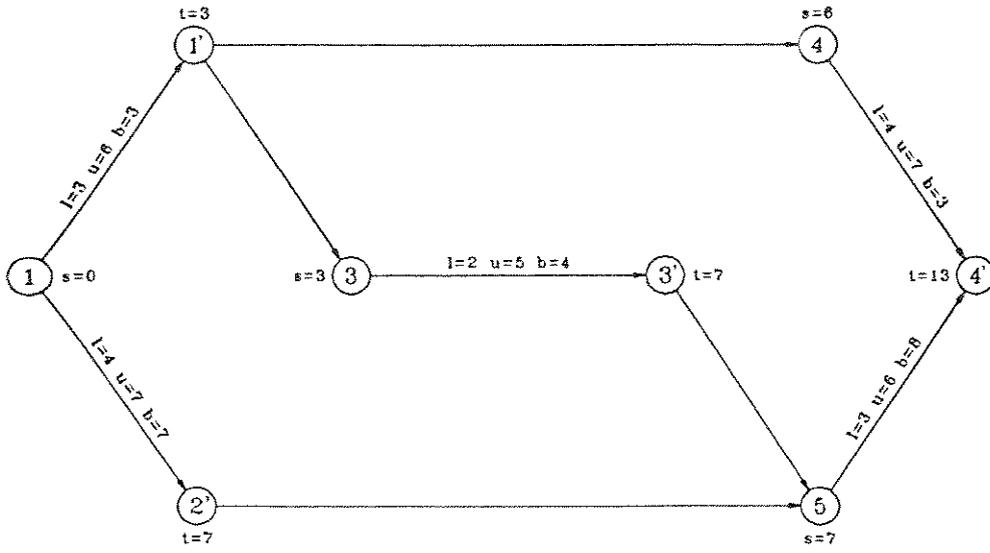


$$G_1 = (V, E_1)$$

Este problema resolvido, por um dos métodos apresentados no capítulo 4, tem como uma de suas soluções ótimas:  $x_1 = 0$ ;  $x_3 = 0$ ;  $x_4 = 3$ ;  $x_5 = 2$ ;  $y_1 = 0$ ;  $y_2 = 3$ ;  $y_3 = 2$ ;  $y_4 = 3$ .

Logo, o problema  $P$  tem como uma de suas soluções ótimas:  $s_1 = 0$ ;  $s_3 = 3$ ;  $s_4 = 6$ ;  $s_5 = 7$ ;  $t_1 = 3$ ;  $t_2 = 7$ ;  $t_3 = 7$ ;  $t_4 = 13$ .

A seguir, o digrafo que representa o problema  $P$ , com indicação de um cronograma ótimo para sua execução. O custo associado com este cronograma é igual a 13.



$$G = (V, E)$$

## CAPÍTULO 4.

### RESOLUÇÃO DO PROBLEMA DE SELEÇÃO GENERALIZADA.

#### 4.1- Problema de seleção generalizada.

Seja  $G = (V, E)$ , um digrafo com um conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$  e um conjunto de arcos  $E = \{e_1, e_2, \dots, e_m\}$ , onde cada arco é um par ordenado de vértices distintos, a cada arco  $e_k = (v_i, v_j)$  é associado um valor inteiro não negativo  $c_{ij}$ , e a cada vértice  $v_i$  são associados um valor inteiro  $b_i$  e dois limitantes inteiros  $l_i$  (limitante inferior) e  $u_i$  (limitante superior).

Associando a cada vértice  $v_i$  uma variável  $x_i$ , deseja-se determinar um vetor de números inteiros não negativos  $X = (x_1, x_2, \dots, x_n)$  solução ótima de :

$$\begin{array}{ll}
 \text{P:} & \text{máx } f(x) = \sum_{i=1}^n b_i \cdot x_i \\
 & \text{s.a. } x_i - x_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E \\
 & \quad \quad \quad l_i \leq x_i \leq u_i \quad \text{para todo vértice } v_i \in V \\
 & \quad \quad \quad x_i \text{ inteiro} \quad \quad \quad \text{para todo vértice } v_i \in V
 \end{array}$$

O estudo do problema acima pode, sem perda de generalidade, como visto no capítulo 3, ser restrito à :

$$\left[ \begin{array}{ll}
 l_i = 0 & \text{para } i=1, 2, \dots, n, \\
 0 \leq c_{ij} \leq u_i \leq c_{ij} + u_j & \text{para todo arco } (v_i, v_j) \in E \text{ e} \\
 |V^+| \leq |V^-| & \text{onde } V^+ = \{v_i : b_i \geq 0\} \text{ e } V^- = \{v_i : b_i < 0\}
 \end{array} \right.$$

A importância deste problema é baseada, como apresentado no capítulo 3, entre outros, na relação existente entre ele e o dual dos seguintes problemas clássicos : fluxo de custo mínimo em um digrafo, b-emparelhamento de custo máximo em um grafo bipartido (subgrafo restrito em grau de custo máximo de um multigrafo bipartido), emparelhamento de custo máximo bipartido (designação de custo máximo).

Problema de seleção, apresentado no capítulo 2, é o problema  $P$ , quando  $c_{ij} = 0$  para todo arco  $(v_i, v_j) \in E$  e  $l_i = 0$ ,  $u_i = 1$  para todo vértice  $v_i \in V$ . O problema de seleção consiste na determinação de um subconjunto de vértices  $V^*$  de soma  $\sum [b_i : v_i \in V^*]$  máxima, tal que se  $v_i \in V^*$  então todos os antecessores de  $v_i$  também pertencem a  $V^*$ .

Algoritmos para determinação da solução ótima do problema de seleção em um digrafo a partir da solução ótima de um problema de seleção em um digrafo bipartido e deste a partir da solução ótima de um problema de fluxo máximo em um grafo bipartido ou a partir da solução ótima de um problema de b-emparelhamento máximo bipartido (subgrafo máximo restrito em grau de um multigrafo bipartido) são apresentados no capítulo 2.

Neste capítulo, são propostos três algoritmos iterativos diferentes para a determinação da solução ótima de  $P$  (generalização do problema de seleção), em que a cada iteração, um problema de seleção é resolvido.

O primeiro algoritmo é teoricamente o mais simples.

O segundo algoritmo, modifica o problema, transformando-o em um problema de seleção generalizada em um digrafo bipartido completo com  $\sum [b_i : i=1, 2, \dots, n]=0$ . Este problema de seleção generalizada em um digrafo bipartido é, então resolvido através do primeiro algoritmo.

O terceiro algoritmo, além da transformação do problema em um problema de seleção generalizada em um digrafo bipartido completo com  $\sum [b_i : i=1, 2, \dots, n]=0$ , utiliza o conceito de escalonamento de um dos parâmetros do problema.

Os três algoritmos têm sua complexidade dependente do algoritmo utilizado na resolução dos problemas de seleção em um digrafo (capítulo 2).

Os três algoritmos são algoritmos do tipo primal-dual. No primeiro a solução ótima de  $P$  é aproximada por soluções viáveis do primal, isto é, a viabilidade do primal é mantida em todas as iterações. Nos dois últimos a viabilidade do primal não é mantida em todas as iterações, é mantida a viabilidade do dual; e no terceiro é utilizada a técnica de escalonamento introduzida por Edmonds e Karp [EdK72].

#### 4.2- Validade do procedimento iterativo.

Dados um problema  $P$  de seleção generalizada, e uma solução viável  $x^0$  do problema  $P$ , sejam os problemas  $P(x^0)$  (seleção generalizada) e  $P_0(x^0)$  (seleção), definidos a seguir.

$$\begin{aligned}
 P(x^0): \quad \text{máx } f(x) &= \sum_{i=1}^n b_i \cdot x_i \\
 \text{s.a. } x_i - x_j &\leq c_{ij} - x^0_i + x^0_j && \text{para todo arco } (v_i, v_j) \in E \\
 0 \leq x_i &\leq u_i - x^0_i && \text{para todo vértice } v_i \in V \\
 x_i &\text{ inteiro} && \text{para todo vértice } v_i \in V
 \end{aligned}$$

$$\begin{aligned}
 P_0(x^0): \quad \text{máx } f(y) &= \sum_{i=1}^n b_i \cdot y_i \\
 \text{s.a. } y_i - y_j &\leq 0 && \text{para todo } (v_i, v_j) \in E \text{ com } c_{ij} - x^0_i + x^0_j = 0 \\
 y_i &= 0, 1 && \text{para todo vértice } v_i \in V \text{ com } u_i - x^0_i \geq 1 \\
 y_i &= 0 && \text{para todo vértice } v_i \in V \text{ com } u_i - x^0_i = 0
 \end{aligned}$$

Os problemas  $P(x^0)$  e  $P_0(x^0)$  têm as seguintes características:

- $(0, 0, \dots, 0)^T$  é solução viável de  $P(x^0)$  e de  $P_0(x^0)$ .
- Qualquer solução ótima  $y^0$  de  $P_0(x^0)$  é solução viável de  $P(x^0)$ .
- Qualquer solução ótima  $y^0$  de  $P_0(x^0)$  tem  $f(y^0) \geq 0$ , como consequência imediata da definição de  $P_0(x^0)$  e de  $(0, 0, \dots, 0)^T$  ser solução viável de  $P_0(x^0)$ .
- A função objetivo  $f(x)$  de  $P(x^0)$  e de  $P_0(x^0)$  é linear e as restrições dos dois problemas, também são lineares.
- $P(x^0)$  pode ser obtido a partir de  $P$ , através de mudança da variável  $x$  por  $x - x^0$ , desprezo, na função objetivo, do termo constante e substituição no conjunto de restrições de  $-x^0_i \leq x_i \leq u_i - x^0_i$  por  $0 \leq x_i \leq u_i - x^0_i$ . Esta substituição no conjunto de restrições corresponde a uma diminuição do conjunto de soluções viáveis; isto é,  $\{x + x^0 : x \text{ é viável de } P(x^0)\}$  está contido em  $\{x : x \text{ é viável de } P\}$ .

-Se  $y^0$  é uma solução ótima de  $P_0(x^0)$  e  $\alpha = \min(\{u_i - x^0_i : y^0_i = 1\} \cup \{c_{ij} - x^0_i + x^0_j : y^0_i - y^0_j = 1\})$ , então  $\alpha y^0$  é solução viável de  $P(x^0)$ ,  $x^0 + \alpha y^0$  é solução viável de  $P$  e  $f(x^0 + \alpha y^0) = f(x^0) + \alpha f(y^0) \geq f(x^0)$ .

Estas características de  $P(x^0)$  e  $P_0(x^0)$ , estabelecem uma maneira de obter, a partir de uma solução viável  $x^0$  de  $P$ , uma outra solução viável  $x^1$  de  $P$ , tal que  $f(x^1) \geq f(x^0)$ . Este processo pode ser utilizado de maneira iterativa para determinar uma seqüência  $x^0, x^1, x^2, \dots, x^k$  de soluções viáveis de  $P$  satisfazendo  $f(x^i) \leq f(x^{i+1})$  para  $i=0, 1, \dots, k-1$ .

Este processo iterativo, quando iniciado com  $(0, 0, \dots, 0)^T$ , é provado pelos Teoremas 4.1 e 4.2, terminar com a determinação de uma solução ótima de  $P$ . O Teorema 4.1 prova a existência de uma solução ótima de  $P$  do tipo  $x^0 + x^1$ , onde  $x^1$  é uma solução viável de  $P(x^0)$ . O Teorema 4.2 prova que  $x^0$  é solução ótima de  $P$ , quando  $(0, 0, \dots, 0)^T$  é solução ótima de  $P_0(x^0)$ , estabelecendo um critério de parada do processo iterativo.

Seja  $P_0$ , o problema  $P_0(x^0)$ , quando  $x^0 = (0, 0, \dots, 0)^T$ ; isto é:

$$P_0: \quad \max f(y) = \sum_{i=1}^n b_i \cdot y_i$$

$$\text{s.a. } y_i - y_j \leq 0 \quad \text{para todo } (v_i, v_j) \in E \text{ com } c_{ij} = 0$$

$$y_i = 0, \quad 1 \quad \text{para todo vértice } v_i \in V \text{ com } u_i \geq 1$$

$$y_i = 0 \quad \text{para todo vértice } v_i \in V \text{ com } u_i = 0$$

**Teorema 4.1:** Se  $y^0$  é uma solução ótima de  $P_0$  então existe uma solução ótima  $x^*$  de  $P$  com  $x^*_i \geq \alpha \cdot y^0_i$  para  $i=1, 2, \dots, n$ , onde  $\alpha = \min(\{u_i : y^0_i = 1\} \cup \{c_{ij} : y^0_i - y^0_j = 1\})$ .

**Demonstração:** Seja  $x^0$  uma solução ótima de  $P$ . Os vetores  $x^1$  e  $y^1$ , definidos abaixo, são soluções viáveis de  $P$  e  $P_0$  respectivamente.

$$x^1_i = \begin{cases} \alpha & \text{se } x^0_i < \alpha \cdot y^0_i \\ x^0_i & \text{se } x^0_i \geq \alpha \cdot y^0_i \end{cases} \quad y^1_i = \begin{cases} 0 & \text{se } x^0_i < \alpha \cdot y^0_i \\ y^0_i & \text{se } x^0_i \geq \alpha \cdot y^0_i \end{cases}$$

A verificação das viabilidades de  $x^1$  e  $y^1$  é feita, a seguir, através de exame de todas as alternativas.

-  $x^1$  é solução viável de  $P$ , isto é:

$$x^1_i \geq 0 \quad ,$$

$$x^1_i = \begin{cases} \alpha \leq u_i & \text{se } x^0_i < \alpha \cdot y^0_i \text{ e} \\ x^0_i \leq u_i & \text{se } x^0_i \geq \alpha \cdot y^0_i \text{ ,} \end{cases}$$

$x^1_i$  é inteiro e

$$x^1_i - x^1_j = \begin{cases} \alpha - \alpha = 0 \leq c_{ij} & \text{se } x^0_i < \alpha \cdot y^0_i \text{ e } x^0_j < \alpha \cdot y^0_j \\ \alpha - x^0_j \leq c_{ij} - x^0_j \leq c_{ij} & \text{se } x^0_i < \alpha \cdot y^0_i \text{ e } x^0_j \geq \alpha \cdot y^0_j \\ x^0_i - x^0_j \leq c_{ij} & \text{se } x^0_i \geq \alpha \cdot y^0_i \text{ e } x^0_j \geq \alpha \cdot y^0_j \\ x^0_i - \alpha \leq x^0_i - x^0_j \leq c_{ij} & \text{se } x^0_i \geq \alpha \cdot y^0_i \text{ e } x^0_j < \alpha \cdot y^0_j \end{cases}$$

para todo  $i = 1, 2, \dots, n$  e todo arco  $(v_i, v_j) \in E$ .

-  $y^1$  é solução viável de  $P_0$ , isto é:

$$y^1_i = 0, 1 \quad \text{e}$$

$$y^1_i - y^1_j = \begin{cases} 0 - 0 \leq 0 & \text{se } x^0_i < \alpha \cdot y^0_i \text{ e } x^0_j < \alpha \cdot y^0_j \\ y^0_i - y^0_j \leq c_{ij} = 0 & \text{se } x^0_i \geq \alpha \cdot y^0_i \text{ e } x^0_j \geq \alpha \cdot y^0_j \\ 0 - y^0_j \leq 0 & \text{se } x^0_i < \alpha \cdot y^0_i \text{ e } x^0_j \geq \alpha \cdot y^0_j \\ y^0_i - 0 \leq x^0_i / \alpha - y^0_j - 1 \leq (x^0_i - x^0_j) / \alpha \leq c_{ij} / \alpha = 0 & \text{se } x^0_i \geq \alpha \cdot y^0_i \\ & \text{e } x^0_j < \alpha \cdot y^0_j \end{cases}$$

para todo  $i = 1, 2, \dots, n$  e todo arco  $(v_i, v_j) \in E$  com  $c_{ij} = 0$ .

De  $y^0$  ser solução ótima de  $P_0$  e  $y^1$  ser solução viável de  $P_0$ , tem-se:

$$f(y^0) - f(y^1) \geq 0, \quad \text{logo}$$

$$\sum [b_i \cdot y^0_i : x^0_i < \alpha \cdot y^0_i] = \sum [b_i : x^0_i < \alpha \cdot y^0_i] \geq 0 \quad (1)$$

De  $x^0$  ser solução ótima de  $P$  e  $x^1$  ser solução viável de  $P$ , tem-se:

$$f(x^0) - f(x^1) \geq 0, \quad \text{logo}$$

$$\sum [b_i \cdot x^0_i : x^0_i < \alpha \cdot y^0_i] - \sum [b_i \cdot \alpha : x^0_i < \alpha \cdot y^0_i] \geq 0, \quad \text{mas}$$

$$\sum [b_i \cdot (-1) : x^0_i < \alpha \cdot y^0_i] \geq \sum [b_i \cdot (x^0_i - \alpha) : x^0_i < \alpha \cdot y^0_i], \quad \text{logo}$$

$$-\sum [b_i : x^0_i < \alpha \cdot y^0_i] \geq 0 \quad (2)$$

De (1) e (2) tem-se :  $\sum [b_i : x^0_i < \alpha \cdot y^0_i] = 0$  e  $f(x^0) = f(x^1)$ , logo  $x^1$  é solução ótima de  $P$  e satisfaz  $x^1_i \geq \alpha \cdot y^0_i$  para  $i=1, 2, \dots, n$ . A verificação da validade destas desigualdades é feita através do exame das duas alternativas :

$$x^1_i = \begin{cases} \alpha = \alpha \cdot y^0_i & \text{se } x^0_i < \alpha \cdot y^0_i \\ x^0_i \geq \alpha \cdot y^0_i & \text{se } x^0_i \geq \alpha \cdot y^0_i \end{cases},$$

portanto,  $x^1$  é o  $x^*$  do enunciado do teorema. ■

**Teorema 4.2 :** Se  $y^0 = (0, 0, \dots, 0)^T$  é solução ótima de  $P_0$  então  $x^* = (0, 0, \dots, 0)^T$  é solução ótima de  $P$ .

**Demonstração :** Seja, por absurdo,  $x^0$  uma solução ótima de  $P$  com  $f(x^0) > f(x^*) = 0$ .

Os vetores  $x^1$  e  $y^1$ , definidos abaixo, são soluções viáveis de  $P$  e  $P_0$  respectivamente.

$$y^1_i = \begin{cases} 1 & \text{se } x^0_i \geq 1 \\ 0 & \text{se } x^0_i = 0 \end{cases} \quad x^1 = x^0 - y^1$$

A verificação das viabilidades de  $x^1$  e  $y^1$  é feita, a seguir, através de exame de todas as alternativas :

-  $y^1$  é solução viável de  $P_0$ , isto é:

$$y^1_i = 0, 1 \quad e$$

$$y^1_i - y^1_j = \begin{cases} 0 - 0 \leq 0 & \text{se } x^0_i = x^0_j = 0 \\ 1 - 1 \leq 0 & \text{se } x^0_i \geq 1 \text{ e } x^0_j \geq 1 \\ 0 - 1 \leq 0 & \text{se } x^0_i = 0 \text{ e } x^0_j \geq 1 \\ 1 - 0 \leq x^0_i - x^0_j \leq c_{ij} = 0 & \text{se } x^0_i \geq 1 \text{ e } x^0_j = 0 \end{cases}$$

para todo  $i = 1, 2, \dots, n$  e todo arco  $(v_i, v_j) \in E$  com  $c_{ij} = 0$ .

-  $x^1$  é solução viável de  $P$ , isto é:

$$x^1_i \geq 0 \quad e$$

$$x^1_i = x^0_i - y^0_i \leq x^0_i \leq u_i,$$

$x^1_i$  é inteiro e

$$x^1_i - x^1_j = x^0_i - x^0_j - y^1_i - y^1_j = \begin{cases} 0 - 0 - 0 - 0 \leq c_{ij} & \text{se } x^0_i = x^0_j = 0 \\ 0 - x^0_j - 0 - 1 \leq 0 \leq c_{ij} & \text{se } x^0_i = 0 \text{ e } x^0_j \geq 1 \\ x^0_i - x^0_j - 1 - 1 \leq c_{ij} & \text{se } x^0_i \geq 1 \text{ e } x^0_j \geq 1 \\ x^0_i - 0 - 1 - 0 = x^0_i - x^0_j - 1 \leq c_{ij} & \text{se } x^0_i \geq 1 \text{ e } x^0_j = 0 \end{cases}$$

para todo  $i = 1, 2, \dots, n$  e todo arco  $(v_i, v_j) \in E$ .

De  $y^0$  ser solução ótima de  $P_0$  e  $y^1$  ser solução viável de  $P_0$ , tem-se:

$$f(y^1) \leq f(y^0) = 0 \quad (3)$$

De  $x^0$  ser solução ótima de  $P$  e  $x^1$  ser solução viável de  $P$ , tem-se:

$$f(x^0) \geq f(x^1) = f(x^0) - f(y^1), \quad \text{logo } f(y^1) \geq 0 \quad (4)$$

De (3) e (4) tem-se:  $f(y^1) = 0$  e  $f(x^1) = f(x^0)$ ,  
então  $x^1$  é solução ótima de  $P$  e  $x^0_i > x^1_i \geq 0$  ou  $x^0_i = x^1_i = 0$  para  
 $i = 1, 2, \dots, n$ .

A construção acima pode ser repetida para a solução ótima  $x^1$ , em lugar de  $x^0$ , até obter um vetor binário  $x^1$ , o que ocorre após um número finito de repetições. A função objetivo permanece com o mesmo valor positivo, para os vetores  $x^1$  obtidos. Logo, o vetor binário obtido  $x^1$  é solução viável de  $P$  e de  $P_0$  e  $f(x^1) > 0$ , então  $y^0$  não é solução ótima de  $P_0$ , absurdo. ■

#### 4.3- Algoritmo I de resolução do problema de seleção generalizada.

Os Teoremas 4.1 e 4.2 garantem que é possível determinar uma solução ótima de  $P$ , através de transformações lineares em  $P$  e como consequência em  $P_0$ , e resolução de uma seqüência de problemas de seleção.

##### 4.3.1- Algoritmo I de resolução do problema de seleção generalizada em um digrafo $G = (V, E)$ .

$$I) \quad x^0 := (0, 0, \dots, 0)^T.$$

II) Repetir passos 1, 2, 3, 4 até o passo 1 mandar parar.

1) Determinar  $P_0$  e sua solução ótima  $y^0$ .

Se  $y^0 = (0, 0, \dots, 0)^T$  então parar,  $x^0$  é solução ótima de  $P$ .

2)  $\alpha := \min(\{u_i : y^0_i = 1\} \cup \{c_{ij} : (v_i, v_j) \in E \text{ e } y^0_i - y^0_j = 1\})$

3) Determinar nova solução viável de  $P$  :  $x^0 := x^0 + \alpha \cdot y^0$ .

4) Atualizar  $P$  :

$$u_i := u_i - \alpha \cdot y^0_i \quad i=1, 2, \dots, n$$

$$c_{ij} := c_{ij} - \alpha \cdot (y^0_i - y^0_j) \quad (v_i, v_j) \in E$$

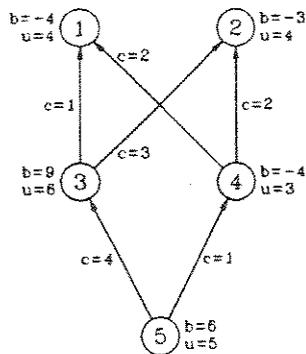
Este algoritmo consiste, em determinar, em cada iteração, uma melhora do valor da função objetivo. A solução ótima  $y^0$  de  $P_0$ , corresponde a uma direção viável de subida da função. (Deslocamentos na direção do gradiente ou de uma sua projeção, no conjunto de soluções viáveis, não garantem alcançar a fronteira em um ponto com componentes inteiras). O valor de  $\alpha$  corresponde ao máximo, que pode ser alterada a solução viável  $x^0$ , na direção  $y^0$ , de maneira a manter a viabilidade.

O processo envolve no máximo a resolução de  $\sum [u_i : v_i \in V^+]$  problemas de seleção, visto que para todo vértice  $v_i$ , os valores de  $x^0_i$  ao longo do processo iterativo, formam uma seqüência não decrescente de inteiros não negativos, que quando  $y^0 = (0, 0, \dots, 0)^T$  é solução ótima de  $P_0$ , o processo iterativo termina e que quando  $y^0_i = 1$  para  $v_i \in V^-$  existe  $v_j \in V^+$  com  $y^0_j = 1$ .

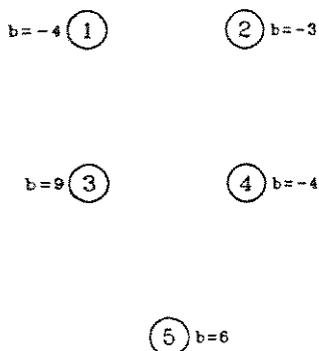
#### 4.3.2- Exemplo de resolução do problema de seleção generalizada pelo Algoritmo I:

$$\begin{array}{ll}
 P: & \text{máx } f(x) = -4 \cdot x_1 - 3 \cdot x_2 + 9 \cdot x_3 - 4 \cdot x_4 + 6 \cdot x_5 \\
 & \text{s.a } x_3 - x_1 \leq 1 \\
 & \quad x_4 - x_1 \leq 2 \\
 & \quad x_3 - x_2 \leq 3 \\
 & \quad x_4 - x_2 \leq 2 \\
 & \quad x_5 - x_3 \leq 4 \\
 & \quad x_5 - x_4 \leq 1 \\
 & \quad 0 \leq x_1 \leq 4 \\
 & \quad 0 \leq x_2 \leq 4 \\
 & \quad 0 \leq x_3 \leq 6 \\
 & \quad 0 \leq x_4 \leq 3 \\
 & \quad 0 \leq x_5 \leq 5 \\
 & \quad x_i \text{ inteiro} \qquad \qquad i = 1, 2, \dots, 5
 \end{array}$$

Este problema pode ser representado no digrafo abaixo.

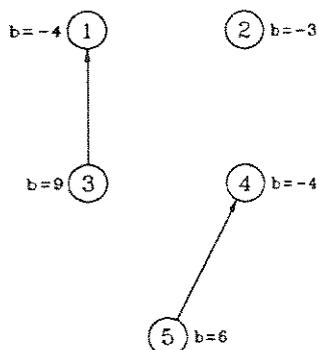


1º passo :



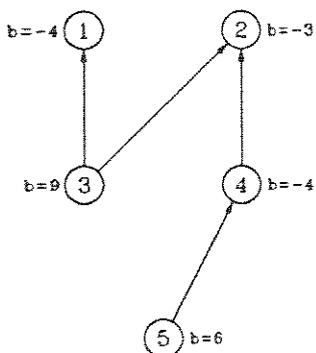
solução ótima  $y^0 = (0, 0, 1, 0, 1)$   
 $\alpha = \min(\{5, 6\}U\{1, 3, 1\}) = 1$   
 $x^0 = (0, 0, 1, 0, 1)$

2º passo :

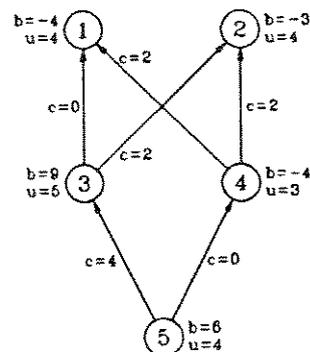


solução ótima  $y^0 = (1, 0, 1, 1, 1)$   
 $\alpha = \min(\{4, 5, 3, 4\}U\{2, 2\}) = 2$   
 $x^0 = (2, 0, 3, 2, 3)$

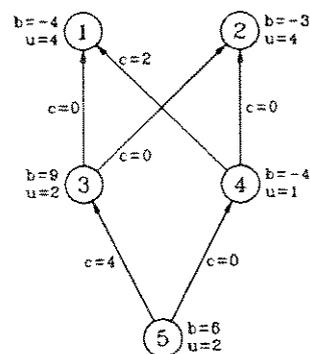
3º passo :



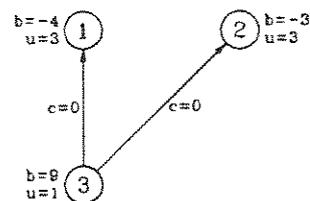
solução ótima  $y^0 = (1, 1, 1, 1, 1)$   
 $\alpha = \min(\{4, 4, 2, 1, 2\}) = 1$   
 $x^0 = (3, 1, 4, 3, 4)$



G após a atualização.

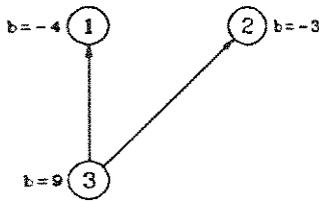


G após a atualização.



G após a atualização.  
 (vértice  $v_4$  sai e com este  $v_5$ )

4º passo :



solução ótima  $y^0 = (1, 1, 1, 0, 0)$

$$\alpha = \min(\{3, 3, 1\}) = 1$$

$$x^0 = (4, 2, 5, 3, 4)$$

5º passo :



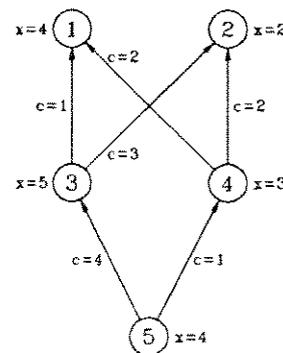
solução ótima  $y^0 = (0, 0, 0, 0, 0)$

$x^0 = (4, 2, 5, 3, 4)$  é ótima de P.



G após a atualização.

(vértice  $v_3$  sai)



G com indicação de uma seleção generalizada ótima.

**Observação:** O método acima pode não convergir se como solução inicial, ao invés da solução viável  $x^0 = (0, 0, \dots, 0)^T$ , for utilizada uma solução viável qualquer.

Se no exemplo acima, for utilizada como solução inicial a solução viável  $x^0 = (4, 3, 5, 3, 4)^T$ , o digrafo fica restrito ao vértice  $v_2$ , e do fato de  $b_2 = -3$  tem-se que  $y^0 = (0, 0, 0, 0, 0)^T$ , mas  $x^0 = (4, 3, 5, 3, 4)^T$  não é ótima de P.

### 4.3.3- Complexidade do Algoritmo I.

O número de iterações é limitado, como consequência imediata de  $y^0_i = 1$  para  $v_i \in V^-$  implicar na existência de  $v_j \in V^+$  com  $y^0_j = 1$ , do Teorema 4.2 e das restrições  $0 \leq x_i \leq u_i$  e  $x_i$  inteiro para  $i = 1, 2, \dots, n$ , a um número máximo de  $\sum [u_i : v_i \in V^+]$ .

Em cada iteração é calculada a solução ótima de um problema de seleção. A complexidade do Algoritmo I depende, portanto, da complexidade do algoritmo utilizado para a resolução do problema de seleção.

A complexidade do Algoritmo I é, então, limitada por :

$$U_1 \cdot (\text{Complexidade do algoritmo de seleção utilizado}),$$

onde  $U_1 = \sum [ u_i : v_i \in V^* ]$ .

Quando o algoritmo utilizado na solução do problema de seleção é o Algoritmo II-A (Algoritmo II do problema de seleção, utilizando o Algoritmo A para a resolução do problema de b-emparelhamento máximo bipartido, apresentados no capítulo 2), é possível estabelecer outro limitante para a complexidade do Algoritmo I do problema de seleção generalizada. A seguir, são feitas as considerações necessárias para o estabelecimento deste outro limitante.

O grafo utilizado para o problema de b-emparelhamento máximo em uma iteração é modificado para a iteração seguinte : seja pela introdução de vértices, seja pela introdução ou retirada de arestas.

Arestas  $e_k = (v_i, v_j)$ , que pertencem ao b-emparelhamento máximo em uma iteração ( $r_k > 0$ ), têm  $y_i^0 = y_j^0$  na solução ótima de  $P_0$  nesta iteração e, portanto  $c_{ij}$  não é alterado; isto é, a aresta  $e_k$  continua no grafo na próxima iteração e o b-emparelhamento  $r_k$  pode ser utilizado como valor inicial de b-emparelhamento do grafo da próxima iteração.

A retirada de um vértice  $v_i$  ( $u_i = 0$ ) ao fim de uma iteração implica na retirada, também, de todos os arcos incidentes a este vértice, neste caso o valor do b-emparelhamento máximo calculado decresce de no máximo  $|b_i|$ , e no pior caso este b-emparelhamento terá que ser refeito. Isto corresponde a calcular no total, no máximo um b-emparelhamento de valor  $2 \cdot b$ , supondo que cada referente calculado possui apenas um caminho aumentante e que cada caminho aumentante permite uma ampliação do b-emparelhamento de apenas 1, tem-se no máximo  $2 \cdot b$  referentes e  $2 \cdot b$  caminhos aumentantes calculados. Na construção de um referente, para um grafo com  $m_1$  arestas, no máximo  $2 \cdot m_1$  arestas são examinadas e na determinação de um caminho aumentante no referente  $m_1$  arestas são examinadas. No cálculo de todos os b-emparelhamentos necessários, no máximo  $6 \cdot b \cdot m_1$  exames de arestas são executados.

O b-emparelhamento máximo obtido em uma iteração é utilizado como b-emparelhamento inicial na próxima iteração, quando ao fim da primeira não há retirada de vértice. Para o cálculo do número máximo de vezes que o b-emparelhamento das arestas é alterado, ao longo de iterações consecutivas, entre duas iterações com retirada de vértices, o resultado do Teorema 2.6 é necessário:

“Se  $S$  é um  $b$ -emparelhamento e  $R$  é um  $b$ -emparelhamento máximo de um grafo bipartido  $G$  então existe em  $G$ , um caminho aumentante em relação a  $S$  com comprimento menor ou igual a  $2 \cdot \lfloor \|R\| / (\|R\| - \|S\|) \rfloor - 1$ , onde  $\|R\| = \sum [r_i : i=1, 2, \dots, m]$  e  $\|S\| = \sum [s_i : i=1, 2, \dots, m]$ .”

**Teorema 4.3 :** O número de alterações no  $b$ -emparelhamento das arestas de  $G_1$ , é no máximo  $O(b^2)$  entre duas iterações em que há retirada de vértice e durante a execução do algoritmo proposto, é no máximo  $O(b^2 \cdot n)$ .

**Demonstração :** Para calcular a solução ótima da generalização do problema de seleção é necessário calcular o  $b$ -emparelhamento máximo de  $G_1$ . Este  $b$ -emparelhamento é calculado através do cálculo de uma seqüência de  $b$ -emparelhamentos máximos  $R_1, R_2, \dots, R_k$  de subgrafos de  $G_1$ . Enquanto não há retirada de vértices de  $G_1$ , tem-se  $\|R_1\| \leq \|R_2\| \leq \dots \leq \|R_k\|$ . Para determinar o  $b$ -emparelhamento  $R_i$  a partir do  $b$ -emparelhamento  $R_{i-1}$ , obtido na iteração anterior, supondo caminhos aumentantes que aumentem de apenas 1 o  $b$ -emparelhamento, tem-se que a soma dos comprimentos de todos os caminhos aumentantes necessários é menor ou igual a:

$$\sum [ 2 \cdot \lfloor \|R_i\| / (\|R_i\| - j) \rfloor - 1 : j = \|R_{i-1}\|, \|R_{i-1}+1\|, \dots, \|R_i-1\| ]$$

Para o cálculo da seqüência dos  $b$ -emparelhamentos máximos acima, nas mesmas hipóteses, a soma dos comprimentos de todos os caminhos aumentantes necessários é menor ou igual a :

$$\begin{aligned} & \sum_{i=1}^k [ 2 \cdot \lfloor \|R_i\| / (\|R_i\| - j) \rfloor - 1 : j = \|R_{i-1}\|, \|R_{i-1}+1\|, \dots, \|R_i-1\| ] = \\ & = \sum_{i=1}^k [ 2 \cdot \lfloor (\|R_i\| - j + j) / (\|R_i\| - j) \rfloor - 1 : j = \|R_{i-1}\|, \|R_{i-1}+1\|, \dots, \|R_i-1\| ] \leq \\ & \leq \sum_{i=1}^k [ 2 \cdot \lfloor j / (\|R_i\| - j) \rfloor + 1 : j = \|R_{i-1}\|, \|R_{i-1}+1\|, \dots, \|R_i-1\| ] \leq \\ & \leq \sum_{i=1}^k [ 2 \cdot j + 1 : j = \|R_{i-1}\|, \|R_{i-1}+1\|, \dots, \|R_i-1\| ] \leq \\ & \leq \sum_{k=1}^b ( 2 \cdot k + 1 ) = b^2 + 2 \cdot b \quad , \end{aligned}$$

portanto, o número de alterações entre duas retiradas de vértices é  $O(b^2)$ . Em  $G_1$  existem  $n$  vértices, no máximo  $n-1$  operações de retirada de vértices podem ser efetuadas, logo no total tem-se um máximo de  $O(b^2 \cdot n)$  alterações de  $b$ -emparelhamento de arestas.

■

O fecho transitivo de  $G$  é calculado em  $O(n \cdot m)$  operações. O algoritmo calcula em cada iteração o fecho transitivo de um subgrafo de  $G$ . O fecho transitivo de um subgrafo de  $G$  de uma iteração ao fim da qual nenhum arco ou vértice é retirado do subgrafo, pode ser utilizado para o cálculo do fecho transitivo do novo subgrafo; e os dois fechos transitivos são calculados em um total de  $O(n \cdot m)$  operações. O número de iterações em que pelo menos um vértice é retirado do digrafo é limitado por  $n-1$ . O número de iterações em que pelo menos um arco é retirado do subgrafo, sem a retirada de nenhum de seus vértices do digrafo, é limitada pelo valor do  $b$ -emparelhamento máximo e este último é limitado por  $b$ . O número de operações necessárias para o cálculo do fecho transitivo de todos os subgrafos é limitado superiormente por  $(b + n) \cdot n \cdot m$ . Se  $G$  é bipartido com bipartição  $(V^+, V^-)$  não é necessário calcular o fecho transitivo de seus subgrafos.

Quando o algoritmo utilizado na solução dos problemas de seleção é o Algoritmo II-A do capítulo 2, a complexidade do Algoritmo I do problema de seleção generalizada é limitada por :

$$O(b \cdot m_1 + n \cdot b^2 + b \cdot n \cdot m_2)$$

$$\text{Onde se } G \text{ é qualquer:} \quad m_2 = |E| = m \quad \text{e} \quad m_1 \leq |V^+| \cdot |V^-|$$

$$\text{se } G \text{ é bipartido:} \quad m_2 = 0 \quad \text{e} \quad m_1 \leq m$$

A complexidade do algoritmo apresentado tem três limitantes superiores, dependendo da forma que o problema de seleção é resolvido :

- Quando o Algoritmo I (fluxo máximo) do capítulo 2 é utilizado para determinar as soluções ótimas dos problemas de seleção, a complexidade é :

$$O(U_1 \cdot (|V^+|^2 \cdot |V^-|))$$

- Quando o Algoritmo II-A do capítulo 2, é utilizado para determinar as soluções ótimas dos problemas de seleção, a complexidade é :

$$O(\min(U_1 \cdot (\min(|V^+|, b^{1/2}) \cdot |V^+| \cdot |V^-| + b \cdot \log_2 b + n \cdot m_2), (b \cdot m_1 + n \cdot b^2 + b \cdot n \cdot m_2)))$$

- Quando o Algoritmo II-B do capítulo 2, (que utiliza o Algoritmo B, com escalonamento, para a resolução do problema de b-emparelhamento bipartido) é utilizado para determinar a solução ótima dos problemas de seleção, a complexidade é :

$$O(U_1 \cdot (|V^+| \cdot m_2 + (\log_2 B_1)^2 \cdot n^{\frac{1}{2}} \cdot |V^+| \cdot |V^-| + n^2 \cdot \log_2 B_1)).$$

A complexidade do Algoritmo I para o problema de seleção generalizada depende do algoritmo utilizado na solução dos problemas de seleção. Na tabela, a seguir, um resumo destas complexidades.

Complexidade do Algoritmo I, de acordo com o algoritmo utilizado na resolução dos problemas de seleção:

$$\text{Algoritmo I : } O(U_1 \cdot |V^+|^2 \cdot |V^-|)$$

$$\text{Algoritmo II-A : } O(\min(U_1 \cdot A_1, (b \cdot m_1 + n \cdot b^2 + b \cdot n \cdot m_2)))$$

$$\text{Algoritmo II-B : } O(U_1 \cdot (|V^+| \cdot m_2 + A_3))$$

$$\text{Onde se } G \text{ é qualquer: } m_2 = |E| = m \text{ e } m_1 \leq |V^+| \cdot |V^-|$$

$$\text{se } G \text{ é bipartido: } m_2 = 0 \text{ e } m_1 \leq m.$$

$$b = \min(\sum[b_i : v_i \in V^+], \sum[b_i : v_i \in V^-]).$$

$$B_1 = \max(|b_i| : v_i \in V).$$

$$U_1 = \sum[u_i : v_i \in V^+].$$

$$|V^+| \leq |V^-|.$$

$$A_1 = (\min(|V^+|, b^{\frac{1}{2}}) \cdot |V^+| \cdot |V^-| + b \cdot \log_2 b + n \cdot m_2).$$

$$A_3 = (\log_2 B_1)^2 \cdot n^{\frac{1}{2}} \cdot |V^+| \cdot |V^-| + (\log_2 B_1) \cdot n^2.$$

#### 4.4- Transformação em um problema de seleção generalizada bipartido.

Seja  $P_1$ , o problema obtido a partir de  $P$  através da mudança de variáveis  $x_i = z_i - z_0$  para  $i = 1, 2, \dots, n$ , onde  $z_0$  é inteiro não negativo.

$$\begin{aligned}
 P_1: \quad & \text{máx } f_1(z) = \sum_{i=1}^n b_i \cdot z_i - \left( \sum_{i=1}^n b_i \right) \cdot z_0 \\
 & \text{s.a. } z_i - z_j \leq c_{ij} \quad \text{para todo arco } (v_i, v_j) \in E \\
 & \quad z_i - z_0 \leq u_i \quad \text{para todo vértice } v_i \in V \\
 & \quad z_0 - z_i \leq 0 \quad \text{para todo vértice } v_i \in V \\
 & \quad z_i \text{ inteiro} \quad \text{para todo } v_i \in V \text{ ou para } i=0 \\
 & \quad z_0 \geq 0
 \end{aligned}$$

Qualquer solução viável de  $P_1$  satisfaz à restrição  $z_i \geq 0$  para  $i = 1, 2, \dots, n$  de  $P$ , visto que satisfaz às restrições  $z_0 - z_i \leq 0$  e  $z_0 \geq 0$ .

O problema  $P_1$  é equivalente ao problema  $P_2$  de seleção generalizada, definido em  $G_2 = (V_B, E_2)$  :

$$\begin{aligned}
 P_2: \quad & \text{máx } f_1(z) = \sum_{i=0}^n b_i \cdot z_i \\
 & \text{s.a. } z_i - z_j \leq h_{ij} \quad \text{para todo arco } (v_i, v_j) \in E_2 \\
 & \quad z_i \geq 0 \quad \text{para todo vértice } v_i \in V_B \\
 & \quad z_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V_B
 \end{aligned}$$

onde  $V_B = V \cup \{v_0\}$ ,  $E_2 = E \cup \{(v_i, v_0) : v_i \in V\} \cup \{(v_0, v_i) : v_i \in V\}$ , a cada vértice  $v_i$  é associado um valor  $b_i$  (com  $b_0 = -\sum [b_i : i=1, 2, \dots, n]$ ) e a cada arco  $(v_i, v_j)$  é associado um inteiro não negativo  $h_{ij}$  com

$$h_{ij} = \begin{cases} c_{ij} & \text{para todo } (v_i, v_j) \in E \\ u_i & \text{para } v_j = v_0 \\ 0 & \text{para } v_i = v_0 \end{cases}$$

O digrafo  $G_2$ , por construção, tem as seguintes características:

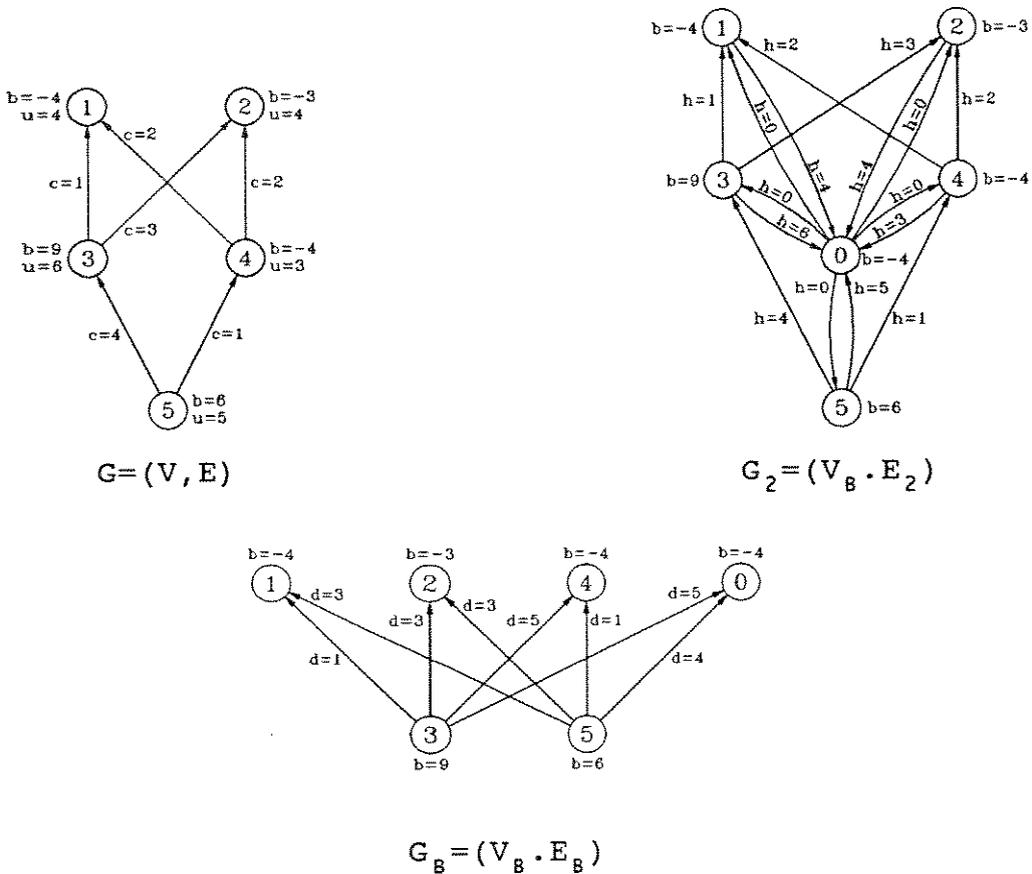
- Entre dois vértices  $v_i$  e  $v_j$  de  $V_B$  existe sempre um caminho em  $G_2$ .
- Sendo  $h_{ij}$  considerado como o comprimento do arco  $(v_i, v_j)$ , para todo  $(v_i, v_j) \in E_2$ , o comprimento do menor caminho entre dois vértices  $v_r \in V^+$  e  $v_s \in V^-$  quaisquer é menor ou igual a  $u_r$ , para  $r = 1, 2, \dots, n$  e igual a zero para  $r = 0$ .
- Todo vértice  $v_j \in V^-$  tem um vértice de  $V^+$  que o antecede em  $G_2$  e todo vértice  $v_i \in V^+$  tem um vértice de  $V^-$  que o sucede em  $G_2$ .

Seja  $P_B$  o problema de seleção generalizada definido em  $G_B = (V_B, E_B)$ :

$$\begin{array}{ll}
 P_B: & \text{máx } f_1(z) = \sum_{i=1}^n b_i \cdot z_i \\
 & \text{s.a. } z_i - z_j \leq d_{ij} \quad \text{para todo arco } (v_i, v_j) \in E_B \\
 & \quad z_i \geq 0 \quad \text{para todo vértice } v_i \in V_B \\
 & \quad z_i \text{ inteiro} \quad \text{para todo vértice } v_i \in V_B
 \end{array}$$

onde  $V_B = (V^+, V^-)$ ,  $V^+ = \{v_i : v_i \in V_B \text{ e } b_i \geq 0\}$ ,  $V^- = \{v_i : v_i \in V_B \text{ e } b_i < 0\}$ ,  $E_B = \{(v_i, v_j) : v_i \in V^+, v_j \in V^-\}$  e a cada arco  $(v_i, v_j) \in E_B$  é associado um número inteiro não negativo  $d_{ij}$  igual ao comprimento do menor caminho de  $v_i$  para  $v_j$  em  $G_2$ .

Exemplo :



Como conseqüência das definições de  $G_B$  e de  $G_2$ , tem-se que :

- $G_B$  é um grafo bipartido e completo com  $|V^+| \cdot |V^-|$  arcos.
- Entre dois vértices quaisquer  $v_i \in V^+$  e  $v_j \in V^-$  sempre existe em  $G_2$  um caminho, que contém  $v_0$ , com comprimento  $u_i$ , portanto  $d_{ij} \leq u_i$ .

Na secção 4.5, é proposta a resolução de  $P$  através da resolução de  $P_2$  e deste através da resolução de  $P_B$ . O problema  $P_B$  é um caso particular de  $P_2$  e este um caso particular de  $P$ . As soluções ótimas de  $P$  e  $P_2$  são relacionadas através dos Teoremas 4.4 e 4.5. As soluções ótimas de  $P_2$  e de  $P_B$  são relacionadas através do Teorema 4.6. Os Lemas 4.1 e 4.2 são utilizados na demonstração do Teorema 4.6.

**Teorema 4.4 :** Se  $z^* = (z^*_0, z^*_1, \dots, z^*_n)^T$  é solução ótima de  $P_2$  então  $x^* = (z^*_1 - z^*_0, z^*_2 - z^*_0, \dots, z^*_n - z^*_0)^T$  é solução ótima de  $P$ .

Conseqüência imediata da construção de  $P_2$ .

■

**Teorema 4.5 :** Se  $x^* = (x^*_1, x^*_2, \dots, x^*_n)^T$  é solução ótima de  $P$  então  $z^* = (0, x^*_1, x^*_2, \dots, x^*_n)^T$  é solução ótima de  $P_2$ .

**Demonstração :**  $z^*$  é solução viável de  $P_2$ , isto é:

$$x^*_i - x^*_j \leq c_{ij} \quad \text{logo} \quad x^*_i - x^*_j \leq h_{ij} \quad \text{se } i \neq 0 \text{ e } j \neq 0$$

$$x^*_i \leq u_i \quad \text{logo} \quad x^*_i - 0 \leq u_i \quad \text{ou} \quad x^*_i - x^*_0 \leq h_{i0} \quad \text{se } j=0 \text{ e } x^*_0=0$$

$$x^*_j \geq 0 \quad \text{logo} \quad 0 - x^*_j \leq 0 \quad \text{ou} \quad x^*_0 - x^*_j \leq h_{0j} \quad \text{se } i=0 \text{ e } x^*_0=0$$

Seja  $z^0 = (x^0_0, x^0_1, \dots, x^0_n)^T$  uma solução ótima de  $P_2$ , logo  $x^0 = (x^0_1 - x^0_0, x^0_2 - x^0_0, \dots, x^0_n - x^0_0)^T$  é também solução ótima de  $P$ , isto é:  $f(x^*) = f(x^0)$ , portanto:

$$\sum_{i=1}^n b_i \cdot x^*_i = \sum_{i=1}^n b_i \cdot x^0_i - x^0_0 \cdot \sum_{i=1}^n b_i - x^0_0 \cdot b_0 + x^0_0 \cdot b_0 = \sum_{i=0}^n b_i \cdot x^0_i$$

logo  $z^*$  é solução ótima de  $P_2$

■

**Lema 4.1 :** Se  $z^* = (z^*_0, z^*_1, \dots, z^*_n)^T$  é solução ótima de  $P_B$  e  $v_j$  é um vértice qualquer de  $V^-$  então existe pelo menos um vértice  $v_i$  pertencente a  $V^+$ , tal que  $z^*_i - z^*_j = d_{ij}$ .

**Demonstração :** Seja  $v_k$  um vértice qualquer de  $V^+$ ; de  $z^*$  ser solução ótima de  $P_B$  tem-se:  $z^*_k - z^*_j \leq d_{kj}$ .

Se, por absurdo,  $z^*_k - z^*_j < d_{kj}$  para todo  $v_k \in V^+$ , seja  $z^0 = (z^0_0, z^0_1, \dots, z^0_n)^T$ , definido por:

$$z^0_i = \begin{cases} z^*_i & \text{para } v_i = v_j \\ z^*_i + 1 & \text{para todo } v_i \in V - \{v_j\} \end{cases}$$

logo  $z^0$  é solução viável de  $P_B$  e  $f_1(z^0) = f_1(z^*) + \sum [b_i : v_i \in V - \{v_j\}]$ , portanto  $f_1(z^0) > f_1(z^*)$  e  $z^*$  não é solução ótima de  $P_B$ . ■

**Lema 4.2:** Se  $z^* = (z^*_0, z^*_1, \dots, z^*_n)^T$  é solução ótima de  $P_B$  e  $v_i$  é um vértice qualquer de  $V^+$ , então existe pelo menos um vértice  $v_j$  pertencente a  $V^-$ , tal que  $z^*_i - z^*_j = d_{ij}$ .

**Demonstração:** Seja  $v_k$  um vértice qualquer de  $V^-$ ; de  $z^*$  ser solução ótima de  $P_B$  tem-se:  $z^*_i - z^*_k \leq d_{ik}$ .

Se, por absurdo,  $z^*_i - z^*_k < d_{ik}$  para todo  $v_k \in V^-$ , seja  $z^0 = (z^0_0, z^0_1, \dots, z^0_n)^T$ , definido por:

$$z^0_r = \begin{cases} z^*_r & \text{para todo } v_r \in V - \{v_i\} \\ z^*_r + 1 & \text{para } v_r = v_i \end{cases}$$

logo  $z^0$  é solução viável de  $P_B$  e da linearidade de  $f_1(z)$  tem-se:

$f_1(z^0) = f_1(z^*) + b_i$ , portanto  $f_1(z^0) > f_1(z^*)$  e  $z^*$  não é solução ótima de  $P_B$ . ■

**Teorema 4.6:**  $z^* = (z^*_0, z^*_1, \dots, z^*_n)^T$  é solução ótima de  $P_2$  se e somente se  $z^*$  é solução ótima de  $P_B$ .

**Demonstração:**  $P_2$  e  $P_B$  possuem a mesma função objetivo; desta forma, é suficiente provar que:

I:- se  $z^*$  é solução viável de  $P_2$ , então  $z^*$  é solução viável de  $P_B$  e

II:- se  $z^*$  é solução ótima de  $P_B$ , então  $z^*$  é solução viável de  $P_2$ .

I:- Se  $z^*$  é solução viável de  $P_2$  e  $(v_i, v_j)$  é um arco qualquer de  $E_B$ , seja  $v_i = v_{i1}, v_{i2}, \dots, v_{ik} = v_j$  um caminho qualquer de  $v_i \in V^+$  para  $v_j \in V^-$  em  $G_2$ , então  $z^*$  satisfaz as desigualdades:

$$z^*_{i1} - z^*_{i2} \leq h_{i1i2}, \quad z^*_{i2} - z^*_{i3} \leq h_{i2i3}, \dots, z^*_{ik-1} - z^*_{ik} \leq h_{ik-1ik},$$

$$\text{logo } z^*_{i1} - z^*_{ik} \leq \sum_{l=1}^{k-1} h_{ilil+1} \quad \text{ou} \quad z^*_i - z^*_j \leq \sum_{l=1}^{k-1} h_{ilil+1}$$

mas  $\sum_{l=1}^{k-1} h_{ilil+1}$  é o comprimento de um caminho qualquer de  $v_i$  para  $v_j$  em  $G_2$ , e  $d_{ij}$  é o comprimento de um destes caminhos, logo  $z^*_i - z^*_j \leq d_{ij}$ , isto é:  $z^*$  é viável de  $P_B$ .

II:- Se  $z^*$  é solução ótima de  $P_B$  e  $(v_i, v_j)$  é um arco qualquer de  $E_2$ , tem-se quatro possibilidades :

$$1) v_i \in V^+ \text{ e } v_j \in V^-,$$

Neste caso,  $(v_i, v_j) \in E_B$ ,  $z^*_i - z^*_j \leq d_{ij}$  e  $d_{ij} \leq h_{ij}$  logo,  $z^*_i - z^*_j \leq h_{ij}$ .

$$2) v_i \in V^+ \text{ e } v_j \in V^+,$$

Existe em  $G_2$ , pelo menos um vértice de  $V^-$  que sucede  $v_j$  e pelo Lema 4.2, existe em  $G_2$ , pelo menos um vértice  $v_k \in V^-$ , tal que  $z^*_j - z^*_k = d_{jk}$ , mas  $d_{ik} \leq h_{ij} + d_{jk}$  e  $z^*_i - z^*_k \leq d_{ik}$  logo,  $z^*_i - z^*_j \leq h_{ij}$ .

$$3) v_i \in V^- \text{ e } v_j \in V^-,$$

Existe em  $G_2$ , pelo menos um vértice de  $V^+$  que antecede  $v_i$  e pelo Lema 4.1, existe em  $G_2$ , pelo menos um vértice  $v_k \in V^+$ , tal que  $z^*_k - z^*_i = d_{ki}$ , mas  $d_{kj} \leq h_{ij} + d_{ki}$  e  $z^*_k - z^*_j \leq d_{kj}$  logo,  $z^*_i - z^*_j \leq h_{ij}$ .

$$4) v_i \in V^- \text{ e } v_j \in V^+,$$

Existe em  $G_2$ , pelo menos um vértice de  $V^+$  que antecede  $v_i$  e pelo menos um vértice de  $V^-$  que sucede  $v_j$ . Pelos Lema 4.1 e Lema 4.2, existem em  $G_2$ , pelo menos dois vértices  $v_k \in V^+$  e  $v_l \in V^-$ , tais que  $z^*_k - z^*_i = d_{ki}$  e  $z^*_j - z^*_l = d_{jl}$ , mas  $d_{kl} \leq h_{ij} + d_{ki} + d_{jl}$  e  $z^*_k - z^*_l \leq d_{kl}$  logo,  $z^*_i - z^*_j \leq h_{ij}$ .

logo, para qualquer arco  $(v_i, v_j) \in E_2$ ,  $z^*$  satisfaz  $z_i - z_j \leq h_{ij}$ , de  $z^*$  ser um vetor de inteiros não negativos, tem-se que  $z^*$  é solução viável de  $P_2$ .



#### 4.5- Algoritmo II de resolução do problema de seleção generalizada.

A transformação de um problema de seleção generalizada em um digrafo qualquer em um problema de seleção generalizada bipartido, apresentada na secção 4.4, e a validade do Algoritmo I, apresentado em 4.5, garantem a validade do Algoritmo II, apresentado a seguir.

##### 4.5.1- Algoritmo II de resolução do problema de seleção generalizada.

I) Determinar  $P_B$ .

II)  $z^0 := (0, 0, \dots, 0)^T$ .

III) Repetir passos 1, 2, 3, 4 até o passo 1 mandar ir para IV.

1) Determinar  $P_0$ , relativo a  $P_B$ , e sua solução ótima  $y^0$ . Se  $y^0 = (0, 0, \dots, 0)^T$  então ir para IV,  $z^0$  é solução ótima de  $P_B$ .

2)  $\alpha := \min(\{d_{ij} : (v_i, v_j) \in E_B \text{ e } y_i^0 - y_j^0 = 1\})$ .

3)  $z^0 := z^0 + \alpha \cdot z^0$ .

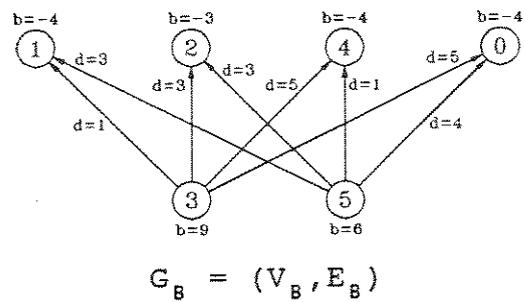
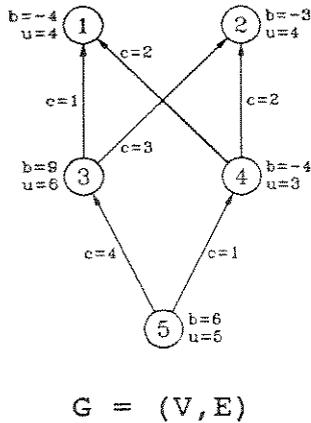
4) Atualizar  $P_B$ :  $d_{ij} := d_{ij} - \alpha \cdot (y_i^0 - y_j^0) \quad (v_i, v_j) \in E_B$

IV) Determinar  $x^*$  solução ótima de  $P$ :  $x_i^* := z_i^0 - z_0^0 \quad i=1, 2, \dots, n$ .

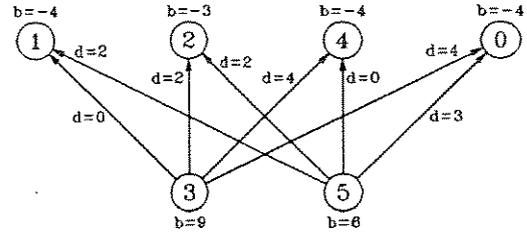
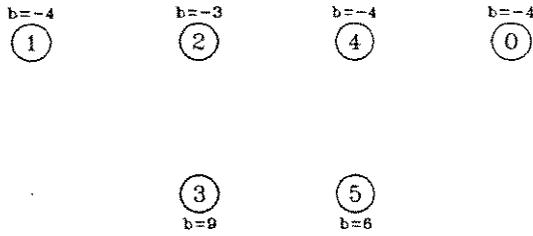
## 4.5.2- Exemplo de resolução do problema de seleção generalizada pelo Algoritmo II.

$$\begin{aligned}
 P: \quad & \text{máx } f(x) = -4 \cdot x_1 - 3 \cdot x_2 + 9 \cdot x_3 - 4 \cdot x_4 + 6 \cdot x_5 \\
 \text{s. a} \quad & x_3 - x_1 \leq 1 \\
 & x_4 - x_1 \leq 2 \\
 & x_3 - x_2 \leq 3 \\
 & x_4 - x_2 \leq 2 \\
 & x_5 - x_3 \leq 4 \\
 & x_5 - x_4 \leq 1 \\
 & 0 \leq x_1 \leq 4 \\
 & 0 \leq x_2 \leq 4 \\
 & 0 \leq x_3 \leq 6 \\
 & 0 \leq x_4 \leq 3 \\
 & 0 \leq x_5 \leq 5 \\
 & x_i \text{ inteiro} \qquad i = 1, 2, \dots, 5
 \end{aligned}$$

Este problema pode ser representado no digrafo  $G$  e, como demonstrado, uma solução ótima pode ser determinada a partir de uma solução ótima do problema representado pelo digrafo  $G_B$ , indicados abaixo.



1º passo:  $z^0 = (0, 0, 0, 0, 0, 0)$



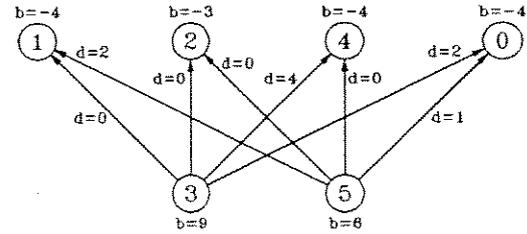
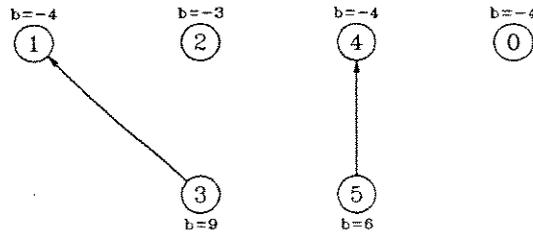
$G_B$  após a atualização.

solução ótima  $y^0 = (0, 0, 0, 1, 0, 1)$

$\alpha = \min(\{1, 1\}) = 1$

$z^0 = (0, 0, 0, 1, 0, 1)$

2º passo:  $z^0 = (0, 0, 0, 1, 0, 1)$



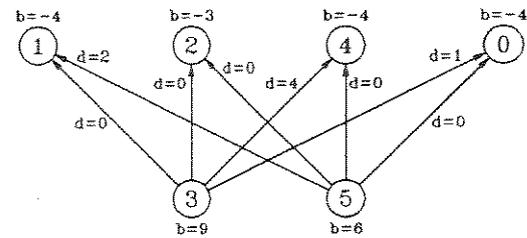
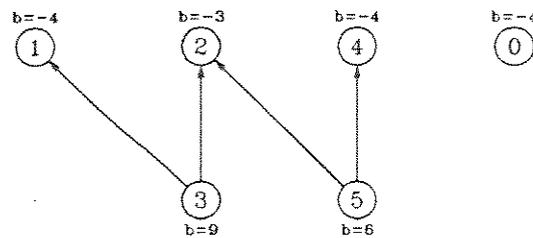
$G_B$  após a atualização.

solução ótima  $y^0 = (0, 1, 0, 1, 1, 1)$

$\alpha = \min(\{2, 2, 3, 4\}) = 2$

$z^0 = (0, 2, 0, 3, 2, 3)$

3º passo:  $z^0 = (0, 2, 0, 3, 2, 3)$



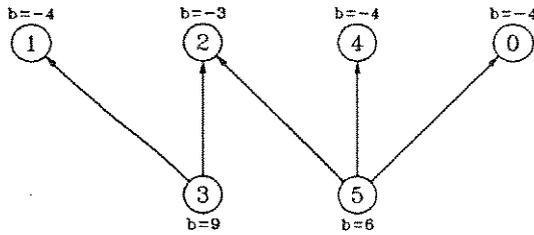
$G_B$  após a atualização.

solução ótima  $y^0 = (0, 1, 1, 1, 1, 1)$

$\alpha = \min(\{2, 1\}) = 1$

$z^0 = (0, 3, 1, 4, 3, 4)$

4º passo :  $z^0 = (0, 3, 1, 4, 3, 4)$

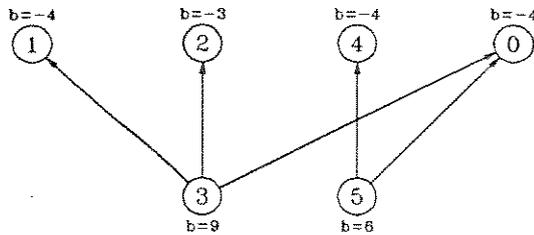


solução ótima  $y^0 = (0, 1, 1, 1, 0, 0)$

$$\alpha = \min(\{4, 1\}) = 1$$

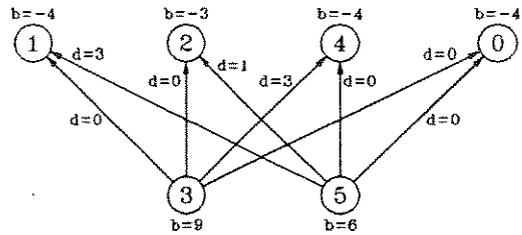
$$z^0 = (0, 4, 2, 5, 3, 4)$$

5º passo :  $z^0 = (0, 4, 2, 5, 3, 4)$

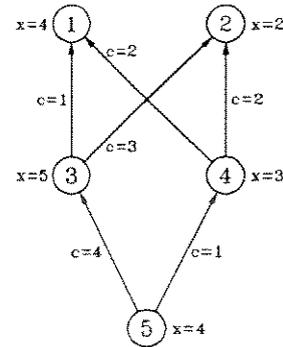


solução ótima  $y^0 = (0, 0, 0, 0, 0, 0)$

$x^0 = (4, 2, 5, 3, 4)$  é ótima de P.



$G_B$  após a atualização.



G com indicação de uma seleção generalizada ótima.

### 4.5.3- Complexidade do Algoritmo II.

A determinação de  $P_B$  é feita através do cálculo dos caminhos mínimos de todos os vértices de  $V^+$  para todos os vértices de  $V^-$  em  $G_2$ . Os melhores algoritmos, para o cálculo de todos estes caminhos mínimos, são, atualmente, o de Fredmann e Tarjan [FmT87], com complexidade  $O(|V^+| \cdot (m+n \cdot \log_2 n))$ , e o de Ahuja, Mehlhorn, Orlin e Tarjan [AMO90] com complexidade  $O(|V^+| \cdot (m+n \cdot (\log_2 C)^{1/2}))$ , onde  $C = \max\{c_{ij} : (v_i, v_j) \in E\}$ .

**Teorema 4.7 :** O número de iterações (problemas de seleção resolvidos) pelo Algoritmo II é limitado por  $U + 1$ , onde  $U = \max\{d_{ij} : (v_i, v_j) \in E_B\} \leq \max\{u_i : v_i \in V^+\}$ .

**Demonstração :**  $z^0_i$  para  $i = 1, 2, \dots, n$  não tem limitante superior; e como conseqüência, em nenhuma iteração existe retirada de vértices do digrafo  $G_B$  ou retirada de arcos, que correspondem às arestas com b-emparelhamento diferente de zero. Portanto, existe

pelo menos um vértice  $v_k \in V^+$  que termina todas as iterações a menos da última não saturado, ou seja  $y^0_k = 1$  em todas as iterações exceto a última.

Todos os arcos do digrafo  $G_B$  têm  $d_{ij} \leq U$ . No final de uma iteração ao atualizar  $P_B$  o valor de  $d_{ij}$ , de todos os arcos de vértices  $v_i \in V^+$  não saturados para vértices  $v_j \in V^-$  não saturados, diminui de pelo menos uma unidade.

Portanto, após no máximo  $U$  iterações todos os arcos de vértices  $v_i \in V^+$  não saturados para vértices  $v_j \in V^-$  não saturados têm  $d_{ij} = 0$ , logo na próxima iteração, o b-emparelhamento é perfeito e  $y^0 = (0, 0, \dots, 0)^T$ . ■

A determinação de  $P_0$  para cada  $P_B$  é imediata. No cálculo da solução ótima de  $P_0$  (problema de seleção) não é necessário calcular o fecho transitivo do subgrafo de  $G_B$  ( $G_B$  é bipartido).

Em cada iteração é calculada a solução ótima de um problema de seleção bipartido. A complexidade do Algoritmo II depende, portanto, da complexidade do algoritmo utilizado para a resolução do problema de seleção.

A complexidade do Algoritmo II é, então, limitada por :

$$O(|V^+| \cdot (m+n \cdot \min(\log_2 n, (\log_2 C)^{1/2}))) + U \cdot (\text{Complexidade do algoritmo de seleção bipartida})$$

Quando o algoritmo utilizado na solução do problema de seleção é o Algoritmo II-A, do capítulo 2, as seguintes considerações são válidas para o estabelecimento da complexidade do Algoritmo II do problema de seleção generalizada.

Não é realizada retirada de vértices no final de nenhuma iteração, uma vez que as variáveis associadas aos vértices de  $P_B$  não têm limitante superior.

Os arcos retirados no final de uma iteração não pertencem ao b-emparelhamento. O b-emparelhamento máximo calculado em um subgrafo de  $G_B$ , em uma iteração, é utilizado como b-emparelhamento inicial do subgrafo de  $G_B$  da iteração seguinte. Tem-se, portanto o cálculo do b-emparelhamento máximo de  $G_B$ , que é um b-emparelhamento perfeito de valor  $B = \max(\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-])$ .

Na determinação do b-emparelhamento máximo, pelo Teorema 4.3 tem-se, no total :  $O(B^2)$  alterações no b-emparelhamento das arestas. Supondo que cada referente calculado possui apenas um caminho aumentante e que cada caminho aumentante permite uma ampliação do b-emparelhamento de apenas 1, tem-se no máximo  $B$  referentes e  $B$  caminhos aumentantes

calculados. Na construção de um referente, no máximo  $2 \cdot |V^+| \cdot |V^-|$  arcos são examinados e na determinação de um caminho aumentante no referente  $|V^+| \cdot |V^-|$  arcos são examinados. No cálculo de todos os b-emparelhamentos necessários, no máximo  $3 \cdot B \cdot |V^+| \cdot |V^-|$  exames de arestas são executados; portanto,  $O(B \cdot |V^+| \cdot |V^-|)$  exames de arestas na construção dos referentes e dos caminhos aumentantes.

Complexidade do Algoritmo II utilizando o Algoritmo II-A é :

$$O(|V^+| \cdot (m+n \cdot \min(\log_2 n, (\log_2 C)^{\frac{1}{2}})) + B \cdot |V^-| + B^2)$$

$$\text{onde } B = \max(\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-]) \quad e$$

$$C = \max\{h_{ij} : (v_i, v_j) \in E_2\} = \max\{(c_{ij} : (v_i, v_j) \in E) \cup \{u_i : v_i \in V\}\}$$

A complexidade do Algoritmo II para o problema de seleção generalizada depende do algoritmo utilizado na solução dos problemas de seleção. Na tabela, a seguir, um resumo destas complexidades.

Complexidade do Algoritmo II, de acordo com o algoritmo utilizado na resolução dos problemas de seleção:

$$\text{Algoritmo I : } O(D + U \cdot |V^+|^2 \cdot |V^-|)$$

$$\text{Algoritmo II-A : } O(D + \min(B \cdot |V^+| \cdot |V^-| + B^2, U \cdot A_2))$$

$$\text{Algoritmo II-B : } O(D + U \cdot A_3)$$

$$\text{Onde } B = \max\{\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-]\}$$

$$B_1 = \max\{|b_i| : v_i \in V\}$$

$$C = \max\{h_{ij} : (v_i, v_j) \in E_2\} = \max\{(c_{ij} : (v_i, v_j) \in E) \cup \{u_i : v_i \in V\}\}$$

$$|V^+| \leq |V^-|$$

$$A_2 = \min(|V^+|, (B)^{\frac{1}{2}}) \cdot |V^+| \cdot |V^-| + B \cdot \log_2 B$$

$$A_3 = (n^{\frac{1}{2}} \cdot |V^+| \cdot |V^-| \cdot \log_2 B_1 + n^2) \cdot \log_2 B_1$$

$$U = \max\{d_{ij} : (v_i, v_j) \in E_B\} \leq \max\{u_i : v_i \in V^+\}$$

$$D = |V^+| \cdot (n \cdot \min(\log_2 n, (\log_2 C)^{\frac{1}{2}}) + m)$$

#### 4.6- Técnica de escalonamento de um dos parâmetros do problema.

A base do algoritmo a ser apresentado são a transformação do problema de seleção generalizada  $P$  em um problema de seleção generalizada bipartido, apresentado na secção 4.4, e as propriedades de problemas de programação linear. ( $G_B$  é um grafo bipartido, logo a matriz dos coeficientes de  $P_B$  é totalmente unimodular, então qualquer solução básica de  $P_B$  sem a restrição  $z_i$  inteiro é inteira e satisfaz  $P_B$ .)

É um algoritmo iterativo em que um problema de seleção é resolvido em cada iteração. O problema de seleção a ser resolvido em cada iteração é determinado a partir do escalonamento do valor de  $U = \max \{d_{ij} : (v_i, v_j) \in E_B\}$ .

##### 4.6.1- Introdução.

Métodos iterativos para solução de problemas de redes, utilizando escalonamento de um dos parâmetros do problema, foram inicialmente apresentados por Edmonds e Karp [EdK72]. Recentemente, têm sido apresentados alguns trabalhos utilizando este tipo de procedimento para, principalmente, o problema de fluxo de custo mínimo.

A validade dos algoritmos, com escalonamento de um dos parâmetros do problema, é fortemente baseada no Teorema de ortogonalidade de soluções ótimas de problemas duais e na linearidade da função objetivo e do conjunto de restrições.

A condição de folgas complementares, para a generalização do problema de seleção definido em  $G_B$ , como consequência do Teorema de ortogonalidade de soluções ótimas de problemas duais, é:

“ Um b-emparelhamento perfeito  $r^*$  de  $G_B$  é solução ótima do problema de b-emparelhamento de custo máximo de  $G_B$  se e somente se existe uma solução viável  $z^0$  de  $P_B$ , tal que  $z^0_i - z^0_j < d_{ij}$  implica em  $r^*_k = 0$ , para todo  $e_k = (v_i, v_j) \in E_B$ ; e  $z^0$  é, neste caso, uma solução ótima de  $P_B$ .”

Uma relaxação da condição de folgas complementares, dita condição de  $\beta$ -otimalidade, para o problema de seleção generalizada em  $G_B$ , é dada a seguir.

**Definição :** Dado  $\beta \geq 0$ , um b-emparelhamento perfeito  $r^*$  de  $G_B$  é dito  $\beta$ -ótimo, se e somente se existe uma solução viável  $z^0$  de  $P_B$ , tal que, se  $d_{ij} - z^0_i + z^0_j \geq \beta$  então  $r^*_k = 0$ , para todo  $e_k = (v_i, v_j) \in E_B$ .

Dados uma solução viável  $z^0$  de  $P_B$  e um número inteiro positivo  $\beta$ , sejam  $G_B(z^0, \beta)$  o subgrafo de  $G_B$  restrito aos arcos  $(v_i, v_j) \in E_B$  com  $d_{ij} - z^0_i + z^0_j < \beta$  e  $P_B(z^0, \beta)$  o problema de seleção definido neste subgrafo de  $G_B$ ; isto é:

$$P_B(z^0, \beta): \quad \text{máx} \quad f_1(y) = \sum_{i=1}^n b_i \cdot y_i$$

$$\text{s.a.} \quad y_i - y_j \leq 0 \quad (v_i, v_j) \in E_B \quad \text{com} \quad d_{ij} - z^0_i + z^0_j < \beta$$

$$y_i = 0, 1 \quad \text{para todo vértice } v_i \in V_B$$

#### 4.6.2- Algoritmo III para resolução do problema $P$ de seleção generalizada.

I) Determinar  $P_B$ ;

II)  $U := \text{máx}\{d_{ij} : (v_i, v_j) \in E_B\}$ ;  $\beta := 2 \cdot \lfloor \log_2 U \rfloor$ ;  $z^0 := (0, 0, \dots, 0)^T$ ;

III) Enquanto  $\beta \geq 1$  fazer:

1) Determinar  $P_B(z^0, \beta)$  e sua solução ótima  $y^*$ ;

2)  $z^0 := z^0 + \beta \cdot y^*$ ;

3) Atualizar  $P_B$ :  $d_{ij} := d_{ij} - \beta \cdot (y^*_i - y^*_j)$   $(v_i, v_j) \in E_B$ ;

4) Se  $y^* = (0, 0, \dots, 0)^T$  então  $\beta := \beta / 2$ ;

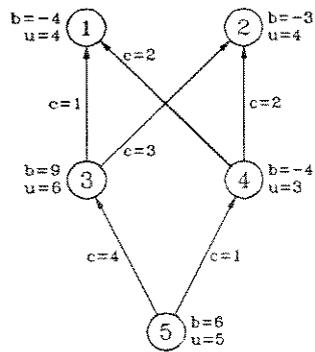
IV) Determinar  $x^*$  solução ótima de  $P$ :

$$x^*_i := z^0_i - z^0_0 \quad i = 1, 2, \dots, n. \quad (z^0 \text{ é solução ótima de } P_B.)$$

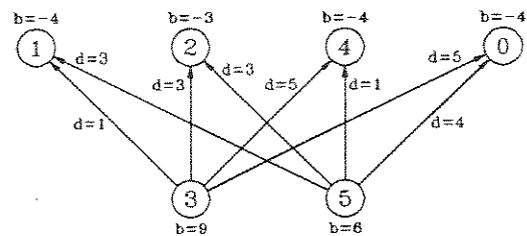
## 4.6.3- Exemplo de resolução do problema de seleção generalizada pelo Algoritmo III.

$$\begin{aligned}
 P: \quad & \max f(x) = -4 \cdot x_1 - 3 \cdot x_2 + 9 \cdot x_3 - 4 \cdot x_4 + 6 \cdot x_5 \\
 \text{s.a} \quad & x_3 - x_1 \leq 1 \\
 & x_4 - x_1 \leq 2 \\
 & x_3 - x_2 \leq 3 \\
 & x_4 - x_2 \leq 2 \\
 & x_5 - x_3 \leq 4 \\
 & x_5 - x_4 \leq 1 \\
 & 0 \leq x_1 \leq 4 \\
 & 0 \leq x_2 \leq 4 \\
 & 0 \leq x_3 \leq 6 \\
 & 0 \leq x_4 \leq 3 \\
 & 0 \leq x_5 \leq 5 \\
 & x_i \text{ inteiro} \qquad i = 1, 2, \dots, 5
 \end{aligned}$$

Este problema pode ser representado no digrafo  $G$  e, como demonstrado, uma solução ótima pode ser determinada a partir de uma solução ótima do problema representado pelo digrafo  $G_B$ , indicados abaixo.



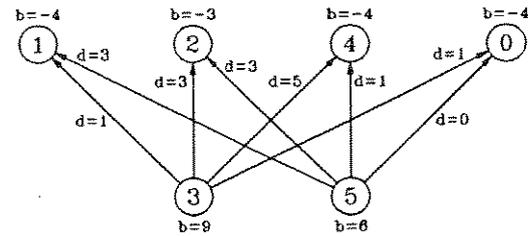
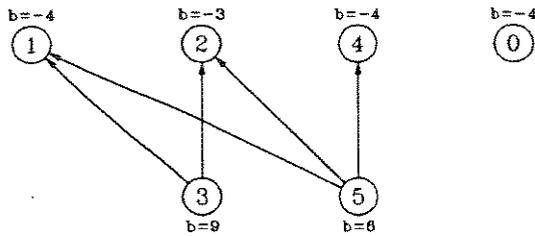
$$G = (V, E)$$



$$G_B = (V_B, E_B)$$

$$U = 5 \quad \beta = 4 \quad z^0 = (0, 0, 0, 0, 0, 0)$$

1º passo:  $\beta = 4$        $z^0 = (0, 0, 0, 0, 0, 0)$

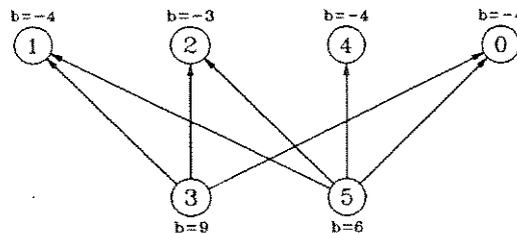


$G_B$  após a atualização.

solução ótima  $y^* = (0, 1, 1, 1, 1, 1)$

$z^0 = (0, 4, 4, 4, 4, 4)$

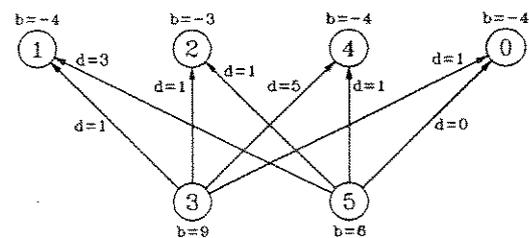
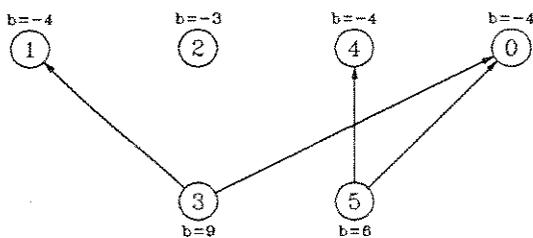
2º passo:  $\beta = 4$        $z^0 = (0, 4, 4, 4, 4, 4)$



solução ótima  $y^* = (0, 0, 0, 0, 0, 0)$

$z^0 = (0, 4, 4, 4, 4, 4)$

3º passo:  $\beta = 2$        $z^0 = (0, 4, 4, 4, 4, 4)$

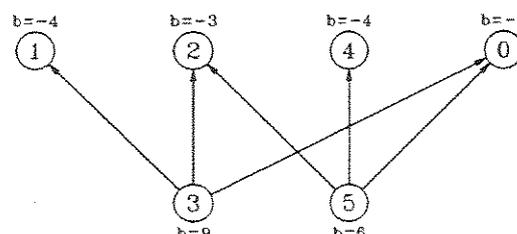


$G_B$  após a atualização.

solução ótima  $y^* = (1, 1, 0, 1, 1, 1)$

$z^0 = (2, 6, 4, 6, 6, 6)$

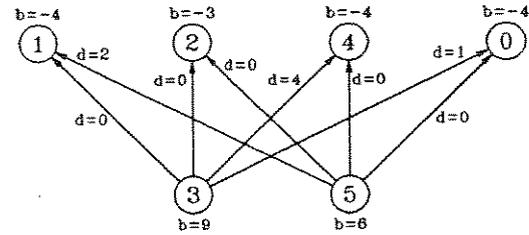
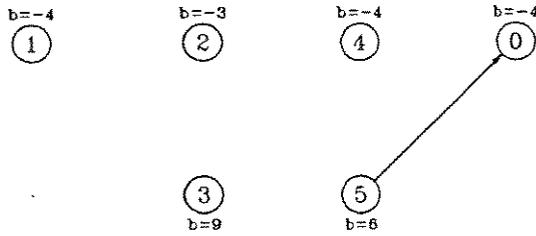
4º passo:  $\beta = 2$        $z^0 = (2, 6, 4, 6, 6, 6)$



solução ótima  $y^0 = (0, 0, 0, 0, 0, 0)$

$z^0 = (2, 6, 4, 6, 6, 6)$

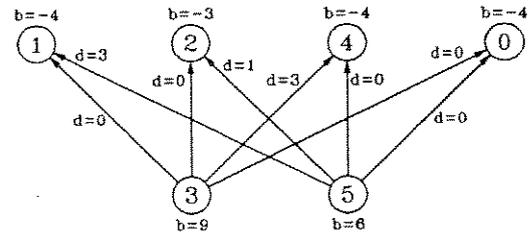
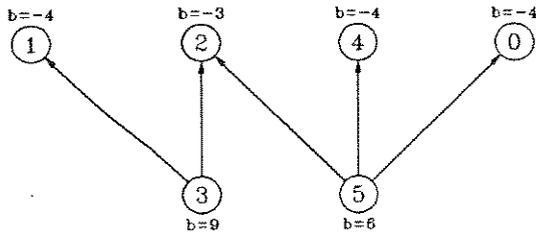
5º passo :  $\beta = 1 \quad z^0 = (2, 6, 4, 6, 6, 6)$



solução ótima  $y^* = (1, 0, 0, 1, 0, 1)$   
 $z^0 = (3, 6, 4, 7, 6, 7)$

$G_B$  após a atualização.

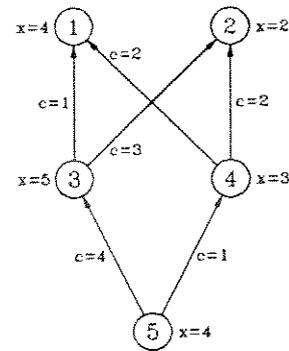
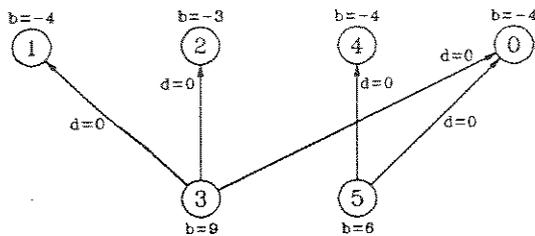
6º passo :  $\beta = 1 \quad z^0 = (3, 6, 4, 7, 6, 7)$



solução ótima  $y^* = (0, 1, 1, 1, 0, 0)$   
 $z^0 = (3, 7, 5, 8, 6, 7)$

$G_B$  após a atualização.

7º passo :  $\beta = 1 \quad z^0 = (3, 7, 5, 8, 6, 7)$



solução ótima  $y^0 = (0, 0, 0, 0, 0, 0)$   
 $z^0 = (3, 7, 5, 8, 6, 7)$  é ótima de  $P_B$ .  
 $x^* = (4, 2, 5, 3, 4)$  é ótima de  $P$ .

$G$  com indicação de uma solução ótima de  $P$ .

4.6.4- Validade do Algoritmo III.

O Algoritmo III (com escalonamento) é um algoritmo iterativo do tipo primal-dual. A validade do Algoritmo III é demonstrada no Teorema 4.8. Os Lemas 4.3 e 4.4, a seguir, são utilizados na demonstração do Teorema 4.8. O Lema 4.4 demonstra que o algoritmo apresentado preserva a  $\beta$ -otimalidade.

**Lema 4.3:** Se  $z^0$  é uma solução viável de  $P_B$  e  $y^*$  é uma solução ótima de  $P_B(z^0, \beta)$ , então  $z^0 + \beta \cdot y^*$  é solução viável de  $P_B$  e  $f_1(z^0) \leq f_1(z^0 + \beta \cdot y^*)$

**Demonstração:** A viabilidade é verificada, a seguir, através de exame das duas alternativas possíveis para todo arco  $(v_i, v_j) \in E_B$ , isto é:  $(v_i, v_j)$  pertence a  $G_B(z^0, \beta)$  e  $(v_i, v_j)$  não pertence a  $G_B(z^0, \beta)$ .

- Para todo  $(v_i, v_j) \in E_B$  e  $(v_i, v_j) \in G_B(z^0, \beta)$ , tem-se  
 $y_i^* - y_j^* \leq 0$  e  $z_i^0 - z_j^0 \leq d_{ij}$ , logo  
 $z_i^0 + \beta \cdot y_i^* - (z_j^0 + \beta \cdot y_j^*) = z_i^0 - z_j^0 + \beta \cdot (y_i^* - y_j^*) \leq z_i^0 - z_j^0 \leq d_{ij}$

- Para todo  $(v_i, v_j) \in E_B$  e  $(v_i, v_j)$  não pertencente a  $G_B(z^0, \beta)$ , tem-se  
 $y_i^* - y_j^* \leq 1$  e  $d_{ij} - z_i^0 + z_j^0 \geq \beta$ , logo  
 $z_i^0 + \beta \cdot y_i^* - (z_j^0 + \beta \cdot y_j^*) = z_i^0 - z_j^0 + \beta \cdot (y_i^* - y_j^*) \leq z_i^0 - z_j^0 + \beta \leq d_{ij}$

De  $z_i^0$  ser um inteiro não negativo,  $y_i^* = 0, 1$  e  $\beta$  ser um inteiro não negativo, tem-se que:  $z_i^0 + \beta \cdot y_i^*$  é um inteiro não negativo.

Portanto,  $z^0 + \beta \cdot y^*$  é solução viável de  $P_B$ .

De  $y^*$  ser solução ótima de  $P_B(z^0, \beta)$  e  $(0, 0, \dots, 0)^T$  ser solução viável de  $P_B(z^0, \beta)$  tem-se:  $f_1(y^*) \geq f_1((0, 0, \dots, 0)^T) = 0$ , logo, como  $f_1$  é linear e  $\beta$  é um número inteiro positivo,

$$f_1(z^0 + \beta \cdot y^*) = f_1(z^0) + \beta \cdot f_1(y^*) \geq f_1(z^0) .$$

■

**Lema 4.4:** Se  $r^*$  é uma extensão de um b-emparelhamento ótimo de  $G_B(z^0, \beta)$ , com  $r_k^* = 0$  para todo arco não pertencente ao subgrafo, e  $y^*$  é a solução ótima correspondente de  $P_B(z^0, \beta)$ , então  $r^*$  é  $\beta$ -ótimo com relação à  $z^0 + \beta \cdot y^*$  (solução viável de  $P_B$ ).

**Demonstração:** Por construção  $r^*$  é um b-emparelhamento perfeito de  $G_B$ . Os arcos não pertencentes ao subgrafo satisfazem trivialmente a condição de  $\beta$ -ótimo.

Os arcos do subgrafo  $G_B(z^0, \beta)$  têm  $d_{ij} - z_i^0 + z_j^0 < \beta$  e se  $d_{ij} - (z_i^0 + \beta \cdot y_i^*) + (z_j^0 + \beta \cdot y_j^*) \geq \beta$  então  $y_i^* - y_j^* < 0$ , logo pela condição de folgas complementares, para estes arcos o b-emparelhamento é igual à zero.

■

**Teorema 4.8 :**  $x^*$  determinado pelo Algoritmo III é uma solução ótima de  $P$ .

**Demonstração :** Pelo Teorema 4.4 é suficiente provar que  $z^0$  é solução ótima de  $P_2$ . Pelo Teorema 4.6 a condição necessária e suficiente para que  $z^0$  seja solução ótima de  $P_2$ , é que  $z^0$  seja solução ótima de  $P_B$ .

A última iteração do algoritmo é para  $\beta = 1$  e termina quando a solução ótima de  $P_B(z^0, 1)$  é  $y^* = (0, 0, \dots, 0)^T$ . A solução ótima de  $P_B(z^0, 1)$  é  $y^* = (0, 0, \dots, 0)^T$ , quando ao determinar esta solução através do cálculo do b-emparelhamento tem-se um b-emparelhamento perfeito de  $G_B(z^0, 1)$ . Seja  $r^*$ , uma extensão deste b-emparelhamento perfeito de  $G_B(z^0, 1)$ , com  $r_k^* = 0$  para todo arco não pertencente ao subgrafo. Este b-emparelhamento  $r^*$  é perfeito de  $G_B$  e as únicas arestas utilizadas em  $P_B(z^0, 1)$  têm  $d_{ij} - z_i^0 + z_j^0 < 1$ , mas pelo Lema 4.3,  $z^0$  é solução viável; isto é,  $d_{ij} - z_i^0 + z_j^0 \geq 0$  ou seja, as arestas de  $G_B$  com b-emparelhamento diferente de zero têm  $d_{ij} - z_i^0 + z_j^0 = 0$ . Portanto de  $z^0$  ser uma solução viável de  $P_B$ , pela condição das folgas complementares  $z^0$  é solução ótima de  $P_B$ . ■

#### 4.6.5- Complexidade do Algoritmo III.

A determinação de  $P_B$  é, novamente, feita através do cálculo dos caminhos mínimos de todos os vértices de  $V^+$  para todos os vértices de  $V^-$  em  $G_2$ . Os melhores algoritmos, para o cálculo de todos estes caminhos mínimos são, atualmente, o de Fredmann e Tarjan [FmT87], com complexidade  $O(|V^+| \cdot (m + n \cdot \log_2 n))$  e o de Ahuja, Mehlhorn, Orlin e Tarjan [AMO90] com complexidade  $O(|V^+| \cdot (m+n \cdot (\log_2 C)^{1/2}))$ , onde  $C = \max\{h_{ij} : (v_i, v_j) \in E_2\}$ .

O número de operações necessárias para o cálculo de  $P_B$  é, novamente, limitado por:  $O(|V^+| \cdot (m+n \cdot \min(\log_2 n, (\log_2 C)^{1/2})))$ .

**Teorema 4.9 :** O número de iterações (problemas de seleção resolvidos) pelo Algoritmo III é no máximo  $n \cdot (\lfloor \log_2 U \rfloor + 1)$ , onde  $U = \max\{d_{ij} : (v_i, v_j) \in E_B\} \leq \max\{u_i : v_i \in V^+\}$ .

**Demonstração :**  $z_i^0$  para  $i = 1, 2, \dots, n$  não tem limitante superior; e na seqüência de iterações com um mesmo valor de  $\beta$  não é feita retirada de vértices do subgrafo  $G_B(z^0, \beta)$ , ou retirada de arestas com b-emparelhamento diferente de zero. Como conseqüência, o número

de b-emparelhamentos de um vértice, no final das iterações com um mesmo valor de  $\beta$ , formam uma seqüência de inteiros não decrescente.

Todos os arcos  $(v_i, v_j)$  do digrafo  $G_B$  têm  $d_{ij} < 2 \lfloor \log_2 U \rfloor + 1$ . Na primeira iteração participam todos os arcos  $(v_i, v_j)$  de  $G_B$  com  $d_{ij} < \beta = 2 \lfloor \log_2 U \rfloor$  e no final desta iteração ao atualizar  $P_B$  o valor de  $d_{ij}$ , de todos os arcos de vértices  $v_i \in V^+$  não saturados para vértices  $v_j \in V^-$  não saturados, diminui de  $\beta$ ; logo como  $G_B$  é completo os demais arcos, que podem ser necessários para a obtenção de um b-emparelhamento perfeito de  $G_B(z^0, \beta)$  passam a ter  $d_{ij} < \beta$ . Neste caso, estes vértices participam da segunda iteração em  $\beta$ , que também é a última com  $\beta = 2 \lfloor \log_2 U \rfloor$ . Logo, são realizadas no máximo duas iterações com  $\beta = 2 \lfloor \log_2 U \rfloor$ .

Na última iteração com um valor qualquer  $2 \cdot \beta$ , obtém-se  $y^0 = (0, 0, \dots, 0)^T$  e um b-emparelhamento perfeito de  $G_B(z^0, 2 \cdot \beta)$  e os arcos utilizados têm  $d_{ij} < 2 \cdot \beta$ .

Na iteração seguinte, a primeira em  $G_B(z^0, \beta)$ , participam os arcos de  $G_B(z^0, 2 \cdot \beta)$  que têm  $d_{ij} < \beta$  e no final desta iteração ao atualizar  $P_B$  o valor de  $d_{ij}$ , de todos os arcos de vértices  $v_i \in V^+$  com  $y_i^0 = 1$  para vértices  $v_j \in V^-$  com  $y_j^0 = 0$ , diminui de  $\beta$ ; logo os arcos de  $G_B(z^0, \beta)$ , que estão nestas condições passam a ter  $d_{ij} < \beta$  e participam da segunda iteração em  $\beta$ . Qualquer caminho alternante presente em  $G_B(z^0, 2 \cdot \beta)$ , no final da iteração  $2 \cdot \beta$ , tem no máximo  $n-1$  arcos e em cada iteração um novo arco de cada um destes caminhos de vértices não saturados de  $V^+$  para vértices não saturados de  $V^-$  entram em  $G_B(z^0, \beta)$ . Logo, ao fim de no máximo  $n$  (número de vértices) iterações, o b-emparelhamento perfeito de  $G_B(z^0, \beta)$  é obtido.

Logo, são necessárias no máximo  $n$  iterações para cada valor de  $\beta$ , portanto o Algoritmo III realiza ao todo no máximo  $n \cdot (\lfloor \log_2 U \rfloor + 1)$  iterações. ■

O processo iterativo é realizado no máximo  $n \cdot (\lfloor \log_2 U \rfloor + 1)$  vezes. Em cada iteração um problema de seleção em um digrafo bipartido é resolvido.

A complexidade do Algoritmo III é, então, limitada por:

$$O(|V^+| (m+n \cdot \min(\log_2 n, (\log_2 C)^{1/2})) + n \cdot \log U \cdot (\text{complexidade do algoritmo de seleção bipartida}))$$

Não é realizada retirada de vértices no final de nenhuma iteração, uma vez que as variáveis associadas aos vértices de  $P_B$  não têm limite superior. Os arcos retirados no final de uma iteração em que  $y^*$  é diferente de  $(0, 0, \dots, 0)^T$ ; isto é, em que não houve alteração no valor de  $\beta$ , não pertencem ao b-emparelhamento (ou têm fluxo igual a zero).

No cálculo da solução ótima do problema de seleção se não houve mudança de valor de  $\beta$ , então a solução ótima do dual (problema de fluxo ou b-emparelhamento máximo) da iteração anterior é utilizada como solução viável inicial do novo problema de seleção.

Quando o algoritmo utilizado na solução do problema de seleção é o Algoritmo II-A do capítulo 2 (utilizando a resolução do problema de b-emparelhamento bipartido sem escalonamento), os seguintes resultados são válidos para o estabelecimento da complexidade do Algoritmo III do problema de seleção generalizada.

Na determinação dos b-emparelhamentos máximos no conjunto de iterações com um mesmo  $\beta$  tem-se, o cálculo de um b-emparelhamento perfeito de valor  $B$  e, pelo Teorema 4.3 um total de  $O(B^2)$  alterações no b-emparelhamento das arestas. Supondo que cada referente calculado possui apenas um caminho aumentante e que cada caminho aumentante permite uma ampliação do b-emparelhamento de apenas 1, tem-se no máximo  $B$  referentes e  $B$  caminhos aumentantes calculados. Na construção de um referente, no máximo  $2 \cdot |V^+| \cdot |V^-|$  arcos são examinados e na determinação de um caminho aumentante no referente  $|V^+| \cdot |V^-|$  arcos são examinados. No cálculo de todos os b-emparelhamentos necessários, no máximo  $3 \cdot B \cdot |V^+| \cdot |V^-|$  exames de arcos são executados; portanto,  $O(B \cdot |V^+| \cdot |V^-|)$  exames de arestas na construção dos referentes e dos caminhos aumentantes.

O número de operações necessárias para resolver os problemas de b-emparelhamentos máximos no conjunto de iterações com um mesmo  $\beta$  é dado por:  $O(B \cdot (B + |V^+| \cdot |V^-|))$ .

A complexidade do Algoritmo III para o problema de seleção generalizada utilizando o algoritmo II-A, para determinar a solução dos problemas de seleção (b-emparelhamento máximo sem escalonamento) é:

$$O(B \cdot (B + |V^+| \cdot |V^-|) \cdot \log_2 U + |V^+| (m+n \cdot \min(\log_2 n, (\log_2 C)^{\frac{1}{2}})))$$

onde  $B = \max(\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-])$  e

$$C = \max\{h_{ij} : (v_i, v_j) \in E_2\} = \max(\{c_{ij} : (v_i, v_j) \in E\} \cup \{u_i : v_i \in V\})$$

$$U = \max\{d_{ij} : (v_i, v_j) \in E_B\} \leq \max\{u_i : v_i \in V\}$$

A complexidade dos algoritmos para a resolução do problema de seleção generalizada depende do algoritmo utilizado na resolução dos problemas de seleção. Na tabela, a seguir, estão estas complexidades :

Complexidade do Algoritmo III, de acordo com o algoritmo utilizado na resolução dos problemas de seleção:

$$\text{Algoritmo I : } O(D + n \cdot \log_2 U \cdot |V^+|^2 \cdot |V^-|)$$

$$\text{Algoritmo II-A : } O(D + \min(n \cdot A_2, B \cdot (|V^+| \cdot |V^-| + B)) \cdot \log_2 U)$$

$$\text{Algoritmo II-B : } O(D + n \cdot A_3 \cdot \log_2 U)$$

$$\text{Onde } B = \max(\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-])$$

$$b = \min(\sum[b_i : v_i \in V^+], \sum[b_i : v_i \in V^-])$$

$$B_1 = \max(|b_i| : v_i \in V)$$

$$|V^+| \leq |V^-|$$

$$C = \max\{h_{ij} : (v_i, v_j) \in E_2\} = \max\{(c_{ij} : (v_i, v_j) \in E) \cup \{u_i : v_i \in V\}\}$$

$$D = |V^+| \cdot (n \cdot \min(\log_2 n, (\log_2 C)^k) + m)$$

$$A_2 = \min(|V^+|, B^k) \cdot |V^+| \cdot |V^-| + B \cdot \log_2 B$$

$$A_3 = (n^k \cdot |V^+| \cdot |V^-| \cdot \log_2 B_1 + n^2) \cdot \log_2 B_1$$

$$U = \max\{d_{ij} : (v_i, v_j) \in E_B\} \leq \max\{u_i : v_i \in V\}$$

## CAPÍTULO 5.

### COMPARAÇÕES ENTRE OS ALGORITMOS E CONCLUSÕES.

#### 5.1- Introdução.

Neste capítulo são feitas comparações entre os algoritmos apresentados nos capítulos 2 e 4, para resolução dos problemas: b-emparelhamento máximo bipartido, seleção de vértices e seleção de vértices generalizada. Na secção 5.2 é apresentado um estudo comparativo das complexidades destes algoritmos. Na secção 5.3, é apresentado um estudo comparativo dos tempos de execução destes algoritmos, para diversos problemas gerados aleatoriamente. Na secção 5.4 são apresentadas conclusões e algumas sugestões para novos trabalhos.

#### 5.2- Algumas comparações entre as complexidades dos algoritmos.

Nesta secção são feitos estudos comparativos do comportamento assintótico dos algoritmos apresentados neste trabalho, ou seja, as complexidades dos algoritmos em notação  $O$  são comparadas. Na secção 5.2.1 são comparados os algoritmos para a resolução do problema de b-emparelhamento máximo bipartido. Na secção 5.2.2, os algoritmos de resolução do problema de seleção são comparados. Na secção 5.2.3, os algoritmos de resolução do problema de seleção generalizada são comparados

##### 5.2.1- Problema de b-emparelhamento máximo bipartido.

As complexidades dos algoritmos de resolução do problema de b-emparelhamento máximo são encontradas na tabela a seguir. Conforme descrito no capítulo 2, o Algoritmo A é uma generalização do algoritmo de Hopcroft e Karp [HpK-72], para o problema de emparelhamento máximo bipartido, enquanto o Algoritmo B utiliza a técnica de escalonamento de  $b_i$ .

**Complexidade dos algoritmos de b-emparelhamento máximo bipartido.**

Algoritmo A	$O(\min( V^+ , (b)^{\frac{1}{2}}) \cdot m + b \cdot \log_2 b)$
Algoritmo B (escalonamento)	$O((\log_2 B_1) \cdot (\log_2 B_1 \cdot n^{\frac{1}{2}} \cdot m + n^2))$
Algoritmo Gabow e Tarjan [GbT88]	$O(b^{\frac{1}{2}} \cdot m' + b' \cdot \log_2 b')$

$$\begin{aligned} \text{onde } b &= \min\{\sum [b_i : v_i \in V^+], \sum [ |b_i| : v_i \in V^- ]\}, \\ B_1 &= \max\{|b_i| : v_i \in V\}, \\ V^+ &= \{v_i : b_i \geq 0\}, \quad V^- = \{v_i : b_i < 0\} \quad \text{e} \quad |V^+| \leq |V^-|, \\ b' &= \sum_{i=1}^n b_i \quad \text{e} \quad m' = \sum_{i=1}^m \min\{b_j : j=1, 2, \dots, n \text{ e } v_j \in e_i\}. \end{aligned}$$

Uma comparação teórica entre os Algoritmos A e B acima, é apresentada, a seguir. Esta comparação, entretanto devido a forma como são feitos os cálculos das complexidades, só é válida para problemas de grandes dimensões. Para esta comparação é razoável supor que  $O(b) = n^{O(1)}$  e  $O(B) = n^{O(1)}$ , o que é bastante comum na literatura (vide por exemplo [GdT87], [GbT89]).

O Algoritmo A tem maior complexidade que o Algoritmo B, quando  $O(b) > n^{\frac{1}{2}} \cdot m \cdot \log_2 n$  e  $O(b) > n^2$  e o Algoritmo B tem maior complexidade que o Algoritmo A quando  $O(n^{\frac{1}{2}} \cdot m \cdot \log_2 n) > b$  e  $O(n^{\frac{1}{2}} \cdot m \cdot \log_2 n) > b^{\frac{1}{2}} \cdot m$ .

### 5.2.2- Problema de seleção.

As complexidades dos algoritmos de resolução do problema de seleção são encontradas na tabela a seguir. Conforme descrito no capítulo 2, o Algoritmo I utiliza o problema de fluxo máximo bipartido, enquanto os algoritmos II-A e II-B utilizam o b-emparelhamento máximo bipartido, sendo que o Algoritmo II-A através do Algoritmo A e o Algoritmo II-B através do Algoritmo B.

**Complexidade dos algoritmos de seleção.**

Algoritmo I :	$O( V^+ ^2 \cdot  V^- )$
Algoritmo II-A :	$O( V^+  \cdot m + \min( V^+ , (b)^{\frac{1}{2}}) \cdot  V^+  \cdot  V^-  + b \cdot \log_2 b)$
Algoritmo II-B :	$O( V^+  \cdot m + \log_2 B_1 \cdot (n^2 + n^{\frac{1}{2}} \cdot  V^+  \cdot  V^-  \cdot \log_2 B_1))$
Algoritmo de Piccard [Pic76] (com Karzanov, Gabow e Tarjan, e Galil)	$O(\min(n^3, n \cdot m \cdot \log_2(n^2/m), n^{5/3} \cdot m^{2/3}))$

Onde  $V^+ = \{v_i : b_i \geq 0\}$ ,  $V^- = \{v_i : b_i < 0\}$  e  $|V^+| \leq |V^-|$   
 $b = \min\{\sum [b_i : v_i \in V^+], \sum [ |b_i| : v_i \in V^- ]\}$ ,  
 $B_1 = \max\{|b_i| : v_i \in V\}$ .

Uma comparação teórica entre os algoritmos acima, subdividida em dois casos é apresentada, a seguir. Esta comparação, entretanto devido à forma como são feitos os cálculos das complexidades, só é válida para problemas de grandes dimensões. Para esta comparação é razoável supor que  $O(b) = n^{O(1)}$  e  $O(B_1) = n^{O(1)}$  (vide por exemplo [GdT87], [GbT89]).

- Se G é denso, então o Algoritmo I tem menor complexidade, sendo que o Algoritmo II-A tem igual complexidade quando  $O(|V^+|^2 \cdot |V^-| / \log_2 n) > b$  e o Algoritmo de Picard tem também a mesma complexidade, quando  $O(V^+) = n$ .

- Se G é esparso, o Algoritmo I tem menor complexidade quando  $O((n \cdot \log_2 n)^{\frac{1}{2}}) > |V^+|$ , sendo que neste caso o Algoritmo II-A tem complexidade igual à do Algoritmo I se  $O(|V^+|^2 \cdot |V^-| / \log_2 n) > b$ . O Algoritmo de Picard tem menor complexidade quando  $O(|V^+|) > (n \cdot \log_2 n)^{\frac{1}{2}}$ .

### 5.2.3- Problema de seleção generalizada.

A complexidade dos algoritmos para a resolução do problema de seleção generalizada são apresentados na tabela, a seguir. Os algoritmos para o problema de seleção generalizada são iterativos, resolvendo em cada iteração um problema de seleção. A complexidade destes algoritmos para a resolução do problema de seleção generalizada depende do algoritmo utilizado na resolução dos problemas de seleção. No método I (Algoritmos I-I, I-II-A, I-II-B) o processo iterativo é aplicado diretamente sobre o digrafo dado; enquanto que nos métodos II e III (Algoritmos II-I, II-II-A, II-II-B, III-I, III-II-A e III-II-B) o digrafo dado é inicialmente transformado através do cálculo de caminhos mínimos, em um digrafo bipartido. Além desta

transformação, o método III utiliza a técnica de escalonamento de  $d_{ij}$ .

**Complexidade dos Algoritmos I, II e III, de acordo com o algoritmo utilizado na resolução dos problemas de seleção:**

Algoritmo I-I:	$O(U_1 \cdot  V^+ ^2 \cdot  V^- )$
Algoritmo II-I:	$O(D + U \cdot  V^+ ^2 \cdot  V^- )$
Algoritmo III-I:	$O(D + n \cdot \log_2 U \cdot  V^+ ^2 \cdot  V^- )$
Algoritmo I-II-A:	$O(\min(U_1 \cdot A_1, b \cdot m_1 + n \cdot b^2 + b \cdot n \cdot m_2))$
Algoritmo II-II-A:	$O(D + \min(U \cdot A_2, B \cdot  V^+  \cdot  V^-  + B^2))$
Algoritmo III-II-A:	$O(D + \min(n \cdot A_2, B \cdot ( V^+  \cdot  V^-  + B)) \cdot \log_2 U)$
Algoritmo I-II-B:	$O(U_1 \cdot ( V^+  \cdot m_2 + A_3))$
Algoritmo II-II-B:	$O(D + U \cdot A_3)$
Algoritmo III-II-B:	$O(D + n \cdot A_3 \cdot \log_2 U)$

Onde se  $G$  é qualquer:  $m_2 = |E| = m$  e  $m_1 \leq |V^+| \cdot |V^-|$

se  $G$  é bipartido:  $m_2 = 0$  e  $m_1 \leq m$

$U_1 = \sum[u_i : v_i \in V^+]$  e  $|V^+| \leq |V^-|$

$A_1 = \min(|V^+|, b^{1/2}) \cdot |V^+| \cdot |V^-| + b \cdot \log_2 b + n \cdot m_2$

$A_2 = \min(|V^+|, B^{1/2}) \cdot |V^+| \cdot |V^-| + B \cdot \log_2 B$

$A_3 = (n^{1/2} \cdot |V^+| \cdot |V^-| \cdot \log_2 B_1 + n^2) \cdot \log_2 B_1$

$D = |V^+| \cdot (m + n \cdot \min(\log_2 n, (\log_2 C)^{1/2}))$

$b = \min(\sum[b_i : v_i \in V^+], \sum[b_i : v_i \in V^-])$

$B = \max(\sum[b_i : v_i \in V^+], \sum[|b_i| : v_i \in V^-])$

$B_1 = \max(|b_i| : v_i \in V)$

$U = \max\{d_{ij} : (v_i, v_j) \in E_B\} \leq \max\{u_i : v_i \in V\}$

$C = \max\{h_{ij} : (v_i, v_j) \in E_2\} = \max\{\{c_{ij} : (v_i, v_j) \in E\} \cup \{u_i : v_i \in V\}\}$

O Algoritmo II tem maior complexidade que o Algoritmo III, que utilize o mesmo algoritmo para resolver os problemas de seleção, quando  $O(U) > n \cdot \log_2 U$  e menor complexidade caso contrário.

Para uma comparação teórica entre os algoritmos acima, considerando apenas as partes relativas às complexidades da resolução dos problemas de seleção, o Algoritmo I teria maior complexidade que o Algoritmo II, visto que o Algoritmo II resolve não só um número menor de problemas de seleção, mas também estes problemas, para este algoritmo, são bipartidos. Entretanto nos Algoritmos II e III, o digrafo é inicialmente transformado em um digrafo bipartido, através do cálculo de caminhos mínimos ( $O(|V^+| \cdot (m+n \cdot \min(\log_2 n, (\log_2 C)^{\frac{1}{2}})))$ ). Portanto, o Algoritmo II tem menor complexidade que o Algoritmo I, que utilize o mesmo algoritmo para resolver os problemas de seleção, sempre que a complexidade do Algoritmo I é maior que  $O(|V^+| \cdot n \cdot \min(\log_2 n, (\log_2 C)^{\frac{1}{2}}))$ . O Algoritmo I tem menor complexidade no caso contrário. Da mesma forma, o Algoritmo III tem menor complexidade que o Algoritmo I, que utilize o mesmo algoritmo para resolver os problemas de seleção, sempre que a complexidade do Algoritmo I é maior que  $O(|V^+| \cdot n \cdot \min(\log_2 n, (\log_2 C)^{\frac{1}{2}}))$  e  $O(U_1) > n \cdot \log_2 U$ . O Algoritmo I tem menor complexidade que o Algoritmo III, no caso contrário.

Contudo, devido a forma como são feitos os cálculos das complexidades, estas comparações são válidas apenas para problemas de grandes dimensões.

### 5.3- Algumas comparações práticas entre os algoritmos.

Nesta secção são feitos estudos comparativos do comportamento prático destes algoritmos. Ou seja, é apresentada uma comparação dos tempos de execução de uma implementação dos algoritmos em Turbo Pascal, para alguns problemas gerados aleatoriamente, em um PC-compatível, que utiliza um microprocessador 8088 e opera à 8 MHz. Os problemas testados têm cem vértices e foram considerados grupos de problemas, em que em cada grupo, apenas um dos parâmetros mencionados tem seu valor alterado. Foram gerados, resolvidos e medidos vinte problemas de cada tipo. As medidas para cada tipo de problema foram número de iterações (mínimo, máximo e médio) e tempo de execução (mínimo, máximo, e médio). O número de iterações representa para estes problemas, também, o número de vezes que o referente é calculado. São apresentadas, nesta secção tabelas contendo os tipos de problemas testados e para cada um destes tipos o número médio de iterações e o tempo médio de execução (em centésimos de segundo). Os tempos de execução no microcomputador utilizado podem ter, entretanto, um erro de 6 centésimos de segundo. Na secção 5.3.1, são apresentados os dados relativos aos programas dos algoritmos A e B de resolução do problema de b-emparelhamento máximo bipartido. Na secção 5.3.2, são apresentados os dados relativos aos programas dos algoritmos II-A e o de Picard-Dinic de resolução do problema de seleção. Na secção 5.3.3, são

apresentados os dados relativos aos programas dos Algoritmos I-II-A, II-II-A, III-II-A de resolução do problema de seleção generalizada.

### 5.3.1- Problema de b-emparelhamento máximo bipartido.

Os números médios de iterações e os tempos médios de execução dos programas dos dois algoritmos (A, B, vide capítulo 2) de resolução do problema de b-emparelhamento máximo, para alguns problemas gerados aleatoriamente, são encontradas na tabela a seguir. Cada linha da tabela corresponde a um tipo de problema e para cada tipo foram utilizados vinte problemas gerados aleatoriamente. Não foram encontradas grandes variações entre o máximo e o mínimo, nem no número de iterações nem no tempo de execução, para problemas de um mesmo tipo, por este motivo a tabela apresenta apenas o número médio de iterações e o tempo médio de execução.

Para problemas de dimensão semelhante aos testados o Algoritmo A mostrou-se superior. Ou seja, a técnica de escalonamento, embora seja, na teoria para alguns casos, superior, nos testes realizados (problemas relativamente pequenos) foi consistentemente inferior.

O parâmetro com maior influência sobre o tempo médio de execução e sobre o número médio de iterações para o Algoritmo A foi o número de arestas do grafo. Para o Algoritmo B, o parâmetro com maior influência sobre o tempo médio de execução e sobre o número de iterações foi, também o número de arestas do grafo, mas neste caso notou-se também uma considerável influência do valor de  $B_i$  ( $=\max\{|b_i| : i=1,2,\dots,n\}$ ) e do valor de  $|V^+|$ . Para os problemas testados, verifica-se que os tempos médios de execução dos Algoritmos A e B, crescem com o número de arestas, enquanto os números de iterações decrescem com o número de arestas.

				Algoritmo A		Algoritmo B	
$ V^+ $	$ V^- $	m	$B_i$	iterações	tempo	iterações	tempo
50	50	200	200	6	27	31	108
50	50	500	200	3	32	23	157
50	50	1000	200	3	54	21	263
50	50	1500	200	2	72	19	339
50	50	2000	200	2	98	17	412

Tabela 1 com variação do número de arestas.

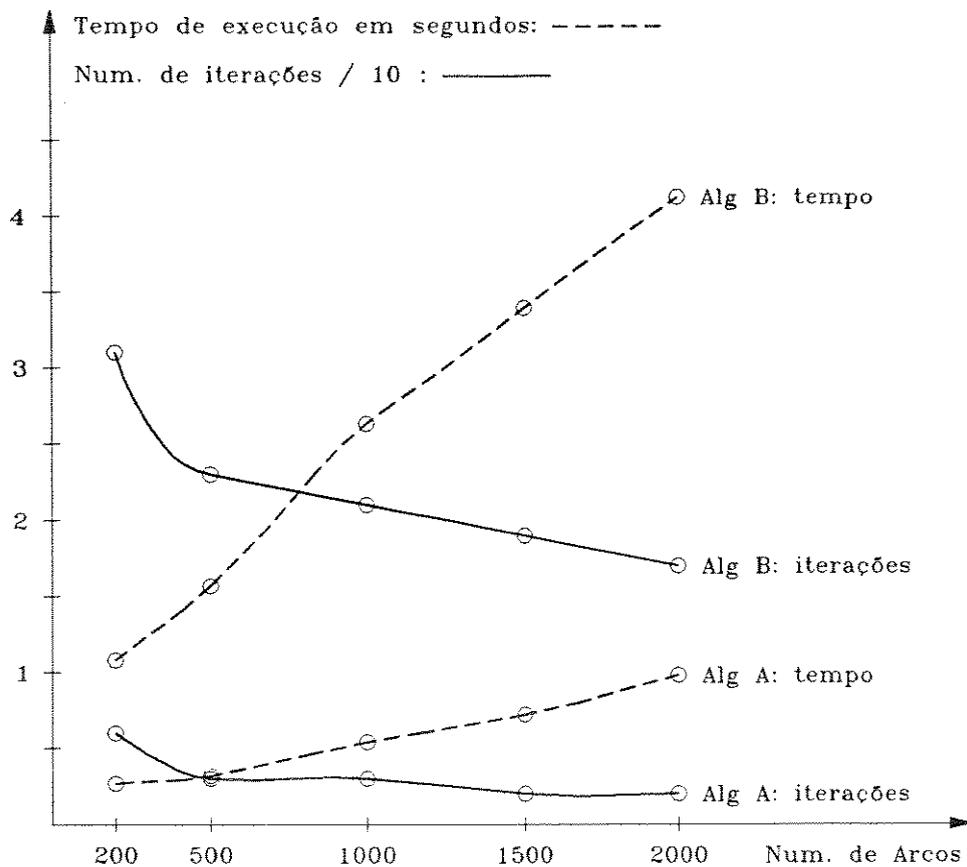
				Algoritmo A		Algoritmo B	
$ V^+ $	$ V^- $	m	$B_1$	iterações	tempo	iterações	tempo
50	50	500	10	3	32	12	82
50	50	500	100	3	32	20	138
50	50	500	200	3	32	23	157
50	50	500	400	3	34	26	184
50	50	500	800	3	34	28	213
50	50	500	1000	3	35	28	206
50	50	500	1500	3	33	31	229
50	50	500	2000	3	35	30	233

Tabela 2 com variação de  $B_1$  ( $=\text{Máx}\{|b_i| : i=1,2..n\}$ ).

				Algoritmo A		Algoritmo B	
$ V^+ $	$ V^- $	m	$B_1$	iterações	tempo	iterações	tempo
10	90	500	200	2	26	15	132
20	80	500	200	2	25	16	130
30	70	500	200	2	31	16	126
40	60	500	200	3	31	19	132
50	50	500	200	3	32	23	157

Tabela 3 com variação do número de vértices de  $V^+$ .

O gráfico, a seguir, ilustra as variações dos tempos de execução e do número de iterações para os algoritmos A e B com o número de arestas do grafo.



### 5.3.2- Problema de seleção.

Os números médios de iterações e os tempos médios de execução dos programas de resolução do problema de seleção em um digrafo qualquer, para alguns problemas gerados aleatoriamente, através dos algoritmos II-A e Picard são encontrados nas tabelas apresentadas a seguir. O Algoritmo II-B não está sendo considerado devido ao seu fraco desempenho (secção 5.3.1) e o Algoritmo I não está sendo considerado devido a utilizar o método de Picard sobre o digrafo transformado. O método de Picard foi implementado utilizando o método de Dinic, para o cálculo do fluxo máximo, adaptado para digrafos com arcos com capacidade ilimitada. As tabelas 4, 5 e 6 correspondem a problemas de seleção em digrafos que não são bipartidos com bipartição  $(V^+, V^-)$  e as tabelas 7, 8 e 9, a problemas de seleção em digrafos que são bipartidos com bipartição  $(V^+, V^-)$ . Cada linha das tabelas corresponde a um tipo de problema e para cada tipo foram utilizados vinte problemas gerados aleatoriamente. Não foram encontradas grandes variações entre o máximo e o mínimo, nem no número de iterações nem no tempo de execução, para problemas de um mesmo tipo, por este motivo as tabelas apresentam apenas o número médio de iterações e o tempo médio de execução.

Para problemas de dimensão semelhante aos testados o Algoritmo de Picard, embora apresentando um número um pouco maior de iterações, apresentou um tempo menor de execução. Isto é provocado pelo tempo utilizado no Algoritmo II-A, na transformação do problema de seleção em um problema de seleção em um digrafo bipartido. Este fato justifica a proximidade dos resultados apresentados nas tabelas para digrafos bipartidos do tipo  $((V^+, V^-), E)$ .

O parâmetro com maior influência sobre o tempo médio de execução dos dois algoritmos para os dois tipos de digrafos testados, foi o número de arcos do digrafo. Para o Algoritmo II-A aplicado a problemas de seleção, definidos em digrafos que não são bipartidos do tipo  $((V^+, V^-), E)$ , notou-se, também, que o tempo médio de execução apresentou grande variação com relação ao número de vértices em  $V^+$  e em  $V^-$ ; em muito menor escala esta variação, também está presente para os mesmos problemas, resolvidos pelo Algoritmo de Picard.

				Algoritmo II-A		Alg. Picard-Dinic.	
$ V^+ $	$ V^- $	m	$B_1$	iterações	tempo	iterações	tempo
50	50	200	200	2	494	10	50
50	50	500	200	2	982	7	66
50	50	1000	200	2	1069	5	91
50	50	1500	200	2	1140	4	134
50	50	2000	200	2	1215	4	201

Tabela 4 com variação do número de arcos para G qualquer.

				Algoritmo II-A		Alg.Picard-Dinic.	
$ V^+ $	$ V^- $	m	$B_1$	iterações	tempo	iterações	tempo
50	50	500	10	2	988	7	64
50	50	500	100	2	988	7	63
50	50	500	200	2	982	7	66
50	50	500	400	2	987	7	65
50	50	500	800	2	988	6	60
50	50	500	1000	2	978	7	68
50	50	500	1500	2	982	7	67
50	50	500	2000	2	983	7	71

Tabela 5 com variação de  $B_1$  ( $=\text{Máx}\{|b_i| : i=1,2..n\}$ ) para G qualquer.

				Algoritmo II-A		Alg.Picard-Dinic.	
$ V^+ $	$ V^- $	m	$B_1$	iterações	tempo	iterações	tempo
10	90	500	200	2	218	2	24
20	80	500	200	2	413	3	30
30	70	500	200	2	615	4	34
40	60	500	200	2	795	5	42
50	50	500	200	2	982	7	66

Tabela 6 com variação do número de vértices de  $V^+$ , para G qualquer.

				Algoritmo II-A		Alg.Picard-Dinic.	
$ V^+ $	$ V^- $	m	$B_1$	iterações	tempo	iterações	tempo
50	50	200	200	6	53	7	48
50	50	500	200	3	82	4	67
50	50	1000	200	2	157	3	128
50	50	1500	200	2	252	3	213
50	50	2000	200	2	364	2	318

Tabela 7 com variação do número de arcos para  $G=(V^+,V^-),E$  bipartido.

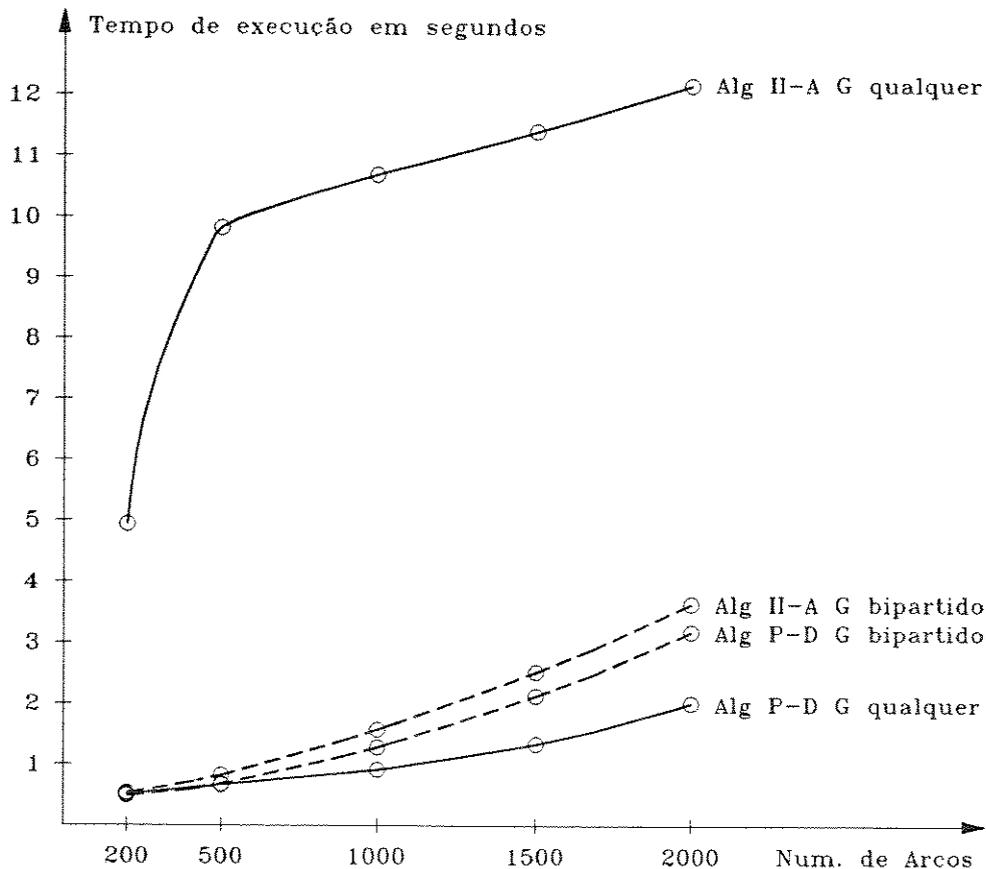
$ V^+ $	$ V^- $	m	$B_1$	Algoritmo II-A		Alg. Picard-Dinic.	
				iterações	tempo	iterações	tempo
50	50	500	10	3	79	4	65
50	50	500	100	3	82	4	68
50	50	500	200	2	82	4	67
50	50	500	400	3	82	4	70
50	50	500	800	3	80	4	68
50	50	500	1000	3	82	4	67
50	50	500	1500	3	81	5	69
50	50	500	2000	3	81	4	67

Tabela 8 com variação de  $B_1$  ( $=\text{Máx}\{|b_i| : i=1,2..n\}$ ), para  $G$  bipartido.

$ V^+ $	$ V^- $	m	$B_1$	Algoritmo II-A		Alg. Picard-Dinic.	
				iterações	tempo	iterações	tempo
10	90	500	200	2	83	2	69
20	80	500	200	2	73	2	55
30	70	500	200	2	76	2	55
40	60	500	200	3	78	4	63
50	50	500	200	3	82	4	67

Tabela 9 com variação do número de vértices de  $V^+$  para  $G=(V^+,V^-),E$  bipartido.

O gráfico a seguir ilustra as variações dos tempos de execução com o número de arcos do digrafo  $G$ , nos dois casos  $G$  bipartido e  $G$  qualquer, para o Algoritmo I e o Algoritmo de Picard.



### 5.3.3- Problema de seleção generalizada.

Os números médios de iterações e os tempos médios de execução dos programas de resolução do problema de seleção generalizada em um digrafo qualquer, para alguns problemas gerados aleatoriamente, através dos Algoritmos I-II-A, II-II-A e III-II-A são encontrados nas tabelas apresentadas a seguir. O Algoritmo II-A foi escolhido para resolver os problemas de seleção para efeito de comparação dos métodos I, II e III; entretanto o uso de outro algoritmo não faria alterações qualitativas na análise. As tabelas 10 a 14 correspondem aos problemas de seleção generalizada em digrafos que não são bipartidos com bipartição  $(V^+, V^-)$  e as tabelas 15 a 19, aos problemas de seleção generalizada em digrafos que são bipartidos com bipartição  $(V^+, V^-)$ . Cada tabela corresponde a problemas de tipos que diferem pelo valor de um parâmetro. Cada linha das tabelas corresponde a um tipo de problema e para cada tipo foram utilizados vinte problemas gerados aleatoriamente. Não foram encontradas grandes variações entre o máximo e o mínimo, nem no número de iterações nem no tempo de execução, para problemas de um mesmo tipo, por este motivo as tabelas apresentam apenas o número médio de iterações e o tempo médio de execução.

Para problemas de dimensão semelhante aos testados o Algoritmo I, embora apresentando, em geral, um número um pouco maior de iterações, apresentou um tempo menor de execução. Isto é provocado pelo tempo utilizado nos Algoritmos II e III, na transformação do problema em um problema de seleção generalizada em um digrafo bipartido. Os três algoritmos utilizam transformações de digrafos em digrafos bipartidos. Os Algoritmos II e III, modificam logo no início o digrafo dado através do cálculo de caminhos mínimos em um digrafo bipartido, enquanto o Algoritmo I modifica alguns subgrafos do digrafo dado, através do cálculo do fecho transitivo destes subgrafos. O procedimento utilizado nos Algoritmos II e III, embora teoricamente interessante, para os problemas testados demonstrou-se pouco interessante, pelo tempo utilizado nesta transformação. O procedimento utilizado nos Algoritmos I, apesar de ser realizado mais de uma vez, por ser mais simples e por envolver sempre subgrafos do digrafo dado, em média digrafos esparsos, mostrou-se mais eficiente para os problemas testados.

Os tempos médios de execução em função do número de arcos do digrafo para os Algoritmos II e III apresentam comportamentos diferentes dependendo de se  $G$  é bipartido  $((V^+, V^-), E)$  ou não. Se  $G$  é bipartido o tempo de execução para os dois algoritmos é uma função decrescente do número de arcos do digrafo; enquanto se  $G$  não é bipartido é uma função decrescente para a primeira metade do conjunto de número de arcos testados e é uma função crescente para a segunda metade do mesmo conjunto. A diferença de comportamento é explicada pelo tempo utilizado no cálculo de  $G_B$ , que para  $G$  qualquer é uma função crescente do número de arcos. O tempo de execução para o cálculo de  $G_B$  não é tão significativo para digrafos bipartidos do tipo  $((V^+, V^-), E)$ . Este fato justifica a proximidade dos resultados apresentados para estes digrafos, pelos três algoritmos.

Para variações dos demais parâmetros não foram observadas diferenças significativas de comportamento do tempo médio de execução para digrafos quaisquer ou bipartidos.

O parâmetro com menor influência sobre o tempo médio de execução dos três algoritmos foi o  $B_1$  e o parâmetro com maior influência foi o número de vértices de  $V^+$  no digrafo. Os tempos médios de execução dos Algoritmos II e III foram bastante próximos, sendo interessante notar, que o Algoritmo II apresentou tempo médio de execução menor que o Algoritmo III para os menores valores de  $U_{\max}$ ,  $C_{\max}$  e  $|V^+|$  e maior, que o Algoritmo III para os maiores valores destes parâmetros. Como esperado, para os maiores valores de  $C_{\max}$  e  $U_{\max}$  utilizados o Algoritmo III apresentou menor tempo que o Algoritmo II; mas o Algoritmo I foi o menor tempo, também neste caso.

Algorit.I Algorit.II Algorit.III											
$V^+$	$V^-$	m	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
50	50	200	200	200	100	119	1368	111	6042	75	5549
50	50	500	200	200	100	141	1749	115	4493	90	4253
50	50	1000	200	200	100	127	1807	97	3642	86	3814
50	50	1500	200	200	100	112	1863	88	3664	82	4077
50	50	2000	200	200	100	99	1850	77	4103	81	4706

Tabela 10 com variação do número de arcos para G qualquer.

Algorit.I Algorit.II Algorit.III											
$V^+$	$V^-$	m	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
50	50	500	10	200	100	133	1632	105	4100	84	3942
50	50	500	100	200	100	146	1743	114	4398	89	4301
50	50	500	200	200	100	141	1749	115	4493	90	4253
50	50	500	400	200	100	143	1740	114	4376	87	4159
50	50	500	800	200	100	136	1681	112	4532	85	4172
50	50	500	1000	200	100	152	1802	117	4472	93	4337
50	50	500	1500	200	100	137	1719	114	4668	88	4412
50	50	500	2000	200	100	146	1686	111	4271	87	4222

Tabela 11 com variação de  $B_1$  ( $=\text{Máx}\{|b_i| : i=1,2..n\}$ ) para G qualquer.

Algorit.I Algorit.II Algorit.III											
$V^+$	$V^-$	m	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
10	90	500	200	200	100	29	233	28	695	43	1221
20	80	500	200	200	100	54	541	52	1650	57	2024
30	70	500	200	200	100	90	942	77	2738	77	3118
40	60	500	200	200	100	111	1273	94	3755	81	3869
50	50	500	200	200	100	141	1749	115	4493	90	4253

Tabela 12 com variação do número de vértices de  $v^+$  para G qualquer.

						Algorit.I		Algorit.II		Algorit.III	
$V^+$	$V^-$	$m$	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
50	50	500	200	10	10	22	268	23	1346	32	2658
50	50	500	200	100	100	105	1225	94	4439	70	4511
50	50	500	200	200	100	141	1749	115	4493	90	4253
50	50	500	200	300	100	168	1972	123	4444	98	4083

Tabela 13 com variação de  $U_{\max} = \max\{|u_i| : i=1, 2, \dots, n\}$  para  $G$  qualquer.

						Algorit.I		Algorit.II		Algorit.III	
$V^+$	$V^-$	$m$	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
50	50	500	200	300	10	65	800	51	2391	65	3556
50	50	500	200	300	100	168	1972	123	4444	98	4083
50	50	500	200	300	200	164	1949	129	5434	93	4643
50	50	500	200	300	300	143	1676	120	5696	87	5058

Tabela 14 com variação de  $C_{\max} = \max\{|c_{ij}| : (v_i, v_j) \in E\}$  para  $G$  qualquer.

						Algorit.I		Algorit.II		Algorit.III	
$V^+$	$V^-$	$m$	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
50	50	200	200	200	100	158	1601	125	4063	92	3673
50	50	500	200	200	100	155	1602	109	2782	97	2593
50	50	1000	200	200	100	126	1519	92	2061	98	2278
50	50	1500	200	200	100	105	1437	78	1735	88	2193
50	50	2000	200	200	100	100	1459	73	1669	87	2172

Tabela 15 com variação do número de arcos para  $G = ((V^+, V^-), E)$  bipartido.

Algorit.I Algorit.II Algorit.III											
$V^+$	$V^-$	m	$B_1$	$U_{\max}$	$C_{\max}$	iter tempo		iter tempo		iter tempo	
50	50	500	10	200	100	142	1502	98	2698	88	2512
50	50	500	100	200	100	152	1631	111	2902	97	2574
50	50	500	200	200	100	155	1602	109	2782	97	2593
50	50	500	400	200	100	153	1650	113	2920	97	2680
50	50	500	800	200	100	145	1536	105	2749	96	2641
50	50	500	1500	200	100	159	1687	114	2957	101	2576
50	50	500	2000	200	100	153	1613	109	2779	97	2541

Tabela 16 com variação de  $B_1 (= \max\{|b_i| : i=1,2,..n\})$  para  $G$  bipartido.

Algorit.I Algorit.II Algorit.III											
$V^+$	$V^-$	m	$B_1$	$U_{\max}$	$C_{\max}$	iter tempo		iter tempo		iter tempo	
10	90	500	200	200	100	16	128	15	385	29	676
20	80	500	200	200	100	37	320	33	776	45	1226
30	70	500	200	200	100	66	663	57	1439	63	1857
40	60	500	200	200	100	104	1086	84	2187	81	2366
50	50	500	200	200	100	155	1602	109	2782	97	2593

Tabela 17 com variação do número de vértices de  $V^+$  para  $G=(V^+,V^-),E$  bipartido.

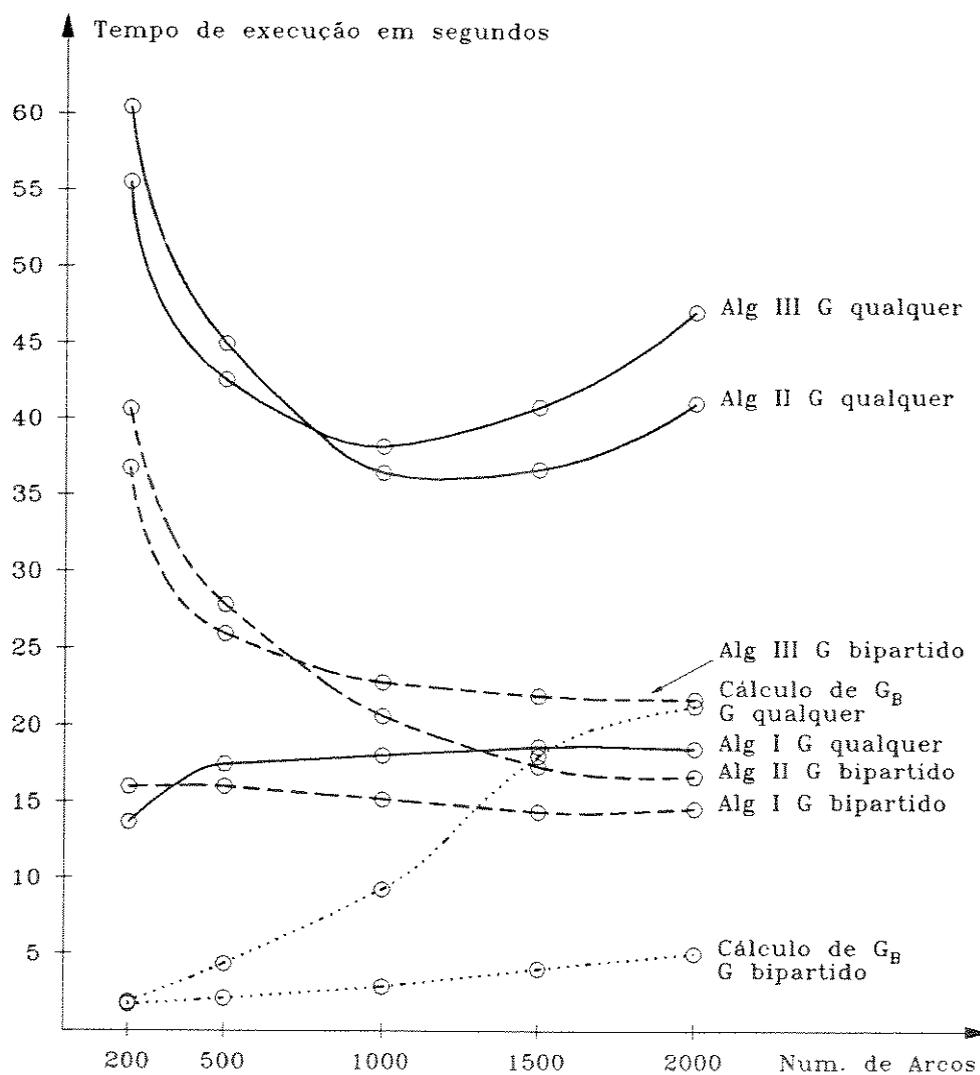
Algorit.I Algorit.II Algorit.III											
$V^+$	$V^-$	m	$B_1$	$U_{\max}$	$C_{\max}$	iter tempo		iter tempo		iter tempo	
50	50	500	200	10	10	23	226	22	713	36	1719
50	50	500	200	100	100	117	1190	94	2746	81	3074
50	50	500	200	200	100	164	1697	115	2901	100	2532
50	50	500	200	300	100	163	1753	115	2811	103	2660

Tabela 18 com variação de  $U_{\max}=\max\{|u_i| : i=1,2,..n\}$  para  $G$  bipartido.

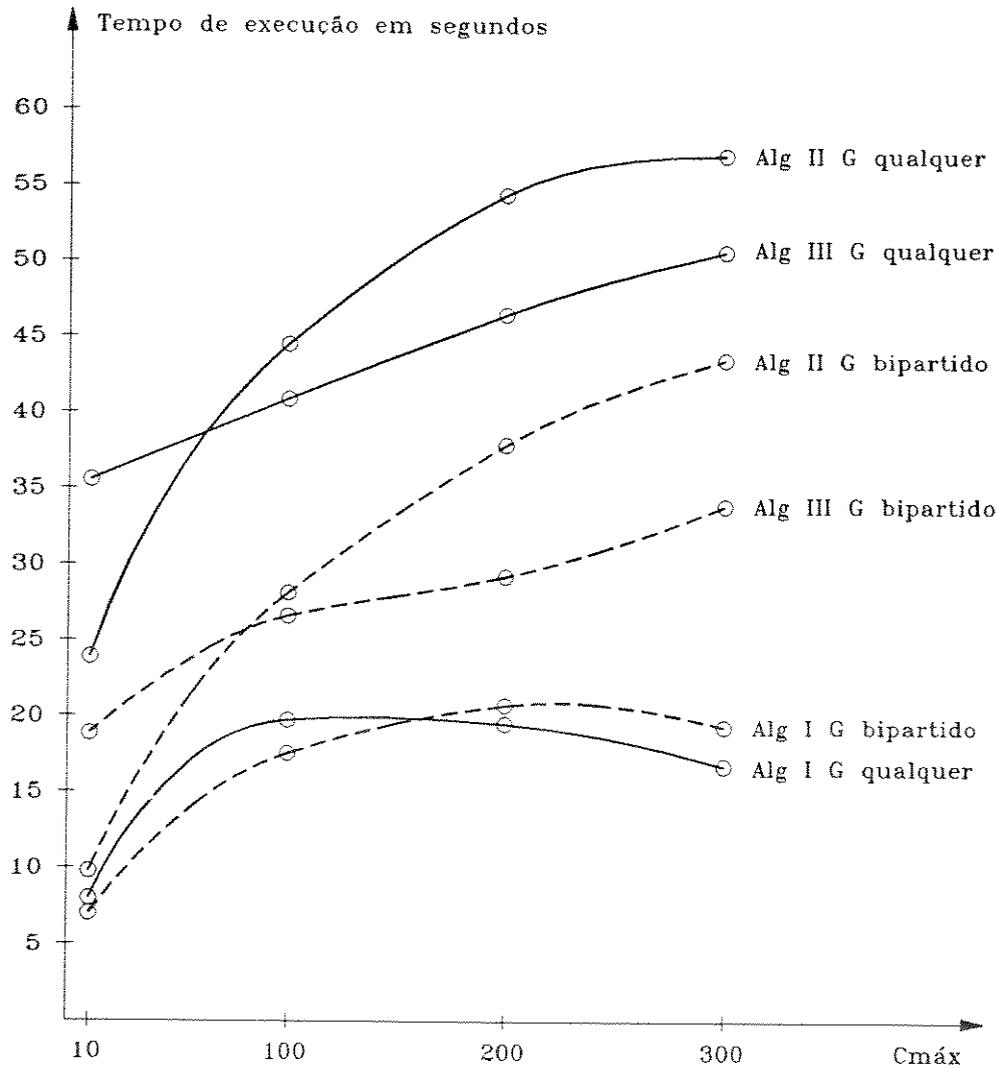
<div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px;"> <span>Algorit.I</span> <span>Algorit.II</span> <span>Algorit.III</span> </div>											
$V^+$	$V^-$	$m$	$B_1$	$U_{\max}$	$C_{\max}$	iter	tempo	iter	tempo	iter	tempo
50	50	500	200	300	10	76	701	56	979	77	1885
50	50	500	200	300	100	163	1753	115	2811	103	2660
50	50	500	200	300	200	195	2073	134	3785	106	2922
50	50	500	200	300	300	190	1933	136	4347	99	3390

Tabela 19 com variação de  $C_{\max} = \max\{|c_{ij}| : (v_i, v_j) \in E\}$  para  $G$  bipartido.

O gráfico a seguir ilustra as variações dos tempos de execução com o número de arcos do digrafo  $G$ , nos dois casos  $G$  bipartido e  $G$  qualquer, para os algoritmos I, II e III e, também, para a transformação do digrafo utilizada pelos algoritmos II e III.



O gráfico a seguir ilustra as variações dos tempos de execução com o valor de  $C = \max\{c_{ij} : (v_i, v_j) \in E\}$  do digrafo  $G$ , nos dois casos  $G$  bipartido e  $G$  qualquer, para os algoritmos I, II e III.



#### 5.4- Conclusões.

No estudo do problema de b-emparelhamento máximo bipartido dois novos algoritmos são propostos. No primeiro (Algoritmo A) é feita uma generalização do Algoritmo de Hopcroft e Karp [HpK73]. Esta generalização consiste em adaptar um algoritmo de um problema 0,1 para um problema de programação inteira. A complexidade do Algoritmo A é provada através de uma série de novos teoremas que generalizam os resultados obtidos por Hopcroft e Karp [HpK73] e um resultado obtido por Even e Tarjan [EvT75]. No segundo algoritmo

(Algoritmo B) é feito um estudo original da utilização da técnica de escalonamento de Edmonds e Karp [EdK72] (apresentada para o problema de fluxo de custo mínimo) para o problema de b-emparelhamento máximo bipartido.

No estudo do problema de seleção dois novos algoritmos são propostos. Os dois algoritmos utilizam a equivalência entre um problema de seleção em um digrafo qualquer e um problema de seleção em um digrafo bipartido, estabelecida no Teorema 2.1 (novo). Este resultado foi importante para a transformação do problema, utilizada pelo Algoritmo II, em um problema de b-emparelhamento máximo bipartido.

A resolução do problema de seleção através do problema de fluxo máximo, não tira proveito, no cálculo do fluxo dos arcos do digrafo  $G$  não terem limitante superior. Na realidade nenhum corte mínimo pode apresentar um arco de  $E$  (com capacidade infinita) direcionado de  $S$  para  $T$ , onde  $(S, T)$  é a bipartição de vértices determinada pelo corte mínimo com  $s \in S$  e  $t \in T$ . No problema de b-emparelhamento máximo os arcos não têm limitante superior, logo o problema de seleção se enquadra melhor neste problema, isto é o cálculo da complexidade do algoritmo que utiliza o problema de b-emparelhamento fica mais próximo do esforço computacional real. Entretanto, para os problemas testados (problemas relativamente pequenos) a transformação do digrafo em um digrafo bipartido, foi relativamente demorada e na implementação do fluxo máximo utilizada os arcos foram considerados sem limitação em sua capacidade. A implementação realizada não permitiu trabalhar com digrafos com número muito grande de vértices, devido a restrições do Turbo Pascal.

No estudo do problema de seleção generalizada apresentado, a ênfase foi dada na determinação das variáveis duais do problema de fluxo de custo mínimo. Os métodos, mais recentes, de resolução do problema de fluxo de custo mínimo utilizam o cálculo de um grande número de caminhos mínimos em um digrafo ou o cálculo de um certo número de fluxos máximos. Um processo de resolução para o problema de seleção generalizada, que não calculasse um número muito grande de caminhos mínimos e tirasse proveito de não haver para este problema canalização nos arcos foi procurado. Um novo algoritmo de resolução que ao invés de resolver em cada iteração problemas de fluxo máximo resolvem problemas de b-emparelhamento máximo bipartido, possibilitando uma melhor avaliação do esforço computacional, foi obtido. Os algoritmos propostos para o problema de seleção generalizada utilizam um processo de deslocamento da solução viável e de uma redução da região viável, que não é aplicável a um programa linear qualquer, mas é válida para o problema de seleção generalizada, como demonstrado no Teorema 4.1 (novo).

Um dos algoritmos apresentados consiste, em determinar, em cada iteração, uma melhora do valor da função objetivo. A direção de deslocamento da solução, corresponde a uma direção viável de subida da função, e está relacionada com o gradiente da função objetivo e com os limites da região de viabilidade, mas leva sempre em consideração que a solução ótima desejada deve ser inteira. (Deslocamentos na direção do gradiente ou de uma sua projeção no conjunto de soluções viáveis, não garantem alcançar a fronteira em um ponto com componentes inteiras).

Tendo por base o problema de determinação de caminhos mínimos em um digrafo, foi desenvolvido algoritmo para modificação do problema transformando-o em um problema de seleção generalizada em um digrafo bipartido, esta transformação é feita com o cálculo de no máximo  $n$  (número de vértices) problemas de caminho mínimo.

Foram utilizados, para a resolução dos problema de seleção estudados, transformações do problema em um digrafo qualquer para um problema do mesmo tipo em um digrafo bipartido. Seria interessante verificar para que outros tipos de problemas, esta transformação é possível e facilita ou possibilita a determinação de uma solução ótima.

Diversos trabalhos têm sido apresentados para os problemas de fluxo, utilizando algumas estruturas de dados especiais para a representação dos dados do problema, de maneira a teoricamente diminuir o número de operações necessárias para a resolução do problema. Um trabalho interessante a ser realizado seria um estudo sobre a melhor estrutura de dados a utilizar, para melhorar a complexidade dos algoritmos e verificar, se a esta melhora teórica corresponderia uma melhora, em termos práticos dos algoritmos. Por exemplo a árvore dinâmica utilizada por Sleator e Tarjan [SIT83] para o problema do fluxo máximo, embora teoricamente seja melhor do que o método de Dinic apresenta na prática, um comportamento pior, segundo os próprios autores.

Diversos trabalhos têm sido apresentados recentemente com algoritmos paralelos para a resolução de problemas combinatórios. Um estudo da possibilidade de uma implementação paralela para os algoritmos apresentados poderia ser realizado.

## REFERÊNCIAS BIBLIOGRÁFICAS.

- [AHU76]- Aho,A.V., Hopcroft,J.E. e Ullman,J.D.- “The Design and Analysis of Computer Algorithms.”- Addison-Wesley Publishing Co., (1976).
- [AMO90]- Ahuja,R.K., Mehlhorn,K., Orlin,J. e Tarjan,R.E.- “Faster Algorithms for the Shortest Path Problem.”- Journal of the ACM, Vol 37, N° 2 (April 1990).
- [AOT89]- Ahuja,R.K., Orlin,J.B. e Tarjan,R.E.- “Improved Time Bounds for the Maximum Flow Problem.”- Siam Journal Comput., Vol 18, N° 5 (October 1989).
- [BdM76]- Bondy,J.A. e Murty,U.S.R.- “Graph Theory with Applications.”- American Elsevier Publishing Co.,Inc., (1976).
- [Bge62]- Berge,C.- “The Theory of Graphs and its Applications.”- Methuen and Co. Limited, London, (1962).
- [Bge76]- Berge,C.- “Graphs and Hypergraphs.”- North-Holland / American Elsevier, 2nd Edition (1976).
- [Bkr74]- Baker,K.R.- “Introduction to Sequencing and Scheduling.”- John Willey and Sons, Inc., (1974).
- [Blk70]- Balinski,M.L.- “On a Selection Problem.”- Management Science, Vol 17, N° 3 (November 1970).
- [Blk86]- Balinski,M.L.- “A Competitive (Dual) Simplex Method for the Assignment Problem.”- Mathematical Programming, 34 (1986).
- [BsB88]- Brassard,G. e Bratley,P.- “Algorithmics Theory and Practice.” - Prentice Hall International,Inc, (1988).
- [CfC87]- Conforti,M. e Cornuejois,G.- “An Algorithmic Framework for the Matching Problem in Some Hypergraphs”- Networks, Vol 17, (1987).
- [Cmn65]- Camion,P.- “Characterization of Totally Unimodular Matrices”- Proc. of the American Math. Soc., Vol 16, N° 5, (October 1965).
- [Djk59]- Dijkstra, E. W.- “A Note on Two Problems in Connexion with Graphs.”- Numer. Math. I, 4, (1959).
- [Dnc70]- Dinic,E.A.- “Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation.”- Soviet Math. Dokl., Vol 11, N° 5 (1970).

- [Drg86]- Derigs,U.- “Solving Large-Scale Matching Problems Efficiently : A New Primal Approach.”- Network, Vol 16, (1986).
- [Dtz63]- Dantzig,G.B.- “Linear Programming and Extensions.”- Princeton University Press, (1963).
- [EdJ73]- Edmonds,J e Johnson,E.L.- “Matching, Euler Tours and the Chinese Postman.”- Mathematical Programming, 5, (1973).
- [EdK68]- Edmonds,J. e Karp,R.M.- “A Technique for Acelerating the Solution of Transportation Problems.”- 2nd Annual Princeton Conf. on Information Sc., (March 1968).
- [EdK72]- Edmonds,J. e Karp,R.M.- “Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems.”- Journal of the A.C.M., Vol 19, N° 2 (April 1972).
- [Edm64]- Edmonds,J.- “Maximum Matching and a Poliedron with 0,1-Vertices”- Journal of Research of the National B. Standards, vol 69B, N° 1,2, (1965).
- [Edm65]- Edmonds,J.- “Paths, Trees and Flowers.”- Canadian Journal Math., 17, (1965).
- [Evn79]- Even,S.- “Graph Algoritms.”- Computer Science Press, Inc., (1979).
- [EvT75]- Even,S. e Tarjan,R.E.- “Network Flow and Testing Graph Connectivity.”- SIAM Journal Comput., Vol 4, N° 4 (December 1975).
- [FBM89]- Fernandez-Baca,D. e Martel,C.U.- “On the Efficiency of Maximum-Flow Algorithms on Networks with Small Integer Capacities”- Algorithmica, N° 4 (1989).
- [FdF62]- Ford, L.R e Fulkerson,D.R.- “Flows in Networks.”- Princeton University Press, (1962).
- [FdF56]- Ford, L.R e Fulkerson,D.R.- “Maximal Flow through a Network.”- Canadian J. Math., Vol 8, (1956).
- [Fjs86]- Fujishige,S.- “A Capacity-Rounding Algorithm for the Minimum-Cost Circulation Problem: A Dual Framework of the Tardos Algorithm.”- Mathematical Programming 35, (1986).
- [Fks61]- Fulkerson,D.R.- “A Network Flow Computation for Project Cost Curves.”- Management Science (1961).
- [FmT87]- Fredman,M.L. e Tarjan,R.A.- “Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms.”- Journal of the A.C.M.,Vol 34, N° 3 (July 1987).
- [GbT88]- Gabow,H. e Tarjan,R.E.- “Almost-Optimum Speed-ups of Algorithms for Bipartite Matching and Related Problems.”- Proc. of the 20<sup>th</sup> Annual ACM Sym. on Theory of Computing, (1988).

- [GbT89]- Gabow,H. e Tarjan,R.E.- "Faster Scaling Algorithms for Network Problems."- Siam Journal of Comput., Vol 18, N° 5 (October 1989).
- [Gbw85]- Gabow,H.N.- "Scaling Algorithms for Network Problems."- Journal of Computer and System Sciences 31, (1985).
- [GdT87]- Goldberg,A.V e Tarjan,R.E.- "Solving Minimum-Cost Flow Problems by Successive Approximation."- Proc. of the 19<sup>th</sup> Annual ACM Sym. on Theory of Computing, (1987).
- [GdT88a]- Goldberg,A.V e Tarjan,R.E.- "Finding Minimum-Cost Circulations by Canceling Negative Cycles."- Proc. of the 20<sup>th</sup> Annual ACM Sym. on Theory of Computing, (1988).
- [GdT88b]- Goldberg,A.V e Tarjan,R.E.- "A New Approach to the Maximum-Flow Problem."- Journal of the A.C.M., Vol 16, N° 4 (October 1988).
- [Gll80]- Galil,Z.- "An  $v^{5/3} \cdot e^{2/3}$  Algorithm for the Maximal Flow Problem."- Acta Inf. 14 (1980).
- [GIN80]- Galil,Z. e Naamad,A.- "An  $O(EV \log^2 v)$  Algorithm for the Maximal Flow Problem."- Journal of Computer and System Sc., Vol 21, (1980).
- [GIT86]- Galil,Z. e Tardos,E.- "An  $O(n^2(m+n \cdot \log n) \cdot \log n)$  Min-Cost Flow Algorithm."- Proc. of the 27<sup>th</sup> Annual Symp. of Found. of Comput. Science, (1986).
- [GMF87]- Gushfield,D., Martel,C. e Fernandez-Baca,D.- "Fast Algorithms for Bipartite Network Flow."- Siam J. Comput., Vol 16, N°2, (April 1987).
- [GrJ79]- Garey,M.R. e Johnson,D.S.- "Computers and Intratibility, A Guide to the Theory of NP-Completeness."- W.H.Freeman and Company (1979).
- [HIL74]- Hillier,F.S. e Lieberman,G.J.- "Introduction to Operations Research."- Holden-Day, Inc. 2<sup>nd</sup> Edition (1974).
- [Hmc82]- Hamacher,H.- "Determining Minimal Cuts with a Minimal Number of Arcs"- Networks, Vol 12, (1982).
- [HoS78]- Horowitz,E. e Sahni,S.- "Fundamentals of Computer Algorithms."- Computer Science Press, Inc., (1978).
- [HpK73]- Hopcroft,J.E. e Karp,R.M.- "An  $n^{5/2}$  Algorithm for Maximum Matchings in Bipartite Graphs."- SIAM Journal Comput., Vol 2, N° 4 (December 1973).
- [Hrr72]- Harary,F.- "Graph Theory."- Addison-Wesley Publishing Company, (1972).
- [Hsn79]- Hansen,P.- "Methods of Nonlinear 0-1 Programming."- Annals of Discrete Mathematics, Vol 5, (1979).

- [Hu-70]- Hu,T.C.- “Integer Programming and Network Flows.”- Addison-Wesley Publishing Company, (1970).
- [Krp80]- Karp,R.M.- “An Algorithm to Solve the  $m \times n$  Assignment Problem in Expected Time  $O(m \cdot n \cdot \log n)$ .”- Networks, Vol 10 (1980).
- [Kzn74]- Karzanov,A.V.- “Determining the Maximal Flow in a Network by the Method of Preflows.”- Soviet Math. Dokl., Vol 15, N° 2 (1974).
- [LcG65]- Lerchs,H. e Grossman,I.F.- “Optimum Design of Open Pit Mines.”- Canadian Min. Met. Bull., Vol 58, N° 633 (January 1965).
- [LSK79]- Lucchesi,C., Simon,I., Simon,I., Kowaltowski,T.- “Aspectos Teóricos da Computação”- IMPA / CNPQ (1979).
- [LvP86]- Lovasz,L. e Plummer,M.D.- “Matching Theory.”- Elsevier Science Publishers (1986).
- [Lwl76]- Lawler,E.L.- “Combinatorial Optimization: Networks and Matroids.”- Holt. Rinehart and Winston, (1976).
- [McP89]- Machado,A.F. e Perin,C.F.- “Resolução do Problema de Seleção em um Digrafo Através do Problema de b-emparelhamento em um Grafo Bipartido.”- XXII Congresso da SOBRAPO, Fortaleza (outubro 1989).
- [McP90a]- Machado,A.F. e Perin,C.F.- “Resolução do Problema de Seleção em um Digrafo Através do Problema de b-emparelhamento em um Grafo Bipartido.”- Pesquisa Operacional, Vol 10, N° 1 (junho 1990).
- [McP90b]- Machado,A.F. e Perin,C.F.- “Uma Generalização do Problema de Seleção em um Digrafo.”- V CLAIO, Congresso da SADIO, Buenos Aires (setembro 1990).
- [McP90c]- Machado,A.F. e Perin,C.F.- “Escalonamento em uma Generalização do Problema de Seleção.”- XIII CNMAC, Congresso da SBMAC, Aguas de Lindóia (Novembro 1990).
- [McP91a]- Machado,A.F. e Perin,C.F.- “A Generalized Selection Problem Solved by Iterative Methods.”- Proc of the 13<sup>th</sup> World Congress on Computation and Applied Mathematics, Dublin, Irlanda, (July 1991).
- [McP91b]- Machado,A.F. e Perin,C.F.- “A Generalized Selection Problem Solved by an Iterative Maximum Flow Method.”- Submitted to the 14<sup>th</sup> International Symp. on Mathematical Programming, Amsterdam, Holanda, (August 1991).
- [McP91c]- Machado,A.F. e Perin,C.F.- “A Generalized Selection Problem in a Digraph Solved by Iterative Methods.”- Submitted to TIMS XXX - Sobrapo XXIII, RIO de Janeiro, (July 1991).
- [MKM78]- Malhotra,V.M., Kumar,M.P. e Maheshwari,S.N.- “An  $O(|V|^3)$  Algorithm for

Finding Maximum Flows in Networks.”- Information Processing Letters, Vol 7, N° 6 (October 1978).

- [Mnd89]- Mander,U.- “Introduction to Algorithms, A Creative Approach.”- Addison-Wesley Publishing Co., (1989).
- [Mrt83]- Murty,K.G.- “Linear Programming.”- John Willey and Sons, (1983).
- [Mrt89]- Martel,C.- “A comparison of Phase and Nonphase Networkflow Algorithms.”- Networks, Vol 19 (1989).
- [Ore62]- Ore,O.- “Theory of Graphs.”- American Mathematical Society, (1962).
- [Orl88]- Orlin,J.B.- “A Faster Strongly Polynomial Minimum Cost Flow Algorithm.”- Proc. of the A.C.M. Symp. Theory of Computing (1988).
- [PbR82]- Padberg,M.W. e Rao,M.R.- “Odd Minimum Cut-Sets and B-matching.”- Mathematics of Operations Research, Vol 7, N° 1 (February 1982).
- [PcR73]- Picard,J.C. e Ratliff,H.D.- “A Graph-Theoretic Equivalence for Integer Programs.”- Operations Research, Vol 21, (1973).
- [PcR73]- Picard,J.C. e Ratliff,H.D.- “Minimal Cost Cut Equivalent Networks.”- Management Science, Vol 19, N° 9 (May 1973).
- [PcR75]- Picard,J.C. e Ratliff,H.D.- “Minimum Cuts and Related Problems.”- Networks, Vol 5, (1975).
- [PcR80]- Picard,J.C. e Ratliff,H.D.- “A Cut Approach to a Class of Quadratic Integer Programming Problems.”- Networks, Vol 10, (1980).
- [Pic76]- Picard,J.C.- “Maximal Closure of a Graph and Applications to Combinatorial Problems.”- Management Science, Vol 22, N° 11 (July 1976).
- [Plb80]- Pulleyblank,W.- “Dual Integrality in B-Matching Problems.”- Mathematical Programming Study, 12, (1980).
- [PpS82]- Papadimitriou,C.H. e Steiglitz,K.- “Combinatorial Optimization: Algorithms and Complexity.”, Prentice Hall, (1982).
- [Rhy70]- Rhys,J.M.W.- “A Selection Problem of Shared Fixed Cost and Network Flows.”- Management Science, Vol 17, N° 3 (November 1970).
- [Roc80]- Rock,H.- “Scaling Techniques for Minimal Cost Network Flows.”- Discrete Structures and Algorithms, Page, Carl Hansen, Munich (1980).
- [Scr86]- Schrijver,A.- “Theory of Linear and Integer Programming.”- John Wiley & Sons (1986).
- [SIT83]- Sleator,D.D. e Tarjan,R.- “A Data Structure for Dynamic Trees.”- Journal of Computer and System Science, N° 26 (1983).

- [Spr79]- Shapiro,J.F.- "A Survey of Lagrangean Techniques for Discrete Optimization."- Annals of Discrete Mathematics, 5, (1979).
- [Szw86]- Szwarcfiter,J.L.- "Grafos e Algoritmos Computacionais."- Editora Campus, 2ª Edição, (1986).
- [Ths78]- Thesen,A.- "Computer Methods in Operations Research."- Academic Press, (1978).
- [Trd85]- Tardos,E.- "A Strongly Polynomial Minimum Cost Circulation Algorithm."- Combinatoria 5, N° 3 (1985).
- [Trd86]- Tardos,E.- "A Strongly Polynomial Algorithm to Solve Combinatorial Linear Programs."- Operations Research, Vol 34, N° 2 (April 1986).
- [Zdh73]- Zadeh,N.-"Bad Network Problems for the Simplex Method and Other Minimum Cost Flow Algorithms."- Mathematical Programming, Vol 5, (1973)