

5

Resultados Experimentais

Neste capítulo, são apresentados resultados experimentais obtidos utilizando os mecanismos propostos. Estes resultados são comparados às métricas citadas anteriormente, especificamente, a receita ótima, a receita ótima de preço único e a receita obtida pelo mecanismo VCG. Além disso, os resultados também são comparados aos que seriam obtidos pelo mecanismo proposto por Hartline et al. em (09).

5.1

Ambiente de Testes

O ambiente de teste utilizado consistiu em um computador com processador Pentium IV HT de 3,06GHz equipado com 2GB de memória. Todo o programa de testes foi implementado utilizando a linguagem C de programação e foi compilado utilizando o Microsoft Visual Studio .NET. O sistema operacional utilizado foi o Windows XP Professional Edition e apenas os processos essenciais estavam sendo executados simultaneamente aos testes.

5.2

Instâncias de Testes

As instâncias de testes foram criadas com base no que foi proposto por Shoham et al. em (13). Neste trabalho, os autores sugerem um método para gerar lances para leilões de múltiplos bens, que precisou ser adaptado para o Leilão de Demanda Unitária.

Nas instâncias criadas, cada consumidor está potencialmente interessado em todos os itens e pode haver um número arbitrário de diferentes classes de itens. Definimos uma classe de item como sendo um conjunto de parâmetros que são utilizados para gerar os valores dos lances oferecidos pelos itens pertencentes à classe em questão.

As instâncias foram geradas da seguinte maneira:

Geração de Instâncias de Testes

- Decida o preço médio ($avgprice_{base}^c$), a variação do preço médio ($avgprice_{var}^c$) e a variação do preço ($price_{var}^c$) para cada classe de item c ;
- Decida o número de consumidores participantes do leilão e também o número de itens que pertencem a cada classe;
- Para cada item i da classe c , seu preço médio ($avgprice_i$) deve ser escolhido sorteando, com probabilidade uniforme, um número real pertencente ao intervalo $[(avgprice_{base}^c - avgprice_{var}^c), (avgprice_{base}^c + avgprice_{var}^c)]$;
- Para criar um lance, escolha um consumidor j e um item i :
 - Sendo c a classe a qual o item i pertence, o valor do lance deve ser $r \times avgprice_i$, onde r é escolhido com probabilidade uniforme do intervalo $[1 - price_{var}^c, 1 + price_{var}^c]$.
- O último passo deve ser repetido tantas vezes quanto o número de lances que se deseja criar.

Exemplo 6 *Seja $avgprice_{base}^c = 100$, $avgprice_{var}^c = 20$ e $price_{var}^c = 10\%$. Considere um item i que pertence à classe c . Seu preço médio $avgprice_i$ deve ser um valor entre 80 and 120. Digamos, então, que seja 90. Qualquer lance oferecido pelo item i , portanto, deve valer pelo menos $0.9 \times 90 = 81$ e nunca mais que $1.1 \times 90 = 99$.*

5.2.1**Resumo das Instâncias Criadas**

Na Tabela 5.1, pode-se perceber os parâmetros das diferentes classes de itens criadas e, na Tabela 5.2, são mostradas as características das meta-instâncias utilizadas como base para criar as instâncias com as quais testamos os mecanismos. Para cada meta-instância foram criadas dez instâncias de testes, chamadas de A1, A2, ..., A10, B1, B2, ..., B10, e assim por diante.

A razão pela qual optou-se por este método de geração das instâncias de teste, é devido ao fato que ele permite simularmos mercados com uma grande variedade de produtos substitutos, que são, de fato, os alvos de leilões

Classe	$avgprice_{base}$	$avgprice_{var}$	$price_{var}$
Inferior	50	25	50%
Padrão	90	35	40%
Superior	500	100	30%
Luxo	800	150	20%
Alto-Luxo	1000	200	50%

Tabela 5.1: Parâmetros das classes

Meta-Inst.	Cons.	Lances	Inf.	Padrão	Sup.	Luxo	Alto-Luxo
A	1000	10000	80	0	0	0	0
B	1000	10000	70	10	0	0	0
C	1000	10000	60	15	5	0	0
D	1000	10000	40	25	10	0	5
E	1000	20000	165	0	0	0	0
F	1000	20000	140	25	0	0	0
G	1000	20000	130	25	10	0	0
H	1000	20000	120	30	12	3	3
I	2000	40000	165	0	0	0	0
J	2000	40000	140	25	0	0	0
K	2000	40000	130	25	10	0	0
L	2000	40000	120	25	15	5	0

Tabela 5.2: Descrição das meta-instâncias

de demanda unitária. Resumidamente, dizemos que dois produtos distintos são substitutos quando um é utilizado em detrimento do outro, mesmo que haja uma grande diferença entre a qualidade de ambos. Exemplos de bens substitutos são diferentes marcas de pasta de dente.

O oposto de bens substitutos são bens complementares. Dois bens são ditos complementares quando eles devem ser utilizados em conjunto, porque completam um ao outro, como, por exemplo, o pé esquerdo e o direito de um calçado.

Formalmente, ver (14), sejam A e B dois bens distintos. Seja v_A o valor atribuído ao bem A , v_B o valor do bem B e v_{A+B} o valor dos itens A e B como um conjunto:

- Se A e B são substitutos, então $v_A + v_B > v_{A+B}$;
- Se A e B são complementares, então $v_A + v_B < v_{A+B}$; e
- Se $v_A + v_B = v_{A+B}$, então diz-se que não há relação entre os bens A e B , que eles são independentes entre si.

Podemos utilizar o método proposto para modelar uma grande faixa de mercados de bens substitutos, desde o mercado de pasta de dentes, já citado

anteriormente e onde não se espera uma grande variação de preços, até o mercado de bens imobiliários, onde a variação de preços pode ser enorme.

5.3

Emparelhamento Ótimo de Tamanho Fixo

Nesta seção, apresentamos alguns métodos para encontrar o emparelhamento ótimo de tamanho fixo, cujo cálculo é necessário quando precisamos limitar o número de bens vendidos. Foram implementados dois algoritmos baseados em redes de fluxo, especificamente o algoritmo de caminhos mais curtos sucessivos e o de eliminação de ciclos negativos, já mencionados no Capítulo 3. Também foi utilizada a implementação de Volgenant e Jonker do método húngaro para calcular um emparelhamento ótimo. Esta implementação está descrita em (12).

5.3.1

Implementações Baseadas em Redes de Fluxo

Para utilizarmos as implementações em redes de fluxo já descritas anteriormente, precisamos apenas limitar a quantidade de fluxo que o nó fonte é capaz de injetar na rede. Fazendo com que esta quantidade seja igual ao tamanho do emparelhamento e, por consequência, igual a quantidade de itens que devem ser vendidos, os algoritmos irão encontrar o emparelhamento ótimo de tamanho fixo.

5.3.2

Usando o Método Húngaro

O método húngaro é um algoritmo primal-dual bastante conhecido e é utilizado para resolver problemas de escalonamento, que consistem em encontrar um emparelhamento perfeito de peso mínimo para um grafo bipartido completo $G \in K_{m,m}$.

No método húngaro, é criada uma variável dual u_i para cada vértice i pertencente ao primeiro grupo de vértices e, semelhantemente, uma variável dual v_j para cada vértice j pertencente ao segundo grupo. O peso da aresta que conecta um vértice i do primeiro grupo a um vértice j do segundo é representado por $w_{i,j}$. O sub-grafo de igualdade G_f contém todos os vértices de G e apenas as arestas (i, j) tal que $u_i + v_j = w_{i,j}$. Uma atribuição de valores às variáveis duais u e v é válida se, para todo par (i, j) , temos que $u_i + v_j \leq w_{i,j}$.

Teorema 5.1 *Seja uma atribuição válida de valores às variáveis duais $\{u, v\}$. Se o sub-grafo de igualdade G_f contém um emparelhamento perfeito*

M , então M é um emparelhamento perfeito de peso mínimo para o grafo G .

Prova. Como G_f contém exatamente os mesmos vértices que G , então qualquer emparelhamento perfeito de G_f é também um emparelhamento perfeito para G . Como cada vértice de G é coberto uma e somente uma vez por M , e G_f é o sub-grafo de igualdade, temos que $w(M) = \sum_{(i,j) \in M} w_{i,j} = \sum_i u_i + \sum_j v_j$. Seja N qualquer emparelhamento perfeito para G , então $w(N) = \sum_{(i,j) \in N} w_{i,j} \geq \sum_i u_i + \sum_j v_j$ ■

Logo, encontrar o emparelhamento perfeito de peso mínimo para G é equivalente a maximizar o tamanho do emparelhamento máximo do sub-grafo de igualdade G_f , sujeito a $u_i + v_j \leq w_{i,j}$, para todo par (i, j) . A seguir, demonstramos como podemos utilizar o mecanismo húngaro para implementar as variações do mecanismo VCG apresentadas neste trabalho.

Seja n o número de consumidores e k a quantidade de bens e assumamos $n > k$. Para podermos utilizar o método húngaro, é necessário modelarmos nosso leilão como uma matriz quadrada, representando um grafo bipartido completo $K_{n,n}$. Uma saída é criarmos $(n - k)$ itens “artificiais” de modo a igualarmos o número de bens e de consumidores do leilão. Na nossa matriz, cada linha representa um consumidor e cada coluna representa um bem, sendo que as k primeiras colunas representam os bens “verdadeiros”.

Seja h o valor do lance mais alto do leilão. Preenchemos a matriz da seguinte forma:

1. Para entradas (i, j) que correspondam a um lance $l_{i,j}$ “real”, isto é $j \leq k$ e $i \leq n$, seja $w_{i,j} = 2 \times h - l_{i,j}$; e
2. Todas as demais entradas devem ser preenchidas com zeros.

Para facilitar a discussão, sempre que estivermos nos referindo a alocação considerando os valores da matriz passada ao método húngaro, utilizaremos a palavra “escalonamento”. Quando estivermos nos referindo aos valores reais dos lances, utilizaremos a palavra “emparelhamento”.

A subtração no primeiro passo é importante pois o método retorna o escalonamento de custo mínimo e estamos interessados em maximizar o custo do emparelhamento. Utilizando tais valores na matriz e sempre lembrando que o método húngaro busca minimizar o custo do escalonamento, claramente atribuiremos itens “reais” a k consumidores, sendo que os demais $n - k$ consumidores serão alocados a itens “artificiais”. É fácil perceber que a alocação dos itens “reais” retornada pelo método corresponde ao emparelhamento de custo máximo de tamanho k : o custo total do escalonamento será

de $(n - k).0$, relativo aos itens artificiais, somado ao custo do escalonamento que envolve os itens “reais”. Então, para minimizar o custo total do escalonamento, o método húngaro precisa apenas minimizar o custo da alocação dos itens reais, que, devido a subtração no primeiro passo, é equivalente a maximizar o custo do emparelhamento destes itens.

Agora, precisamos encontrar uma maneira de modificarmos nossa matriz de forma que possamos encontrar o emparelhamento ótimo de tamanho $f \leq k$. Conseguimos isso criando $(m = k - f)$ itens e consumidores “artificiais”, o que nos deixa com uma matriz de tamanho $(n + k - f \leq 2n)$ onde existem $(n - k + m = n - f)$ bens “artificiais” e m consumidores “artificiais”. A matriz deve ser preenchida de acordo com as mesmas regras estabelecidas anteriormente. Por construção, toda entrada na matriz relativa a um lance real será estritamente maior que zero, logo, quando o método buscar minimizar o custo total do escalonamento, todo bem “artificial” será alocado a um consumidor “real” e, de forma equivalente, todo consumidor “artificial” será alocado a um bem “real”. O custo de todas estas atribuições é zero. Existem ainda $(n + m - (n - f) - m = f)$ alocações que precisam ser feitas pelo método húngaro, e estas obrigatoriamente devem envolver apenas bens e consumidores “reais”. Portanto, minimizar o custo total do escalonamento fica reduzido a encontrar uma alocação de custo mínimo de tamanho f envolvendo apenas bens e consumidores “reais”. Finalmente, minimizar o custo desta alocação no grafo passado ao método húngaro é equivalente a encontrar o emparelhamento de custo máximo e tamanho f no grafo original.

5.3.3

Comparando os Algoritmos

Foram testados os diferentes métodos utilizando todas as instâncias disponíveis. Nesses testes, sempre buscamos encontrar o emparelhamento de custo máximo vendendo todos os itens. Na Tabela 5.3, mostramos o tempo médio que cada um dos algoritmos utilizados demorou para calcular o emparelhamento de custo máximo e tamanho igual ao número de bens. Como pode ser percebido, o método húngaro parece ser a melhor opção.

Inst.	Cons.	Bens	Met. Hung.	CMC	Ciclos Neg.
A-D	1000	80	51.13	634.28	1412.05
E-H	1000	165	65.20	2460.58	6180.80
I-L	2000	165	236.75	6333.53	13157.15

Tabela 5.3: Tempo em ms para calcular emparelhamentos de custo máximo utilizando o método húngaro, o algoritmo de caminhos mais curtos sucessivos e o algoritmo de eliminação de ciclos negativos

5.4

Calculando o Mecanismo VCG

O mecanismo VCG, ou uma de suas variações é uma parte fundamental dos mecanismos que desejamos avaliar. Como descrito no Capítulo 2, sua alocação é dada pelo emparelhamento de custo máximo M^* . Para calcular o preço que deve ser pago por cada consumidor c que pertence a M^* , precisamos calcular M_{-c}^* , que é o emparelhamento de custo máximo tal que nenhum item é alocado ao consumidor c e $|M_{-c}^*| \leq |M^*|$, isto é, seu tamanho deve ser menor ou igual ao tamanho de M^* .

Utilizando um algoritmo comum de emparelhamento de custo máximo com complexidade $O(n^3)$, podemos implementar o mecanismo VCG em $O(n^4)$. Deve-se imaginar, porém, ser possível melhorar esta complexidade, dado que é provável que possamos re-utilizar alguns dos cálculos feitos para encontrarmos M^* . Felizmente, isto é verdade e, utilizando uma modificação do método húngaro muito semelhante à que foi proposta por O’Sullivan e Holland em (11), podemos projetar um algoritmo capaz de implementar o mecanismo VCG em $O(n^3)$.

5.4.1

Modificação do Método Húngaro

Devemos calcular a alocação do mecanismo VCG utilizando o método húngaro conforme descrito anteriormente. Como foi mencionado, o método húngaro é um algoritmo primal-dual, logo, a solução primal ótima encontrada é derivada de uma solução dual correspondente. A idéia é utilizarmos esta solução dual para acelerarmos o cálculo dos preços dos bens.

Seja (u^{opt}, v^{opt}) a solução dual que corresponde à solução primal ótima. Executando os passos a seguir, calculamos o preço de venda do bem alocado a um consumidor c qualquer:

- Inicialmente, devemos perceber que remover o consumidor c do leilão é equivalente a alocá-lo a um bem “artificial”;

- Uma solução dual é considerada viável desde que $u_i + v_j \leq w_{i,j}$, para todo i e j ;
- Se aumentarmos os valores de $w_{c,j}$ para todo bem “real” j :
 - A solução (u^{opt}, v^{opt}) continuará viável, embora deixe de ser ótima;
 - A cardinalidade do maior emparelhamento que pode ser obtido utilizando apenas as arestas

$$(i, j) | u_i^{opt} + v_j^{opt} = w_{i,j}$$

é claramente maior ou igual a $s - 1$, onde s é o número de linhas ou colunas da matriz; e

- Se os novos valores de $w_{c,j}$ forem suficientemente grandes, digamos duas vezes o valor de qualquer outra entrada da matriz, qualquer solução ótima produzida pelo método húngaro irá alocar o consumidor c a um dos itens “artificiais”.
- Execute o método húngaro utilizando (u^{opt}, v^{opt}) como a solução dual inicial após realizadas as modificações descritas acima na matriz.

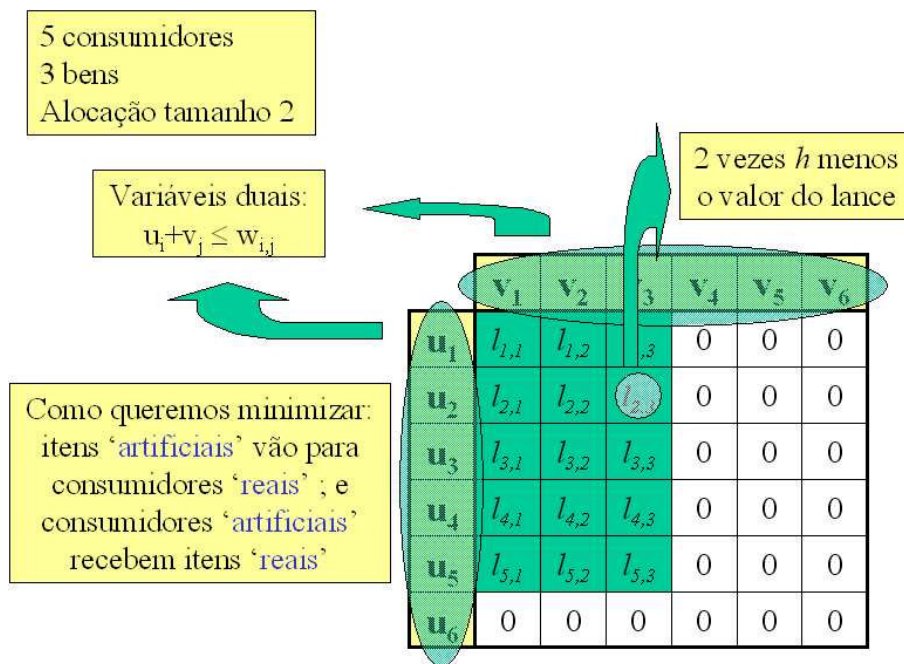


Figura 5.1: Método húngaro Modificado – Passo 1

Proposição 5.2 A complexidade computacional de calcular os preços VCG utilizando a versão modificada do método húngaro é $O(n^3)$, onde n é o número de consumidores.

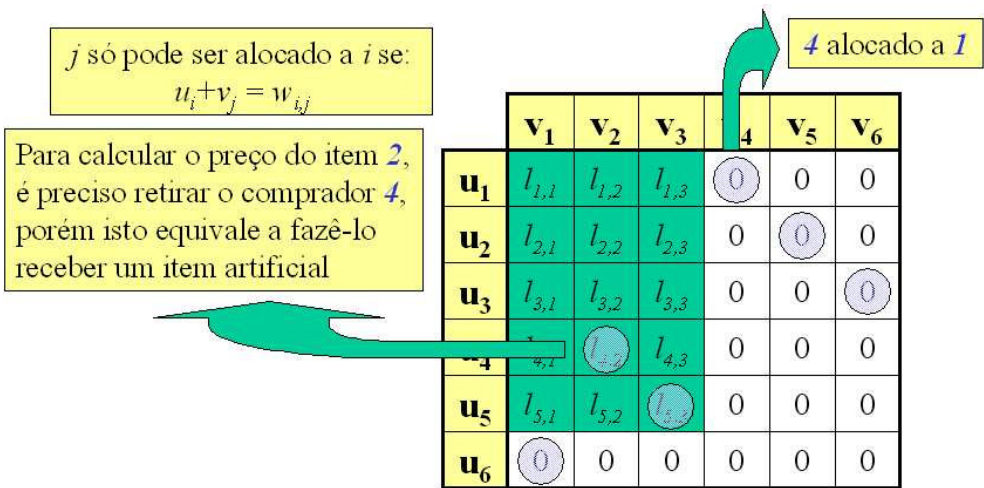


Figura 5.2: Método húngaro Modificado – Passo 2

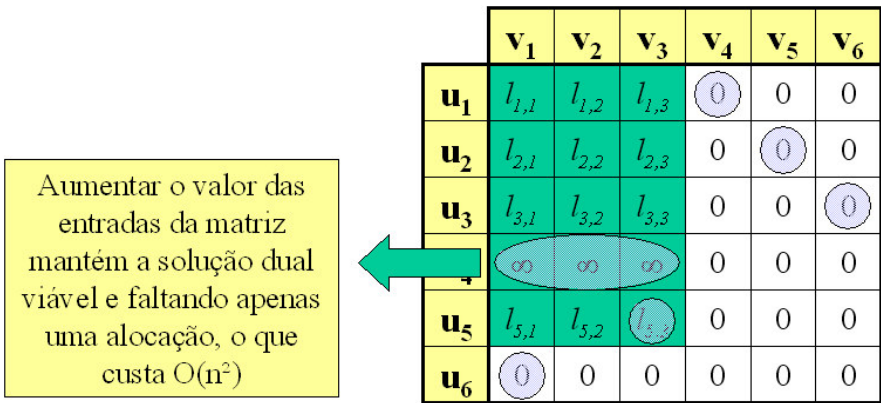


Figura 5.3: Método húngaro Modificado – Passo 3

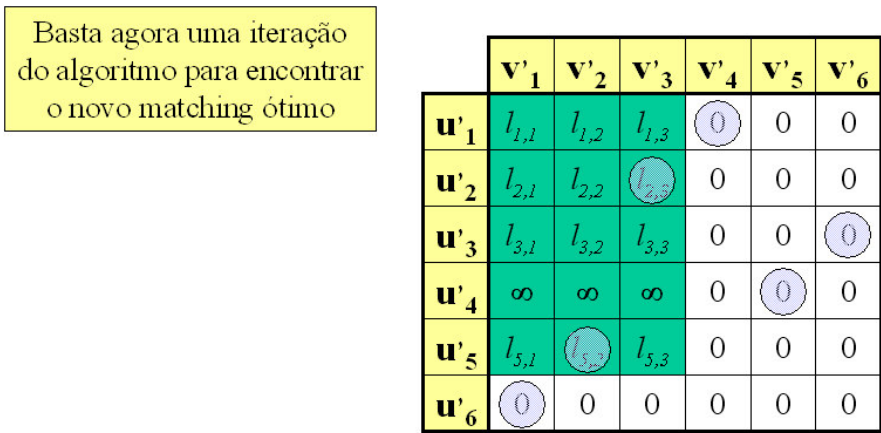


Figura 5.4: Método húngaro Modificado – Passo 4

Prova. Primeiramente, devemos notar que modelamos o nosso problema utilizando uma matriz com um número de entradas s^2 nunca maior que $(2n)^2$, onde n é o número de consumidores. Cada iteração do algoritmo de Volgenant e Jonker para calcular o emparelhamento de custo mínimo tem complexidade $O(s^2)$ e aumenta a cardinalidade do emparelhamento sendo formado em pelo menos uma unidade. Quando a cardinalidade deste emparelhamento é igual a dimensão da matriz, o algoritmo termina. Para calcular o emparelhamento ótimo de tamanho fixo, portanto, precisamos de tempo $O(s^3)$, já que é necessário calcular um emparelhamento de cardinalidade n .

Para calcular o preço que deve ser pago por um consumidor, o algoritmo demora $O(s^2)$, pois, utilizando o método descrito anteriormente, basta aumentar a cardinalidade da alocação em uma unidade. Logo, para calcular os preços a serem pagos por todos os n consumidores, o algoritmo demora $O(n.s^2)$. O tempo levado para executar o mecanismo VCG é, portanto, $O(s^3 + s.n^2) = O(n^3)$. ■

Felizmente, esta melhoria na complexidade computacional teórica reflete-se exatamente como esperado em nossos experimentos. Na Tabela 5.4, mostramos o tempo médio utilizado para executarmos o mecanismo VCG vendendo todos os itens, utilizando o grupo de instâncias A a D, E a H e I a L. Os valores da coluna “ganho” da tabela são iguais a divisão do tempo de execução utilizando a implementação padrão do método húngaro pelo tempo de execução da implementação utilizando a versão modificada.

Inst.	Cons.	Bens	Húng. Pad.	Húng. Mod.	Ganho
A-D	1000	80	2610.50	91.93	28.40
E-H	1000	165	7375.88	193.43	38.13
I-L	2000	165	33661.83	636.30	52.90

Tabela 5.4: Tempo em ms para calcular os preços do mecanismo VCG utilizando o método húngaro padrão e o modificado

5.4.2 Comparação

É apresentada na Tabela 5.5 uma comparação entre a versão modificada do método húngaro e ambos os mecanismos de redes de fluxo, sendo que os dois últimos utilizam o algoritmo de caminhos mais curtos sucessivos para calcular o emparelhamento ótimo inicial. Como foi mencionado anteriormente, o algoritmo de eliminação de ciclos negativos tem um desempenho

melhor que o algoritmo de caminhos mais curtos sucessivos quando o emparelhamento inicial é quase-ótimo. Os resultados apresentados justificam a adoção do método húngaro modificado como o principal algoritmo para a execução do mecanismo VCG ou uma de suas variações.

Inst.	Cons.	Bens	Húng. Mod.	CMC	Ciclos Neg.
A1	1000	80	140	46594	5516
B1	1000	80	110	50359	5406
C1	1000	80	109	52234	7375
D1	1000	80	125	49500	5563
A-D	1000	80	121.00	49671.75	5965.00
E1	1000	165	297	401953	23828
F1	1000	165	234	393532	24406
G1	1000	165	219	382171	23360
H1	1000	165	218	381844	23110
E-H	1000	165	242.00	389875.00	23676.00

Tabela 5.5: Tempo em ms para calcular os preços do mecanismo VCG utilizando o método húngaro modificado e ambos algoritmos de redes de fluxo

5.5

Calculando o Emparelhamento de Custo Máximo com Preço Único

O emparelhamento ótimo de preço único é a alocação de bens a consumidores que provê a maior receita quando temos que decidir um preço de venda p e, para cada item, ou vendemos o item por este preço ou não o vendemos. Podemos dizer que a receita de preço único de uma alocação bem-consumidor é igual ao tamanho da alocação vezes o valor do menor lance de um par bem-consumidor pertencente a alocação, ou seja, modelando o leilão como um grafo bipartido e a alocação como um emparelhamento, o preço único é igual ao peso da aresta de menor peso do emparelhamento e a receita de preço único é igual ao peso desta aresta vezes a cardinalidade do emparelhamento.

Sejam n e k o número de consumidores e de itens, respectivamente. O emparelhamento ótimo de preço único pode ser facilmente calculado em tempo polinomial da seguinte maneira:

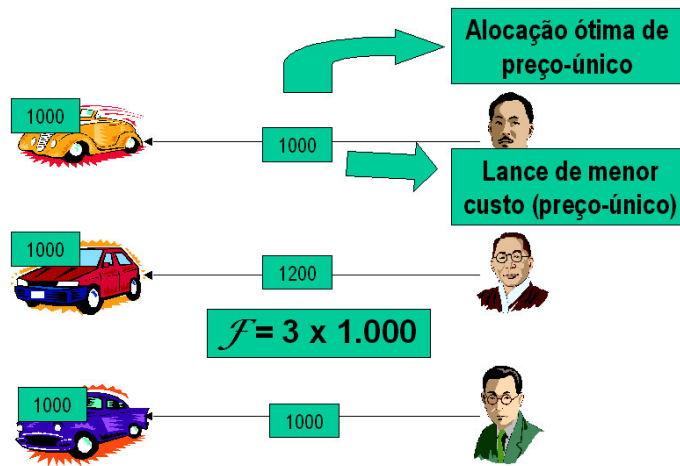


Figura 5.5: Emparelhamento ótimo de preço único

Emparelhamento Ótimo de Preço Único

1. Arrume as arestas do grafo G em ordem não-crescente;
2. Crie um novo grafo G' sem arestas e consistindo dos mesmos nós que G ;
3. Seja $M_{opt} = \emptyset$ e $rev = 0$;
4. Adicione a G' a aresta e de maior custo de G ainda não pertencente a G' ;
5. Calcule o emparelhamento de cardinalidade máxima M' de G' usando, por exemplo, o algoritmo de Edmonds-Karp e, exceto na primeira iteração, partindo do último M' calculado;
6. Se $|M'| \times custo(e) > rev$ então seja $M_{opt} = M'$ e $rev = |M'| \times custo(e)$;
7. Repita os passos 4-6 até que a cardinalidade de M' seja $\min\{n, k\}$.

5.5.1

Melhorias no Desempenho

Apesar do algoritmo proposto ser polinomial, ele é razoavelmente lento, principalmente pelo número de vezes que se tenta aumentar, sem sucesso, a cardinalidade de M' . Felizmente, ele pode ser acelerado realizando apenas

algumas pequenas modificações que evitam que em todo passo seja tentado aumentar a cardinalidade de M' . Apesar destas modificações não alterarem a complexidade computacional do algoritmo, elas representam, como veremos na comparação mais adiante, um grande ganho de desempenho.

A primeira mudança que podemos fazer é bastante evidente e se refere a condição de parada do passo 7. Ao invés de repetir os passos 4-6 até que a cardinalidade de M' seja $\min\{n, k\}$, podemos parar assim que $\min\{n, k\} \times \text{custo}(e) \leq \text{rev}$. De fato, consideramos que essa modificação está presente tanto na versão original quanto na modificada ao realizarmos a comparação de desempenho entre as duas.

A outra mudança que pode ser feita é no passo 4. Ao invés de adicionar uma única aresta em cada iteração, em alguns casos podemos determinar que, com certeza, é necessário adicionarmos um maior número de arestas para que possamos achar um emparelhamento com receita de preço único maior que rev . Isto é alcançado calculando limites superiores para o tamanho de M' após a adição de algumas arestas a G' . Seja M'_{old} o emparelhamento de cardinalidade máxima calculado para G' . Então, um limite superior para a cardinalidade de M' após a adição de m arestas a G' é o mínimo dentre:

- $|M'_{old}| + m$;
- o número de bens distintos em G' tocados por pelo menos uma aresta;
e
- o número de consumidores distintos em G' tocados por pelo menos uma aresta.

Se o limite superior para $|M'|$ multiplicado pelo custo da última aresta inserida for menor que rev , então uma nova aresta deve ser adicionada a G' . Este procedimento deve ser repetido enquanto pudermos garantir não ser possível encontrar um emparelhamento em G' melhor que M_{opt} ou até que possamos afirmar que M_{opt} é, de fato, o emparelhamento ótimo de preço único. No primeiro caso devemos proceder com o algoritmo executando o passo 5 e, no último caso, devemos interromper a execução do algoritmo.

O ganho de desempenho devido ao cálculo do limite superior pode ser percebido na Tabela 5.6, aonde comparamos o tempo levado para calcular o emparelhamento ótimo de preço único para algumas instâncias dos grupos A-H usando a implementação padrão e a melhorada do nosso algoritmo. Os valores da coluna “ganho” na tabela foram calculados dividindo a duração da implementação padrão pela duração da implementação melhorada.

Inst.	Cons.	Bens	Impl. Pad.	Impl. Melh.	Ganho
A1	1000	80	29437	688	42.8
B1	1000	80	28235	641	44.1
C1	1000	80	28688	500	57.4
D1	1000	80	4641	172	27.0
E-H	1000	80	22750.25	500.25	45.5
E1	1000	165	148188	2781	53.3
F1	1000	165	146094	2656	55.0
G1	1000	165	144719	2594	55.8
H1	1000	165	142531	1453	98.1
A-D	1000	165	145383.00	2371.00	61.3

Tabela 5.6: Tempo em ms para calcular alocações ótimas de preço único utilizando a implementação padrão e a melhorada

5.6

Resultados dos Testes

Nesta seção, são apresentados os resultados dos testes dos mecanismos de leilão utilizando as instâncias descritas na Seção 5.2.

5.6.1

Metodologia Utilizada

Os resultados dos testes referentes a uma meta-instância qualquer refletem a média dos resultados para as 10 instâncias que pertencem à meta-instância em questão. Quando o teste se refere a um mecanismo aleatorizado, o resultado de cada instância é, na verdade, a média de 10 iterações do mecanismo utilizando a instância. Nesse último caso, o valor do desvio padrão é utilizado para nos fornecer uma idéia de quanto varia a receita nas várias iterações realizadas.

Logo, quando é testado um algoritmo determinístico, o resultado para uma meta-instância M qualquer é a média dos resultados obtidos executando uma iteração do mecanismo para cada instância pertencente a M . Quando testamos um mecanismo aleatorizado, por sua vez, o resultado para uma meta-instância M é a média dos resultados obtidos executando dez iterações do mecanismo para cada instância de M .

Os mecanismos testados foram:

- **VCG**: O mecanismo VCG vendendo todos os itens;
- **PU**: O mecanismo ótimo de preço único;
- **FR**: Nosso mecanismo orientado à receita 4.4.1;
- **FE**: Nosso mecanismo orientado à eficiência 4.4.2;

- **FR2**: Nosso mecanismo orientado à receita e multiplicando por 6 a estimativa dos número de itens que devem ser vendidos 4.4.1;
- **FE2**: Nosso mecanismo orientado à receita e multiplicando por 12 a estimativa dos número de itens que devem ser vendidos 4.4.2;
- **Hart**: O mecanismo proposto por Hartline et al. em (09) e descrito na Seção 2.3.3.

Os dois primeiros são mecanismos determinísticos e os demais são aleatorizados. Deve-se observar que o desvio padrão é medido como um percentual em relação à receita média. Não haverá indicação da eficiência obtida pelo mecanismo VCG, pois esta é sempre igual a 100%.

5.6.2

Resultados Utilizando as Versões Originais dos Mecanismos

Os resultados obtidos utilizando as versões não-modificadas dos nossos mecanismos, isto é, sem considerar os mecanismos FR2 e FE2, são mostrados nas Tabelas 5.7 e 5.8.

Inst.	VCG	PU	FR	FR DP	FE	FE DP	Hart	Hart DP
A	99,5%	58,5%	21,2%	4,82%	10,9%	0,33%	67,6%	4,91%
B	99,5%	56,5%	23,6%	1,81%	12,1%	0,22%	72,9%	3,75%
C	99,6%	42,9%	42,8%	0,46%	34,0%	0,23%	72,2%	26,59%
D	99,5%	44,4%	62,5%	0,76%	38,5%	0,47%	86,4%	12,77%
E	99,4%	58,3%	21,8%	2,28%	11,2%	0,24%	72,6%	3,43%
F	99,5%	56,7%	23,6%	0,15%	12,6%	0,18%	73,5%	3,65%
G	99,5%	42,6%	43,0%	0,71%	34,2%	0,15%	71,2%	28,21%
H	99,6%	36,6%	51,8%	0,14%	40,2%	0,17%	72,1%	26,08%
I	99,3%	72,0%	17,2%	1,36%	15,2%	0,17%	96,1%	4,16%
J	99,4%	69,9%	19,0%	1,40%	16,0%	0,08%	95,7%	3,28%
K	99,7%	47,3%	43,6%	0,36%	34,4%	1,00%	82,5%	29,96%
L	99,8%	43,9%	54,2%	0,40%	31,2%	0,51%	84,2%	20,82%

Tabela 5.7: Receita utilizando as versões originais dos nossos mecanismos

Podemos perceber nos resultados experimentais obtidos que, assim como era esperado, os mecanismos propostos neste trabalho apresentam um desvio padrão muito menor que o daquele proposto por Hartline et al em (09). Entretanto, está bastante claro que, apesar do desempenho com alta regularidade dos nossos mecanismos, no sentido que eles sempre alcançam uma proporção semelhante da receita e da eficiência máximas, seu desempenho em relação a esses dois parâmetros ainda assim é fraco. Isto se deve, entre outros, ao fato de que decidimos vender um número muito

Meta-Inst.	PU	FR	FE	Hart
A	72,3%	21,4%	11,0%	68,4%
B	72,8%	23,8%	12,2%	73,3%
C	75,8%	43,1%	34,2%	72,7%
D	57,4%	63,2%	38,9%	86,8%
E	71,0%	22,0%	11,3%	73,2%
F	70,8%	23,8%	12,7%	74,2%
G	74,4%	43,2%	34,3%	71,7%
H	69,2%	52,1%	40,4%	72,5%
I	81,6%	17,2%	15,3%	96,7%
J	82,2%	19,1%	16,1%	96,2%
K	82,6%	43,8%	34,5%	82,8%
L	53,1%	54,3%	31,3%	84,4%

Tabela 5.8: Eficiência utilizando as versões originais dos nossos mecanismos

reduzido de bens o que afeta o desempenho de forma direta. Por essa razão, foram testadas versões ligeiramente modificadas dos mecanismos projetados onde se busca elevar o número de bens que devem ser vendidos.

5.6.3

Resultados Usando as Versões Modificadas dos Mecanismos

Os resultados obtidos utilizando as versões modificadas dos nossos mecanismos, isto é, considerando os mecanismo FR2 e FE2 ao invés de FR e FE, são mostrados nas Tabelas 5.9 e 5.10.

Inst.	VCG	PU	FR2	FR2 DP	FE2	FE2 DP	Hart	Hart DP
A	99,5%	58,5%	92,1%	3,09%	92,9%	0,16%	67,6%	4,91%
B	99,5%	56,5%	95,8%	0,96%	93,4%	0,14%	72,9%	3,75%
C	99,6%	42,9%	95,9%	0,60%	95,0%	0,13%	72,2%	26,6%
D	99,5%	44,4%	97,9%	0,28%	96,7%	0,21%	86,4%	12,8%
E	99,4%	58,3%	95,2%	1,69%	95,3%	0,15%	72,6%	3,43%
F	99,5%	56,7%	95,8%	0,69%	95,5%	0,12%	73,5%	3,65%
G	99,5%	42,6%	97,9%	0,11%	96,6%	0,12%	71,2%	28,2%
H	99,6%	36,6%	98,1%	0,11%	97,1%	0,10%	72,1%	26,1%
I	99,7%	58,0%	97,6%	0,06%	96,3%	0,06%	70,7%	4,25%
J	99,7%	57,4%	97,5%	0,83%	96,5%	0,05%	75,6%	3,22%
K	99,8%	41,9%	98,7%	0,04%	97,4%	0,05%	71,0%	26,8%
L	99,8%	39,5%	99,0%	0,05%	98,0%	0,04%	78,7%	17,4%

Tabela 5.9: Receita utilizando as versões modificadas dos nossos mecanismos

A elevação do número de bens que nossos mecanismos devem vender causa uma grande melhoria no seu desempenho. É válido notar que, apa-

Meta-Inst.	PU	FR2	FE2	Hart
A	72,3%	93,1%	93,9%	68,4%
B	72,8%	96,8%	94,3%	73,3%
C	75,8%	96,7%	95,8%	72,7%
D	57,4%	99,0%	97,8%	86,8%
E	71,0%	96,4%	96,5%	73,2%
F	70,8%	96,9%	96,7%	74,2%
G	74,4%	98,9%	97,6%	71,7%
H	69,2%	99,1%	98,0%	72,5%
I	71,1%	98,2%	96,9%	71,1%
J	71,2%	98,0%	97,0%	76,0%
K	74,6%	99,1%	97,8%	71,2%
L	47,8%	99,4%	98,4%	78,9%

Tabela 5.10: Eficiência utilizando as versões modificadas dos nossos mecanismos

rentemente, nossos mecanismos não parecem ter perdido suas principais propriedades: desempenho regular e pequeno desvio padrão. Há de se notar, também, que o mecanismo VCG obteve um excelente desempenho para todas as instâncias de testes, o que era até certo ponto previsível, dado que a regra de formação das instâncias juntamente com os parâmetros utilizados tornam altamente improvável a existência de entradas anômalas.

5.6.4

Leilões com Número Reduzido de Consumidores

Sabe-se que o desempenho do mecanismo VCG tende a piorar quando se reduz o número de consumidores, sendo que, em geral, a arrecadação tem uma queda especialmente acentuada quando este número passa a ser menor ou igual ao número de bens. Para testarmos o desempenho dos mecanismos neste caso, criamos uma instância de testes com 90 consumidores e 90 bens, todos pertencentes a uma mesma classe com as seguintes características: preço médio entre 95 e 105 e variação do preço de 5%. Os resultados podem ser vistos nas Tabelas 5.11 e 5.12. Deve-se notar que foram utilizadas as versões modificadas dos nossos mecanismos.

Como era esperado, o desempenho do mecanismo VCG foi bastante ruim para estas instâncias de teste, sendo que isso afetou diretamente o desempenho dos mecanismos proposto por Hartline et al. Isto acontece porque como o número de bens é igual ao número de consumidores, cada consumidor passa a ser muito importante para a arrecadação do mecanismo ótimo, ou seja, quando excluimos um consumidor do leilão, a arrecadação do mecanismo ótimo diminui de um valor próximo ao valor do lance vencedor do consumidor excluído.

Inst.	VCG	PU	FR2	FR2 DP	FE2	FE2 DP	Hart	Hart DP
equal1	4,9%	95,1%	43,5%	6,71%	39,4%	10,52%	18,3%	56,01%
equal2	4,7%	95,3%	44,5%	6,91%	40,9%	11,70%	13,6%	63,54%
equal3	4,6%	95,4%	44,5%	5,91%	40,8%	11,79%	20,8%	58,5%
equal4	4,9%	95,1%	44,6%	5,94%	40,9%	11,85%	21,1%	57,5%
equal5	4,9%	95,1%	45,7%	6,68%	41,8%	3,75%	18,2%	53,63%
equal6	5,0%	95,0%	45,8%	6,71%	41,9%	3,77%	18,3%	53,52%
equal7	5,6%	94,4%	45,1%	7,31%	41,8%	4,13%	14,3%	56,6%
equal8	5,1%	94,9%	43,8%	6,66%	41,5%	5,57%	11,6%	70,2%
equal9	4,9%	95,1%	43,7%	6,66%	41,4%	5,54%	11,3%	71,75%
equal10	5,1%	94,8%	45,7%	5,10%	41,5%	4,85%	15,8%	67,46%

Tabela 5.11: Receita quando existe o mesmo número de bens e consumidores

Meta-Inst.	PU	FR2	FE2	Hart
equal1	99,3%	43,5%	39,4%	100,0%
equal2	95,3%	44,6%	40,9%	100,0%
equal3	95,4%	44,5%	40,8%	100,0%
equal4	95,1%	44,6%	40,9%	100,0%
equal5	95,1%	45,8%	41,9%	100,0%
equal6	95,0%	45,8%	41,9%	100,0%
equal7	94,4%	45,1%	41,8%	100,0%
equal8	94,9%	43,8%	41,6%	100,0%
equal9	95,1%	43,7%	41,4%	100,0%
equal10	94,8%	45,7%	41,5%	100,0%

Tabela 5.12: Eficiência quando existe o mesmo número de bens e consumidores

Os mecanismos propostos neste trabalho conseguem diminuir a importância de cada consumidor através da redução do número de bens que devem ser vendidos. Quando vendemos um número de bens inferior ao número de possíveis compradores, isto significa que, na prática, alguns consumidores não participam da alocação do leilão. Quando retiramos um consumidor e recalculamos a alocação ótima, o impacto desta exclusão é atenuada pelo fato de que existe um consumidor semelhante que poderá substituí-lo na alocação. Evidentemente, quanto mais homogêneo for o grupo de consumidores, melhor será o desempenho desta estratégia.