CATALINA HIGUITA SALAZAR

**SELEÇÃO DE FORNECEDORES DE SERVIÇO DE TRANSPORTE
UTILIZANDO LEILÃO COMBINATÓRIO DE COMPRAS: ADAPTAÇÃO
E APLICAÇÃO DO ALGORITMO *ITERATIVE DEEPENING SEARCH*
A (IDA*)*.**

São Paulo

2012

CATALINA HIGUITA SALAZAR

**SELEÇÃO DE FORNECEDORES DE SERVIÇO DE TRANSPORTE
UTILIZANDO LEILÃO COMBINATÓRIO DE COMPRAS: ADAPTAÇÃO
E APLICAÇÃO DO ALGORITMO *ITERATIVE DEEPENING SEARCH*
A (IDA*)*.**

**Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Mestre em Engenharia**

**Área de concentração:
Engenharia de Produção**

**Orientador: Prof. Livre – Docente
Hugo Tsugunobu Yoshida
Yoshizaki**

São Paulo

2012

DEDICATORIA

A minha família com gratidão, especialmente aos meus pais Luis Eduardo e Rosa Angélica, por apoiar minhas decisões, incentivar-me e acompanhar-me, inclusive na distância. Por haver-me ensinado que com confiança e dedicação posso vencer os desafios.

Ao Michel que sempre me deu força para seguir em frente; seu amor e compreensão foram fundamentais para culminar esta etapa da minha vida.

AGRADECIMENTOS

Ao Eduardo e a Luciana por seu acolhimento, apoio e carinho incondicional.

Ao meu orientador, Professor Hugo Yoshizaki, por sua orientação ao longo deste trabalho, por compartilhar um pouco de seu vasto conhecimento na área e por contribuir para meu crescimento intelectual e pessoal.

Ao professor Miguel Santoro por ceder parte de seu tempo e contribuir com valiosas sugestões para o aprimoramento deste trabalho.

Ao professor Marcos Brinati por sua disponibilidade para participar deste trabalho e por seus importantes aportes.

Ao professor Jorge Biazzi, por dedicar seu precioso tempo ao desenvolvimento deste trabalho, participando com sua experiência e conhecimento na área.

Aos amigos do laboratório GOL e CISLog, em especial a, Celso H., Celso S., Everton, Irineu B., Juliana, Sergio e Vanessa por me auxiliar e incentivar ao longo deste período.

Aos amigos de turma pelo esforço em conjunto ao longo do curso.

Aos professores Gilberto Montibeller, Claudio Barbieri, Dario Miyake e Afonso Fleury pelos ensinamentos recebidos.

Ao departamento de Engenharia de Produção pela oportunidade de realizar o curso de mestrado.

A agência de fomento CAPES pela concessão de bolsa para a realização desta pesquisa.

RESUMO

A seleção de fornecedores de transporte é um desafio cada vez maior. O crescimento da rede de clientes a ser coberta demanda uma alocação eficiente em termos de custo não suprida por mecanismos tradicionais de negociação. Neste âmbito, o leilão combinatório torna-se uma alternativa de negociação ao permitir capturar sinergias entre os trajetos que devem ser atendidos. Em consequência disso, diminui-se o custo de transporte do fornecedor que se reflete nos menores preços de suas propostas e finalmente no custo total de compra do serviço. Por outro lado, esta decisão envolve fatores além do custo total; a mensuração destes torna-se importante para identificar fornecedores que melhor se ajustam aos requerimentos do comprador. No entanto, é fundamental escolher um método adequado para sua avaliação porque este influencia a decisão final.

Este problema de compra de serviços de transporte é conhecido na literatura como *Winner Determination Problem (WDP)* que, devido a sua complexidade, possui uma resolução limitada. Após revisão teórica, foi observado que os estudos relacionados à área de transporte focalizavam o desenvolvimento de modelos matemáticos que fossem representativos da realidade. Alguns destes modelos abordam a utilização de múltiplos critérios atribuindo um coeficiente que pondera cada critério. Evidenciou-se a necessidade do desenvolvimento de um algoritmo alternativo que além de facilitar sinergias entre trajetos, fosse abrangente o suficiente para tratar múltiplos critérios em instâncias compatíveis com problemas reais. Logo, com o intuito de contribuir com a literatura foi adaptado um algoritmo matemático otimizante ao problema de compras de fornecedores de transporte com base no algoritmo de Sandholm (2002). Este algoritmo aplica leilão combinatório de compras, apoiando-se na teoria da análise de decisão para mensurar critérios relevantes do comprador. Inicialmente, o algoritmo minimiza o custo total do comprador designando combinações de trajetos e fornecedores; depois é modificado para o tratamento multi-critério. Os resultados obtidos foram comparados com o software comercial CPLEX.

Palavras chave: Leilão combinatório. Algoritmo otimizante. Fornecedor. Múltiplos critérios. *Winner Determination Problem (WDP)*.

ABSTRACT

Selecting suppliers is a crescent challenge for the enterprises. The extent of the client web that needs to be served demands efficient allocations, in terms of cost, that are not addressed by traditional mechanisms. In this scenario, another mechanism came to be: the combinatorial auction. In this one, suppliers can express their synergies on routes they wish to supply. This leads to lowering their transportation costs, which is reflected in lower bidding prices as well as in the total cost of service. On the other hand, the selection of a supplier involves other criteria besides cost. The definition of these is essential to define which supplier fits the needs of the buyer. That is why it is of most importance to choose the right method to evaluate these needs, as it defines the final choice.

This problem is known as Winner Determination Problem (WDP) and due to its complexity, possesses a feeble solution. After compiling what has been done about the subject, it was noticed that in the field of transport, studies are focused on mathematical models that represent reality. Some models address criteria assigning coefficients to the objective function by weighting on it. Clearly, there was a need for alternative algorithms that would, besides promoting synergies on routes, also treat multi-criteria problems close to reality. Therefore, searching for a valid contribution in the field, an adaption of an optimizing algorithm based on Sandholm (2002)'s was made. The algorithm applies combinatorial auction, supported by decision analysis for measuring relevant buyer's criteria. First, the main algorithm's objective is to minimize buyer's costs by combining routes and suppliers; then, a modified approach considers multi criteria. Results were then compared to the commercial software CPLEX.

Keywords: Combinatorial auction. Optimizing algorithm. Supplier. Multi-criteria. Winner Determination Problem (WDP)

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de uma rede formada pelas linhas que precisam ser atendidas pelas transportadoras.....	17
Figura 2 - Processo de negociação do leilão combinatório na compra de serviços de transporte	35
Figura 3 - Exemplo de uma rota formada destinos A, B e C.	59
Figura 4 - Exemplo de conversão das propostas para o formato binário	60
Figura 5 - Exemplo da árvore BMC com as possíveis soluções (caminhos).....	63
Figura 6 - Exemplo da estrutura da árvore de propostas. Cada caminho é uma oferta. O <i>Stopmask</i> indica a direção em que se deve descer pela árvore dependendo das linhas que já foram alocadas.	66
Figura 7 - Diagrama de fluxo do algoritmo mono-critério: <i>Iterative Deepening Search (IDA*)</i> com <i>Depth First Search (DFS)</i>	72
Figura 8 - Estrutura da proposta multi-critério	73
Figura 9 - Formato da função de valor	75
Figura 10 - Diagrama de fluxo do Algoritmo multi-critério: <i>Iterative Deepening Search (IDA*)</i> com <i>Depth First Search (DFS)</i>	79

LISTA DE GRÁFICOS

Gráfico 1 - Tempo médio de execução de cada instância vs. número de propostas.	88
Gráfico 2 - Função de valor do custo	96
Gráfico 3 - Função de valor das avarias.....	96
Gráfico 4 - Função de valor do <i>lead time</i>	97
Gráfico 5 - Função de valor da emissão de CO2	98
Gráfico 7 – Comparação do tempo médio de execução entre os algoritmos mono-critério e multi-critério	104
Gráfico 8 – Comparação entre o custo total ótimo e o custo total multi-critério.....	105

LISTA DE TABELAS

Tabela 1 - Exemplo da oferta das propostas comerciais.....	18
Tabela 2 - Exemplo da oferta do leilão combinatório, em forma de pacotes, e sua redução de custos.	18
Tabela 3 - Exemplo da utilização do <i>Stopmask</i> para encontrar o caminho que leva ao pacote.....	66
Tabela 4- Fator de variação do fornecedor	81
Tabela 5 - Tamanho das instâncias com 15 fornecedores	84
Tabela 6 - Resultados das instâncias executadas para validar o algoritmo	85
Tabela 7 – Resultados da execução das instâncias no algoritmo multi-critério e o GAMS/CPLEX 10.2	99
Tabela 8 – Comparação das propostas e seus valores entre o GAMS/CPLEX 10.2 e o algoritmo do cenário 1500_100_15_2.	102
Tabela 9 - Comparação das propostas e seus valores entre o GAMS/CPLEX 10.2 e o algoritmo do cenário 500_100_15_3.	102
Tabela 10 – Variação dos pesos dos critérios.....	106
Tabela 11 – Análise de sensibilidade da instância 500_100_15_1	108

LISTA DE ABREVIATURAS E SIGLAS

AFT	Alternative Focused Thinking
AHP	Analytic Hierarch Process
BMC	Busca do Mínimo Custo
BPF	Busca do Próximo Filho
CA	Combinatorial Auction
CASS	Combinatorial Auction Structured Search
CBR	Case-Based Reasoning
DEA	Data Envelopment Analysis
DFS	Depth First Search
GP	Goal Programing
IDA*	Iterative Deepening Search A*
LDS	Limited Discrepancy Search
MAVT	Multi-Attribute Value Theory
MCD	Multi-Criteria Decision Analysis
RFP	Request for Proposals
WDP	Winner Determination Problem
VFT	Value Focused Thinking
VCG	Vikrey-Clarke-Groves

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CARACTERIZAÇÃO DO PROBLEMA DE COMPRAS DE SERVIÇOS DE TRANSPORTE DE CARGA LOTAÇÃO.....	15
1.2	OBJETIVOS E CONTRIBUIÇÃO DO TRABALHO.....	19
1.3	ORGANIZAÇÃO DO TRABALHO	21
2	REVISÃO DA LITERATURA.....	22
2.1	FUNDAMENTOS DO SISTEMA DE TRANSPORTE DE CARGA	22
2.1.1	Definições fundamentais	22
2.1.2	Propostas comerciais tradicionais (Request for Proposals – RFP)	24
2.1.3	Economia de escopo em transporte de carga lotação	26
2.2	LEILÃO COMBINATÓRIO (COMBINATORIAL AUCTION – CA) E COMPRAS DE SERVIÇOS DE TRANSPORTE DE CARGA LOTAÇÃO	29
2.2.1	Fundamentos do leilão combinatório	29
2.2.2	Leilão combinatório para seleção de serviços de transporte de carga lotação	33
2.2.3	Iterações do leilão combinatório	38
2.2.4	Problema de determinação do vencedor (Winner Determination Problem – WDP) em leilão combinatório de compras	38
2.3	MÉTODOS UTILIZADOS PARA SOLUCIONAR O WDP.....	41
2.3.1	Métodos Exatos	41
2.3.2	Métodos de aproximação	45
2.4	COMPRAS DE SERVIÇOS DE TRANSPORTE COM MÚLTIPLOS CRITÉRIOS	47
2.4.1	Abordagem de custo total	47
2.4.2	Análise de decisão	49
2.4.3	Modelo matemático de múltiplos critérios para leilão combinatório de compras de serviços de transporte	53
2.5	CONCLUSÃO DA SEÇÃO 2	55
3	ALGORITMO ADAPTADO PARA COMPRAS DOS SERVIÇOS DE TRANSPORTE DE CARGA LOTAÇÃO	57

3.1	ALGORITMO PARA SOLUCIONAR O PROBLEMA WDP MONO-CRITÉRIO	57
3.1.1	Parâmetros de entrada e pressupostos	58
3.1.2	Execução do algoritmo	61
3.2	ALGORITMO PARA SOLUCIONAR O PROBLEMA WDP MULTICRITÉRIO..	73
3.2.1	Modificação das estratégias de busca	74
3.3	CONCLUSÃO DA SEÇÃO 3	80
4	VALIDAÇÃO E RESULTADOS DOS ALGORITMOS PROPOSTOS	81
4.1	ALGORITMO MONO-CRITÉRIO	82
4.1.1	Geração de instâncias	82
4.1.2	Resultados	84
4.2	ALGORITMO MULTI-CRITÉRIO	88
4.2.1	Geração de instâncias	88
4.2.2	Resultados	98
5	CONCLUSÕES E TRABALHOS FUTUROS	110
	REFERÊNCIAS	116
	ANEXO A – Código do algoritmo mono-critério para solucionar o problema de seleção de transportadoras.	122
	ANEXO B – Código do algoritmo multi-critério para solucionar o problema de seleção de transportadoras.	130

1 INTRODUÇÃO

A compra de serviços de transportes tem-se transformado numa atividade complexa. Conforme aumenta o tamanho da rede de clientes a ser coberta, cresce também o número de fornecedores necessários para atendê-la. Portanto, os interesses do comprador e fornecedores ficam mais difíceis de serem conciliados durante a negociação.

No transporte de carga lotação (*Truckload* - *TL*), mecanismos de negociação tradicionais como propostas comerciais (*Request for Proposal* - *RFP*) não têm a abrangência necessária para alocar fornecedores de maneira eficiente. Nas propostas comerciais tradicionais, designam-se as linhas (par origem-destino) aos fornecedores que oferecem menor custo. Geralmente, o frete de transporte cobrado pelo fornecedor para atender uma linha leva em conta custos adicionais derivados da incerteza de conseguir outras linhas. Esses custos adicionais crescem conforme se incrementa o tamanho da rede a ser alocada; cenário desfavorável para o comprador cujo objetivo é atender seus clientes com o menor custo possível dentro do nível de serviço requerido. (CAPLICE e SHEFFI, 2006) (LEE, ROY H. e ZHONG, 2007)

As ineficiências apresentadas pelas propostas comerciais tradicionais, traduzidas em maior custo de transporte, podem ser superadas permitindo aos fornecedores (transportadoras) submeter propostas em conjuntos de linhas (pacotes). Um pacote formado pelo próprio fornecedor possibilita sinergias entre as linhas pertencentes ao conjunto, que levam à diminuição do frete de transporte. Ou seja, o custo de servir um pacote é menor que o custo de servir cada linha separadamente. A economia gerada é conhecida no mercado de transportes como economia de escopo, a qual depende, além da interdependência entre linhas, da habilidade do fornecedor para avaliar sua estrutura de custo (SHEFFI, 2004), (CAPLICE e SHEFFI, 2006).

Com a motivação de capturar economias de escopo na compra de serviços de transporte, vários pesquisadores, Vries e Vohra (2003), Song e Regan (2004) e Caplice e Sheffi (2006), começaram a estudar o leilão combinatório como um mecanismo de negociação alternativo.

Esse tipo de transação tem sido amplamente aceito na alocação de bens e serviços, como uma técnica para o suporte e automatização das negociações. Segundo

Cramtom, Shoham e Steinberg (2006), sua principal vantagem é permitir ao ofertante (fornecedor) expressar abertamente suas preferências com relação aos itens (linhas) que deseja adquirir. Isto é particularmente importante quando existe complementaridade, ou seja, quando um conjunto de itens gera uma utilidade maior do que as somas das utilidades de itens individuais.

O leilão combinatório de compras é projetado para que o fornecedor (ofertante) elabore pacotes utilizando todas as linhas (L) que estão sendo leiloadas, portanto, podem ser submetidos até $2^L - 1$ pacotes por cada fornecedor. Depois de realizar os lances, o embarcador deve combinar todos estes pacotes até encontrar o conjunto deles que minimize o custo total de compra. Isso significa que quanto maior a quantidade de linhas, maior a complexidade do processo. Este problema é representado pelo modelo matemático de determinação do vencedor (*Winner Determination Problem – WDP*). Devido à sua complexidade de resolução, alguns casos não apresentam solução em tempo polinomial; por causa disso é conhecido na literatura como NP – difícil (LEHMANN, MULLER e SANDHOLM, 2006), (SANDHOLM, 2000); (SONG e YANG, 2004), (SONG e REGAN, 2005).

Visando melhorar o desempenho do leilão combinatório, em termos de abrangência computacional e obtenção da melhor solução, diferentes abordagens têm sido adotadas ao longo do tempo e vários algoritmos otimizantes foram desenvolvidos. A respeito da compra de serviços de transporte, as pesquisas encontradas limitam-se à expansão do modelo de otimização tradicional, buscando maior aproximação com o ambiente real de negociação (CAPLICE, 1996), (CAPLICE e SHEFFI, 2003), (SHEFFI, 2004), (SONG e REGAN, 2004), (CAPLICE e SHEFFI, 2006).

As pesquisas na área de transporte convergem em outro aspecto importante do processo de compra; a avaliação do desempenho dos fornecedores participantes. Um método intuitivo é realizar a avaliação antes de iniciar o leilão, utilizando estratégias que avaliem o desempenho com base em alguns fatores; desse modo, os fornecedores que atendam aos pré-requisitos continuam na negociação (CAPLICE e SHEFFI, 2003). Esse método de pré-seleção avalia com base nas características da rede e desconsidera características individuais das linhas; como consequência, incorre-se no custo de oportunidade de perder potenciais fornecedores. Em busca de aprimorar o processo de avaliação, outras abordagens têm sido estudadas; destacam-se a modificação do coeficiente de custos de

modelos tradicionais como *WDP* (abordagem do custo total) e a utilização de ferramentas de análise de decisão (BICHLER e KALAGNANAM, 2005).

Esta última tem sido pouco estudada pela literatura, apesar de apresentar vantagens como ponderar *trade-offs* e avaliar os critérios em sua unidade de medida natural (através das funções de valor), diferente da abordagem de custo total, que exige transformação em unidade monetária.

Em resumo, o leilão combinatório é uma estratégia eficiente de negociação; sua aplicação na compra de serviços de transporte tem-se mostrado vantajosa a respeito das propostas comerciais tradicionais. Entretanto, sua implementação é complexa e requer algoritmos de otimização eficientes; além disso, é fundamental considerar vários critérios de seleção. As abordagens existentes não conseguem atender em sua totalidade a necessidade do processo de compra, ou seja, eficiência na alocação, tempo reduzido e mensuração adequada dos critérios. Desse modo, existe a oportunidade de contribuir com novas aplicações que incorporem melhores estratégias para compra de serviços de transporte.

1.1 CARACTERIZAÇÃO DO PROBLEMA DE COMPRAS DE SERVIÇOS DE TRANSPORTE DE CARGA LOTAÇÃO

O mercado de transporte de carga lotação é composto de empresas embarcadoras e empresas transportadoras. As empresas embarcadoras são aquelas que precisam transportar sua mercadoria, tais como, fabricantes, varejistas e distribuidores. As empresas transportadoras são aquelas que possuem caminhões e instalações adequadas para fornecer o serviço de transporte da mercadoria.

A necessidade de transportar mercadorias pode ser suprida através da contratação de transportadoras para realizar este serviço. Essa contratação, geralmente, é realizada através de um processo de licitação conhecido também como leilão de compras de serviço de transporte.

Qual a decisão a ser tomada no leilão de compras?

Toma-se a decisão de quais transportadoras contratar para servir a rede ao menor custo de transporte. A decisão também pode ser com base em múltiplos critérios, nesse caso, selecionam-se as transportadoras que melhor satisfaçam esses critérios.

O que se compra no leilão de compras de serviços de transporte?

A unidade de contratação é a linha, a qual é definida por Jara-Diaz (1988) como um trajeto origem-destino que possui fluxo por período de tempo. Para esta dissertação, incorpora-se parte da definição anterior “trajeto origem-destino” e se assume que a transportadora realiza o número de viagens estipuladas pelo embarcador. Uma linha pode ser expressa, por exemplo, como o percurso entre a origem São Paulo e o destino Ribeirão Preto (ver Figura 1).

O que significa vencer uma linha?

Quando a transportadora vence a negociação de uma linha significa que ela ganha o direito de transportar a mercadoria destinada para essa linha durante determinado período de tempo. Esses direitos são cedidos mediante um contrato realizado com o embarcador (ver Seção 2.2.2.1).

Quais são as decisões que devem ser tomadas antes de iniciar o leilão?

1. O conjunto de linhas (rede) a ser leilado: um ponto de origem ou destino é um local para realizar carga e descarga da mercadoria. Na prática, pode ser uma região, cidade, terminal de carga e descarga, centro de distribuição, armazém, loja, entre outros (ZHONG, 2008). Apresentando um exemplo hipotético, na figura 1 as cidades representam os pontos de origem e destino; os trajetos que ligam esses pontos correspondem às linhas da rede.

Figura 1 - Exemplo de uma rede formada pelas linhas que precisam ser atendidas pelas transportadoras.

2. Quais transportadoras participam do leilão: essa decisão geralmente é tomada avaliando a transportadora com relação a características que o embarcador considera importantes. A pré-seleção é realizada com base na informação fornecida pela transportadora e, se a mesma trabalha ou trabalhou com o embarcador, nos históricos de desempenho do embarcador (CAPLICE e SHEFFI, 2003), (ZHON, 2008).
3. Tipo de leilão a utilizar: tradicionalmente são utilizadas as propostas comerciais (ver seção 2.1.2), porém, os custos adicionais de movimentações vazias e reposicionamentos incrementam consideravelmente o custo de transporte e, conseqüentemente reduzem a eficiência da negociação. Por exemplo, se fosse utilizado esse método para leiloar as linhas da Figura 1, ao custo de transporte da linha (São Paulo – Ribeirão Preto) é acrescentado um valor de reserva que representa o custo de procurar carga em Ribeirão, lugares próximos ou retornar vazio para São Paulo. Para exemplificar, foi decidido arbitrariamente que o valor de reserva equivale a 20% do custo de transporte.

Um método alternativo são as propostas combinatórias ou pacotes; um exemplo deste tipo de proposta pode ser (São Carlos – São Paulo – Campinas – São Paulo – Ribeirão Preto – São José), ver Figura 1. Embora não seja de ciclo

fechado, conseguem-se formar sinergias que contribuem na redução do custo de transporte.

Como é realizado o leilão de compras?

1. Transportadoras enviam propostas que incluem as linhas e o preço, estas podem ser individuais ou em pacotes, dependendo do tipo de leilão;
2. O embarcador avalia as propostas e decide quais transportadoras atendem quais linhas, buscando minimizar o custo de comprar seus serviços;
3. As transportadoras selecionadas são contratadas.

As tabelas 1 e 2 mostram exemplos da diferença entre as propostas individuais e combinatórias, considerando como único critério de avaliação o custo total:

Tabela 1 - Exemplo da oferta das propostas comerciais.

Nº	Proposta <i>RFP</i>	Custo de transporte (\$)	Valor de reserva = Custos procura e reposicionamento (\$)	Custo da Oferta (\$)
1	São Paulo – Campinas	500	100	600
2	São Paulo – Ribeirão	850	170	1020
3	São Paulo - Registro	460	92	552
4	São Carlos – São Paulo	550	110	660
5	São Carlos - Bauru	710	142	852
6	Bauru – Registro	865	173	1038
7	Ribeirão – São José	930	186	1116
Custo Total (\$)				5838

Tabela 2 - Exemplo da oferta do leilão combinatório, em forma de pacotes, e sua redução de custos.

Nº	Oferta Pacote (leilão combinatório)	Custo de transporte (\$)	Valor de reserva = Custos procura e reposicionamento (\$)	Custo da Oferta (\$)
1	São Paulo – Campinas	500	100	600
2,7,4	São Carlos – São Paulo - Ribeirão – São José	550 + 850 + 930	Rep. e Proc. (São José) =186	2516
3	São Paulo - Registro	460	92	552
5,6	São Carlos – Bauru - Registro	710 + 865	Rep. e Proc. (Registro) =173	1748
Custo Total (\$)				5416

A Tabela 2 exemplifica como o leilão combinatório consegue capturar economias reduzindo o valor de reserva (ver Tabela 1). Pode-se observar que as propostas 2,7 e 4 se convertem em uma única oferta cujo valor (\$ 2516) é menor que a somatória das propostas individuais ($\$1020 + \$1116 + \$660 = \2796), do mesmo modo as propostas 5 e 6 diminuiram seu custo de \$1890 para \$1748. As economias, geradas em ambos lograram diminuir o custo total da transportadora em \$422.

1.2 OBJETIVOS E CONTRIBUIÇÃO DO TRABALHO

Conforme exposto anteriormente, existe a necessidade de métodos que consigam capturar as características do processo de compra de serviços de transporte para fazer melhores alocações de fornecedores. O objetivo principal deste trabalho é implementar e, se necessário, adaptar um algoritmo que permita obter a melhor solução do problema em estudo. Entenda-se como melhor solução aquela que atende de forma eficiente as necessidades do embarcador. Em primeiro lugar, uma solução eficiente seria obtida através de um algoritmo otimizador, ao minimizar o custo total de compra de embarcador, neste caso, seria a solução ótima. Em segunda instância, o algoritmo será adaptado para avaliar critérios, além do custo, considerados relevantes nessa área, de modo que a melhor solução será aquela que mais se aproxime aos requerimentos do embarcador, relacionados aos critérios. O foco deste trabalho é o transporte de carga lotação por sua importância na compra de serviços logísticos. Assim mesmo é importante ressaltar que o leilão combinatório será estudado desde o ponto de vista do comprador, portanto o algoritmo desenvolvido terá como base o leilão combinatório de compras, conhecido também como leilão reverso, cujo objetivo principal é minimizar o custo do comprador, capturando economias de escopo.

Com a finalidade de alcançar os objetivos propostos, os seguintes objetivos específicos devem ser desenvolvidos:

- Realizar uma revisão da literatura que permita aprofundar os conhecimentos sobre o leilão combinatório e as abordagens que já foram utilizadas para solucionar o problema de *WDP* na área de transportes e em outros ambientes de negociação.

- Selecionar o algoritmo que melhor represente as necessidades do problema que se pretende solucionar.
- Implementar e adaptar um algoritmo que solucione otimamente e em tempo adequado o problema em estudo, a partir do algoritmo selecionado da literatura.
- Avaliar a eficiência do algoritmo em termos de tempo computacional e obtenção da solução ótima.
- Realizar uma revisão da literatura acerca das abordagens multi-critério utilizadas na área de compras de transporte e selecionar os critérios mais relevantes.
- Selecionar, a partir da teoria multi-critério, um método para avaliar os critérios.
- Adaptar o algoritmo mono-critério, de forma que se possam avaliar múltiplos critérios.
- Comparar os resultados obtidos com ambos os algoritmos e definir conclusões.

A contribuição esperada está direcionada à área de transportes e pesquisa operacional. O algoritmo implementado busca melhorar o desempenho dos modelos existentes na literatura, para resolver o problema da procura de transporte de carga lotação. Este algoritmo tem como base o leilão combinatório para alocar as linhas aos melhores fornecedores, sendo considerados como melhores fornecedores aqueles que ofertam o menor custo em caso dos algoritmos mono-critério e aqueles que mais se ajustam aos critérios definidos pelo comprador no caso do algoritmo multi-critério. Com este algoritmo, espera-se contribuir com um método alternativo para solucionar o problema, que seja uma base para continuar a expandir o conhecimento na área e superar as limitações sujeitas aos modelos matemáticos.

Espera-se também que a abordagem multi-critério seja uma contribuição à literatura que, embora esteja interessada em aproximar os modelos às negociações reais, demonstra pouca maturidade no estudo de múltiplos critérios. Devido ao prevailecimento de otimalidade para resolução dos problemas, na área de transporte, outros critérios importantes são estudados superficialmente utilizando abordagens que não exploram suas verdadeiras características; por esse motivo, métodos da área de análise de decisão são aplicados ao algoritmo.

Em síntese, espera-se que esta pesquisa contribua com a literatura e sirva como base para trabalhos futuros relacionados à seleção de fornecedores.

1.3 ORGANIZAÇÃO DO TRABALHO

Levando em conta os objetivos propostos e resultados esperados, o trabalho está estruturado da seguinte forma: na Seção 2, é realizada uma revisão da literatura com a finalidade de caracterizar a compra de serviços de transportes, explicar como o leilão combinatório pode contribuir para gerar economias de escopo, apresentar os algoritmos mais relevantes que buscam solucionar este problema e descrever como se realiza a compra desde uma perspectiva de múltiplos critérios. Na Seção 3 é realizada a descrição do algoritmo proposto para solucionar o problema utilizando abordagens mono-critério e multi-critério. Na Seção 4, ambos os algoritmos são implementados e seus resultados são analisados. Na Seção 5, são definidas as conclusões e os trabalhos futuros.

2 REVISÃO DA LITERATURA

A pesquisa bibliográfica começa na Seção 2.1, caracterizando o sistema de transporte de carga lotação, explicando como é realizado o processo de compras de serviço de transporte e como este poderia ser melhorado capturando economias de escopo. Em seguida, na Seção 2.2, a pesquisa é direcionada a descrever os fundamentos do leilão combinatório, como este contribui na compra de serviços de transporte e como tem sido seu desenvolvimento ao longo do tempo. Depois, na Seção 2.3, estudam-se as abordagens multi-critério para esse tipo de negociação. Finalmente, na Seção 2.4 são elaboradas as conclusões derivadas da revisão da literatura.

2.1 FUNDAMENTOS DO SISTEMA DE TRANSPORTE DE CARGA

Esta seção apresenta conceitos essenciais do sistema de transportes e como este pode ser beneficiado com a incorporação de propostas combinatórias às negociações de compra. A seção 2.1.1 introduz conceitos básicos do sistema de transporte de carga. A seção 2.1.2 define o processo de compra tradicional baseada em propostas comerciais. Finalmente, a seção 2.1.3 explica quais fatores favorecem a geração de economias de escopo e como podem ser aproveitados.

2.1.1 Definições fundamentais

Todo sistema de transporte tem como objetivo deslocar itens desde seu lugar de origem até seu lugar de destino; conforme esta afirmação, Caplice (1996) apresenta as seguintes definições:

Um item é a unidade básica que o contratante do serviço deseja transportar. Uma remessa ou mercadoria é uma coleção de itens com origem e destino comuns. Uma

carga é um conjunto de remessas que estão se deslocando no mesmo veículo seja durante uma parte do trajeto ou até o final do mesmo.

Durante o processo de entrega de carga as principais atividades a serem executadas são as seguintes: (1) movimentação fracionada, (2) movimentação direta (*line-haul*), (3) carga/ descarga, e (4) ordenação de itens:

- Movimentação fracionada: refere-se à viagem do veículo que contém paradas intermediárias para receber e entregar mercadoria; comumente começa e termina na mesma localização. Os fatores relacionados aos custos de movimentação local são: o número de paradas durante a viagem, o tempo de processamento em cada ponto de parada e a distância entre os pontos.
- Movimentação direta: consiste numa viagem realizada sem paradas intermediárias que começa e termina em lugares diferentes. Os fatores que influenciam os custos relacionados à movimentação direta são: a distância da viagem, o tempo de processamento na origem e no destino e o balanço da rede de clientes do transportador. Entre estes fatores, a distância tem a maior influência devido à relação direta com os custos operacionais (combustível, pneus, manutenção de equipamentos, salário dos motoristas, entre outros); por causa disso, a movimentação direta exhibe diminuição nos custos unitários em função da distância. Por outro lado, qualquer custo associado ao tempo de processamento é incluído nos custos fixos. Adicionalmente, o balanço da rede de clientes acarreta um custo, relacionado às movimentações necessárias para ligar o ponto de descarga com o novo ponto de carga; conhecido como reposicionamento (*deadhead*) ou movimentações vazias.
- Manipulação de carga: envolve a carga e descarga de mercadorias, cujo custo é influenciado pela quantidade e características físicas das mesmas.
- Ordenação de itens: trata da coleta, classificação e sequenciamento da mercadoria. Esta atividade é realizada em terminais, onde se recebe a mercadoria e se ordena de acordo com o próximo destino.

2.1.1.1 Classificação do sistema de transporte

Segundo Caplice (1996), Caplice e Sheffi (2003), o sistema de transporte pode ser classificado em consolidado e direto. Entende-se como consolidação o processo de combinar diferentes itens ou remessas a fim de compor a carga de um veículo em particular. Esse sistema utiliza uma ou varias estratégias de consolidação, a partir dela(s) são delineados outros aspectos operacionais, entre eles o *trade-off* entre custo e nível de serviço.

- Consolidação por tempo: acontece quando os itens a serem transportados são acumulados até conseguir suficiente volume para realizar a entrega. Isso não implica que todos têm o mesmo destino.
- Consolidação por veículo: significa que um único veículo realiza múltiplas paradas para coletar (entregar) mercadorias que têm o mesmo destino (origem).
- Consolidação por terminais: ocorre quando mercadorias de múltiplas origens são descarregadas numa unidade comum, classificadas e re-carregadas em novos veículos conforme seu destino.

Transportadoras que utilizam algumas dessas estratégias são denominadas de carga fracionada – Less-than-truckload (LTL); elas realizam rotas regulares que envolvem movimentação fracionada para recepção ou entrega de remessas e movimentação direta para consolidação de carga entre terminais.

Por outro lado, o sistema de transporte direto realiza entregas diretas entre origem e destino, sem paradas intermediárias para consolidação de carga. As transportadoras que realizam operações com estas características são conhecidas como transportadoras de carga lotação - Truckload (TL). Seus roteiros de operação são irregulares, ou seja, o veículo não necessariamente regressa ao lugar de origem.

O intuito desta dissertação é transporte de carga lotação.

2.1.2 Propostas comerciais tradicionais (*Request for Proposals – RFP*)

Usualmente contratos sobre serviços de transportes são negociados através do processo de propostas comerciais tradicionais e sua unidade básica de contratação

é a linha. Uma proposta comercial tradicional é composta de uma linha e seu frete de transporte (LEE, ROY H. e ZHONG, 2007). Em termos gerais, a descrição do processo é a seguinte:

Em primeiro lugar, o embarcador (comprador do serviço) estima a carga que precisa transportar, usualmente calculando as movimentações principais. Depois, transmite a informação às transportadoras junto com a lista de linhas disponíveis. Com base nessa informação, as transportadoras decidem o preço para servir cada linha que seja de seu interesse e lançam suas propostas. Em seguida, o embarcador analisa as propostas submetidas de forma separada para cada linha. Finalmente, seleciona para cada linha a proposta que estiver com menor custo e anuncia as transportadoras ganhadoras. O custo total de compra é composto pela somatória das propostas selecionadas (SHEFFI, 2004).

Pode-se observar que o tamanho da rede que o embarcador deseja atender é um fator que condiciona o desempenho da negociação, já que impacta na complexidade do problema através da quantidade de transportadoras e quantidade de propostas que precisam ser avaliadas. Outros fatores importantes são as características das linhas e fornecedores. Por exemplo, dependendo da linha pode ser necessária a utilização de várias transportadoras para evitar dependências na entrega de mercadoria; pode-se também identificar sinergias entre as linhas e a própria rota operacional dos fornecedores.

Evidentemente a complexidade intrínseca ao tamanho da rede cresce conforme os mercados são expandidos. Nesse contexto, a utilização de propostas comerciais representa um grande esforço de execução; além disso, lances baseados em estimativas levam a soluções menos eficientes (ver Seção 1.1.3). A fim de compreender a dimensões do processo de compra Caplice e Sheffi (2006) elaboraram, com base na experiência adquirida entre 1996 e 2001 projetando e administrando leilões deste tipo (ver Tabela 3). Pode-se observar como o número de linhas impacta diretamente nas dimensões da compra; por outro lado, o investimento anual é um indicador da importância de realizar processos eficientes para obter maiores economias.

Tabela 3 - Características dos leilões de compra de serviços de transporte. Caplice e Sheffi (2006).
Tabela 21.1

Características	Mínima	Mediana	Média	Máxima
Número de linhas	136	800	1800	~5000
Número de embarques anuais	~6000	88000	~200000	~1500000
Custos anuais dos serviços de transporte	\$3M	\$75M	\$175M	\$700M
Número de fornecedores incumbentes	5	100	162	700
Número de fornecedores participantes	15	75	120	470
Número de fornecedores designados do leilão	5	40	64	300
Redução da quantidade de fornecedores	17%	48%	52%	88%
Redução base em custos de transporte (sem considerar fatores de serviço)	3%	14%	13%	24%
Redução final nos custos de transporte (considerando fatores de serviço e outras restrições do negocio)	0%	6%	6%	17%
Duração do processo de compra	<1	3	3	6+

Visando melhorar o desempenho da negociação, vários pesquisadores (ver Seção 2.2.2.2) introduziram a utilização de propostas combinatórias como uma alternativa de compra de serviços de transporte (ver seção 1.2); seu desempenho é confirmado pelos casos bem sucedidos em empresas de grande porte. Ledyard et al. (2002) descrevem sua aplicação na *Sears Logistics Services*. Vries e Vohra (2003) explicam sua utilização em *Colgate-Palmolive*, *Procter and Gamble*, *Wal-Mart*, *Ford Motor Company*, *Compaq Computer Corporation*, entre outras. Elmaghraby e Keskinocak (2004) descrevem a experiência da *Home Depot*, grande empresa varejista.

2.1.3 Economia de escopo em transporte de carga lotação

O mecanismo de negociação estipula os custos a incorrer no processo de compra. Em transporte de carga lotação quando uma transportadora utiliza a estimativa como sua estratégia de negociação, o qual acontece nas propostas comerciais, o valor de suas propostas é influenciado por custos adicionais produto da incerteza implícita nos próximos trajetos a vencer. O preço estipulado para cada linha inclui um valor de reserva que é estimado levando em consideração: o custo de procurar carga no lugar de destino e o custo de reposicionamento. O primeiro acontece

quando não existe uma linha ligada a esse trajeto e se deve esperar até encontrar uma nova carga. O segundo ocorre quando é necessário deslocar o caminhão vazio para cobrir uma nova linha. Ambos os custos são classificados por Caplice e Sheffi (2003) como custos de conexão, os quais, em conjunto com os custos de movimentação compreendem os custos de transporte mais relevantes.

Pode-se observar que a incerteza dos custos de conexão se deve principalmente ao tempo de espera ocasionado pela variabilidade espacial e temporal da demanda. Particularmente, o reposicionamento de caminhões vazios compromete grande parte das despesas operacionais. Por causa disso, fornecedores preferem formar conjuntos de linhas que formem sinergias a respeito destes custos (LEE, ROY H. e ZHONG, 2007).

Embora os custos de movimentação possam ser estimados com maior precisão, eles também sofrem modificações por causa da incerteza, visto que em sua maioria dependem da distância percorrida. Por exemplo, combustível, salário dos motoristas e manutenção do veículo.

Dado que a estimação dos custos de transporte depende das linhas a serem servidas, torna-se difícil obter uma alocação eficiente leiloando linha por linha. Conforme Caplice e Sheffi (2003), numa alocação eficiente as linhas a serem leiloadas são atribuídas aos fornecedores que as valorizam mais.

Por outro lado, as propostas combinatórias permitem interdependência entre os custos das linhas, ao aceitar varias linhas dentro de uma mesma proposta, levando à diminuição do custo total. Caplice e Sheffi (2003) Consideram que os fornecedores possuem maior capacidade para elaborá-las dada sua habilidade para:

- Decidir sobre quais regiões realizar propostas de forma a manter um bom balanço entre as novas rotas e as já existentes. Isso é importante para uma melhor utilização da sua infraestrutura e controle da periodicidade para manutenção dos caminhões e descanso de seus trabalhadores;
- Selecionar as linhas a fim de garantir um mercado suficiente para prestar seus serviços sem obter prejuízos;
- Criar configurações que estejam condicionadas de forma que ele possa ganhar uma delas, mas não todas. O fornecedor é livre para incluir uma linha em várias propostas.

Levando em conta esses critérios, as propostas resultantes podem expressar complementaridade ou substituíbilidade. Estes termos, comumente utilizados em

leilão combinatório são definidos por Song e Regan (2005) como segue: seja $C(AB)$ o custo para uma transportadora servir a linha que vai do ponto A ao ponto B ; seja o custo $C(BA)$ de servir uma linha que vai de B para A e, seja $C(BCA)$ o custo de transportar mercadoria de B para C e de C para A . Pode-se dizer que as linhas são:

- Complementares, se; $C(AB) + C(BA) > C(AB \cup BA)$
- Substituíveis, se; $C(AB \cup BA) + C(AB \cup BCA) < C(AB \cup BCA \cup BA)$
- Aditivas, se; $C(AB) + C(BA) = C(AB \cup BA)$

Para ilustrar as relações, suponha que uma transportadora oferta os conjuntos $\overline{AB}$ e $\overline{BA}$, complementares entre si, dado que ao juntá-los num pacote incorre-se em menores custos de movimentações vazias. Agora, supondo que existe outro conjunto $\overline{BCA}$, então se diz que $\{\overline{AB}, \overline{BA}\}$ e $\{\overline{AB}, \overline{BCA}\}$ são substituíveis a respeito de $\overline{AB}$ já que servindo as três linhas gera-se um custo de movimentação vazia em $\overline{AB}$ no momento em que o caminhão é reposicionado para servir $\overline{BA}$. Finalmente, uma relação aditiva existe se o custo das linhas permanece invariável quando combinadas em uma oferta.

Uma vez que o interesse do fornecedor seja trabalhar com conjuntos de linhas complementares, os custos de procura são desconsiderados visto que se conhece o próximo destino. Os custos de reposicionamento podem ser calculados com maior precisão conhecendo a distância entre as linhas.

A redução de custos dos fornecedores ocasionada pela interdependência entre linhas é conhecida como economia de escopo. Cabe ressaltar que o transporte de carga lotação exibe significativas economias, quando os volumes de carga são consistentes e os fornecedores possuem habilidade para identificá-las (CAPLICE e SHEFFI, 2003).

Contudo, capturar essas economias requer a utilização de algoritmos de otimização que garantam uma avaliação e seleção eficientes. Deve-se considerar que uma linha pode formar parte de várias propostas, portanto, o comprador pode receber uma quantidade de propostas que cresce exponencialmente ao número de linhas, o qual incrementa a complexidade da seleção (ver Seção 2.1.4).

2.2 LEILÃO COMBINATÓRIO (*COMBINATORIAL AUCTION* – CA) E COMPRAS DE SERVIÇOS DE TRANSPORTE DE CARGA LOTAÇÃO

Uma vez compreendidas as vantagens que pode trazer às negociações a utilização de propostas combinatórias, esta Seção apresenta sua descrição formal. A seção 2.2.1 caracteriza o leilão combinatório dentro do ambiente de negociação. Em seguida, a Seção 2.2.2 define o leilão combinatório como uma ferramenta para as compras de serviços de transporte. Depois, a Seção 2.2.3 explica as vantagens e desvantagens de realizar o leilão combinatório em uma rodada. Finalmente, a Seção 2.2.4 define o problema e o modelo matemático das compras de serviços de transporte, conhecido como problema de determinação do vencedor (*WDP*).

2.2.1 Fundamentos do leilão combinatório

Os leilões têm como propósito a alocação de bens e serviços entre múltiplos agentes através de um processamento eficiente de dados.

Dada sua extensa abrangência, torna-se fundamental escolher o tipo de leilão com base no cenário. Com esse propósito, Kalagnanam e Parkes (2004) elaboraram seis dimensões valendo-se dos componentes mais representativos dentro de um ambiente de negociação:

- Recursos: identificam-se o conjunto de bens ou serviços que serão negociados. Esses recursos podem ser: único item, múltiplos itens, ambos com uma ou múltiplas unidades de cada item. E ainda, em cenários reais, o item pode ser do tipo *commodity* ou multi-atributo.
- Estrutura do mercado: deve-se estabelecer a posição dos compradores e vendedores dentro da negociação. Em leilões tradicionais (*forward auctions*) um único vendedor está comercializando seus recursos com múltiplos compradores.

Em contrapartida, nos leilões reversos ou de compra (*reverse auctions*) um único comprador está adquirindo recursos de vários fornecedores.

- Estruturação das preferências: estipulam-se as regras/preferências relacionadas à utilidade dos agentes envolvidos.
- Estrutura da proposta: define-se a flexibilidade para expressar propostas. Uma oferta é composta por quantidade e preço. As formas mais comuns são: propostas de itens divisíveis, indivisíveis, em conjuntos, em configurações (multi-atributos). Quanto ao preço, ele pode ser fixo ou variável em função da quantidade.
- Adequação da oferta à demanda: escolhe-se a quantidade de vendedores ou compradores que vão participar do leilão. Esta seleção influencia na complexidade do problema, o qual varia entre classificação simples até NP - difíceis.
- Feedback: finalmente, procura-se saber se o mecanismo de leilão é direto ou indireto. Se for direto, o lance se realiza sem receber informação do estado do leilão. Se for indireto, tal como leilões ascendentes, os agentes recebem informação do estado do leilão, portanto, eles têm a oportunidade de ajustar suas propostas.

Essas dimensões contribuem para uma descrição detalhada de qualquer tipo de leilão. As escolhas realizadas em cada uma delas têm um impacto significativo na complexidade estrutural do problema.

2.2.1.1 Leilão combinatório

Segundo Sandholm (2000), a carência dos leilões que analisam itens de forma separada é não reconhecer que em muitos ambientes de negociação os ofertantes dão importância aos itens que ganharão em próximos lances e, por esse motivo, utilizam estratégias como a estimação numa tentativa de conseguir uma boa configuração.

O leilão combinatório busca superar as ineficiências inerentes desses leilões permitindo aos ofertantes submeter propostas em combinações de itens, chamados "pacotes" (CRAMTOM, SHOHAM e STEINBERG, 2006)

O mais famoso leilão combinatório é a generalização combinatória *Vikrey-Clarke-Groves* (VCG). De modo geral, no VCG ofertantes submetem seus pacotes; ganham os pacotes que minimizem o custo total (leilão reverso) ou maximizem a receita total do leiloeiro (leilão tradicional), mas o valor que é pago é o preço do pacote que ficou em segundo lugar. Desse modo, o leilão incentiva os ofertantes a revelar seus verdadeiros preços (AUSUBEL e MILGROM, 2006).

Classificação do leilão combinatório

Segundo Sandholm et al. (2002), o leilão combinatório pode ser classificado com base em quatro fatores que diferenciam os itens pela sua singularidade; multiplicidade e disposição, livre e restrita:

- Leilão de único item: o leiloeiro disponibiliza as unidades que deseja vender; os ofertantes realizam lances segundo as quantidades que pretendem adquirir. O leilão consiste em selecionar os lances com o maior preço, para maximizar a receita de venda do leiloeiro, considerando que cada item deve ser alocado a pelo menos um ofertante.
- Leilão de múltiplos itens: refere-se ao leilão com diferentes tipos de itens, possivelmente em várias unidades. Nesse caso o ofertante pode realizar um único lance que contenha a quantidade de unidades desejada de cada item. O leiloeiro seleciona os lances de maior preço que combinados aloquem todas as unidades negociadas (SANDHOLM, 2000).
- Leilão reverso ou leilão de compras: entende-se como leilão reverso aquele onde o comprador está interessado em adquirir itens ao mínimo custo e, um grupo de vendedores está disposto a proporcionar esses itens. Assim, cada vendedor oferta os conjuntos de itens que está disposto a prover. Esse mecanismo é utilizado na compra de serviços de transporte, o foco específico desta pesquisa, onde é necessário incorporar complementaridade entre as unidades (linhas) para obter alocações eficientes (CAPLICE e SHEFFI, 2003), (LEE, ROY H. e ZHONG, 2007).
- Intercâmbios ou leilões duplos: Esse tipo de transação pode envolver vários vendedores e compradores dependendo dos negócios determinados pelo mercado. Cada participante tem a possibilidade de realizar tanto transações de compra quanto transações de venda.

Quando não existe livre disposição de itens significa que tanto o leiloeiro, seja vendedor ou comprador, quanto os ofertantes, devem limitar suas negociações às unidades estipuladas no começo de leilão. Assim, cada unidade deve ser alocada exatamente a um comprador.

O desafio do leilão combinatório é decidir: como realizar uma alocação, dado um conjunto de propostas/lances. Esse problema, conhecido como *Winner Determination Problem (WDP)* (ver Seção 2.2.4), tem recebido bastante atenção na literatura e, dada sua complexidade (ver Seção 2.2.1.4) existe a preocupação por desenvolver algoritmos eficientes que estejam em capacidade de obter boas combinações em tempo computacional moderado. (ROTHKOPF, PEKE e HARSTAD, 1998); (SANDHOLM, 2000); (SANDHOLM et al. 2002); (DANG e JENNINGS, 2003).

2.2.1.2 Complexidade do leilão combinatório

A complexidade computacional do leilão combinatório é calculada mediante a análise do pior caso; isto significa que é calculado o máximo número de operações executado para encontrar uma solução do problema. Usualmente a complexidade é reportada em notação assintótica conhecida como *big-O*, a qual indica que o valor é um limitante superior da complexidade do problema. (HUNTER, 2009)

Quando não há combinação entre itens, uma solução do problema pode ser encontrada selecionando para cada item a proposta que possua o maior preço ou o menor custo, no caso de leilão de compra. Encontrar essa solução leva $O(TL)$, onde T significa número de ofertantes e L significa número de itens (SANDHOLM, 2002). Sua complexidade polinomial indica que no pior caso cada ofertante submeterá uma proposta para cada item, assim para encontrar o menor preço de cada, seria necessario procurar em T multiplicado por L propostas.

Em contraste, o leilão combinatório precisa avaliar todas as possíveis soluções formadas a partir do conjunto de propostas, para encontrar uma solução factível que maximize a receita de venda (leilão tradicional) ou minimize o custo de compra (leilão reverso) do leiloeiro. Em outras palavras, seja L o conjunto de itens que está à venda ou à compra, então qualquer ofertante i , pode submeter $2^L - 1$ propostas,

as quais devem ser combinadas até encontrar o conjunto que inclua todos os itens leiloados e otimize o valor total do leiloeiro (seja a receita de venda ou o custo de compra) (SANDHOLM e BOUTILIER, 2006).

Uma forma intuitiva para solucioná-lo seria enumerar todas as propostas e fazer as combinações, mas devido a seu crescimento exponencial com o número de itens, as soluções são restritas a instâncias pequenas. Dependendo da abordagem utilizada para resolver o problema e da quantidade de propostas submetidas, este pode ser não solucionável em tempo polinomial. O conceito de tempo polinomial significa que, no pior caso, o tempo de execução é da ordem $O(p)$ para algum polinômio p . Esse tipo de problema é denominado na literatura como NP- difícil (SANDHOLM, 2000); (SONG e YANG, 2004) ; (SONG e REGAN, 2005); (LEHMANN, MULLER e SANDHOLM, 2006).

Uma questão que auxilia na diminuição da complexidade é a realização de um pré-processamento das propostas, de tal forma que somente seja considerada uma proposta de cada combinação de itens submetida. Essa proposta é a de maior preço/menor custo. Dessa forma minimiza-se o número de alternativas a ser analisado. Uma análise detalhada sobre a complexidade do *WDP* é realizada por Lehmand, Muller e Sandholm (2006).

2.2.2 Leilão combinatório para seleção de serviços de transporte de carga lotação

Depois de introduzir o leilão combinatório e demonstrar sua importância e tratamento dado na literatura, esta Seção pretende definir o ambiente de negociação e, mostrar como os pesquisadores têm abordado esse mecanismo no transporte de carga de lotação (LT).

2.2.2.1 Cenário de negociação

Conforme as dimensões de Kalagnanam e Parkes (2004), o formato das compras de serviços de transporte é o seguinte:

- Recursos: as linhas que o embarcador precisa atender são as unidades que serão negociadas. Ou seja, são múltiplos itens (linhas) com uma unidade de cada (todas as linhas são diferentes).
- Estrutura do mercado: por se tratar da compra de um serviço, o tipo de leilão utilizado é o leilão reverso, conhecido também como leilão de compras. Aqui, um único embarcador está adquirindo o serviço de várias transportadoras. Desse modo, o embarcador é o leiloeiro e as transportadoras são as ofertantes.
- Estrutura de preferência: em primeira instância, o embarcador deseja minimizar seu custo total de compra e as transportadoras devem realizar lances ao menor custo possível para vencer o leilão. Em segunda instância, o leilão será projetado para avaliar múltiplos critérios de cada linha; neste formato o embarcador deve maximizar o valor total. Assim, cada proposta terá um valor, dependendo do desempenho dos critérios avaliados, entre maior o valor melhor a proposta. Portanto, as transportadoras devem intentar atender os níveis de desempenho desejados em suas propostas para se tornarem vencedoras.
- Estrutura da proposta: neste caso a linha é um item indivisível. Devido a isso, a estrutura da proposta (P) é representada da seguinte forma:

L = Conjunto de linhas totais

I_p = Conjunto de linhas da proposta; $I_p \in L$

p_p = Preço da proposta

C_p = Desempenho dos critérios da proposta; $C_p = \{c_p^1, c_p^2, \dots, c_p^n\}$

Para minimizar o custo total de compra; $P = \langle I_p, p_p \rangle$.

Para maximizar o valor total; $P = \langle I_p, c_p^1, c_p^2, \dots, c_p^n \rangle$ onde o custo é inserido como critério.

- Adequação da oferta à demanda: a quantidade de transportadoras que participam do leilão depende da quantidade de linhas que se deve leiloar.

- Feedback: o mecanismo de leilão é direto, porque uma vez que as propostas são lançadas não existe possibilidade de modificá-las e não se troca informação sobre o estado do leilão.

A Figura 2 mostra uma breve descrição do processo do leilão combinatório de compras:

Figura 2 - Processo de negociação do leilão combinatório na compra de serviços de transporte

Finalizando a negociação da Figura 2, as transportadoras selecionadas:

- Ganham o direito de atender exatamente o conjunto de linhas da(s) proposta(s) vencedora(s).
- O valor pago pelo serviço é o valor estipulado na(s) proposta(s) vencedora(s).
- Neste caso, a transportadora deve transportar toda a mercadoria destinada para as linhas que ganhou pelo período acordado. Em outras situações, quando a mercadoria é leiloada por volume e não por linha, várias transportadoras ganham o direito de atender uma mesma linha em um determinado volume.

2.2.2.2 Evolução do leilão combinatório na compra de serviços de transporte

Segundo Caplice e Sheffi (2006), o primeiro registro sobre a solução do *WDP* para um leilão de serviços de transporte possivelmente foi realizado pela *Reynolds Metals Company*, ao final de 1980. Eles desenvolveram um modelo de programação inteira

mista (MIP) que minimiza custos de transporte, no entanto as propostas submetidas foram tradicionais.

Posteriormente, Caplice (1996) analisou o efeito das economias de escopo na relação entre o fornecedor e o comprador e introduziu a utilização de propostas combinatórias.

Em seguida Ledyard et al. (2002) descreveram, provavelmente, a primeira aplicação de leilão combinatório em serviços de transporte, o que eles chamaram de *Combined –value auctions (CVA)*. O primeiro leilão da *Sears Logistics Services*, sem CVA, envolveu 854 linhas com custo de serviço de aproximadamente US\$190 milhões por ano. Depois de implementar o CVA seu custo de serviço diminuiu para US\$165 milhões por ano, o que significou 13% de economia.

Uma vez conhecidos os benefícios do leilão combinatório, aumentou-se o interesse dos pesquisadores por aprimorar esse processo de negociação. Nesse contexto, Caplice e Sheffi (2003) realizaram amplas análises e apresentaram modelos matemáticos envolvendo novas restrições, adicionadas para proporcionar um ambiente seguro de negociação e próximo à realidade. Os autores também reportaram experiência em projetar e conduzir mais de 100 leilões de compra, dos quais mais da metade foram leilões combinatórios; foram movimentados US\$8 bilhões em serviços de transporte e suas economias superaram os US\$500 milhões. Nesta época já existiam softwares comerciais que permitiam combinações, entre eles estão: *e-Mediator*, *Manugistics Inc (RFQ optimizer®²)*, *Manhattan Associates Inc.(OptiBid®³)*, *i2 Inc. (Transportation Bid Collaborator®)*, *Baan Inc. (BidPro®)*, *Saitech (SBids®)*, e *Schneider Logistics Inc. (SUMMIT CVA®)* (SANDHOLM, 2000), (VRIES e VOHRA, 2003). Muitos desses softwares têm sido aproveitados por companhias como *Colgate-Palmolive*, *Procter and Gamble*, *Wal-Mart*, *Ford Motor Company*, *Compaq Computer Corporation*, entre outras (VRIES e VOHRA, 2003).

Depois, Sheffi (2004), além de expandir as restrições de negociação mencionadas por Caplice e Sheffi (2003), explicou a variabilidade dos custos administrativos e operacionais com relação ao número de fornecedores e ao método utilizado para realizar sua avaliação.

Nesse mesmo período, Elmaghraby e Keskinocak (2004) descreveram a utilização do leilão combinatório na empresa varejista *Home Depot*. Nesse leilão, as transportadoras foram habilitadas para submeter propostas combinatórias. A

experiência foi descrita como bem sucedida já que, além de diminuir o custo total de transporte, várias transportadoras expressaram maior satisfação com sua parte do negócio.

Posteriormente, Guo et al. (2006) desenvolveram e resolveram dois modelos de otimização que incluíam restrições de cobertura e desempenho. Os resultados apresentados foram obtidos utilizando B&B e heurísticas.

Simultaneamente, Sandholm et al. (2006) detalharam como a empresa de bens de consumo *Procter & Gamble* aplicou a interface web REV, da empresa de software *CombineNet*, a suas negociações de compras, entre elas, a de compras de serviço de transporte utilizando propostas combinatórias. No total a empresa desembolsava US\$3 bilhões com esse tipo de negociação e com a utilização do REV obteve economia de 9.6%, dos quais 3.6% foram na área de transportes.

Em termos gerais as pesquisas do leilão combinatório de compras têm sido direcionadas à solução do *WDP* da a perspectiva do comprador. Entretanto, alguns autores exploram o tema pensando em gerar maiores benefícios para a contraparte (fornecedores) através de estratégias que possibilitem uma melhor configuração de propostas. Nessa linha, Song e Regan (2004) construíram um método que determina heurísticamente potenciais linhas que formariam parte das rotas de um fornecedor e resolve o modelo de Partição de Conjuntos para determinar quais rotas selecionar. Finalmente as rotas selecionadas no modelo são submetidas ao comprador. Os resultados demonstram que é vantajoso selecionar rotas utilizando essas estratégias. Mais detalhes são encontrados em um posterior artigo dos mesmos autores, Son e Regan (2005).

Mas adiante, Zhong (2008) desenvolveu um modelo de otimização para a geração das propostas das transportadoras. Apresenta também um leilão combinatório iterativo (processo executado várias vezes) que integra os problemas de otimização do embarcador e do transportador. Aqui, os preços duais derivados da solução do *WDP* são utilizados pelas transportadoras para identificar quais linhas são mais lucrativas. Finalmente, estende o problema determinístico para um problema estocástico em dois estágios sobre incerteza de volume, demonstrando que sua solução é aproximada à do modelo determinístico.

Depois, Zhong et al. (2010) expandiram o modelo estocástico de Zhong (2008) que busca restringir o risco do embarcador, produto da incerteza ocasionada pela sazonalidade da demanda. O modelo, além de conter as restrições consideradas por

2.2.3 Iterações do leilão combinatório

Outro aspecto que deve ser tratado diz respeito ao número de iterações do leilão. Dependendo das características da negociação, os leilões podem ser efetuados uma única vez ou múltiplas vezes (leilões iterativos) (JIN, WU e ERKOC, 2006). O primeiro caso exige que as propostas sejam submetidas no começo do leilão sem possibilidade de ajustes durante o processo. O segundo caso possibilita o intercâmbio de informação a fim de incentivar ajustes nas propostas, os quais são realizados ao começo de cada iteração. Esse último implica o estabelecimento de critérios de parada e regras que facilitem as transações e diminuam o tempo de execução.

Os serviços de transporte são negociados envolvendo ambos os tipos de leilão dependendo do interesse e condições do comprador. Dada a complexidade do problema, intuitivamente parece mais viável criar vários cenários considerando diferentes restrições e ajustá-los até chegar a uma alocação que seja coerente com as exigências e necessidades do comprador. Essa prática implica realizar um leilão iterativo de múltiplas rodadas, o qual exige esforço e tempo e, conseqüentemente, incrementa os custos administrativos (CAPLICE e SHEFFI, 2003); (SHEFFI, 2004); (CAPLICE e SHEFFI, 2006).

Por outro lado, ao realizar uma rodada consegue-se reduzir o custo administrativo e, o mais importante, diminuir a probabilidade de “danificar os preços” entre fornecedores, que poderia resultar em baixos níveis de serviço. Porém, isso somente pode ser feito desenvolvendo algoritmos mais completos e eficientes (em termos de tempo computacional e obtenção da solução) (CAPLICE e SHEFFI, 2003).

2.2.4 Problema de determinação do vencedor (*Winner Determination Problem* – *WDP*) em leilão combinatório de compras

Conforme mencionado na Seção 2.2.1, o leilão combinatório é discutido e analisado na literatura através do modelo de otimização combinatória para a determinação do

vencedor (*WDP*). Para o caso específico das compras de serviços de transporte de carga lotação, o leilão representado pelo *WDP* é o leilão combinatório reverso.

O problema, caracterizado na Seção 1.1, possui os seguintes pressupostos:

- Volume não especificado; assume-se que, se a transportadora decide inserir uma linha em suas propostas, está disposta a transportar todo o volume designado para essa linha.
- Assume-se que, se a transportadora submete uma proposta para determinadas linhas, possui a capacidade requerida para transportar a mercadoria de todas as linhas durante o período estipulado no contrato.
- Uma vez submetidos, os pacotes não podem ser modificados.
- O preço de cada pacote corresponde ao valor que o embarcador paga à transportadora em troca do serviço prestado para o conjunto de linhas especificadas no pacote, se o pacote for selecionado.
- Cada linha é atendida somente por uma transportadora, dado que as propostas são lançadas em função da linha como um todo e não em função do seu volume.

2.2.4.1 Modelo de otimização para leilão combinatório de compras de serviços de transporte

O modelo matemático de otimização combinatória tem como objetivo minimizar o custo total de compra do embarcador, sujeito a restrição de cobertura de todas as linhas da sua rede. Sua formulação matemática é a seguinte:

Parâmetros:

P = Conjunto dos pacotes submetidos pelas transportadoras

T = Conjunto de transportadoras que participam do leilão

L = Conjunto de linhas, pertencentes à rede do embarcador

l = Linha, $l \in L$

p = Pacote (conjunto de linhas), $P = \{L | l \in L\}, p \in P$

t = Transportadora, $t \in T$

a'_{pt} = Vetor binário associado ao pacote p do fornecedor t que indica se a linha l pertence ou não ao pacote. Assim, $a'_{pt} = 1$ se a linha está no pacote submetido pelo fornecedor, $a'_{pt} = 0$ em caso contrário

C_{pt} = Custo associado ao pacote p submetido pela transportadora t

Variável decisão:

x_{pt} = Variável binária associada ao pacote p submetido pela transportadora t . Se $x_{pt} = 1$, o pacote p da transportadora t é um dos vencedores do leilão combinatório. Em caso contrário, $x_{pt} = 0$

Função objetivo:

$$\text{Minimizar o Custo Total de Compra do Embarcador} = \sum_{p \in P} \sum_{t \in T} C_{pt} x_{pt} \quad (1)$$

Restrições:

$$\text{Cobertura da rede do embarcador:} \quad \sum_{p \in P} \sum_{t \in T} a'_{pt} x_{pt} = 1 \quad \forall l \in L, \quad (2)$$

$$\text{Binária:} \quad x_{pt} \in \{0,1\} \quad \forall p \in P, \forall t \in T. \quad (3)$$

A função objetivo da eq. (1) assegura que os pacotes selecionados x_{pt} minimizem o custo total de compra. A restrição de cobertura da eq.(2) estabelece que cada linha da rede do embarcador seja atendida por uma transportadora, e a restrição da eq.(3) assegura a binariedade da variável de decisão.

Dadas as características do problema de pesquisa, o *WDP* não aceita livre disposição de itens. Isto é, não se permite incluir ou excluir linhas dos pacotes selecionados. O embarcador deve garantir à transportadora que as linhas incluídas em seu pacote serão todas designadas a mesma.

Um problema com configurações semelhantes é conhecido na literatura como Partição de Conjuntos, caso especial do problema de Cobertura de Conjuntos.

2.3 MÉTODOS UTILIZADOS PARA SOLUCIONAR O *WDP*

Diferentes autores têm procurado resolver o problema de determinação do vencedor através de diferentes perspectivas. A maioria dos métodos apresentados nesta Seção corresponde a aplicações de leilões tradicionais. Embora o foco da atual pesquisa seja resolver um problema de leilão reverso, considerou-se pertinente expô-los dado que são problemas análogos e, conseqüentemente, podem ser resolvidos utilizando as mesmas abordagens.

2.3.1 Métodos Exatos

2.3.1.1 *Branch and Bound (B&B)*

Um busca *B&B* pode ser utilizada para examinar todas as possíveis designações das linhas às transportadoras para instâncias pequenas, onde o tempo de processamento depende da função de corte empregada, vide Viswanathkumar e Srinivasan (2002). O *B&B* começa com uma solução vazia e divide o problema em K estágios recursivos. No estágio k , quando as linhas de 1 até $l-1$ têm sido designadas, as transportadoras que fizeram lances para a linha l são examinadas. A linha l é designada a uma transportadora t se a atual melhor solução supera o limitante inferior; em caso contrário, a busca é descartada e se continua a examinar a próxima transportadora. O processo continua recursivamente até designar todas as linhas. (GUO et al. 2006)

2.3.1.2 *Restrição de combinações para garantir determinação ótima do vencedor em tempo polinomial*

Um método trivial para lidar com a complexidade do problema é limitar o espaço de busca a uma quantidade que possua solução plausível em tempo polinomial. Algumas regras que inibem a negociação nesse sentido são: limitar o número de participantes do leilão (ofertantes), limitar diretamente a quantidade de propostas emitidas por cada ofertante e criar propostas, a priori, sobre as quais os ofertantes realizarão os lances.

Embora essas práticas sejam uma ferramenta para diminuir o *trade-off* entre otimalidade e tempo de execução, comprometem a eficiência do leilão (SANDHOLM, 2000). Por um lado, incorrem num custo de oportunidade ao deixar de admitir propostas com grande potencial para formar uma boa solução. Por outro lado, ofertantes possivelmente ficarão insatisfeitos ao fazer lances sobre propostas criadas pelo leiloeiro, já que não expressam suas preferências.

2.3.1.3 *Enumeração exaustiva*

Segundo Sandholm (2002), a enumeração exaustiva pode ser utilizada em casos onde o espaço de busca é pequeno, visto que sua aplicação exige listar todas as propostas submetidas e comparar todas as possíveis combinações. Dessa forma, se fosse analisado o pior caso, isto é, cada combinação de itens que recebe pelo menos uma proposta, o espaço de busca seria exponencial ao número de itens.

Em termos gerais, as propostas chegam a ser imensuráveis, exceto se a quantidade de itens for pequena. Por tal razão, a enumeração exaustiva não é um método de busca viável na maioria dos casos.

2.3.1.4 *Programação dinâmica*

Outro método utilizado para encontrar a solução ótima do *WDP* é a programação dinâmica. A diferença da enumeração exaustiva é que precisa computar as propostas repetidas vezes para chegar à solução ótima, já a programação dinâmica processa as propostas uma única vez. Por tal motivo, a programação dinâmica

apresenta menor complexidade computacional. Ainda assim, continua sendo um método de difícil implementação para grandes conjuntos de itens (acima de 20 - 30). (SANDHOLM, 2002)

Rothkopf, Peke e Harstad (1998) desenvolveram um algoritmo de programação dinâmica onde se determina para cada conjunto de itens a maior receita que poderia ser obtida. O algoritmo começa explorando a partir dos menores conjuntos até chegar aos maiores. Com a exploração sistemática a solução ótima pode ser obtida com qualquer combinação de subconjuntos de itens. Pode-se observar que sua complexidade não depende do número de propostas, mas sim do tamanho do conjunto de itens. Isto é, o algoritmo gera todas as possíveis combinações, embora nem todas tenham sido ofertadas, gerando um número de passos computados que é sempre o mesmo. Por outro lado, cada item acrescentado incrementa sua complexidade de maneira exponencial.

2.3.1.5 Algoritmos com curvas de oferta lineares e segmentadas

Dang e Jennings (2003) realizaram duas contribuições importantes à literatura. Primeiro, desenvolvem uma forma compacta para a representação das propostas dos leilões combinatórios multi-unidades, com curvas de propostas que permitem expressar correlação entre itens. Segundo, constroem algoritmos que obtêm a solução ótima em alguns casos. Finalmente, analisam a complexidade computacional dos casos onde as curvas de demanda/oferta são curvas lineares segmentadas (*piece-wise linear functions*). Uma curva de demanda/oferta representa o preço designado para determinado conjunto de itens. Dessa forma, a posição i da abscissa representa um conjunto de itens com quantidades específicas de cada um, e a ordenada na posição j representa o preço relativo a esse conjunto. No entanto, os autores reconhecem que a complexidade dos algoritmo restringe sua aplicação em instâncias reais.

2.3.1.6 Algoritmos irrestritos

Uma alternativa às abordagens anteriores que está sendo amplamente utilizada, pretende melhorar a eficiência dos leilões evitando restringir os ofertantes de submeter suas propostas conforme sua preferência. Esses algoritmos apelam principalmente à estrutura em árvore para melhorar sua eficiência de busca. Alguns dos mais apontados pela literatura são apresentados a seguir:

- Combinatorial Auction Structured Search (CASS): consiste em encontrar a melhor alocação com base numa busca em profundidade - *Depth First Search (DFS)*. A fim de reduzir seu espaço de busca, o algoritmo propõe estruturar as propostas classificando-as segundo seu item de menor valor. Esse método chamado de *BINS* pretende facilitar a busca dos filhos (propostas) que posteriormente serão inseridos na árvore de busca. Um caminho na árvore representa uma solução factível e está composto por uma proposta de cada categoria, já que as propostas pertencentes a uma categoria são mutuamente excludentes (FUJISHIMA, LEYTON-BROWN e SHOHAM, 1999).
- Pré-processamento e IDA^* : Sandholm (2002) construiu um algoritmo em árvore, onde cada caminho corresponde a uma alocação factível. O valor de cada alocação é comparado com o melhor valor encontrado até esse instante. Garante-se a solução ótima porque o algoritmo percorre todas as alocações potenciais. Para uma busca mais eficiente foi implementada uma estratégia chamada *Iterative Deepening A^* (IDA^*)*. Essa estratégia aproveita a *DFS*, a qual é executada em tempo linear. Seu desempenho possibilita soluções rápidas. Por outro lado, se necessário, o algoritmo pode ser finalizado prematuramente obtendo como resultado a melhor solução factível encontrada até esse ponto. Antes de iniciar a busca, as propostas são submetidas a pré-processamentos que visam identificar aquelas não competitivas com o propósito de descartá-las. Assim como CASS, o IDA^* consegue reduzir seu tempo de busca cortando os galhos da árvore de busca que atendem à seguinte condição: $g(n) + h(n) < T(A)$ onde $g(n)$ denota a soma das propostas adicionadas ao caminho acima do nó n , e $h(n)$ denota a receita esperada dos itens que ainda não foram alocados.

- Busca híbrida: Song e Yang (2004) desenvolveram um algoritmo que utiliza o método de *B&B* conjuntamente com programação linear. A visitação é realizada com *Best First Search (BFS)* em todo o espaço de busca e, eventualmente utiliza *DFS* quando existe uma pequena porção de propostas pendentes. O autor afirma que sua estratégia híbrida reduz o uso da memória computacional. A atualização da solução é condicionada ao limitante superior em cada iteração. O limitante superior é calculado uma única vez utilizando programação linear, depois o mesmo é modificado com outro critério; isso evita cálculos repetitivos e contribui à diminuição do tempo computacional.
- Combinatorial Auctions Branch On Bids (CABOB): Sandholm et al. (2005) desenvolvem um algoritmo em árvore que utiliza uma busca em profundidade e *B&B* para ramificar as propostas. Conjuntamente se mantém um grafo que guarda as propostas ainda não expandidas na árvore de busca. Os vértices são removidos do grafo ao passo que se desce na árvore e, são reinseridos quando se faz o *backtracking*. O algoritmo utiliza limitante inferior e superior, os quais são calculados solucionando o problema linear (*LP*) a priori. Além disso, a contribuição à receita dada por cada item é calculada explorando técnicas de vizinhança. O *benchmark* demonstra uma solução ótima obtida em menor tempo ao comparar-se com softwares comerciais como o CPLEX.

2.3.2 Métodos de aproximação

Outra abordagem do problema é a utilização de algoritmos que geram boa solução em tempo computacional reduzido, em vez de uma solução ótima com custos computacionais elevados. Para garantir um bom desempenho do algoritmo, deve-se atender ao limite do pior caso (*worst-case bound*), ou seja, garantir que o valor da melhor alocação é menor ou igual a uma constante, k , vezes o valor da melhor alocação encontrada pelo algoritmo (SANDHOLM, 2002).

Alguns exemplos de algoritmos aproximados são apresentados a seguir:

2.3.2.1 *Limited Discrepancy Search (LDS)*

Esta abordagem limita o esforço de busca a um intervalo onde existe uma alta probabilidade de encontrar a solução ótima. A fim de gerar a árvore de busca, uma pontuação, $dis(n)$, é estabelecida para a proposta n ; assim, propostas são adicionadas à árvore da esquerda para a direita começando pela menor pontuação (SAKURAI, YOKOO e KAMEI, 2000).

2.3.2.2 *Relaxação Lagrangiana*

Determina a alocação e os preços através de Multiplicadores de Lagrange, como uma forma de adaptar-se a ambientes dinâmicos onde licitantes e itens podem mudar com o tempo. Os Multiplicadores de Lagrange são, em termos econômicos, o custo marginal de um item, quando são utilizados para relaxar restrições de demanda. Igualmente podem ser interpretados como o benefício marginal de usar um item adicional. Dessa forma, o comprador anuncia os conjuntos de itens e ajusta os preços para cada item. Baseados nos preços de cada item, licitantes submetem propostas; ganha quem oferecer o menor preço. Um mecanismo de ajuste iterativo de preço permite que uma solução seja encontrada.

Hsieh e Huang (2010) adotaram uma abordagem de relaxação lagrangiana para um caso de múltiplos compradores. O algoritmo propriamente dito consiste em: (1) um algoritmo para solucionar subproblemas, (2) um método para solucionar o problema dual e (3) uma heurística para encontrar uma solução factível aproximada. Não foi comprovada sua aplicação para problemas de grande porte.

Além da obtenção de soluções eficientes em tempo computacional moderado, considera-se fundamental ter a possibilidade de alocar fornecedores com base em avaliações abrangentes que mensurem todos os critérios relevantes do cenário de negociação. A seção 2.4, descreve com mais detalhe abordagens com esse propósito.

2.4 COMPRAS DE SERVIÇOS DE TRANSPORTE COM MÚLTIPLOS CRITÉRIOS

Esta Seção apresenta a evolução da utilização de múltiplos critérios nas compras de serviços de transporte. A seção 2.4.1 descreve a abordagem mais utilizada nesta área, conhecida como abordagem de custo total. A seção 2.4.2 explica como teorias da área de análise de decisão são utilizadas para tratar esse problema.

Estudos descritivos que analisaram o *trade-off* entre o custo e o nível de serviço foram a principal iniciativa de abordagem multi-critério (CAPLICE, 1996). Posteriormente veio a aplicação de estratégias genéricas de compras à seleção de transportadoras; tais estratégias auxiliavam na pré-seleção das transportadoras participantes do leilão (CAPLICE e SHEFFI, 2003). Aquelas selecionadas entram no “*core carrier group*” o que significa que elas possuem características necessárias ou suficientes para satisfazer as exigências do embarcador relacionadas ao serviço de transporte. Esse tipo de estratégia avalia o potencial das transportadoras para atender todas as linhas ignorando suas características individuais, especialmente o nível de serviço (CAPLICE e SHEFFI, 2003), (SHEFFI, 2004).

Visando analisar cada linha de forma particular, Caplice (1996), Caplice e Sheffi (2003), Sheffi (2004) propõem permitir a participação de todas as transportadoras no processo de seleção e só avaliar seu desempenho durante a negociação.

Com esse propósito algumas abordagens têm sido exploradas. Estas vão desde a modificação do coeficiente de custos de modelos tradicionais como *WDP*, denominada também abordagem do custo total (Ver Seção 2.4.1), até a utilização de ferramentas de análise de decisão que designam pesos e funções de valor para priorização de diferentes critérios (ver Seção 2.4.2) (BICHLER e KALAGNANAM, 2005).

2.4.1 Abordagem de custo total

Modelos que avaliam critérios diferentes ao custo por linha, segundo uma abordagem de custo total, são sugeridos por Caplice (1996) e Caplice e Sheffi

(2003), Sheffi (2004) e Guo et al. (2006). Trata-se de uma modificação do coeficiente de custos da função objetivo utilizando fatores multiplicativos ou aditivos, a fim de mensurar o desempenho da transportadora.

Os autores sugerem modificar o modelo mono-critério de Partição de Conjuntos (*Carriers Assignment Model – CAM*). Esse modelo tem como único objetivo minimizar o custo total do embarcador, sujeito à restrição de cobertura e às restrições de contorno. O *CAM* deve ser modificado através de penalidades (fatores de desempenho, Caplice e Sheffi (2003), ou *penalty cost*, Guo et al. (2006)) impostas aos coeficientes da função objetivo a fim de representar o nível de serviço das transportadoras.

Um fator multiplicativo pode ser construído baseado em *rankings* subjetivos ou objetivos da transportadora. Por exemplo, o embarcador define que o nível de serviço é composto de três componentes: entregas no prazo, taxa de recusa e facilidade de fazer negócios. As transportadoras são classificadas de acordo com esses critérios. Os *rankings* são combinados em um multiplicador aplicado aos pacotes de linhas correspondentes (CAPLICE, 1996).

O fator aditivo consiste em transformar o nível de serviço em custo para inseri-lo nos respectivos coeficientes. No exemplo apresentado, uma transportadora pode diminuir seu tempo de entrega de 5 dias para 3 dias, porém há um incremento no custo. Esse incremento pode ser estimado e subtraído do preço do pacote específico. Dessa forma, a transportadora é recompensada por prover uma rápida entrega (CAPLICE, 1996).

Os custos que somente são alocados a instalações ou transportadoras como um todo devem ser incluídos como custos fixos. Assim mesmo, independente do método utilizado, os coeficientes modificados só servem para realizar a seleção de transportadoras. O valor pago pelo embarcador é o valor original do pacote.

As pesquisas mencionadas posicionam-se de forma similar à respeito da dificuldade em selecionar os fatores de desempenho e especificar seu valor monetário para realizar uma análise adequada. Além disso, concordam sobre a importância de adicionar esses fatores ao modelo de otimização pela possibilidade de realizar *trade-offs*.

2.4.2 Análise de decisão

Além das características de cada linha, consideram-se importantes as características próprias de cada critério, por exemplo, sua relevância e seu comportamento frente a mudanças. Portanto, avaliar cada critério em termos de custo e incorporá-lo na função objetivo (abordagem de custo total) pode não ser o tratamento mais adequado.

Por exemplo, se o percentual de avarias apresenta diminuição não linear com relação à melhoria do desempenho, torna-se difícil mensurá-lo monetariamente para inseri-lo como penalidade. Essa valoração deve estar condicionada à percepção do embarcador sobre as implicações de aceitar um índice maior de avarias em quanto pode ser valorado.

O impacto de um atributo e a forma como este é mensurado causam diferenças na seleção final de fornecedores. Por causa disso, ferramentas alternativas para a tomada de decisão em problemas multi-critérios têm sido abordadas. Weber e Current (1993) afirmam que uma das vantagens da análise multi-critério é permitir que os critérios sejam analisados em suas unidades de medida natural, eliminando a convergência em uma unidade única, a qual implica monetarização de intangíveis.

Em termos gerais, as técnicas de análise de decisão consistem em designar pesos e criar funções de valor para cada critério, para calcular pontuações (valores). Os ofertantes (transportadoras) podem competir diante desses valores melhorando o desempenho dos atributos (BICHLER e KALAGNAMANG, 2005).

Existem vários métodos de tomada de decisão, com relação à seleção de fornecedores, que envolvem múltiplos critérios. Ho et al. (2010) realizaram uma revisão da literatura das principais aplicações entre os anos 2000 e 2008, e identificam que entre as abordagens individuais mais utilizadas estão: Análise Envoltória de Dados (*Data Envelopment Analysis – DEA*), Programação Matemática, Análise Hierárquica (*Analytic Hierarch Process – AHP*), e Raciocínio Baseado em Casos (*Case-Based Reasoning – CBR*). Quando analisadas as abordagens integradas, a mais popular entre as pesquisadas é *AHP* com Programação Multi-objetiva (*Goal Programming – GP*). A respeito dos critérios mais relevantes durante o processo de seleção, o autor menciona a qualidade, seguida do tempo de entrega e o preço.

Bichler e Kalagnaman (2005) descreveram dois pressupostos fundamentais subjacentes ao tratamento de leilões multi-atributo na literatura. O primeiro refere-se às propostas como pontuais e especificadas em pares (atributo-valor) para cada atributo, devido às dificuldades técnicas em expressar propostas de maior complexidade. O outro assume que os contratos do leilão são adjudicados a uma única oferta (*sole sourcing*), baseado no argumento que leilões multi-critério são utilizados principalmente para licitações com alta especificidade de ativos. Para rebater esses pressupostos os autores, além de introduzir vários fornecedores, recomendaram propostas configuráveis permitindo aos ofertantes especificar os possíveis valores de um atributo/critério em formato funcional (em função do preço). A respeito da seleção de serviços de transporte, recentemente, Prado (2011) desenvolveu um estudo aplicando o processo de Apoio à Decisão Multi-critério (*Multi-Criteria Decision Analysis – MCDA*) utilizando Pensamento Focado em Valor (*Value Focused Thinking - VFT*) para conduzir o processo decisório. Esta última consiste em duas atividades: primeiro decidir o que se quer com a decisão e posteriormente imaginar as alternativas para alcançar o objetivo. A abordagem *VFT* deve produzir melhores resultados se comparada com a abordagem de Pensamento Focado em Alternativas (*Alternative Focused Thinking – AVT*) (KEENEY, 1992). Prado (2011) resolveu um problema de seleção de fornecedores cujo objetivo era escolher uma transportadora para cobrir uma linha levando em conta vários critérios; nesse caso; o autor não utilizou propostas combinatórias. Inicialmente, o autor realizou uma pré-qualificação de transportadoras utilizando critérios de inclusão e rejeição. Os critérios selecionados foram nível de serviço e custo de transporte. O modelo proposto pelo autor está baseado na Teoria de Valor Multi-atributo (*Multi-Attribute Value Theory - MAVT*) que se fundamenta na utilização de otimização com múltiplos critérios.

2.4.2.1 Teoria de Valor Multi-atributo (*MAVT*)

O modelo de decisão *MAVT* geralmente possui uma função objetivo baseada nos *trade-offs* de valor. Essa função é conhecida como função de utilidade aditiva ou função de valor (KEENEY, 2002).

$$V(x, y) = w_1 v_1(x) + w_2 v_2(y), \quad (4)$$

$$w_1 + w_2 = 1 \quad (5)$$

A função de valor linear aditiva $V(x, y)$ é composta por duas funções de valor v_1 e v_2 , pertencentes aos objetivos (critérios x e y) avaliados (ver eq.(4)). Os valores w_1 e w_2 são fatores de dimensionamento (pesos), comumente normalizados (ver eq.(5)), os quais são especificados fazendo *trade – off* entre os critérios (ver Seção *trade – offs*). Na prática, a priorização das preferências do tomador de decisão (embarcador), e conseqüentemente a construção de funções de valor apropriadas são de fundamental importância (BICHLER E KALAGNAMAN, 2005).

Adicionalmente, torna-se essencial que a função $V(x, y)$ assuma independência preferencial em todos os critérios. Ou seja, um critério x é preferencialmente independente de y se preferências por resultados específicos de x não dependem do valor do atributo y (OLSON, 1996).

A abordagem *MAVT* utiliza as funções de valor para dar as pontuações aos critérios que estão sendo avaliados. Ou seja, o tomador de decisão constrói uma função de valor para cada critério, a partir dela deve atribuir uma pontuação (valor), em escala de 0 a 100, ao desempenho do critério (neste caso, o desempenho é relativo a cada oferta enviada pela transportadora) (MONTIBELLER e FRANCO, 2010). Esse valor seria um reflexo do comportamento (algumas vezes não linear) do critério quando medido em sua escala natural (BELTON, 1986).

Um pressuposto importante a respeito da função de valor é sua monotonicidade. A principal justificativa é que o valor de qualquer atributo é estritamente crescente ou decrescente (BICHLER E KALAGNAMAN, 2005).

Uma alocação ótima do leilão nesse modelo de decisão é designar as propostas aos fornecedores de modo que o valor do leiloeiro/ tomador de decisão ($V(x, y)$) seja maximizado (BICHLER E KALAGNAMAN, 2005).

2.4.2.2 Trade-offs

Segundo Keeney (2002), “Fazer o julgamento: o quanto se está disposto a ceder de um objetivo para melhorar outro objetivo, é a essência dos *trade-offs* de valor”.

Uma preocupação constante no campo da análise de decisão é a realização de *trade-offs* adequados, para designar os pesos relacionados aos critérios. Embora a designação dos pesos esteja baseada em julgamentos subjetivos, várias técnicas têm sido propostas na literatura para contribuir na determinação de pesos consistentes. Para Yoshizaki e Montibeller (2011), as principais são:

- Swing Weights: Deve-se designar 100 pontos ao critério que desejam melhorar primeiro (primeiro *swing*), do menos ao mais preferido nível. Os *swings* subsequentes, dos demais critérios, são ancorados ao primeiro *swing*. Depois, os pesos são normalizados com cada *swing* dividido pela soma dos *swings*. Por exemplo, ao designar pesos para avarias e *lead time*: 100 pontos são dados à diminuição de avarias de 1% para 0,5%, de acordo com este valor/*swing*, a melhoria de 5 para 2 dias no *lead time* pode ter um valor/*swing* de 70. Assim, o peso dado às avarias seria de $w_{AV} = 100 / 170 = 0.59$, e para *lead time* de $w_{LT} = 70 / 170 = 0.41$.
- Método trade-off: os tomadores de decisão devem considerar duas situações hipotéticas, onde um critério está no maior nível de preferência e o outro está no menor nível de preferência. Eles devem selecionar entre as duas opções e então reduzir o desempenho do critério que esteja no maior nível para o nível x , até que ambos sejam indiferentes. Por exemplo: o decisor deve escolher entre 1% de avarias e 2 dias de *lead time* ou 0,2% de avarias e 5 dias de *lead time*, no caso de selecionar a segunda opção, o decisor deve aumentar as avarias até que o valor: (1% avarias, 2 dias *lead time*) = (x % avarias, 5 dias *lead time*). Assim, os pesos são obtidos resolvendo:

$$w_{AV} \cdot v_{AV}(1\%) + w_{LT} \cdot v_{LT}(2\text{dias}) = w_{AV} \cdot v_{AV}(x\%) + w_{LT} \cdot v_{LT}(5\text{dias})$$
, onde o valor do maior nível é 100, o valor do menor nível é 0, e o valor de x pode ser obtido da função de valor.
- Pricing out: Se as funções de valor são aproximadamente lineares, então é possível avaliar os pesos como uma unidade equivalente ao custo. Assim, o valor

do atributo é convertido através da função de valor para uma moeda virtual (BICHLER et al., 2005). Responde-se a pergunta: quanto poderia custar incrementar em uma unidade um determinado critério? Esta abordagem é útil quando os critérios são quantitativos e é possível estimar custos unitários por atributo.

Na prática, se as funções de valor são não lineares, o método *Swing Weights* é o mais utilizado, já que é eficiente e de fácil compreensão (YOSHIZAKI E MONTIBELLER, 2011).

2.4.3 Modelo matemático de múltiplos critérios para leilão combinatório de compras de serviços de transporte

No problema de compras as questões mais relevantes para o embarcador são a ponderação dos critérios e as funções de valor. A respeito da transportadora, pode-se dizer que suas propostas serão mais bem direcionadas, para atender requisitos específicos de cada critério, se o embarcador disponibilizar informação com a importância relativa (peso) que tem cada critério.

A seleção dos critérios também é uma questão fundamental na análise de decisão e existem técnicas projetadas especialmente para esse feito, mas este não é o foco da pesquisa. Para mais detalhes ver Prado (2011). Para esta dissertação assume-se que os critérios já foram selecionados.

O modelo matemático combinatório multi-critério proposto para solucionar o problema de compras de serviços de transporte, baseado na Teoria de Valor Multi-Atributo (MAVT), é apresentado a seguir:

Parâmetros:

P = Conjunto de propostas submetidas pelas transportadoras

T = Conjunto de transportadoras que participam do processo de compra

L = Conjunto de linhas que compõem a rede do embarcador

C = Conjunto de critérios que o embarcador quer avaliar

p = Proposta (pacote), $p \in P$

t = Transportadora, $t \in T$

l = Linha, $l \in L$

c = Critério, $c \in C$

w_c = Peso do critério

a_{pt}^l = Vetor binário que toma o valor de 1 se o pacote p do fornecedor t tem a linha l e, toma o valor 0 em caso contrario

V_c^{pt} = Valor da função do critério c para o pacote p do fornecedor t

Função:

$F(c)$ = Função de valor do critério c

Variável decisão:

X_{pt} = Recebe o valor de 1 se o pacote p do fornecedor t é vencedor do leilão, 0 em caso contrario

Função objetivo:

$$\text{Maximizar o Valor Total} = \sum_{p \in P} \sum_{t \in T} x_{pt} \left[l \sum_{c \in C} w_c * V_c^{pt} \right] \quad (6)$$

Restrições:

$$\text{Cobertura da rede:} \quad \sum_{p \in P} \sum_{t \in T} a_{pt}^l x_{pt} = 1 \quad \forall l \in L, \quad (7)$$

$$\text{Binária:} \quad x_{pt} \in \{0,1\} \quad \forall p \in P, \forall t \in T \quad (8)$$

$$\text{Normalização:} \quad \sum_{c \in C} w_c = 1 \quad (9)$$

Valores:

$$V_c^{pt} = F(c), \quad \forall p, \forall t, \forall c$$

A função objetivo da eq.(6) assegura que as propostas selecionadas maximizem o valor total da compra, isso significa que as propostas selecionadas serão aquelas que tiverem maior valorização e satisfaçam as restrições. A valorização das propostas realiza-se da seguinte forma: o nível de desempenho descrito na proposta para cada critério $c \in C$, é avaliado utilizando as funções de valor, $F(c)$. A função

de valor normalmente dá uma valoração para cada critério entre 0 e 100. Esses valores V_c^{pt} são multiplicados pelo peso W_c correspondente a cada critério.

Finalmente, da somatória dessas multiplicações resulta o valor da proposta.

O valor de cada proposta é uma medida de quão boa ela é, e através desta valoração a proposta é selecionada ou não. Se a variável $X_{pt} = 1$ a proposta p do fornecedor t foi selecionada, em caso contrario $X_{pt} = 0$.

As restrições da eq.(7) e da eq.(8) são originais do *WDP*; garantem a cobertura de todas as linhas e a binariedade da variável decisão. A restrição da eq.(9) estabelece que os pesos sejam normalizados de forma que sua somatória seja igual a 1.

2.5 CONCLUSÃO DA SEÇÃO 2

Este capítulo introduziu o leilão combinatório como um mecanismo de negociação que possibilita expressar preferências dos ofertantes, ao ter a liberdade de submeter o conjunto de itens que melhor beneficiam sua estrutura de custos. Ao mesmo tempo pode satisfazer, as exigências do mínimo custo do leiloeiro como consequência das economias geradas pelos fornecedores, e exigências adicionais inserindo as respectivas regras. Esse problema é tratado na literatura a partir do *WDP*. Devido a sua complexidade, problemas de maiores dimensões precisam de algoritmos adequados para serem resolvidos (SANDHOLM, 2000); (SONG e YANG, 2004); (SONG e REGAN, 2005); (LEHMANN, MULLER e SANDHOLM, 2006).

Foram também analisadas as vantagens que esse tipo de leilão traz à compra de serviços de transporte. Entre elas a mais relevante é a economia de escopo, produto das propostas complementares que favorecem os fornecedores na diminuição de custo e, consequentemente, aos compradores na redução do custo total de movimentações vazias. Adicionalmente, foi comprovado que o leilão combinatório é um problema pouco explorado na área de transporte, cujas principais contribuições estão relacionadas ao desenvolvimento de modelos matemáticos e análises a respeito das condições ideais de negociação.

Da utilização de múltiplos critérios, pode-se afirmar que a técnica mais explorada pela literatura é a de custo total, a qual se resume em penalidades que são

adjudicadas aos coeficientes de custo. Quando o custo de um critério é descontado dos coeficientes do modelo, aquelas propostas que possuem “menor custo” são favorecidas, ou seja, aquelas que apresentam um melhor desempenho do critério. Mas, na hora de alocar, o valor cobrado pelo serviço será o valor original da oferta.

A técnica de análise de decisão tem sido pouco estudada para seleção de transportadoras. Uma aplicação que vale a pena ressaltar é Prado (2011) onde o processo da *MCDA* foi aplicado para escolher o fornecedor que atenderá uma linha. Esta técnica utiliza uma função de valor aditiva para mensurar a interação entre os critérios e selecionar as transportadoras que maximizam o valor da função. No entanto, essa aplicação não explora economias de escopo porque a seleção da transportadora é feita para uma linha.

As pesquisas também mostraram a importância do nível de serviço como critério de seleção e a dificuldade enfrentadas pelas empresas para mesurá-lo e avaliar seu *trade-off* com o custo de transporte.

Adicionalmente foram mencionadas algumas vantagens da utilização de técnicas multi-critério sobre as mono-critério. As primeiras fornecem ao decisor (empresa) um conjunto alternativo de soluções associadas às ponderações dadas a cada critério, em vez de uma solução ótima. Além disso, permitem avaliar os critérios em sua unidade de medida natural (através das funções de valor); diferente da abordagem de custo total que exige transformação em unidade monetária.

Conforme já exposto, foi evidenciada a necessidade de desenvolver algoritmos que ajudem a capturar as economias de escopo presentes nos fornecedores, como uma forma para chegar a um menor custo de compra. Mostrou-se também a importância da incorporação da análise multi-critério devido à relevância que tem durante o processo de seleção de fornecedores. Com esse propósito, esta pesquisa descreve um algoritmo que pode ser utilizado para resolver o problema de leilão combinatório direcionado à procura de transportadoras detalhado no Capítulo 3. Em primeira instância o algoritmo busca minimizar o custo total de compra do embarcador. Posteriormente, o algoritmo é adaptado para avaliar vários critérios, utilizando métodos da análise de decisão.

3 ALGORITMO ADAPTADO PARA COMPRAS DOS SERVIÇOS DE TRANSPORTE DE CARGA LOTAÇÃO

3.1 ALGORITMO PARA SOLUCIONAR O PROBLEMA *WDP* MONO-CRITÉRIO

Uma vez revisadas as diferentes abordagens para este problema, pode-se dizer que os algoritmos irrestritos são uma boa alternativa para ser aplicada nas compras de serviços de transportes. Estes se mostram mais flexíveis a respeito de fatores relevantes para o tratamento deste caso; por exemplo, não limitam o tamanho do problema ao número de ofertantes, itens ou propostas submetidas. No leilão de compra, estes parâmetros têm um espaço de variação muito amplo que depende do tamanho da rede do embarcador a ser leiloada. Em outras palavras, quando a rede a ser coberta é muito extensa, cresce a quantidade de fornecedores e, conseqüentemente cresce o número propostas submetidas. A limitação de algum desses parâmetros implica perder a oportunidade de obter uma alocação mais eficiente, porque está deixando de considerar propostas com potencial ganhador; por esse motivo algoritmos irrestritos devem ser empregados.

Outro aspecto, não menos importante, é a questão da otimalidade da solução. Entre esses algoritmos, alguns como *CASS*, *CABOB* e *IDA** comprovaram soluções ótimas em um tempo de execução linear $O(P)$ a respeito do número, P , de propostas. Isso demonstra sua capacidade para tratar a complexidade do problema sem reduzir sua eficiência nos resultados. Uma análise mais detalhada revelou características estruturais semelhantes entre esse grupo de algoritmos, principalmente estratégias que se beneficiavam da estrutura em árvore para explorar o espaço de busca.

Dessa forma chegou-se ao algoritmo base desta pesquisa, desenvolvido por Sandholm (2002). Além de possuir as características mencionadas anteriormente, sua estratégia de busca, *Iterative Deepening A** (*IDA**), é uma das mais sofisticadas da literatura (ver Seção 3.1.2.3). O algoritmo mono-critério de Sandholm busca solucionar o problema tradicional de determinação do vencedor, no seguinte contexto: Um vendedor disponibiliza um conjunto de itens e um comprador submete

propostas combinatórias. As propostas são avaliadas pelo algoritmo e são selecionadas aquelas que maximizam a receita do vendedor. A estrutura é simplificada porque o algoritmo não possui aplicação a um ambiente específico de negociação.

Enquanto Sandholm (2002) procura maximizar a receita do vendedor, este trabalho procura minimizar o custo do comprador (embarcador). Além disso, o autor desenvolve o algoritmo permitindo lances só de um ofertante, restrição que é desconsiderada neste trabalho a fim de admitir várias transportadoras para aumentar a abrangência do algoritmo. Como consequência dessa divergência entre objetivos, é também necessário modificar o cálculo de algumas funções e parâmetros do algoritmo. Essas mudanças serão devidamente ressaltadas ao longo do texto para facilitar a compreensão do leitor.

Antes de executar o algoritmo é realizada a “preparação dos parâmetros de entrada” onde as propostas recebidas são adequadas ao formato binário, excetuando preço e fornecedor. Em seguida, na implementação do algoritmo são executadas várias estratégias de busca (ver Seção 3.1.2), a fim de encontrar uma alocação ótima que minimize o custo do comprador. A estrutura em árvore é a principal ferramenta utilizada durante a exploração do espaço de busca. O algoritmo é executado em uma única rodada e a inserção dos múltiplos critérios será realizada na Seção 3.2.

3.1.1 Parâmetros de entrada e pressupostos

3.1.1.1 Parâmetros de entrada

- Pacotes (propostas): são derivados da rede de clientes que o comprador precisa servir. Uma linha é a menor unidade sendo leiloada que corresponde ao par (origem-destino), ou seja, não há diferenciação entre volume e linha. Se uma transportadora for selecionada para atender uma linha, ela deve transportar toda a mercadoria destinada (demandada) para essa linha. Sua denotação é feita da seguinte forma: na Figura 3 as linhas formadas pelos pares origem-destino:

(A,B); (B,C); (C,B); (C,A), são representadas através de um número inteiro, assim, (A,B)=1; (B,C)=2; (C,B)=3; (C,A)=4.

Figura 3 - Exemplo de uma rota formada destinos A, B e C.

- Número de fornecedores: é irrestrito, ou seja, são admitidas todas as transportadoras que tenham interesse em participar do leilão e sejam aprovadas pelo comprador.
- Número de pacotes por fornecedor: é controlado pelas transportadoras e depende das linhas selecionadas para formar suas rotas. Cada uma é livre para formar rotas testando todas as possíveis combinações se desejar. As propostas são enviadas no seguinte formato:
 - Se o único critério de seleção for custo, a proposta deve conter: rota, preço e fornecedor. Por exemplo, supondo que o fornecedor 1 quer fazer um lance para (linha1,linha2,linha3), com um preço R\$ 950, a oferta submetida é $P_1 = \{ (\text{linha1}, \text{linha2}, \text{linha3}, \$950, 1) \}$.
 - Se forem vários os critérios de seleção, a proposta incluirá: rota, desempenho de cada critério e fornecedor. Por exemplo, supondo que o fornecedor do exemplo anterior também deve especificar seu percentual de avarias considerando toda a rota, o qual é de 0.5%, a oferta submetida é $P_1 = \{ (\text{linha1}, \text{linha2}, \text{linha3}, \$950, 0.5\%, 1) \}$

Pressupostos

- Pode-se observar que, no problema de compra de serviços de transporte considerado neste trabalho, a demanda de cada rota será atendida por um único fornecedor. Portanto, pressupõe-se que qualquer fornecedor que submete uma

oferta para determinada rota está com capacidade de atendê-la completamente. Não existe a possibilidade de que uma mesma rota seja atendida por vários fornecedores, embora na prática isso aconteça como uma medida para evitar a dependência de um único prestador de serviço.

- Neste trabalho não se especifica o volume nem a sazonalidade da demanda nas linhas. Assume-se que, na informação disponibilizada pelo embarcador antes de iniciar o leilão está especificada a quantidade demandada por linha e sua frequência. Dessa forma, a transportadora conhece as movimentações de todas as linhas. Assim, quando formar pacotes deve levar em conta essas movimentações para manter um volume balanceado na rota e conseguir gerar economias de escopo.

3.1.1.2 Preparação das propostas

Uma vez submetidas, as propostas são processadas e armazenadas em formato binário como parte da estratégia adotada para buscá-las e alocá-las quando for preciso. Continuando o exemplo da Seção 3.1.1.1, o novo formato é ilustrado na Figura 4.

Figura 4 - Exemplo de conversão das propostas para o formato binário

Pode-se observar que o tamanho da proposta depende da quantidade de linhas que estão sendo leiloadas, já que a representação em forma binária assim o exige. Dessa forma, todas as linhas leiloadas são incluídas na proposta, se a linha forma parte proposta recebe o valor de “1”, caso contrário seu valor é “0”. Somente as propostas submetidas pelos fornecedores formam parte do espaço de busca; isso limita a busca dentro de um espaço factível, aumentando sua eficiência nas alocações.

3.1.2 Execução do algoritmo

Envolve a aplicação de duas estratégias de busca. A primeira corresponde à busca principal do algoritmo cuja finalidade é escolher as melhores propostas. A segunda tem como objetivo administrar de forma eficiente o espaço de busca ao selecionar as propostas que vão sendo alocadas. Ambas são detalhadas nas Seções 3.1.2.1 e 3.1.2.2 respectivamente. Cabe ressaltar que, as estratégias de busca são explicadas considerando como objetivo a minimização do custo de compra do embarcador; entretanto, na Seção 3.1.3 são detalhadas as modificações realizadas em ambas as estratégias para alcançar o objetivo de maximização de valor (multi-critério).

3.1.2.1 Busca do Mínimo Custo (BMC)

Utiliza-se uma árvore de busca para alcançar esse objetivo. O principal parâmetro de entrada é uma lista de propostas em formato binário. Um caminho na árvore consiste em uma sequência de propostas que não compartilham linhas entre si, $(P_j \cap P_k = \Phi) \forall j, k$ no mesmo caminho. Em qualquer parte da busca um caminho corresponde a uma solução factível (ver Figura 5).

Seja A o conjunto de linhas atualmente alocadas no caminho (ver eq.(10)):

$$A = \bigcup_{P_p \text{ no caminho}} P_p \quad (10)$$

Sendo L o total de linhas, o conjunto E de linhas que esperam por alocação é eq.(11):

$$E = L - A \quad (11)$$

Um caminho é finalizado quando nenhum pacote pode ser adicionado. Isso acontece quando para cada proposta, alguma de suas linhas já foi adicionada ao caminho. Enquanto as propostas vão sendo adicionadas, seu preço vai incrementando o custo

do caminho, Cc da eq.(12), cujo valor final é a somatória das propostas alocadas nele:

$$Cc = \sum_{P_p \text{ no caminho}} \text{Custo } P_p \quad (12)$$

Em cada nó de busca o custo estimado do caminho (custo do caminho+ estimativa do custo das linhas ainda não alocadas) é comparado com o menor custo encontrado até o momento na árvore de busca, para determinar se a atual solução (caminho) é a melhor encontrada. Se o custo estimado é maior que a solução atual, então não vale a pena continuar descendo na árvore (construindo esse caminho), portanto, essa ramificação é cortada e a busca continua. Esse ponto de corte foi adaptado do algoritmo a fim de obter o menor custo total; foram efetuadas mudanças no limitante de forma que atendesse ao objetivo (ver Seção 3.1.2.3 – Função de custo estimado).

Em cada folha da árvore, se o custo do caminho é menor que a atual solução, este custo é estocado como a melhor solução encontrada. Uma vez que a busca é finalizada, a solução guardada é a solução ótima.

Os filhos de um nó (proposta), na árvore da BMC, são todos os nós (propostas) que estão um nível abaixo desse nó em particular (ver Figura 5).

Um método simples de construção da árvore seria considerar toda a lista de propostas como potenciais filhos para cada nó; porém, isso implicaria um esforço computacional $O(PL)$, porque além de procurar em P propostas da lista, seria necessário percorrer todas as L linhas em cada uma delas.

Como alternativa foi utilizado um algoritmo que reduz o espaço de busca considerando os filhos que cumprem a seguinte proposição, demonstrada por Sandholm (2002):

Proposição 1: Cada partição relevante (solução) é representada exatamente por um caminho da raiz até uma folha se os filhos de cada nó são aquelas propostas que:

- Possuem a linha com o menor índice entre as linhas que ainda não foram utilizadas no caminho, $l^* = \min\{l \in (1...L)\}$, e;
- Não incluem linhas que já estão alocadas no caminho.

$$\text{filho} = \{P \in (P_1, \dots, P_n) : l^* \in P, P \cap A = \Phi\} \quad (13)$$

mais a folha. Cada nó pode assumir dois valores: “1” ou “0”, no primeiro caso significa que a linha pertence à proposta e no outro caso a linha não faz parte da proposta. Um caminho da árvore (começa na raiz e finaliza numa folha) representa uma proposta ou pacote. A rota (conjunto de linhas) é formada pelos nós que antecedem a folha, já que esta última é destinada para o preço da rota e o fornecedor que a submeteu.

A partir da raiz, a árvore é construída descendo cada nível pela esquerda ou direita. Se uma linha pertence oferta cria-se nó à esquerda em caso contrario à direita. Conforme representado na Figura 6, quando um pacote $P = \{(linha2, linha5); \$7; 3\}$ é inserido, o caminho construído é $\langle 0 \ 1 \ 0 \ 0 \ 1 \rangle$, cujo preço \$7 foi submetido pelo fornecedor 3.

Na árvore, um caminho corresponde a uma proposta. Em caso de vários fornecedores submeterem a mesma rota, todas as propostas são conservadas e especificadas na folha dessa rota; por exemplo, na Figura 3 o caminho que denota a rota formada pelas linhas 2 e 3, $\langle 0 \ 1 \ 1 \ 0 \ 0 \rangle$, recebeu uma proposta do fornecedor 1 e uma do fornecedor 2, ambas especificadas na folha do caminho. Esse método de armazenamento teve uma modificação com relação ao algoritmo de Sandholm (2002); este último armazenava uma proposta por caminho após efetuar um pré-processamento, além disso, o algoritmo não suportava vários ofertantes.

O mesmo fornecedor não pode realizar várias propostas com a mesma rota, deve-se levar em conta que uma linha pode formar parte de várias rotas, seja de forma individual ou em conjuntos diferentes. Nesse caso, se ocorrer submissão de propostas iguais, será selecionada aquela que tiver menor custo. Assim, número máximo de propostas sobre uma mesma rota equivale ao número de fornecedores.

Cabe notar que a estrutura das propostas influencia a ordem de inserção na árvore de busca principal. Visto que a BPF é uma busca em profundidade pela árvore de propostas, os primeiros caminhos a serem explorados são aqueles que possuem a maior quantidade de linhas, favorecendo as propostas que produzem maior complementaridade. Desse modo, aqueles caminhos com uma linha só são os últimos a serem explorados. Isso gera algumas vantagens, como encontrar boas soluções rapidamente e não ter a necessidade de terminar o algoritmo para obter uma boa solução factível.

Estratégia de visitação

Compreende a criação de um vetor, chamado “*Stopmask*”, com uma variável para cada linha $l \in L$. O vetor *Stopmask*[l] pode tomar os seguintes valores: *Blocked*, *Must*, *Any*, os quais apontam o estado de uma linha antes de realizar uma busca na árvore e limitam a visitação dos nós:

- *Stopmask*[l] = *Blocked* quando a linha l pertence a um pacote que foi inserido no caminho da árvore de busca principal. A visitação na árvore de propostas é restringida pela esquerda do nível l , ou seja, ela só avança pela direita indicando que as propostas que contêm a linha l são descartadas da busca atual.
- *Stopmask*[l] = *Must* é designado à linha l que possui o menor índice l^* entre as linhas que ainda não foram alocadas, indicando que os pacotes que contêm essa linha são o foco da busca. A visitação é feita pela esquerda da árvore no nível l .
- *Stopmask*[l] = *Any* significa que a linha l ainda não foi alocada, portanto, não sofre punição durante a visitação.

Em síntese, o *Stopmask* corta os caminhos das propostas que contêm linhas alocadas (*Blocked*) e ordena a visitação de modo que a próxima oferta encontrada cumpra com os requisitos de ter linhas não alocadas (*Any*), e entre elas aquela com menor índice (*Must*). Depois que uma oferta é adicionada ao caminho o *Stopmask* é atualizado conforme o novo estado de cada linha.

Quando um caminho é formado, é necessário realizar um backtracking que possibilite continuar a busca pela melhor solução. Uma vez que o último filho é adicionado ao caminho, o *Stopmask* de cada linha retorna ao valor que tinha antes de adicionar este último nó, assim: as linhas *Must* e *Blocked* tornam-se *Any*, e o valor *Must* é retomado pela linha que o tinha antes de inserir o último filho.

O Caminho exemplificado na Figura 6 e na Tabela 4 auxilia na ilustração do papel do *Stopmask* na busca de propostas. O pacote $P = \langle \langle 0 \ 1 \ 0 \ 0 \ 1 \rangle, \$7, 3 \rangle$ foi obtido a partir do seguinte caminho:

Figura 6 - Exemplo da estrutura da árvore de propostas. Cada caminho é uma oferta. O *Stopmask* indica a direção em que se deve descer pela árvore dependendo das linhas que já foram alocadas.

Tabela 4 - Exemplo da utilização do *Stopmask* para encontrar o caminho que leva ao pacote $P = \{ \langle 0 \ 1 \ 0 \ 0 \ 1 \rangle, \$7, 3 \}$

Linha	Stopmask	Significado
1	<i>Stopmask</i> [1] = <i>Blocked</i>	A linha 1 já foi inserida na árvore BMC, através de outro pacote. Portanto, desce-se pela direita.
2	<i>Stopmask</i> [2] = <i>Must</i>	A linha 2, ainda não foi inserida mas é a próxima porque tem o menor índice. Portanto, desce-se pela esquerda.
3	<i>Stopmask</i> [3] = <i>Blocked</i>	A linha 3 já foi inserida na árvore BMC, através de outro pacote. Portanto, desce-se pela direita.
4	<i>Stopmask</i> [4] = <i>Blocked</i>	A linha 4 já foi inserida na árvore BMC, através de outro pacote. Portanto, desce-se pela direita.
5	<i>Stopmask</i> [5] = <i>Any</i>	A linha 2, ainda não foi inserida mas não tem prioridade ainda. Mesmo assim, desce-se primeiro pela esquerda e se não houver nó se procura pela direita.

3.1.2.3 Iterative Deepening Search A* (IDA*)

A busca principal BCM é implementada através da estratégia *Iterative Deepening Search A* (IDA*)* com ajuda da BPF para adicionar os filhos à árvore. Esta estratégia encontra a solução ótima procurando em frações menores do espaço de busca, um espaço que é linear na busca em profundidade, e evita expandirem nós cujo custo supera o custo encontrado até então.

Utiliza-se um “ponto de corte” ou limitante superior (LS) cujo valor inicial é dado por uma estimativa do custo total e é designado custo da raiz, *IDA**; reforçando o exposto na Seção 3.1.2.1, a busca de Sandholm (2002) utilizava limitante inferior, o qual se modificou para limitante superior a fim construir uma condição que seleccionasse o menor custo total.

Executa-se uma série de buscas em profundidade - *Depth-First-Search (DFS)*. Em cada iteração, um nó é gerado, e é expandido se satisfizer a condição (custo do caminho percorrido até esse nó + o custo estimado das linhas ainda não alocadas $\leq$ LS). A melhor solução é armazenada e comparada com as próximas soluções encontradas. Se a solução ótima for alcançada o algoritmo é finalizado. Em caso contrário, o LS é incrementado no custo do nó que foi gerado, mas não expandido, somado a estimativa do custo das linhas não alocadas, e uma nova iteração começa.

O algoritmo da Figura 7 descreve detalhadamente a aplicação de *IDA** e *DFS* no problema de minimização do custo de compra dos serviços de transporte:

Algoritmo mono-critério: *Iterative Deepening Search A* - IDA** com *Depth First Search (DFS)*.

- A busca começa calculando o custo estimado das linhas, representado pela variável $Cest(E)$. A função utilizada é discutida mais adiante na Seção (função de custo estimado).
- Depois são inicializadas as variáveis, entre as principais estão: o limitante superior LS que no instante inicial é igual à estimativa do custo, $Cest(E)$; e os vetores *Solução* e *Ganhadores* que no instante inicial são iguais a nulo. Em qualquer instante, o vetor *Solução* guardará soluções parciais e o vetor *Ganhadores* guardará a melhor solução encontrada. Assim mesmo, o

custo do caminho, Cc , e a variável que altera o limitante superior, $NovoLS$, são iguais a zero. A variável que guarda o mínimo custo $CustoMinimo$ é igual a infinito; foi estipulado este valor para viabilizar sua comparação inicial com custo do caminho.

- Uma vez inicializadas as variáveis começa a busca em profundidade DFS , cujos parâmetros principais são: as linhas em espera de serem alocadas, $Espera$, as linhas alocadas, $Alocados$, e o custo dos pacotes (nós) alocados, Cc . Cada iteração DFS devolve a melhor solução encontrada até este instante denotada pelas variáveis: $Solucao, Cc$ em que a primeira significa o conjunto de propostas selecionado (nós expandidos) e a segunda o custo das mesmas. A DFS realiza-se da seguinte forma:
 - O algoritmo DFS começa com a inicialização das variáveis. Nesse instante, $Espera = L$ – linhas da proposta, $Alocados = nulo, Cc = 0$
 - Em seguida inicia a construção da árvore BMC (Busca do Mínimo Custo), descendo até completar um caminho, da esquerda para a direita. Para adicionar os nós, o algoritmo compara em cada nó de busca uma estimativa do custo do caminho dada pela função $f = Cc + Cest(E)$ com o limitante superior LS .
 - Se a estimativa de custo for menor que LS , realiza-se a busca dos filhos do nó pela árvore de ofertas (BPF).
 - Se um filho (proposta) for encontrado, atualizam-se as variáveis: $Espera = L$ – linhas da proposta, $Alocados = P(\text{proposta})$, $Cest(E)$, $Cc = Cc + CustoP(\text{custo da proposta})$, se insere a proposta ao vetor $Solucao$ e, começa uma nova DFS . Isso é realizado para cada nó de busca até completar um caminho.
 - Se não for encontrado filho:
 - Se o caminho estiver completo (com todas as linhas no conjunto de propostas escolhidas), uma solução factível foi encontrada. No caso de ser o primeiro caminho, os nós da solução factível e seu valor ($Solucao, Cc$) são os valores iniciais das variáveis $Ganhadorese$ $CustoMinimo$, e $LS = Cc$ para assegurar que as demais

soluções serão melhores que a atual; este valor só será modificado quando uma melhor solução for encontrada.

- Se o caminho não estiver completo (até a folha), isso significa que não foi encontrada uma solução. Realiza-se *backtracking* até encontrar um nó que faça parte de outra ramificação, a partir da qual se começa a construir outro caminho. Se todos os nós foram explorados, finalizar a busca; os nós vencedores estão guardados no vetor **Ganhadores**. Se ainda há nós para expandir, a busca continua.
- Se a estimativa de custo for maior que LS , interrompe-se a visitação e calcula-se o novo limitante. O $NovoLS$ compreende o custo Cc dos nós que foram expandidos mais o $Cest(E)$ das linhas não alocadas.
- Se já foi encontrada uma solução factível ($Ganhadores \neq nulo$), corta-se o caminho que estava sendo visitado, porque este não melhora a solução atual. Realiza-se *backtracking* para continuar a busca.
- Se ainda não existe solução factível, então, deve-se expandir o espaço de busca modificando o ponto de corte, limitante superior, $LS = NovoLS$, liberam-se os nós visitados e começa a *DFS* do instante inicial

Função de custo estimado

O custo estimado corresponde a uma estimativa do custo máximo que podem ter as linhas que ainda não foram alocadas (ver eq.(14)). Tem como propósito servir de limitante superior inicial do espaço de busca. Auxilia também no cálculo do custo potencial de um caminho, fornecendo o valor das linhas que ainda não foram inseridas. Para isso, calcula-se uma contribuição ao custo dada por cada linha.

A contribuição de cada linha é obtida dividindo-se o preço da proposta pelo número de linhas a qual pertence; portanto uma linha pode ter tantas contribuições quanto o número de vezes que aparece nas propostas. Logo, sua contribuição corresponde ao menor valor encontrado.

Esta função foi adaptada da heurística apresentada em Sandholm (2002) para estimar o maior valor dos itens ainda não inseridos; a função adaptada busca encontrar o menor custo estimado das linhas não inseridas.

$$Cest(E) = \sum_{I \in E} c(I),$$

$$\text{em que } c(I) = \text{mínimo} \left[\frac{\text{Custo } P}{\text{número de linhas } P} \forall P / I \in P \right] \quad (14)$$

Proposição 2: O $Cest(E)$ dá um limite superior do custo das linhas não alocadas.

Prova. Para qualquer oferta P que for considerada ganhadora, cada uma de suas linhas contribuirá com uma fração de custo dada pela divisão ($\text{preço } P / \text{número de linhas } P$). Por outro lado, cada item só forma parte de uma oferta ganhadora. Assim, o custo de cada item pode ser no máximo $c(I)$, e sua somatória fornece um limite do custo total.

3.1.2.4 Complexidade das estratégias de busca

3.1.2.4.1 Complexidade da BMC

No pior caso, o tempo de execução da busca principal BMC é polinomial em propostas e exponencial em linhas. Sandholm (20002) demonstrou que o número de nós que compõem a árvore de busca equivalem a $O(L(P/L)^L)$ em que L é o número de linhas e P o número de propostas. Demonstrou também que o tempo gasto executando funções em cada nó de busca é de $O(PL)$. Adicionalmente, analisando o decréscimo do limitante e a regeneração dos nós de busca depois que são cortados, o autor concluiu que na pior das hipóteses a complexidade permanece polinomial em propostas. Devido ao fato das linhas poderem ser controladas pelo comprador, o crescimento exponencial do problema também o é.

3.1.2.4.2 Complexidade da BPF

Por outro lado, a BPF acontece em um tempo menor que a $O(LP)$ que é o tempo que leva para percorrer uma lista de propostas. Graças ao formato em árvore, a complexidade do algoritmo não é maior do que o número de caminhos da árvore de propostas vezes dois (*backtracking*). Considerando que a árvore tem P folhas

(propostas), uma profundidade de $L + 1$ contando a raiz e cada nó tem dois filhos, Sandholm (2002) demonstrou que no pior caso o número de caminhos é $O(P)$. Além disso, a complexidade é ainda menor devido ao corte imposto utilizando o *Stopmask* (ver Seção 3.2.2.2), o qual leva a diminuir a quantidade avaliada de nós para encontrar a solução ótima. Por causa disso o pior caso, aqui descrito, não aconteceu.

Figura 7 - Diagrama de fluxo do algoritmo mono-critério: *Iterative Deepening Search (IDA*)* com *Depth First Search (DFS)*

3.2 ALGORITMO PARA SOLUCIONAR O PROBLEMA WDP MULTICRITÉRIO

A principal modificação é transformar o algoritmo mono-critério de redução de custo em um algoritmo multi-critério de maximização de valor. Vale ressaltar que o algoritmo multi-critério aplica as mesmas estratégias de busca de Sandholm (2002), explicadas na Seção 3.1; porém, o algoritmo base para aplicar estratégias multi-critério será o algoritmo mono-critério adaptado nesta dissertação (ver Seção 3.1).

Com a diferença entre objetivos, as funções e limitantes que condicionam o algoritmo também são modificados. Os parâmetros de entrada e preparação das propostas explicados na Seção 3.1.1 são mantidos. Assim, o formato da proposta multi-critério é composto de: rota, desempenho dos critérios e fornecedor.

Não existe uma unidade comum para designar o desempenho dos critérios. Cada um deles é descrito com sua medida inerente (natural). Por exemplo, o custo é dado em valor monetário (\$), as avarias em percentual (%). A figura 6 mostra a estrutura da proposta que inclui a rota (linha1, linha2, linha3) submetida pelo fornecedor 1 e avalia os critérios de custo (\$950) e avarias (0.5%).

$$P = \{\langle \text{Rota} \rangle, \text{critério1}, \text{critério2} \dots \text{critérioN}, \text{fornecedor}\}$$

$$P_1 = \{\langle 11100 \rangle, \$950, 0.5\%, 1\}$$

Figura 8 - Estrutura da proposta multi-critério

Cabe ao embarcador informar a ponderação que terá cada critério, antes das transportadoras submeterem os pacotes.

Ponderação das linhas de cada proposta

Na prática as linhas diferem umas das outras conforme seu impacto na lucratividade da empresa. Desse modo, os clientes que demandam prioridade no transporte de suas mercadorias são aqueles que contribuem em maior proporção com os lucros.

Esse comportamento influencia a sequência de atendimento das linhas da rede; dá-se prioridade às linhas que servem os clientes mais importantes da empresa.

Esta dissertação considera que todas as linhas têm a mesma importância. Embora, em alguns casos, o pressuposto não seja coerente com a realidade, é necessário para viabilizar os cálculos das funções de valor. Com sua incorporação assume-se um cenário onde todas as linhas possuem um volume similar de mercadorias, portanto contribuem de forma equivalente ao negócio. Cenários diferentes estão fora do escopo do trabalho.

3.2.1 Modificação das estratégias de busca

3.2.1.1 Busca do Máximo Valor (BMV) e Busca do Próximo Filho (BPF)

A valoração das propostas é diferente aqui se busca maximizar valor, portanto quando estes vão sendo adicionados ao caminho, seu valor vai incrementado o Vc , valor do caminho (ver eq.(15)), cujo valor final é a somatória das propostas alocadas nele:

$$Vc = \sum_{P_p \text{ no caminho}} ValorP_p \quad (15)$$

O $ValorP$ de qualquer proposta é calculado através das funções de valor (ver Figura 9). O desempenho mais satisfatório, segundo o decisor, toma o valor de 100 e o menos satisfatório toma o valor de 0. Os pontos intermediários variam conforme a natureza do critério, assim a função pode ser linear, quadrática ou segmentada. As funções são estabelecidas pelo decisor, que nesse caso seria a empresa embarcadora, e assim é construída uma função para cada critério.

Figura 9 - Formato da função de valor

Cada proposta submetida inclui o desempenho de cada critério. Cada desempenho é avaliado na sua função de valor a partir dela e selecionado o valor dado a esse desempenho específico (ver Figura 9). Desse modo, o *ValorP* é a somatória dos valores dos critérios (ver eq.(16)).

$$ValorP_p = \sum_{k \in C} ValorCritério_k \quad (16)$$

Em cada nó de busca é feita uma estimativa do valor do caminho (valor do caminho até esse nó, VC + estimativa do valor das linhas não alocadas, $Vest(E)$), o qual é comparado com o maior valor encontrado até o momento, para determinar se essa é a melhor solução encontrada. Quando o valor estimado do caminho é menor que a melhor solução, esse caminho é cortado. Uma vez explorados todos os caminhos a busca é finalizada, e a solução guardada é a melhor encontrada.

A BPF (ver Seção 3.1.2.2) permanece igual. A árvore de propostas que é utilizada em conjunto com o *Stopmask* é também utilizada para selecionar os próximos filhos. A proposição 1, da seção 3.1.2.1, que explica como devem ser os filhos de cada nó também se aplica para este caso.

3.2.1.2 Iterative Deepening Search A* - IDA*

O ponto de corte da busca é um limitante inferior (LI), cujo valor inicial zero vai incrementando conforme o número de iterações. São executadas buscas em profundidade, onde em cada nó é avaliada a condição (valor do caminho percorrido até esse nó + valor estimado das linhas ainda não alocadas $\leq$ LI), a fim de guardar sempre a melhor solução e reduzir o espaço de busca cortando os caminhos não promissores. O diagrama de fluxo da figura 10 descreve o funcionamento do algoritmo utilizando a busca em profundidade com IDA*:

Algoritmo multi-critério: Iterative Deepening Search A* - IDA* com Depth First Search (DFS)

- A busca começa calculando o valor das propostas, $ValorP_p$, mediante as funções de valor.
- Em seguida calcula o valor estimado das linhas, representado pela variável $Vest(E)$; a função utilizada é discutida mais adiante na Seção (função de valor estimado).
- Depois são inicializadas as variáveis, entre as principais estão: o limitante inferior LI que no instante inicial é igual à estimativa do valor, $Vest(E)$; e os vetores Solução e Ganhadores que no instante inicial são iguais a nulo. O vetor Solução guarda soluções parciais e o vetor Ganhadores guarda a melhor solução encontrada. Assim mesmo, o valor do caminho (nós alocados) Vc e a variável que altera o limitante inferior $NovoLI$ são iguais a zero; enquanto a variável que guarda o máximo valor $ValorMaximo$ é igual a infinito.
- Uma vez inicializadas as variáveis começa a busca em profundidade DFS, cujos parâmetros principais são: as linhas em espera de serem alocadas, $Espera$, as linhas alocadas, $Alocados$, e o valor dos pacotes (nós alocados), Vc . Cada iteração DFS devolve a melhor solução encontrada até este instante denotada pelas variáveis: Solução, Vc onde a primeira significa o conjunto de propostas

selecionado (nós expandidos) e a segunda o valor das mesmas. A *DFS* realiza-se da seguinte forma:

- O algoritmo *DFS* começa com a inicialização das variáveis. Nesse instante, $Espera = L(\text{conjunto de linhas}), Aloca\text{dos} = \text{nulo}, Vc = 0$.
- Em seguida inicia a construção da árvore BMV (Busca do Máximo Valor), descendo até completar um caminho, da esquerda para a direita. Para adicionar os nós, o algoritmo compara em cada nó de busca uma estimativa do valor do caminho dada pela função $f = Vc + Vest(E)$ com o limitante inferior, LI .
- Se a estimativa de valor for maior que LI , realiza-se a busca dos filhos do nó pela árvore de ofertas (BPF).
 - Se um filho (proposta) for encontrado, atualizam-se as variáveis: $Espera = L - \text{linhas contidas na proposta}, Aloca\text{dos} = P(\text{proposta}), Vest(E), Vc = Vc + ValorP(\text{valor da proposta})$, adiciona-se a proposta ao vetor *Solução* e começa uma nova *DFS*. Isso é realizado para cada nó de busca até completar um caminho.
 - Se não for encontrado filho:
 - Se o caminho estiver completo (com todas as linhas no conjunto de propostas escolhidas), uma solução factível foi encontrada. Em caso de ser o primeiro caminho, os nós da solução factível e seu valor (*Solução*, Vc) são os valores iniciais das variáveis *Ganhadores* e *ValorMaximo*, e $LI = Vc$ para assegurar que as demais soluções sejam melhores que a atual, esse valor só será modificado quando uma melhor solução for encontrada.
 - Se o caminho não estiver completo (até a folha), significa que não foi encontrada uma solução. Realiza-se *backtracking* até encontrar um nó que faça parte de outra ramificação, a partir da qual se começa a construir outro caminho. Se todos os nós foram explorados pode-se finalizar a busca. Os nós vencedores estão guardados na variável *Ganhadores*. Se ainda há nós para expandir, a busca continua.

- Se a estimativa do valor for menor que LI , interrompe-se a visitação e calcula-se o novo limitante. O $NovoLI$ compreende o valor VC dos nós que foram expandidos mais o $Vest(E)$ das linhas não alocadas nesses nós.
- Se já foi encontrada uma solução factível ($Ganhadores \neq nulo$) corta-se o caminho que estava sendo visitado, porque esse não melhora a solução atual. Realiza-se *backtracking* para continuar a busca.
- Se ainda não existe solução factível, deve-se expandir o espaço de busca modificando o ponto de corte, limitante inferior, $LI = NovoLI$; liberam-se os nós visitados e começa a *DFS* do instante inicial.

Função de valor estimado

O valor estimado corresponde a uma estimativa do valor máximo das linhas que ainda não foram alocadas. Tem como propósito servir de limitante inferior inicial do espaço de busca e auxiliar no cálculo da contribuição o valor do caminho, que podem fazer as linhas não alocadas. Para isso, calcula-se a máxima contribuição ao valor dada por cada linha.

A contribuição de cada linha é obtida dividindo o preço da oferta a qual pertence, pelo número de linhas que a compõe, portanto, uma linha pode ter tantas contribuições quanto o número de vezes que aparece nas propostas. Logo, sua máxima contribuição corresponde ao maior valor encontrado.

$$Vest(E) = \sum_{I \in E} V(I),$$

$$\text{onde } V(I) = \max \left[\frac{ValorP}{\text{número de linhas } P} \mid \forall P \mid I \in P \right] \quad (17)$$

Figura 10 - Diagrama de fluxo do Algoritmo multi-critério: *Iterative Deepening Search (IDA*)* com *Depth First Search (DFS)*

3.3 CONCLUSÃO DA SEÇÃO 3

Nesta seção foram descritos os algoritmos adaptados do algoritmo de Sandholm (2002). O primeiro algoritmo adaptado foi mono-critério, cujo objetivo principal é minimizar o custo total de transporte do embarcador. Em seguida, esse algoritmo foi modificado para tratar múltiplos critérios onde o objetivo principal é maximizar o valor do embarcador.

As principais diferenças a respeito do algoritmo de Sandholm (2002) residem em que o algoritmo foi adaptado e aplicado ao problema de compras de serviços de transporte; permite inclusão de vários fornecedores e todas as propostas são avaliadas, visto que não há pré-processamento.

A estratégia utilizada foi *IDA** com busca em profundidade. O ponto de corte utilizado é calculado mediante uma heurística que estima o melhor valor a ser obtido em cada nó, e condiciona o avanço da busca a essa estimativa até encontrar a solução inicial. Depois se continua com busca em profundidade comparando as soluções encontradas com a melhor armazenada, simultaneamente, são atualizados variáveis e vetores. No final, quando foram explorados todos os nós fatíveis, a melhor solução encontrada é aquela armazenada.

As buscas utilizadas são BMC e BPF, a primeira refere-se ao processo de alocar as propostas atendendo a condição de corte; a outra permite selecionar de forma mais eficiente as propostas a serem inseridas na árvore.

O algoritmo multi-critério incorpora uma estratégia adicional para calcular os valores das propostas. Esta estratégia trata da inserção de funções de valor para avaliar o desempenho das propostas e designar uma valoração.

Ambos os algoritmos são implementados e validados na Seção 4.

4 VALIDAÇÃO E RESULTADOS DOS ALGORITMOS PROPOSTOS

Esta seção apresenta a geração de instâncias e validação de resultados do algoritmo. A Seção 4.1 explica a geração de instâncias do algoritmo mono-critério e analisa os resultados obtidos a partir dessas instâncias. A seção 4.2 descreve como são complementadas as instâncias para analisar múltiplos critérios, são também selecionados os critérios do teste, descrevem-se suas funções de valor e realiza-se uma análise de sensibilidade.

Para determinar a eficiência dos algoritmos na prática, foram realizados experimentos em um computador com processador E7500 2,93GHz e 2G de RAM. Para sua implementação foi utilizada a linguagem de programação *Java*, cuja *Virtual Machine* - VM foi configurada para utilizar 1G de memória e 64M de pilha.

As instâncias de teste para ambos os algoritmos (mono-critério e multi-critério) foram geradas a partir da classificação das transportadoras. Categorizam-se as transportadoras em transportadoras do tipo A, B e C em relação ao nível de serviço. As transportadoras do tipo A são consideradas muito boas, as do tipo B enquadram-se como transportadoras boas e as do tipo C são as regulares. Para fazer essa classificação são estabelecidos intervalos de variação (Tabela 5) que são aplicados aos valores relacionados às transportadoras. Com esses intervalos consegue-se criar os *trade-offs* entre os critérios que estão sendo avaliados. Cabe notar que as transportadoras são classificadas de forma aleatória, mas procura-se manter quantidades proporcionais entre os diferentes tipos. Na prática, a classificação é elaborada de forma qualitativa; utiliza-se informação fornecida pelas transportadoras, as quais devem garantir sua confiabilidade. Nesses casos existe a possibilidade de incluir uma multa contratual referente ao cumprimento dos índices de desempenho informados (PRADO, 2011).

Tabela 5 - Fator de variação do fornecedor

Classificação do fornecedor	Fator de variação do fornecedor
A	1.00 – 0.90
B	0.90 - 0.80
C	0.80 - 0.70

4.1 ALGORITMO MONO-CRITÉRIO

4.1.1 Geração de instâncias

As instâncias resolvidas a fim de validar o algoritmo apresentam os seguintes parâmetros: o número de linhas disponibilizadas pelo comprador, o número de propostas submetidas, o número de fornecedores participantes do leilão e detalham as propostas submetidas por cada fornecedor, estas últimas geradas da seguinte forma:

4.1.1.1 Rota

Para sua configuração cada linha tem a mesma probabilidade (0,5) de pertencer à rota. Se uma linha faz parte da rota esta é representada com o valor 1, em caso contrário toma o valor 0. Cada rota pertence a uma solução factível do problema, quer dizer, sempre há rotas complementares entre si. Devido à existência de vários fornecedores há necessidade de dividir as rotas em comuns e particulares: as primeiras incluem aquelas rotas ofertadas várias vezes, cada uma é submetida pelo menos por 20% dos fornecedores, e representam as rotas que conectam os pontos principais acessíveis para qualquer transportadora. As outras correspondem às rotas próprias de cada fornecedor.

4.1.1.2 Custo do pacote

Sendo esse o valor que um fornecedor está disposto a receber para cobrir determinada rota, sua geração está diretamente relacionada com o custo de transporte de cada rota e com o nível da transportadora (qualidade da transportadora) (ver Seção 4.1.2.1).

Cálculo dos custos

Antes de entrar em detalhe com o cálculo dos custos, é importante especificar dois parâmetros que serão utilizados: Distância da oferta e o Fator de variação do fornecedor.

Considera-se uma distância média para cada linha leiloadada, desta forma, a distância da rota é calculada multiplicando a distância média pelo número de pontos atendidos na rota.

O Fator de variação estipulado para transportadoras do tipo A, B e C respectivamente pode ser observado na Tabela 5. Cada fator indica que o custo das propostas submetidas por um fornecedor tipo A pode diminuir até 10% do valor original, se for um fornecedor tipo B o custo diminui num intervalo de 10% e 20% e para um fornecedor tipo C o custo pode ser entre 20% e 30% menor. Além disso, como o número de fornecedores não forma parte das restrições, a decisão de diferenciá-los tem o propósito de manter uma coerência entre os custos e o tipo de transportadora. Mas, para o caso mono-critério, é óbvio que as melhores instâncias pertencerão aos fornecedores que se encontrem na classificação C.

Considerando os parâmetros anteriores, calculam-se os custos de transporte para cada rota. A fim de contabilizar esses custos em função da distância percorrida em cada rota, os mesmos são divididos arbitrariamente em custos fixos e custos variáveis.

Os custos fixos compreendem principalmente aos custos não relacionados a movimentação de caminhões, tais como, custos de aquisição e manutenção do direito de tráfego, instalações de terminais, equipamentos de transporte e administrativos. Os custos variáveis referem-se aos custos de movimentação, entre eles, os gastos com combustível, salários dos motoristas, equipamentos de manutenção, manuseio, coleta e entrega (BALLOU, 2006).

Neste caso, o custo fixo (CF) mensal e o custo variável (CV) foram gerados de forma aleatória. Com esses valores e supondo uma velocidade de 50 km/hora, uma jornada de trabalho de 26 dias/mês e 12 horas/dia, são construídos os custos fixos (eq.(18)) e variáveis (eq.(19)) de cada oferta.

$$CF_{pacote} = \frac{CF(\$ / \text{mês})}{\text{Horas trabalhadas ao mês}(h / \text{mês})} \left(\frac{\text{DistanciaPacote}(km)}{50(km/h)} \right) \quad (18)$$

$$CV_{pacote} = CV(\$ / km) * \text{DistanciaPacote}(km) \quad (19)$$

$$CustoPacote = (CV_pacote + CF_pacote) * Fator de Variação * complementaridade \quad (20)$$

Para calcular o custo do pacote conforme a eq.(20), multiplica-se a somatória do custo fixo e do custo variável pelo fator de variação da transportadora. Além disso, para manter a complementaridade, evitando que os custos de uma oferta sejam iguais à somatória dos custos individuais de cada linha, aplica-se um desconto de 5% a cada oferta que inclua duas ou mais linhas.

4.1.2 Resultados

Com o intuito de validar a otimalidade do algoritmo para o problema proposto, foram geradas instâncias de tamanhos variáveis com respeito ao número de propostas submetidas e número de linhas participantes do leilão (ver tabela 6). O número de fornecedores manteve-se fixo, já que o problema não possui restrições com respeito ao número máximo ou mínimo de fornecedores que podem vencer o leilão. Isso significa que todo fornecedor que submete uma oferta pode ser um vencedor. A combinação entre tamanhos de propostas e tamanhos de linhas formou 16 cenários de teste e para cada cenário foram geradas 5 instâncias; assim, no total, foram testadas 80 instâncias.

Tabela 6 - Tamanho das instâncias com 15 fornecedores

Nº Propostas	Nº linhas
500	100
1000	200
1500	300
2000	400

Os resultados do algoritmo foram comparados com os resultados do modelo matemático executado em GAMS utilizando o solver CPLEX 10.2 (ver tabela 7).

Tabela 7 - Resultados das instâncias executadas para validar o algoritmo

Cenário	Instância	Solução ótima (R\$)	Algoritmo	GAMS/CPLEX 10.2
			Tempo de execução (seg.)	Tempo de execução (seg.)
500_100_15	1	15895	0,266	0,340
	2	16487	0,132	0,227
	3	16323	0,127	0,242
	4	16480	0,153	0,306
	5	16853	0,147	0,202
500_200_15	1	33259	0,358	0,245
	2	36152	0,440	0,308
	3	33004	0,357	0,307
	4	32819	0,334	0,317
	5	32871	0,342	0,314
500_300_15	1	48692	0,617	0,315
	2	49967	0,642	0,313
	3	49573	0,643	0,320
	4	50185	0,601	0,326
	5	49285	0,586	0,307
500_400_15	1	65901	0,928	0,405
	2	65359	0,916	0,364
	3	68384	0,974	0,397
	4	66649	0,880	0,404
	5	65960	0,855	0,404
1000_100_15	1	16273	0,648	0,310
	2	18312	5,345	0,657
	3	16648	3,589	0,428
	4	17017	2,928	0,435
	5	17051	2,228	0,347
1000_200_15	1	33083	1,544	0,413
	2	34711	1,441	0,420
	3	33785	1,405	0,420
	4	32803	1,498	0,433
	5	33101	1,556	0,487
1000_300_15	1	48725	2,273	0,486
	2	52213	2,295	0,441
	3	50464	2,214	0,440
	4	50580	2,293	0,548
	5	49346	2,417	0,490
1000_400_15	1	67393	3,004	0,599
	2	65926	3,311	0,635
	3	64728	3,510	0,842
	4	68046	3,402	0,709
	5	65614	3,336	0,613

Continua

1500_100_15	1	17553	4,368	1,361
	2	17020	5,507	6,653
	3	17886	5,041	1,290
	4	17058	2,798	0,523
	5	16094	3,055	0,413
1500_200_15	1	33850	3,482	0,530
	2	32736	3,152	0,531
	3	33879	3,526	0,807
	4	35073	7,678	0,588
	5	32919	3,505	0,576
1500_300_15	1	48230	5,036	0,651
	2	49388	5,052	0,678
	3	49704	5,354	0,738
	4	48680	5,372	0,740
	5	49322	5,389	0,689
1500_400_15	1	68652	7,843	1,696
	2	64840	7,089	0,909
	3	64829	7,829	1,057
	4	65293	7,453	0,872
	5	67393	6,807	0,815
2000_100_15	1	17337	9,977	1,164
	2	16280	8,545	0,643
	3	16938	9,053	1,200
	4	16572	8,835	1,026
	5	16800	10,399	0,952
2000_200_15	1	32390	5,689	0,717
	2	32065	6,093	0,678
	3	32424	6,105	0,668
	4	32903	5,516	0,725
	5	34443	6,542	0,867
2000_300_15	1	50485	9,589	0,874
	2	49201	10,775	1,389
	3	49593	8,698	0,851
	4	51544	10,388	1,059
	5	51501	7,557	0,830
2000_400_15	1	68231	12,995	1,375
	2	65343	13,396	1,297
	3	64740	12,964	1,215
	4	67538	12,589	1,231
	5	65013	12,090	1,095

Na tabela 7, o cenário está representado da seguinte forma: propostas_ linhas_ fornecedores; para cada cenário foram geradas cinco instâncias diferentes.

A comparação da tabela 6 mostra que o custo obtido pelo CPLEX e o custo obtido pelo algoritmo foi o mesmo em todas as instâncias avaliadas. Este resultado confirma que o algoritmo consegue chegar ao custo total ótimo dos problemas de compras de serviços de transporte.

A respeito dos tempos de execução, pode-se observar que o algoritmo apresenta um desempenho superior ao CPLEX no primeiro cenário, já no segundo cenário, ambos os tempos são similares. A partir do terceiro cenário o algoritmo tem maior dificuldade para encontrar a solução ótima se comparado com o CPLEX; isso leva a tempos mais elevados.

A diferença nos tempos de execução pode ser atribuída às técnicas utilizadas pelo software comercial para resolver o problema. O CPLEX está projetado para encontrar soluções ótimas rapidamente. Em primeiro lugar analisa o problema visando diminuir seu tamanho eliminando redundâncias. Depois aplica técnicas de *branch and cut*, desempenhadas de forma que sejam encontradas soluções factíveis sem sacrificar muito tempo de processamento, o qual poderia ser utilizado testando a otimalidade da solução encontrada (ILOG S.A, 2006).

Algumas instâncias também pertencentes ao mesmo cenário apresentam grandes diferenças no tempo de execução. Isso demonstra que a estrutura da árvore formada a partir dos pacotes influencia a busca da solução e consequentemente afeta o desempenho do algoritmo. Por exemplo, o cenário 1000_100_15 possui 5 instâncias diferentes (diferentes estruturas da árvore de propostas), cujo efeito reflete-se nos tempos de execução, onde pode ser observada maior dificuldade do algoritmo em obter algumas soluções.

Ao calcular os tempos médios de execução do algoritmo (ver Gráfico 1) para cada cenário, constatou-se um comportamento polinomial conforme aumenta o número de propostas. Cada curva do Gráfico 1 representa a variação do tempo médio dos cenários que possuem a mesma quantidade de linhas. Observa-se um comportamento com tendência polinomial de ordem 2 nas curvas de 400 e 300 linhas. A curva de 200 linhas apresenta um comportamento semelhante ao linear. Já a curva de 100 linhas não é passível de ajuste. Seus tempos médios inferem que houve maior dificuldade para encontrar a solução ótima quando comparados com cenários que têm a mesma quantidade de propostas e diferentes quantidades de linhas.

Gráfico 1 - Tempo médio de execução de cada instância vs. número de propostas. Cada curva representa o comportamento do algoritmo com determinado número de linhas e sofrendo variação no número de propostas.

O comportamento polinomial deve-se ao fato do algoritmo não explorar todas as possíveis combinações que formariam uma solução. Devido à condição de visitação imposta, quando o custo aproximado supera o limitante superior sabe-se que a próxima solução encontrada não vai superar a atual solução, portanto a exploração é interrompida e redirecionada. Com a diminuição do espaço de busca os tempos de execução são menores.

4.2 ALGORITMO MULTI-CRITÉRIO

4.2.1 Geração de instâncias

As instâncias utilizadas na avaliação do algoritmo mono-critério são adaptadas para utilização no algoritmo multi-critério, a fim de possibilitar as comparações e análises entre ambos os algoritmos. Dado que as instâncias mono-critério contêm só as propostas, os custos das propostas e os fornecedores; deve-se adicionar os parâmetros dos critérios a serem avaliados em conjunto com o custo da proposta.

Por outro lado, o algoritmo também precisa dos requerimentos do cliente (comprador) a respeito das funções de valor que são utilizadas durante a avaliação de cada critério e o peso (importância relativa) dado a cada critério.

Esta seção apresenta quais critérios além do custo são utilizados e qual a metodologia utilizada para seu cálculo (seção 4.2.1.1). Em seguida, mostra-se o modelo matemático multi-critério, específico para os critérios selecionados (seção 4.2.1.2). Depois, são designados os pesos de cada critério (seção 4.2.1.3). Finalmente, explica-se a construção das funções de valor (seção 4.2.1.4).

4.2.1.1 Critérios selecionados

Custo e Nível de serviço

Os critérios utilizados neste algoritmo foram selecionados levando em conta a revisão da literatura realizada na Seção 2. Foi comprovado que o *trade-off* mais relevante é o custo de transporte vs. o nível de serviço; portanto, ambos critérios são avaliados.

Por se tratarem de propostas combinatórias o custo é calculado por pacote e não por linha. Calcula-se o custo do pacote conforme descrito na Seção 4.1.1.2.

O nível de serviço depende unicamente da transportadora e para mensurá-lo foram selecionados dois critérios que os embarcadores consideram relevantes; o tempo de entrega (*lead time*) e as avarias. (PRADO, 2011):

- O lead time é o prazo acordado contratualmente em dias úteis para que a mercadoria seja efetivamente entregue. A satisfação dos clientes é inversamente proporcional ao prazo definido e um maior *lead time* pode obrigar a contratante a ter que alterar os seus processos para se adequar às necessidades de seus clientes. Neste caso, o *lead time* é avaliado por linha já que cada linha possui características e prioridades de entrega diferentes.

Para calcular o *lead time* de um pacote atribuem-se limitantes inferiores e superiores a cada linha, indicando o tempo mínimo e máximo de entrega nessa linha. Em seguida o intervalo formado pelos limitantes é dividido em três partes iguais, a primeira parte corresponde ao intervalo onde são gerados os valores da transportadora A (menor tempo de entrega), a segunda parte é o intervalo para

os valores da transportadora B (tempo médio de entrega) e a última parte, o intervalo para os valores da transportadora C (maior tempo de entrega). Em seguida, para cada linha do pacote, gera-se aleatoriamente um valor que esteja dentro do intervalo correspondente da linha e da transportadora. Finalmente a somatória desses valores (dias) indica o *lead time* do pacote. Assim, esse processo é feito para todos os pacotes. Com este formato, busca-se manter a relação inversa entre o custo e o nível de serviço.

- As avarias são o percentual de cargas danificadas ou extraviadas durante o processo. Embora esse índice seja integralmente indenizado pelas transportadoras, existem dezenas de aspectos envolvidos em uma não efetivação da entrega: atraso para o cliente, necessidade de reposição (quando possível), prazo de retorno, custos financeiros decorrentes do atraso, etc. Embora o cálculo das avarias dependa do tipo de cliente, ou seja, o faturamento que este representa para a transportadora, neste trabalho não é abordado esse nível de detalhe. As avarias são então diferenciadas por transportadora e não por pacote, ou seja, uma transportadora terá o mesmo percentual de avarias em todos os seus pacotes.

Para calcular as avarias de uma transportadora, eq. (21), seleciona-se um valor médio de avarias utilizando os dados encontrados em Prado (2011). Em seguida, aplica-se o fator de variação dependendo da transportadora conforme a tabela 4. Sendo assim, as transportadoras do tipo A aumentam o valor original em até 10,0%, as do tipo B aumentam entre 10,1% e 20,0% e as do tipo C entre 20,1% e 30%. Isso significa que o menor percentual de avarias ocorre para a transportadora tipo A, quer dizer, muito boa em relação ao nível de serviço.

$$AvariaTransportadora = AvariaMédia * Fator de Variação \quad (21)$$

Emissão de CO₂

Um terceiro critério é a emissão de dióxido de carbono (CO₂), cuja importância na indústria é cada vez maior por causa das políticas de sustentabilidade das empresas que buscam reduzir os gases do efeito estufa (GEE).

O CO₂ foi escolhido porque é o principal GEE produzido pelo setor de transporte (GREENE e SCHAFER, 2003). A respeito do transporte rodoviário, uma recente

pesquisa do instituto de Logística e Supply Chain (ILOS) revelou que esse meio de transporte tem o maior volume de emissões de gás carbônico na atmosfera (88,6%) e responsável também pela maior quantidade de carga movimentada no Brasil (62,8%).

No âmbito internacional, várias regulamentações foram desenvolvidas com o objetivo de estabilizar a concentração dos gases de efeito estufa na atmosfera. Uma das mais importantes foi um acordo estabelecido na “Convenção do Clima” realizada em Kyoto em 1997, conhecido como Protocolo de Kyoto. Esse acordo determina que entre o período de (2008-2012) os países industrializados deverão reduzir suas emissões de GEE em pelo menos 5.2% em relação aos níveis de 1990 (ONU, 1998).

Embora o Brasil não possua nenhuma meta de redução dentro do acordo, tem demonstrado comprometimento com as reduções de GEE. Assim o confirma o estudo do ILOS, cujo foco principal é mostrar como as empresas brasileiras estão agindo para reduzir o impacto de seus negócios no meio ambiente e nas atividades logísticas. Essa pesquisa envolveu 109 empresas brasileiras, entre as mil com maior faturamento do país, de 14 setores diferentes, das quais 60% realizam inventário para mapear emissões GEE. A maioria (72%) apresenta iniciativas que buscam minimizar os efeitos ambientais das atividades no setor de logística. Apenas uma em cada cinco delas não pensa em ter áreas dedicadas à sustentabilidade. Assim, a logística aparece como a terceira área mais envolvida (76%) em ações voltadas para a sustentabilidade ambiental dentro das corporações, ficando atrás do setor de compras e suprimento (82%) e de produção (77%) (REVISTA FATOR, 2011).

Neste trabalho, calcula-se a emissão de CO₂ por pacote. O valor de referência utilizado para calcular as emissões (0,43kgCO₂e/km) foi obtido através da ferramenta de cálculo de GEE disponibilizada pela *GHG Protocol* ([HTTP://www.ghg.protocol.org/calculation-tools/all-tools](http://www.ghg.protocol.org/calculation-tools/all-tools)) sob o nome de “CO₂ emissions for transport”.

Segundo o *GHG Protocol*, a emissão de CO₂ pode ser calculada baseando-se no consumo de combustível ou na distância percorrida. Apesar de a primeira ser apontada como uma metodologia mais precisa, as informações necessárias são muito específicas (tipos de veículos) e este trabalho não pretende esse nível de detalhe. Por causa disso, a emissão é calculada com base na distância percorrida pelos veículos.

O cálculo da emissão por pacote é feito da seguinte forma: multiplica-se a emissão (0,43kgCO₂e/km) pela distância percorrida na rota do pacote, conforme a eq. (22). Ao valor obtido, aplica-se o fator de variação da Tabela 4. Desta forma, a emissão da transportadora tipo A será menor do que a emissão da transportadora tipo B; a transportadora tipo C tem a maior emissão.

$$EmissaoPacote(p) = 0.43(kgCO_2e / km) * DistanciaPacote(km/pacote) * FatordeVariação \quad (22)$$

4.2.1.2 Modelo matemático de otimização multi-critério

O modelo matemático apresentado na Seção 2.4.3 foi modificado para inserir os critérios selecionados (Seção 4.2.1) visando avaliar os pacotes submetidos pelas transportadoras. Assim, o modelo matemático combinatório para avaliar o custo, nível de serviço e emissão de CO₂ é o seguinte:

Parâmetros:

P = Conjunto de propostas submetidas pelas transportadoras

T = Conjunto de transportadoras que participam do processo de compra

L = Conjunto de linhas que compõem a rede do embarcador

C = Conjunto de critérios que o embarcador quer avaliar

p = Proposta (pacote), $p \in P$

t = Transportadora, $t \in T$

l = Linha, $l \in L$

W_c = Peso do custo

W_{NS} = Peso do nível de serviço

W_{EC} = Peso da emissão de carbono

a_{pt}^l = Vetor binário que toma o valor de 1 se o pacote p do fornecedor t tem a linha

l e, toma o valor 0 em caso contrario

V_c^{pt} = Valor da função de custo para o pacote p do fornecedor t

V_{NS}^{pt} = Valor da função de nível de serviço para o pacote p do fornecedor t

V_{EC}^{pt} = Valor da função de emissão de carbono para o pacote p do fornecedor t

$\alpha_{pt} = \left(\frac{n^\circ \text{linhas}_{pt}}{n^\circ \text{linhas totais}} \right)$ Fator de cobertura do pacote p do fornecedor t

Variável decisão:

X_{pt} = Recebe o valor de 1 se o pacote p do fornecedor t é vencedor do leilão, 0 em caso contrario

Função objetivo:

$$\text{Maximizar o Valor Total} = \sum_{p \in P} \sum_{t \in T} \alpha_{pt} x_{pt} [w_C V_C^{pt} + w_{NS} V_{NS}^{pt} + w_{EC} V_{EC}^{pt}] \quad (23)$$

Restrições:

Cobertura da rede:
$$\sum_{p \in P} \sum_{t \in T} a'_{pt} x_{pt} = 1 \quad \forall l \in L, \quad (24)$$

Binária:
$$x_{pt} \in \{0,1\} \quad \forall p \in P, \forall t \in T \quad (25)$$

Normalização:
$$\sum_{c \in C} w_c = 1 \quad (26)$$

Valores:

$$V_{NS}^{pt} = w_{LT} * \frac{\sum_{l \in pt} V_{LT}^{ptl}}{n^\circ \text{linhas}_{pt}} + w_{AV} * V_{AV}^{pt}, \quad \forall p, \forall t \quad (27)$$

$$V_{LT}^{ptl} = F(LT), \quad \forall p, \forall t, \forall l \quad (28)$$

$$V_{AV}^{pt} = F(AV), \quad \forall p, \forall t \quad (29)$$

$$V_{EC}^{pt} = F(EC), \quad \forall p, \forall t \quad (30)$$

A valoração dos critérios na função objetivo é realizada conforme explicado na Seção 2.4.3 utilizando a função de valor, eq. (23). A maximização do valor total deve garantir cobertura de todas as linhas, eq.(24), e considerar as restrições de binariedade, eq.(25), e normalização, eq.(26), conforme explicado na Seção 2.4.3.

Os valores de cada critério (ver eq.(27) à eq.(30) são obtidos a partir da sua função de valor.

Visando manter a mesma ponderação para todas as linhas foi elaborado um fator de cobertura por pacote, α_{pt} . Esse fator divide a quantidade de linhas do pacote entre o total de linhas, indicando que não existe priorização.

4.2.1.3 Definição dos pesos de cada critério

Para tomar essa decisão são considerados dois aspectos: em primeiro lugar o algoritmo não está sendo aplicado a um caso real, portanto, não é possível estabelecer uma ponderação de acordo com características específicas do negócio, como é feito na prática. Em segundo lugar, cada critério parece muito importante para os embarcadores que utilizam o transporte rodoviário para distribuir suas mercadorias, embora o *trade-off* entre o custo e o nível de serviço seja mais mencionado na literatura, a emissão de CO₂ é também um fator a ser controlado dada a atual preocupação das empresas de se tornarem sustentáveis. Com essas considerações são designadas ponderações iguais para cada critério (33,33%).

A respeito do nível de serviço, Prado (2011) na avaliação de seus sub-critérios dá uma das ponderações mais altas para o *lead time*. Nesse caso, seguindo o raciocínio do autor o *lead time* tem um peso de 59%, e em complemento as avarias ficam com um peso de 41%.

A ponderação é apresentada na árvore de valores da Figura 11:

Figura 11 - Árvore de valores da seleção de transportadoras

4.2.1.4 Funções de valor

Custo

Foi decidido de forma arbitrária que o custo do pacote será avaliado através de uma função segmentada (ver Gráfico 2), onde sua abscissa representa o fator de custo da eq.(32) e sua ordenada o valor equivalente ao fator.

$$CustoMedioLinha = \frac{CustoOtimo(\$)}{\text{total de linhas}} \quad (31)$$

$$FatorCusto = \frac{CustoPacote(l)(\$ / pacote)}{CustoMedioLinha(\$ / linha) * n^{\circ} \text{ linhas do pacote}}, \forall l \quad (32)$$

O *CustoMedioLinha* (ver eq.(31)) multiplicado pelas linhas contidas no pacote é o custo médio do pacote (l). O fator custo avalia quão bom é o custo de um pacote em relação ao custo médio proveniente do custo ótimo. Dado que esse fator é uma proporção aproximada, o melhor caso (valor 100) não é quando seja igual a 1, já que o custo do pacote poderia superar o custo médio. Então, optou-se por dar o valor de 100 para um fator cujo custo do pacote seja a metade do custo médio (0,5). O Valor de 0 significa que o custo do pacote é duas vezes maior do que o custo médio (2). A

reta formada pelos valores intermediários (1 e 1,4) significa que entre esses pontos a valoração é mais sensível a mudança do fator de custo.

Valor	Fator de Custo
0	2
25	1,4
75	1
100	0,5

Gráfico 2 - Função de valor do custo

Avarias:

Este critério é avaliado levando em conta a análise de Prado (2011). A função (ver Gráfico 3) constrói-se atribuindo 100, maior pontuação, para o menor percentual de avarias recebido e 0, menor pontuação, para o maior percentual recebido. O eixo horizontal é dividido em cinco partes, em cada uma dessas partes aumenta-se a penalização em 10%, 15%, 20%, 25% e 30% respectivamente, por considerar que quanto maior o percentual de avarias menor a valoração, porque aumentam os problemas no relacionamento com os clientes e os custos não mensuráveis como multas por não recebimento.

Valor	Avarias (%)
0	4
30	3,35
55	2,7
75	2,05
90	1,4
100	0,75

Gráfico 3 - Função de valor das avarias

Lead time

Conforme Prado (2011) este critério é representado através de uma função linear, uma vez que é possível adequar os processos internos para atender uma variação no *lead time*. Seguindo esse raciocínio, a função (Gráfico 4) é construída considerando o menor e maior tempo de entrega atribuído a cada linha através das ofertas. Dado que o *lead time* é por linha, a quantidade de funções de valor é igual ao total de linhas sendo leiloadas.

Gráfico 4 - Função de valor do *lead time*. Este gráfico exemplifica uma linha com *lead time* permitido de 1 a 6 dias, assim, ao menor *lead time* é dada uma valoração de 100 e ao maior de 0, quer dizer, o valor diminui na medida em que aumenta o *lead time*.

Emissão de CO₂

O critério é avaliado utilizando uma função linear (ver Gráfico 5) porque na mesma proporção que diminui a emissão aumenta a pontuação do pacote. O cálculo do fator de emissão considera o pior caso do leilão em relação a esse critério. A situação que maximiza as emissões de CO₂ é uma transportadora atender cada linha retornando vazia ao ponto de origem, sem sinergia entre linhas. Nesse caso, as emissões totais podem ser calculadas multiplicando a emissão de CO₂ pela somatória do dobro da distância de todas as linhas conforme a eq.(33). A partir delas calcula-se a emissão média por linha, eq.(34). Finalmente, calcula-se o fator de emissão dividindo a emissão de CO₂ por pacote pela emissão média por pacote, eq.(35). A emissão por pacote é o valor submetido pela transportadora para esse critério.

$$EmissoesTotais(pior\ caso) = 0,43(kgCO_2e/km) \sum_{k \in I} [DistLinha_k(km) * 2] \quad (33)$$

$$EmissaoMedaLinha = \frac{EmissoesTotais(kgCO_2e)}{\text{total de linhas}} \quad (34)$$

$$FatorEmissao = \frac{EmissaoPacote(I)(kgCO_2e)}{EmissaoMedaLinha(kgCO_2e) * n^oLinhasPacote(I)} \quad (35)$$

Gráfico 5 - Função de valor da emissão de CO

4.2.2 Resultados

Conforme exposto na seção 4.2.1, as instâncias utilizadas para validar o algoritmo são formadas a partir das utilizadas na validação do algoritmo mono-critério, dessa forma são mantidas as mesmas propostas com seu respectivo custo. Os parâmetros adicionais de cada proposta são: o percentual de avarias e o *lead time* que avaliam o nível de serviço e a emissão de CO₂. Nesse caso, os pesos são os mesmos para cada critério, 33,33%, (ver Figura 14) e os parâmetros que constroem as funções de valor são especificados na Seção 4.2.1.4. Os cenários criados a partir da variação no número de propostas e número de linhas podem ser observados na Tabela 5. No total são testadas 80 instâncias.

As instâncias são executadas tanto no algoritmo quanto no GAMS/CPLEX 10.2. A informação extraída de cada execução é: o custo total da instância, o tempo de execução e o valor máximo. Este último mede a porção em que os requerimentos (critérios) são atendidos pela a melhor combinação (solução), considerando 100 como o máximo valor que se pode obter (combinação ideal). Os resultados são apresentados na Tabela 8.

Tabela 8 – Resultados da execução das instâncias no algoritmo multi-critério e o GAMS/CPLEX 10.2

Cenário	Instância	Custo (R\$)	Algoritmo		GAMS/CPLEX 10.2	
			Valor máximo (R\$)	Tempo de execução (seg.)	Valor máximo (R\$)	Tempo de execução (seg.)
500_100_15	1	16040	65,642	0,110	65,708	0,314
	2	17604	66,542	0,142	66,609	0,371
	3	16631	65,542	0,136	65,674	0,432
	4	16553	65,689	0,151	65,755	0,394
	5	19911	64,761	0,175	64,826	0,476
500_200_15	1	37960	65,228	0,399	65,293	0,440
	2	38226	68,482	0,375	68,551	0,446
	3	33004	65,047	0,411	65,112	0,443
	4	33710	64,065	0,380	64,128	0,477
	5	32941	64,919	0,362	64,984	0,456
500_300_15	1	49003	64,180	0,571	64,244	0,577
	2	61354	64,282	0,658	64,347	0,578
	3	49573	64,236	0,656	64,299	0,572
	4	52332	65,031	0,637	65,095	0,573
	5	49302	64,153	0,667	64,217	0,576
500_400_15	1	72518	65,112	0,843	65,177	0,709
	2	66716	64,312	0,789	64,376	0,946
	3	73241	65,981	1,019	66,047	0,721
	4	70856	64,411	0,925	64,475	0,676
	5	66152	64,872	0,761	64,936	0,680
1000_100_15	1	16273	64,745	0,969	64,809	0,993
	2	19239	70,278	3,836	74,302	0,789
	3	17934	66,602	5,672	66,613	0,887
	4	19067	66,006	3,942	66,132	0,711
	5	18908	67,669	10,777	67,779	0,748
1000_200_15	1	33217	64,583	1,762	64,648	0,755
	2	37085	66,574	1,936	66,641	0,783
	3	40703	65,831	1,965	65,897	1,712
	4	39183	64,299	1,699	64,363	0,716

Continua

	5	35095	65,087	1,904	65,152	0,822
1000_300_15	1	48725	64,598	2,621	64,662	0,995
	2	61424	64,982	2,829	65,735	0,935
	3	57669	64,709	2,404	64,774	0,958
	4	53949	65,923	2,700	65,990	1,036
	5	51468	63,927	2,540	63,990	0,986
1000_400_15	1	69354	64,679	3,982	64,744	1,310
	2	68507	65,071	3,813	65,136	1,306
	3	65142	65,089	3,621	65,154	1,337
	4	75513	65,284	4,021	65,349	1,368
	5	65614	64,730	3,662	64,803	1,322
1500_100_15	1	18704	69,939	4,245	69,225	0,885
	2	19091	65,984	5,841	72,810	1,862
	3	19300	69,482	4,734	73,560	1,203
	4	19724	68,359	4,275	68,427	1,005
	5	16711	64,746	5,401	65,549	0,748
1500_200_15	1	40198	65,875	5,338	65,940	2,727
	2	32736	64,691	4,086	64,756	2,602
	3	40169	66,824	4,663	66,891	2,668
	4	39170	67,613	8,605	68,562	1,111
	5	34538	64,076	4,826	64,140	4,368
1500_300_15	1	48306	65,123	6,474	65,188	1,584
	2	49415	64,609	6,705	64,674	1,693
	3	54483	65,160	7,081	65,225	1,779
	4	48786	64,288	6,630	64,353	1,621
	5	50044	64,577	6,948	64,642	1,604
1500_400_15	1	80285	65,479	9,594	65,544	1,984
	2	68564	64,756	8,424	64,821	2,097
	3	66932	64,447	8,752	64,511	2,186
	4	65505	64,447	9,422	64,511	2,207
	5	71815	65,820	9,500	65,885	2,040
2000_100_15	1	18526	68,634	14,513	69,312	1,225
	2	17104	66,029	12,800	66,096	1,664
	3	18496	67,793	10,947	67,861	1,421
	4	17650	66,442	12,202	66,630	2,281
	5	17585	68,801	12,218	68,870	0,740
2000_200_15	1	32558	64,542	11,637	64,607	3,072
	2	32209	65,040	7,847	65,105	1,575
	3	33841	65,133	10,634	65,198	4,875
	4	32903	64,818	15,906	64,883	5,042
	5	40132	67,121	13,402	67,189	5,146
2000_300_15	1	53991	65,825	12,249	65,892	4,250
	2	50987	64,660	12,636	64,725	4,595
	3	55071	65,516	12,419	65,582	2,851

Continua

	4	60652	66,495	12,669	66,562	2,474
	5	58074	66,532	9,128	66,598	4,481
2000_400_15	1	73122	66,492	16,646	66,559	3,211
	2	65536	64,685	16,661	64,751	3,202
	3	64899	64,667	16,193	64,732	2,997
	4	71872	65,628	17,222	65,694	4,034
	5	65013	64,361	14,898	64,426	2,808

Conforme o esperado, dado os resultados do algoritmo mono-critério (ver Tabela 7), os tempos de execução do algoritmo multi-critério crescem mais rápido quando comparados com o CPLEX (ver Tabela 8). Os primeiros cenários de teste (500 propostas) apresentam um bom desempenho superando o software comercial no início, quando são avaliadas 100 e 200 linhas. Quando avaliadas 300 e 400 linhas o algoritmo compete com tempos similares. Na medida em que aumenta o tamanho das instâncias, o desempenho do algoritmo cai com relação ao CPLEX.

Pode-se observar na Tabela 8, que os valores máximos calculados por ambos diferem por centésimos. Embora os cálculos realizados sejam os mesmos, foi constatado que as aproximações feitas aos números no decorrer do processo criam a diferença final. Vale ressaltar que embora exista divergência no valor máximo, as propostas selecionadas são iguais em ambos. Para exemplificar foram selecionadas duas instâncias que são apresentadas nas tabelas 9 e 10:

Tabela 9 – Comparação das propostas e seus valores entre o GAMS/CPLEX 10.2 e o algoritmo do cenário 1500_100_15_2.

GAMS/CPLEX 10.2		ALGORITMO MULTICRITÉRIO	
Número da proposta	Valor da proposta	Número da proposta	Valor da proposta
233	0,817	233	0,816
332	0,769	332	0,768
421	0,797	421	0,796
534	0,765	534	0,764
628	0,792	628	0,791
759	0,782	759	0,781
782	0,666	782	0,665
856	0,568	856	0,567
1217	0,828	1217	0,827
1306	10,374	1306	10,364
1307	10,384	1307	10,374
1338	0,796	1338	0,795
1360	10,598	1360	10,588
1361	11,394	1361	11,382
1362	9,918	1362	9,908
1363	10,433	1363	10,423
TOTAL	70,681	TOTAL	70,610

Tabela 10 - Comparação das propostas e seus valores entre o GAMS/CPLEX 10.2 e o algoritmo do cenário 500_100_15_3.

GAMS/CPLEX 10.2		ALGORITMO MULTICRITÉRIO	
Número da proposta	Valor da proposta	Número da proposta	Valor da proposta
34	32,725	34	32,693
50	13,85	50	13,836
118	8,39	118	8,382
147	5,507	147	5,502
220	4,114	220	4,110
266	0,718	266	0,717
487	0,756	487	0,755
TOTAL	66,06	TOTAL	65,995

Pode-se observar que as propostas selecionadas são iguais, mas existe uma diferença entre seus valores. Acredita-se que esta divergência é causada pelos métodos de aproximação próprios de cada programa. Com esses dois exemplos é possível conferir que a solução final não é alterada, no entanto as decisões devem ser tomadas com base num único método; isto significa que ambos não podem ser utilizados de forma simultânea numa análise.

4.2.2.1 Comparação de resultados entre algoritmos mono-critério e multi-critério.

Conforme observado no Gráfico 6, os tempos médios de execução são similares nos cenários com 500 propostas, já nos cenários restantes o algoritmo mono-critério supera ao multi-critério. Essa vantagem pode acontecer devido à quantidade de informação que deve ser processada e a dificuldade intrínseca na busca pela melhor combinação de propostas. Os cenários com 1000 propostas apresentam um leve aumento no tempo médio, exceto o cenário de 100 linhas que além de ter um tempo médio elevado em ambos os algoritmos apresenta também uma diferença maior. Os cenários com 1500 propostas têm um desempenho que diminui conforme aumenta o número de linhas e também a diferença entre os tempos com relação ao cenário anterior. Finalmente, os cenários com 2000 propostas apresentam um comportamento similar ao de 1000 propostas só que em maior escala, ou seja, o tempo médio e sua diferença crescem proporcionalmente ao número de linhas exceto quando são 100 linhas, onde o desempenho diminui drasticamente.

Gráfico 6 – Comparação do tempo médio de execução entre os algoritmos mono-critério e multi-critério

Quando utilizados múltiplos-critérios não existe mais a prioridade para minimizar o custo total, e sim para selecionar a melhor combinação segundo a importância dada a cada critério, inclusive o custo total. Sabe-se que ao beneficiar um critério na maioria dos casos se prejudica outro. Assim, neste caso, dada a igualdade dos pesos de custo, nível de serviço e emissão de CO₂, cada vez que se intenta melhorar o nível de serviço ou emissão, o custo pode aumentar

Embora a solução multi-critério não deva ser analisada só em termos de custo, a modo de ilustração o Gráfico 7 compara o custo ótimo contra o custo multi-critério. Nesse gráfico pode-se observar em termos monetários a influência dos pesos dos critérios. Na obtenção do custo ótimo, toda a importância é voltada para o custo enquanto o nível de serviço e a emissão são nulos. Por outro lado, o custo multi-critério é calculado dando igual importância aos pesos, sendo natural o incremento nos custos. Em algumas instâncias o custo de considerar o nível de serviço e emissão parece bastante alto, mas de qualquer forma é o comprador que avalia quão relevantes são os critérios e se vale a pena ou não incluí-los na análise.

Gráfico 7 – Comparação entre o custo total ótimo e o custo total multi-critério

4.2.2.2 *Análise de sensibilidade entre os pesos dos critérios*

Antes de realizar a análise de sensibilidade vale a pena esclarecer que, na prática, no leilão multi-critério os pesos são fixados e publicados antecipadamente, desse modo o ofertante (transportadora) pode realizar lances mais aproximados dos objetivos que o leiloeiro (comprador) deseja alcançar. Pode-se dizer que pesos fixos asseguram a objetividade na avaliação das propostas. Por tal razão, no caso que o leiloeiro deseja mudar o peso de algum critério para avaliar uma solução alternativa, é conveniente informar ao ofertante e solicitar uma nova proposta, do contrário o leilão não se realizará de forma balanceada para todos os ofertantes.

O objetivo da análise de sensibilidade é mostrar que a partir do mesmo conjunto de propostas são obtidas soluções diferentes dependendo dos pesos designados a cada critério. Esses pesos representam as preferências do comprador com relação a cada critério, as quais são definidas considerando o mercado de atuação do comprador.

Neste caso, foi avaliada a instância 500_100_15_1 com custo ótimo de R\$ 15.895,00 considerando sete cenários de variação nos pesos dos critérios (ver Tabela 11). A primeira variação é a inicialmente utilizada na análise de resultados. A segunda beneficia o custo diminuindo o nível de serviço e a emissão de CO₂. A terceira beneficia o nível de serviço e mantém os outros critérios com a mesma ponderação. A quarta variação designa pesos iguais ao custo e ao nível de serviço e diminui a emissão de CO₂. Finalmente, as últimas três variações correspondem à designação do maior peso aos critérios nível de serviço, emissão de CO₂ e custo, respectivamente.

Tabela 11 – Variação dos pesos dos critérios

Variação dos pesos	Custo	Nível de serviço	Emissão de CO ₂	Total
1	0.333	0.333	0.333	1.00
2	0.590	0.205	0.205	1.00
3	0.205	0.590	0.205	1.00
4	0.400	0.400	0.200	1.00
5 (NS)	0.000	1.000	0.000	1.00
6(Emissão)	0.000	0.000	1.000	1.00
7(Custo)	1.000	0.000	0.000	1.00

O algoritmo foi executado para a instância selecionada com cada uma das variações descritas anteriormente. Os resultados analisados foram as soluções encontradas em cada uma das sete variações. Estes resultados permitem analisar a relação entre o conjunto de propostas correspondente a cada solução, e os pesos dos critérios, mostrando a importância que tem na decisão final selecioná-los adequadamente.

A fim de comparar as soluções, foi calculado, em cada uma (conjunto de propostas), a média aritmética do percentual de avarias, o *lead time* e a emissão de CO₂, assim como, o custo total, valor máximo e tempo de execução. Os resultados estão detalhados na Tabela 12. Os valores do nível de serviço, emissão e custo obtidos na variação 5, 6 e 7, respectivamente, servem de base para comparação uma vez que estes são os valores mínimos de cada critério para esta instância em particular. Assim, o menor nível de serviço está representado em média por 2,51% de avarias e 4 dias de *lead time*. Seu valor máximo (52,642) indica que o conjunto de propostas que possui o melhor nível de serviço, não atinge em sua totalidade os requerimentos do comprador, os quais seriam, obter o menor *lead time* em cada uma das 100 linhas, em torno de 1 a 3 dias dependendo da linha, e as menores avarias (0,75 %) segundo o Gráfico 3.

A respeito da emissão, na variação de pesos número 6, o menor valor obtido é um total de 11669,19 KgCO₂e. Isso significa que, as emissões totais diminuiriam quase a metade se comparadas com a emissões totais no pior caso, 21744,23 KgCO₂e, que corresponderia a um fator de aproximadamente 0,54. Segundo o Gráfico 5, esse valor está bastante próximo ao melhor fator (0,5) que corresponderia ao valor de 10872,12 KgCO₂e, por este motivo seu valor máximo ficou em 92,668.

A variação número 7 consegue um custo de R\$15895, valor igual ao custo ótimo obtido para esta instância (ver Tabela 7). No entanto, o valor máximo é de 74,652 porque a função de valor do custo está construída para calcular o valor por proposta (ver Gráfico 2), como consequência o fator utilizado (ver eq.(32)) baseia-se no custo médio por linha, ou seja, dado que tem um valor calculado com média aritmética, um fator com valor de 1 pode não ser o melhor, porque o custo de determinada proposta poderia ser menor que o custo ótimo médio (custo ótimo médio por linha vezes o número de linhas da proposta). Então somando os valores obtidos por cada proposta, pode acontecer que o valor máximo seja menor e mesmo assim obter uma

boa solução. O caso anterior demonstra a importância de analisar os resultados acompanhando o raciocínio de sua respectiva função de valor.

Tabela 12 – Análise de sensibilidade da instância 500_100_15_1

Variação dos pesos	Valor máximo	Custo	Avaria média (%)	Lead time médio (dias)	Emissão de CO2 (KgCO ₂ e)	Tempo de execução
1	66,167	16439	2,63	6	13229,57	0,130
2	69,094	15902	2,78	7	13514,54	0,166
3	59,473	18835	2,58	4	11922,67	0,211
4	63,641	16040	2,74	7	13476,95	0,139
5 (NS)	52,642	20673	2,51	4	11861,97	1,110
6(Emissão)	92,668	19805	2,55	4	11669,16	1,371
7(Custo)	74,652	15895	2,80	7	13574,49	1,240

Uma vez estabelecida a base de comparação da Tabela 10, as variações restantes (ver Tabela 9) são analisadas: a primeira variação consegue uma solução intermediária, ou seja, os critérios não se enquadram nos valores máximos ou mínimos, sendo evidente o equilíbrio entre eles. Com relação aos valores mínimos, há uma diferença de R\$544 no custo, 0,12% nas avarias, 2 dias no *lead time* e 1560,41KgCO₂e na emissão, com esta solução o comprador não beneficia nenhum critério em particular.

A segunda variação, busca favorecer o custo atribuindo-lhe 0,59 e dividindo 0,41 por igual entre os demais critérios. Como resultado, o custo obtido aproxima-se ao custo ótimo com diferença de R\$7, por outro lado, o nível de serviço diminui, com um incremento de 0,27% nas avarias e 3 dias no *lead time*, e a emissão cresce em 1905,33KgCO₂e.

Na mesma proporção do cenário anterior, o cenário 3, beneficia o nível de serviço deixando iguais o custo e a emissão, neste caso, o aumento do custo com relação ao ótimo é maior (R\$ 2940), a emissão cresce moderadamente mantendo uma diferença com o menor valor de 253.51KgCO₂e, e o nível de serviço mantém-se muito próximo aos melhores valores, com o mesmo *lead time* e uma diferença de 0.07% nas avarias. Embora custo e emissão possuam a mesma ponderação, esta última tem uma relação inversamente proporcional com o nível de serviço, então

para maior nível de serviço, menor emissão, por causa disso o crescimento é menor se comparado ao caso anterior.

Finalmente, o cenário 4, tem o propósito de beneficiar o custo e o nível de serviço dando uma ponderação de 0,4 para cada um, e 0,2 para a emissão. Pode-se observar que não há evidência de favorecimento do custo e do nível de serviço em detrimento da emissão, isto pode acontecer devido ao *trade - off* entre o custo e o nível de serviço. Enquanto o custo diminui o nível de serviço diminui e vice-versa, portanto, não existe solução que favoreça ambos a não ser buscando um equilíbrio, e foi o que aconteceu neste cenário. A respeito da emissão, esta diminui conforme a melhoria o nível de serviço, por causa disso o valor ficou acima da média dos resultados dos 7 cenários.

Com a análise anterior vemos que a seleção dos critérios e o entendimento de seus *trade-offs* são fundamentais na tomada de decisão utilizando o método de múltiplos critérios. Da mesma forma, o nível da decisão depende da capacidade de interpretação dos resultados obtidos, para isso é importante saber como foram construídas as funções de valor e conhecer qual o seu impacto nos resultados.

5 CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação estudou o problema de compras de serviços de transporte de carga lotação desde o ponto de vista do comprador com o objetivo de propor um método de análise que atenda as principais necessidades da compra de serviços de transporte. O método otimizante permite a utilização de propostas combinatórias como mecanismo de negociação e a análise de aspectos importantes diferentes do custo. Com estas características é possível aproveitar a estrutura da rede das transportadoras para gerar economias de escopo e realizar uma seleção baseada nas necessidades do comprador, robustez que não se tem na abordagem tradicional de minimização de custo individual. Para atender este objetivo foram propostos objetivos específicos, conforme discutido.

Foi realizada uma revisão da literatura onde se concluiu que o leilão combinatório é um mecanismo de negociação que beneficia a compra de serviços de transporte. Esse leilão possibilita expressar as preferências das transportadoras permitindo-lhes submeter as rotas que melhor se adaptam a sua própria rede de clientes e, conseqüentemente, beneficiam sua estrutura de custos. Isso ocasiona um aumento na sua economia de escopo. Ao mesmo tempo, o comprador (embarcador) pode obter o mínimo custo total de compra graças aos menores preços, produto da economia gerada pela combinação de linhas.

Foi verificado que este problema, conhecido como *WDP* na literatura, é NP – completo, devido ao crescimento exponencial das combinações de linhas (propostas) com relação ao número de linhas leiloadas pelo embarcador; portanto, o pior caso seria quando todas as possíveis combinações forem submetidas pelas transportadoras. Ainda assim, existem poucas pesquisas que estudam o leilão combinatório para problemas específicos de seleção de transportadoras, e suas principais contribuições estão direcionadas ao desenvolvimento de modelos matemáticos e análises a respeito das condições ideais de negociação. Assim mesmo, dos algoritmos pesquisados nenhum foi elaborado para tratar o problema de compras de serviços de transporte. Somente a interface REV da empresa de software *CombineNet* desenvolvida por Tuomas Sandholm aproveita algumas das estratégias de busca em árvore descritas neste trabalho (SANDHOLM et al. 2006).

A respeito da utilização de múltiplos critérios, a abordagem de custo total demonstrou maior aplicação em comparação com as técnicas de análise de decisão. Ao mesmo tempo, foi constatado que esta última consegue analisar os critérios conservando sua natureza, por intermédio das funções de valor. Desse modo o embarcador tem a vantagem de mensurar os critérios sem traduzi-los a uma unidade comum (custo generalizado) e ficar sujeito a diminuir sua sensibilidade frente a mudanças, as quais que poderiam levar a tomar de decisões equivocadas.

Em virtude da revisão da literatura, decidiu-se adaptar e aplicar um algoritmo que permitisse obter as melhores soluções em tempo polinomial. Para isso foi realizada uma revisão teórica visando conhecer quais algoritmos buscavam solucionar o problema de determinação do vencedor (*WDP*). Ao analisar cada um deles foi selecionado o algoritmo base desta pesquisa (Sandholm (2002)). Vale ressaltar que esse algoritmo base não tem uma aplicação em particular, simplesmente explica algumas estratégias para a obtenção da solução ótima utilizando combinações de itens.

Uma vez definido o algoritmo base, o primeiro algoritmo a ser construído foi um algoritmo exato mono-critério, específico para compras de serviços de transporte. Com esse propósito foram efetuadas diversas mudanças nos parâmetros e funções do algoritmo original e conservadas suas estratégias de busca. A fim de validar a qualidade das soluções obtidas, as mesmas foram comparadas com as soluções geradas pelo software comercial CPLEX. Foi também analisado o tempo de execução, ainda que superar o tempo do software comercial não fizesse parte dos objetivos, já que o único interesse relacionado ao desempenho é a execução do algoritmo em tempo polinomial.

Com relação à qualidade da solução, o algoritmo mono-critério fornece soluções ótimas para instâncias de tamanho real; utilizando como uma referência a Tabela 3 traduzida de Caplice e Sheffi (2006). Nessa tabela uma instância é considerada real desde que possua uma quantidade de linhas maior que 136; valor que se enquadra dentro das instâncias testadas que vão desde 100 até 400 linhas.

A respeito do tempo de execução, o algoritmo teve desempenho polinomial de ordem 2 em todas as instâncias testadas. Por outro lado, as comparações com o CPLEX mostraram que o tempo de execução do algoritmo é menor ou similar nas primeiras instâncias testadas (500 propostas e 100 a 400 linhas), mas superior nas instâncias maiores.

Dado que as pesquisas na área de transporte demonstraram que, a seleção de transportadoras envolve outros aspectos diferentes do custo e que o *trade - off* mais ponderado é o custo e o nível de serviço. Esse trabalho também tinha como objetivo adaptar o algoritmo para avaliar vários critérios (algoritmo multi-critério). Depois de realizar uma revisão da literatura a analisar as abordagens de custo total e de análise de decisão, optou-se por aplicar técnicas da Teoria de Valor Multi-Atributo (MAVT) para o desenvolvimento do algoritmo. Os critérios foram escolhidos com o objetivo de representar necessidades atuais das empresas embarcadoras. Na prática o critério mais importante além do custo é o nível de serviço, cuja ponderação realizou-se considerando o *lead time* e as avarias. Ambos os sub-critérios foram escolhidos de Prado (2011), já que parte de seu trabalho está direcionado a seleção de um operador logístico utilizando múltiplos critérios. Entretanto, a emissão de CO₂ é uma questão que tem sido abordada por muitas empresas (60% segundo o estudo do ILOS) em suas políticas de sustentabilidade, que buscam com sua redução uma oportunidade para aumentar sua vantagem competitiva, e ao mesmo tempo contribuir com o meio ambiente. No Brasil o CO₂ é o principal GEE produzido pelo setor de transporte, sendo 88,6% das emissões no transporte rodoviário.

Nesta linha, foram construídas as funções de valor para cada critério, com base em pesquisas recentes e na experiência de profissionais do setor. O principal desafio era criar funções que representassem a natureza da cada critério e ao mesmo tempo fossem úteis para avaliar combinações de linhas (pacotes). No total foi utilizada uma função de valor por critério a exceção do *lead time*, já que foi construída uma função de *lead time* por cada linha leiloadas. Adicionalmente, foi formulado o modelo matemático específico para propostas combinatórias.

O algoritmo foi adaptado para maximizar valor, objetivo principal dos modelos multi-critério, e foi testado nas mesmas condições do algoritmo mono-critério com instâncias adaptadas para esse feito. Para o teste principal foram designados pesos iguais a cada critério (0.333). Seu desempenho foi avaliado contra o CPLEX, sendo que para isso, foi implementado o modelo matemático desenvolvido no software comercial. Dos resultados obtidos pode-se concluir que:

- As soluções encontradas foram as mesmas em ambos, apesar de seu valor máximo ser diferente por alguns centésimos. Essa divergência foi ocasionada pelas aproximações numéricas feitas no CPLEX no decorrer do processo de

cálculo. Acredita-se que pequenas diferenças não afetam a eficiência do algoritmo com relação à obtenção da melhor solução. Vale ressaltar que na prática deve-se utilizar só uma ferramenta para obter os resultados a serem analisados, já que sua combinação pode prejudicar a análise dos dados.

- O comportamento do tempo de execução foi similar ao primeiro algoritmo. O menor cenário (500 ofertas) apresentou tempos menores ou parecidos, e conforme o tamanho das instâncias era incrementado, aumentava o tempo de execução. No entanto, a tendência foi de tempo polinomial de ordem 2.

Quando comparados os resultados do algoritmo mono-critério e do algoritmo multi-critério, verificou-se que este último apresenta maior tempo de processamento nas instâncias superiores a 500 propostas. Adicionalmente, ambos demonstram dificuldade para tratar instâncias com 100 linhas e 1000 a 2000 propostas. Aparentemente essas instâncias possuem muitas ofertas que contém poucas linhas, fato que aumenta o tempo da busca por soluções já que, na árvore de ofertas os últimos nós a serem explorados são as propostas com menos linhas.

Finalmente, foi realizada uma análise de sensibilidade cujo objetivo era mostrar como a atribuição de pesos diferentes consegue mudar a solução e qual o seu impacto na decisão final. Nesta análise foram criados sete cenários com ponderações diferentes, dos quais três tinham o máximo peso em um dos critérios avaliados. Os demais cenários foram construídos principalmente para observar o *trade-off* do custo vs. o nível de serviço e a relação de ambos com a emissão de CO₂.

- Com essa análise vemos que ponderações que beneficiam o custo ou o nível de serviço são identificadas facilmente porque suas soluções melhoram um, enquanto prejudicam o outro. Da mesma forma, a emissão de CO₂ muda inversamente ao nível de serviço, devido a relação adjudicada que indica que, quanto maior o nível de serviço de uma transportadora maior a chance da empresa ter uma frota menos poluente e a infra-estrutura ser suficiente para sua manutenção.
- Por outro lado, cenários que atribuíram igual ponderação ao nível de serviço e ao custo obtiveram soluções regulares que não satisfaziam nenhum critério. Por causa do *trade-off* não é possível encontrar propostas com custo baixo e bom nível de serviço, então acabam selecionando-se propostas intermediárias. Neste caso, a emissão de CO₂ continua mudando dependendo do nível de serviço.

Contudo, pode-se concluir que esta dissertação adaptou um algoritmo que abordou o problema de compras utilizando um mecanismo de negociação que permite a obtenção de economias de escopo e, ao mesmo tempo, oferece a possibilidade de avaliar aspectos diferentes do custo que também são importantes para a seleção. Mostrou também a importância que tem alinhar os critérios e suas ponderações com os objetivos da empresa, entender os *trade-offs* presentes na decisão e avaliar cenários diferentes para avaliar um panorama global das vantagens e desvantagens que cada um traz. Adicionalmente, mostrou a necessidade de ter pleno conhecimento das funções de valor para realizar uma interpretação dos resultados bem sucedida que seja refletida na decisão final.

Os resultados apresentados neste trabalho buscam contribuir com a comunidade acadêmica que estuda problemas de seleção de fornecedores de serviço de transporte. Intenta-se mostrar como a utilização de ferramentas diferentes, como a análise de decisão, ajuda a aumentar a robustez deste problema complexo, que ainda é tratado por muitos como um problema otimizante cujo objetivo é o menor custo total. Espera-se que esse algoritmo sirva como base para trabalhos futuros na área.

Como limitação do algoritmo proposto, pode-se mencionar o tempo de execução que embora polinomial, incrementa conforme o tamanho das instâncias. Outra limitação é a falta de restrições com relação ao volume transportado em cada linha e a delimitação da quantidade de fornecedores por linha ou região, e demais restrições que tornem o algoritmo mais prático. A última limitação é a falta de aplicação em um caso real para avaliar seu desempenho.

Algumas sugestões de trabalhos futuros que podem ser realizados a partir da dissertação são:

- Continuar pesquisando novas estratégias de busca que permitam melhorar o desempenho do atual algoritmo em termos de tempo computacional e capacidade de processamento de instâncias maiores;
- Aumentar sua robustez inserindo novas restrições de contorno, por exemplo, restrições de volume e restrições por região.
- Incorporar priorização para cada uma das linhas, ou seja, o algoritmo multi-critério estaria capacitado para decidir quais linhas devem atender os critérios com maior rigorosidade.

- Aplicá-lo a um caso real que permita efetuar o processo decisório completo, desde a seleção dos critérios até a implementação da decisão final.

REFERÊNCIAS

AUSUBEL, L.; MILGROM, P. The lovely but lonely Vickrey auction. In: CRAMTOM, P.; SHOHAM, Y.; STEINBERG, R. **Combinatorial auctions**. Cambridge: The MIT press, 2006, Cap. 1, p. 17-40.

BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos: Logística empresarial**. 5 edição. Porto Alegre:Bookman, 2006, 6161 p. ISBN 8536305916.

BELTON, V. A comparison of the analytic hierarchy process and a simple multi-attribute value function. **European journal of operational research**, p. 7-21, Julho 1986.

BICHLER, M.; KALAGNANAM, J. Configurable offers and winner determination in multi-attribute auctions. **European Journal of Operational Research** , v. 160, p. 380-394, 2005.

BICHLER, M.; DAVENPORT, A.; HOHNER, G.; KALAGNANAM, J. Industrial Procurement Auctions. In: CRAMTOM, P.; SHOHAM, Y.; STEINBERG, R. **Combinatorial auctions**. Cambridge: The MIT press, 2006, Cap. 23, p. 593-612.

CAPLICE, C. G. **An optimization based bidding process: a new framework for shipper-carrier relationships**. 1996, PhD (Transportation and Logistics System). School of Engineering, MIT, 1996.

CAPLICE, C.; SHEFFI, Y. Optimization-based procurement for transportation services. **Journal of business logistics**, p. 109-128, Oct. 2003.

CAPLICE, C.; SHEFFI, Y. Combinatorial auctions for truckload transportation. In: CRAMTOM, P.; SHOHAM, Y.; STEINBERG, R. **Combinatorial auctions**. Cambridge: The MIT press, 2006, Cap. 21, p. 539-572.

CRAMTOM, P.; SHOHAM, Y.; STEINBERG, R. **Introduction to combinatorial auctions**. In:_____. **Combinatorial auctions**. Cambridge: The MIT press, 2006, Cap. 1, p. 1-16.

DANG, V. D.; JENNINGS, N. R. Optimal clearing algorithms for multi-unit single item and multi-unit combinatorial auctions with demand/supply function bidding. In: 5TH INTERNATIONAL CONFERENCE ON ELECTRONIC COMMERCE. Pittsburgh. **Proceedings**, p. 25-30. 2003.

DAUER, J. P.; SALEH, O. A. Constructing de set of efficient objective values in multiple objective linear programs. **European Journal Of Operational Research**, v. 46, p. 358-365, 1990.

DAVENPORT, A. J.; KALAGNANAM, J. R. **Price negotiations for procurement of direct inputs**. IBM Research Report RC 22078. 2001.

DE SMET, Y. Multicriteria auctions without full comparability of bids. **European Journal of Operational Research**, v. 177, p. 358-365, 2007.

ELMAGHRABY, W.; KESKINOCAK, P. **Combinatorial auctions in procurement**. The practice of supply chain managment: where theory and aplication converge (International Series in Operations Research & Management Science). V. 62, parte 3, 2004, p 245-258. Edited by: Harrison, T. P.; Lee, H. L.; Neale, J. J. doi: 10.1007/b138860, p. 345.

FUJISHIMA, Y.; LEYTON-BROWN, K.; SHOHAM, Y. Taming the Computational Complexity of Combinatorial Auctions: optimal and aproximate approaches. In: INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE-99. Stockholm. **Proceedings of IJCAI**. 1999.

GREENE, D. L.; SCHAFER, A. Reducing de Greenhouse Emissions: Prepared for the Pew Center on Global Climate Change. **Solutions**, p. 68, 2003.

GUO, Y.; LIM, A.; RODRIGUES, B.; ZHU, Y. Carrier Assignment models in transportation procurement. **Journal Of The Operational Research Society**, v.57, p.1504-1510, 2006.

HSIEH, F.-S.; HUANG, C.-W. A comparative study of two combinatorial reverse auctions models. In: PACIFIC ASIA CONFERENCE ON INFORMATION SYSTEMS (PACIS). Taiwan, p. 248-259, 2010.

HO, W.; XU, X. e DEY, P. K. Multi-criteria Decision Making Approaches for Supplier Evaluation and Selection: A Literature Review. **European Journal of Operational Research**, v. 202, n. 1, p.16-24, 2010.

HUNTER, DAVID J. **Essentials of discrete mathematics**. Jones & Bartlett Publishers. 2009. 420p.

ILOG S.A. **ILOG CPLEX 10.0: Users Manual**, p. 478, 2006.

JARA-DIAZ, S. R. Multioutput analysis of trucking operations using spatially disaggregated flows. **Transportation Research Part B: Methodological**, p. 159-171, Junho 1988.

JENNINGS, N. R.; DANG, V. D. Polynomial algorithms for clearing multi-unit single-item. In: 15TH EUROPEAN CONFERENCE ON AI. **Proceedings**, 2002, p. 23–27.

JIN, M.; WU, S. D.; ERKOC, M. Multiple unit auctions with economies and diseconomies of scale. **European Journal of Operational Research: Production, Manufacturing and Logistics**, v.174, p816-834, 2006, doi:10.1016/j.ejor.2005.02.075.

KALAGNANAM, J.; PARKES, D. C. Auctions, bidding and exchange design. In: _____. **Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era**. Boston, Cap. 1, p. 143-212. 2004.

KAMESHWARAN, S.; NARAHARI, Y.; ROSA C. H.; KULKARNI D. M.; TEW J. D. Multiattribute electronic procurement using goal programming. **European Journal Of Operatinal Research**, v.179, p. 518-536, 2007, doi:10.1016/j.ejor.2006.01.010.

KEENEY, R. L.; **Value focused thinking: a path to creative decision-making**. Cambridge: Harvard University Press, 1992.

KEENEY, R. L. Common mistakes in making value trade-offs. **Operations Research**, v. 50, n. 6, p. 935-945, Nov.- Dez. 2002.

KOSTAMIS, D.; BIL, D. R.; DUENYAS, Y. Total procurement auctions: should the buyer reveal suppliers' quality? **Management Science**, v. 55, p.1985-1999, 2009.

LEDYARD, J. O.; OLSON, M.; PORTER, D.; SWANSON, J. A.; TORMA, D. P. The First Use of a Combined-Value Auction for Transportation Services. **Interfaces**, v. 32, n. 5, p. 4-12, Set-Out. 2002.

LEE, C.-G.; ROY H., K.; ZHONG, M. A carrier's optimal bid generation problem in combinatorial auctions for transportation procurement. **Transportation Research Part E**, v.43, p. 163-191. 2007.

LEHMANN, D.; MULLER, R.; SANDHOLM, T. The Winner Determination Problem. In: CRAMTON, P.; SHOHAM, Y.; STEINBERG, R. **Combinatorial Auctions**. Cambridge: MIT Press, 2006, Cap. 12, p. 297-318.

LORENTIZIADIS, P. Post-objective determination of weights of the evaluation factors in public procurement tenders. **European Journal Of Operations Research**, v.200, n.1, p. 261-267, Jan 2010, doi:10.1016/j.ejor.2008.12.013.

MONTIBELLER, G.; FRANCO, A. Multi-Criteria Decision Analysis for Strategic Decision Making. In: ZOPOUNIDIS, C.; PARDALOS, P. M. **Handbook of Multicriteria Analysis**. Heidelberg: Springer-Verlag Berlin, 2010. Cap. 2, p. 25-47. doi: 10.1007/978-3-540-92828-7_2.

OLSON, D. L. **Decision Aids for Selections Problems**. New York : Springer-Verlag 1996, 198 p. ISBN 978-0-387-94560-6.

ORGANIZAÇÃO NAÇÕES UNIDAS. Kyoto protocol to the united nations framework convention on climate change. United Nations. Kyoto. 1998.

PRADO, A. **Análise de seleção multicritério aplicada na seleção de fornecedores de logística**. 2011, 175 f. Dissertação de mestrado (Sistemas Logísticos). Escola Politécnica da Universidade de São Paulo, São Paulo, 2011.

REVISTA FATOR 19 MAR. 2011. O transporte rodoviário lidera as emissões de CO2. Disponível em: http://www.revistafator.com.br/ver_noticia.php?not=150565. Acesso em: 31 Maio. 2011.

ROTHKOPF, M. H.; ALEKSANDAR, P.; HARSTAD, R. M. Computationally manageable combinatorial auctions. **Management Science**, v.44, n.8, p. 1131–1147. Ago.1998, doi: 10.1287/mnsc.44.8.1131.

SAKURAI, Y.; YOKOO, M.; KAMEI, K. An Efficient Approximate Algorithm for Winner Determination in Combinatorial Auctions. In: ACM CONFERENCE ON ELECTRONIC COMMERCE (EC-00). **Proceedings**, 2000.

SANDHOLM, T. Approaches to winner determination in combinatorial auctions. **Decision Support Systems**, v. 28, p. 165–176, Mar. 2000, doi:10.1016/S0167-9236(99)00066-4.

_____. Algorithm for optimal winner determination in combinatorial auctions. **Artificial Intelligence**, v.135, p. 1-54. Fev. 2002, doi:10.1016/S0004-3702(01)00159-X .

SANDHOLM, T.; SURI S.; GILPIN A.; LEVINE D. Winner determination in combinatorial acution generalizations.In: THE FIRST INTERNATIONAL JOINT

CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEM: PART 1. **Proceedings** Bologna, Italy, 2002, doi:10.1145/544741.544760.

SANDHOLM, T.; SURI S.; GILPIN A.; LEVINE D. CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. **Management Science**, v. 51, n. 3, p. 374-390. Mar. 2005, doi: 10.1287/mnsc.1040.0336.

SANDHOLM, T.; BOUTILIER, C. Preference elicitation in combinatorial auctions. In: CRAMTON, P.; SHOHAM, Y.; STEINBERG, R. **Combinatorial auctions**. Cambridge: The MIT press, 2006, Cap. 10, p. 233-263.

SANDHOLM, T.; DAVID L.; CONCORDIA M.; MARTYN P.; HUGHES R.; JACOBS J.; BEGG D. Changing the game in strategic sourcing at Procter & Gamble: expressive competition enabled by optimization. **Interfaces**, v. 36, p. 55-68, Jan.-Fev. 2006, doi: 10.1287/inte.1050.0185.

SHEFFI, Y. Combinatorial Auctions in the Procurement. **Interfaces**, v. 34, n. 4, p. 245-252. Jul.-Ago. 2004, doi: 10.1287/inte.1040.0075.

SONG, J.; REGAN, A. Combinatorial Auctions for Transportation Service Procurement: The Carrier Perspective. **Transportation Research Record**, v. 1833, p. 40-46, 2004, doi: 0.3141/1833-06.

_____. Approximation algorithms for the bid construction problem in combinatorial auctions for the procurement of freight transportation contracts. **Transportation Research Part B**, v.139, p. 914-933. Dez. 2005, doi:10.1016/j.trb.2004.11.003.

SONG, J.-W.; YANG, S.-B. A Faster Optimal Allocation Algorithm in Combinatorial Auctions. MICAI 2004: **Advances in Artificial Intelligence**, v. 2972, p. 362-369, 2004, doi: 10.1007/978-3-540-24694-7_37.

SOUMYADIP GHOSH, M. E.; KALAGNANAM, J. R.; LADANYI, L. **Bid evaluation in procurement auctions with piece-wise linear supply curves**. IBM Research Report RC 22219, 2001.

VISWANATHKUMMAR, G.; SRINIVASAM, G. A branch and bound algorithm to minimize completion time variance on a single processor. **Computers and Operations Research**. V. 30, n. 8, p. 1135-1150, 2003, doi: 10.1016/S0305-0548(02)00062-X.

VRIES , S.; VOHRA, V. R. Combinatorial auctions: a survey. **Inform Journal on Computing**. V.15, n.3, p. 284-309, 2003, doi: 10.1287/ijoc.15.3.284.16077.

WEBER, C. A.; CURRENT, J. R. A multiobjective approach to vendor selection. **European Journal of Operational Research**, v.68, n. 2, p. 173-184, Jul. 1993, doi:10.1016/0377-2217(93)90301-3.

MONTIBELLER, G.; YOSHIZAKI, H. T. Y. **A framework for locating logistic facilities with multi-criteria decision analysis**. In: Lecture notes in computing science. Springer Verlag, p. 505-519. ISBN 978364219892, 2011.

ZHONG, M. **Combinatorial Auctions for Truckload Transportation Procurement**. 217 f. Tese (Doutorado Engenharia Mecânica). Universidade de Toronto. Toronto, 2008.

ZHONG, M.; KWON, R. H.; LEE, C. A stochastic programming winner determination model for truckload procurement under shipment uncertainty. **Transportation Research Part E: Logistics and Transportation**. v. 46, n.1, p. 49-60, 2010.

ANEXO A – Código do algoritmo mono-critério para solucionar o problema de seleção de transportadoras.

Descrição dos Métodos principais:

```

Public class Monocriterio {
    public static void main(String[] args) {

        IDA bt = new IDA();
        n_linhas= 0;
        n_ofertas = 0;

        APÓS LER ARQUIVO DE ENTRADA:

        // LÓGICA PARA CRIAÇÃO DA ÁRVORE DE OFERTAS

        i=j=0; // Inicializa o While.
        C_item=new double[n_ofertas][n_linhas];    // Cria o vetor que guardará o custo de cada linha
                                                    // custo oferta/numero de linhas da oferta)
        cont_linhas=0;    // Inicializa o contador de linhas
        for(i=0;i<n_ofertas;i++){    // Faça até completar todas as ofertas
            while(j<=n_linhas){
                cont_linhas=0;
                bt.insert(Oferta[i][j],raiz_arvore);    // Função que insere as ofertas na
                                                    // árvore
                raiz_arvore2=raiz_arvore;    // variável temporal criada para não modificar
                                                    // a verdadeira raiz da árvore

                j++;
            }
            j=0;
        }

        // ALGORITMO PRINCIPAL (BUSCA DO MINIMO CUSTO): RETORNA UM CONJUNTO DE
        // OFERTAS VENCEDORAS QUE MINIZAM O CUSTO

        Iniciar_BMC = new double[n_linhas];    // Cria a variável da busca principal
        Linha_Espera=new double[n_linhas];    // Cria a variável que guarda as linhas que
                                                    // ainda estão fora do caminho
        for(j=0; j<n_linhas; j++){    // Inicializa a variável Linha_Espera
            Linha_Espera[j]=1;
        }

        F_Limite=bt.Custo_Esperado(); // Inicializa o Limitante superior que é igual ao custo
                                                    // esperado
        cont_alocados=0;    // inicializa o contador de ofertas alocadas
        Custo=new double[n_ofertas][n_fornecedores];    // Cria variável de custo da oferta
        Stopmask = new String[n_ofertas][n_linhas];    // Cria variável da árvore de ofertas, a
                                                    // qual possui três valores para cada linha
        custo_minimo=new double[n_ofertas];    // Cria variável de custo mínimo da oferta

        // INICIALIZANDO ÁRVORE DE BUSCA DO MINIMO CUSTO

        while(true){

```

```

// 1.
bt.DepthFirstSearch(); // Chama o MÉTODO DEPTH FIRST SEARCH, Função principal do
                        // algoritmo que realiza o IDA* e a busca em profundidade

// 2.
if(fim==1){ // Retornar os ganhadores (FIM DO ALGORITMO)

// 3.
if(ajuste_f_limite==0){ // Condição que permite ajustar o limitante
    F_Limite= Novo_f;
}

} // Fim while principal

} // Fim static main

/**
 * CONSTRUÇÃO DA ÁRVORE DE OFERTAS.
 * Gera a raiz da árvore
 * raiz_arvore2 muda raiz_arvore de lugar sem perder a raiz inicial
 */
public IDA() {
    raiz_arvore = new Nó_ArvoreOfertas(5);
    raiz_arvore.visitado = false;
    raiz_arvore2=raiz_arvore;
    raiz_BMC=new Busca_Principal (null);
    Nó_BMC=raiz_BMC;
    Nó_BMC2=raiz_BMC;
}

// METODO INTERNO PARA INSERIR NA ÁRVORE DE OFERTAS

void insert(double x, Nó_ArvoreOfertas node) {
    for(int a=0;a<j;a++){
        if (Oferta[i][a] == 1){ // Condição que indica guardar à esquerda da árvore
                                // se a linha pertence à oferta
            cont_linhas++;
            raiz_arvore2 = raiz_arvore2.left;
        }
        if (Oferta[i][a] == 0 ) // Condição que indica guardar à direita da árvore se
                                // a linha pertence à oferta
            raiz_arvore2 = raiz_arvore2.right;
    }

    if(j<n_linhas){
        if (x == 1 && raiz_arvore2.left == null){ // Condição para criar um novo nó
                                                    // (filho) à esquerda
            raiz_arvore2.left = new Nó_ArvoreOfertas(x);
            raiz_arvore2.left.pai=raiz_arvore2;
            raiz_arvore2.left.visitado=false;
        }
        if (x == 0 && raiz_arvore2.right == null){ // Condição para criar um novo nó
                                                    // (filho) à direita
            raiz_arvore2.right = new Nó_ArvoreOfertas(x);
            raiz_arvore2.right.pai=raiz_arvore2;
            raiz_arvore2.right.visitado=false;
        }
    }
}

```

```

else {
    if(raiz_arvore2.right == null){
        raiz_arvore2.right=new Nó_ArvoreOfertas(i); // Cria o último nó da oferta que guarda os demais parâmetros
        raiz_arvore2.right.transportadora=new double[n_fornecedores]; // Guarda o fornecedor
        raiz_arvore2.right.custo=new double[n_fornecedores]; // Guarda o custo
        raiz_arvore2.right.bid=new double[n_fornecedores]; // Guarda o número da oferta
        raiz_arvore2.right.Vark=0; // Contador das ofertas repetidas
    }
    raiz_arvore2.right.transportadora[raiz_arvore2.right.Vark] = Oferta[i][n_linhas]; // Guarda os mesmos parâmetros para ofertas repetidas
    raiz_arvore2.right.custo[raiz_arvore2.right.Vark] =Oferta[i][n_linhas+1];
    raiz_arvore2.right.bid[raiz_arvore2.right.Vark]=i;
    for(int a=0;a<j;a++){
        if (Oferta[i][a] == 1 && k==0){
            C_item[i][a]=raiz_arvore2.right.custo[raiz_arvore2.right.Vark]/cont_linhas; //Calcula o custo por linha de cada oferta
        }
        else if (Oferta[i][a] == 1){
            C_item[i][a]=raiz_arvore2.right.custo[raiz_arvore2.right.Vark]/cont_linhas;
        }
    }
    raiz_arvore2.right.Vark++;
}
} // fim método para inserir na árvore de ofertas

```

//CLASSE QUE REALIZA A BUSCA PRINCIPAL (BMC)

```

public class Busca_Principal{
    Busca_Principal(double[] item) {
        if(item!= null){
            element = new double[item.length] ;
            for(int i=0; i<element.length; i++)
                element[i]=item[i];
        }
        children = null;
        pai=null;
    }
    double[] element;
    List <Busca_Principal> children;
    Busca_Principal pai;
}

```

//MÉTODO PARA ADICIONAR FILHOS À ÁRVORE PRINCIPAL

```

public void AddChild( double[] element){
    if(children ==null)
        children = new ArrayList<Busca_Principal>();
        children.add(new Busca_Principal(element));
        children.get(Nó_BMC.children.size()-1).pai=Nó_BMC;
    } // Fim método para adicionar filhos

```

//MÉTODO INTERNO PARA PERCORRER EM PRÉ-ORDEM A BUSCA PRINCIPAL

```

public void listPreOrder(Busca_Principal raiz_BMC2){
    if(raiz_BMC2.children!=null){
        for(Busca_Principal data:raiz_BMC2.children) {
            Busca_BMC.add(data);
        }
    }
}

```

```

        listPreOrder (data);
    }
}
} // Fim método interno pré-ordem

```

//MÉTODO PARA PERCORRER EM PRÉ-ORDEM A BUSCA PRINCIPAL

```

public void PrintSearch1(){
    Busca_BMC=null;
    Busca_BMC = new ArrayList<Busca_Principal>();
    listPreOrder(raiz_BMC);
    for(j=0; j<Busca_BMC.size(); j++){
        for(k=0; k<n_linhas; k++){
            System.out.print(Busca_BMC.get(j).element[k] + "|");
            System.out.print( "\n");
        }
    }
} // Fim método de pré-ordem

} //fim classe da Busca Principal

```

// MÉTODO DA DEPTH FIRST SEARCH (FUNÇÃO PRINCIPAL)

```

void DepthFirstSearch(){ // devolve linhas livres, propostas ganhadoras e custo atual do
                        caminho

```

```

    Caminho=new double[n_linhas];
    while(cont_alocados<n_ofertas){

```

// 1 ATUALIZAR VARIÁVEIS

```

        if(cont_alocados==0){
            nova_busca=0;
            for(j=0; j<n_linhas; j++){
                if(j==0)
                    Stopmask[cont_alocados][j]="MUST";
                else
                    Stopmask[cont_alocados][j]="ANY";
            }
        }

```

```

    Custo_Maximo= (int)Custo_Estimado(); // Contribuição das propostas que estão em espera
                                         (MÉTODO DO CUSTO ESTIMADO)

```

// 2 AVALIAR CONDIÇÃO PARA ENTRADA DE PROPOSTAS

```

if(((Custo_Caminho + Custo_Maximo)> F_Limite)){ // Comparação com o limitante superior para
                                                saber se a propostas pode entrar no caminho

    Ganhadores=null;
    Novo_f=Custo_Caminho+Custo_Maximo; // custo do caminho = atual limitante superior
    return; // Para sair da função com valores de retorno

}
else{

```

//3 BUSCAR NA ÁRVORE DE PROPOSTAS

```

// Classificar linhas dentro das variáveis Stopmask
if(cont_alocados>0){
    cont_linhas2=0;
    for(j=0; j<n_linhas; j++){

```

```

        if(Linha_Espera[j]==0){
            Stopmask[cont_alocados][j]="BLOCKED";
        }
        else if((cont_linhas2==0 &&
            Stopmask[cont_alocados-1][j].compareTo("BLOCKED")<0){
            Stopmask[cont_alocados][j]="MUST";
            cont_linhas2=1;
        }
        else if(Stopmask[cont_alocados-1][j].compareTo("BLOCKED")<0) {
            Stopmask[cont_alocados][j]="ANY";
        }
        else{
            Stopmask[cont_alocados][j]="BLOCKED";
        }
    }        // Fim for
}          // Fim If

cont_linhas2 = 0;
for(k=0;k<n_linhas; k++){
    if(Stopmask[cont_alocados][k].equals("BLOCKED"))
        cont_linhas2++;
}

// Chama O MÉTODO DE VISITAÇÃO DA ÁRVORE
if(cont_linhas2 != n_linhas){
    k=0;
    novavisitação=false;
    CONTA++;
    Visitacao(raiz_arvore);
    while(novavisitação==true){
        k=0;
        novavisitação=false;
        CONTA++;
        Visitacao(raiz_arvore);
        if(raiz_arvore.right.visitado==true)
            novavisitação=false;
    }

    //NÃO HÁ MAIS NÓS PARA PERCORRER, CAMINHO ENCONTRADO:
    else{
        ajuste_f_limite=1;        // O F_limite foi ajustado, caminho encontrado.
        F_Limite= Custo_Caminho; // Atualização do upper bound
        Novo_f=Custo_Caminho;    // Atualização do novo limite, variável auxiliar
        Nó_BMC2=Nó_BMC;

        // Guarda o caminho encontrado no vetor Ganhadores
        if(Nó_BMC2.children==null){
            Ganhadores = null;
            Ganhadores = new ArrayList<double[]>();
            while(Nó_BMC2.element!=null){
                Ganhadores.add(Nó_BMC2.element);
                Nó_BMC2=Nó_BMC2.pai;
            }
        }
        return; // Sai da função Depth First Search
    }
}

```

//4 ATUALIZAÇÃO E BACKTRACKING SE A PROPOSTA NÃO FOI ENCONTRADA

```

// SE NÃO ENCONTROU CAMINHO
if(j==n_linhas){
    // Chama os MÉTODOS DE BACKTRACKING
    BacktrackStopmask(Stopmask);
    raiz_arvore2=raiz_arvore;k=0;
    Backtracking(raiz_arvore2,Caminho);

    // Chama o MÉTODO PARA DESBLOQUEAR PROPOSTAS
    raiz_arvore3=raiz_arvore;
    for(i=0; i<n_linhas; i++){
        if(Linha_Espera[i] == 0 && raiz_arvore3.right!=null)
            raiz_arvore3 = raiz_arvore3.right;
        else if(Linha_Espera[i] == 1 && raiz_arvore3.right!=null){
            raiz_arvore3 = raiz_arvore3.right;
            DesbloquearBid(raiz_arvore3);
            break;
        }
    }
}

// Chama o MÉTODO PARA ATUALIZAR CUSTO
cont_linhas2=1;
encontra_filho=1;
Atualizo_custo_solução();
}

```

// 5 LOOP PARA CONTINUAR A BUSCA RECURSIVA

```

// SE ENCONTROU FILHO FAÇA:
if(j!=n_linhas) {
    encontra_filho=0;cont_linhas2=0;
    Atualizo_custo_solução();

    //PARA CADA PROPOSTA QUE SEJA FILHO DA ATUAL, FAZER:

    // SE FOR A ÚLTIMA OFERTA QUE TEM A LINHA 1:
    if(raiz_arvore.left.visitado==true){ // Continua a última busca depois que a última oferta da árvore
                                        foi alocada
        if(ultimo==1){
            this.DepthFirstSearch(); // Chama uma vez a função principal DFS
            ultimo=0;
        }
        else{
            while( Stopmask[cont_alocados][0].compareTo("MUST")<0){
                this.DepthFirstSearch();
                Nó_BMC2=Nó_BMC;
                if( Ganhadores!=null && Nó_BMC2.children==null && Novo_f < CUSTO_MINIMO){
                    Melhores_i_alocados = null;
                    Melhores_i_alocados = new ArrayList<double[]>();
                    while(Nó_BMC2.element!=null){
                        Melhores_i_alocados.add(Nó_BMC2.element);
                        Nó_BMC2=Nó_BMC2.pai;
                    }
                    CUSTO_MINIMO = Novo_f;
                }
            }
            return;
        }
    }
}

```

```

}    // Fim While

// ENQUANTO EXISTAM PROPOSTAS NÃO EXPLORADAS DO LADO ESQUERDO

while(raiz_arvore.left.visitado==false){

//A)
this.DepthFirstSearch(); // Chama função principal DFS

if(raiz_arvore.left.visitado==false ){

//B)
// Compara o custo do último caminho encontrado com o custo do melhor alocado, se for menor o
último caminho passa a ser o melhor alocado
    Nó_BMC2=Nó_BMC;
    if( Ganhadores!=null && Nó_BMC2.children==null && Novo_f < CUSTO_MINIMO){
        Melhores_i_alocados = null;
        Melhores_i_alocados = new ArrayList<double[]>();
        while(Nó_BMC2.element!=null){
            Melhores_i_alocados.add(Nó_BMC2.element);
            Nó_BMC2=Nó_BMC2.pai;
        }
        CUSTO_MINIMO = Novo_f;
    }

//C)
// Selecionar o valor máximo entre o próximo limitante e o atual limitante
    if(Proximo_f > Novo_f)
        Proximo_f=Novo_f;

// Chama os MÉTODOS DE BACKTRACKING
    BacktrackStopmask(Stopmask);
    raiz_arvore2=raiz_arvore;k=0;
    Backtracking(raiz_arvore2,Caminho);

// chama o MÉTODO PARA DESBLOQUEAR OFERTAS
    raiz_arvore3=raiz_arvore;
    for(i=0; i<n_linhas; i++){
        if(Linha_Espera[i] == 0 && raiz_arvore3.right!=null){
            raiz_arvore3 = raiz_arvore3.right;
        }
        else if(Linha_Espera[i] == 1 && raiz_arvore3.right!=null){
            raiz_arvore3 = raiz_arvore3.right;
            DesbloquearBid(raiz_arvore3);
            break;
        }
    }
}

// Condição que recomeça a busca em caso de ainda não ter alocado nenhum caminho
if(Ganhadores==null&&Melhores_i_alocados==null){
    Custo_Caminho=0;
    raiz_BMC=null;
    raiz_BMC=new Busca_Principal(null);
    Nó_BMC=raiz_BMC;
    DesbloquearBid(raiz_arvore);
    return;
}

```



```

    } // fim if

    // Chama o MÉTODO PARA ATUALIZAR CUSTO
    cont_linhas2=1;
    encontra_filho=1;
    Atualizo_custo_solução();

} //Fim while

// 6 RETORNAR SOLUÇÃO ENCONTRADA

// Se encontrou solução final então retorne a solução e a seu custo
if(Melhores_i_alocados!=null){
    fim=1;
    return ;
}
else{
    Ganhadores=null;
    Novo_f=Proximo_f;
    return ;
}

} // Fim else filhos

} // Fim else

} // Fim while

} // Fim Depth First Search

} // Fim public class Monocriterio

```

ANEXO B – Código do algoritmo multi-critério para solucionar o problema de seleção de transportadoras.

Descrição dos principais métodos e mudanças a respeito do algoritmo mono-critério:

```

public class Multicriterio {

public static void main(String[] args) {
    Multicriterio bt = new Multicriterio();

// APÓS INICIALIZAÇÃO DAS VARIÁVEIS E LEITURA DO ARQUIVO DE ENTRADA:

// FUNÇÕES DE VALOR

    Linhas_Oferta=new int[n_ofertas];
    for(i=0; i<n_ofertas; i++){
        k=0;
        Linhas_Oferta[i]=0;
        for(j=0; j<n_linhas; j++){
            if(Oferta[i][j]==1)
                k++;
        }
        Linhas_Oferta[i]=k;
    }

// Valores gerados a partir da Função de custo

    A1=(Dado[6][1]-Dado[5][1])/(Dado[6][0]-Dado[5][0]);
    A2=(Dado[7][1]-Dado[6][1])/(Dado[7][0]-Dado[6][0]);
    A3=(Dado[8][1]-Dado[7][1])/(Dado[8][0]-Dado[7][0]);
    Custo_Otimo=X;
    Fator_FCosto=new double[n_ofertas];
    Valor_FCosto=new double[n_ofertas];
    for(i=0; i<n_ofertas; i++){
        Fator_FCosto[i] = Oferta[i][n_linhas+1]/((Custo_Otimo/n_linhas)*Linhas_Oferta[i]);

// Segmentos da função de custo
        if(Dado[5][0]<=Fator_FCosto[i]&&Fator_FCosto[i]<Dado[6][0]){
            Valor_FCosto[i]=Dado[5][1]+A1*Fator_FCosto[i]-A1*Dado[5][0];
        }
        else if(Dado[6][0] <= Fator_FCosto[i] && Fator_FCosto[i] < Dado[7][0]) {
            Valor_FCosto[i]=Dado[6][1]+A2*Fator_FCosto[i]-A2*Dado[6][0];
        }
        else if(Dado[7][0] <= Fator_FCosto[i] && Fator_FCosto[i] <= Dado[8][0]) {
            Valor_FCosto[i]=Dado[7][1]+ A3 * Fator_FCosto[i] - A3 * Dado[7][0];
        }
    }

// Valores gerados a partir da Função de Lead Time

    Valor_FLeadTime=new double[n_ofertas];
    for(i=0; i<n_ofertas; i++){
        Valor_FLeadTime[i]=0;
        for(j=0; j<n_linhas; j++){
            if(Oferta[i][j]==1){

```

```

        A1=(0-100)/(Dado[15+j][1]-Dado[15+j][0]);
        Valor_FLeadTime[i]=Valor_FLeadTime[i] + ( A1 * Oferta[i][n_linhas+2+j] + 100 - (A1 *
            Dado[15+j][0]));
    }
}
Valor_FLeadTime[i]=Valor_FLeadTime[i]/Linhas_Oferta[i];
}

// Valores gerados a partir da função de avarias

double d=Dado[11][0]*Dado[11][0]-Dado[9][0]*Dado[9][0];
double e=Dado[10][0]*Dado[10][0]-Dado[9][0]*Dado[9][0];

//Parâmetros
B =(((Dado[9][1]/e)*d + ((Dado[10][1]/e)*d) - Dado[9][1] + Dado[11][1]) /((((Dado[9][0]-
    Dado[10][0])/e)* d +Dado[11][0]-Dado[9][0]));
A = ((B*(Dado[9][0]-Dado[10][0])) - Dado[9][1] + Dado[10][1])/e;
C = Dado[9][1] - (A*(Dado[9][0]*Dado[9][0])) - (B*Dado[9][0]);

//Função
Valor_FAvarias=new double[n_ofertas];
for(i=0; i<n_ofertas; i++)
Valor_FAvarias[i]= A*((Oferta[i][n_linhas*2+2]/1000)*(Oferta[i][n_linhas*2+2]/1000)) +
    B*((Oferta[i][n_linhas*2+2]/1000)) + C;

// Valores gerados a partir da função de Emissão de Carbono

//Parâmetros
A1=(Dado[13][1]-Dado[12][1])/(Dado[13][0]-Dado[12][0]);

//Função
Valor_FEmissao=new double[n_ofertas];
double[] Fator_FEmissao=new double[n_ofertas];
double Emissao_total=Dado[14][0];
for(i=0; i<n_ofertas; i++){
    Fator_FEmissao[i] = (Oferta[i][n_linhas*2+3]/1000)/((Emissao_total/n_linhas)*Linhas_Oferta[i]);
    Valor_FEmissao[i]= Dado[12][1]+A1*Fator_FEmissao[i]-A1*Dado[12][0];
}

// VALOR PARA CADA OFERTA (SOMATORIA: PESO*VALOR CRITERIO)
Valor_Oferta=new double[n_ofertas];
double l=n_linhas;
for(i=0; i<n_ofertas; i++){
    Valor_Oferta[i]=(pesos[0]* Valor_FCusto[i] + pesos[1]*(pesos[2]* Valor_FLeadTime[i]+
        pesos[3]*Valor_FAvarias[i]) + pesos[4]*Valor_FEmissao[i])* Linhas_Oferta[i]/l;
}

// ALGORITMO PRINCIPAL (BUSCA DO MAXIMO VALOR): RETORNA UM CONJUNTO DE
// OFERTAS VENCEDORAS QUE MAXIMIZAM O VALOR

Limitante_Inferior= 0; // Inicializa o limitante inferior
cont_alocados=0; // inicializa o contador de ofertas alocadas
valor=new double[n_ofertas][n_fornecedores]; // Cria variável de custo da oferta
Stopmask = new String[n_ofertas][n_linhas]; // Cria variável da árvore de ofertas, a qual possui três
// valores para cada linha
valor_maximo=new double[n_ofertas]; // Cria variável de custo mínimo da oferta

// INICIALIZANDO ÁRVORE DE BUSCA DO MÁXIMO VALOR

```

```

while(true){

    // 1.
    bt.DepthFirstSearch();           // Chama o MÉTODO DEPTH FIRST SEARCH, Função principal do
                                     // algoritmo que realiza o IDA* e a busca em profundidade

    // 2.
    if(fim==1){ // Retorna os ganhadores (FIM DO ALGORITMO)
        return;
    }

    // 3.
    if(ajuste_f_limite==0){          // Condição que permite ajustar o limitante inferior
        Limitante_Inferior= Novo_f;
    }

} // Fim while principal

} // Fim static main

```

// MÉTODO DO DEPTH FIRST SEARCH (FUNÇÃO PRINCIPAL)

```

void DepthFirstSearch(){           // devolve linhas livres, propostas ganhadoras e valor atual do
                                   // caminho
    Caminho=new double[n_linhas];
    while(cont_alocados<n_ofertas){

        // 1 ATUALIZAR VARIÁVEIS

        if(cont_alocados==0){
            nova_busca=0;
            for(j=0; j<n_linhas; j++){
                if(j==0)
                    Stopmask[cont_alocados][j]="MUST";
                else
                    Stopmask[cont_alocados][j]="ANY";
            }
        }
        Contribuição_maxima= Valor_Estimado(); // Contribuição das propostas que estão em espera
                                                // (MÉTODO DO VALOR ESTIMADO)
    }

```

// 2 AVALIAR CONDIÇÃO PARA INSERIR A PROPOSTA

```

if((Valor_Caminho + Contribuição_maxima)< Limitante_Inferior){ // Comparação com o
                                                                // limitante inferior

    Winners=null;
    Novo_f=Valor_Caminho+Contribuição_maxima; // atualização do limitante
    return; // Para sair da função com valores de retorno
}

else{

```

// 3 VISITAR ÁRVORE DE PROPOSTAS

```

// Classificar linhas dentro das variáveis Stopmask
if(cont_alocados>0){
    cont_linhas2=0;
    for(j=0; j<n_linhas; j++){

```

```

        if(Linha_Espera[j]==0){
            Stopmask[cont_alocados][j]="BLOCKED";
        }
        else if(cont_linhas2==0 && Stopmask[cont_alocados-1][j].compareTo("BLOCKED")<0){
            Stopmask[cont_alocados][j]="MUST";
            cont_linhas2=1;
        }
        else if(Stopmask[cont_alocados-1][j].compareTo("BLOCKED")<0) {
            Stopmask[cont_alocados][j]="ANY";
        }
        else{
            Stopmask[cont_alocados][j]="BLOCKED";
        }
    }
}

```

```

cont_linhas2 = 0;
for(k=0;k<n_linhas; k++){
    if(Stopmask[cont_alocados][k].equals("BLOCKED"))
        cont_linhas2++;
}

```

// Chama o MÉTODO DE VISITAÇÃO

```

if(cont_linhas2 != n_linhas){
    k=0;
    novavisitação=false;
    Visitação(raiz_arvore);

    while(novavisitação==true){
        k=0;
        novavisitação=false;
        Visitação(raiz_arvore);
        if(raiz_arvore.right.visitado==true)
            novavisitação=false;
        j=0;
        while(j<n_linhas && Caminho[j]==0){
            j++;
        }
    }
} // fim if

```

//NÃO HÁ MAIS NÓS PARA PERCORRER, CAMINHO ENCONTRADO:

```

else{
    ajuste_f_limite=1;    // O F_limite foi ajustado, caminho encontrado.
    Limitante_Inferior= Valor_Caminho; // Atualização do upper bound
    Novo_f=Valor_Caminho;    // Atualização do novo limite, variável auxiliar
    Nó_BMC2=Nó_BMC;
}

```

// Guarda o caminho encontrado no vetor Winners

```

if(Nó_BMC2.children==null){
    Winners = null;
    Winners = new ArrayList<double[]>();
    while(Nó_BMC2.element!=null){
        Winners.add(Nó_BMC2.element);
        Nó_BMC2=Nó_BMC2.pai;
    }
}
return; // Sai da função Depth First Search
}

```

//4 ATUALIZAÇÃO E BACKTRACKING SE A PROPOSTA NÃO FOI ENCONTRADA

```

// ENCONTROU CAMINHO?
j=0;
while(j<n_linhas && Caminho[j]==0){           // Faça enquanto o vetor Caminho seja nulo
j++;
}

// SE NÃO ENCONTROU CAMINHO
if(j==n_linhas){

// Chama os MÉTODOS DE BACKTRACKING
BacktrackStopmask(Stopmask);
raiz_arvore2=raiz_arvore;k=0;
Backtracking(raiz_arvore2,Caminho);

// Chama o MÉTODO PARA DESBLOQUEAR AS PROPOSTAS
raiz_arvore3=raiz_arvore;
for(i=0; i<n_linhas; i++){
    if(Linha_Espera[i] == 0 && raiz_arvore3.right!=null)
        raiz_arvore3 = raiz_arvore3.right;
    else if(Linha_Espera[i] == 1 && raiz_arvore3.right!=null){
        raiz_arvore3 = raiz_arvore3.right;
        DesbloquearBid(raiz_arvore3);
        break;
    }
}

// Chama o MÉTODO DE ATUALIZAÇÃO DE VALOR;
cont_linhas2=1;
encontra_filho=1;
Atualizo_valor_solução();
}

// SE ENCONTROU FILHO FAÇA:
if(j!=n_linhas) {
    encontra_filho=0;cont_linhas2=0;
    Atualizo_valor_solução();
}

```

// 5 LOOP PARA CONTINUAR BUSCA RECURSIVA

//PARA CADA BID QUE SEJA FILHO DO ATUAL NÓ, FAZER:

```

// SE FOR A ÚLTIMA OFERTA QUE TEM A LINHA 1:
if(raiz_arvore.left.visitado==true){ // Continua a última busca depois que a última oferta da
                                     árvore foi alocada

    if(ultimo==1){
        this.DepthFirstSearch(); // Chama uma vez a função principal DFS
        ultimo=0;
    }
    else{
        while( Stopmask[cont_alocados][0].compareTo("MUST")<0){
            this.DepthFirstSearch();
            Nó_BMC2=Nó_BMC;
            if( Winners!=null && Nó_BMC2.children==null && Novo_f > VALOR_MAXIMO){
                Melhores_i_alocados = null;
                Melhores_i_alocados = new ArrayList<double[]>();
                while(Nó_BMC2.element!=null){
                    Melhores_i_alocados.add(Nó_BMC2.element);
                }
            }
        }
    }
}

```

```

        Nó_BMC2=Nó_BMC2.pai;
    }
    VALOR_MAXIMO = Novo_f;
}
return;
}
}
} // Fim While

while(raiz_arvore.left.visitado==false){ // Enquanto existam ofertas não exploradas do lado esquerdo

//A)
this.DepthFirstSearch(); // Chama função principal DFS

    if(raiz_arvore.left.visitado==false ){

//B)
        // Compara o valor do último caminho encontrado com o valor do melhor alocado, se for menor o
        último caminho passa a ser o melhor alocado
        Nó_BMC2=Nó_BMC;
        if( Winners!=null && Nó_BMC2.children==null && Novo_f > VALOR_MAXIMO){
            Melhores_i_alocados = null;
            Melhores_i_alocados = new ArrayList<double[]>();
            while(Nó_BMC2.element!=null){
                Melhores_i_alocados.add(Nó_BMC2.element);
                Nó_BMC2=Nó_BMC2.pai;
            }
            VALOR_MAXIMO = Novo_f;
        }

//C)
        // Selecionar o valor máximo entre o próximo limitante e o atual
        if(Proximo_f > Novo_f)
            Proximo_f=Novo_f;

        // Chama o MÉTODO DE BACKTRACKING
        BactrackStopmask(Stopmask);
        raiz_arvore2=raiz_arvore;k=0;
        Backtracking(raiz_arvore2,Caminho);

        // chama o MÉTODO PARA DESBLOQUEAR PROPOSTAS
        raiz_arvore3=raiz_arvore;
        for(i=0; i<n_linhas; i++){
            if(Linha_Espera[i] == 0 && raiz_arvore3.right!=null){
                raiz_arvore3 = raiz_arvore3.right;
            }
            else if(Linha_Espera[i] == 1 && raiz_arvore3.right!=null){
                raiz_arvore3 = raiz_arvore3.right;
                DesbloquearBid(raiz_arvore3);
                break;
            }
        }
    }

    // Condição que recomeça a busca em caso de ainda não ter alocado nenhum caminho
    if(Winners==null&&Melhores_i_alocados==null){
        Valor_Caminho=0;
        raiz_BMC=null;
        raiz_BMC=new Busca_Principal(null);
        Nó_BMC=raiz_BMC;
        DesbloquearBid(raiz_arvore);
    }
}

```

```

        return;
    }

} // fim do if

// Chama o MÉTODO DE ATUALIZAÇÃO DE VALOR
cont_linhas2=1;
encontra_filho=1;
Atualizo_valor_solução();

} // Fim while

// 6 RETORNA SOLUÇÃO ENCONTRADA

// Se encontrarmos solução final então retorne a solução e seu valor
if(Melhores_i_alocados!=null){
    fim=1;
    if (flag==0){
        valores(raiz_arvore);
        flag=1;
    }
    return ;
}
else{
    Winners=null;
    Novo_f=Proximo_f;
    return ;
}

} // Fim do else filhos

} // fim else

} // fim while

} // fim DepthFirstSearch

} // Fim public class Multicriterio

```