4
Algoritmos

4.1
Pré-Processamento

Alguns pré-processamentos são implementados para eliminar restrições e diminuir o número de variáveis do problema. A idéia é realizar simplificações e/ou eliminações de forma a reduzir ao máximo o tamanho do problema sem comprometer as soluções do problema.

4.1.1
Simplificações

A formulação para o problema apresentado baseia-se em um modelo de fluxos sobre uma rede espaço e tempo que controla a movimentação de veículos e cargas variadas em uma malha rodoviária.

Instâncias de interesse, presentes em transportadoras de grande porte no país, possuem malhas rodoviárias que são compostas por um número pequeno de centros de distribuição ($|C| \leq 25$), por um número médio de veículos ($5 \leq |V| \leq 200$) e por um número grande de demandas ($|D| \geq 200$) e instantes de tempo ($|T| \geq 700$).

Tal número elevado de instantes de tempo para as instâncias testadas se deve ao fato de que eventos de chegada e saída de veículos nos centros de distribuição podem ocorrer a qualquer momento ao longo do período de planejamento considerado. Com isso, o tamanho do modelo de programação inteira apresentado torna-se muito grande.

Portanto, uma forma de reduzir consideravelmente o número de variáveis e restrições do problema é através da eliminação de instantes de tempo, considerando apenas momentos realmente relevantes para o problema.

Teoricamente, veículos podem deixar os centros de distribuição em qualquer momento do dia. Entretanto, o que ocorre na prática é que, por questões de segurança e organização, costumam ser definidos instantes diários de saída de veículos dos centros de distribuição.

Portanto, definiu-se que CD's possuem até três instantes diários de saída de veículos. Este número varia de acordo com as necessidades de cada CD.

Além disso, centros de distribuição costumam possuir ao menos um dia na semana em que não há quantidade considerável de demanda que justifique a saída de veículos dos centros de distribuição existentes, o que diminui ainda mais os instantes de saída de veículos a serem considerados.

4.1.2 Eliminação de Vértices de Grau 2

O procedimento de pré-processamento aqui descrito consiste em eliminar vértices intermediários de grau dois.

Em particular, para o PPA, a técnica de pré-processamento foi utilizada para eliminar arcos de deslocamento do fluxo de veículos considerado para o problema.

O que ocorre é que nem todos os centros de distribuição que compõem as rotas disponíveis representam pontos onde podem ocorrer operações de reembarque. Ou seja, pode-se dizer que nestes CD's, mercadorias não são descarregadas de um veículo para serem carregadas em outros.

Considerando esta situação, variáveis de deslocamento de veículos e de movimentação de carga nestes veículos podem ser eliminadas.

Na figura 4.1 nós hachurados representam CD's no tempo onde não são permitidas operações de reembarque. Tais vértices e arestas associadas são substituídos por uma única aresta representando o deslocamento do veículo. Na figura, os dois vértices hachurados e as respectivas arestas associadas são substituídos por duas arestas de deslocamento do veículo, diminuindo o número de variáveis do problema.

A figura 4.3 exibe uma legenda para que seja possível identificar as variáveis presentes na formulação que fazem parte do fluxo ilustrado.

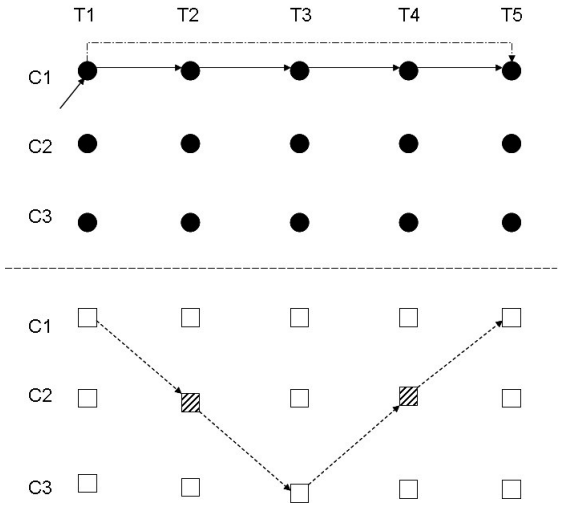

Figura 4.1: Eliminação de variáveis de deslocamento de veículos.

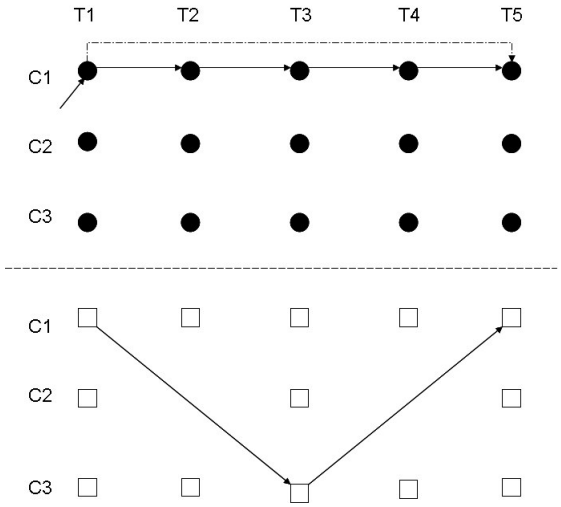

Figura 4.2: Fluxo de veículos após eliminação de variáveis de deslocamento.

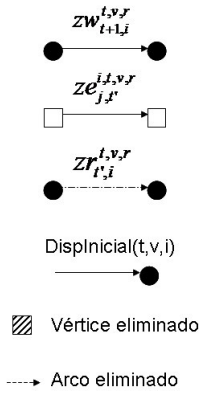

Figura 4.3: Legenda

4.1.3 Eliminação de Arcos de Movimentação de Demandas

Outra técnica de pré-processamento experimentada, tenta eliminar arestas de deslocamento de demandas do fluxo de carga descrito na formulação do problema. O objetivo é encontrar arcos de deslocamento de demandas no tempo $(i, j) \in A$ que certamente não sejam utilizados pelos veículos disponíveis na otimização.

Dado um grafo representando a malha rodoviária considerada no tempo, onde vértices representam centros de distribuição no tempo e arcos representam deslocamentos das demandas, a idéia é verificar se existe caminho desde a origem de uma demanda em seu instante de coleta, até a origem do arco de deslocamento (i, j), e se existe um caminho desde o destino do arco até o destino da demanda em seu instante limite para atendimento. Caso não

exista ao menos um dos caminhos considerados, podemos eliminar o arco (i, j) correspondente, sem prejudicar a solução do problema.

Na figura 4.4 o deslocamento de uma demanda no arco (i, j) é considerado. Portanto, caso não exista um caminho da origem da demanda até i ou um caminho de j até o destino da demanda, o arco (i, j) pode ser eliminado.

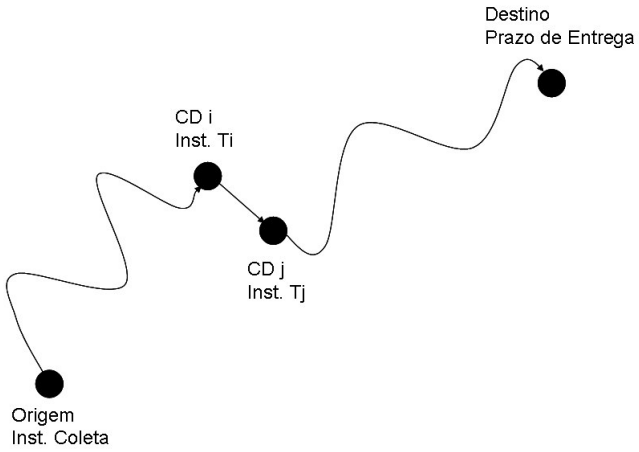

Figura 4.4: Eliminação de variáveis de deslocamento de demandas.

Dondo, Mendez e Cerda (DMC03) apresentam um framework para o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos e propõem regras para eliminações baseadas em janelas de tempo seguindo a mesma idéia.

O método descrito acima foi implementado utilizando-se o algoritmo de Dijkstra no grafo espaço/tempo construído, considerando as possibilidades de deslocamento das demandas existentes. Desta forma, foi possível identificar se existe ou não caminho entre pares de vértices do grafo.

4.2 Estratégias de Resolução

Três estratégias de resolução para o PPA-VE foram consideradas.

As três estratégias utilizam funcionalidades implementadas no ILOG CPLEX para resolver o problema de programação linear inteira gerado. Trata-se de uma implementação para o algoritmo de Branch-and-Cut e da implementação de um heurística usada para melhorar o desempenho sobre certos tipos de problemas chamada *Polishing*.

O *Polishing* normalmente é usado com o objetivo de melhorar a solução obtida ao final da execução do algoritmo de Branch-and-Cut, caso a otimalidade não tenha sido comprovada. Tal heurística pode ainda ser usada em substituição ao procedimento de Branch-and-Cut caso uma solução para o problema inteiro possa ser encontrada na raiz da árvore de busca.

A heurística implementa uma abordagem evolucionária para melhorar soluções de modelos de programação inteira. O método realiza mutações e combinações de soluções do problema, ambas obtidas com um LNS (Large Neighborhood Search). Estas técnicas são integradas com o algoritmo de Branch-and-Bound. A solução resultante da heurística *Polishing* freqüentemente encontra melhorias para modelos MIP difíceis.

Implementações de Local Branching, (FL03) Fischetti e Lodi, também foram testadas para o problema. Entretanto, não foram obtidos bons resultados com esta abordagema, visto que a resolução da relaxação linear para cada iteração do algoritmo mostrou-se demorada. A heurística implementada no *Polishing* apresentou melhor desempenho para as instâncias utilizadas.

4.2.1 Estratégia 1

A primeira estratégia de resolução executa o algoritmo de pré-processamento descrito na seção 4.1 e simplesmente executa o algoritmo de Branch-and-Cut seguido por uma etapa de execução do *Polishing*, caso a otimalidade da solução até então obtida não tenha sido comprovada.

Testes realizados para esta primeira estratégia geraram bons resultados computacionais. Entretanto, estes sugerem a contratação de mais veículos extras do que o normal. Com isso, definiu-se uma estratégia de resolução em que inicialmente, apenas veículos próprios são considerados e veículos extras são utilizados apenas para transportar demandas não atendidas por veículos próprios (Estratégia 2).

As duas estratégias de resolução seguintes sugerem decompor o PPA-VE.

Algoritmo 1 - Estratégia de Resolução para o PPA-VE

pre-processamento()
$p \leftarrow criaPPA\text{-}VE()$
$sol \leftarrow p.Branch\text{-}and\text{-}Cut()$
if *sol* não é ótima **then**
  $sol \leftarrow p.Polishing(sol)$
end if
return *sol*

4.2.2 Estratégia 2

A segunda estratégia de resolução propõe resolver o PPA e, em uma segunda etapa, considerar a contratação de veículos extras (PPA-VE). Tal abordagem permitiu a resolução de instâncias mais complexas do problema.

Como forma de considerar mercadorias que por ventura não possam vir a ser atendidas por veículos próprios, veículos extras devem ser criados.

Portanto, após uma etapa de otimização considerando apenas veículos próprios (resolução do PPA), variáveis e restrições para veículos extras são criadas. Variáveis de decisão de alocação de veículos próprios a rotas são fixadas e variáveis de carregamento de demandas são limitadas inferiormente de forma a garantir que decisões de carregamento já realizadas não sejam replanejadas. Desta forma, veículos extras são utilizados apenas com objetivo de transportar mercadorias até então não atendidas por veículos próprios.

Considerando as alterações realizadas no problema, uma nova etapa de otimização é realizada. Assim, ao término da otimização, temos uma sugestão de contratação de veículos extras que podem vir a realizar o transporte de demandas antes não atendidas por veículos próprios. Veículos extras podem atender demandas desde suas origens até seus destinos ou podem compor o transporte de cargas junto com outros veículos considerados na otimização.

Tal estratégia de resolução implementada executa o algoritmo de pré-processamento descrito na seção 4.1, e chama o algoritmo de *Branch-and-Cut* seguido pela execução do *Polishing*, caso a otimalidade da solução até então obtida não tenha sido comprovada. Por fim, um novo problema de otimização é criado e resolvido através do algoritmo de *Branch-and-Cut* e etapas de *Polishing*.

A figura 4.5 ilustra a movimentação de um veículo próprio e de três veículos extras entre os centros de distribuição existentes. No exemplo, veículos extras realizam o transporte de demandas para que estas sejam reembarcadas no veículo próprio considerado. Ou seja, veículos extras são utilizados para compor o transporte de demandas antes não atendidas junto com o veículo próprio considerado.

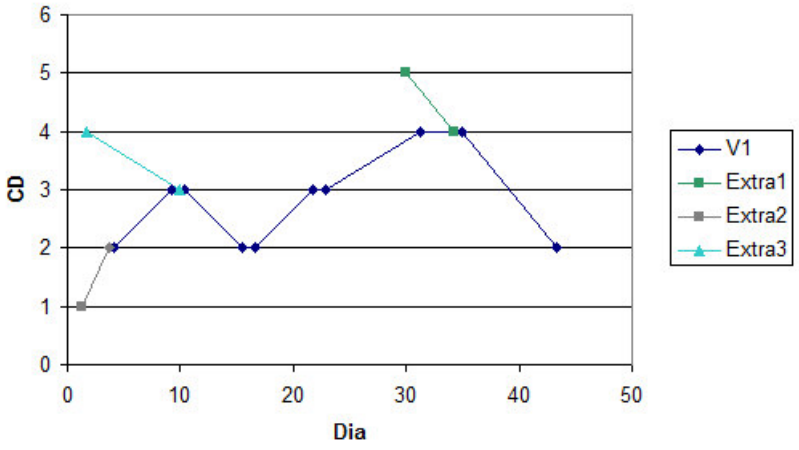

Figura 4.5: Movimentação de veículos extras.

Algoritmo 2 - Estratégia de Resolução para o PPA-VE

$pre\text{-}processamento()$
$p \leftarrow criaPPA()$
$sol \leftarrow p.Branch\text{-}and\text{-}Cut()$
if sol não é ótima **then**
  $sol \leftarrow p.Polishing(sol)$
end if
$p \leftarrow criaPPA\text{-}VE(p, sol)$
$sol \leftarrow p.Branch\text{-}and\text{-}Cut()$
if sol é ótima **then**
  return sol
end if
$sol \leftarrow p.Polishing(sol)$
return sol

4.2.3 Estratégia 3

A terceira estratégia de resolução proposta sugere decompor ainda mais ou problema.

Considerando os testes realizados e a análise dos resultados obtidos utilizando a Estratégia 2, foi possível notar que, normalmente, a primeira solução viável obtida para o problema é muito ruim.

Com o objetivo de reduzir a razão entre tal solução inteira inicial para o problema e a relaxação linear correspondente, *gap* inicial, a estratégia de resolução passou a ser dividida em duas etapas.

Inicialmente, estamos interessados apenas em obter uma solução viável inicial para o problema. Para isso, definimos um novo problema PPA-Simples.

O PPA-Simples trata de uma simplificação para o PPA. O novo problema se resume a maximizar o atendimento das demandas sem se preocupar com os custos de transporte, considerando todas as restrições apresentadas no PPA.

Portanto, a função objetivo para o PPA-Simples fica como segue:

$$Min \sum_{d \in D(i)} vf_{Origem(d),d}^{InstColeta(d)} \tag{4-1}$$

onde $vf_{Origem(d),d}^{InstColeta(d)}$ representa a quantidade não atendida da demanda $d \in D$ coletada no instante $InstColeta(d)$ no centro de distribuição $Origem(d)$.

Uma solução ótima para o PPA-Simples S, normalmente se traduz em uma boa solução inicial para o PPA.

Após obter uma solução para o PPA-Simples gastando no máximo 20 minutos de execução, adicionamos à formulação proposta para o PPA uma desigualdade que simplesmente garante uma quantidade mínima em peso não atendido na solução do problema.

$$\sum_{d \in D} vf^{InstColeta(d)}_{Origem(d),d} \geq objVal(S) \tag{4-2}$$

onde $objVal(S)$ é o valor da função objetivo do PPA-Simples associada à solução S. Ou seja, garantimos com essa desigualdade que ao menos $objVal(S)$ toneladas não poderão ser atendidas na solução final. Observe que esta desigualdade é válida para o PPA somente se objVal(S) é a solução ótima de PPA-Simples.

Tal estratégia reduziu o *gap* inicial obtido para o PPA e melhorou consideravelmente os resultados obtidos para instâncias mais complexas do problema.

Algoritmo 3 - Estratégia de Resolução para o PPA-VE

pre-processamento()
$p \leftarrow criaPPA\text{-}Simples()$
$sol \leftarrow p.Branch\text{-}and\text{-}Cut()$
if *sol* não é ótima **then**
  $sol \leftarrow p.Polishing(sol)$
end if
$p \leftarrow criaPPA(sol)$
Adiciona desigualdades válidas ao problema,
considerando a solução obtida.
$sol \leftarrow p.Branch\text{-}and\text{-}Cut()$
if *sol* não é ótima **then**
  $sol \leftarrow p.Polishing(sol)$
end if
$p \leftarrow criaPPA\text{-}VE(p, sol)$
$sol \leftarrow p.Branch\text{-}and\text{-}Cut()$
if *sol* é ótima **then**
  return *sol*
end if
$sol \leftarrow p.Polishing(sol)$
return *sol*

4.3

Eliminação de Simetria

Visando fortalecer as formulações descritas no capítulo 3, eliminações de simetria são aqui propostas como forma de reduzir o espaço de busca do problema apresentado.

Dada uma solução viável para o problema, muito freqüentemente podemos obter soluções equivalentes por simetria. Basta encontrarmos dois veículos com as mesmas características e realizar uma troca de rotas entre eles. Portanto, dependendo do número de veículos "equivalentes" de uma instância, podemos ter mais ou menos soluções equivalentes por simetria.

Benavent, Corberán, Sanchis e Plana (BCSP07) utilizam cortes de simetria sugeridos por Gendreau (Gendreau); Ghiani, Laganá, Laporte e Musmanno (GLLM00) seguindo idéias parecidas.

Com base nessas idéias, veículos são considerados equivalentes se possuem os mesmos limites para carregamento e se cobram os mesmos custos para a realização de rotas.

As duas desigualdades propostas para o problema aqui estudado estabelecem uma "ordenação" de veículos equivalentes.

Para a primeira desigualdade proposta, a idéia é permitir a utilização de determinado veículo somente caso todos os outros veículos equivalentes e anteriores na ordenação proposta não estejam disponíveis. Ou seja, a idéia é eliminar soluções simétricas no problema onde uma simples troca de veículos equivalentes gere uma nova solução de mesmo valor na função objetivo.

É importante considerar que esta desigualdade não deve ser aplicada para o primeiro veículo na ordenação de veículos equivalentes proposta.

$$N.zr_{t_2,i}^{t_1,v,r} \leq \sum_{v' \in \{1,\ldots,v-1\}} \sum_{\substack{r' \in R(v') \\ t_3 \leq t_1 \text{ e } t_4 > t_1}} zr_{t_4,i}^{t_3,v',r'}; \;\; \forall t_1 \in T, \forall i \in C, \forall v \in V(i) \backslash \{1\}, \forall r \in R(v),$$

onde N é o número de veículos que aparecem no lado direito da desigualdade;

$t_1 = InstInicio(r)$, $t_2 = InstFim(r)$, $t_3 = InstInicio(r')$

e $t_4 = InstFim(r')$ (4-3)

Na figura 4.6 arcos representam realizações de rotas por veículos. A idéia é que um veículo v pode realizar uma rota $r \in R(v)$ apenas se todos os veículos equivalentes e anteriores a ele na ordenação estabelecida não estejam disponíveis no momento. Ou seja, nenhum destes veículos pode estar disponível para realizar rotas no lugar de v.

Na figura 4.6 v e v' são veículos equivalentes. O veículo v pode iniciar a rota $r \in R(v)$ no instante $T2$ apenas se v' estiver ocupado neste instante.

Ou seja, apenas se v' estiver realizando uma dentre as suas opções de rota na figura.

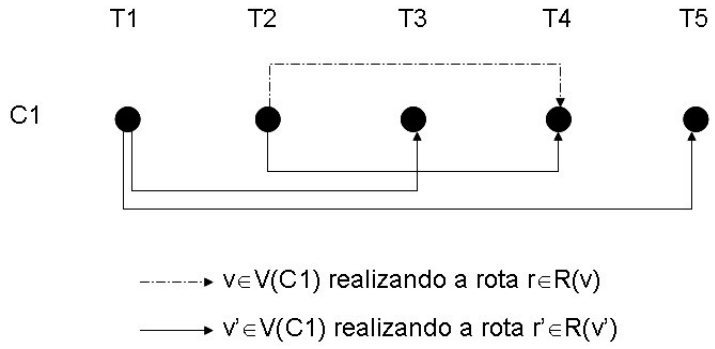

Figura 4.6: Cortes de simetria

Outra idéia seria considerar a ordenação de veículos equivalentes proposta de forma a garantir que um veículo seja mais utilizado que veículos a seguir na ordenação. A restrição a seguir garante que o total em quilômetros percorridos por um veículo v_1, sendo v_2 anterior a v_1 na ordenação de veículos equivalentes, deve ser menor ou igual ao total percorrido por v_2.

$$\sum_{t\in T}\sum_{r\in R(v_1)} Distancia(r).zr_i^{t,v_1,r} \leq \sum_{t\in T}\sum_{r\in R(v_2)} Distancia(r).zr_i^{t,v_2,r};$$
$$\forall i \in C, \forall v_1, v_2 \in V \text{com } v_1 \leq v_2 \text{ e } v_1 \text{ equivalente a } v_2. \quad (4\text{-}4)$$