6
Problema de Roteamento de Veículos sobre Arcos

O problema de roteamento de veículos sobre os arcos de uma malha viária, com restrição de capacidade, mais conhecido pela sua sigla em inglês *CARP* (*Capacitated Arc Routing Problem*), refere-se à determinação de rotas para os veículos de uma frota, para atender demandas em clientes distribuídos ao longo dos arcos. Cada veículo deve começar e encerrar sua rota no depósito que é a sua base. O objetivo é encontrar rotas que atendam todos os clientes, satisfazendo a capacidade dos respectivos veículos, e tais que os custos totais do transporte sejam minimizados.

Este problema, proposto por Golden e Wong em 1981 [55], pertence à classe dos problemas de otimização combinatória $\mathcal{NP}$-difíceis. Ele contém como casos especiais o problema do carteiro rural (*RPP - Rural Postman Problem*) e o problema do carteiro chinês com restrição de capacidade (*CCPP - Capacitated Chinese Postman Problem*), que também pertencem à classe $\mathcal{NP}$-difícil [75, 51].

Modelos e algoritmos para a resolução do *CARP* são descritos em um livro editado por Dror [42]. Diferentes heurísticas foram propostas por Golden et al. [56], Chapleau et al. [28], Ulusoy [105], Pearn [90, 91] e Hertz et al. [63]. Diversas outras heurísticas são descritas em [11], [43] e [42]. Algoritmos exatos foram propostos por Hirabayashi, Saruwatari e Nishida [64], Kiuchi et al. [68], Welz [115], e Belenguer e Benavent [19]. É importante observar que tais algoritmos, em geral, não conseguem garantir a otimalidade de soluções nos casos em que a instâncias contém mais de 30 clientes com demandas.

Diversas alternativas para o cálculo de estimativas inferiores para o valor da solução ótima de instâncias do *CARP* foram propostas por Golden e Wong [55] (*MLB – Matching Lower Bound*), Assad et al. [9] (*NSLB – Node Scanning Lower Bound*), Pearn [89] (*MNSLB – Matching Node-Scanning Lower Bound*), Hirabayashi et al. [64] (*NDLB – Node Duplication Lower Bound*), Benavent et al. [22] (*LB1* e *LB2*), Amberg e Voß [4] (*HRLB – Hierarchical Relaxations Lower Bound*) e Wøhlk [117] (*MCNDLB – Multiple*

PUC-Rio - Certificação Digital Nº 0024952/CA

Cuts Node Duplication Lower Bound).

Contudo, as melhores estimativas para as soluções ótimas de um grande conjunto de instâncias disponíveis na literatura são aquelas encontradas por Belenguer e Benavent [20]. Em 47 instâncias, de um conjunto de 87, essas estimativas igualam-se aos valores das melhores soluções heurísticas previamente encontradas, com o objetivo de minimizar a distância total a ser percorrida. Entretanto, para estas estimativas, ainda não se conhece uma forma de utilizá-las em um algoritmo *branch-and-bound*, não constituindo portanto uma formulação para o *CARP*. Esta discussão é apresentada na seção 6.5.

A abordagem descrita neste capítulo, transformação do *CARP* ao *CVRP* e uso de um algoritmo *branch-and-cut-and-price*, não somente consegue estimativas melhores do que as listadas em [20], como também consegue encontrar as soluções ótimas para diversas dessas instâncias.

Inicialmente é mostrada na seção 6.2 a transformação do *CARP* no *CVRP* proposta por Pearn, Assad e Golden em [88]. Nesta transformação, a instância do $CVRP$ resultante tem $3r + 1$ vértices, onde r é o número de arestas da instância do $CARP$ com demandas associadas. Contudo, as classes de instâncias mais comumente usadas em testes de algoritmos para o $CARP$ são formadas por instâncias com um mínimo de 15 e um máximo de 190 arestas com demandas (veja as tabelas 6.2, 6.3 e 6.4 para detalhes dessas instâncias). A utilização desta abordagem é praticamente inviável, uma vez que os melhores códigos atualmente disponíveis para o *CVRP* [24, 80, 116, 96, 50] somente resolvem de forma consistente instâncias com no máximo 135 vértices. Assim, é proposta na seção 6.3 uma nova transformação que, embora gere uma instância do $CVRP$ com $2r + 1$ vértices, permite ao algoritmo *branch-and-cut-and-price* utilizado (veja seção 6.6) comportar-se como se estivesse resolvendo uma instância com apenas $r + 1$ vértices. Os resultados obtidos com essa abordagem são mostrados na seção 6.7.

6.1 Modelagem do *CARP*

O *CARP*, como descrito na seção 6, pode ser modelado por um grafo conexo não orientado $H = (N, A)$. O conjunto $N = \{v_0, v_1, \ldots, v_n\}$ representa o depósito (v_0) e os clientes ($v_1, \ldots, v_n$). O conjunto $A = \{(v_i, v_j) \mid v_i, v_j \in N, i < j\}$ modela a malha viária associada ao problema. A cada aresta $e \in A$ é associado um custo $\ell_e \in \mathbb{Z}^+$ de percorrê-la. As arestas

PUC-Rio - Certificação Digital Nº 0024952/CA

que representam os clientes, com demandas $w_e \in \mathbb{Z}^+$ a serem atendidas, são definidas por $R = \{e \in A \mid w_e > 0\}$ e $r = |R|$ é o número de tais arestas. O depósito é a base de k veículos de idêntica capacidade $D \geq \max\{w_e \,|\, e \in R\}$ e o ponto de início e término de todas as rotas.

Seja F um conjunto de caminhos fechados, com início e término no depósito, onde as arestas de cada caminho podem ser tanto atendidas (veículo atende a demanda) quanto ignoradas (veículo apenas passa pela aresta). Observe que toda aresta $e \in R$ deve ser atendida por um dos caminhos. Assim, o conjunto F é uma solução factível para o *CARP* se:

- cada aresta $e \in R$ é atendida por um único caminho de F; e
- a soma das demandas atendidas em cada caminho fechado em F não excede a capacidade D.

O objetivo é encontrar uma solução que minimize a soma dos custos dos caminhos a serem percorridos. O valor dado por $\sum_{e \in R} \ell_e$ é uma estimativa inferior trivial no custo de uma solução ótima. A diferença entre esses valores é o custo das arestas ignoradas em cada caminho.

PUC-Rio - Certificação Digital Nº 0024952/CA

6.2 Transformação de Pearn, Assad e Golden

Uma transformação do *CARP* no *CVRP*, que descarta o conjunto de vértices $N - \{v_0\}$ e substitui cada aresta de R por três novos vértices, foi proposta por Pearn, Assad e Golden em [88]. Nessa transformação uma aresta $(v_i, v_j) \in R$ é associada aos vértices s_{ij} e s_{ji} (vértices laterais) e m_{ij} (vértice central). O vértice v_0 é mantido como o depósito e a instância do *CVRP* é então definida no grafo completo não orientado $G = (V, E)$, onde

$$V = \bigcup_{(v_i,v_j) \in R} \{s_{ij}, m_{ij}, s_{ji}\} \bigcup \{v_0\}\,. \tag{6-1}$$

Cada demanda $w_{i,j}$ é distribuída entre os novos vértices s_{ij}, m_{ij} e s_{ji} que substituem a aresta (v_i, v_j). Esta distribuição pode ser arbitrária, desde que satisfaça a seguinte igualdade ($d : V \to \mathbb{Z}^+$):

$$d(s_{ij}) + d(m_{ij}) + d(s_{ji}) = w_{i,j}. \tag{6-2}$$

Os custos $c : E \to \mathbb{Z}^+$ são definidos como:

$$c(s_{ij}, s_{kl}) \;=\; \begin{cases} \alpha.(\ell_{ij} + \ell_{kl}) + dist(v_i, v_k) & \text{se } (i,j) \neq (k,l) \\ 0 & \text{se } (i,j) = (k,l) \end{cases} \tag{6-3}$$

$$c(v_0, s_{ij}) = \alpha.\ell_{ij} + dist(v_0, v_i) \tag{6-4}$$

$$c(m_{ij}, v) = \begin{cases} \beta.\ell_{ij} & \text{se } v = s_{ij} \text{ ou } v = s_{ji} \\ \infty & \text{caso contrário,} \end{cases} \tag{6-5}$$

onde $dist(v_i, v_j)$ é o valor do caminho de custo mínimo do vértice v_i ao vértice v_j no grafo H, calculados com os custos ℓ. Essa transformação será válida se os fatores α e β satisfizerem $2.(\alpha + \beta) = 1$ e $\alpha, \beta > 0$. A figura 6.1 mostra essa transformação para uma aresta $(i, j) \in R$, usando-se fatores $\alpha = \beta = \frac{1}{4}$.

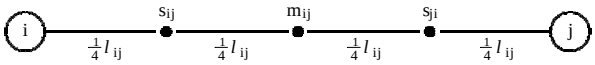

Figura 6.1: *CARP* – Transformação de Pearn, Assad e Golden.

O significado da definição 6-3 é que ao substituir as arestas $(v_i, v_j) \in A$ e $(v_k, v_l) \in A$ pelos conjuntos de vértices $\{s_{ij}, m_{ij}, s_{ji}\}$ e $\{s_{kl}, m_{kl}, s_{lk}\}$, respectivamente, o menor caminho entre os vértices laterais s_{ji} e s_{kl} seria composto pela aresta de s_{ji} ao vértice v_j, as arestas que compõem o menor caminho entre v_j e v_k e a aresta de v_k a s_{kl} (veja figura 6.2). Uma interpretação similar é fornecida pela definição 6-4, para o deposito v_0 e um vértice lateral s_{ij}. A definição 6-5 garante que, em qualquer solução factível de custo finito, o vértice central m_{ij} é acessível somente a partir dos vértices laterais s_{ij} e s_{ji}.

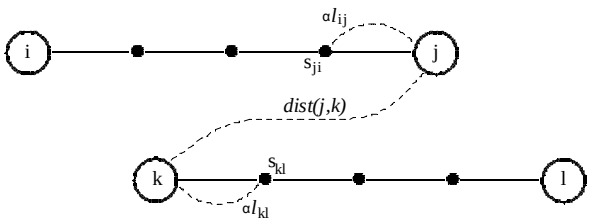

Figura 6.2: *CARP* – Custo do menor caminho entre dois vértices laterais.

A equivalência das duas instâncias ($CVRP$ e $CARP$) decorre de dois fatos: o primeiro é que qualquer solução factível para o $CARP$ pode ser transformada em uma solução factível para o $CVRP$ de custo não superior; e o segundo é que a solução ótima do $CVRP$ corresponde a uma solução do $CARP$ de igual custo.

Considere um caminho fechado C que ocorre em uma solução factível qualquer para o $CARP$. A troca de cada aresta $(v_i, v_j) \in C$, tal que a demanda $w_{ij} > 0$ deve ser atendida por C, por três vértices s_{ij}, m_{ij} e s_{ji}, faz com que os mesmos sejam visitados na mesma ordem que a aresta

PUC-Rio - Certificação Digital Nº 0024952/CA

(v_i, v_j) ($s_{ij} \rightarrow m_{ij} \rightarrow s_{ji}$ ou $s_{ji} \rightarrow m_{ij} \rightarrow s_{ij}$). O custo de ir do vértice s_{ij} ao vértice s_{ji} (ou vice-versa), passando por m_{ij}, é $2.\beta.\ell_{ij}$, enquanto que a fração do custo de chegar ao vértice s_{ij} e sair pelo vértice s_{ji} é de $2.\alpha.\ell_{ij}$. Como $2(\alpha + \beta) = 1$ e $\alpha, \beta > 0$, o custo continua sendo ℓ_{ij}.

Além disso, se (v_i, v_j) e (v_k, v_l), para $j \neq k$, são arestas com demandas a serem atendidas por C e separadas por uma ou mais arestas que não serão atendidas por C, o custo de ir de s_{ji} até s_{kl} é $\alpha.(\ell_{ij} + \ell_{kl}) + dist(j, k)$. O termo $dist(j, k)$ não excede o custo da aresta (v_j, v_k) no grafo H, enquanto os termos com o fator α estão incorporados nos custos das arestas (v_i, v_j) e (v_k, v_l), como descrito anteriormente. Portanto, qualquer solução factível do $CARP$ pode ser transformada em uma solução do $CVRP$ de custo não superior.

Por outro lado, a solução ótima do $CVRP$ corresponde a uma solução do $CARP$ de igual custo. Se C é agora um caminho fechado da solução ótima do $CVRP$, então qualquer vértice central m_{ij} deve sempre ser visitado na seqüência $s_{ij} \rightarrow m_{ij} \rightarrow s_{ji}$ ou $s_{ji} \rightarrow m_{ij} \rightarrow s_{ij}$. Essa restrição é imposta através do custo infinito atribuído às arestas que ligam os vértices m_{ij} aos demais vértices que não s_{ij} e s_{ji}). Assim, há uma correspondência biunívoca entre uma seqüência (s_{ij}, m_{ij}, s_{ji}) de um caminho fechado C da solução ótima do $CVRP$ e a aresta (v_i, v_j) de um caminho fechado C' de uma solução de igual custo do $CARP$.

6.3
Nova Transformação

Na transformação descrita na seção 6.2, o vértice central m_{ij} é inserido com o propósito apenas de garantir que uma rota do $CVRP$ passe nos três vértices correspondentes à aresta (v_i, v_j) em seqüência ($s_{ij} \rightarrow m_{ij} \rightarrow s_{ji}$ ou $s_{ji} \rightarrow m_{ij} \rightarrow s_{ij}$). Esse vértice central pode ser eliminado simplesmente obrigando-se que as r arestas que ligam os vértices s_{ij}, s_{ji} na instância do $CVRP$ resultante façam parte da solução. Em outras palavras, determinando-se previamente que r arestas deverão fazer parte das rotas que compõem as soluções válidas.

Esta idéia resulta numa nova transformação similar à descrita na seção 6.2, mas que reduz a instância do $CVRP$ a um grafo com $2r + 1$ vértices. Agora uma aresta $(v_i, v_j) \in R$ é associada a apenas dois vértices (s_{ij} e s_{ji}). A instância do $CVRP$ é definida no grafo completo não orientado $G = (V, E)$,

PUC-Rio - Certificação Digital Nº 0024952/CA

onde

$$V = \bigcup_{(v_i,v_j)\in R} \{s_{ij}, s_{ji}\} \bigcup \{v_0\}\,. \tag{6-6}$$

O vértice v_0 é também definido como o depósito. Uma demanda $w_{i,j}$ é distribuída de modo que $d(s_{ij}) + d(s_{ji}) = w_{i,j}$ $(d : V \to \mathbb{Z}^+)$ e os custos $c : E \to \mathbb{Z}^+$ são definidos como:

$$c(s_{ij}, s_{kl}) = \begin{cases} 0 & \text{se } (i,j) = (k,l) \\ \ell_{ij} & \text{se } (i,j) = (l,k) \\ dist(v_i, v_k) & \text{se } (i,j) \neq (k,l) \text{ e } (i,j) \neq (l,k) \end{cases} \tag{6-7}$$

$$c(v_0, s_{ij}) = dist(v_0, v_i)\,, \tag{6-8}$$

onde $dist(i,j)$ tem o mesmo significado que o descrito na seção 6.2.

Todas as arestas $\{(s_{ij}, s_{ji}) \,|\, (v_i, v_j) \in R\}$ podem ser previamente fixadas a 1, ou seja, somente são válidas soluções do *CVRP* que visitem s_{ij} e s_{ji} em seqüência ($s_{ij} \to s_{ji}$ ou $s_{ji} \to s_{ij}$). É importante ressaltar que obrigar r arestas a fazer parte da solução não acarreta em esforço computacional extra a muitos dos algoritmos para o *CVRP*, muito pelo contrário, a resolução da instância assim resultante é facilitada. Embora a instância resultante do *CVRP* esteja associada a um grafo com $|V| = 2.r+1$ vértices, a utilização obrigatória desse conjunto de r arestas pode fazer com que os algoritmos comportem-se como se a instância tivesse apenas $|V| = r + 1$ vértices. A seção 6.4 detalha este fato.

As figuras 6.3 à 6.5 exemplificam a transformação de uma instância do *CARP* a uma instância do *CVRP*. O grafo associado à instância do *CARP*, com 4 vértices ($N = \{0, 1, 2, 3\}$) e 5 arestas (todas com demandas), é mostrado na figura 6.3. Os custos das arestas são $\ell(0,1) = 2$, $\ell(0,2) = 4$, $\ell(1,2) = 1$, $\ell(1,3) = 3$ e $\ell(2,3) = 5$. A transformação introduz os vértices s_{ij} e s_{ji} para cada uma das arestas $(i,j) \in R$, como mostrado na figura 6.4. A instância do *CVRP*, o resultado final da transformação, é definida sobre o grafo completo com 11 vértices cuja função de custo c é mostrada na tabela 6.1. A figura 6.5 mostra o grafo resultante associado à nova instância do *CVRP*, com diversas das arestas, do total de 55, omitidas para uma melhor visualização.

PUC-Rio - Certificação Digital Nº 0024952/CA

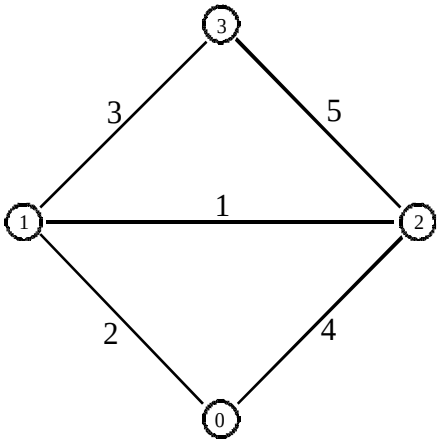

Figura 6.3: *CARP* – Grafo original.

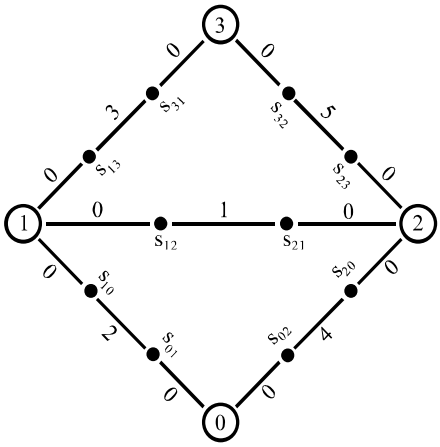

Figura 6.4: *CARP* – Grafo com os vértices s_{ij}.

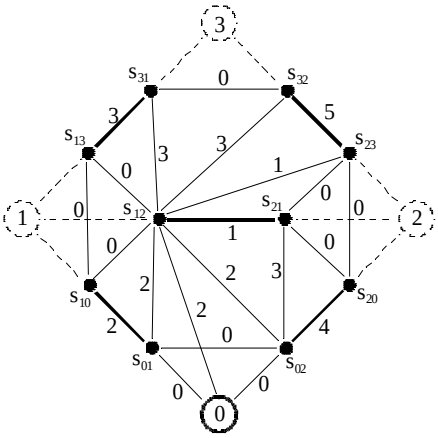

Figura 6.5: *CARP* – Grafo completo associado a instância do *CVRP* (função c mostrada na tabela 6.1). Vértices 1, 2 e 3 não fazem parte do grafo do *CVRP*. Diversas arestas foram omitidas: $(0, s_{31})$, $(0, s_{32})$, $(0, s_{13})$, (s_{02}, s_{32}), (s_{01}, s_{31}), etc.

	0	s_{01}	s_{10}	s_{02}	s_{20}	s_{12}	s_{21}	s_{13}	s_{31}	s_{23}	s_{32}
0	-	0	2	0	3	2	3	2	5	3	5
s_{01}		-	2	0	3	2	3	2	5	3	5
s_{10}			-	2	1	0	1	0	3	1	3
s_{02}				-	4	2	3	2	5	3	5
s_{20}					-	1	0	1	4	0	4
s_{12}						-	1	0	3	1	3
s_{21}							-	1	4	0	4
s_{13}								-	3	1	3
s_{31}									-	4	0
s_{23}										-	5
s_{32}											-

Tabela 6.1: *CARP* – Custos das arestas para o exemplo da figura 6.5.

PUC-Rio - Certificação Digital Nº 0024952/CA

6.4 Formulação do *CARP*

A formulação mestre explícita (*ME-CVRP*) para o *CVRP*, apresentada na seção 5.1, pode ser modificada para considerar as particularidades da transformação do *CARP* ao *CVRP*. O objetivo dessa modificação é a especialização do algoritmo proposto por Fukasawa et al. [50], que será descrita na seção 6.6, para tirar vantagem do fato de existirem r arestas que obrigatoriamente devem pertencer a todas soluções.

A formulação mestre explícita do *CARP* pode ser uma pouco mais simplificada, em relação à formulação *ME-CVRP*, devido ao fato de que $G = (V, E)$, c, d e D resultam de uma instância do *CARP*. Um segundo fator é que todas as arestas $\{(s_{ij}, s_{ji}) \mid (v_i, v_j) \in R\}$ são obrigadas a fazer parte da solução. As simplificações são detalhadas a seguir:

- Como cada veículo que aparece em uma solução do *CARP* deve atender pelo menos uma aresta de R e como toda aresta é representada como dois vértices de G, então todas as variáveis x podem ser restritas a serem menores ou iguais a 1.
- As q-rotas podem ser restringidas àquelas que visitem os vértices s_{ij} e s_{ji} em seqüência, $s_{ij} \rightarrow s_{ji}$ ou $s_{ji} \rightarrow s_{ij}$. Será denotado por p' o número total de q-rotas que satisfazem tal restrição. Somente essa restrição já elimina um grande número das variáveis λ. Por outro lado, esta restrição não altera a complexidade $O(n^2.D)$ de geração das q-rotas.
- A metade das restrições (1) na formulação *ME-CVRP* pode ser eliminada. A redefinição anterior das q-rotas e as restrições (5) garantem que se as restrições de grau associadas a s_{ij} são satisfeitas, então as restrições de grau associadas a s_{ji} também serão satisfeitas. Esta observação tem um grande impacto no desempenho do algoritmo, porque o tamanho dos *LP*s e a convergência da rotina de geração de colunas depende, em grande parte, do número de tais restrições. Em outras palavras, isto sugere que algoritmo comporta-se como se estivesse resolvendo uma instância com r e não $2.r$ clientes.
- Um fato bem conhecido na separação de cortes de capacidade (veja por exemplo [80, 84]) é que arestas (u, v) com valor 1 na solução fracionária devem ser contraídas. Isto acontece porque se um corte de capacidade sobre um conjunto S, com $(u, v) \in \delta(S)$, é violado, então o corte de capacidade sobre o conjunto $S \cup \{u, v\}$ também é violado. O resultado é que a rotina de separação sempre trabalha sobre um grafo com no máximo $r + 1$ vértices.

PUC-Rio - Certificação Digital Nº 0024952/CA

A formulação mestre explícita para o *CVRP*, considerando-se as particularidades da redução do *CARP* ao *CVRP*, chamada de *ME-CARP*, é apresentada a seguir:

$$
\text{ME-CARP}\left\{
\begin{array}{llllll}
\min \sum\limits_{e\in E} c_e.x_e & & & & & (0) \\
\text{sujeito a} & & & & & \\
 & \sum\limits_{e\in\delta(\{s_{ij}\})} x_e & = & 2 & \forall\ (v_i,v_j)\in R & (1') \\
 & \sum\limits_{e\in\delta(\{v_0\})} x_e & \geq & 2.K^* & & (2) \\
 & \sum\limits_{e\in\delta(S)} x_e & \geq & 2.k(S) & \forall\ S\subseteq V\setminus\{v_0\} & (3) \\
 & x_e & \leq & 1 & \forall\ e\in E & (4') \\
\sum\limits_{j=1}^{p'} q_j^e.\lambda_j \quad - & x_e & = & 0 & \forall\ e\in E & (5) \\
 & x_e & \in & \{0,1\} & \forall\ e\in E & \\
\lambda_j & & \geq & 0 & \forall\ j\in\{1,\ldots,p'\}\ . &
\end{array}
\right.
$$

As restrições (3) podem usar $k(S) = \lceil \sum_{v_i\in S} d_i/D \rceil$ como um limite inferior no número mínimo de veículos necessário para atender o subconjunto de clientes $S \subset V_+$. As restrições (5) obrigam as variáveis x a serem combinações lineares das q-rotas (estas restrições não são estritamente necessárias para que *ME-CARP* seja uma formulação do *CARP*). As restrições de integralidade completam a formulação.

Uma formulação mais compacta, a efetivamente usada no algoritmo *branch-and-cut-and-price*, é obtida se toda ocorrência de x_e nas restrições (0), (1'), (2) e (3) for substituida pelo seu equivalente dado por (5). A relaxação das restrições de integralidade resulta em um problema de programação linear, que será chamado de *DWM-CARP*, cuja solução é uma

PUC-Rio - Certificação Digital Nº 0024952/CA

estimativa inferior válida para o valor da solução ótima de *ME-CARP*:

$$
\text{DWM-CARP}\left\{
\begin{array}{lcll}
\min \sum_{l=1}^{p'} \sum_{e \in E} c_e . q_l^e . \lambda_l & & & (7) \\
\text{sujeito a} & & & \\
\sum_{l=1}^{p'} \sum_{e \in \delta(\{s_{ij}\})} q_l^e . \lambda_l & = & 2 \quad \forall\ (v_i, v_j) \in R & (8) \\
\sum_{l=1}^{p'} \sum_{e \in \delta(\{v_0\})} q_l^e . \lambda_l & \geq & 2.K^* & (9) \\
\sum_{l=1}^{p'} \sum_{e \in \delta(S)} q_l^e . \lambda_l & \geq & 2.k(S) \quad \forall\ S \subseteq V \setminus \{v_0\} & (10) \\
\sum_{l=1}^{p'} q_l^e . \lambda_l & \leq & 1 \quad \forall\ e \in E & (11) \\
\lambda_l & \geq & 0 \quad \forall\ l \in \{1, \ldots, p'\}\ . &
\end{array}
\right.
$$

Não somente os planos de cortes comumente denominados *Rounded Capacity Inequalities* podem aparecer no *DWM-CARP*, um corte genérico $\sum_{e \in A} a_e . x_e \geq b$ também pode ser incluído como $\sum_{l=1}^{p} (\sum_{e \in A} a_e . q_l^e) \cdot \lambda_l \geq b$. Assim, outras desigualdades válidas, por exemplo as descritas em [80], podem ser adicionados a esta formulação. Esta mesma abordagem pode ser usada para incorporar à formulação as desigualdades propostas por Belenguer e Benavent [20], o que será discutido na seção 6.5.

6.5
Relaxação de Belenguer e Benavent

O algoritmo *branch-and-cut* apresentado por Belenguer e Benavent [20] é baseado em uma relaxação do *CARP* definida sobre variáveis z, onde $z(e)$ representa o número de vezes que o arco e da malha viária é utilizado sem o atendimento da demanda associada. O programa linear inteiro correspondente minimiza a soma das distâncias percorridas sem atendimento de demanda. As restrições são associadas a subconjuntos $S \subseteq N$ que definem cortes no grafo $H = (N, A)$. Determina-se então um número mínimo de arestas no corte que tem que ser percorridas sem o

PUC-Rio - Certificação Digital Nº 0024952/CA

atendimento da demanda. Esta formulação pode ser expressa como segue:

$$\text{BB-CARP} \begin{cases} \min \sum_{e \in A} \ell_e . z_e \\ \text{sujeito a} \\ \quad \sum_{e \in \delta(\{S\})} z_e \geq \alpha(S) \quad \forall\ S \subseteq N \quad (B.1) \\ \quad z_e \geq 0 \quad \forall\ e \in A \\ \quad z_e \in \mathbb{Z} \quad \forall\ e \in A\,, \end{cases}$$

onde $\alpha(S)$ é o limite inferior no número de arestas a serem percorridas sem atender a demanda para o corte definido pelo conjunto S. Este número é definido para diferentes casos relativamente gerais. Aqui é importante frizar que *BB-CARP* não é uma formulação para o *CARP* tendo por objetivo somente fornecer uma estimativa inferior para o valor da solução ótima.

Este fato torna-se mais evidente quando se observa que é uma questão em aberto determinar, em tempo polinomial, se uma solução (inteira) para *BB-CARP* corresponde a uma solução para o *CARP*. Portanto, *BB-CARP* pode ser usado somente para verificar a qualidade, e eventualmente provar a otimalidade, de soluções obtidas por heurísticas.

Por outro lado, a excelente qualidade das estimativas obtidas por Belenguer e Benavent [20] levam a questionar a possibilidade de utilizá-las como cortes poliédricos em outras formulações. Em particular se isto seria possível na formulação aqui proposta.

A resposta é afirmativa. Para tal, é necessário traduzir as desigualdades (B.1) definidas sobre variáveis z em desigualdades nas variáveis x. Isto corresponde apenas a escrever cada variável z_e em função das variáveis x da formulação original. De fato, isto pode ser facilmente realizado, uma vez que uma aresta (s_{ij}, s_{kl}) contém o caminho mínimo do vértice i ao vértice k no grafo original, o qual é conhecido e permite, portanto, listar todas as arestas que são utilizadas sem o atendimento da demanda associada.

Considere, por exemplo, a aresta (s_{21}, s_{02}) do exemplo mostrado na figura 6.5 e tabela 6.1. Esta aresta é associada ao caminho (o mais curto) $2, 1, 1, 0$ no grafo mostrado na figura 6.3 e corresponde ao caminho de s_{21} a s_{02} detalhado na figura 6.6.

A tradução de uma desigualdade sobre as variáveis z a outra sobre as variáveis x pode, então, ser conseguida pela substituição de cada variável z_e, onde e é uma aresta do grafo original, pela soma de todas as variáveis x_{uv} tais que o caminho mínimo de u a v contenha a aresta e. Seja $Z_e(x)$ tal expressão.

PUC-Rio - Certificação Digital Nº 0024952/CA

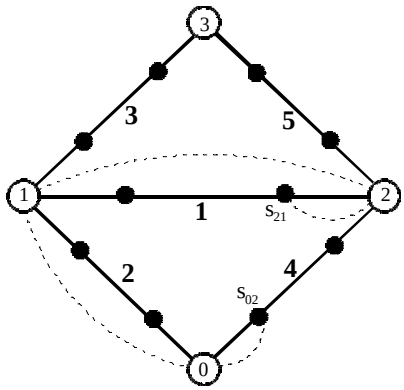

Figura 6.6: *CARP* – Caminho mínimo associado a uma aresta.

Considere, por exemplo, as desigualdades denominadas por Belenguer e Benavent [20] de *Odd Edge Cutset constraints*. Dado o *odd edge cutset* definido pelo vértice 2 na figura 6.3, essa desigualdade é então: $z_{02} + z_{12} + z_{23} \geq 1$. A desigualdade equivalente escrita sobre as variáveis x, $Z_{02}(x) + Z_{12}(x) + Z_{23}(x) \geq 1$, segue:

$$\begin{aligned} & x_{s_{21},s_{32}} + x_{s_{20},s_{32}} + x_{s_{21},s_{31}} + x_{s_{20},s_{31}} + x_{s_{23},s_{31}} + x_{s_{21},s_{10}} + \\ & x_{s_{20},s_{10}} + x_{s_{23},s_{10}} + x_{s_{21},s_{01}} + x_{s_{20},s_{01}} + x_{s_{23},s_{01}} + x_{s_{20},s_{12}} + \\ & x_{s_{23},s_{12}} + x_{s_{21},s_{13}} + x_{s_{20},s_{13}} + x_{s_{23},s_{13}} + x_{s_{21},s_{02}} + x_{s_{23},s_{02}} \quad \geq \quad 1\,, \end{aligned}$$

onde o lado esquerdo equivale à tradução da variável z_{12} em termos das variáveis x, enquanto que as traduções das variáveis z_{02} e z_{23} resultam em expressões nulas.

Uma observação importante é que essa tradução fornece um procedimento eficiente, embora de complexidade não polinomial, para testar se uma solução inteira ótima z^*, dada pela formulação *BB-CARP*, é uma solução factível para o *CARP*. O procedimento consiste em verificar a existência de uma solução factível 0-1 no sistema linear, escrito sobre as variáveis x, dado por $Z_e(x) = z_e^* \;\; \forall\, e \in E$, junto com as restrições (1'), (2), (3) e (4') da formulação *ME-CARP* e $x_{s_{ij},s_{ji}} = 1 \,\forall\, (i,j) \in R$. Caso esse sistema tenha solução factível, então o valor da função objetivo dada por z^* é ótimo. Caso contrário, este valor corresponde apenas a uma estimativa inferior para o valor da solução ótima. No primeiro caso, uma solução ótima para o *CARP*, associada a z^*, pode ser construída diretamente a partir de qualquer solução $\overline{x}$ que satisfaça o sistema descrito.

PUC-Rio - Certificação Digital Nº 0024952/CA

6.6 Algoritmo *branch-and-cut-and-price*

O algoritmo *branch-and-cut-and-price* robusto para o *CVRP* apresentado por Fukasawa et al. [50], combina um algoritmo *branch-and-cut* com um algoritmo de geração de colunas, as quais correspondem às q-rotas definidas no capítulo 5. A formulação mestre explícita (*ME-CVRP*) para o *CVRP*, com número exponencial de restrições e variáveis, é descrita na seção 5.1. A formulação equivalente do *CARP*, *ME-CARP*, é descrita na seção 6.4. A especialização desse algoritmo para o *CARP* trabalha com um problema de programação linear derivado de *ME-CARP*, cuja formulação (*DWM-CARP*) também é descrita na seção 6.4.

A especialização do algoritmo para o *CARP* basicamente consiste na inclusão da transformação do *CARP* ao *CVRP*, descrita na seção 6.3, e na modificação do esquema de geração de colunas para forçar as q-rotas a visitar os vértices s_{ij} e s_{ji} em seqüência. A única outra modificação é a alteração do modelo de programação linear para aquele descrito na seção 6.4 (*DWM-CARP*).

Nas seções 6.6.1 a 6.6.3 são descritas os principais componentes desse algoritmo: a geração de cortes, a geração de colunas e o algoritmo *branch-and-bound*.

6.6.1 Geração de Cortes

Tanto o algoritmo original quanto essa especialização consideram inicialmente na formulação utilizada, *DWM-CVRP* ou *DWM-CARP*, somente restrições relativas aos graus dos vértices (restrições (8) e (9) de *DWM-CARP*) e cortes violados são adicionados à formulação quando necessário.

Os cortes de capacidade não são os únicos que podem aparecer nessas formulações. Um corte genérico $\sum_{e \in E} a_e.x_e \geq b$ pode ser incluído como $\sum_{j=1}^{p}(\sum_{e \in E} a_e.q_j^e).\lambda_j \geq b$. Assim, o algoritmo pode adicionar todas as classes de cortes descritos em [80] (*framed capacity*, *strengthened comb*, *multistar*, *partial multistar*, *generalized multistar* e *hypotour cuts*), utilizando o pacote de rotinas *CVRPSEP* [79] para separar tais cortes. Contudo, os testes computacionais mostraram que os cortes realmente efetivos, além das restrições (10) e (11), foram os chamados *framed capacity* e *strengthened comb cuts*. Assim, somente estes foram utilizados na obtenção dos resultados computacionais descritos na seção 6.7.

PUC-Rio - Certificação Digital Nº 0024952/CA

Os cortes separados com o pacote de rotinas *CVRPSEP* são definidos em termos de variáveis x_e associadas às arestas do grafo que modela a instância do *CVRP*. Como o algoritmo utiliza a formulação *DWM-CARP*, essas variáveis não estão realmente disponíveis. A solução ótima desta formulação, dada em termos de variáveis λ, é representada por λ_j^* para todas as q-rotas j. Desta forma, as variáveis λ_j^* são convertidas em variáveis x_e^* correspondentes para a utilização do pacote de rotinas *CVRPSEP*.

6.6.2
Geração de Colunas

Sejam μ, ν, π, e ω os multiplicadores duais associados às restrições (8), (9), (10), e (11), respectivamente, de *DWM-CARP* (veja definição na seção 6.4). O subproblema de geração de colunas consiste em encontrar q-rotas (variável λ), cuja coluna resultante tenha custo reduzido mínimo, o qual é dado pela soma dos fatores $\bar{c}_e$ relativos às arestas da q-rota:

$$\bar{c}_e = \begin{cases} c_e - \mu_i - \mu_j - \sum\limits_{S \,|\, \delta(S) \ni e} \pi_S - \omega_e & \text{se } e = \{v_i, v_j\} \in E \setminus \delta(\{v_0\}) \\ c_e - \nu - \mu_j - \sum\limits_{S \,|\, \delta(S) \ni e} \pi_S & \text{se } e = \{v_0, v_j\} \in \delta(\{v_0\}) \,. \end{cases} \tag{6-9}$$

Um corte genérico adicional $\sum_{j=1}^{p}(\sum_{e \in E} a_e.q_j^e).\lambda_j \geq b$, com multiplicador dual α, contribui com o valor $-a_e.\alpha$ no cálculo de $\bar{c}_e$.

Como as restrições (8) são omitidas para os vértices s_{ji}, ocorrendo apenas para os correspondentes s_{ij}, a busca por uma q-rota deve ser restringida àquelas que visitam esses vértices em seqüência ($s_{ij} \rightarrow s_{ji}$, ou $s_{ji} \rightarrow s_{ij}$). O subproblema de geração de colunas reduz-se a encontrar q-rotas de custo reduzido mínimo, ou seja, q-rotas cuja soma dos fatores $\bar{c}_e$ é mínima. Esse problema é $\mathcal{NP}$-difícil. Contudo, devido ao fato de todas as demandas serem inteiras, o mesmo pode ser resolvido em tempo pseudo-polinomial [30].

Observe que uma q-rota admite que um mesmo vértice seja visitado mais de uma vez e, conseqüentemente, uma aresta com demanda associada seja percorrida mais de uma vez (considerando essa demanda a cada vez que a atravessa). Assim, o uso de q-rotas no lugar de rotas válidas piora a qualidade da solução da relaxação linear do *DWM-CARP*, ou seja, piora a estimativa inferior da solução ótima. Esta piora pode ser atenuada se são utilizadas apenas q-rotas que não contenham ciclos de pequeno tamanho. Assim, o algoritmo aqui descrito, conforme detalhado em [50], utiliza programação dinâmica para encontrar q-rotas sem s-ciclos, para valores

PUC-Rio - Certificação Digital Nº 0024952/CA

pequenos de s, $s \leq 4$.

Um ciclo de tamanho dois, veja exemplo na figura 6.7, corresponde a visitar um vértice s_{ij}, seguir até um vértice s_{kl} (passando antes por s_{lk}) e então retornar ao s_{ij} (passando antes por s_{ji}). A eliminação desses ciclos na realidade impede que uma aresta com demanda associada seja atravessada na mesma direção em que já havia sido atravessada ao começar um tal ciclo. Por conseqüência, a eliminação de ciclos de tamanho três resulta que uma aresta e, com demanda associada, somente pode ser novamente visitada no mesmo sentido depois que a q-rota passe por três outras arestas com demandas (o que pode até incluir essa aresta e em sentido contrário).

Os resultados computacionais descritos na seção 6.7 foram obtidos com a eliminação de ciclos de tamanho três, que foi foi implementada segundo a técnica descrita por Irnich e Villeneuve [66].

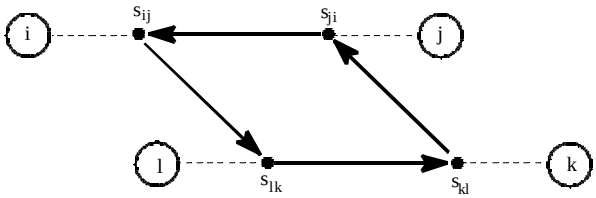

Figura 6.7: *CARP* – Ciclo de tamanho dois.

PUC-Rio - Certificação Digital Nº 0024952/CA

6.6.3
Branch-and-Bound

O algoritmo, que utiliza a regra de busca em profundidade na árvore de enumeração, começa com a adição de colunas à formulação *DWM-CARP* até que não existam mais colunas de custo reduzido negativo, quando então tenta-se adicionar cortes violados. Este esquema é repetido até que os subproblemas de geração de colunas e de separação não possam mais fornecer novos elementos à formulação.

No caso de particionamento do conjunto de soluções (*branching*) do *LP* corrente, é utilizada uma adaptação do esquema proposto em [80]. O *branching* dá-se pela imposição da disjunção $(x(\delta(S)) = 2) \vee (x(\delta(S)) \geq 4)$, onde S é um conjunto de vértices tal que $2 < x^*(\delta(S)) < 4$ e x^* é a solução ótima do LP em termos das variáveis x. Essa disjunção implica que ou os vértices no conjunto S são atendidos por um único veículo ou são atendidos por dois ou mais veículos.

O conjunto S é escolhido a partir de estimativas da solução ótima para cada subproblema resultante, calculada com um pequeno número de iterações da geração de colunas. Para limitar o tempo gasto nessas estimativas,

especialmente quando a profundidade h da árvore de enumeração é alta, somente p conjuntos ($5 \leq p \leq \max\{10 - h, 5\}$) são considerados. Além disso, é dada prioridade aos conjuntos com menores valores $|x^*(\delta(S)) - 2.7|/d(S)$. O valor 2.7 é utilizado porque a restrição $x(\delta(S)) = 2$ tende a causar maior impacto no LP do que $x(\delta(S)) \geq 4$, causando o desbalanceamento da árvore de enumeração. Por outro lado, valores mais próximos de 2 do que de 4 aumentam o impacto da restrição $x(\delta(S)) \geq 4$.

6.7 Resultados Computacionais

Os testes foram realizados em um computador com processador *Pentium* IV, 2.4 GHz, com 1GB de RAM, utilizando os conjuntos de instâncias *kshs*, *gdb*, *bccm* e *egl*, (disponíveis no endereço `http://www.uv.es/~belengue/carp.html`). Essas instâncias foram originalmente utilizadas por Kiuchi et al. [68] (*kshs*), DeArmon [38] e Golden et al. [56] (*gdb*), Benavent et al. [22] (*bccm*), e Li [77] e Li e Eglese [78] (*egl*), respectivamente. Exceto para as instâncias do conjunto *bccm*, cujos nomes são $1.A$, $1.B$, ..., $10D$, as demais têm seus nomes começando pelo nome do conjunto ao qual pertence.

Os conjuntos *kshs*, *gdb* e *bccm* foram gerados aleatoriamente segundo diferentes padrões de construção, com variação do grafo associado, da capacidade dos veículos e das demandas. Nas instâncias destes conjuntos todas as arestas do grafo contém demandas, ou seja, os conjuntos R e A são iguais. O quarto conjunto, *egl*, foi construído usando-se um grafo associado à malha rodoviária do distrito de Lancashire (GBR). Os custos e as demandas (quando existentes) são proporcionais ao comprimento das arestas. Estes conjuntos de instâncias são os mais utilizados atualmente em testes de algoritmos exatos e heurísticos para o *CARP*, assim como na avaliação comparativa de desempenho entre tais algoritmos.

Nas tabelas 6.2 e 6.3, instâncias *egl* e *bccm* respectivamente, são comparados as estimativas inferiores calculadas no nó raiz pelo algoritmo *branch-and-cut-and-price* (solução do *DWM-CARP*) com as melhores estimativas inferiores conhecidas anteriormente. As quatro primeiras colunas de ambas as tabelas especificam os nomes das instâncias e suas características: o número de vértices V, o número de arestas com demandas r e o número mínimo de veículos K^* necessário para atender todas as demandas. As demais colunas mostram a melhor estimativa superior conhecida LS, a melhor estimativa inferior anteriormente conhecida LI (todas obtidos em [20]), as

PUC-Rio - Certificação Digital Nº 0024952/CA

novas estimativas inferiores LN calculadas no nó raiz pelo algoritmo *branch-and-cut-and-price* e os respectivos tempos de CPU (em segundos) gastos na obtenção destes últimos T_{raiz}. Os valores em negrito indicam a igualdade com a melhor estimativa superior anteriormente conhecido.

Os resultados mostram que as estimativas inferiores calculadas no nó raiz pelo algoritmo *branch-and-cut-and-price* são, para a maioria das instâncias, iguais ou melhores do que as anteriormente conhecidas. Em somente duas instâncias, $2.B$ e $5.B$, a estimativa inferior obtida foi uma unidade menor. Note-se que as novas estimativas são estritamente melhores em todas as instâncias onde $K^* > 5$, o que indica que a sua qualidade é menos sensível ao uso de muitos veículos. As novas estimativas também são estritamente melhores em todas as instâncias do conjunto *egl*.

A tabela 6.4 apresenta os resultados para as instâncias em que o algoritmo *branch-and-cut-and-price* foi utilizado em todos os nós da árvore de enumeração. As quatro primeiras colunas são iguais às das tabelas 6.2 e 6.3. As demais colunas mostram o valor da solução ótima Opt, o tempo de CPU (em segundos) gasto no nó raiz T_{raiz}, o número de nós da árvore de enumeração $N_{nós}$, os tempos de CPU (em segundos) gastos na rotina de geração de colunas T_{cols} e de geração de cortes T_{cortes} e o total geral T_{total}. Os valores em negrito, em dez das instâncias, indicam que a otimalidade foi provada pela primeira vez: a última instância em aberto do conjunto *kshs*, as últimas três do conjunto *gdb*, quatro do conjunto *bccm* e duas do conjunto *egl*. Em uma instância (*gdb13*) o valor da solução ótima encontrada é inferior ao da melhor estimativa superior anteriormente conhecida.

Ressalte-se que em 20 instâncias da classe *bccm* ($1.A$, $1.B$, $2.A$, $3.A$, $3.B$, $4.A$, $4.B$, $4.C$, $5.A$, $6.A$, $7.A$, $7.B$, $8.A$, $8.B$, $9.A$, $9.B$, $9.C$, $10.A$, $10.B$ e $10.C$) a nova estimativa inferior calculada coincide com a melhor estimativa superior conhecida. Contudo, o algoritmo não foi capaz de encontrar as soluções ótimas para várias destas instâncias em um tempo computacional razoável, pois houve uma dispêndio excessivo de tempo na exploração de nós além da profundidade 20 nas respectivas árvores de enumeração.

Os tempos de CPU, satisfatoriamente baixos, e a qualidade das estimativas obtidas na raiz da árvore de enumeração levam a crer que é possível encontrar a solução ótima para mais instâncias dos conjuntos *bccm* e *egl* em um futuro próximo. Além disso, as estimativas das soluções ótimas das instâncias do conjunto *egl* ainda podem ser muito melhoradas com a adição de cortes específicos para o *CARP*, como aqueles propostos por Belenguer e Benavent [19, 20].

PUC-Rio - Certificação Digital Nº 0024952/CA

Instância	|V|	|E|	r	$K*$	LS	LI	LN	T_{raiz}
egl-e1-a	77	98	51	5	3548	3515	**3548**	144,203
egl-e1-b	77	98	51	7	4498	4436	4468	52,594
egl-e1-c	77	98	51	10	5595	5453	5542	46,063
egl-e2-a	77	98	72	7	5018	4994	5011	521,219
egl-e2-b	77	98	72	10	6340	6249	6280	198,531
egl-e2-c	77	98	72	14	8415	8114	8234	66,875
egl-e3-a	77	98	87	8	5898	5869	**5898**	924,859
egl-e3-b	77	98	87	12	7822	7646	7697	375,609
egl-e3-c	77	98	87	17	10433	10019	10163	142,125
egl-e4-a	77	98	98	9	6461	6372	6395	1171,580
egl-e4-b	77	98	98	14	9021	8809	8884	418,984
egl-e4-c	77	98	98	19	11779	11276	11427	202,688
egl-s1-a	140	190	75	7	5018	4992	5014	750,391
egl-s1-b	140	190	75	10	6435	6201	6379	204,500
egl-s1-c	140	190	75	14	8518	8310	8480	66,984
egl-s2-a	140	190	147	14	9995	9780	9824	3260,250
egl-s2-b	140	190	147	20	13174	12886	12968	896,706
egl-s2-c	140	190	147	27	16795	16221	16353	408,878
egl-s3-a	140	190	159	15	10296	10025	10143	1680,371
egl-s3-b	140	190	159	22	14053	13554	13616	1639,530
egl-s3-c	140	190	159	29	17297	16969	17100	635,292
egl-s4-a	140	190	190	19	12442	12027	12143	14318,100
egl-s4-b	140	190	190	27	16531	15933	16093	2761,130
egl-s4-c	140	190	190	35	20832	20179	20375	1119,980
Gap médio %					2,40	1,45		
Gap máximo %					4,27	3,11		

Tabela 6.2: *CARP* – Comparação dos limites inferiores para as instâncias *egl*.

PUC-Rio - Certificação Digital Nº 0024952/CA

Instância	|V|	r	$K*$	LS	LI	LN	T_{raiz}
1.A	24	39	2	247	**247**	**247**	98,278
1.B	24	39	3	247	**247**	**247**	54,561
1.C	24	39	8	319	309	312	771,882
2.A	24	34	2	298	**298**	**298**	79,404
2.B	24	34	3	330	**330**	329	169,010
2.C	24	34	8	528	526	**528**	0,969
3.A	24	35	2	105	**105**	**105**	127,590
3.B	24	35	3	111	**111**	**111**	134,339
3.C	24	35	7	162	161	161	3,156
4.A	41	69	3	522	**522**	**522**	2475,280
4.B	41	69	4	534	**534**	**534**	1178,370
4,C	41	69	5	550	**550**	**550**	824,599
4.D	41	69	9	652	644	648	76,576
5.A	34	65	3	566	**566**	**566**	629,417
5.B	34	65	4	589	**589**	588	388,144
5.C	34	65	5	617	612	613	274,772
5.D	34	65	9	724	714	716	62,779
6.A	31	50	3	330	**330**	**330**	158,667
6.B	31	50	4	340	338	337	169,254
6.C	31	50	10	424	418	420	119,371
7.A	40	66	3	382	**382**	**382**	319,349
7.B	40	66	4	386	**386**	**386**	163,754
7.C	40	66	9	437	436	436	607,637
8.A	30	63	3	522	**522**	**522**	359,426
8.B	30	63	4	531	**531**	**531**	168,885
8.C	30	63	9	663	653	654	376,863
9.A	50	92	3	450	**450**	**450**	17722,600
9.B	50	92	4	453	**453**	**453**	4520,030
9.C	50	92	5	459	**459**	**459**	1460,890
9.D	50	92	10	518	509	512	305,158
10.A	50	97	3	637	**637**	**637**	13336,600
10.B	50	97	4	645	**645**	**645**	13719,400
10.C	50	97	5	655	**655**	**655**	5078,570
10.D	50	97	10	739	732	734	473,632
Gap médio %					0,41	0,32	
Gap máximo %					3,13	2,19	

Tabela 6.3: *CARP* – Comparação dos limites inferiores para as instâncias *bccm*.

PUC-Rio - Certificação Digital Nº 0024952/CA

Instância	|V|	r	$K*$	Opt	T_{raiz}	$N_{nós}$	T_{cols}	T_{cortes}	T_{total}
kshs1	8	15	4	14661	0,719	2	0,656	0,172	0,828
kshs2	10	15	4	9863	0,375	2	0,391	0,047	0,453
kshs3	6	15	4	9320	0,688	4	0,766	0,219	1,016
kshs4	8	15	4	**11498**	0,828	3	0,531	0,391	0,953
kshs5	8	15	3	10957	0,703	1	0,656	0,047	0,703
kshs6	9	15	3	10197	0,734	3	1,391	0,078	1,484
gdb1	12	22	5	316	1,859	10	1,359	1,672	3,125
gdb2	12	26	6	339	0,531	5	1,125	0,141	1,359
gdb3	12	22	5	275	0,406	5	0,531	0,234	0,844
gdb4	11	19	4	287	0,281	8	0,625	0,125	0,813
gdb5	13	26	6	377	0,891	2	0,422	0,578	1,016
gdb6	12	22	5	298	0,250	1	0,203	0,047	0,250
gdb7	12	22	5	325	0,625	10	1,203	0,422	1,750
gdb8	27	46	10	**348**	2,750	28	30,922	0,859	32,469
gdb9	27	51	10	303	12,016	7	22,141	8,313	30,969
gdb10	12	25	4	275	1,063	10	5,500	0,656	6,344
gdb11	22	45	5	395	19,438	29	1351,130	9,688	1364,630
gdb12	13	23	7	**458**	0,703	27	4,422	0,406	5,219
gdb13	10	28	6	**536**	1,125	145	88,063	1,484	93,031
gdb14	7	21	5	100	0,156	1	0,156	0	0,156
gdb15	7	21	4	58	0,781	9	3,094	0,531	3,750
gdb16	8	28	5	127	2,313	19	24,406	1,563	26,422
gdb17	8	28	5	91	0,953	23	33,328	0,703	34,828
gdb18	9	36	5	164	5,578	16	53,641	2,172	56,391
gdb19	8	11	3	55	0,063	1	0,047	0	0,063
gdb20	11	22	4	121	1,375	7	3,719	0,656	4,500
gdb21	11	33	6	156	3,969	13	17,781	2,734	20,859
gdb22	11	44	8	200	6,406	29	105,109	5,156	112,078
gdb23	11	55	10	233	9,547	30	136,438	6,625	145,453
1C	24	39	8	**319**	774,906	6119	7802,220	830,156	8916,78
2B	24	34	3	330	169,484	14	581,656	86,938	671,250
2C	24	34	8	**528**	0,938	1	0,891	0,047	0,953
3C	24	35	7	**162**	3,141	363	314,719	4,438	328,891
7C	31	66	9	**437**	39,781	3	54,609	9,500	131,969
egl-e1-a	77	51	5	**3548**	143,844	1	128,391	15,1719	143,875
egl-e3-a	77	87	8	**5898**	923,656	1	831,188	91,7969	923,766

Tabela 6.4: *CARP* – Resultados do algoritmo *BCP* para as instâncias *kshs*, *gdb*, *bccm* e *egl*.

PUC-Rio - Certificação Digital Nº 0024952/CA