

HEURÍSTICA PARA A SOLUÇÃO DE PROBLEMAS DE TRANSPORTE COLABORATIVO DE GRANDE PORTE

Enrico Barnaba Ferri (POLI/USP)

enrico.ferri@gmail.com

Hugo Tsugunobu Yoshida Yoshizaki (POLI/USP)

hugo@usp.br

Rafael Alzuguir Rosin (POLI/USP)

rafael.rosin@poli.usp.br



A necessidade de redução de custos logísticos tem obrigado as empresas a colaborar entre si. O problema de logística colaborativa aqui enfocado é assim definido: identificar ciclos (ou seja, um percurso fechado) em um conjunto de rotas de carga de lotação (onde o caminhão coleta carga em um ponto e vai diretamente ao local de descarga, pois é completamente preenchido) de vários embarcadores de forma a minimizar o reposicionamento (isto é, viagens sem carga útil) de caminhões, dado que o subconjunto de rotas de um determinado embarcador pode conter rotas que complementam aquelas de outro. Desta maneira, vários embarcadores combinados podem oferecer aos transportadores um conjunto de ciclos com movimentação regular de veículos com carga completa e com mínimo reposicionamento. Esse problema pode ser modelado como um problema particular de cobertura de conjuntos com restrição de ciclos, o problema de cobertura de rotas com cardinalidade restrita (PCRCR), que é NP-Hard. Este estudo apresenta uma heurística alternativa que obtém resultados, em média melhores que a literatura existente, além de solucionar instâncias maiores. Ademais, o tempo de execução da heurística cresce de forma polinomial em função do tamanho do problema, ao contrário dos demais métodos aqui avaliados, que possuem comportamento exponencial.

Palavras-chaves: Transporte colaborativo, logística, pesquisa operacional, heurística

1. Introdução

A preocupação com operações mais eficientes, sustentáveis e competitivas levou as empresas a colaborem de muitas maneiras diferentes onde quer que estejam, na mesma ou diferentes cadeias de fornecimento. São exemplos dessas práticas: distribuição conjunta, consolidação de cargas, compra conjunta, bem como compartilhamento de capacidades e serviços. Estas atividades de cooperação são fomentadas pelo atual desenvolvimento dos setores de telecomunicações e tecnologias de informação on-line.

No mercado de FTL (full truckload) os embarcadores geralmente compram os serviços de transporte rota a rota, selecionando os transportadores pelo menor custo por rota. Uma operação de FTL (full truckload) é o movimento de um caminhão diretamente a partir da origem para o destino, com nenhuma parada entre eles para carregar ou entregar cargas parciais. Parece ser uma operação simples, mas existem custos ocultos associados ao reposicionamento do caminhão (deadhead). O reposicionamento ocorre quando o transportador tem que mover um caminhão vazio após descarregar sua carga até o próximo local de coleta. Em geral, transportadores fazem reposicionamento de veículos de forma a satisfazer diferentes necessidades de embarcadores. Assim, o custo das rotas realizadas por um único transportador envolve também o custo do reposicionamento de seus veículos, o que é uma função de como origens e destinos de seus clientes estão distribuídos sobre o território.

Uma estratégia para os embarcadores é oferecer aos transportadores um conjunto de rotas com uma configuração onde a distância percorrida para reposicionamentos seja mínima, e desta forma o transportador pode oferecer melhores preços. Um pré-requisito para este agrupamento de rotas é que estas devem ocorrer de uma maneira regular, tal como a reposição de um centro de distribuição por uma fábrica. Trabalhos anteriores trataram este problema, tais como Savelsbergh et al (2004), Ergun et al (2004; 2007a; 2007b), Özener and Ergun (2008), variando de questões estratégicas a operacionais, incluindo como compartilhar ganhos entre embarcadores, inspirados pelos serviços colaborativos baseados na Internet, como Nistevo e Transplace. Este tipo de análise pode ser empregada como uma preliminar para um leilão de transporte (Chen et al, 2009), e tem semelhanças com problemas de colaboração de transportadores (Liu et al, 2010).

Uma questão fundamental é como identificar uma configuração ótima formada por diferentes rotas de embarcadores. Ergun et al (2007a) definiu o problema de transporte colaborativo como: dado um conjunto de rotas, encontrar um conjunto ciclos que abranjam todas as rotas minimizando o reposicionamento de veículos, obedecendo os limites de cardinalidade ou tempo de viagem. Este problema é explosivamente combinatório e, portanto, não podem ser obtidos resultados ótimos mesmo para instâncias relativamente pequenas. Estes autores propuseram uma Heurística (Ergun et al, 2007a) que apresentou um excelente desempenho, mas que ainda depende da geração de todos os conjuntos possíveis de ciclos. A contribuição do nosso trabalho é propor uma Heurística que não dependa da geração de todos os ciclos viáveis e, portanto, pode resolver problemas maiores do que a literatura anterior dentro de tempos de processamento razoáveis.

2. Definição do Problema e Revisão da Literatura

Ergun et al (2007b) definiram o problema de cobertura de rotas com cardinalidade restrita (PCRCR) como uma variação do Problema de Cobertura de Rotas (PCR). O PCR pode ser definido da seguinte forma: dado um dígrafo direcionado $D = (V, A)$, onde V é o conjunto dos

vértices e A , o conjunto dos arcos, onde para cada arco há um custo associado $c(a)$ para $a \in A$ e um conjunto L de rotas $\subseteq A$, o objetivo é minimizar o custo de cobertura das rotas L utilizando-se ciclos. O PCRCR difere do PCR por possuir a restrição de que o número de arcos que compõem o ciclo deve ser menor que um dado inteiro k . Do ponto de vista prático, o PCRCR fornece soluções mais factíveis de serem postas em prática pela dificuldade em se formar ciclos com números muito altos de rotas, o que envolve a sinergia entre diferentes embarcadores e localidades. Savelsbergh et al (2004) provou que o PCRCR é NP-hard, e Immorlica et al (2005) mostrou que algumas variações do PCR são APX-hard. Além de Savelsbergh et al (2004) e Ergun et al (2004; 2007a), nenhuma literatura sobre o PCRCR foi encontrada pelos autores.

Seguindo os autores anteriores, é possível formular o PCRCR como um problema de cobertura de conjuntos, e assim podemos utilizar uma formulação de PI para avaliar o desempenho da heurística para pequenas instâncias.

A formulação do problema de cobertura de conjuntos (representado no Box 1) consiste em: dado o conjunto C_k previamente gerado como a combinação de todos os ciclos (cobrindo pelo menos uma rota com a restrição de cardinalidade e número de reposicionamentos), um subconjunto de ciclos $c \in C_k$ deve ser escolhido de forma a minimizar o custo total dos ciclos cobrindo todas as rotas de modo que $c \cap L \neq \emptyset$.

Ergun et al (2007a) propuseram um algoritmo guloso (GH) baseado na formulação de cobertura de conjuntos e obtiveram excelentes resultados. Uma desvantagem do GH é a necessidade de se gerar todos os ciclos viáveis. Desta forma, o tempo de processamento para a geração dos ciclos aumenta exponencialmente em relação ao número de rotas, mesmo quando é aplicado um procedimento para inibir a geração de ciclos de baixa qualidade (Savelsbergh et al, 2004). Para algumas instâncias, o tempo de processamento para a geração dos ciclos viáveis chega a ser duas vezes maior que o tempo de processamento da heurística GH.

PARÂMETROS:

C_c : custo do ciclo $c \in C_k$.

l_c : rotas que pertencem ao ciclo $c \in C$. Se $l_c = 1$, então a rota $l \in L$ pertence a um ciclo.

Se $l_c = 0$, então a rota $l \in L$ não pertence a um ciclo $c \in C$.

VARIÁVEIS:

x_c : indica se ciclo c foi selecionado ou não. Se $x_c = 1$, o ciclo faz parte da solução. Caso contrário, o ciclo não foi selecionado.

FUNÇÃO OBJETIVO:

Minimizar o custo total dos ciclos selecionados:

$$\min \sum_{c \in C_k} C_c x_c$$

RESTRICÇÕES:

Restrição que faz com que todas as rotas pertençam a, pelo menos, um ciclo:

$$\sum_{c \in C_k} l_c x_c \geq l, \text{ para } \forall l \in L$$

Variável é binária:

$$x_c \in \{0,1\}, \text{ para } \forall c \in C_k$$

Box 1 – Modelo de Cobertura de Conjuntos

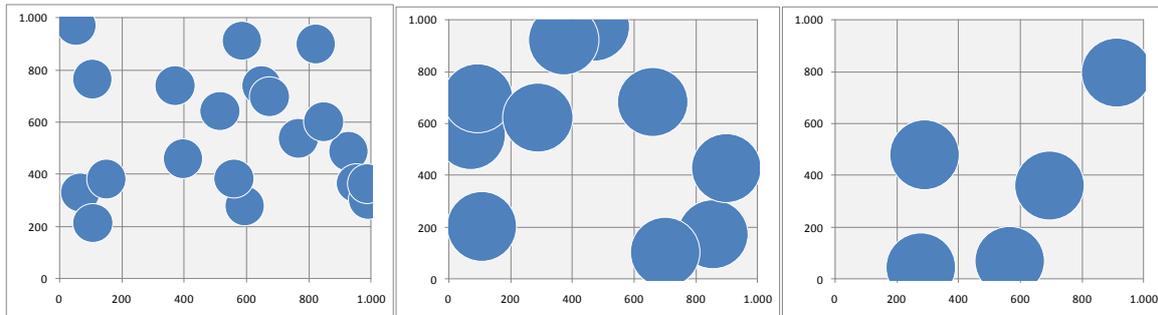


Figura 1 – Exemplo de Dígrafo 1, 2 e 3 respectivamente

As instâncias utilizadas por Ergun et al (2007a) para testar o algoritmo GH seguiram as seguintes características: existem três conjuntos de instâncias por meio de três tipos de dígrafos, cada um com 500 nós. Cada dígrafo difere do outro em relação ao número de agrupamentos e fator de agrupamento (raio do agrupamento como uma fração do tamanho da grade), com o mesmo percentual de curtas (dentro do agrupamento) – 20% – e longas rotas (entre agrupamentos). Dígrafo 1 tem vinte agrupamentos e fator de agrupamento de 0.05; dígrafo 2, dez agrupamentos e fator de agrupamento de 0.1; e dígrafo três, cinco agrupamentos e fator de agrupamento de 0.1. A grade/espço da instância corresponde a um quadrado com tamanhos de lados arbitrários de 1.000 por 1.000 unidades.

A Figura 1 mostra exemplos para cada um dos dígrafos.

A cardinalidade é limitada em cinco. Um ou dois reposicionamentos são permitidos, sujeita à restrição de cardinalidade máxima (cinco arcos).

3. Descrição da Heurística e Resultados Computacionais

A Heurística proposta aqui (CD, de construtiva destrutiva) é uma alternativa ao GH e PI. Ela não requer que sejam geradas todas as combinações viáveis de ciclos, e desta forma ocupa um menor espaço de memória permitindo que sejam resolvidos problemas maiores. Ela também possui um desempenho similar aos outros métodos.

Existem quatro conceitos principais na criação da heurística CD, listados abaixo:

- Durante o início da construção dos ciclos, as rotas mais longas têm prioridade para serem alocadas;
- A construção é dada adicionando-se a rota mais próxima a uma das extremidades do ciclo em construção;
- A fim de reduzir a “miopia” da heurística, devem-se construir ciclos em paralelo e ao final do processo de construção, selecionar o que apresenta melhor “qualidade” (este conceito será explicado mais adiante);
- Após a construção de um ciclo, esse deve ser destruído caso não atenda a um limite de “qualidade” mínima. Esse limite de “qualidade” não deve ser fixo. Deve ser muito rigoroso

no início e ir reduzindo à medida que novos ciclos são construídos.

Os seguintes parâmetros são utilizados pela heurística CD com suas respectivas descrições:

- L é o conjunto das rotas.
- U é dado por $U = L$, é o conjunto das rotas que estão ordenadas em ordem decrescente de comprimento.
- l_{max} é o número de rotas contidas em L ;
- l_1 é o índice atrelado ao conjunto U de modo que $l_1 = [1; l_{max}]$. Dado isso, quando $l_1 = 1$ refere-se à rota de maior comprimento de U e $l_1 = l_{max}$ refere-se à de menor comprimento. Ao final da criação de um ciclo, este índice é sempre incrementado;
- p é o fator de “qualidade” do ciclo dado pela proporção entre o custo do reposicionamento e o custo total de um determinado ciclo;
- p_{min} é a menor proporção p que o ciclo deve ter para não ser destruído. $p_{min} = [0;1]$;
- p_{incr} é o quanto a proporção p_{min} será incrementada após todas as rotas forem inspecionadas. $p_{incr} = [0;1]$;
- \check{C} é o conjunto dos ciclos criados pela heurística (solução final);
- ζ é o conjunto dos ciclos que estão sendo criados em paralelo e ainda não foram selecionados para a solução final;
- C_c é o ciclo candidato a ser considerado na solução final \check{C} ;
- j é o índice utilizado para os ciclos $C_j \in \zeta$;
- j_{max} é o número máximo de ciclos que serão construídos em paralelo;
- l_{min} é a proporção mínima de rotas que devem ser alocadas em ciclo antes de p_{min} ser incrementado. $l_{min} = [0;1]$;
- p_{mult} é o multiplicador que será aplicado a p_{incr} , caso a proporção mínima de rotas l_{min} não seja atingida. Este parâmetro é utilizado para deixar a heurística mais rápida.

Dadas as nomenclaturas e os conceitos explicados anteriormente é possível iniciar a descrição da heurística.

Primeiramente, deve-se criar o conjunto U de modo que $U = L$ e que contém as rotas $l \in U$ em ordem decrescente de comprimento atrelando-se o índice l_1 a este conjunto. Em seguida, escolhe-se $l_1 = 1$ para iniciar a construção dos ciclos $C_j \in \zeta$. A primeira rota do ciclo será chamada de rota base.

Para todos os passos a seguir, sempre que um ciclo atingir as restrições de número máximo de reposicionamento e/ou cardinalidade, sua construção é concluída. Além disso, sempre que uma rota $l \in U$ for selecionada, é removida de U .

Os ciclos $C_j \in \zeta$ serão construídos em paralelo e ao final do processo somente aquele que apresentar a menor proporção p será um candidato ao conjunto \check{C} . A construção desses ciclos se dá adicionando-se ao final do ciclo a j -ésima rota que possui seu vértice inicial mais próximo do vértice final da rota base. Por exemplo, para criar o ciclo C_1 , será inserida a primeira rota mais próxima, para o C_2 , a segunda mais próxima e assim por diante até a j_{max} -ésima rota.

É importante enfatizar que o processo de construção de ciclos em paralelo é feito de modo que a construção de um ciclo não afete a de outro. Por exemplo, se um ciclo utilizar uma determinada rota, essa estará disponível para ser utilizada para os outros ciclos em construção.

A seguir, adiciona-se ao início do ciclo em construção a rota $l \in U$ que possui seu vértice final

mais próximo do vértice inicial da rota base do ciclo C_j .

Para concluir a construção, adiciona-se ao final de C_j a rota $l \in U$ que possui seu vértice inicial mais próximo do vértice da extremidade final de C_j . Repete-se este passo até que a restrição de cardinalidade e/ou reposicionamento sejam atingidas.

Após a construção dos ciclos C_j , verifica-se, para cada um, todas as combinações possíveis e viáveis de fechamento do ciclo de modo que todas elas contenham a rota base e calcula-se o fator p para cada combinação. Escolhe-se, para cada ciclo C_j , a combinação com o menor p eliminando as rotas desnecessárias de cada ciclo. Por exemplo, se um ciclo C_l for composto por apenas duas rotas, existem duas formas de fechamento: a primeira é com as duas rotas e a segunda é com somente a rota base. Se a segunda opção tiver o menor p , o ciclo será fechado com uma rota e a outra é devolvida ao conjunto U .

A seguir, escolhe-se o ciclo $C_c \in C_j$ que é aquele que possui o menor valor p . Esse é o ciclo candidato para fazer parte da solução. Os demais $j-1$ ciclos C_j são descartados.

Se o ciclo candidato tiver p maior que p_{min} , então ele será descartado. Caso contrário é adicionado a \check{C} .

Por último, devem-se atualizar os parâmetros. l_l é incrementado até que a primeira rota $l \in U$ esteja disponível.

Se l_l for maior que l_{max} , então l_l é igual ao menor índice disponível em U e ainda, caso a proporção de rotas alocadas aos ciclos \check{C} em relação ao total de rotas disponíveis (l_{max}) for menor que l_{min} , então p_{min} será multiplicado por $(1 + p_{incr} * p_{mult})$, caso contrário, será multiplicado apenas por $(1 + p_{incr})$.

Os fatores l_{min} e p_{mult} são utilizados quando o número de ciclos gerados é muito baixo devido ao fator p_{min} estar rejeitando a maior parte dos ciclos C_c . Entretanto a utilização desses parâmetros é opcional, porém utilizá-los reduz o tempo de execução da heurística sem afetar o significativamente o resultado.

Inicia-se a construção do próximo ciclo selecionando-se a rota base até que o conjunto U esteja vazio.

A Heurística CD está representada no Box 2.

Os testes foram realizados utilizando-se os mesmos critério utilizados por Ergun et al (2007a), como descrito anteriormente. Foram testadas cinco instâncias para cada combinação de dígrafo, número de rotas e reposicionamentos. Desta forma, os resultados apresentados na Tabela 1 são médias do mesmo conjunto de problemas. Devido restrições de memória, não foi possível obter as solução por PI para os problemas com mais de 3.000 rotas.

Os testes foram realizados em um computador Intel ® Core 2 Quad, 2.33 GHz com 8 GB de memória RAM, sistema operacional Microsoft ® Vista 64bits. GAMS 22.7.2 e ILOG CPLEX 11.0 foram utilizados para os testes em PI; as heurísticas GH e CD foram implementadas em MS Visual C++ 2008.

Início Heurística Construtivo-Destrutiva (CD)

$U := L;$

Ordenar rotas $l \in U$ em ordem decrescente de comprimento utilizando o índice l_l ;

$l_l := 1;$

Enquanto $U \neq \emptyset$ **faça:**

Adicionar a rota base $l \in U$ de índice l_l ao ciclo $C_j \in \zeta$ e removê-la do conjunto U ;

Para cada C_j , adicionar ao final do ciclo a j -ésima rota que possui seu vértice inicial mais próximo do vértice final da rota base e removê-la de U ;

Se atingir restrição de reposicionamento ou de cardinalidade

então vá até *fim_construção*;

Adicionar ao início de C_j a rota $l \in U$ que possui seu vértice final mais próximo do vértice inicial da rota base e removê-la do conjunto U .

Enquanto não atingir restrição de reposicionamento ou de cardinalidade **faça:**

Adicionar ao final de C_j a rota $l \in U$ que possui seu vértice inicial mais próximo do vértice da extremidade final de C_j e removê-la de U .

Fim Enquanto

fim_construção;

Para cada C_j , verificar todas as combinações viáveis de fechamento de ciclo que contenham a rota base;

Para cada combinação de C_j , calcular a proporção p ;

Para cada combinação de C_j , escolher a combinação que possua o menor p e devolver rotas não usadas ao conjunto U ;

Escolhe-se o ciclo $C_c \in C_j$ que possui o menor valor p e descartar os demais C_j de modo que $\zeta = \emptyset$;

Se C_c tiver $p > p_{min}$, **então** C_c é descartado;

caso contrário, $\check{C} := \check{C} \cup C_c$;

l_l é incrementado até que a próxima rota $l \in U$ esteja disponível.

Se $l_l > l_{max}$, **então:**

l_l é igual ao menor índice disponível em U ;

Se (número de rotas alocadas a \check{C}) / $l_{max} < l_{min}$ **então:**

$$p_{min} = p_{min} * (1 + p_{incr} * p_{multi});$$

Caso contrário:

$$p_{min} = p_{min} * (1 + p_{incr});$$

Fim Se

Fim Se

Fim Enquanto

Fim Algoritmo

Box 2 – Heurística CD

DÍGRAFO	ROTAS	Resultados (#)			Diferenças (%)		
		PI	GH	CD	CD X GH	CD x PI	GH X PI
1	100	96,872.14	97,110.76	96,975.58	-0.14%	0.11%	0.25%
1	200	169,442.99	171,764.97	171,135.20	-0.37%	1.00%	1.37%
1	500	338,856.33	358,777.58	357,332.80	-0.40%	5.45%	5.88%
1	1000	621,360.37	686,025.76	683,448.80	-0.38%	9.99%	10.41%
1	1500	828,851.60	918,721.56	910,723.00	-0.87%	9.88%	10.84%
1	3000	1,603,193.08	1,805,524.28	1,761,072.00	-2.46%	9.85%	12.62%
1	5000		3,049,304.03	2,938,458.00	-3.64%		
SUBTOTAL DÍGRAFO 1					-2.37%	8.80%	10.37%
2	100	94,153.69	94,315.11	94,315.00	0.00%	0.17%	0.17%
2	200	157,752.13	159,175.92	158,972.40	-0.13%	0.77%	0.90%
2	500	309,631.20	328,042.02	326,172.20	-0.57%	5.34%	5.95%
2	1000	546,413.84	608,978.00	600,711.00	-1.36%	9.94%	11.45%
2	1500	789,015.56	889,440.60	872,138.00	-1.95%	10.53%	12.73%
2	3000	1,654,982.77	1,858,356.37	1,792,328.00	-3.55%	8.30%	12.29%
2	5000		2,776,382.20	2,659,818.00	-4.20%		
SUBTOTAL DÍGRAFO 2					-3.13%	8.24%	10.88%
3	100	81,018.44	81,024.10	81,019.18	-0.01%	0.00%	0.01%
3	200	150,130.76	151,117.65	150,835.60	-0.19%	0.47%	0.66%
3	500	314,158.93	332,513.16	331,576.20	-0.28%	5.54%	5.84%
3	1000	479,838.33	546,747.99	534,086.40	-2.32%	11.31%	13.94%
3	1500	738,953.22	850,096.26	818,111.60	-3.76%	10.71%	15.04%
3	3000	1,339,474.40	1,514,192.25	1,439,456.00	-4.94%	7.46%	13.04%
3	5000		2,843,777.17	2,672,684.00	-6.02%		
SUBTOTAL DÍGRAFO 3					-4.62%	8.10%	11.99%
TOTAL					-3.33%	8.40%	11.03%

Tabela 1 – Resultados para problemas com um reposicionamento (DH=1)

A Tabela 1 compara os três métodos (PI, GH, e CD) para o caso de um reposicionamento permitido (DH=1). Comparando com o algoritmo GH, a Heurística CD obteve melhores resultados para todos os casos. Além disso, quanto maior a concentração dos dígrafos, melhor foi o desempenho da heurística CD.

Os resultados do Algoritmo CD comparado com a Heurística GH variaram de -0,01% a -6,02%, sendo que a média permaneceu em -3,33%. Comparando-se os resultados da Heurística CD com PI a diferença foi em média 8,40%, enquanto que a diferença entre GH e PI foi de 11,03%.

A Tabela 2 mostra os tempos de execução para problemas com um reposicionamento (DH=1). Para PI e GH os tempos de geração dos ciclos viáveis foram adicionados, pois fazem parte do processo de solução nestes casos.

A Heurística CD obteve tempos de processamento superiores aos tempos da Heurística GH e PI para problemas com menos de 3.000 rotas mas com um crescimento menos pronunciado. Para problemas maiores (3.000 e 5.000 rotas), a Heurística CD obteve tempos melhores. Como exemplo, para os problemas com 5.000 rotas, a Heurística CD foi executada entre 1 e 2 minutos dependendo do Dígrafo, enquanto que GH necessitou de mais de 11 minutos. Este comportamento é evidente na Figura 2, que mostra os tempos de execução versus o número de rotas para o Dígrafo 1.

DÍGRAFO	ROTAS	TEMPO DE PROCESSAMENTO (s)		
		PI	GH	CD
1	100	0.02	0.00	0.03
1	200	0.02	0.02	0.11
1	500	0.05	0.04	2.33
1	1000	0.45	0.33	6.33
1	1500	9.99	1.92	11.07
1	3000	4,507.56	53.34	29.54
1	5000		680.58	58.43
2	100	0.02	0.01	0.16
2	200	0.03	0.01	0.72
2	500	0.06	0.05	2.58
2	1000	0.48	0.36	6.74
2	1500	9.96	1.91	12.15
2	3000	4,823.14	52.87	30.78
2	5000		680.12	68.23
3	100	0.02	0.01	0.08
3	200	0.03	0.01	0.16
3	500	0.05	0.05	4.05
3	1000	0.45	0.35	10.72
3	1500	8.50	1.83	19.64
3	3000	4,356.73	51.29	48.77
3	5000		663.97	101.29

Tabela 2 – Tempos de execução para problemas com um reposicionamento (DH=1)

A Tabela 3 mostra os resultados para dois reposicionamentos permissíveis (DH=2), que são instâncias mais complexas pelo superior número de combinações permitidas. Neste caso, os resultados da Heurística CD em relação a GH variaram de -1,00% a 0,12%, sendo que a média permaneceu em -0,11%. Comparando-se os resultados da Heurística CD com PI, a diferença permaneceu entre 0,32% e 2,17%, com uma média de 1,10%, enquanto que comprando-se os resultados da Heurística GH com PI, a diferença permaneceu entre 0,33% e 2,59%, com uma média de 1,53%. Como observado, para DH=2 os resultados das Heurísticas ficaram mais próximos das soluções obtidas por PI.

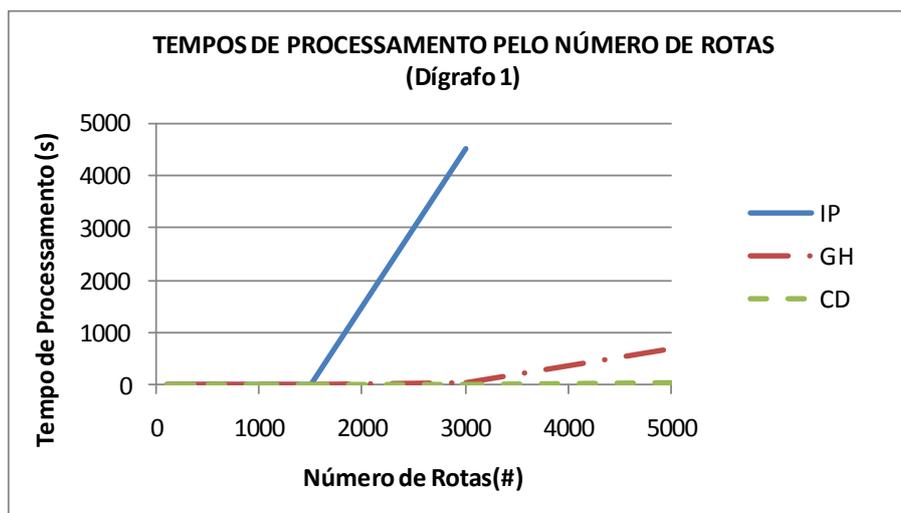


Figura 2 – Tempos de execução para problemas com um reposicionamento (DH=1) e Dígrafo 1

É importante enfatizar que a Heurística CD encontrou solução para todos os problemas sem exceção. Por limitações de memória, foram obtidas soluções para PI apenas até 500 rotas e GH até 1.500 rotas.

A Tabela 4 apresenta os tempos de processamento para dois reposicionamentos permissíveis (DH=2). Os tempos da Heurística CD cresceram de maneira menos acentuada comparada a PI e GH. Este comportamento é evidente na Figura 3, que mostra os tempos de execução versus o número de rotas para o Dígrafo 1.

DÍGRAFO	ROTAS	Resultados (#)			Diferenças (%)		
		PI	GH	CD	CD X GH	CD x PI	GH X PI
1	100	59,751.44	61,297.89	61,049.20	-0.41%	2.17%	2.59%
1	200	114,033.55	116,695.76	115,929.20	-0.66%	1.66%	2.33%
1	500	257,593.37	263,045.72	261,924.40	-0.43%	1.68%	2.12%
1	1000		542,223.43	542,132.80	-0.02%		
1	1500		834,578.42	833,766.60	-0.10%		
1	3000			1,543,930.00			
1	5000			2,565,374.00			
SUBTOTAL DÍGRAFO 1					-0.17%	1.74%	2.24%
2	100	56,052.07	57,481.30	56,989.06	-0.86%	1.67%	2.55%
2	200	100,763.85	102,686.13	101,913.50	-0.75%	1.14%	1.91%
2	500	254,204.86	257,562.58	256,715.20	-0.33%	0.99%	1.32%
2	1000		477,793.62	477,747.80	-0.01%		
2	1500		757,613.50	757,932.40	0.04%		
2	3000			1,481,806.00			
2	5000			2,479,096.00			
SUBTOTAL DÍGRAFO 2					-0.11%	1.12%	1.63%
3	100	52,854.45	53,163.76	53,132.30	-0.06%	0.53%	0.59%
3	200	98,424.45	99,914.85	98,914.20	-1.00%	0.50%	1.51%
3	500	251,236.56	252,073.47	252,037.00	-0.01%	0.32%	0.33%
3	1000		489,138.50	489,070.20	-0.01%		
3	1500		621,364.35	622,081.80	0.12%		
3	3000			1,484,356.00			
3	5000			2,500,852.00			
SUBTOTAL DÍGRAFO 3					-0.03%	0.39%	0.66%
TOTAL					-0.11%	1.10%	1.53%

Tabela 3 – Resultados para problemas com dois reposicionamentos (DH=2)

DÍGRAFO	ROTAS	TEMPO DE PROCESSAMENTO (s)		
		PI	GH	CD
1	100	0.07	0.03	0.17
1	200	0.30	0.12	0.71
1	500	18.37	3.48	2.17
1	1000		44.98	6.30
1	1500		210.65	10.35
1	3000			34.66
1	5000			70.47
<hr/>				
2	100	0.07	0.02	0.16
2	200	0.28	0.12	0.47
2	500	17.68	3.50	2.29
2	1000		45.34	8.96
2	1500		210.72	14.85
2	3000			41.90
2	5000			95.87
<hr/>				
3	100	0.07	0.02	0.09
3	200	0.37	0.12	0.30
3	500	17.73	3.31	2.94
3	1000		43.44	10.52
3	1500		202.25	22.70
3	3000			63.69
3	5000			144.74

Tabela 4 – Tempos de execução para problemas com dois reposicionamentos (DH=2)

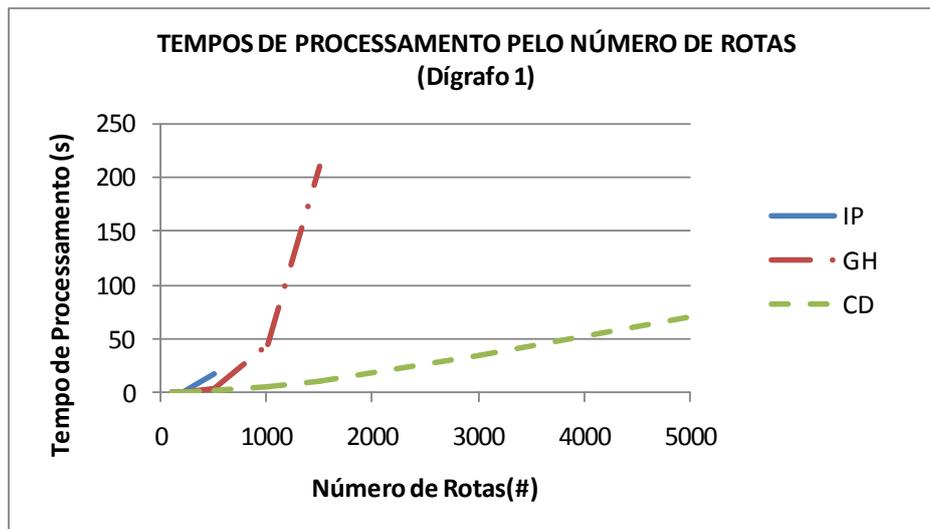


Figura 3 - Tempos de execução para problemas com dois reposicionamentos (DH=2) e Dígrafo 1

4. Análise dos tempos de execução

Em função dos resultados dos tempos de execução obtidos, fez-se uma última análise para verificar a tendência de crescimento desses tempos em função do número de rotas. Para isso, gerou-se outro conjunto de problemas e, a partir de seu resultado, analisou-se o tempo de execução com gráficos e regressões.

A análise foi feita separadamente para os problemas com um ou dois reposicionamentos.

Para o caso de DH=1, tanto GH quanto PI foram ajustadas pela regressão por uma curva exponencial com boa aderência (ambas com R² superior a 95%). A Heurística CD foi ajustada

por uma curva polinomial de grau dois com excelente aderência ($R^2 = 99,91\%$). Dado isso, é possível estimar os tempos para qualquer tamanho de problema. Dessa forma, nota-se que para executar problemas maiores que 5000 rotas, os tempos necessários para PI e GH serão muito maiores se comparados com os de CD.

Para DH=2 GH e PI foram ajustadas novamente pela regressão por uma curva exponencial com boa aderência (ambas com R^2 superior a 95%). O R^2 obtido por PI foi igual a 1, devido ao número pequeno de pontos obtidos (somente 3 pontos).

A Heurística CD foi ajustada por uma curva polinomial de grau dois com excelente aderência ($R^2 = 99,89\%$). Isso permite conjecturar que seu tempo de execução pode ser realmente polinomial.

5. Conclusões

O problema de transporte colaborativo e seu problema de otimização combinatória (PCRCR) são muito interessantes do ponto de vista prático e de vista metodológico. Este trabalho propõe uma heurística Construtiva-Destrutiva que, nesta fase de desenvolvimento, apresentou resultados razoáveis em relação a literatura mas com tempos de processamento excelentes para instâncias maiores quando comparados a literatura anterior, uma vez que não exige que todos os ciclos viáveis sejam gerados. Também demanda menor memória computacional, permitindo que instâncias maiores sejam processadas.

Constatou-se que o tempo de execução de CD é menor para os problemas maiores e mais complexos (por exemplo, aqueles que possuem dois reposicionamentos). Além do mais, o aumento do tempo de execução em função do número de rotas apresentou característica polinomial, ao contrário de GH e PLI que apresentaram comportamento exponencial.

Um método de busca local está sendo atualmente pesquisado para melhorar seu desempenho.

Referências

- CHEN, R. L., S.A. BEYGI, A. COHN, D. R. BEIL, A. SINHA (2009). *Solving truckload procurement auctions over an exponential number of bundles*. Transportation Sci., 43-4, 493–510.
- ERGUN, O., G. KUYZU & M. SAVELSBERGH (2004). *The lane covering problem*. Working paper, GeorgiaTech. <http://www2.isye.gatech.edu/~mwps/publications/ijoc030404.pdf>.
- ERGUN, O., G. KUYZU & M. SAVELSBERGH (2007a). *Shipper collaboration*. Comput. Oper. Res., 34, 1551-1560.
- ERGUN, O., G. KUYZU & M. SAVELSBERGH (2007b). *Reducing Truckload Transportation Costs Through Collaboration*. Transportation Sci., 41-2, 206–221.
- IMMORLICA, N., M. MAHDIAN, & V. S. MIRROKNI (2005). *Cycle cover with short cycles*. In: V. Diekert and B. Durand (Eds.): STACS 2005, LNCS 3404, pp. 641-653. Springer-Verlag, Berlin Heidelberg.
- LIU, R., Z. JIANGA, R.Y.K. FUNGB, F. CHENA, X. LIU (2010). *Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration*. Comput. Oper. Res., 37, 950–959.
- ÖZENER, O. Ö. & O. ERGUN (2008). *Allocating costs in a collaborative transportation procurement network*. Transportation Sci.42-2, 146–165.
- SAVELSBERGH, M., O. ERGUN and G. KUYZU (2004). *Collaborative logistics*. 35th Annual Conference of the Italian Operational Research Society, Lecce, Italy. Available at www2.isye.gatech.edu/~mwps/presentations/AIRO04.pdf.