

Problema do Carteiro Chinês com F.O. não Linear

Franklin Angelo Krukoski

23/11/2020

Resumo

Este trabalho propõe uma abordagem não linear ao problema de roteamento em arcos conhecido como Problema do Carteiro Chinês Capacitado e busca implementá-lo utilizando a linguagem Julia, o pacote de modelagem JuMP e o solver Gurobi.

1 Introdução

O Problema do Carteiro Chinês foi inicialmente proposto por [Mei-Ko, 1962], denominado *Route Inspection Problem*, e trata-se de um problema de otimização que tem por objetivo percorrer todas as arestas de um grafo, de forma a minimizar a distância percorrida. Diferentemente do *TSP* onde cada nó do grafo deve ser percorrido uma única vez, no *PCC* cada aresta pode ser percorrida mais de uma vez, desde que a distância percorrida seja mínima.

Este problema é considerado como problema de roteamento de veículos por possuir forte associação a dois problemas reais, a coleta de resíduos sólidos e a distribuição de correspondências [Beltrami e Bodin, 1974], mas também tem outras aplicações como a montagem de fragmentos de DNA [Pevzner et. al., 2001].

O PCC é formulado matematicamente conforme a seguir:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n C_{i,j} x_{i,j} \quad (1)$$

$$\text{sujeito à } \sum_{j=1}^n x_{j,i} - \sum_{j=1}^n x_{i,j} = 0, \quad i = 1, \dots, n \quad (2)$$

$$x_{i,j} + x_{j,i} \geq 1, \quad \forall (i, j) \in A \quad (3)$$

$$x_{i,j} \in \mathbf{Z}^+ \quad (4)$$

onde:

$x_{i,j}$: número de vezes em que a aresta (i, j) é percorrido de i para j ;

$C_{i,j}$: comprimento ou o custo da aresta (i, j) .

A função objetivo (1) minimiza o a distância total a ser percorrida, que pode ser interpretada como o custo para percorrer a aresta (i,j) . A restrição (2) garante a continuidade da rota, a restrição (3) garante que nenhuma aresta deixará de ser percorrida e a restrição (4) impõem que as variáveis sejam não negativas e inteiras.

Nas últimas décadas desenvolveram-se diversas variantes do problema, dentre as quais destacam-se: *PCCR - Problema do Carteiro Chinês Rural*, onde apenas um subconjunto das arestas precisam ser percorridas, *CARP - Problema de Roteamento em Arcos Capacitados* em que cada aresta possui uma demanda a ser atendida, *PCW - Problema do Carteiro ao Vento* que consiste em determinar um passeio fechado de custo mínimo passando por todas as arestas do grafo e *PCCH - Problema do Carteiro Chinês Hierárquico* onde os arcos são divididos em grupos e uma relação de precedência é definida entre os grupos.

Posteriormente tais variantes complementam-se como em [Nossack, 2017], onde o problema do carteiro rural ao vento com opção de percorrer a rua em zigzag durante certos períodos do dia, para ruas com tráfego leve e apontam esse tipo de problema para situações reais como coleta de resíduos e entrega de jornais. O trabalho de [Kayaci e Yilmaz, 2018] introduzem uma nova abordagem denominada Problema do Carteiro Chinês Hierárquico dependente de tempo, que tem como objetivo minimizar o tempo total de viagem enquanto obedece relações de precedência entre os arcos. [Tirkolaee, 2018] propõe um modelo para o Problema de Roteirização de Arcos Periódicos Capacitados, considerando múltiplas viagens e motoristas e o período de trabalho da tripulação e tem por objetivo minimizar a distância total percorrida e custo total de utilização dos veículos.

Porém, a maioria dos problemas pesquisados e apresentados correspondem exclusivamente a problemas de programação linear. Não foram constatados, na breve pesquisa que fundamentou este trabalho, abordagens não lineares para a Função Objetivo ou restrições. Evidencia-se que a complexidade de tais problemas, por serem do tipo NP-Hard, exigem na maioria dos casos e principalmente para os problemas de grande porte, abordagens Heurísticas de solução.

Este trabalho busca apresentar uma abordagem não linear para compor o custo de deslocamento dos veículos em um Problema do Carteiro Chinês Capacitado, por constatar-se aumento significativo no consumo de combustível a medida que a capacidade de caminhões de coleta aumente. Para tal, é considerado nesse problema o momento que o caminhão realiza o atendimento no percurso (i, j) . O problema e sua formulação matemática está descrito na seção 2. Os resultados estão brevemente dispostos na seção 3 e no repositório do GitHub, conforme referências e a seção 4 apresenta a análise e discussão dos resultados.

2 Problema

No PCC Capacitado o custo normalmente é tratado de forma linear, representando a distância entre os pontos i e j [Ding, 2014]. Além disso, considerando que o trajeto não é realizado em uma única viagem, por questões de tempo ou capacidade do veículo, serão consideradas as restrições de capacidade do veículo. No entanto sabemos, por exemplo, que existem custos associados ao consumo de combustível que são não lineares, e que o gasto de combustível é diferente dependendo da carga do caminhão no momento em que a aresta (i, j) percorrida.

No caso da coleta de resíduos sólidos, temos gastos diferentes de combustível a medida que o caminhão realiza coleta. O consumo de combustível pode quadruplicar, dependendo do modelo e da capacidade do veículo ¹. Dessa forma, considerar a ordem que cada aresta será atendida, afim de minimizar a distância percorrida com o caminhão em sua capacidade máxima pode reduzir consideravelmente os custos de deslocamento.

Considerando estes custos não lineares associados ao gasto de combustível, e as demais restrições referentes ao PCC Capacitado, temos o seguinte modelo:

$$\min_x Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{p \in S} C_{i,j} x_{i,j}^p * C_{i,j}^{t-1,p} \quad (5)$$

onde: $C_{i,j}^{t-1,p} = (1 + t/T)$ ou $C_{i,j}^{t-1,p} = \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^{t-1} \sum_{p=1}^P l_{i,j}^{t-1,p} q_{i,j}^p$

$$\text{sujeito a } \sum_{j=1}^n \sum_{t=1}^T x_{i,j}^{p,t} = \sum_{j=1}^n \sum_{t=1}^T x_{j,i}^{p,t} \quad i = 1, \dots, n, p \in S \quad (6)$$

$$\sum_{t=1}^T x_{i,j}^{p,t} = x_{i,j}^p \quad \forall (i, j) \in A, p \in S \quad (7)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{i,j}^{p,t} \leq 1 \quad \forall t \in T, p \in S \quad (8)$$

$$\sum_{p=1}^S \sum_{t=1}^T (l_{i,j}^{p,t} + l_{j,i}^{p,t}) = \lceil \frac{q_{i,j}}{w} \rceil \quad \forall (i, j) \in A \quad (9)$$

$$\sum_{t=1}^T l_{i,j}^{p,t} = l_{i,j}^p \quad \forall (i, j) \in A, p \in S \quad (10)$$

$$\sum_{t=1}^T x_{i,j}^{p,t} \geq \sum_{t=1}^T l_{i,j}^{p,t} \quad \forall (i, j) \in A, p \in S \quad (11)$$

¹Conforme pesquisa realizada na secretaria de Meio Ambiente da prefeitura de Francisco Beltrão - Pr, em outubro de 2020

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^T l_{i,j}^{p,t} q_{i,j} \leq w \quad p \in S \quad (12)$$

$$\sum_{j=1}^n f_{i,j}^p - \sum_{j=1}^n f_{j,i}^p = \sum_{j=1}^n \sum_{t=1}^T l_{i,j}^{p,t} \quad \forall i \in A \setminus \{1\}, p \in S \quad (13)$$

$$f_{i,j}^p \leq (n^2) x_{i,j}^p, \quad \forall (i,j) \in A, p \in S \quad (14)$$

$$f_{i,j}^p \geq 0 \quad \forall (i,j) \in A, p \in S \quad (15)$$

$$q_{i,j} > 0 \quad \forall (i,j) \in A \quad (16)$$

$$l_{i,j}^{p,t} \in 0, 1 + \quad \forall (i,j) \in A, p \in S, t \in T \quad (17)$$

$$x_{i,j}^{p,t} \in \mathbf{Z}^+ \quad \forall (i,j) \in A, p \in S, t \in T \quad (18)$$

onde: w é a capacidade dos $s \in S$ caminhões, $q_{i,j}$ são as demandas de cada arco, $l_{i,j}^p$ assume valor 1 se o caminhão p atende o arco (i,j) e $f_{i,j}^p$ assume valor positivo se $x_{i,j}^p \geq 1$.

As demais restrições referem-se à: (6) continuidade das rotas dos caminhões, (7) atendimento da aresta (i,j) em um único período t , por algum caminhão, (8) cada caminhão atravessa uma aresta uma única vez no período t , (9) arestas sejam atendidas em suas demandas, (10) atravessar a aresta (i,j) em um único período t , por algum caminhão, (11) obrigam que os caminhões percorram os arcos que lhes forem designados a atender, (12) que os recolhimentos não ultrapassem a capacidade do caminhão, (13) e (14) garantem que não sejam criadas sub-rotas, de acordo com o modelo proposto por [Golden e Wong, 1981] e (15), (16), (17) e (18) definem o domínio das variáveis do modelo.

As expressões que calculam $Cc_{i,j}^{t-1,p}$ de duas formas e cada uma delas é trabalhada separadamente, em dois grupos distintos de cenários. A primeira fórmula para Cc acrescenta um percentual ao custo de travessia de um arco, a medida que as coletas são realizadas. Esse acréscimo corresponde a razão entre arestas visitadas e previstas de serem visitas. Dessa forma espera-se que as arestas com maior custo sejam coletadas primeiro, na medida do possível. Enquanto a segunda acrescenta ao custo proporcionalmente à demanda previamente coletada em relação a capacidade total do caminhão, no mesmo sentido que o caso anterior. Destaca-se que a segunda forma de formular o custo é mais próxima da realidade.

Foram criados diferentes cenários para medir a capacidade do modelo em encontrar soluções ótimas em tempo computacional razoável, em um computador com configurações medianas², cujos parâmetros e resultados são descritos e discutidos a seguir.

3 Resultados e Discussão

Os *scripts* contendo as funções e os cenários utilizados neste trabalho estão dispostos em <https://github.com/FKrukoski/Projeto1-Otimiz.N.L.>

²Intel Core i5-3230M, 2.6GHz(x4), 8GB de RAM

Alguns cenários foram elaborados com o intuito de validar e testar o modelo que contem a primeira descrição da função Cc em situações diversas, com as seguintes características: Número de nós variando entre 9, 16 ou 25. As arestas são construídas em até 3 pares de pontos, considerando-se os mais próximos. O grafo é sempre bidirecionado, ou seja, a distância de i para j é igual a distância de j para i . As demandas de cada aresta são valores aleatórios entre 50 e 100, uniformemente distribuídos e independem do sentido em que são atendidas (de i para j ou vice-versa). Serão considerados tantos períodos quantos forem necessários para que o número de caminhões escolhido, que varia em cada situação, tenha capacidade de coletar toda a demanda com alguma folga, ou seja, a soma da demanda do grafo será menor que a soma das capacidades dos caminhões. Desta forma a demanda dos caminhões também se ajusta aos cenários para garantir a factibilidade. Para esse primeiro conjunto de cenários, o tempo de solução foi limitado em 180s. Os cenários são descritos na Tabela 1, a seguir:

Tabela 1: Cenários - Modelo 1						
Cenários	Adjac	Demanda	CapacViagens	Períodos	Veículos	Variáveis
Cenário 1	28	906.0	500	16	2	5670
Cenário 2	28	906.0	350	10	3	5589
Cenário 3	48	1858.0	1000	25	2	27136
Cenário 4	48	1858.0	700	17	3	28416
Cenário 5	48	1858.0	600	14	4	31744
Cenário 6	48	1858.0	500	10	5	29440
Cenário 7	72	2699.0	2750	72	1	91875
Cenário 8	72	2699.0	1500	38	2	98750
Cenário 9	72	2699.0	1000	26	3	103125

Fonte: O autor, 2020

Os cenários 1 e 2 foram criados com 9 pontos, os cenários 3, 4, 5 e 6 com 16 e os cenários 7, 8 e 9 com 25 pontos. Observa-se que ao se considerar mais pontos no grafo, o número de variáveis cresce consideravelmente, porém, conforme a Tabela 2, o que torna o problema realmente difícil e faz com que o solver não seja capaz de encontrar uma solução ótima (nem factível, em alguns casos) em pouco tempo é o aumento no número de veículos, já que essa condição aumenta significativamente o número de restrições e consequentemente exige maior tempo computacional para cada iteração.

Conforme a Tabela 2, foi possível obter soluções ótimas nos cenários que apresentaram menor número de veículos, e os tempos computacionais não variaram muito, mesmo aumentando significativamente o número de variáveis.

As soluções obtidas nos Cenários 2, 3 e 8 são representadas na Figura 1. As rotas com veículos diferentes são representadas por cores diferentes e deslocadas para facilitar a visualização.

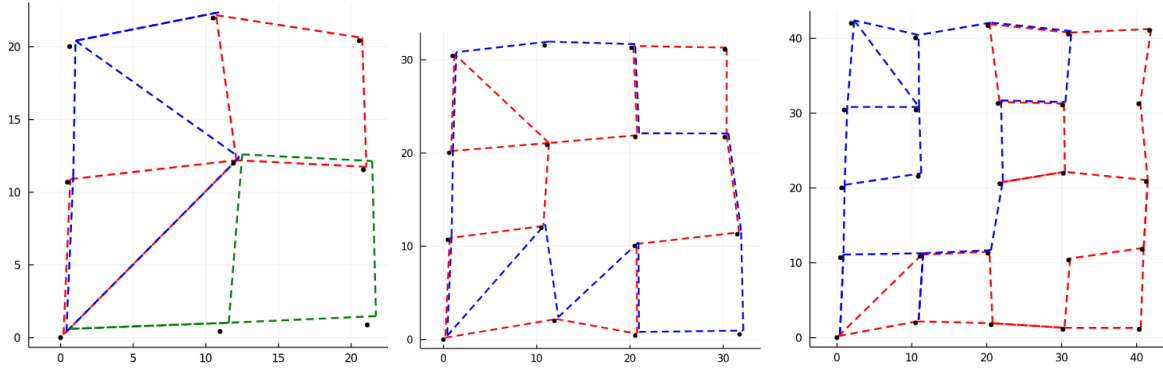
Os cenários 4, 5 foram computados novamente, ampliando-se o limite de tempo para 30 minutos e as métricas obtidas estão dispostas na Tabela 3. Com essa alteração, o Solver foi capaz de obter a solução ótima no cenário 4*, após realizar mais de 10 vezes o número de iterações. Para o cenário 5*, não foi obtida uma solução ótima, mas o Gap reduziu consideravelmente e acredita-se que em mais algum tempo, a solução ótima seria obtida. Esse resultado mostra a fragilidade em se tratar esse modelo com uma matriz mal

Tabela 2: Resultados - Modelo 1

Cenários	SimplexIters	TimeElapsed	Solução	Gap
Cenário 1	15965.0	2.16	ótima	0.0
Cenário 2	1.08e6	59.61	ótima	0.0
Cenário 3	27142.0	5.11	ótima	0.0
Cenário 4	2.44e6	180	Limitada pelo tempo	0.12
Cenário 5	905969.0	180	Limitada pelo tempo	0.27
Cenário 6	492683.0	180	Sem soluções viáveis	100.0
Cenário 7	12856.0	6.69	ótima	0.0
Cenário 8	672522.0	63.92	ótima	0.0
Cenário 9	319552.0	180	Sem soluções viáveis	100.0

Fonte: O autor, 2020

Figura 1: Cenários 2, 3 e 8 - Resultados



condicionada e esparsa.

Tabela 3: Cenários 4 e 5 - 30 min

Cenários	SimplexIters	TimeElapsed	Solução	Gap
Cenário 4*	3.21e7	1678.07	ótima	0.0
Cenário 5*	3.27e7	1800	Limitada pelo tempo	0.09

Fonte: O autor, 2020

Os cenários 6 e 9 foram executados novamente com um limite de tempo de 60 minutos, e os resultados estão apresentados na Tabela 4. Apesar do aumento no limite no tempo de execução, nenhum dos dois cenários obteve solução ótima. Porém, pode ser observado que o cenário 9*, por ter menos veículos que o cenário 6*, chegou mais próximo da solução exata, apesar de ter quase três vezes mais variáveis binárias e inteiras. Isso evidencia que a maior dificuldade desse tipo de problema é adicionar mais veículos para percorrerem as arestas.

Posteriormente, partindo dos cenários que obtiveram solução ótima em menos que 3 minutos no primeiro grupo, foram executados para o modelo 2, conforme descrito anteriormente. Porém, apenas o cenário 1 pode ser executado integralmente. Novos cenários intermediários foram criados para explorar as limitações do modelo, com um número menor de pontos e veículos que o cenário 3. Os resultados dos cenários criados para o Modelo 2 estão dispostos na Tabela 5.

Tabela 4: Cenários 6 e 9 - 60 min

Cenários	SimplexIters	TimeElapsed	Solução	Gap
Cenário 6*	1.72e7	3600	Limitada pelo tempo	0.96
Cenário 9*	4.87e7	3600	Limitada pelo tempo	0.04

Fonte: O autor, 2020

Tabela 5: Cenários - Modelo 2

Cenários	Adjac	Demanda	CapacViagens	Períodos	Veículos	Variáveis
Cenário M2-1	28	906.0	500	16	2	5670
Cenário M2-2	30	1041.0	600	16	2	7000
Cenário M2-3	36	1310.0	700	19	2	11808
Cenário M2-4	42	1542.0	1600	44	1	20475

Fonte: O autor, 2020

Os cenários descritos na Tabela 5 são compostos por 9 (3x3 pontos), 10 (5x2 pontos), 12(3x4 pontos) e 15 (3x5 pontos), respectivamente. Não foi possível criar cenários maiores por conta da memória interna limitada do computador utilizado³. Os cenários foram resolvidos até a solução ótima, em poucos segundos, como disposto na Tabela 6. A estrutura do modelo, da forma como está implementado não contém nem muitas variáveis, nem muitas restrições, mas a função objetivo contém uma expressão com muitas variáveis (evidentemente, linearmente dependentes). Algumas das soluções estão dispostas na Figura 2.

Tabela 6: Resultados - Modelo 2

Cenários	SimplexIters	TimeElapsed	Solução	Gap
Cenário M2-1	20109.0	5.83	ótima	0.0
Cenário M2-2	99715.0	14.12	ótima	0.0
Cenário M2-3	69608.0	19.16	ótima	0.0
Cenário M2-4	3107.0	24.19	ótima	0.0

Fonte: O autor, 2020

Os resultados evidenciados na Figura 2 indicam as soluções ótimas com poucas arestas sendo atravessadas duas vezes. Infelizmente o padrão de grafo, criado aleatoriamente, não permitiu que se verificasse graficamente o real sentido da função objetivo, pois as arestas tem tamanhos semelhantes e demandas que variam em um intervalo comportado.

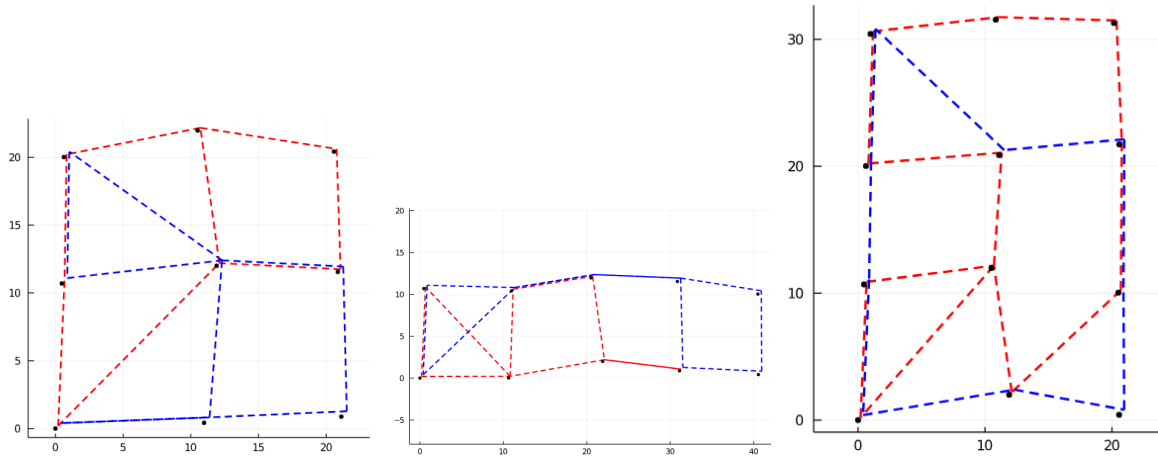
Nota-se que a maneira que a função objetivo do Modelo 2 foi escrita não está em sua forma mais eficiente. Dessa maneira uma matriz muito grande de expressões precisa ser calculada mesmo para uma pequena quantidade de pontos, veículos e tempos de atendimento.

4 Considerações Finais

O Problema do Carteiro Chinês é um problema clássico muito estudado em programação linear, tem sido trabalhado e solucionado principalmente por Heurísticas e

³O limite de memória atingido e o modelo não criado

Figura 2: Cenários 1, 2 e 3 - M2 - Resultados



Meta-heurísticas nos últimos anos. Torná-lo não linear, adicionando o tempo como fator condicionante para a escolha das designações das arestas à múltiplos caminhões tornou-o ainda mais complexo e de difícil solução, principalmente tratando do modelo matemático exato. Apesar da evolução dos *solvers* comerciais, como o Gurobi, essa abordagem não é recomendada, principalmente para execução em computadores medianos, pois isso limita muito o tamanho dos problemas que podem ser tratados. Apesar disso, o Solver foi capaz de solucionar problemas pequenos com solução exata.

Uma possível abordagem seria a utilização de heurísticas para a criação de uma solução factível inicial e, a partir dela, buscar por melhorias com o método exato. Porém essa metodologia não foi contemplada nesse trabalho. Outra possibilidade seria implementar o problema considerando apenas as variáveis cujas adjacências existam, ao invés da matriz completa, e esparsa, que foi considerada no problema. Isso reduziria consideravelmente o número de variáveis do problema e talvez permitisse explorar uma variedade maior de problemas.

Outras formas de não linearização poderiam ser pensadas para a função objetivo, porém a apresentada nesse trabalho foi julgada como a mais significativa em relação ao custo em se atravessar um arco qualquer. Percebeu-se que a maneira que a função objetivo foi implementada, com uma expressão recursiva não foi eficiente, pois não permitiu explorar problemas maiores por falta de memória.

Referências

- Beltrami, E. J. Bodin, L. D. Networks and vehicle routing for municipal waste collection. *Networks*, vol. 4, 1974. p.65-94.
- Ding, H. Li, J. Lih, K-H. Approximation algorithms for solving the constrained arc routing problems in mixed graphs. *European Journal of Operational Research*, 239(1), 2014. p.80-88.
- Golden, B. L. Wong, R. T. Capacitated Arc Routing Problems. *Networks*, vol. 11, 1981.

p.305-315.

Kayacı Çodur, M., Yılmaz, M. A time-dependent hierarchical Chinese postman problem. *Central European Journal of Operational Research* v. 28, p. 337–366, 2018.

Mei-Ko, K. Graphic Programming Using Odd or Even Points. *Chinese Mathematics*, vol.1, 1962. p.273-277.

Nossack, J et al. The windy rural postman problem with a time-dependent zigzag option. *European Journal of Operational Research*, v. 258, p. 1131-1142, 2017.

Pevzner, P. A. Tang, H. Waterman, M. An Eulerian path approach to DNA fragment assembly, In: *Proceedings of the National Academy of Sciences* 98 v.17, 2001, p. 9748-9753.

Tirkolaee, E. B.; Mahdavi, I.; Esfahani, M. M. S. A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time. *Waste Management*, v.76, p.138-146, 2018.

Comentários adicionais pós-apresentação

Foram executados novamente os cenários do modelo 1, após as considerações levantadas, principalmente em relação à variável $x_{i,j}^p$, onde passei a desconsiderá-la na função objetivo e nas restrições, considerando o somatório correspondente no lugar dela. As funções foram ajustadas conforme o arquivo mais recente, postado no GitHub. A seguir, seguem a repetição das tabelas 1 e 2, intercaladas com as tabelas resultantes das correções. Apesar da diminuição do número de variáveis, o modelo precisou de muito mais iterações e mais tempo para atingir a solução ótima, quando o fez. Apesar de não saber de antemão, acredito que seja interessante manter as variáveis que expressam a soma dos períodos em cada arco, tanto para x, como para l.

Tabela 7: Cenários - Modelo 1

Cenários	Adjac	Demanda	CapacViagens	Períodos	Veículos	Variáveis
Cenário 1	28	906.0	500	16	2	5670
Cenário 2	28	906.0	350	10	3	5589
Cenário 3	48	1858.0	1000	25	2	27136
Cenário 4	48	1858.0	700	17	3	28416
Cenário 5	48	1858.0	600	14	4	31744
Cenário 6	48	1858.0	500	10	5	29440
Cenário 7	72	2699.0	2750	72	1	91875
Cenário 8	72	2699.0	1500	38	2	98750
Cenário 9	72	2699.0	1000	26	3	103125

Tabela 8: Cenários - Modelo 1 - após considerações

Cenários	Adjac	Demanda	CapacViagens	Períodos	Veículos	Variáveis
Cenário 1	28	906.0	500	16	2	5346
Cenário 2	28	906.0	350	10	3	5103
Cenário 3	48	1858.0	1000	25	2	26112
Cenário 4	48	1858.0	700	17	3	26880
Cenário 5	48	1858.0	600	14	4	29696
Cenário 7	72	2699.0	2750	72	1	90625
Cenário 8	72	2699.0	1500	38	2	96250
Cenário 9	72	2699.0	1000	26	3	99375

Tabela 9: Resultados - Modelo 1

Cenários	SimplexIters	TimeElapsed	Solução	Gap
Cenário 1	15965.0	2.16	ótima	0.0
Cenário 2	1.08e6	59.61	ótima	0.0
Cenário 3	27142.0	5.11	ótima	0.0
Cenário 4	2.44e6	180	Limitada pelo tempo	0.12
Cenário 5	905969.0	180	Limitada pelo tempo	0.27
Cenário 6	492683.0	180	Sem soluções viáveis	100.0
Cenário 7	12856.0	6.69	ótima	0.0
Cenário 8	672522.0	63.92	ótima	0.0
Cenário 9	319552.0	180	Sem soluções viáveis	100.0

Tabela 10: Resultados - Modelo 1 - após considerações

Cenários	SimplexIters	TimeElapsed	Solução	Gap
Cenário 1	421235	28.09	ótima	7.57e-6
Cenário 2	2.76e6	180.01	Limitada pelo tempo	0.087
Cenário 3	36528.0	8.02	ótima	0.0
Cenário 4	1.06e6	180	Limitada pelo tempo	0.23
Cenário 5	1.07e6	180	Limitada pelo tempo	0.31
Cenário 7	291911	52.66	ótima	0.0
Cenário 8	491806	180	Limitada pelo tempo	0.02
Cenário 9	568337	180	Limitada pelo tempo	0.35