

Utilização de programação dinâmica para geração de trechos iniciais na resolução de um problema de inspeção da malha ferroviária

Guilherme Vinicyus Batista (UFPR) guivbatista89@gmail.com

Cassius Tadeus Scarpin (UFPR) cassiusts@gmail.com

José Eduardo Pécora Jr. (UFPR) pecora@ufpr.br

Resumo:

O problema de inspeção em ferrovias se enquadra na classe dos Problemas de Roteamento Periódico e Capacitado em Arcos (PCARP). Um ou mais carros devem atender a demanda periódica de inspeção dos trilhos dentro de um horizonte de planejamento. É um problema de difícil resolução que requer novas abordagens almejando soluções melhores. Esse artigo inicialmente utiliza programação dinâmica para gerar rotas, e então reconhecer trechos potenciais que podem estar presente em boas soluções. Os trechos foram usados para formar espaços restritos que, quando explorados, trouxeram boas soluções com baixo tempo computacional.

Palavras chave: PCARP, Programação Dinâmica, Espaço Restrito.

Using dynamic programming to generate initial stretches for the resolution of the inspection problem in the railroad network

Abstract

The problem of inspection in railroads receives the classification of Periodic Capacitated Arc Routing Problem (PCARP). One or more vehicles must perform the periodical demands of rails inspection during the planning horizon. It is a problem of difficult resolution and it requires new approaches to achieve better solutions. This paper, firstly, uses dynamic programming to generate routes and then recognizes potential stretches that may be present in good solutions. The stretches were used to create restricted spaces, its exploration allowed the achievement of good solutions with small computation time.

Key-words: PCARP, Dynamic Programming, Restricted Space.

1. Introdução

A inspeção dos trilhos é uma atividade fundamental para as empresas que atuam no ramo ferroviário. O objetivo dessa tarefa é garantir o fluxo de veículos e evitar possíveis acidentes em virtude do desgaste dos trilhos, ou seja, evitar grandes prejuízos. Esse problema é classificado como Problema de Roteamento Periódico e Capacitado em Arcos (BATISTA; SCARPIN, 2015). Veículos homogêneos e com capacidade limitada são responsáveis pela inspeção dos trilhos para garantir o fluxo nas vias. Essa capacidade é associada a deslocamentos diários que os carros conseguem fazer entre duas cidades.

Como a malha ferroviária, em um país como o Brasil, é extensa, os veículos devem fazer diariamente um grande percurso, tornando inviável seu retorno até um depósito. Sendo assim, a rota deve ser bem planejada e contínua, de modo que o local onde um movimento é finalizado em um respectivo dia é o ponto de partida no próximo. A malha ferroviária está dividida em

trechos, segmentos da malha ligando duas cidades. A periodicidade de cada trecho é dada segundo um intervalo de tempo máximo em que eles devem ser analisados ao menos uma vez. O horizonte de tempo utilizado para planejar a inspeção dos trilhos deve ser no mínimo igual a maior periodicidade e a solução encontrada deve ser cíclica. A periodicidade de cada trecho pode até não ser respeitada, porém, quando isso ocorre, um custo de atraso, punição, deve ser levado em conta.

O objetivo desse artigo é apresentar um algoritmo que auxilia na limitação do espaço solução do problema na tentativa de alcançar resultados bons e rápidos. O PCARP é um problema NP-Hard, resultados ótimos podem ser alcançados para pequenas instâncias, porém isso não ocorre para instâncias maiores. Para aumentar a eficiência do modelo proposto limitou-se o espaço de busca do problema. Um modelo de Programação Dinâmica gera rotas que são analisadas e, a partir da análise, são identificados movimentos padrões para gerar espaços restritos. A busca de soluções em espaços restritos tende a ser mais efetiva, isso porque muitas das possíveis soluções são descartadas.

O Capítulo 2 apresenta uma revisão da literatura sobre os problemas de roteamento em arcos e suas aplicações que, com o decorrer do tempo, culminam no PCARP. O Capítulo 3 descreve mais detalhes do problema de inspeção de trilhos, apresenta um modelo matemático e soluções para dois grupos de instâncias. O Capítulo 4 apresenta a metodologia adotada na geração de espaços restritos e os resultados de tal abordagem. Finalmente, as considerações finais estão no Capítulo 5.

2. Revisão da literatura

O principal objetivo do Problema de Roteamento em Arcos (PRA), em inglês *Arc Routing Problems*, é determinar o menor custo para atravessar um conjunto de arcos de um grafo, com ou sem restrições (EISELT; GENDREAU; LAPORTE, 1995). Um apanhado dos problemas de roteamento em arcos estudados na última década dos anos 2000 pode ser encontrado em (CORBERÁN; PRINS, 2010). Muitos fatores podem influenciar os problemas de roteamento, gerando uma complexidade cada vez maior. Entre os fatores pode-se encontrar: número de domicílios, tempo para servir um determinado nó ou arco, tamanho da frota, natureza da demanda, localização da demanda, restrições na capacidade dos veículos, natureza do grafo, entre outros (KONOWALENKO, 2012).

Alguns dos problemas de roteamento em arcos já estão bem consolidados na literatura e sofrem variações que visam deixar o comportamento de seus modelos cada vez mais parecido com a realidade. É o caso do Problema do Carteiro Chinês (TAUBE, 1992), o Problema do Carteiro Rural (PCR) e o Problema de Roteamento Capacitado em Arcos, *Capacitated Arc Routing Problem* (CARP). Por exemplo, o Problema do Carteiro Rural Periódico (GHIANI et al., 2005), é uma variante do PCR que vem ao encontro do desenvolvimento desse trabalho. Nele, uma rota deve ser obtida com custo mínimo, sendo planejada para múltiplos períodos e cada arco deve ser visitado um determinado número de vezes nesse longo período.

Normalmente os veículos tem uma determinada capacidade que está associada a carga máxima, ou a máxima capacidade de trabalho ou, ainda, a uma restrição de distância máxima a ser percorrida (CORBERÁN; PRINS, 2010). Baseado nisso, o CARP foi introduzido em 1981, um problema muito rico em teoria e aplicabilidade que era negligenciado até então (GOLDEN; WONG, 1981). Uma demanda é associada a cada arco de um grafo: veículos de uma frota com capacidade W devem atravessar esses arcos coletando ou entregando as determinadas demandas sem exceder a capacidade. O CARP básico da literatura envolve apenas períodos simples, um dia, e um grafo não direcionado no qual cada arco pode ser atravessado em

qualquer direção (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a). Para decisões que são tomadas diariamente, o CARP é muito utilizado e eficaz. Quando necessário abordar problemas que vão além do nível operacional, sendo ele o nível tático ou estratégico, o CARP deixa a desejar. Trabalhar com um horizonte de tempo maior implica em problemas mais complexos de serem resolvidos, dando origem ao PCARP.

Entre as primeiras vezes que o PCARP foi abordado (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002b) ele foi chamado de *arc routing planning problem*, problema de planejamento em roteamento em arcos. Recebeu esse nome por fazer uma correspondência com o problema de planejamento da produção e tratar de decisões táticas. A importância do período no roteamento de veículos vem sendo abordada a muito tempo, pois isso permite a redução de custos de distribuição ao longo dos dias em questão (CHRISTOFIDES; BEASLEY, 1984). Um trabalho pioneiro relacionado ao assunto pode ser encontrado em (EGLESE, 1994) no qual um roteamento com características de um PCARP é proposto. O problema proposto por Eglese (1994) estava relacionado a um contexto de remoção de neve nas vias dos países com inverno rigoroso, algumas vias precisavam ser tratadas com “sal” a cada duas horas e outras a cada quatro horas.

O PCARP pode ser visto como uma extensão natural do CARP, ao invés do problema ser tratado para um único dia é então expandido para múltiplos períodos (MONROY; AMAYA; LANGEVIN, 2013). Aproximações através do simples modelo do CARP são dificilmente aplicadas (MEI et al., 2011). A resolução do PCARP implica em conciliar as decisões de nível tático e operacional simultaneamente levando em conta todo o horizonte de tempo. Porém, em sua resolução algumas complicações podem aparecer como, por exemplo, demandas flutuantes ao longo do horizonte de tempo ou espaçadas em intervalos de tempo (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005).

Aplicações diferenciadas do PCARP vem sendo publicadas recentemente; na tentativa de englobar todas suas variantes, O PCARP pode ser definido como: problema de roteamento em arcos envolvendo um horizonte de tempo composto por mais de um período, sendo que esses períodos podem ser formados por qualquer fração temporal. Um ou mais veículos ao fazerem suas rotas devem executar da melhor maneira possível as tarefas que estão associadas a cada arco sem exceder sua capacidade e respeitando as frequências exigidas.

A abordagem padrão do PCARP é possuir custos associados ao atravessamento dos arcos e outros custos para servi-los. São custos não negativos que as vezes podem até mesmo ser iguais. Um arco pode ser atravessado quantas vezes forem necessárias, porém os atendimentos devem ocorrer nos momentos planejados. É assumido também que as demandas individuais nunca excedem a capacidade do carro, a exemplo do CARP, para evitar atendimentos germinados (CHU; LABADI; PRINS, 2006).

O PCARP está restrito ao que pode ser transportado por um veículo (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a), isto é, pode-se entender como o aproveitamento máximo do esforço/capacidade deste. É um problema considerado NP-Hard por englobar o CARP. O PCARP por ser periódico pode ter um espaço de solução muito maior que o CARP. Além disso, é necessário resolver um problema de alocação e definir os dias em que os arcos devem ser servidos (ZHANG et al., 2016). Em uma variação mais recente, até mesmo um problema de localização de facilidades foi incorporado ao contexto do PCARP (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2016).

Aplicações do PCARP são encontradas quanto trata-se, por exemplo, de: coleta de lixo (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005), remoção de neve em países com inverno rigoroso, aplicação de herbicidas em ferrovias, entrega de correspondências, inspeção de linhas

elétricas, monitoramento, inspeção e manutenção de rodovias (MARZOLF; TRÉPANIER; LANGEVIN, 2006; MONROY; AMAYA; LANGEVIN, 2013) e ferrovias (BATISTA, 2014), irrigação de estradas não pavimentadas em minas ao céu aberto (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a, 2014b, 2016), inspeção de encanamentos, irrigação de árvores na beira de estradas (HUANG; LIN, 2014). A resolução muitas vezes não é exata, sendo ela por heurísticas como: Algoritmos Meméticos (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005; MEI et al., 2011; ZHANG et al., 2016); *Scatter Search* (CHU; LABADI; PRINS, 2006); heurística baseada em duas fases, primeiramente agrupamento e em seguida roteirização (MONROY; AMAYA; LANGEVIN, 2013); Colônia de Formigas (KANSOU; YASSINE, 2009).

3. O problema

A inspeção da malha ferroviária é um problema cíclico, pois após o fim do horizonte planejado a mesma rota deve ser utilizada para o próximo planejamento. Ou seja, deve existir uma replicação do que foi feito, de modo que os dias seguintes ao horizonte de tempo devem ser considerados na resolução do problema. A inspeção é um tipo de serviço regular no qual a frequência de passagem em cada trecho é determinada por um intervalo de tempo entre atendimentos, a chamada periodicidade. A frequência tratada como intervalos de tempo ou periodicidade consiste na definição de um mínimo espaço de tempo entre a execução de cada tarefa e/ou um máximo espaço de tempo, esse caso trata-se de um máximo espaço de tempo entre serviços.

A formulação é baseada em um grafo não direcionado $G = (X, E)$ com n pontos, logo, $X = \{x_1, x_2, \dots, x_n\}$, e m arestas com $E = \{e_1, e_2, \dots, e_m\}$ que devem ser percorridos por nk carros definidos pelo conjunto $K = \{1, 2, \dots, nk\}$. Cada arco e é formado por um par de nós e é chamado de $x_{ij} = (x_i, x_j)$ ou $[i, j]$ e é associado a um custo c_{ij} . Para o roteamento, cada aresta é formada por um par de arcos direcionados que representam cada um dos sentidos possíveis de locomoção. A capacidade de cada carro é convertida na máxima distância que um carro pode percorrer em um dia, o que corresponde a uma aresta, fato que simplifica o modelo.

A periodicidade de cada arco é expressa na quantidade máxima de períodos consecutivos em que o arco deve ser atendido ao menos uma vez $MP(x_{ij})$. Por exemplo, se a periodicidade for de 15 dias, entre o dia 1 e o dia 15 deve ocorrer ao menos um atendimento, entre os dias 2 e 16 também e assim por diante. Os arcos que requerem apenas uma passagem ao longo de todo horizonte de tempo formam o subconjunto R , já os demais formam o subconjunto S , ou seja, $E = R \cup S$. Tem-se que o horizonte de tempo H , é formado por np períodos $H = \{1, 2, \dots, np\}$, no qual cada período é simbolizado pela letra p . O modelo é formulado a partir das seguintes variáveis:

x_{ijkp}	$\left\{ \begin{array}{l} 1, \text{ se o carro } k \text{ se desloca do ponto } i \text{ para o ponto } j \text{ no período } p \\ 0, \text{ caso contrário} \end{array} \right.$
pn_{ijp}	$\left\{ \begin{array}{l} 1, \text{ se o arco } x_{ij} \text{ não respeita a periodicidade no período } p \\ 0, \text{ caso contrário} \end{array} \right.$
f_{ikp}	$\left\{ \begin{array}{l} 1, \text{ se o carro } k \text{ fica parado no ponto } i \text{ no dia } p \\ 0, \text{ caso contrário} \end{array} \right.$
y_i	Contabiliza a quantidade de carros que sai do ponto i no dia 1 e chega no ponto i no dia np

Os índices i e j estão associados a cada um dos arcos $[i, j]$, podendo ser eles seus pontos finais ou iniciais. O índice k aos carros e o índice p aos períodos. As variáveis pn_{ijp} e f_{ikp} foram criadas para que o problema tenha viabilidade. Cada variável pn_{ijp} tem associado a ela um custo PU_{ij} , que é uma punição caso a periodicidade de um arco não seja atendida. Isso permite que na modelagem proposta haja um atraso no atendimento de alguns arcos, visto que essa quantidade de atrasos deve ser a menor possível. O uso de punições também é feito em (EGLESE, 1994) e mais recentemente em (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a, 2014b, 2016), quando alguma demanda não é atendida a tempo, a punição começa a interferir na função objetivo, piorando-a.

A variável f_{ikp} permite que o carro folgue, ou seja, ficar parado em algum ponto em determinado dia, pois, algumas vezes é possível obter uma solução com menos deslocamentos. Quando essa variável assume valor 1, significa que as periodicidades podem ser atendidas em menor tempo, ou seja, o horizonte de tempo pode diminuir.

A função objetivo é dada pela equação (1):

$$\min Z = \sum_{[i,j] \in E} \sum_{k=1}^{nk} \sum_{p=1}^{np} c_{ij} * x_{ijkp} + \sum_{[i,j] \in E} \sum_{p=1}^{np} PU_{ij} * pn_{ijp} \quad (1)$$

Sujeito às seguintes restrições:

$$\sum_{[i,j] \in E} x_{ijkp} + f_{jkp} = \sum_{[i,j] \in E} x_{jik,p+1} + f_{jk,p+1} \quad \forall j \in X, \forall k \in K, p = 1, \dots, np - 1 \quad (2)$$

$$\sum_{[i,j] \in E} x_{ijk1} + f_{ik1} = y_i \quad \forall i \quad (3)$$

$$\sum_{[i,j] \in E} x_{ijk,np} + f_{jk,np} = y_j \quad \forall j \quad (4)$$

$$\sum_{[i,j] \in E} x_{ijkp} + \sum_{[i,j] \in E} x_{jikp} + \sum_{i=1}^n f_{ikp} = 1 \quad \forall p \in H, \forall k \in K \quad (5)$$

$$\sum_{k=1}^{nk} \left(x_{ijkp} + x_{jikp} + x_{ijk,p+1} + x_{jik,p+1} + \dots + x_{ijk,p+MPIj-1} + x_{jik,p+MPIj-1} \right) + p_{ijp} \geq 1 \quad (6)$$

$$\forall [i, j] \in S, \forall p \in H$$

$$\sum_{k=1}^{nk} \sum_{p=1}^{np} (x_{ijkp} + x_{jikp}) \geq 1 \quad \forall [i, j] \in R \quad (7)$$

$$x_{ijkp}, p_{ijp}, f_{ikp} \in \{0,1\} \quad \forall [i, j] \in E, \forall k \in K, \forall p \in H$$

$$y_i \in N^+ \quad (8)$$

A restrição (2) garante o fluxo diário dos carros permitindo a folga. As equações (3) e (4) garantem que a quantidade de carros que sai de um ponto no dia 1, será igual a quantidade de carros que voltará nesse ponto no final do horizonte de tempo. A equação (5) garante que a cada dia cada carro terá uma tarefa, seja ela deslocamento ou ficar parado. A restrição (6) é a inequação que visa garantir os atendimentos de todos os arcos com periodicidade menor que o tamanho do horizonte de tempo, se algum desses arcos for atrasado ocorrerá uma punição na função objetivo. A ideia de conversão dos períodos acima de np continua a valer para a equação

(6), o período $np + 1$ se torna o período 1, $np + 2$ o período 2 e assim por diante. Essa conversão pode ser vista como a função resto (*mod*) onde os períodos acima de np se tornam o resto da divisão por np . A restrição (7) garante que algum atendimento será feito em todos os arcos com periodicidade igual ao horizonte de tempo. Um atraso em algum desses arcos não pode ser permitido, pois significaria que eles nunca seriam atendidos. Em (8) as variáveis são dimensionadas.

O modelo foi validado na aplicação em 10 pequenos problemas cujas soluções eram possíveis de serem encontradas de maneira intuitiva; ou ainda, partindo de uma solução prévia, o problema era criado. Essas 10 instâncias receberam o nome de *gvb* e seus respectivos resultados foram os listados na Tabela 1. Outras 23 instâncias foram criadas a partir de (GOLDEN; DEARMON; BAKER, 1983) e seus resultados são encontrados na Tabela 2.

Tabela 1 - Relação dos 10 problemas gerados, propriedades e resultados.

Problema	Nós	Arestas	Period.	Carros	Neces.	Dias	Solução	GAP	Atrasos	Folgas	Tempo
<i>gvb1</i>	16	24	2	9	36	4	36	0,00%	0	0	0,68
<i>gvb2</i>	16	24	3	3	36	12	36	0,00%	0	0	278,27
<i>gvb3</i>	16	24	3	6	36	6	36	0,00%	0	0	1093,85
<i>gvb4</i>	16	24	2	4	32	8	32	0,00%	0	0	59682
<i>gvb5</i>	6	7	2	3	18	6	18	0,00%	0	0	0,03
<i>gvb6</i>	6	7	2	1	8	8	8	0,00%	0	0	0,1
<i>gvb7</i>	12	24	2	7	42	6	42	0,00%	0	0	1,07
<i>gvb8</i>	8	12	2	2	16	8	16	0,00%	0	0	0,02
<i>gvb9</i>	14	23	3	10	60	6	60	0,00%	0	0	8,02
<i>gvb10</i>	11	12	2	1	16	24	24	0,00%	0	0	407,03

Tabela 2 - Resultados para 23 instâncias adaptadas de Golden, Dearmon e Baker (1983)

Problema	Nós	Arestas	Period.	Carros	Neces.	Dias	Solução	GAP	Atrasos	Folgas	Tempo
<i>pgdbcm1</i>	12	22	2	1	32	38	451	0,00%	4	0	39000
<i>pgdbcm2</i>	12	26	3	1	44	53	592	12,33%	11	1	70423
<i>pgdbcm3</i>	12	22	2	2	31	19	340	7,35%	0	3	36906
<i>pgdbcm4</i>	11	19	3	2	34	21	2519	82,14%	3	1	49595
<i>pgdbcm5</i>	13	26	4	2	53	32	17733	96,28%	1	1	86400
<i>pgdbcm6</i>	12	22	2	3	34	14	456	8,33%	0	3	86400
<i>pgdbcm7</i>	12	22	3	3	37	45	508	2,95%	0	5	55767
<i>pgdbcm8</i>	27	46	4	3	67	27	19363	98,32%	12	0	86400
<i>pgdbcm9</i>	27	51	3	4	78	24	24465	98,21%	13	2	86400
<i>pgdbcm10</i>	12	25	4	4	70	21	745	8,19%	0	6	86400
<i>pgdbcm11</i>	22	45	2	1	50	60	459	10,46%	64	2	31510
<i>pgdbcm12</i>	13	23	3	1	33	40	527	8,16%	0	0	86400
<i>pgdbcm13</i>	10	28	2	2	33	20	620	1,45%	0	5	86400
<i>pgdbcm14</i>	7	21	3	2	32	20	145	0,00%	0	7	4,21
<i>pgdbcm15</i>	7	21	4	2	48	29	140	2,14%	0	7	63135
<i>pgdbcm16</i>	8	28	2	3	40	24	182	3,30%	0	28	35301
<i>pgdbcm17</i>	8	28	3	3	52	21	168	4,17%	0	4	77693
<i>pgdbcm18</i>	9	36	4	3	81	33	385	4,16%	0	6	86400
<i>pgdbcm19</i>	8	11	3	4	29	12	125	3,20%	0	17	47566
<i>pgdbcm20</i>	11	22	4	4	53	16	279	12,54%	2	1	67903
<i>pgdbcm21</i>	11	33	2	5	46	12	212	3,77%	0	10	38181
<i>pgdbcm22</i>	11	44	4	5	88	22	426	0,70%	0	17	86400
<i>pgdbcm23</i>	11	55	2	6	82	17	336	2,08%	0	15	86400

As Tabelas 1 e 2 trazem um resumo das características de cada instância e seus resultados. A primeira coluna apresenta o nome das instâncias, a segunda a quantidade de nós, a terceira a quantidade de arestas, a quarta a quantidade de diferentes periodicidades, a quinta a quantidade

de carros, a sexta as necessidades (soma total da quantidade mínima de atendimentos que devem ser feitos), a sétima o horizonte de tempo, a oitava a solução, a nona o GAP, a décima a quantidade de atrasos na solução, a décima primeira a quantidade de folgas em cada solução e a décima segunda o tempo de resolução em segundos. Os problemas foram resolvidos pelo software Gurobi 6.5 em um computador Intel Core i7 com 2 Ghz de processamento e 4Gb de memória RAM e sistema operacional 64 bits.

4. Metodologia e resultados

O problema proposto tem características bem peculiares no qual uma rota deve ter início e fim bem definidos, sendo que entre esse início e fim deve ser feito uma quantidade específica de deslocamentos, igual ao tamanho do horizonte de tempo. Vale lembrar que podem existir folgas dentro desses deslocamentos. Muitos dos algoritmos existentes na literatura para geração de rotas acabam por não contemplar a formação de subciclos, contudo esse caso a contempla, de modo que foi necessário pensar em uma abordagem rápida e eficaz para gerar novas soluções.

A Programação Dinâmica (PD) quando bem utilizada resolve problemas de forma exata. Para sua execução é necessário que o problema tenha estados e estágios muito bem definidos. Cada estágio acontece sempre que é preciso tomar uma decisão, os estados dizem respeito a como está o andamento do problema dependendo da decisão tomada. A PD pode ser aplicada a problemas nos quais é necessário encontrar a melhor ordem de performance dado um conjunto de operações, por exemplo, em problemas de sequenciamento e no problema do caixeiro viajante (BELLMAN, 1962; HELD; KARP, 1962). Esses problemas devem ter estruturas que possam trabalhar com contas recursivas.

No algoritmo, um problema complexo deve ser decomposto em estruturas mais simples, sendo que seu esforço computacional cresce quanto mais estágios houver. O processo busca o ótimo em cada estágio, sendo mais barato do que enumerar todas as possibilidades de um problema combinatório (SAKAGUTI, 2007). A programação dinâmica é uma ferramenta promissora e atraente para utilizar no problema, pois ao se olhar sua estrutura completa e montada pode-se perceber que o caminho ótimo deve estar em alguma combinação de ligação entre os estágios. Além disso, em suas ligações estão apenas os possíveis deslocamentos e folgas.

Para gerar pedaços de soluções iniciais, um modelo simples de programação dinâmica foi concebido. Como o atraso é indesejado e interfere muito no problema, a seguinte proposição foi feita: sempre que um carro for se deslocar, ele deve avaliar a condição de todos os demais arcos e fazer o movimento que gerará menos atrasos nas periodicidades. Os cálculos feitos são muito simples, porém não modela o problema real, apenas evidencia movimentos que tenham certa lógica. A Figura 1 exemplifica um problema.

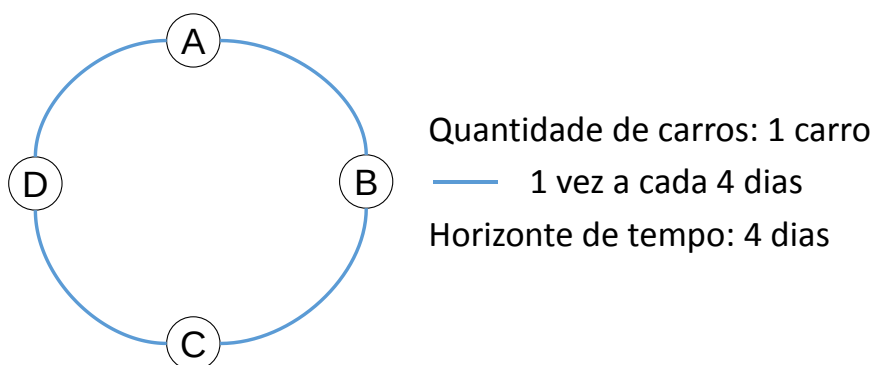


Figura 1 - Exemplo para aplicação da programação dinâmica

Foi definido um total de 5 estágios para o exemplo da Figura 1, começando em 0 e terminando no último dia do horizonte de tempo. O carro disponível deve sair de qualquer ponto e retornar a esse mesmo ponto no final do período de planejamento. Ao sair de um estágio e ir para o outro, são avaliadas todas as decisões de sair de um ponto e ir até outro. Após isso, é feita uma avaliação de qual movimento causa menor impacto nas periodicidades por meio de um valor T, mostrando a melhor decisão a ser tomada e gerando o estado para cada ponto durante os estágios. O valor T vai sendo acumulado a cada estágio e ao final é só necessário recuperar a rota. A Figura 2 é uma representação de como foi feita a programação dinâmica para Figura 1.

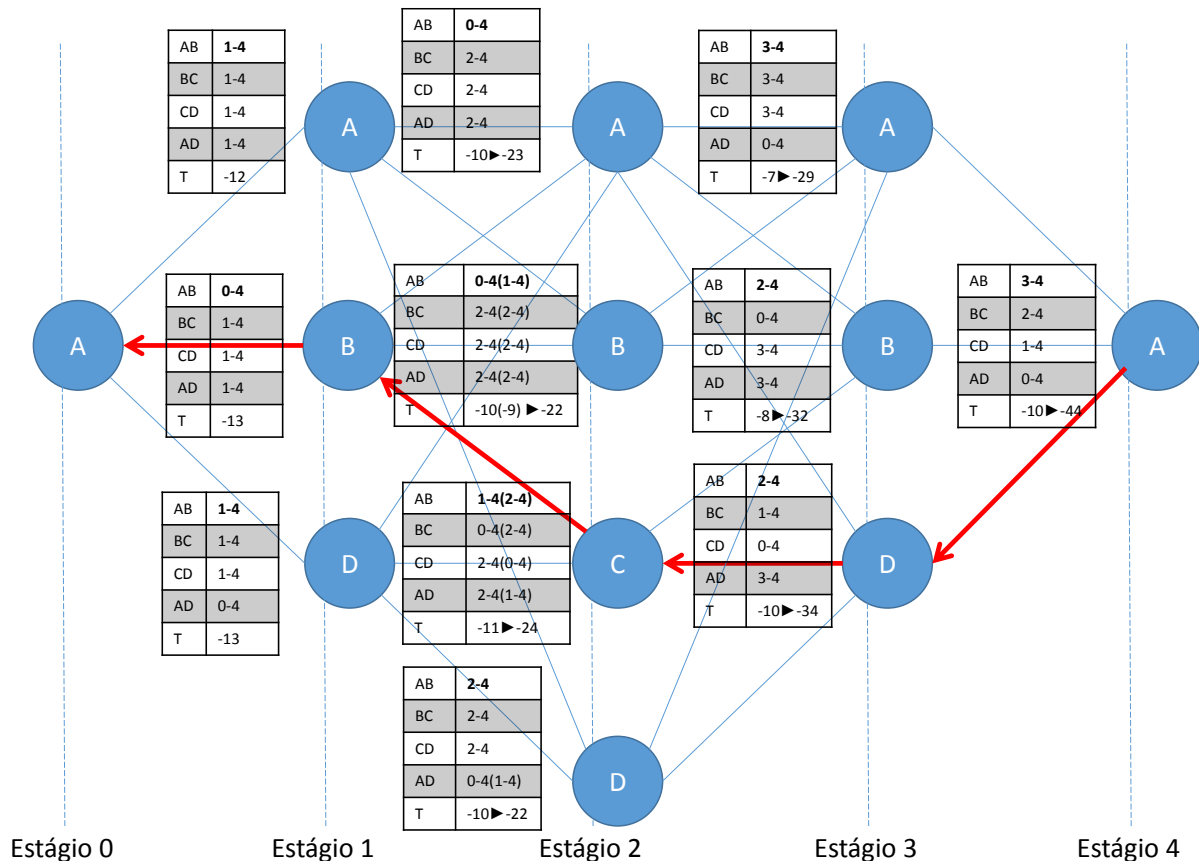


Figura 2 - Representação da Programação Dinâmica para o exemplo

É possível ver que a melhor rota é dada por A-B-C-D-A, todas as periodicidades são satisfeitas e nenhum atraso ocorre. Porém, como já citado, essa abordagem não modela o problema real, pois leva em conta apenas atrasos e não os deslocamentos. Além disso, não consegue ter uma visão geral do problema, pois uma decisão depois de tomada não pode ser recalculada. O futuro imediato é o mais importante, se algo valer a pena, será feito na ocasião sem pensar mais à frente. Quando aplicado o método a problemas maiores é possível ver que cada ponto escolhido como inicial pode gerar uma rota diferente, arcos podem até mesmo deixar de ser atendidos se não forem cobertos no momento oportuno. Problemas cíclicos dificilmente são abordados através de algoritmos de programação dinâmica ou heurísticas sequenciais que calculam as soluções do PCARP diariamente (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002b).

4.1 Utilização dos trechos

Por meio do algoritmo previamente apresentado foram geradas todas as possíveis soluções para cada problema. Dentro dessas soluções, foi avaliada qual a sequência de caminhos que mais aparece levando em conta uma quantidade de dias. A instância *pgdbcm1*, por exemplo,

apresentou 12 rotas como solução, uma para cada pondo de partida, relembrando que é uma instância com 12 nós, 22 arestas, 1 carro e 38 dias. Nem todas as rotas são factíveis, porém ela mostra trechos interessantes em suas soluções. As rotas obtidas pelo algoritmo proposto estão na **Erro! Fonte de referência não encontrada..**

Rota 1	1	7	8	10	9	2	4	1	2	3	5	6	7	12	1	10	11	5	3	4	1	7	1	2	9	2	4	3	2	9	11	5	12	1	10	8	11	10	1
Rota 2	2	4	1	2	3	5	6	7	1	12	7	8	11	10	9	2	4	3	2	1	10	1	4	3	5	12	1	7	12	6	5	11	9	2	4	1	2	3	2
Rota 3	3	4	2	1	12	7	1	10	9	2	3	5	11	8	7	6	5	12	1	4	3	4	2	1	7	1	10	9	2	3	5	11	10	8	7	12	1	4	3
Rota 4	4	2	1	4	3	5	6	7	8	11	5	12	7	1	10	9	2	3	4	2	1	12	1	4	3	5	11	10	8	7	6	7	1	10	9	2	3	2	4
Rota 5	5	6	7	12	1	7	8	10	9	2	4	1	2	3	5	11	10	1	4	3	5	6	12	1	7	8	11	9	2	4	1	2	3	5	12	1	10	11	5
Rota 6	6	5	12	7	1	2	3	5	11	9	2	4	1	12	6	7	8	10	1	4	3	2	1	7	12	5	3	4	2	9	10	1	12	6	5	11	8	7	6
Rota 7	7	12	1	4	2	3	5	12	6	7	1	10	8	11	9	2	1	4	3	5	12	1	4	2	3	5	11	10	1	7	8	10	9	2	1	12	7	6	7
Rota 8	8	11	10	1	4	2	3	5	11	9	2	1	12	5	6	7	1	4	3	2	9	10	1	2	4	3	5	11	8	7	12	1	7	1	4	2	9	10	8
Rota 9	9	2	4	1	2	3	5	6	7	1	12	7	8	10	9	11	10	1	4	3	4	2	9	2	1	2	3	5	12	1	7	6	5	11	10	1	4	2	9
Rota 10	10	1	7	8	10	9	2	4	1	2	3	5	12	6	7	12	1	10	11	5	3	4	1	7	1	2	9	2	4	3	2	9	11	5	12	1	10	8	10
Rota 11	11	9	2	1	4	2	3	5	6	7	1	12	5	11	8	10	1	4	3	2	9	2	1	4	2	3	5	12	1	7	8	11	9	10	1	4	3	5	11
Rota 12	12	7	1	12	6	7	8	10	9	2	4	1	2	3	5	11	10	1	4	3	5	12	1	7	8	11	9	2	4	1	2	3	5	6	12	1	10	1	12

Figura 3 - Resultado da Programação Dinâmica para a instância pgdbcm1.

A **Erro! Fonte de referência não encontrada.** dá ênfase em amarelo para a sequência de pontos que mais aparece quando buscadas os movimentos de 6 dias. A sequência 9-2-4-1-2-3-5 aparece 7 vezes na solução. Logo, ela pode ser um bom ponto de partida para o carro 1 no dia 1 que finalizara sua rota no final do sexto dia no ponto 5.

Tem-se a intuição de que essas sequencias que mais aparecem podem estar presentes em boas soluções do problema e eventualmente até mesmo na solução ótima. Então, aproveita-se dessa sequência de trechos para gerar o que chama-se espaço restrito (ER). Pécora Jr. (2008) desenvolveu um algoritmo que avalia iterativamente esses espaços restritos. O espaço restrito é um subconjunto do espaço de solução global, que geralmente é pequeno e rápido de ser explorado. A criação de um ER está vinculada a inserção de uma restrição, nesse caso o ER irá conter o trecho que mais aparece. Para isso, adicionamos uma restrição ao modelo obrigando um carro fazer a sequência de trechos que mais apareceu. No exemplo da Figura 3 adicionamos a restrição dada pela equação (9).

$$ER: -x_{9_2_1_1} - x_{2_4_1_2} - x_{4_1_1_3} - x_{1_2_1_4} - x_{2_3_1_5} - x_{3_5_1_6} = -6 \quad (9)$$

4.2 Resultados

Para avaliar o desempenho do algoritmo proposto, usou-se 4 espaços restritos para cada uma das 23 instâncias *pgdbcm*. Os espaços gerados compreendem os movimentos para 4, 5, 6 e 7 dias consecutivos. Cada problema teve rotas geradas por meio da PD considerando cada ponto como ponto inicial, e a partir delas foi feita a identificação dos movimentos que mais se repetiam. A busca dentro dos espaços restritos foi limitada a um tempo de processamento de 10 minutos utilizando o solver Gurobi 6.5 e a configuração do computador utilizado foi a mesma dos testes anteriores. O resultado da exploração dentro de cada espaço restrito está presente na Tabela 3.

Na Tabela 3, as melhores soluções conhecidas são as mesmas do Capítulo 3, ou seja, é o resultado de 24 horas de processamento para as 21 instâncias em que o valor ótimo não é conhecido. Percebe-se que em 10 das 23 instâncias foram encontrados resultados iguais aos melhores conhecidos, em outras 6 instâncias o resultado alcançado está a menos de 5,2% da melhor solução, nas outras 7 instâncias o melhor resultado conhecido não foi superado e o GAP permanece alto em virtude das punições com atrasos.

Tabela 3 - Resultados utilizando espaços restritos

Problema	Melhor solução conhecida	GAP	4 dias	5 dias	6 dias	7 dias	GAP Melhor solução
<i>pgdbcm1</i>	451	0,00%	2451	451	19391	3450	0,00%
<i>pgdbcm2</i>	592	12,33%	620	4590	26600	25576	4,52%
<i>pgdbcm3</i>	340	7,35%	1340	345	353	340	0,00%
<i>pgdbcm4</i>	2519	82,14%	5515	5532	7578	6527	54,32%
<i>pgdbcm5</i>	17733	96,28%	30719	24775	45723	49711	28,42%
<i>pgdbcm6</i>	456	8,33%	456	456	456	456	0,00%
<i>pgdbcm7</i>	508	2,95%	530	571	565	542	4,15%
<i>pgdbcm8</i>	19363	98,32%	46377	73372	67370	58390	58,25%
<i>pgdbcm9</i>	24465	98,21%	101426	113470	111456	147449	75,88%
<i>pgdbcm10</i>	745	8,19%	6769	7754	785	15746	5,10%
<i>pgdbcm11</i>	459	10,46%	20474	3469	75455	36506	86,77%
<i>pgdbcm12</i>	527	8,16%	536	13535	1503	2521	1,68%
<i>pgdbcm13</i>	620	1,45%	620	620	620	620	0,00%
<i>pgdbcm14</i>	145	0,00%	145	145	145	145	0,00%
<i>pgdbcm15</i>	140	2,14%	142	140	140	140	0,00%
<i>pgdbcm16</i>	182	3,30%	182	182	182	182	0,00%
<i>pgdbcm17</i>	168	4,17%	170	168	168	168	0,00%
<i>pgdbcm18</i>	385	4,16%	5434	16414	5420	13415	92,90%
<i>pgdbcm19</i>	125	3,20%	125	125	125	125	0,00%
<i>pgdbcm20</i>	279	12,54%	3290	5272	2264	4275	87,68%
<i>pgdbcm21</i>	212	3,77%	212	212	212	212	0,00%
<i>pgdbcm22</i>	426	0,70%	3495	443	3495	457	3,84%
<i>pgdbcm23</i>	336	2,08%	350	341	340	337	0,30%

5. Considerações finais

A busca em espaços restritos se mostra uma boa alternativa para a resolução de um problema como o PCARP. Os resultados, logo após 10 minutos de processamento, chegam a valores iguais a melhor solução conhecida até então. No entanto, a qualidade dos movimentos usados para a limitação dos espaços é algo importante a ser considerado. A Programação Dinâmica proposta, apesar de não se importar com a factibilidade das rotas, é uma ferramenta rápida que gera movimentos inteligentes, e por meio dela foi possível a identificação dos padrões de movimentos que culminaram nos bons resultados. O método final se preocupa apenas com a movimentação de um carro e a geração de um espaço restrito, ele ainda pode ser aperfeiçoado gerando rotas por outro método e a explorando iterativamente mais espaços restritos. O PCARP é um problema recente e em alta, visto a quantidade de aplicações que oferece. O desenvolvimento de novas técnicas para a resolução do problema está em aberto e deve ser mais explorado, pois os ganhos relativos a suas aplicações, como na inspeção de trilhos, são objeto de interesse de grandes empresas.

Referências

- BATISTA, G. V.** PROPOSTA DE UM MODELO MATEMÁTICO PARA O PROBLEMA DE ROTEAMENTO EM ARCOS CAPACITADO E PERIÓDICO, 2014. Universidade Federal do Paraná.
- BATISTA, G. V.; SCARPIN, C. T.** PROBLEMA DE ROTEAMENTO EM ARCOS CAPACITADO E PERIÓDICO APLICADO A UM CONTEXTO REAL. Revista Produção Online, v. 15, n. 3, p. 1080–1098, 2015.

BELLMAN, R. *Dynamic Programming Treatment of the Travelling Salesman Problem*. J. ACM, v. 9, n. 1, p. 61–63, 1962. New York, NY, USA: ACM. Disponível em: <<http://doi.acm.org/10.1145/321105.321111>>. .

CHRISTOFIDES, N.; BEASLEY, J. E. *The Period Routing Problem*. Networks, v. 14, p. 237–256, 1984.

CHU, F.; LABADI, N.; PRINS, C. *A Scatter Search for the periodic capacitated arc routing problem*. European Journal of Operational Research, v. 169, p. 586–605, 2006.

CORBERÁN, A.; PRINS, C. *Recent Results on Arc Routing Problems : An Annotated Bibliography*. Networks, p. 50–69, 2010.

EGLESE, R. W. *Routeing winter gritting vehicles*. Discrete Applied Mathematics, v. 48, n. 3, p. 231–244, 1994. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/0166218X92000035>>. .

EISELT, H. A.; GENDREAU, M.; LAPORTE, G. *Arc Routing Problems, Part I: The Chinese Postman Problem*. Operations Research, v. 43, n. 3, p. 231–242, 1995. Disponível em: <<http://or.journal.informs.org/cgi/doi/10.1287/opre.43.2.231>>. .

GHIANI, G.; MUSMANNO, R.; PALETTA, G.; TRIKI, C. *A heuristic for the periodic rural postman problem*. Computers & Operations Research, v. 32, n. 2, p. 219–228, 2005. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0305054803002132>>. Acesso em: 28/8/2013.

GOLDEN, B. L.; DEARMON, J. S.; BAKER, E. K. *Computational experiments with algorithms for a class of routing problems*. Computers & Operations Research, v. 10, n. 1, p. 47–59, 1983. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0305054883900266>>. Acesso em: 13/3/2014.

GOLDEN, B. L.; WONG, R. T. *Capacitated Arc Routing Problem*. Networks, v. 11, p. 305–311, 1981.

HELD, M.; KARP, R. M. *A dynamic programming approach to sequencing problems*. J. Soc. Indust. Appl. Math., v. 10, n. 1, p. 196–210, 1962.

HUANG, S.; LIN, T. *Using Ant Colony Optimization to solve Periodic Arc Routing Problem with Refill Points*. , n. January 2015, p. 37–41, 2014.

KANSOU, A.; YASSINE, A. *Ant Colony System for the Periodic Capacitated Arc Routing Problem Problem*. International Network Optimization Conference. Anais... . p.1–7, 2009.

KONOWALENKO, F. *PROBLEMA DO CARTEIRO CHINÊS NÃO-ORIENTADO E MISTO PARA A OTIMIZAÇÃO DE ROTAS NA CIDADE DE IRATI / PR*, 2012. Universidade Federal do Paraná.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. *Evolutionary Algorithms for Multiperiod Arc Routing Problems*. In: IPMU 2002 (Ed.); 9th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based systems. Anais... . p.1–8, 2002a. Annecy, France: ESIA-University of Savoie.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. *Planning Problems in Arc Routing*. In: F. U.-E. de Valencia (Ed.); Actes de PMS 2002 (EURO Working Group on Project Management and Scheduling). Anais... . p.232–235, 2002b. Valencia, Espanha.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. *Evolutionary algorithms for periodic arc routing problems*. European Journal of Operational Research, v. 165, p. 535–553, 2005.

MARZOLF, F.; TRÉPANIER, M.; LANGEVIN, A. *Road network monitoring: algorithms and a case study*. Computers & Operations Research, v. 33, n. 12, p. 3494–3507, 2006. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0305054805000717>>. Acesso em: 20/1/2014.

MEI, Y.; MEMBER, S.; TANG, K.; YAO, X. *A Memetic Algorithm for Periodic Capacitated Arc Routing Problem*. , v. 41, n. 6, p. 1654–1667, 2011.

MONROY, I. M.; AMAYA, C. A.; LANGEVIN, A. *The periodic capacitated arc routing problem with irregular services*. Discrete Applied Mathematics, v. 161, p. 691–701, 2013.

PÉCORA JR, J. E. *Iterative Restricted Space Search: a solving approach based on hybridization*, 2008. Université Laval.

RIQUELME-RODRÍGUEZ, J.-P.; GAMACHE, M.; LANGEVIN, A. *Periodic capacitated arc-routing problem with inventory constraints*. Journal of the Operational Research Society, v. 65, n. 12, p. 1840–1852, 2014a. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84925938293&partnerID=tZ0tx3y1>>. .

RIQUELME-RODRÍGUEZ, J.-P.; GAMACHE, M.; LANGEVIN, A. *Adaptive Large Neighborhood Search for the Periodic Capacitated Arc Routing Problem with Inventory Constraints*. Networks, v. 47, n. 1, p. 26–36, 2014b.

RIQUELME-RODRÍGUEZ, J.-P.; GAMACHE, M.; LANGEVIN, A. *Location arc routing problem with inventory constraints*. Computers & Operations Research, v. 76, p. 84–94, 2016.

SAKAGUTI, Y. F. *Otimização do processo de picking de um centro de distribuição através da programação dinâmica*, 2007. Universidade Federal do Paraná.

TAUBE, J. DE M. *O problema do Carteiro Chinês*, 1992.

ZHANG, Y.; MEI, Y.; TANG, K.; JIANG, K. *Memetic algorithm with route decomposing for periodic capacitated arc routing problem*. Applied Soft Computing Journal, p. 17–21, 2016. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.asoc.2016.09.017>>.