

Desenvolvimento de um algoritmo baseado em Hill-Climbing para o problema de roteamento periódico de veículos

Rodrigo Faria Dayrell (UTFPR-LD) rodrigofariadayrell@gmail.com

Rafael Henrique Palma Lima (UTFPR-LD) rafaelhlma@utfpr.edu.br

Resumo:

O objetivo deste trabalho é descrever o desenvolvimento um algoritmo para solucionar problemas de roteamento periódico de veículos (PVRP, do inglês, Periodic Vehicle Routing Problem). O algoritmo proposto busca encontrar soluções próximas ou iguais às da literatura para o problema em um curto espaço de tempo. O problema tratado é de alta complexidade e busca definir rotas que atendam às demandas dos clientes que podem acontecer ao longo de um horizonte de tempo, de maneira periódica, respeitando a restrição de capacidade do veículo. O método de resolução proposto pode ser dividido em três fases. Na primeira fase são designados os períodos de cada ponto aleatoriamente, procurando equilibrar a carga entre os períodos. A segunda fase consiste em designar os pontos aos veículos de cada período de forma factível utilizando o algoritmo do vizinho mais próximo como regra de agrupamento. Na terceira fase o algoritmo 2-opt é utilizado em cada rota de cada veículo. Os resultados se mostraram próximos aos da literatura.

Palavras chave: Problema de Roteamento Periódico de Veículos, Hill-climbing, Vizinho Mais Próximo, Algoritmo 2-opt.

Development of a algorithm based in Hill-Climbing for the periodic vehicle routing problem

Abstract

The aim of this paper is to describe the development of an algorithm to solve periodic vehicle routing problem (PVRP). The proposed one intends to find solution close to or suchlike to those in the literature to the problem, in a short space time. The problem dealt with is highly complex and try to define routes that reach customer demands that can occur over a period of time, in a periodic way, respecting the capacity of the vehicle. The proposed resolution method can be divided into three phases. In the first phase the periods of each point are randomly assigned, trying to balance the load between the periods. The second phase consists of assigning the points to the vehicles of each period in a feasible way using the algorithm of the nearest neighbor as a grouping rules. In the third phase the 2-opt algorithm is used in each route of each vehicle. The results were close to those found on the literature.

Key-words: Periodic Vehicle Routing Problem, Hill-climbing, Nearest Neighbor, 2-opt Algorithm.

1. Introdução

O transporte é responsável por uma parte considerável do custo total do produto. Com o objetivo de reduzir os custos de transporte, heurísticas têm sido criadas para minimizar as

distâncias totais das rotas dos veículos. Por essa razão, a literatura científica tem se dedicado a desenvolver abordagens metaheurísticas para encontrar boas soluções para problemas de roteamento, o que proporcionaria grandes reduções de custos logísticos para as organizações.

O problema de roteamento de veículos, o VRP, é sem dúvida um dos os problemas de otimização combinatória mais estudados na literatura. Pode ser definido como um problema de projetar rotas de entrega para veículos de determinadas capacidades, para conjunto de clientes com localizações e demandas conhecidas, com os veículos partindo de um único depósito. São projetadas rotas para os veículos com o objetivo de minimizar a distância total percorrida (PACHECO *et al.* 2012).

Este trabalho trata do Problema de Roteamento Periódico de Veículos (PVRP), no qual um período de planejamento de vários dias é considerado e os clientes devem ser visitados mais de uma vez, com periodicidade pré-definida. Diferentes clientes geralmente exigem números diferentes de visitas em um determinado horizonte de tempo. Clientes com maiores demandas ou menores capacidades de armazenamento requerem mais visitas do que clientes com demandas menores ou maiores capacidades de armazenamento (HEMMELMAYR *et al.* 2007).

Nesse contexto, o objetivo deste trabalho é propor uma heurística que combina o Hill Climbing e o algoritmo 2-opt para encontrar soluções factíveis em um curto espaço de tempo. Assim, o algoritmo 2-opt serve como mecanismo de exploração da vizinhança até que não seja mais possível melhorar a solução atual. O procedimento proposto foi repetido por uma quantidade determinada de iterações, sendo registrada a melhor resposta obtida.

O restante do artigo está organizado da seguinte maneira. A Seção 2 descreve o problema de roteamento periódico de veículos e a Seção 3 apresenta o algoritmo proposto para o problema. A Seção 4 descreve as instâncias de benchmark usadas para testar o algoritmo e a Seção 5 discute os resultados numéricos obtidos. Finalmente, a Seção 6 faz as considerações finais do artigo.

2. Problema de Roteamento Periódico de Veículos

No problema de roteamento de veículos, os clientes procuram serviços em múltiplos dias durante um período de tempo. Os clientes precisam primeiramente ser designados a sequências de dias de serviço e então um VRP é resolvido para cada dia do período para todos os clientes planejados para aquele dia. O objetivo é minimizar a distância total viajada pela frota de cada período.

O PVRP recebe muita atenção na literatura. Drummond, Ochi e Vianna (2001) utilizaram busca local e um algoritmo genético paralelo para a resolução de um PVRP. Alegre, Laguna e Pacheco (2007) desenvolveram um algoritmo de busca dispersa para instâncias com períodos longos. Coene, Arnout e Spieksma (2010) implementaram um algoritmo com uma fase de programação e outra de roteamento PVRP. Hemmelmayr *et al.* (2007) desenvolveram um algoritmo de busca variada de vizinhança.

Existem diversas variações do PVRP. Vidović, Popović e Ratković (2014) propuseram uma heurística de pesquisa variada de vizinhança para solucionar um problema de roteamento de inventário de multi produtos e multi períodos na entrega de combustíveis com veículos homogêneos e multi compartimento. Avella, Boccia e Sforza (2004) e Cornillier *et al.* (2008) estudam o mesmo caso propondo algoritmos exatos e heurísticos. Angelelli e Speranza (2002)

Propuseram uma busca tabu para resolver uma generalização do PVRP na qual os veículos podem renovar suas capacidades com facilidades intermediárias. Vidal *et al.* (2012) propuseram um algoritmo genético híbrido para resolver um PVRP e um PVRP com multi depósitos com veículos de capacidade heterogênea e duração de rota imposta. Nguyen, Crainic e Toulouse (2014) propuseram um algoritmo genético para um PVRP com janelas de tempo.

3. Proposta do algoritmo

O problema de roteamento periódico de veículos é de alta complexidade. Para cada combinação de períodos existe uma série de possíveis designações de pontos para os veículos e cada veículo possui uma série de possíveis roteamentos. Isso faz com que o problema tenha um espaço de soluções muito grande, sendo a maioria das soluções de baixa qualidade ou infactíveis.

Como existe um grande espaço a ser explorado, foram escolhidos métodos e heurísticas que proporcionassem ao algoritmo um baixo custo computacional, para tentar explorar uma maior quantidade de soluções. Foram consideradas também heurísticas que possibilitassem uma exploração de uma região mais promissora do espaço de soluções.

Foram geradas soluções aleatórias factíveis na qual as restrições de periodicidade e de capacidade dos veículos sejam atendidas. Em seguida as soluções foram melhoradas através de uma busca local até que as soluções não pudessem mais ser otimizadas.

3.1. Geração da solução inicial

Para a obtenção da solução inicial foi gerado aleatoriamente uma série de combinações P , onde P_n é o código que informa em quais períodos o ponto n será visitado. Em seguida, é gerada uma série V , onde V_n é o veículo pelo qual o ponto n será visitado, ou seja, inicialmente todos os pontos serão sempre visitados pelo mesmo veículo em qualquer período. Nessa primeira solução somente as restrições de periodicidade são respeitadas, a capacidade do veículo é desconsiderada. O procedimento é ilustrado na Figura 1.

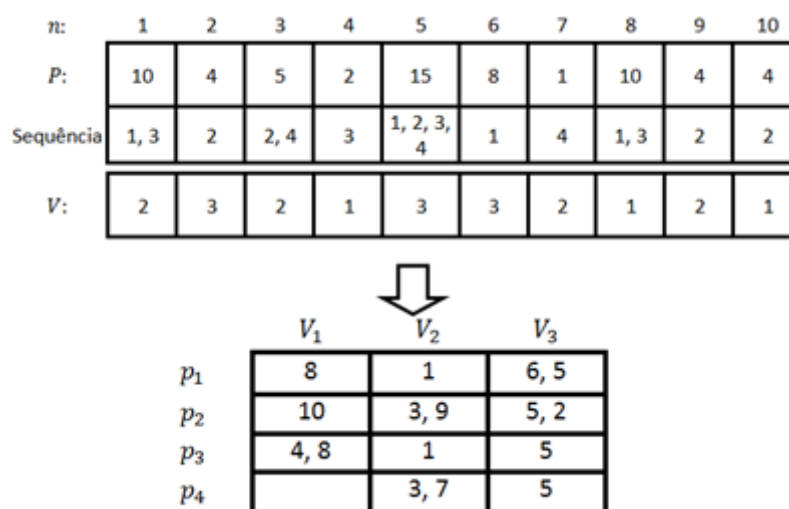


Figura 1 – Designação dos pontos

Para tornar a solução factível, uma sequência de trocas deve ser realizada. Essa sequência pode ser dividida em três etapas. Na primeira etapa são realizadas k_1 trocas de pontos entre as rotas $R_{p,v}$ que serão percorridas no período p pelo veículo v . Para decidir quais rotas serão trocadas é realizada uma verificação de quais rotas possuem carga $Q_{R_{p,v}}$ acima da capacidade C e a primeira rota $R_{p,v}$ detectada será escolhida para perder um ponto aleatório para a rota de menor carga $Q_{R_{p,u}}$ localizada no mesmo período p . Na segunda etapa serão realizadas k_2 trocas de pontos entre períodos. Será novamente realizada uma verificação de quais veículos possuem carga excedida e o primeiro detectado terá um ponto n sorteado para trocar seu código P_n e se manter no mesmo veículo V_n . A terceira etapa consiste em repetir a primeira etapa k_3 vezes. Se mesmo após esse procedimento a solução continuar infactível, a solução é descartada e uma nova solução inicial é gerada e os processos anteriores são repetidos até que se consiga uma solução factível. Os valores utilizados para k_1 , k_2 e k_3 são iguais a 20.

A solução factível obtida pelos procedimentos anteriores possui qualidade muito baixa. É necessário fazer uma nova designação dos pontos para as rotas. O algoritmo do vizinho mais próximo é utilizado como uma espécie de guia na designação dos pontos n para cada rota $R_{p,v}$. Primeiramente foram reunidos todos os pontos n de cada período p e colocados em uma série L , onde L_p é o conjunto de pontos no período p .

Em seguida foi aplicado o algoritmo do vizinho mais próximo entre os pontos de cada conjunto L_p . Após o ordenamento dos pontos, cada sequência L_p deve ser fragmentada em m sub-rotas, onde m é a quantidade máxima de veículos disponíveis em um período. As sub-rotas futuramente vão se transformar em rotas $R_{p,v}$ para cada um dos m veículos v .

Para fragmentação das sequências L_p será distribuído para as rotas $R_{p,1}$ até $R_{p,m-1}$ um valor de pontos igual ao quociente da quantidade de pontos contidos na sequência L_p pela quantidade m de veículos. Os pontos que sobrem ficarão na rota $R_{p,m}$. Em seguida, as extremidades de cada sub-rota vão se ligar ao depósito, que é identificado como o ponto 0. A fragmentação é ilustrada na Figura 2.

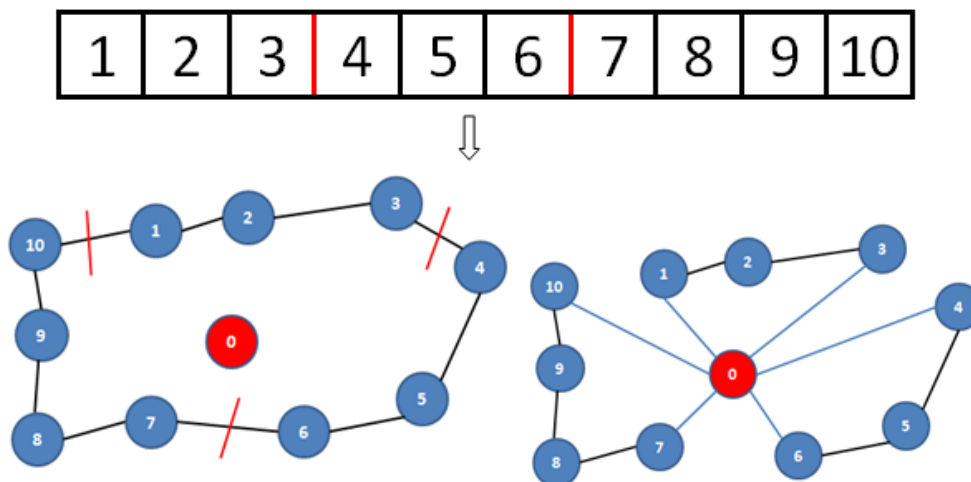


Figura 2 – Fragmentação das rotas

Após realizar essa designação K_4 rotas $R_{p,v}$ são sorteadas para deslocar seu último ou o primeiro ponto para uma das rotas vizinhas $R_{p,v+1}$ ou $R_{p,v-1}$. O valor utilizado para K_4 é igual a 10.

3.2. Busca local

As rotas construídas no procedimento anterior ainda não estão devidamente otimizadas. Para melhorá-las, o algoritmo 2-opt é aplicado em cada uma das rotas ao longo do horizonte de planejamento. O algoritmo 2-opt gera todas as soluções possíveis retirando-se 2 arcos e conectando novamente os pontos. Dessa maneira, pode-se utilizar essa estrutura de vizinhança em um framework Hill-Climbing, gerando vizinhanças até que nenhuma nova melhoria possa ser obtida, chegando a um ótimo local. Quando um ótimo local é obtido, a iteração do algoritmo é finalizada.

4. Instâncias de benchmark para o PVRP

Em um PVRP os pontos podem variar sua designação entre os períodos, porém devem respeitar as possíveis combinações determinadas pela instância. As combinações são informadas através de códigos nos quais cada número significa uma sequência de dias em que um veículo pode visitar um cliente. As instâncias informam a posição do ponto, demanda necessária e todas as combinações possíveis, como mostrado na Tabela 1.

Ponto	X	Y	Demanda	Frequência	Nºcomb.	Combinações				
0	30	40	0	0	0					
1	37	52	7	1	5	1	2	4	8	16
2	49	49	30	5	1	31				
3	52	64	16	2	3	5	10	20		
4	20	26	9	1	5	1	2	4	8	16
5	40	30	21	2	3	5	10	20		
6	21	47	15	2	3	5	10	20		
7	17	63	19	2	3	5	10	20		
8	31	62	23	2	3	5	10	20		
9	52	33	11	2	3	5	10	20		
10	51	21	5	1	5	1	2	4	8	16

Fonte: Cordeau, Gendreau e Laporte (1997)

Tabela 1 – Parte da instância p02 de Cordeau, Gendreau e Laporte (1997)

A Tabela 1 mostra uma parte da instância p02 como exemplo de uma das instâncias usadas no trabalho. A primeira coluna informa o número dos pontos, sendo o ponto 0 o depósito. A segunda e terceira coluna informa as coordenadas de cada ponto em um plano cartesiano. A quarta coluna informa a demanda de cada ponto. A quinta coluna se refere ao número períodos em que o ponto deve ser visitado. Da sexta coluna em diante aparecem os códigos disponíveis de combinação para cada ponto. Os códigos são números de 1 a $2^p - 1$ onde p é o número de períodos da instância. É possível definir qual combinação de períodos cada

código se refere através da conversão do número decimal informado para o número binário equivalente. A Figura 3 ilustra como funciona esse processo.

	1	2	3	4	5		
1	0	0	0	0	1	→	5
2	0	0	0	1	0	→	4
4	0	0	1	0	0	→	3
8	0	1	0	0	0	→	2
16	1	0	0	0	0	→	1
5	0	0	1	0	1	→	3, 5
10	0	1	0	1	0	→	2, 4
20	1	0	1	0	0	→	1, 3
31	1	1	1	1	1	→	1, 2, 3, 4, 5

Figura 3 – Leitura dos códigos de combinações

A Figura 3 mostra alguns exemplos de possíveis códigos para um horizonte de planejamento de cinco períodos. O número 20, por exemplo, pode ser convertido para números binários como 10100, o que significa que pontos que forem designados com o código 20 devem ser visitados nos períodos 1 e 3, pois o algarismo 1 está presente apenas na primeira e terceira casas do número binário equivalente.

5. Resultados

Para uma futura análise, foram utilizadas instâncias que simulam diferentes situações que tornam possível avaliar a flexibilidade do algoritmo proposto. Foi gerada uma tabela que informa a quantidade de iterações, o tempo real gasto o número de pontos, períodos e veículos em cada período, para que seja possível relacionar essas características com os resultados obtidos em comparação com as melhores respostas conhecidas. O algoritmo foi executado em 29 instâncias padrão encontradas na literatura. Foram executadas sucessivas iterações durante o período de uma hora para cada instância. O algoritmo foi testado em instâncias obtidas na literatura propostas por Cordeau, Gendreau e Laporte (1997). Os resultados da aplicação do algoritmo podem ser vistos na Tabela 2.

Instância	Iterações	Tempo(s)	n	m	p	$\frac{n}{m * p}$	Custo	Melhor Conhecida	%
p01	47346	3600,04	51	3	2	8,5	636,66	524,61	21,36
p02	42965	3600,05	50	3	5	3,3	1547,97	1322,87	17,02
p03	85266	3600,02	50	1	5	10,0	797,24	524,61	51,97
p04	41766	3600,02	75	5	2	7,5	959,13	835,43	14,81
p05	22988	3600,2	75	6	5	2,5	2477,64	2027,99	22,17
p06	651	3603,11	75	1	10	7,5	1549,72	836,37	85,29
p07	56535	3600,03	100	4	2	12,5	1005,17	826,14	21,67
p08	22260	3600,02	100	5	5	4,0	2580,96	2034,15	26,88
p09	40518	3600,04	100	1	8	12,5	1547,45	826,14	87,31
p10	40119	3600,08	100	4	5	5,0	2210,07	1595,84	38,49
p11	23008	3600,05	139	4	5	7,0	1052,39	779,29	35,05
p12	13285	3600,02	163	3	5	10,9	1621,01	1195,28	35,62
p13	1174	3606,28	417	9	7	6,6	5073,01	3511,62	44,46
p14	304473	3600,00	20	2	4	2,5	954,81	954,81	0,00
p15	125920	3600,02	38	2	4	4,8	1977,28	1862,63	6,16
p16	59933	3600,01	56	2	4	7,0	3195,42	2875,24	11,14
p17	125165	3600,01	40	4	4	2,5	1802,02	1597,75	12,78
p18	49750	3600,14	76	4	4	4,8	3598,60	3147,24	14,34
p19	23017	3600,38	112	4	4	7,0	5754,08	4834,34	19,03
p20	6846	3600,35	184	4	4	11,5	11014,58	8367,4	31,64
p21	61847	3600,12	60	6	4	2,5	2608,69	2184,04	19,44
p22	26049	3600,16	114	6	4	4,8	5134,13	4271,11	20,21
p23	11047	3600,04	168	6	4	7,0	7883,98	6602,59	19,41
p24	90226	3600,01	51	3	6	2,8	4503,66	3687,46	22,13
p25	94853	3600,02	51	3	6	2,8	4511,28	3777,15	19,44
p26	70356	3600,00	51	3	6	2,8	4505,70	3795,33	18,72
p27	24206	3600,09	102	6	6	2,8	27303,46	21956,46	24,35
p28	23387	3600,11	102	6	6	2,8	26998,98	22934,71	17,72
p29	24588	3600,02	102	6	6	8,5	26793,12	22909,36	16,95

Fonte: Autoria Própria

Tabela 2 – Resultados do PVRP comparados a Cordeau, Gendrea e Laporte (1997)

A Tabela 2 Mostra os resultados obtidos pelo algoritmo proposto. A primeira coluna indica o nome da instância. As segunda e terceira colunas informam, respectivamente, o número de iterações e o tempo real decorrido para realizá-las. As quarta, quinta e sexta colunas mostram o número de pontos, número de veículos, e número de períodos respectivamente. A sétima coluna mostra a quantidade média de pontos em cada sub-rotas. A oitava coluna mostra o melhor custo obtido pelo algoritmo proposto e a nona coluna mostra o melhor custo conhecido. A décima coluna informa o desvio entre o custo obtido pelo algoritmo proposto e o melhor conhecido.

Desvios acima de 40% foram considerados altos, desvios entre 30% e 40% foram

considerados regulares, desvios abaixo de 30% e acima de 15% foram considerados satisfatórios e desvios abaixo de 15% são considerados muito bons. A maioria das respostas foram satisfatórias porém foram obtidos apenas cinco desvios considerados muito bons e quatro desvios altos. Pode-se observar que a qualidade da resposta está relacionada com a quantidade de veículos por período, pois as piores soluções possuíam apenas 1 veículo por período. Outra relação relevante é a quantidade total de pontos. A instância com maior número de pontos foi classificada como tendo alto desvio. A média de pontos por sub-rotas também deve ser considerada, pois quanto menor essa relação, mais rápida é a busca local e, consequentemente, maior é a quantidade de iterações em um intervalo de tempo. O algoritmo foi construído com o objetivo de criar rotas com pontos bem agrupados e sem cruzamentos entre os arcos. A Figura 4 apresenta os cinco períodos obtidos na instância p02.

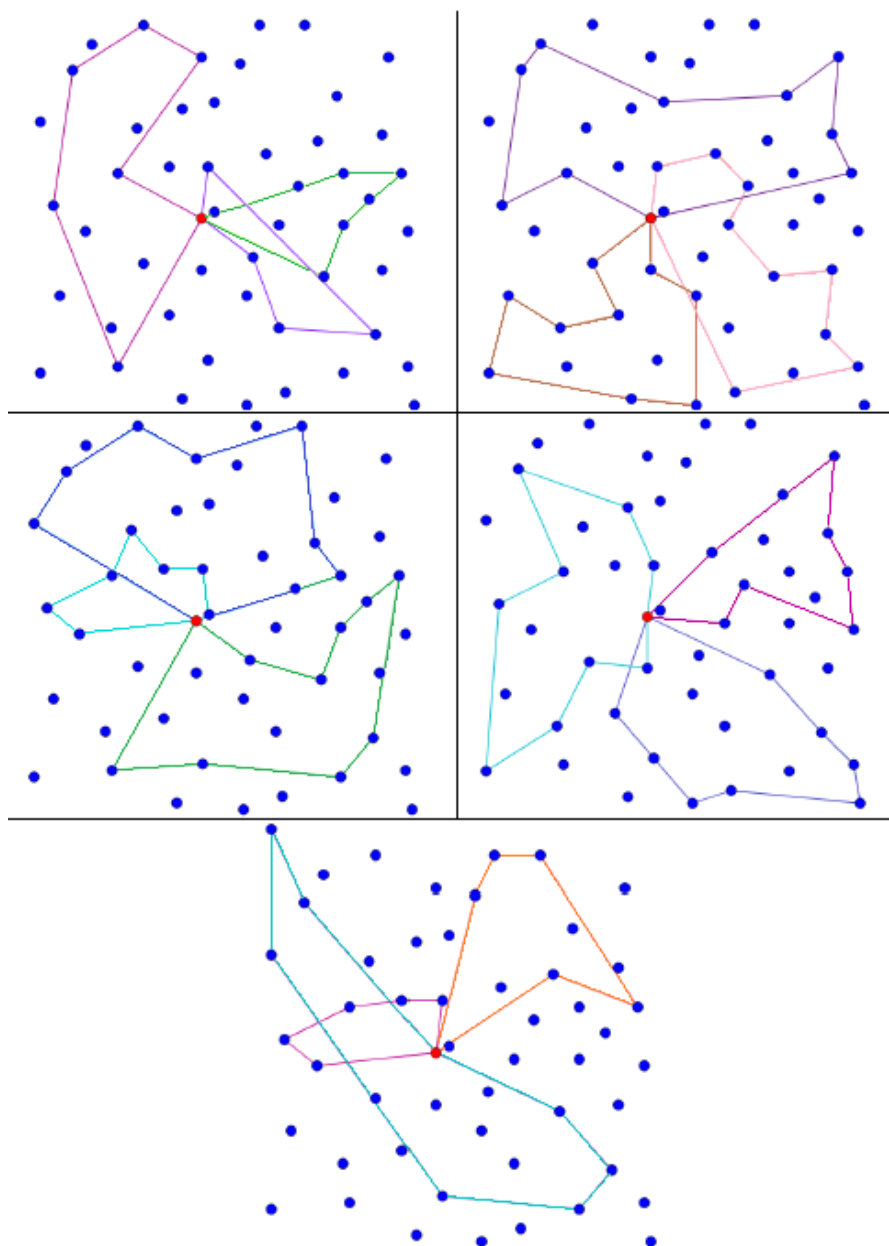


Figura 4 – Desenho das rotas obtidas na instância p02

A Figura 4 representa graficamente as rotas da melhor resposta obtida para a instância p02, que possui cinco períodos e três veículos em cada período. É possível observar que não há cruzamentos entre os arcos de uma mesma rota e o algoritmo construiu as rotas utilizando pontos próximos uns dos outros. Assim foi possível obter respostas satisfatórias quando comparadas com a literatura.

6. Conclusões

Este trabalho apresentou um algoritmo que combina Hill Climbing com o algoritmo 2-opt e o algoritmo do vizinho mais próximo para o Problema de Roteamento Periódico de Veículos (PVRP). O algoritmo se inicia como soluções totalmente aleatórias e em seguida o algoritmo do vizinho mais próximo foi usado como atalho na designação dos pontos para as rotas. As soluções aleatórias são melhoradas aplicando o algoritmo 2-opt até que nenhuma solução melhor possa ser obtida.

Com relação ao tempo de execução, o algoritmo pode ser considerado bem-sucedido, pois encontrou uma média de 15 soluções por segundo e é eficiente com problemas de poucos pontos por período. O algoritmo chegou a soluções factíveis e satisfatórias. Contudo, em apenas uma das instâncias foi encontrada solução igual a da literatura, o algoritmo não tem capacidade de realizar possíveis eliminações de sub-rotas e mostrou-se ineficiente quando aplicado em instâncias que necessitam de um grande número de pontos ou instâncias com apenas um veículo por período.

Algumas mudanças simples podem ser realizadas para resolver alguns dos problemas, como adicionar a possibilidade de outro tipo de solução inicial para o caso de o número de veículos por período ser igual a 1. Heurísticas que possibilitem eliminação de sub-rotas através de uma busca local mais eficiente podem ser implementadas para o melhoramento das respostas. O algoritmo pode ser adaptado para a resolução de problemas de roteamento periódico de veículos com restrições de tempo devido a sua eficiência para dividir as rotas de forma homogênea.

Referências

- HEMMELMAYR, VERA C.; DOERNER, KARL F.; HARTL, RICHARD F. *A variable neighborhood search heuristic for periodic routing problems*. European Journal of Operational Research. Vol. 195, p.791-802, 2007.
- PACHECO, J.; ALVAREZ, A.; GARCIA, I.; ANGEL-BELLO, F. *Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates*. Journal of the Operational Research Society. . Vol. 63, p.569-581, 2012.
- DRUMMOND, L.; OCHI, L.; VIANNA, D. *An asynchronous parallel metaheuristic for the period vehicle routing problem*. Future Generation Computer Systems. Vol. 17, n.4, p.379-386, 2001.
- ALEGRE, J.; LAGUNA, M.; PACHECO, J. *Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts*. European Journal of Operational Research. Vol. 179, n.3, p.736-746, 2007.
- COENE, S.; ARNOUT, A.; SPIEKSMAN, F. *On a periodic vehicle routing problem*. Journal of the Operational Research Society. Vol. 61, n.12, p.1719-1728, 2010.
- VIDOVIĆ, M.; POPOVIĆ, D.; RATKOVIĆ, B. *Mixed integer and heuristics model for the inventory routing problem in fuel delivery*. International Journal of Production Economics. Vol. 147, p.593-604, 2014.
- AVELLA, P.; BOCCIA, M.; SFORZA, A. *Solving a fuel delivery problem by heuristic and exact approaches*. European Journal of Operational Research. Vol. 152, p.170-179, 2004.



VIII CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO

Ponta Grossa, PR, Brasil, 05 a 07 de dezembro de 2018

CORNILLIER, F.; BOCTOR, F.F.; LAPORTE, G. RENAUD, J.; *An exact algorithm for the petrol station replenishment problem.* Journal of the Operational Research Society. Vol. 59, p.607-615, 2008.

ANGELELLI, E.; SPERANZA, M. *The period vehicle routing problem with intermediate facilities.* European Journal of Operational Research. Vol. 137, n.2, p.233–247, 2002.

VIDAL, T.; CRAINIC, T.G.; GENDREAU M. LAHRICHI, N.; REI, W.; *A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems.* Operations Research. Vol. 60, n.3, p.611-624, 2012.

NGUYEN, P.K.; CRAINIC, T.G.; TOULOUSE, M. *A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows.* Journal of Heuristics. Vol. 20, n.4, p.383-416, 2014.

CORDEAU, J.F., GENDREAU, M., LAPORTE, G. *A tabu search heuristic for periodic and multi-depot vehicle routing problems.* Networks. Vol. 30, p.105-119, 1997.