

A Relax-and-Fix Decomposition Strategy Based on Adjacent Nodes Applied to the Periodic Capacitated Arc Routing Problem (PCARP)

J. Oliveira, and C. Scapin

Abstract—Periodic Capacitated Arc Routing Problem (PCARP) is a combinatorial problem of hard resolution and due to its intrinsic intractability, researchers have concentrated on designing heuristic methods to obtain good quality solutions. In this work, the aim is to use the Relax-and-Fix heuristic (R&F) as a method of resolution of PCARP. The R&F heuristic consists of decomposing the original model in smaller submodels that are probably easier to be solved. Some factors that contribute to the success of the heuristics are applied decompositions. In this work we introduce a new form of decomposition which is not found in the literature yet as far as we know called Relax-and-Fix Adjacent Nodes (R&F-AN). This new strategy considers the characteristics of the PCARP and the decomposition of the model occurs by exploring in adjacencies in the associated graph. The efficiency of this new approach is evaluated on a set of 23 instances in the literature. Results have shown a reduction of 45% in the number of infeasible problems, when it is compared with other classic strategies of R&F heuristic decomposition. The R&F-AN has obtained 6 new better quality solutions than those already known, being 4 times faster in mean than the attempt of the resolution of the PCARP in the exact form.

Index Terms—Heuristic algorithms, PCARP, Relax-and-fix.

INTRODUÇÃO

OS problemas de roteamento em veículos podem ser classificados em: problemas de roteamento em nós e problemas de roteamento em arcos [1]. Os Problemas de Roteamento em Arcos (ARP) são os mais antigos problemas relacionados a grafo, a primeira referência vem do problema das pontes de Königsberg de 1736 [2]. Os ARPs referem-se aos problemas de roteamento onde os clientes e/ou atividades estão localizados nos arcos [3].

O Problema de Roteamento em Arcos Capacitados (CARP), proposto em [4], pode estar associado à capacidade dos veículos, a um tempo máximo de trabalho ou, também, a uma distância percorrida. O objetivo é determinar um conjunto de rotas de custo mínimo, onde as arestas com demandas devem ser atendidas respeitando a capacidade dos veículos [5].

O CARP é considerado *NP-hard* e, portanto, não se espera que instâncias de grande porte sejam resolvidas por algoritmo exato, uma vez que, os melhores algoritmos exatos são limitados a instâncias moderadas com 140 vértices e 190 arestas [5].

Em [6] propõem-se uma nova classe de problemas. A proposta é uma extensão do CARP, estendendo o período de trabalho de um único dia para um horizonte de tempo de vários dias, denominados de Problemas de Roteamento em Arcos Capacitado e Periódico (PCARP) ou *periodic* CARP. Os autores destacam que o PCARP é útil para modelagem de problemas que exigem tomadas de decisões em níveis táticos e operacionais, uma vez que essa classe de problemas é capaz de abordar ambos os níveis em paralelo.

Devido sua natureza teórica e importância prática, o PCARP tem atraído esforços de pesquisadores nas últimas décadas [7]. Dentre estes, pode-se citar: coleta de resíduos urbanos ([6], [8] e [9]), monitoramento de redes rodoviárias ([1] e [10]), supressão de poeira em estradas ([11] e [12]), manutenção preditiva e preventiva de linhas férreas ([13] e [14]) e irrigação de árvores ([15]), entre outras. O objetivo do PCARP é determinar uma rota de custo mínimo, na qual todas as viagens comecem e terminem em um depósito e as tarefas sejam atendidas de acordo com suas frequências ao longo do horizonte de tempo, respeitando o espaçamento e a capacidade dos veículos [6].

Sob a ótica da otimização, o PCARP pertence à categoria *NP-hard*, pois inclui o CARP como caso particular. Em cenários reais, o modelo matemático pode conter centenas ou milhares de restrições, fazendo com que a complexidade e tempo de resolução cresçam exponencialmente. Portanto, algoritmos são propostos com o intuito minimizar o custo computacional [16]. Consequentemente, devido a intratabilidade intrínseca do PCARP, pesquisadores concentram-se em projetar métodos heurísticos para determinar soluções de boa qualidade em menor tempo computacional [7]. Alguns métodos heurísticos propostos na literatura para PCARP: *Scatter Search* ([17] e [9]), *Decreasing Frequencies Nearest Insertion Heuristic* ([18]), *Best Insertion Heuristic* ([18]), *Lower Bound Heuristic* ([18]), *Adaptive Large Neighborhood Search* ([11] e [12]) e *Two Phased Hybrid Local Search* ([7]).

Outros métodos como as metaheurísticas são utilizadas na resolução do PCARP. As metaheurísticas são baseadas em métodos determinísticos ou aleatórios para determinar soluções viáveis ou ótimas [19]. Como descrito em: [6] que propõem um *Memetic Algorithm*, que envolve um *Genetic Algorithm*; [20] desenvolvem um método híbrido, combinando a metaheurística

Este trabalho foi financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

J. D. Oliveira, Universidade Estadual do Centro-Oeste / Research Group of Technology Applied to Optimization (GTAO), Irati, Paraná, Brasil, jailson-deoliveira@hotmail.com.

C. T. Scapin, Universidade Federal do Paraná (UFPR) / Research Group of Technology Applied to Optimization (GTAO), Curitiba, Paraná, Brasil, cassiusts@gmail.com.

Ant Colony Optimization com a heurística *Insertion In Best Combination*; [21] incorporam a heurística *Route Merging a um Memetic Algorithm*; [15] propõem a combinação da *Ant Colony Optimization* com uma heurística de melhoria *2-opt*.

Neste trabalho, foi proposto, o uso da heurística *relax-and-fix* (R&F) como método de solução para o PCARP. A heurística R&F descrita em [22], consiste em um método de decomposição de um problema de programação inteira mista em subproblemas menores disjuntos, que podem ser resolvidos rapidamente, porém sem a garantia de resolução do problema original de forma ótima.

A heurística R&F tem sido aplicada para solucionar problemas de otimização combinatoria, por exemplo, em: [23], [24], [25], [26], [27], [28] e [29]. Estratégias do tipo R&F tem sido utilizada com sucesso para resolver problemas cuja decisões são tomadas ao longo de um horizonte de tempo, onde cada decisão tem um impacto nas decisões futuras. Porém, esse impacto tende a diminuir ao longo do horizonte [28].

O objetivo das heurísticas é explorar de forma direcionada as características do modelo, determinando soluções aceitáveis em tempo computacional curto. A heurística R&F tem sido amplamente aplicada para solucionar problemas de dimensionamento de lotes: [23], [24], [25], [26] e outros. A R&F possui um conjunto de estratégias de decomposição clássica para explorar os modelos de dimensionamento de lotes, mas essas estratégias não são suficientes para tratar o PCARP. Em razão disso, é proposto nesse trabalho uma estratégia inédita de decomposição denominada *Relax and Fix Adjacent Nodes* (R&F-AN), na qual, as características do PCARP são exploradas.

A estratégia R&F-AN é baseada em adjacência de nós/vértices a partir de um determinado nó. A decomposição do modelo ocorre levando em consideração seus nós adjacentes. A estratégia proposta apresentou bom desempenho quando comparada com as estratégias clássicas do R&F.

O artigo está organizado da seguinte forma: a Seção 2 é apresentado a formulação matemática do PCARP abordado nesse trabalho. Seção 3 apresenta a descrição da heurística R&F e as estratégias desenvolvidas. Testes computacionais são apresentados na Seção 4. As conclusões gerais do trabalho são apresentadas na Seção 5.

II. FORMULAÇÃO MATEMÁTICA

No PCARP cada aresta do grafo possui demandas periódicas, as quais devem ser atendidas dentro de um horizonte de tempo composto por mais de um período. O objetivo é definir rotas de custo mínimo para cada veículo de modo que as demandas sejam atendidas sem exceder a capacidade do veículo.

O modelo a seguir é descrito em [14] e é definido da seguinte forma: Seja um grafo não direcionado $G = (X, E)$ com n nós (vértices) $X = \{1, 2, \dots, n\}$ e m arestas com $E = \{1, 2, \dots, m\}$ que devem ser percorridos por nk veículos definidos pelo conjunto $K = \{1, 2, \dots, nk\}$. Cada aresta que conecta o nó i ao nó j pode ser denotada por $[i, j]$, com $i, j \in X$ e $i \neq j$. O custo associado a cada aresta é c_{ij} .

A capacidade dos veículos é associada a máxima distância que estes podem se deslocar em um dia, nesse modelo é considerada uma aresta por dia. O horizonte de tempo H é formado por np períodos $H = \{1, 2, \dots, np\}$, onde cada período

é representado por p . A demanda é considerada a periodicidade de cada aresta, que é expressa na quantidade máxima de períodos em que uma aresta deve ser atendida ao menos uma vez $MP[i, j]$. O modelo descrito em [14] é baseado em Programação Linear, em que as variáveis são:

$$\begin{aligned} x_{ijkp} & \begin{cases} 1, \text{ se o carro } k \text{ se desloca do ponto } i \text{ para o ponto } \\ j \text{ no período } p \\ 0, \text{ caso contrário} \end{cases} \\ pn_{ijp} & \begin{cases} 1, \text{ se o arco } [i, j] \text{ não respeita a periodicidade no } \\ \text{período } p \\ 0, \text{ caso contrário} \end{cases} \\ f_{ikp} & \begin{cases} 1, \text{ se o carro } k \text{ fica parado no ponto } i \text{ no dia } p \\ 0, \text{ caso contrário} \end{cases} \end{aligned}$$

Os índices i e j estão associados as arestas $[i, j]$, podendo ser eles os pontos iniciais ou finais. O índice p está associado aos períodos e k aos veículos. Cada variável pn_{ijp} está associada a uma punição PU_{ij} , caso uma aresta não tenha sua periodicidade atendida. Essa variável permite atrasos no atendimento das arestas, se necessário. A variável f_{ikp} permite que um veículo folgue em um determinado dia, possibilitando em alguns casos obter uma solução com menor quantidade de deslocamentos. As variáveis pn_{ijp} e f_{ikp} foram propostas para que o modelo seja viável.

Modelo:

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

Sujeito à:

$$\begin{aligned} \sum_{[i,j] \in E} x_{ijkp} + f_{jkp} - \sum_{[i,j] \in E} x_{jik,p+1} - f_{jk,p+1} &= 0 \\ \forall j \in X, \forall k \in K, \forall p \in H \end{aligned} \quad (2)$$

$$\begin{aligned} \sum_{[i,j] \in E} x_{ijkp} + \sum_{[i,j] \in E} x_{jikp} + \sum_{i=1}^n f_{ikp} &= 1 \forall p \in H, \\ \forall k \in K \end{aligned} \quad (3)$$

$$\begin{aligned} \sum_{k=1}^{nk} \left(x_{ijkp} + x_{jikp} + x_{ijk,p+1} + x_{jik,p+1} + \dots \right. \\ \left. + x_{ijk,p+MP(x_{ij})-1} + x_{jik,p+MP[i,j]-1} \right) + pn_{ijp} \\ \geq 1 \forall [i, j] \in E, \forall p \in H \end{aligned} \quad (4)$$

$$x_{ijkp}, pn_{ijp}, f_{ikp} \in \{0,1\} \forall [i, j] \in E, \forall k \in K, \forall p \in H \quad (5)$$

O objetivo do modelo (1) é minimizar custos de deslocamentos e atrasos. As restrições (2) garantem o fluxo diário dos veículos e que o destino de um veículo ao final de um dia deve ser o ponto de partida no próximo dia. As restrições (3) garantem que todos os veículos terão uma designação para cada dia p , isto é, cada veículo no dia p pode se deslocar de i para j , ou de j para i , ou ainda permanecer parado em um ponto qualquer i . As restrições (4) se referem à periodicidade com que cada aresta deve ser atendida. Observa-se que caso a variável pn_{ijp} assumia valor 1, indica que a aresta $[i, j]$ teve sua periodicidade atrasada em um dia implicando em uma punição PU_{ij} na função objetivo. Observa-se que, nos casos em que um conjunto de arestas A é formado por arestas que demandam apenas uma passagem em todo o horizonte de tempo, as restrições (4) podem ser simplificadas e a variável pn_{ijp} descartada, dando origem as restrições (6):

$$\sum_{k=1}^{nk} \left(x_{ijk1} + x_{jik1} + x_{ijk,2} + x_{jik,2} + \dots \right) \geq 1 \quad (6)$$

$$\forall [i, j] \in A$$

Por fim, as restrições (5) determinam que todas as variáveis do modelo sejam binárias.

III. HEURÍSTICA RELAX-AND-FIX

A heurística propõe solucionar o problema em etapas, sendo que, cada etapa, resolve de forma exata um subproblema derivado do problema original. Espera-se que cada subproblema seja, possivelmente, mais fácil de ser resolvido, pois ocorre uma redução no esforço computacional proporcionado pela diminuição do número de variáveis inteiras. Para tanto, baseia-se na partição das variáveis inteiras ou binárias do modelo, criando R conjuntos disjuntos $Q^r, r = 1, \dots, R$.

O conjunto de variáveis originais é dividido em três grupos: o primeiro grupo contém as variáveis inteiras e/ou binárias; o segundo grupo abrange as variáveis relaxadas (livre), isto é, variáveis binárias são consideradas contínuas no intervalo $[0, 1]$ e variáveis inteiras contínuas \mathbb{R}^+ ; e o terceiro grupo é formado por variáveis fixas, as quais são fixadas de acordo com a solução de submodelos precedentes a iteração atual. O número R determina o número de iterações da heurística.

O comportamento clássico dessa heurística é descrito em [30]. Considere um problema inteiro misto (MIP), supondo que as variáveis binárias y são particionadas em R conjuntos disjuntos Q^1, Q^2, \dots, Q^R . Então, resolve-se sequencialmente os R MIP's, denotados por MIP^r , cada MIP^r define um subproblema referente a uma partição r com $1 \leq r \leq R$. Ao final do processo, espera-se determinar uma solução factível para o MIP original. Uma descrição formal do processo pode ser observada no Algoritmo 1.

A heurística deve ser interrompida caso algum subproblema gerado no passo 1 do algoritmo seja infactível. Isso significa que não foi possível obter uma solução para o submodelo MIP^r , considerando as variáveis fixadas nas partições anteriores. A fim de evitar esse status de solução, que pode ocorrer no algoritmo R&F clássico, em [31] é apresentado uma variante

para o algoritmo R&F, o qual é denominado *Enhanced Fix-and-Relax*. Ao detectar a infactibilidade na resolução de um subproblema MIP^r , a heurística tem o potencial de retroceder um ou mais passos. Ao realizar esse processo o algoritmo gera um subproblema MIP^{aux} , nesse subproblema as variáveis pertencentes aos conjuntos Q^i ($i = aux, \dots, r$) são integralizadas, aux pode assumir os valores $aux = r - 1, r - 2, \dots, 1$ de acordo com o número de passos que a estratégia retrocedeu. A abordagem do subproblema tende a se aproximar do problema original. O algoritmo 2 descreve a heurística *Enhanced Relax-and-Fix* (E-R&F).

Algoritmo 1: Relax-and-Fix Clássico.

Defina uma partição $Q^r, r = 1, \dots, R$ para as variáveis;

Para $r = 1, \dots, R$ **faça**

Relaxe as variáveis inteiras/binárias para os subconjuntos $Q^i, i = r + 1, \dots, R$;

Mantenha a integralidade das variáveis do subconjunto Q^r ;

se $r > 1$

Fixe as variáveis pertencentes aos subconjuntos $Q^i, i = 1, \dots, r - 1$, conforme a solução dos MIP^r anteriores;

Fim

Resolva o MIP^r gerado

se MIP^r infactível **então**

O particionamento não foi capaz de determinar uma solução viável;

Exit para

Fim

$r = r + 1$;

Fim

Algoritmo 2: Enhanced Relax-and-Fix

Defina uma partição $Q^r, r = 1, \dots, R$ para as variáveis.

Relaxe as variáveis inteiras/binárias para os subconjuntos $Q^r, r = 1, \dots, R$;

$r = 0$

enquanto $r < R$ **faça**

$r = r + 1$;

Integralize as variáveis do subconjunto Q^r ;

Resolva o subproblema MIP^r gerado;

se $r > 1$ e MIP^r infactível **então**

$aux = r$;

enquanto $aux > 1$ **faça**

$aux = aux - 1$;

Integralize as variáveis dos subconjuntos $Q^i, i = aux, \dots, r$;

Resolva o subproblema MIP^{aux} gerado;

se MIP^{aux} factível **então**

exit enquanto;

Fim

Fim

Fim

Fixe as variáveis do subconjunto $Q^1 \cup \dots \cup Q^r$;

Fim

No pior dos casos a heurística E-R&F resolve o problema *MIP* original. A escolha das partições das variáveis do *MIP* afeta o desempenho da heurística. Portanto, as partições do problema devem contemplar as características e particularidades de cada modelo, para que seja possível gerar soluções factíveis e de boa qualidade.

O modelo PCARP da seção II foi adaptado para ser resolvido pela heurística R&F. Foram propostas 5 estratégias distintas de decomposição das variáveis x_{ijkp} , pn_{ijp} e f_{ikp} . Quatro consistem na decomposição do horizonte de tempo H e uma considera a decomposição por adjacência ou vizinhança de vértices (nós). A estratégia que realiza a decomposição por nós adjacentes, ainda não tinha sido proposta na literatura, no melhor do nosso conhecimento. Estas estratégias são descritas a seguir.

Estratégia Relax-and-Fix Forward (R&F-F)

Nessa estratégia as variáveis do problema são divididas em R subconjuntos Q^r , com $r = 1, \dots, R$, a serem resolvidos de modo que cada subproblema MIP^r corresponda a um período/dia r do horizonte de tempo H . A ordem da resolução é a cronológica, isto é, inicia-se a resolução no subproblema correspondente ao primeiro período do horizonte ($r = 1$) de tempo e finalizando no subproblema correspondente ao último período ($r = R$), subproblema MIP^R .

A Fig. 1 a) ilustra essa estratégia. Para $r = 1$ tem-se: variáveis x_{ijkp} , pn_{ijp} , $f_{ikp} \in Q^1$ como binária; variáveis x_{ijkp} , pn_{ijp} , $f_{ikp} \in Q^r$ $r = 2, \dots, R$ relaxadas, contínuas no intervalo $[0,1]$. Na segunda iteração $r = 2$: variáveis x_{ijkp} , pn_{ijp} , $f_{ikp} \in Q^1$ são fixadas, tornando-se parâmetros; variáveis x_{ijkp} , pn_{ijp} , $f_{ikp} \in Q^2$ são mantidas binárias e x_{ijkp} , pn_{ijp} , $f_{ikp} \in Q^r$ $r = 3, \dots, R$ relaxadas.

Estratégia Relax-and-Fix Backward (R&F-B)

É similar à estratégia R&F-F, porém, a sequência de resolução dos subproblemas MIP^r ocorrem em ordem cronológica inversa, a resolução é iniciada pelo subproblema MIP^R e é finalizada com a resolução do subproblema MIP^1 .

Estratégia Relax-and-Fix Overlapping (R&F-O)

Baseia-se na partição do problema original em R períodos com sobreposição de partições. Dada uma partição r , as variáveis pertencentes as partições Q^r e Q^{r-1} , com $2 \leq r \leq R$ são mantidas binárias e as demais variáveis subsequentes a esses períodos são relaxadas. Na iteração seguinte, as variáveis da partição Q^{r-1} são fixadas, as variáveis dos subconjuntos Q^r e Q^{r+1} mantidas binárias e as demais variáveis relaxadas. O processo ocorre até que o último subproblema seja resolvido. A Fig. 1 b) ilustra as duas primeiras iterações para essa estratégia.

Estratégia Relax-and-Fix Overlapping 2

É baseada na sobreposição de partições, similar à estratégia anterior. Nesta proposta, cada subproblema MIP^r é resolvido mantendo as variáveis correspondentes aos períodos $r - 3$, $r - 2$, $r - 1$ e r binárias ($3 \leq r \leq R$) e as demais variáveis dos períodos subsequentes relaxadas. No próximo subproblema, as variáveis correspondentes aos períodos $r - 3$ e $r - 2$ são fixadas, as variáveis dos períodos $r - 1$, r , $r + 1$ e $r + 2$ são

binárias e as demais variáveis subsequentes relaxadas e, assim, sucessivamente até o último subproblema ser resolvido.

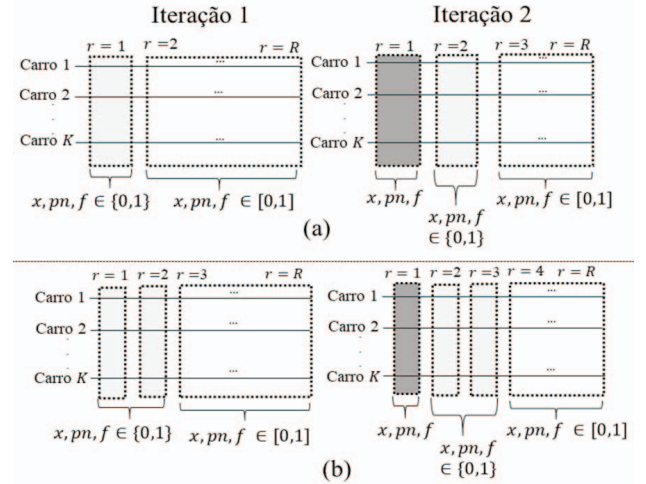


Fig. 1. Estratégias Relax-and-Fix.

Relax-and-Fix Adjacent Nodes (R&F-AN)

Essa estratégia de decomposição é baseada em adjacência ou vizinhança de nós. Se $[i, j]$ é uma aresta, então i e j são nós adjacentes ou vizinhos. Denota-se por $N(i)$ o conjunto de vértices pertencentes a vizinhança de i e $N[i] = N(i) \cup \{i\}$ como vizinhança fechada de i .

O modelo descrito nesse trabalho baseia-se em um grafo não direcionado $G = (X, E)$. Para inicializar a heurística R&F-AN, escolhe-se um nó $i^* \in X$ para ser o nó raiz (nó de origem), na primeira iteração as variáveis binárias x_{ijkp} , pn_{ijp} e f_{ikp} são decompostas em dois subconjuntos disjuntos. As variáveis x_{ijkp} , pn_{ijp} e f_{ikp} que possuem combinações do índice i^* (nó raiz) com sua vizinhança aberta $N(i^*)$ são mantidas binárias e as demais variáveis são relaxadas. A partir da segunda iteração os nós $i \in N(i^*)$ tornam-se nós raízes i^* e para cada subproblema gerado, as variáveis x_{ijkp} , pn_{ijp} e f_{ikp} são decompostas em três subconjuntos disjuntos:

Primeiro subconjunto – As variáveis x_{ijkp} , pn_{ijp} e f_{ikp} com índices formados pelas combinações dos nós i^* da iteração atual com os nós i_{ant}^* anteriores são fixadas de acordo com a solução do subproblema anterior;

Segundo subconjunto – As variáveis x_{ijkp} , pn_{ijp} e f_{ikp} com índices formados pelas combinações dos nós i^* com suas respectivas vizinhanças abertas $N(i^*) - \{i_{ant}^*\}$ são mantidas binárias;

Terceiro subconjunto – As variáveis x_{ijkp} , pn_{ijp} e f_{ikp} não pertencentes a nenhum dos subconjuntos anteriores são relaxadas.

Para ilustrar o funcionamento dessa estratégia, considere o grafo não direcionado $G = (X, E)$ com $|X| = 8$ e $|E| = 10$. Na Fig. 2 pode ser observado a representação do grafo G , bem como, as iterações da heurística. Nesse exemplo foi adotado o nó 6 como nó raiz inicial. Para esse primeiro subproblema, as variáveis x_{ijkp} , pn_{ijp} e f_{ikp} com índices formados pelas combinações do nó 6 com os nós $i \in N(6) = \{1, 8\}$ são mantidas binárias e as demais variáveis são relaxadas. Na segunda iteração tem-se o primeiro subproblema resolvido e os nós 1 e

8 passam a ser nós raízes atuais. As variáveis com índices formados pelas combinações dos nós 1 e 8 com o nó raiz anterior 6 são fixadas. As variáveis com índices formados pelas combinações dos nós 1 e 8 com suas respectivas vizinhanças $N(1) - \{6\} = \{2, 4, 5\}$ e $N(8) - \{6\} = \{7\}$ tornam-se binárias e as demais variáveis são relaxadas.

Na terceira e última iteração, os nós 2, 4 e 5 passam a ser nós raízes e as variáveis que possuem índices formados pelas combinações desses nós com os nós raízes anteriores $\{1,8\}$ são fixadas de acordo com a solução do subproblema anterior. Para as variáveis formadas pelas combinações dos nós raízes e suas respectivas vizinhanças $N(2) - \{1, 8\}$, $N(4) - \{1, 8\}$ e $N(5) - \{1, 8\}$ tornam-se binárias. No final do procedimento, caso nenhum subproblema gerado seja infactível encontra-se uma solução viável para o problema original.

Para a estratégia R&F-AN não é possível determinar o número de iterações a priori para a heurística, uma vez que esse número está vinculado com o nó que é adotado como nó raiz. No exemplo, tomando o nó 6 como nó raiz tem-se três iterações, caso inicialize pelo nó 3 como raiz tem-se 4 iterações. Como pode ser observado nessa estratégia optou-se pela busca em largura.

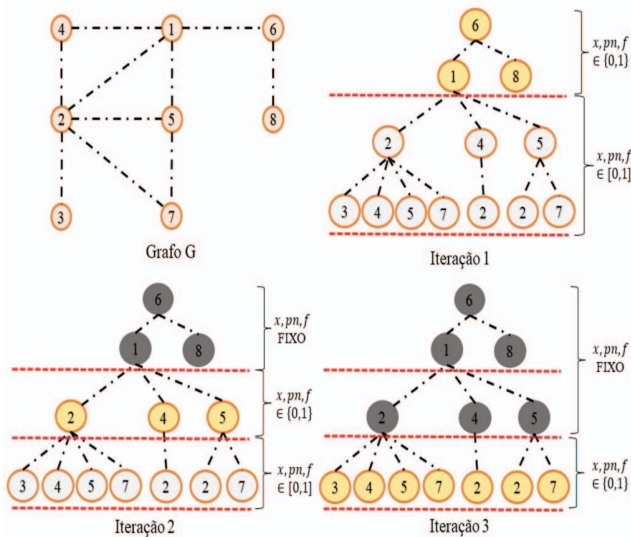


Fig. 2. Estratégia R&F-AN.

IV. TESTES COMPUTACIONAIS

As estratégias propostas para a heurística *relax-and-fix* foram implementadas em *Microsoft Visual Studio 2015*, utilizando pacote de otimização *Gurobi 7.0*. Os testes foram realizados utilizando um computador com processador *Intel Core i7-5500U* com velocidade de *2.4GHz*, memória de RAM de 8GB sob o sistema operacional *Windows 10*.

Como critério de parada adotou-se o tempo de execução de 600 segundos por iteração, embora nenhum problema tenha utilizado mais de 600 segundos por iteração. Para cada instância foi calculado o desvio entre a solução obtida pela heurística R&F (Z_H) e as soluções encontradas em [13], os quais foram executados no *software CPLEX 12.4* (configuração *default*) (Z_C) com um tempo limite para obtenção de uma solução de 86400 segundos (24 horas). Este desvio é calculado da seguinte forma:

$$Desvio = \frac{Z_H - Z_C}{Z_C} \tag{7}$$

Para avaliar as estratégias propostas foram utilizadas as 23 instâncias proposta por [13], sendo estas adaptadas de [32]. Para maiores detalhes consultar [13]. As características das instâncias são apresentadas na Tabela I, onde necessidades é a soma das necessidades de cada arco (número de atendimentos no arco).

A Tabela II apresenta os resultados computacionais relativos a aplicação das heurísticas *relax-and-fix* clássica, R&F-F, R&F-B, R&F-O, R&F-O2 e R&F-AN. O símbolo “--” indica que a estratégia gerou subproblemas infactíveis, não sendo capaz de determinar uma solução viável para o problema original, Z indica a solução e D o desvio calculado com a equação (7).

TABELA I
CARACTERÍSTICAS DAS INSTANCIAS GDB

Ins (gdb)	Vértices	Arestas	Veículos	Necessidades	Dias
1	12	22	1	32	38
2	12	26	1	44	53
3	12	22	2	31	19
4	11	19	2	34	21
5	13	26	2	53	32
6	12	22	3	32	14
7	12	22	3	37	45
8	27	46	3	67	27
9	27	51	4	78	24
10	12	25	4	70	21
11	22	45	1	50	60
12	13	23	1	33	40
13	10	28	2	33	20
14	7	21	2	32	20
15	7	21	2	48	29
16	8	28	3	40	24
17	8	28	3	52	21
18	9	36	3	81	33
19	8	11	4	29	12
20	11	22	4	53	16
21	11	33	5	46	12
22	11	44	5	88	22
23	11	55	6	82	17

A estratégia R&F-AN apresenta o melhor desempenho quando comparada com as demais, não sendo capaz de determinar solução viável para apenas 4 instâncias (gdb8, gdb9, gdb10 e gdb21). As instâncias gdb8 e gdb9 são de difícil resolução, pois apresentam uma quantidade elevada de vértices, arestas e necessidades. É possível evidenciar ainda que a solução é de baixa qualidade mesmo no uso do CPLEX com tempo limite de 24 horas.

A estratégia R&F-AN determinou três soluções viáveis (gdb2, gdb4 e gdb18) de melhor qualidade que as encontradas em [13]. Obteve-se ao todo 19 soluções viáveis, sendo que dessas os problemas gdb13, gdb14, gdb15, gdb16, gdb17, gdb21 e gdb23 apresentam desvio igual a 0, já os problemas gdb3, gdb6, gdb7, gdb19 e gdb22 apresentaram desvio igual ou inferior a 0,04, ficando muito próximas da solução encontrada pelo método exato. O desvio médio da heurística é de 3,13, valor abaixo das demais estratégias. A estratégia R&F-AN reduziu em 45% o número de problemas infactíveis, quando comparada com as demais estratégias. Observa-se ainda que em nenhum dos problemas esgotou-se o tempo limite por iteração.

TABELA II
RESULTADOS OBTIDOS COM APLICAÇÃO DAS ESTRATÉGIAS

Ins gdb	R&F-F		R&F-B		R&F-O		R&F-O2		R&F-AN		[13]
	Z	D	Z	D	Z	D	Z	D	Z	D	Z
1	--	--	--	--	--	--	--	--	3841	7.63	4450
2	--	--	--	--	--	--	--	--	7606	-0.35	1162
3	--	--	--	--	--	--	--	--	344	0.01	340
4	--	--	--	--	--	--	1450	3.08	2562	-0.28	3555
5	--	--	--	--	--	--	--	--	2070	10.8	1750
6	--	--	--	--	447	8.82	459	0.01	465	0.02	456
7	--	--	--	--	--	--	1552	28.9	526	0.02	518
8	--	--	--	--	--	--	--	--	--	--	1239
9	--	--	--	--	--	--	--	--	--	--	1345
10	--	--	--	--	--	--	--	--	5788	6.89	734
11	--	--	--	--	--	--	--	--	--	--	6443
12	--	--	--	--	--	--	--	--	1751	31.6	536
13	--	--	--	--	620	--	620	620	620	0.00	620
14	--	--	--	--	--	--	145	145	145	0.00	145
15	--	--	--	--	--	--	3144	21.4	140	0.00	140
16	--	--	18	0.0	185	0.02	184	0.01	182	0.00	182
17	--	--	--	--	318	17.9	--	--	168	0.00	168
18	--	--	--	--	0	3	--	--	389	-0.06	415
19	213	16.0	13	0.0	129	0.03	135	0.08	127	0.02	125
20	--	--	--	--	--	--	6258	1.74	--	--	2280
21	--	--	--	--	--	--	212	--	212	0.00	212
22	--	--	--	--	--	--	--	--	434	0.00	433
23	--	--	--	--	--	--	336	--	336	0.00	336

A Tabela III apresenta os tempos em segundos ($T(s)$) gastos na execução de cada estratégia. Valores indicados “--” representam problemas que utilizaram toda a memória do computador antes de chegar a uma solução ótima ou atingir o tempo limite. Observa-se que a resolução de [13] evidencia, na maioria dos casos, tempo de execução superior às estratégias propostas.

Esses resultados já eram previstos, uma vez que as estratégias possuem tempo predefinido. [13] conseguiu determinar solução ótima para a instância gdb14 com tempos de 261s. As estratégias R&F-O2 e R&F-AN chegaram a mesma solução para a instância gdb14.

A inactibilidade ocorreu com grande frequência nas estratégias em que a decomposição por período foi realizada, conforme observado na Tabela II. Para compreender o elevado número de problemas inactíveis será analisada a instância gdb19 resolvida pela estratégia R&F-F. A Fig. 3 ilustra a configuração da solução do subproblema 10 gerado na 10ª iteração da heurística na resolução da instância gdb19.

Nota-se que a instância possui arestas com três periodicidades diferentes, que devem ser atendidas a cada 4, 6 e 12 dias com demandas 1, 2 e 3 passagens no horizonte de 12 dias. Observa-se que o veículo 1 inicia sua rota no vértice 1, veículo 2 no vértice 2, veículo 3 no vértice 4 e veículo 4 no vértice 7. O problema original foi decomposto em 12 subproblemas a serem resolvidos pela estratégia R&F-F, ocorrendo inactibilidade na resolução do 11º subproblema.

A seguir são descritos os movimentos realizados por cada veículo até o 10º dia do horizonte de tempo.

Carro 1: 1-2-5-5-1-6-6-1-4-2-5.
Carro 2: 2-2-7-3-2-1-1-5-5-5-7.
Carro 3: 4-2-2-2-2-2-5-7-7-7-2.
Carro 4: 7-5-1-1-4-2-7-3-2-1-1.

TABELA III
TEMPO DE EXECUÇÃO DAS ESTRATÉGIAS

Ins gdb	R&F-F	R&F-B	R&F-O	R&F-O2	R&F-AN	[13]
	$T(s)$	$T(s)$	$T(s)$	$T(s)$	$T(s)$	$T(s)$
1	--	--	--	--	199	39000 ¹
2	--	--	--	--	2155	70423 ¹
3	--	--	--	--	1749	36906 ¹
4	--	--	--	105	1292	49595 ¹
5	--	--	--	--	3480	86400
6	--	--	260	180	1284	86400
7	--	--	--	270	3746	55767 ¹
8	--	--	--	--	--	86400
9	--	--	--	--	--	86400
10	--	--	--	--	718	86400
11	--	--	--	--	--	31510 ¹
12	--	--	--	--	383	86400
13	--	--	111	111	602	86400
14	--	--	--	105	790	261
15	--	--	--	1681	6850	63135 ¹
16	--	182	182	145	8249	35301 ¹
17	--	--	305	--	9327	77693 ¹
18	--	--	--	--	10444	86400
19	122	135	145	125	1640	47566 ¹
20	--	--	--	137	--	67903 ¹
21	--	--	--	122	7380	38181 ¹
22	--	--	--	--	8272	86400
23	--	--	--	1432	14686	86400
<i>t médio</i>	122	158,5	200.6	401.18	4624.78	64227.9

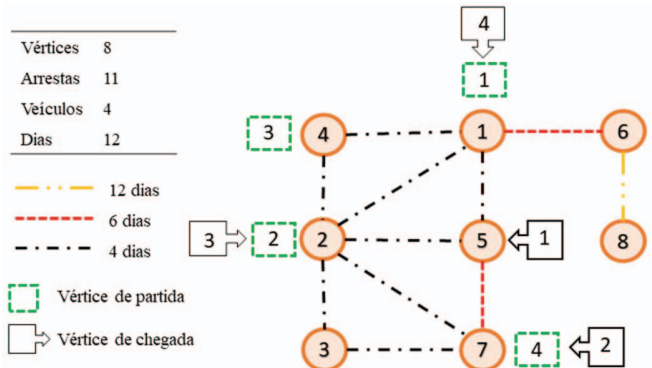


Fig. 3. Configuração da solução subproblema 10 da instância gdb19.

No final do 10º dia, o veículo 1 está no vértice 5, o veículo 2 no vértice 7, o veículo 3 no vértice 2 e o 4º no vértice 1. As arestas 1-5 e 3-7 necessitam passagens no 11º dia, já as arestas 1-4, 1-6, 2-3, 2-4 e 6-8 no 12º dia, totalizando 7 necessidades. Considerando a existência de dois dias no horizonte de tempo e um total de 4 veículos, o problema disponibiliza 8 movimentos sobre o grafo que, teoricamente, poderia atender as 7 arestas que ainda exigem passagens com folga de um dia para um veículo.

Observa-se que a aresta 6-8 requer atendimento no 12º dia, que poderia ser atendido pelo veículo 4 que se encontra no vértice 1 no 10º dia. Todavia, devido às restrições do modelo, as quais exigem que no final do horizonte de tempo os veículos retornem para o nó onde iniciaram suas rotas, não é possível satisfazer tais restrições, tornando o problema inactível. Portanto, nesse caso, essas restrições não podem ser atendidas,

pois com dois movimentos o veículo 4 não conseguiria voltar ao vértice que marca o início da sua rota.

Além disso, um fator que pode contribuir para a infactibilidade é número de folgas por veículos. Na resolução do 11º subproblema, o veículo 1 havia folgado 2 vezes, o veículo 2 folgado 4 vezes, o veículo 3 folgado 6 vezes e o veículo 4 folgado 2 vezes totalizando 14 folgas até o 10º dia, o que comprometeu a viabilidade de solução do subproblema. A fim de evitar esse comportamento, foi adaptado o algoritmo *Enhanced Relax-and-Fix* que permite à heurística voltar algumas iterações caso ocorra infactibilidade, podendo estruturar de melhor forma as rotas e evitar essa configuração de solução na iteração corrente.

Como observado na Tabela II à estratégia R&F-AN apresentou desempenho superior às demais estratégias propostas. Portanto, nesse último conjunto de testes optou-se em adaptar apenas a estratégia R&F-AN, a qual foi denominada *Enhanced Relax-and-Fix Adjacent Nodes* (E-R&F-AN). A Tabela IV apresenta os resultados.

Os valores em negrito correspondem a melhor solução encontrada para cada instância. Observa-se que [13] encontrou solução ótima apenas para o problema gdb14. Para 12 instâncias o modelo exato esgotou a memória do computador antes de chegar a uma solução ótima ou atingir o tempo limite de 24 horas.

TABELA IV
RESULTADOS E-R&F-NA

Ins gdb	E-R&F-AN			[13]	
	Z	D	t(s)	Z	t(s)
1	1456	-0.673	6144.6	4450	39000 ¹
2	3613	-0.689	9053	11622	70423 ¹
3	344	0.012	8274	340	36906 ¹
4	2562	-0.279	8045	3555	49595 ¹
5	6754	2.859	19207	1750	86400
6	456	0.000	6613	456	86400
7	526	0.015	10810	518	55767 ¹
8	23373	0.886	71184	12395	86400
9	45450	2.378	68779	13454	86400
10	3757	4.119	19634	734	86400
11	2451	-0.962	26732	64437	31510 ¹
12	1503	1.804	6028	536	86400
13	620	0.000	3827	620	86400
14	145	0.000	706.02	145	261
15	140	0.000	8189	140	63135 ¹
16	182	0.000	8182	182	35301 ¹
17	168	0.000	9952	168	77693 ¹
18	389	-0.063	10466	415	86400
19	127	0.016	2550	125	47566 ¹
20	1279	-0.439	17308	2280	67903 ¹
21	212	0.000	10581	212	38181 ¹
22	434	0.002	9370	433	86400
23	336	0.000	16192	336	86400

Para as instâncias gdb1, gdb2, gdb4, gdb11, gdb18 e gdb20 a heurística E-R&F-AN obteve soluções com desempenho superior as determinadas em [13]. Para os problemas gdb6, gdb13, gdb14, gdb15, gdb16, gdb17, gdb21 e gdb23 a heurística determinou soluções com desvio igual à zero da melhor solução conhecida. Soluções com desvio igual ou inferior a 0,04 são encontradas para os problemas gdb3, gdb7, gdb19 e gdb22. Os piores desempenhos da estratégia correspondem à resolução das instâncias gdb5, gdb9 e gdb12 apresentando desvio superior a 1,8. Destaca-se ainda que a estratégia em nenhuma instância esgotou a memória do

computador, fato esperado uma vez que o objetivo da heurística R&F é a resolução de subproblemas menores.

Com relação ao tempo de execução, houve um ganho significativo, a E-R&F-AN teve um tempo médio de 15659,4 segundos (aproximadamente 4 horas) por instância, já a resolução pelo método exato obteve um tempo médio de 64218,8 segundos (aproximadamente 17 horas) por instância.

A aplicação da E-R&F-AN no modelo apresentou bom desempenho, pois obteve 6 soluções melhores do que as já conhecidas na literatura, 11 soluções iguais ou muito próximas e apenas 3 soluções com valores piores. Com relação ao tempo de execução a estratégia foi em média 4 vezes mais rápida que o método exato.

V. CONCLUSÃO

O PCARP possui uma natureza *NP-hard* e quanto maior for a rede (grafo) e o número de restrições, sejam estas vinculadas à capacidade, tempo ou periodicidades, mais difícil é determinar uma solução viável. Tal dificuldade é compensada pelo ganho que o modelo pode trazer em aplicações reais.

Como método de solução para o modelo proposto em [13] e [14], propõe-se o uso da heurística *relax-and-fix*, que consiste na decomposição de um problema de programação inteira mista em subproblemas menores, descrito em [22]. Um dos fatores que determinam o sucesso da heurística é a escolha das partições das variáveis do problema, para o PCARP foi considerado duas estratégias de partição das variáveis: a decomposição em relação ao período e a decomposição em adjacência em nós. A decomposição em adjacência em nós é uma das principais contribuições desse trabalho, por ser uma estratégia inédita, na qual, as características do PCARP são exploradas de forma direcionadas.

Os resultados computacionais mostram que as heurísticas que utilizaram a decomposição por período possuem desempenho inferior à decomposição por adjacência. A R&F-AN obteve tempo de processamento maior, mas em contrapartida determinou soluções viáveis de melhor qualidade, conseguindo determinar soluções para 19 das 23 instâncias. O maior tempo gasto pela R&F-AN é 5.9 vezes menor que a execução quando comparada com os resultados de [13].

Devido ao alto número de problemas infactíveis encontrados na aplicação da R&F clássica, foi proposta uma adaptação da heurística *Enhanced Relax-and-Fix*. Os resultados desse segundo conjunto de testes, mostram a potencialidade da estratégia, visto que o número de problemas infactíveis foi nulo. A estratégia E-R&F-AN obtendo 6 soluções melhores que as já conhecidas e em nenhuma das instâncias esgotou a memória do computador, fato que foi comum na resolução pelo método exato (*solver* CPLEX 12.4), o qual esgotou a memória em 12 dos 23 problemas. Com relação ao tempo de execução a estratégia foi em média 4 vezes mais rápida que a execução do modelo exato.

A heurística *relax-and-fix*, em especial as estratégias R&F-AN e *enhanced* R&F-AN mostraram-se boas estratégias para determinar soluções viáveis para o PCARP estudado. Na maioria dos casos essas soluções são de boa qualidade. Houve ganho em tempo computacional em relação à resolução dos modelos por um *software* comercial e nenhum caso esgotou a memória do computador. Em casos em que as soluções

determinadas pela R&F encontram-se distantes de soluções aceitáveis, estas podem ser posteriormente aprimoradas com a aplicação de uma heurística de melhoria.

AGRADECIMENTOS

Agradecemos aos colegas do Grupo de Tecnologia Aplicada à Otimização (GTAO).

REFERÊNCIAS

- [1] I. M. Monroy, C.A. Amaya and A. Langevin, "The periodic capacitated arc routing problem with irregular services," *Discrete Applied Mathematics*, vol. 161, pp. 691–701, 2013.
- [2] M. Arenales, V. Armentano, R. Morabito and H. Yanasse, "Pesquisa Operacional," Rio de Janeiro: Elsevier, 2007.
- [3] C. Archetti, A. Corberán, I. Plana, J. M. Sanchis and M. G. Speranza, "A mathuristic for the team orienteering arc routing problem," *European Journal of Operational Research*, vol. 245, pp. 392–401, 2015.
- [4] B. L. Golden and R. T. Wong, "Capacitated arc routing problem," *Networks*, vol. 11, pp. 305–315, 1981.
- [5] Y. Chen, J. Hao and F. A. Glover, "A hybrid metaheuristic approach for the capacitated arc routing problem," *European Journal of Operational Research*, vol. 253, pp. 25–39, 2016.
- [6] P. Lacomme, C. Prins and W. Ramdane-Chérif, "Evolutionary algorithms for periodic arc routing problems," *European Journal of Operational Research*, vol. 165, pp. 535–553, 2005.
- [7] Y. Chen and J. Hao, "Two phased hybrid local search for the periodic capacitated arc routing problem," *European Journal of Operational Research*, vol. 264, pp. 55–65, 2018.
- [8] P. Lacomme, C. Prins, and W. Ramdane-Chérif, "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... pp.1–8, 2002. Annecy, France: ESIA-University of Savoie.
- [9] F. Chu, N. Labadi and C. Prins, "A scatter search for the periodic capacitated arc routing problem," *European Journal of Operational Research*, vol. 169, pp. 586–605, 2006.
- [10] F. Marzolf, M. Trépanier and A. Langevin, "Road network monitoring: algorithms and a case study," *Computers & Operations Research*, vol. 33, no. 12, pp. 3494–3507, 2006.
- [11] J. P. Riquelme-Rodríguez, A. Langevin and M. Gamache, "Adaptive large neighborhood search for the periodic capacitated arc routing problem with inventory constraints," *Network*, vol. 64, pp. 125–139, 2014.
- [12] J. P. Riquelme-Rodríguez, A. Langevin and M. Gamache, "Location arc routing problem with inventory constraints," *Computers & Operations Research*, v. 76, p. 84–94, 2016.
- [13] G. V. Batista, "Proposta de um modelo matemático para o problema de roteamento em arcos capacitado e periódico," dissertação, Mestrado em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, Brasil, 2014.
- [14] G. V. Batista and C. T. Scarpin "Problema de roteamento em arcos capacitado e periódico aplicado a um contexto real," *Revista Produção Online*, Florianópolis, vol. 15, no. 3, pp. 1080–1098, set. 2015.
- [15] S. H. Huang and T. H. Lin, "Using ant colony optimization to solve periodic arc routing problem with refill points," *Journal of Industrial and Production Engineering*, vol. 31, no. 7, pp. 441–451, 2014.
- [16] V. O. Silva, P. Y. Ekel and C. A. P. S. Martins, "An efficient parallel implementation of an optimized simplex method in GPU-CUDA," *IEEE Latin America Transactions*, vol. 16, no. 2, pp. 564–573, Feb. 2018.
- [17] F. Chu, N. Labadi and C. Prins, "The periodic capacitated arc routing problem linear programming model, metaheuristic and lower bounds," *Journal of Systems Science and Systems Engineering*, vol. 13, no. 4, pp. 423–435, 2004.
- [18] F. Chu, N. Labadi and C. Prins, "Heuristics for the periodic capacitated arc routing problem," *Journal of Intelligent Manufacturing*, vol. 16, no. 2, pp. 243–251, 2005.
- [19] P. H. S. Palhares and L. C. Brito, "Constrained mixed integer programming solver based on the compact genetic algorithm," *IEEE Latin America Transactions*, vol. 16, no. 3, pp. 1493–1498, May. 2018.
- [20] A. Kansou and A. Yassine, "Ant colony for the periodic capacitated arc routing problem," *Journal of Industrial and Production Engineering*, 2009.
- [21] Y. Mei, K. Tang and X. Yao, "A memetic algorithm for periodic capacitated arc routing problem," *IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 41, no. 6, pp. 1654–1667, 2011.
- [22] L. A. Wolsey, "Wiley-Interscience series in discrete mathematics and optimization," *Integer Programming*, John Wiley & Sons, 1998.
- [23] G. Belvaux and L. A. Wolsey, "bc-prod: A specialized branch-and-cut system for lot-sizing problems," *Management Science*, vol. 46, no.5, pp. 724–738, 2000.
- [24] C. Mercé and G. Fontan, "MIP-based heuristics for capacitated lotsizing problems," *Int. J. Production Economics*, vol. 85, pp. 97–111, 2003.
- [25] D. Ferreira, R. Morabito and S. Rangel, "Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants," *Computers & Operations Research*, vol. 37, no. 4, pp. 684–691, 2010.
- [26] T. Wu, L. Shi and J. Song, "Na MIP-based interval heuristic for the capacitated multi-level lot sizing problem with setup times," *Annals of Operations Research*, vol. 196, no. 1, pp. 65–650, 2011.
- [27] L. Guimarães, D. Klabjan and B. Almada-Lobo, "Princing, relaxing and fixing under lot sizing and scheduling," *European Journal of Operational Research*, vol. 230, pp. 399–411, 2013.
- [28] B. B. Oliveira, M. A. Carravilla, J. F. Oliveira and F. M. B. Toledo, "A relax-and-fix based algorithm for the vehicle-reservation assignment problem in car Rental Company," *European Journal of Operational Research*, vol. 237, pp. 729–737, 2014.
- [29] L. Oliveira, C. C. Souza and T. Yunes, "Improved bounds for the traveling umpire problem: A stronger formulation and relax-and-fix heuristic," *European Journal of Operational Research*, vol. 236, pp. 592–600, 2014.
- [30] Y. Pochet and L. A. WOLSEY, "Production planning by mixed integer programming," Science Business Media, Inc, New York, Springer, 2006.
- [31] L. F. Escudero, J. Salmeron, "On a fix-and-relax framework for a class of project scheduling problems," *Operations Research*, no. 140, pp. 163–188, 2005.
- [32] B. L. Golden and J. S. Dearmon, and E. K. Baker, "Computational experiments with algorithms for a class of routing problems," *Computers & Operations Research*, vol. 10, no. 1, pp. 47–59, 1983.



Jailson Domingos de Oliveira é Mestre (2017) em Pesquisa Operacional pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, área de concentração: Programação Matemática, na Universidade Federal do Paraná; Licenciado em Matemática (2010) pela UNICENTRO. Atualmente é estudante de Doutorado no Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da UFPR e Professor substituto na UNICENTRO.



Cassius Tadeu Scarpin é Doutor (2012) e Mestre (2007) em Pesquisa Operacional pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, área de concentração: Programação Matemática, na Universidade Federal do Paraná. Engenheiro de Produção (2010) e Licenciado em Matemática (2002) pela mesma universidade. Atualmente ocupa o cargo de Professor Adjunto no Departamento de Administração Geral e Aplicada (DAGA) da Universidade Federal do Paraná (UFPR) e coordena o Grupo de Pesquisa de Tecnologia Aplicada à Otimização (GTAO).