Heurística Relax-and-Fix Aplicada ao Problema de Roteamento Periódico em Arcos Capacitado

Jailson Domingos de Oliveira
PPGMN/ Departamento de Matemática
UFPR/UNICENTRO
Curitiba, Brasil
jailson-de-oliveira@hotmail.com

Cassius Tadeu Scarpin
Departamento de Administração Geral e Aplicada
UFPR
Curitiba, Brasil

***Resumo*— O presente trabalho trata do problema de roteamento periódico em arcos capacitado (PCARP). Considerou-se o caso especial onde os veículos não têm a necessidade de voltar ao depósito no final de um dia e, ainda, têm a possibilidade de folgar em qualquer dia do horizonte de tempo [6]. O PCARP é um problema pouco explorado na literatura. Classificado como um problema NP-hard, sendo comumente aplicado em coleta de lixo urbano, inspeção de linhas de força despejo de sal em vias com neves, entrega de correspondência entre outros [7]-[8]. Nesse trabalho, desenvolve-se métodos de solução do tipo *relax-and-fix*, foram propostas 4 estratégias diferentes para a heurística e avaliado o seu desempenho para determinar soluções para o PCARP. Os testes computacionais realizados mostram que a heurística proposta pode ser um método rápido para determinar soluções iniciais para o problema.**

Palavras-chave*—*Relax-and-fix; Problema de Roteamento Periódico em Arcos Capacitado ; Método Exato.

I. Introdução

Avanços tecnológicos, crescimento populacional e mudanças no cenário econômico com aumento da oferta e demanda de produtos, acarretaram em desafios às empresas, em especial no ramo logístico, conduzindo a revisão constante de seus modelos de distribuições cada vez mais complexas. Dessa forma, os processos logísticos se tornaram importantes nas operações gerenciais fazendo com que a logística consolide-se como elemento chave nas estratégias competitivas das empresas [1].

Segundo a Fundação Dom Cabral[1] os resultados da pesquisa Custo Logístico no Brasil em 2015, cujo objetivo é avaliar os custos logísticos para as empresas, revelou que, das 142 empresas participantes (que correspondem a 15% do PIB brasileiro), 11,73% de sua receita é consumida pelo custo logístico. É estimado que os custos de distribuição somam quase a metade dos custos totais logísticos, podendo chegar em alguns casos em 70% do valor como, por exemplo, no caso das indústrias de alimentos e bebidas [2].

A procura por otimização desses custos acarretou no aumento de pesquisas e artigos publicados. A literatura traz estudos em três grandes problemas logísticos: gerenciamento de armazéns, localização de facilidades e roteamento de veículos. Nesse último, em especial, a literatura atual sugere dois objetivos explorados: a redução do número de veículos utilizados e/ou a redução da distância total percorrida [3].

O Problema de Roteamento de Veículos (PRV) ou *Vehicle Routing Problems* (VRP), é um problema clássico da literatura [4]. O PRV tem por objetivo determinar a melhor rota para uma frota de veículos partindo de um depósito central para atender um conjunto de usuários finais (clientes). O problema tem sido amplamente estudado devido sua larga aplicação em problemas reais e sua complexidade, pertencendo a classe de problemas *NP-Hard* e, portanto, não há algoritmos em tempo polinomial para encontrar soluções ótimas.

O clássico Problema de Roteamento de Veículos (PRV) pode apresentar algumas variações, que é o caso do Problema de Roteamento em Arcos, conhecido pela sua sigla em inglês ARP (*Arc Routing Problem*), que se refere a determinação de rotas para veículos de forma a atender demandas de clientes ao longo dos arcos. Os ARP's são similares aos PRV, exceto que as demandas são distribuídas pelos arcos e não pelos nós. Existem um grande número de situações reais que podem ser modelados por esse tipo de problema tais como: a coleta de lixo urbano, a inspeção de linhas de forças, o despejo de sal em vias com neves e entrega de correspondências entre outros [5].

O problema das pontes de *Königsberg*, o Problema do Carteiro Chinês, o Problema do Carteiro Rural são exemplos de problemas que se encaixam nessa classificação ARP's. Temos ainda o Problema de Roteamento em Arcos Capacitado - *Capacitated Arc Routing Problem* (CARP), que é caracterizado por ter uma demanda associada a cada arco, sendo utilizados para tomadas de decisões de nível operacional, tomadas diariamente [6].

Ainda nessa categoria de problemas de roteamento de arcos encontra-se o Problema de Roteamento Periódico em Arcos Capacitado- *Periodic Capacitated Arc Routing Problem* (PCARP), que são uma extensão natural do CARP, ao invés do problema ser resolvido para tomada de decisões de nível operacional são utilizados para decisões de nível tático, para horizontes de tempo maiores com restrições de frequência [7].

Em muitos problemas, alguns arcos possuem uma necessidade de serem atendidos com maior frequência. Por exemplo, no caso da coleta de lixo que em grandes centros a produção é maior do que em bairros. Logo, essas regiões demandam uma frequência maior de coleta, em que se faz

[1]ttp://www.fdc.org.br/blogespacodialogo/Lists/Postagens/Post.aspx?List=95696fb1-15d4-444c-9a1e-506231d17962&ID=482&Web=e06f7d1a-c7ed-49ae-95f1-a5c1408f0875

necessário um planejamento baseado em múltiplos períodos. Portanto, resolver PCARP implica a determinação simultânea de decisões táticas: determinando para cada rua um número de tratamento de acordo com sua frequência; e operacionais: determinar uma rota de forma que, para cada período, as ruas sejam atendidas. O custo total depende da combinação das decisões de designação e roteamento [8].

Em estudo [9] tratando do roteamento de veículos que despejam sal em vias, apesar do problema não ter sido nomeado de PCARP, apresenta características desse tipo de problema, por apresentar intervalos de atendimento das vias [9]. O autor resolve o problema dividindo em duas fases, resolve o Problema do Carteiro Chinês, e para melhorias utiliza metaheurística *Simulated Anneling*. Em outra pesquisa [10], é apresentado o PCARP como extensão do CARP, e aplicam dois algoritmos para resolução. O primeiro é uma heurística gulosa e o segundo é o algoritmo *Scatter Search* (SS) baseado em busca local, utilizados para avaliar dois conjuntos de casos. Em [6] apresenta um modelo com características diferenciadas das já propostas na literatura, como não necessitar voltar a um depósito ao final de um dia e a possibilidade de um veículo folgar em um determinado dia. Seu modelo foi aplicado na manutenção preditiva e preventiva de linhas férreas, usou o *software* comercial CPLEX 12.4 para resolver, conseguindo chegar na solução ótima em apenas uma das 23 instâncias proposta pelo autor.

O PCARP é um problema pouco explorado na literatura e é um problema *NP-hard* uma vez que inclui CARP como caso particular e, quanto mais restrições mais difícil é encontrar uma solução [7]. Logo, métodos exatos para sua solução são, até o momento na literatura, ineficientes quando trabalhado com problemas reais devido sua complexidade. O que leva ao desenvolvimento e proposta de novos métodos para determinar boas soluções para o PCARP.

Nesse trabalho, considerou-se o problema de roteamento periódico de arcos capacitado no qual um carro não tem a necessidade de voltar ao depósito no fim de um dia e, tendo a possibilidade de folgar em um determinado dia. Baseado na formulação matemática proposta em [6]. Como abordagem de solução, utilizou-se heurística do tipo *relax-and-fix* que decompõem o modelo original em submodelos que podem ser resolvidos de modo exato. Além de propor diferentes estratégias para heurística, esse trabalho procurou contribuir com a literatura verificando a viabilidade e desempenho da heurística quando aplicada no PCARP.

O trabalho está organizado em seções. Na Seção 2 é apresentado a formulação matemática para o PCARP. Na Seção 3 é apresentado o método de solução, ou seja, a heurística *relax-and-fix* e algumas estratégias propostas. Na seção 4, os experimentos computacionais realizados são descritos. E por fim, as conclusões e sugestões de trabalhos futuros estão na Seção 5.

II. Formulação Matemática

O modelo proposto em [6], parte de um grafo não direcionado $G=(X,E)$ com n pontos, $X=\{x_1,x_2,\dots,x_n\}$, e m arestas com $E=\{e_1,e_2,e_3,\dots,e_m\}$ 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 chamado de $x_{ij}=(x_i,x_j)$, os quais são associado a um custo c_{ij}.

A referência [1] considerou a capacidade dos veículos usados, como sendo a máxima distância que estes podem se deslocar, ou seja, 1 arco 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 nesse modelo é considerado a periodicidade de cada arco que é expressa na quantidade máxima de períodos em que um arco deve ser atendido ao menos uma vez $MP(x_{ij})$.

O modelo é baseado em Programação Linear Binária, com três variáveis descritas abaixo:

$$x_{ijkp}\begin{cases}1, & \textit{se o carro } k \textit{ se desloca do ponto } i \textit{ para o } j \textit{ no período } p\\ 0, & \textit{caso contrario}\end{cases}$$

$$p_{ijp}\begin{cases}1, & \textit{se o arco } x_{ij} \text{ não } \textit{respeita a periodicidade no período } p\\ 0, & \textit{caso contrario}\end{cases}$$

$$f_{ikp}\begin{cases}1, & \textit{se o carro } k \textit{ fica parado no ponto } i \textit{ no período } p\\ 0, & \textit{caso contrario}\end{cases}$$

As variáveis p_{ijp} e f_{ikp} são propostas para que o modelo tenha viabilidade. Cada variável p_{ijp} está associada a uma punição PU_{ij}, caso um arco não tenha sua periodicidade atendida o que permite que haja atrasos no atendimento de alguns arcos.

Já a variável f_{ikp} permite que um carro folgue em um determinado dia, permitindo em algumas vezes obter uma solução com menor quantidade de deslocamentos.

Modelo:

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

Sujeito à:

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

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

$$\sum_{k=1}^{nk}\begin{pmatrix}x_{ijkp+}x_{jikp}+x_{ijk,p+1}+x_{jik,p+1}+\cdots\\ +x_{ijk,p+MP(x_{ij})-1}+x_{jik,p+MP(x_{ij})-1}\end{pmatrix}+p_{ijp}\geq 1$$
$$\forall[i,j]\in E,\forall p\in H \quad (3)$$

$$x_{ijkp},p_{ijp},f_{ikp}\in\{0,1\} \quad \forall[i,j]\in E,\forall k\in K,\forall p\in H \quad (4)$$

O objetivo do modelo (1) é a minimização dos custos de deslocamentos e atrasos. Restrição (2) garante o fluxo diário dos carros permitindo folgar. Pode ser observado nessa restrição, se um carro que chega em ponto em um determinado dia somando a possibilidade do veículo já estar no ponto, deve ser igual ao movimento no dia seguinte ou permanência no mesmo.

Restrição (3) garante que todos os carros terão uma designação para cada dia p, isto é, que cada veículo no dia p pode se deslocar de i para j, ou de j para i, ou ainda permanecer parado em um ponto i.

A restrição (4) se refere à periodicidade que deve ser atendida em cada arco. Observa-se que se as variáveis p_{ijp} assumam valor 1, indica que o arco e_{ij} teve sua periodicidade atrasada em um dia implicando em uma punição PU_{ij} na função objetivo.

Por fim a restrição (5) determina que todas as variáveis do modelo sejam binárias.

III. Heurística Relax-and-Fix

A heurística *relax-and-fix* tem sido aplicada nas últimas décadas para solucionar problemas de otimização combinatória, em especial, problemas de dimensionamento de lotes [11]-[12]-[13]. A heurística foi desenvolvida por [14], consiste em um método de decomposição de um modelo de programação inteira mista em submodelos menores disjuntos, que podem ser resolvidos rapidamente, porém sem a garantia de resolução do problema original de forma ótima.

A heurística baseia-se na partição das variáveis inteiras ou binárias do modelo criando n conjuntos distintos, $P_i, i = 1, \dots n$. Observa-se que o conjunto de variáveis originais são divididos em três grupos: um contendo variáveis inteiras e binárias, um segundo contendo as variáveis relaxadas, e o último com variáveis fixadas. O número n indica o número de iterações da heurística.

Em uma iteração i, as variáveis do conjunto P_i são definidas como inteiras e binárias, as variáveis (1,..., $i-1$) são fixadas e as demais variáveis $(i+1, \dots, n)$ são relaxadas. O submodelo resultante é resolvido. Caso o submodelo seja infactível, então se para a metodologia, pois não pode ser encontrado uma solução viável para o modelo original, com as variáveis (1, ..., $i-1$) fixadas. Caso contrário, as variáveis do conjunto P_i são fixadas, repete-se o processo para os demais conjuntos até que todas as variáveis $(i, \dots, n-1)$ sejam todas fixadas e as variáveis da partição P_n sejam inteiras ou binárias. Assim obtém-se uma solução inicial para o modelo original.

Segundo a literatura [11]-[13] a heurística *relax-and-fix* tem demonstrado bom desempenho para determinar soluções iniciais para modelos de dimensionamento de lotes. Esses resultados são posteriormente melhorados com a aplicação de uma heurística de melhoramento, como a heurística *fix-and-optimize*, por exemplo.

Os processos de escolha das partições das variáveis do problema afetam o desempenho da heurística, uma vez que seu objetivo é resolver submodelos inteiros mistos menores [15].

Nesse trabalho foram implementadas quatro estratégias diferentes para a heurística *relax-and-fix* e aplicadas no problema de roteamento periódico em arcos capacitado proposto [6]. Todas as estratégias propostas nesse trabalho são baseadas em partições por período. Descreve-se cada estratégia a seguir:

Relax-and-Fix Forward (RFF) - Nessa estratégia o problema original foi dividido em P subproblemas a serem resolvidos de modo que cada subproblema corresponda a um dia p do horizonte de tempo. A ordem da resolução é a cronológica, isto é, iniciando a resolução no subproblema correspondente ao primeiro período do horizonte de tempo e finalizando no subproblema correspondente ao último período.

A fig. 1 (a) ilustra o funcionamento dessa estratégia. Para o primeiro subproblema, primeira iteração da heurística, as variáveis binárias foram mantidas e as mesmas variáveis para os demais períodos foram relaxadas (valores no intervalo [0,1]). Após a resolução do primeiro subproblema, as variáveis binárias do primeiro período são fixadas, tornam-se parâmetros, as variáveis do segundo período ($p = 2$) são mantidas binárias, e as demais dos período $(3, \dots, p)$ são relaxadas. O processo se repete até que o último subproblema, seja resolvido.

Relax-and-fix Backward (RFB) - Essa estratégia também utiliza as partições baseadas em períodos, porém, ao contrário da estratégia anterior RFF, a resolução dos subproblemas ocorre em ordem cronológica inversa do último período do horizonte de tempo, até a última iteração correspondente ao primeiro subproblema.

Relax-and-fix com Overlapping 1 (RFO1) - Essa estratégia, se baseia na partição do problema original em P períodos com sobreposição de partições. Seja uma partição $p \geq 2$, as variáveis correspondente aos subproblemas p e $p-1$ são mantidas binárias e as demais variáveis subsequentes a esses períodos são relaxadas. Na próxima iteração, as variáveis do período $p-1$ são fixadas, as variáveis dos períodos p e $p+1$ mantidas binárias e as demais variáveis são relaxadas, o processo ocorre até que o último subproblema seja resolvido.

A fig. 1 (b) ilustra essa estratégia. Para o primeiro subproblema temos as variáveis correspondente ao primeiro e segundo período ($p = 1$ e $p = 2$) do horizonte de tempo, mantidas binárias e para os demais períodos foram relaxadas. Após a resolução desse primeiro subproblema, as variáveis binárias do primeiro período são fixadas (tornando-se parâmetro), as variáveis referentes ao segundo e terceiro período ($p = 2$ e $p = 3$) são mantidas binárias, e as demais variáveis são relaxadas. O processo ocorre da mesma forma até a resolução do último subproblema correspondente ao último período.

Relax-and-fix com Overlapping 2 (RFO2) – Essa estratégia, baseia-se na sobreposição de partições, semelhante a estratégia anterior RFO1. Nessa proposta cada subproblema é resolvido mantendo as variáveis binárias correspondentes aos períodos $p-3$, $p-2$, $p-1$ e p com $p \geq 3$, sendo as demais variáveis dos períodos subsequentes relaxadas. No próximo subproblema, as variáveis correspondentes aos períodos $p-3$ e $p-2$ são fixadas, as variáveis dos períodos $p-1$, p, $p+1$ e $p+2$ são mantidas binárias e as demais variáveis subsequentes relaxadas, e assim sucessivamente até o último subproblema ser resolvido.

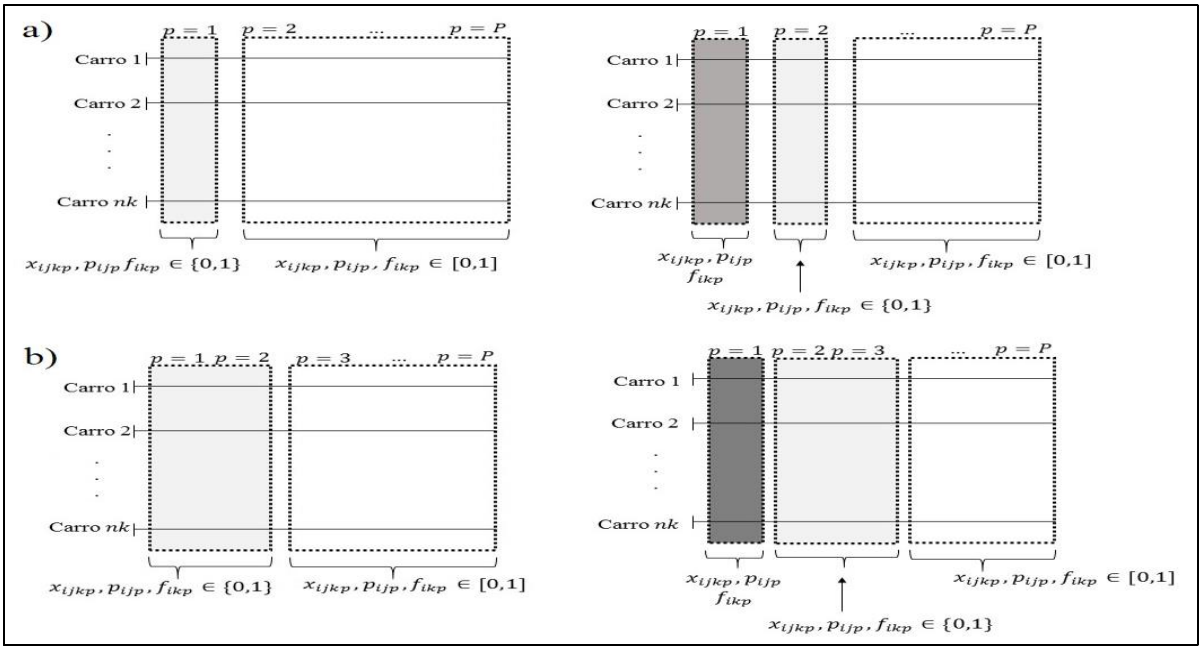

fig.1 Representação da heurística *relax-and-fix*.

IV. Experimentos Computacionais

A fim de verificar o desempenho da heurística *relax-and-fix* e as estratégias aqui propostas para o problema de roteamento periódico em arcos capacitado, foram utilizadas 23 instâncias propostas no trabalho [6]. As características para cada uma das instâncias, como número de nós, número arestas, quantidade de carros e o horizonte de tempo (Dias) estão dispostos na Tabela I. Percebe-se que as instâncias propostas tornam os problemas complexos de serem resolvidos.

TABELA I. Características dos 23 problemas.

Problema	Nós	Arestas	Período	Carros	Neces.	Dias
gdb1	12	22	2	1	32	38
gdb2	12	26	3	1	44	53
gdb3	12	22	2	2	31	19
gdb4	11	19	3	2	34	21
gdb5	13	26	4	2	53	32
gdb6	12	22	2	3	32	14
gdb7	12	22	3	3	37	45
gdb8	27	46	4	3	67	27
gdb9	27	51	3	4	78	24
gdb10	12	25	4	4	70	21
gdb11	22	45	2	1	50	60
gdb12	13	23	3	1	33	40
gdb13	10	28	2	2	33	20
gdb14	7	21	3	2	32	20
gdb15	7	21	4	2	48	29
gdb16	8	28	2	3	40	24
gdb17	8	28	3	3	52	21
gdb18	9	36	4	3	81	33
gdb19	8	11	3	4	29	12
gdb20	11	22	4	4	53	16
gdb21	11	33	2	5	46	12
gdb22	11	44	4	5	88	22
gdb23	11	55	2	6	82	17

As heurísticas propostas foram implementadas no Microsoft Visual Studio 2015 utilizando pacote de otimização IBM ILOG CPLEX 12.4. Os testes foram realizados utilizando um computador com processador Intel Core i7-5500U com velocidade de 2.4GHz, memória de RAM de 8GB e 1TB de HDD sob o sistema operacional Windows 10.

A Tabela II a seguir mostra o valor da função objetivo e tempo gasto em segundos, na execução de cada heurística propostas para cada uma das 23 instâncias, foram estipulados um tempo limite de 1 hora (h) ou 3600 segundos (s) para cada estratégia. Os símbolos "--" indicam que a estratégia gerou subproblemas infactível, enquanto que os símbolos "*" que a heurística utilizou o tempo limite e não chegou em uma solução.

A última coluna da tabela refere-se aos valores da função objetivo de cada instância, as quais foram resolvidas pelo *software* CPLEX 12.4 utilizando a configuração default do solver, para o qual foi estabelecido um tempo limite para obtenção de uma solução de 24 horas ou 86400 segundos. Os símbolos "**" indicam que os problemas esgotaram a memória do computador antes de chegar a uma solução ótima ou atingir o tempo limite. Observa-se que o método exato chegou na solução ótima apenas para instância gdb14, para as demais utilizou o tempo estabelecido ou esgotaram a memória do computador.

Observa-se que a estratégia RFF apresentou o pior desempenho em relação as demais, das 23 instâncias sugeridas 19 dessas foram infactíveis. A infactibilidade ocorreu com grande frequência em todas as estratégias propostas. Esse fato ocorre devido ao funcionamento da heurística *relax-and-fix*, uma vez que, o modelo proposto exige que cada carro volte ao nó de origem no final do horizonte de tempo e, como a heurística fixa período por período a cada iteração, incide que, nas últimas iterações um determinado carro encontra-se em um nó muito distante de seu nó de origem, não havendo dias suficientes no

TABELA II. Resultados das estratégias propostas para heurística *relax-and-fix*

	RFF		RFB		RFO1		RFO2		EXATO	
Problema	***Solu.***	***Tempo***	***Solu.***	***Tempo***	***Solu.***	***Tempo***	***Solu.***	***Tempo***	***Solu.***	***Tempo***
gdb1	--	--	--		--		--		4450	39000**
gdb2	--	--	--		--		--		11622	70423**
gdb3	--	--	--		--		--		340	36906**
gdb4	--	--	--		--		14503		3555	49595**
gdb5	--	--	--		--		--		1750	86400*
gdb6	--	--	--		4478	260	459		456	86400*
gdb7	--	--	--		--		15520		518	55767**
gdb8	--	--	--		--		--		12395	86400*
gdb9		*	--	*	--	*	--	*	13454	86400*
gdb10		*	--	*	--	*	--	*	734	86400*
gdb11	--	--	--		--		--		64437	31510**
gdb12	--	--	--		--		--		536	86400*
gdb13	--	--	--		620	111	620	111	620	86400*
gdb14	--	--	--		--		145	105	145	261
gdb15	--	--	--		--		3144	1681	140	63135**
gdb16	--	--	182	136	185	182	184	145	182	35301**
gdb17	--	--	--	--	3180	305	--		168	77693**
gdb18	--	--	--	--	--		--	*	415	86400*
gdb19	2135	122	135	145	129	145	135	125	125	47566**
gdb20	--	--	--	--	--		6258		2280	67903**
gdb21	--	--	--	--	--		212	122	212	38181**
gdb22	--	*		*	--	*	--	*	433	86400*
gdb23	--	--	--	--	--		336	1432	336	86400*

horizonte de tempo para seu retorno ao nó inicial, o que torna infactível o problema.

A estratégia RFB resolveu duas das 23 instâncias, seu desempenho foi melhor que RFF, porém, abaixo do esperado. Já a estratégia RFO1 apresentou desempenho superior as duas primeiras estratégias, conseguiu resolver 5 das 23 instâncias, chegou no mesmo valor para a função objetivo do problema gdb13 que o método exato, com um tempo menor, enquanto que o método exato levou 24 horas a heurística levou 111 segundos (1 minuto e 51 segundos).

A última estratégia proposta nesse trabalho RFO2, obteve o melhor desempenho, chegou em uma solução para 11 dos 23 problemas, sendo que para as instâncias gdb13, gdb14, gdb21 e gdb23 obteve o mesmo valor que o método exato para a função objetivo, em um tempo de execução inferior ao exato. É possível observar ainda que para as instâncias gdb6, gdb16 e gdb19 a heurística teve bom desempenho chegando em um valor para função objetivo próxima do método exato e, novamente, com um tempo muito inferior a esse. Esses valores podem posteriormente ser melhorados com a aplicação de uma heurística de melhoria como, por exemplo, a heurística *fix-and-optimize*.

V. CONCLUSÕES

Nesse trabalho estudou-se o problema de roteamento periódico em arcos capacitado (PCARP). Foi realizado estudos baseado no modelo [1] e desenvolvido métodos de solução do tipo *relax-and-fix*. Foram implementadas estratégias de decomposição das variáveis binárias do problema e realizado testes com as 23 instâncias propostas na literatura essas com diferentes características números de nós, número de dias, número de veículos e periodicidades, o que torna o problema complexo, fato esperado devido à natureza do problema *NP-hard.*

Nos testes, verificou-se que a heurística *relax-and-fix* é uma boa estratégia para determinar uma solução inicial para o problema. Em alguns casos, os valores da função objetivo podem ficar distantes da solução ótima, o que pode ser futuramente melhorada com uma heurística de melhoria. Conforme analisados os resultados obtidos, verificou-se que a melhor heurística proposta foi a *Relax-and-Fix Overlapping 2*, sendo a que apresentou maior número de soluções comparada com as demais. Porém, não conseguiu chegar em uma solução para as 23 instâncias.

Futuramente, pretende-se estudar e implementar novas estratégias para heurística *relax-and-fix*, baseando-se no particionamento do problema de acordo com o número de carros, ou seja, fixando as variáveis que envolvem o primeiro carro como binário e relaxando para os demais carros. Na segunda iteração fixa as que envolvem o carro 1 (parâmetro), as do segundo carro como binárias e relaxando as variáveis para os demais, continuando o processo até a fixação das variáveis que envolvem o último veículo. Implementar uma heurística de melhoria do tipo *fix-and-optimize*, baseada no particionamento do conjunto de variáveis em subconjunto disjuntos, os quais estão fixados e serão liberados para otimização a cada iteração, objetivando a melhoria das soluções encontradas pelo *relax-and-fix*.

AGRADECIMENTOS

Agradecemos a CAPES REUNI, pelo apoio financeiro.

REFERÊNCIAS

[1] V.P. Heloisa, "Metaheuristica para a Solução de Problemas de Roteamento de Veículos com Janela de Tempo. Dissertação de Mestrado," UNICAMP. 2013. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0166218X92000035>. Acesso em: 02/02/2016.

[2] P.L. Dilson, "Heurísticas e Algoritmo Exato para o Problema de Roteamento de Veículos dom Coleta e Entrega Simultâneas," Dissertação de Mestrado. UFMG. 2010 .Disponível em: <https://www.dcc.ufmg.br/pos/cursos/defesas/1209M.PDF>. Acesso em: 02/02/2016

[3] D.H.H. Seyed and S. Abbas, "Lower and upper bounds for location-arc routing problems with vehicle capacity constraints," European Journal of Operational Research, v. 224, n. 1, p. 189–208, 2013. Elsevier B.V. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0377221712004705>. Acesso em: 02/02/2016.

[4] D.B. George and J.H. Ramser, "The Truck Dispatching Problem," Management Science, 6,p.80-91, 1959.

[5] C. Angel and P. Christian, "Recent Results on Arc Routing Problems : Na Annotated Bibliography," Networks, p. 50–69, 2010.

[6] V.B. Guilherme, "Proposta de Um Modelo Matemático para o Problema de Roteamento em Arcos Capacitado e Periódico," Dissertação de Mestrado. UFPR. 2014. Disponível em: <http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/35767/R%20-%20D%20-%20GUILHERME%20VINICYUS%20BATISTA.pdf?sequence=1>. Acesso em: 25/11/2015.

[7] L. Philippe, P. Christian and R.C Wahiba, "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.

[8] L. Philippe, P. Christian and R.C Wahiba, "Evolutionary algorithms for periodic arc routing problems," European Journal of Operational Research, v. 165, p. 535–553, 2005.

[9] R.W Eglese, "Routeing winter gritting vehicles. Discrete Applied Mathematics," v.48, n. 3, p. 231–244, 1994. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0166218X92000035>. Acesso em: 13/02/2016.

[10] C. Feng, L. Nacima and P. Christian, "A Scatter Search for the periodic capacitated arc routing problem," European Journal of Operational Research, vol. 169, p. 586–605, 2006.

[11] G.D. Satyaki and N. Raskesh, "Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs," Computers & Operations Research, 32, pp. 2987-3005, 2005.

[12] F. Deisemara, M. Reinaldo and R. Socorro, "Solution approaches for the soft drink integrated production lot sizing and scheduling problem," European Journal of Operational Research, 196, pp. 697-706, 2009.

[13] F. Deisemara, M. Reinaldo and R. Socorro, "Relax and fix heuristics to solve one-stage onemachine lot-scheduling models for small-scale soft drink plants," Computers & Operations Research, 37, 684–691, 2010.

[14] W.A. Laurence, "Wiley-Interscience Series in Discrete Mathematics and Optimization," Integer Programming, Wiley. 1998.

[15] M.C.C. Lívia and S.O. Maristela, "Heurística Relax-and-Fix para o Problema de Dimensionamento de Lotes com Janelas de Tempo de Produção," 2012. Disponível em: <http://www.din.uem.br/sbpo/sbpo2012/pdf/arq0479.pdf>.