

An ILS algorithm with RVND for the green vehicle routing problems with time-varying speeds

Darci José Mendes Júnior, Luciana Brugiolo Gonçalves, Stênio Sã R. F. Soares

¹Departamento de Ciência da Computação
- Universidade Federal de Juiz de Fora - UFJF

darcimendes@ice.ufjf.br, lbrugiolo@ice.ufjf.br, ssoares@ice.ufjf.br

Abstract. *The environmental impacts of human action have led several countries to create stricter laws and tax breaks to reduce this damage. Thereby, the Green Logistic has been increasingly sought to meet the requirements and needs for a more sustainable development. This work presents an ILS (Iterated Local Search) algorithm combined with RVND (Random Variable Neighborhood Search) and compare it with a GRASP (Greedy Randomized Search Procedure) algorithm where each one has two variations: minimize distance and minimize emission. The results show the effectiveness of the ILS approach and heuristics that minimize the total distance covered do not present themselves as good solutions in terms of sustainability.*

Resumo. *Os impactos ambientais da ação humana levaram diversos países a criarem leis mais rígidas e incentivos fiscais para reduzir esse dano. Assim, a Logística Verde tem sido cada vez mais buscada para atender as exigências por um desenvolvimento mais sustentável. Este trabalho apresenta um algoritmo ILS (Iterated Local Search) combinado com RVND (Random Variable Neighborhood Search) e o compara com um algoritmo GRASP (Greedy Randomized Adaptive Search Procedure) onde cada um possui duas variações: minimizar distância e minimizar emissão. Os resultados mostram a eficácia da abordagem ILS e que heurísticas que minimizam a distância total percorrida não apresentam soluções sustentáveis.*

1. Introdução

A logística tem como principal objetivo descrever o transporte, manuseio e armazenamento de produtos a fim de maximizar a rentabilidade e minimizar seus custos. Nos últimos anos, a preocupação com o impacto ambiental dessas atividades tem crescido [McKinnon et al. 2010], levando à mudanças na legislação e no perfil do cliente, que agora considera a responsabilidade ambiental das empresas antes de decidir sobre diferentes marcas. Seja por força de novas leis, por incentivos fiscais ou por estratégia de marketing, esta nova realidade fez surgir a Logística Verde. Além disso, o impacto negativo da poluição atmosférica sobre a saúde é motivo suficiente para justificar a relevância de trabalhos cujo o foco seja a redução da emissão de gases poluentes [Rao et al. 2013].

A Logística Verde tem um papel essencial quando se considera problemas relacionados ao transporte, processos de reciclagem, controle de energia e emissão de poluentes. Esta relevância da área pode ser verificada pelo aumento no número de trabalhos publicados sobre o tema [Lin et al. 2014]. Em [Sbihi and Eglese 2010] os autores apresentaram

diversos problemas de Otimização associados à Logística Verde. A partir de então, o aumento no interesse pela área se intensificou e é influenciado pelo apelo crescente sobre a importância de questões relacionadas ao impacto ambiental [Picelli and Georges 2011].

Os aspectos ambientais considerados, que levam o problema de roteamento ao escopo da Logística Verde, denotado Roteamento Verde de Veículos, estão relacionados à redução de emissão de poluentes. O problema considera uma frota de veículos cujas velocidades em cada trecho podem variar, o que define o Problema de Roteamento Verde de Veículos com Velocidades Variáveis no Tempo. O objetivo deste problema é minimizar a emissão de gases poluentes provenientes da queima de combustível.

Dado um horizonte de planejamento, definido para um dia de viagem, considera-se ao longo do dia períodos iguais de tempo que influenciam na velocidade que os veículos podem alcançar, e o cálculo da quantidade de poluentes emitidos depende diretamente dessa velocidade. O problema estudado neste trabalho é o mesmo apresentado por [Qian and Eglese 2016], que utiliza uma estratégia que envolve geração de colunas baseada num algoritmo de busca Tabu.

Outras variantes também têm recebido atenção dos pesquisadores da área. O Problema de Roteamento abordado por [Molina et al. 2014] é definido sobre uma frota heterogênea onde variam a capacidade de carga e o tipo de combustível dos veículos. Além disso, os clientes possuem janela de tempo e demanda. Os autores consideraram três objetivos possíveis: minimizar o custo interno, a emissão de CO_2 e a emissão de NO_x . O cálculo da emissão de CO_2 está relacionado à distância percorrida e ao consumo de combustível, que também sofre influência da carga. O cálculo da emissão de NO_x está relacionado à distância percorrida e à tecnologia do veículo.

Neste trabalho é proposta uma abordagem baseada na metaheurística *Iterated Local Search* (ILS) para resolver o problema. Para isso foram desenvolvidas heurísticas construtivas e de busca local que serão combinadas na construção do ILS. Com o intuito de avaliar o impacto de se considerar como objetivo a minimização da emissão de gases poluentes, uma versão do algoritmo considerando a minimização da distância também é apresentada.

Na próxima seção é apresentada uma descrição detalhada do problema e de como é feito o cálculo da emissão dos gases e da distância. Na Seção 3 serão descritos o algoritmo construtivo, as heurísticas de busca local e as abordagens ILS e RVND. Na Seção 4 são apresentados como as instâncias foram geradas e os resultados e na Seção 5 as conclusões e trabalhos futuros.

2. Problema de Roteamento Verde de Veículos com Velocidades Variáveis no Tempo

Este trabalho aborda o Problema de Roteamento Verde com Velocidades Variáveis no Tempo - PRV-VV. O objetivo deste problema é minimizar a emissão de gases poluentes provenientes da queima de combustível. Todos os clientes a serem visitados possuem demanda e janela de tempo.

Para descrever o problema, considere um grafo completo e direcionado $G = (V, A)$ onde $V = C \cup \{d_{central}\}$, sendo $C = \{c_1, c_2, \dots, c_n\}$, o conjunto de clientes e $d_{central}$ o depósito. Como este problema considera a variação de velocidade durante

o horizonte de tempo, para cada intervalo de tempo t_k , com k variando de 1 até m , há uma velocidade máxima permitida diferente para cada arco. Logo, A contém o conjunto de arcos, onde cada par de nós é ligado por m arcos, um para cada intervalo de tempo. Existe uma velocidade mínima que é a mesma para todos os arcos.

Cada cliente i possui uma demanda d_i e uma janela de tempo que inicia em JI_i e termina no tempo JF_i . A frota é homogênea, possui capacidade máxima de carga C_{max} e o número de veículos é ilimitado. Há também um tempo de espera máximo TE_{max} para os casos em que o veículo chegue antes do início da janela de tempo.

Assim como [Qian and Eglese 2016], foi considerada a emissão equivalente de CO_2 onde é feito primeiro o cálculo do consumo de combustível em litros por quilômetro, sendo que este consumo depende da velocidade. A emissão equivalente de CO_2 é obtida multiplicando o consumo por um fator de 3.1787 que é o fator proposto pelo Governo do Reino Unido em [Department for Environment, Food and Rural Affairs 2010], segundo [Qian and Eglese 2016]. Neste trabalho são consideradas a emissão de 5 diferentes gases. São estes CO , HC , NO_x , PM e CO_2 , sendo a cada um associado um índice s variando de 1 a 5, de acordo com a ordem em que foram apresentados. A emissão de cada um dos gases para o deslocamento do nó i até o nó j com velocidade $v_{i,j}$ é calculada a partir da Equação 1 onde P_s é a emissão em kg/km do gás s :

$$P_s = \frac{a_s + b_s v_{i,j} + c_s v_{i,j}^2 + d_s v_{i,j}^3 + e_s v_{i,j}^4 + f_s v_{i,j}^5 + g_s v_{i,j}^6}{v_{i,j}} \quad (1)$$

Na Tabela 1 são apresentadas as constantes a_s , b_s , c_s , d_s , e_s , f_s e g_s , segundo [P. G. Boulter and McCrae 2009], utilizadas para calcular a emissão individual de cada gás s . Essas constantes estão relacionadas ao veículo de código R166 que possui padrão de emissão EURO V, limite de peso maior que 32 toneladas e diesel como combustível.

Tabela 1. Constantes dos gases - adaptado de [P. G. Boulter and McCrae 2009].

Gás	a_s	b_s	c_s	d_s	e_s	f_s	g_s
CO	$2.0404E+0$	$4.0540E-1$	$-1.7566E-2$	$4.1924E-4$	$-5.7141E-6$	$4.4735E-8$	$-1.4931E-10$
HC	$3.594E-1$	$9.3573E-2$	$-5.3987E-3$	$1.4956E-4$	$-2.2094E-6$	$1.7133E-8$	$-5.4005E-11$
NO_x	$5.0793E+1$	$-1.1020E+0$	$3.3824E-1$	$-1.2620E-2$	$2.0982E-4$	$-1.5928E-6$	$4.5487E-9$
PM	$7.5519E-1$	$8.6426E-2$	$-4.3024E-3$	$1.1100E-4$	$-1.6094E-6$	$1.3039E-8$	$-4.4033E-11$
CO_2	$1.2690E+4$	$1.6564E+1$	$8.6867E+1$	$-3.5533E+0$	$6.1462E-2$	$-4.7730E-4$	$1.3853E-6$

A emissão total destes 5 gases produzidos pelo veículo quando transita do nó i para o nó j , P_{total} , é definida pelo somatório de P_s para s de 1 à 5 e é dada em kg/km (Equação 2). Para se chegar à emissão total em kg, P_{total} é multiplicado pela distância entre os nós $d_{i,j}$. Neste trabalho o peso da carga não é considerado no cálculo da emissão.

$$P_{total} = \sum_{s=1}^5 P_s \quad (2)$$

3. Abordagem Heurística ILS-RVND

Nesta seção serão descritas as heurísticas desenvolvidas para tratar o Problema de Roteamento Verde com Velocidades Variáveis no Tempo. A abordagem desenvolvida neste trabalho é baseada no ILS [Lourenço et al. 2010], uma metaheurística de simples implementação que combina construção, refinamento e perturbação. Para a fase de construção

foi considerado um algoritmo com múltiplos critérios onde cada critério muda o comportamento do algoritmo. Para a fase de refinamento foi utilizado um algoritmo baseado na metaheurística VND, considerando ordem aleatória de exploração das vizinhanças - RVND. A seguir são descritos os algoritmos de construção e de refinamento, bem como a estratégia geral baseada no ILS.

3.1. Algoritmos construtivos

O algoritmo construtivo constrói uma solução atribuindo valores às variáveis de decisão, uma por vez, até a solução estar completa [Talbi 2009]. O Algoritmo 1 descreve o algoritmo construtivo utilizado neste trabalho. Os parâmetros de entrada são: G , o grafo com as informações sobre a instância; α que indica o fator de aleatoriedade do algoritmo e três parâmetros que definem o comportamento do algoritmo, $critOrd$, $critCusto$ e $critInsercao$. O parâmetro $critOrd$ define o critério de ordenação dos nós clientes, $critCusto$ é o critério utilizado para calcular o custo de se inserir o nó na solução e $critInsercao$ o critério que determina quais posições em uma rota serão consideradas como posições candidatas para a inserção do próximo nó.

O primeiro passo do Algoritmo 1 é determinar o número mínimo de rotas necessárias para o atendimento de todos os clientes (linha 2). Para estimar o valor de $numRotas$ a soma das demandas dos clientes é dividida pela capacidade máxima do veículo C_{max} . Considerando que pelo menos $numRotas$ serão necessárias, o passo seguinte é a inicialização destas rotas, onde são criadas cada uma das rotas tendo como ponto inicial e final o depósito - linha 3.

Na linha 4, uma lista ordenada dos nós clientes é criada considerando o critério $critOrd$, parâmetro do algoritmo. Para o critério $critOrd$ duas estratégias diferentes de ordenação foram consideradas: J_{ini} e J_{dif} . A primeira estratégia, J_{ini} , prioriza os clientes que têm o início da janela de tempo mais cedo. Já o segundo critério, J_{dif} , prioriza os clientes com menor janela de tempo, ou seja, prioriza os clientes com menor diferença entre tempo de fim e tempo de início da janela de tempo.

O laço entre as linhas 5 e 27 é executado até que uma solução esteja completa. Na linha 7 é selecionado um nó da lista onde α é um fator de aleatoriedade e assume valores entre 0 e 1. Se α for 0 o nó selecionado é o melhor e se for 1 o nó será sorteado considerando toda a lista. Para valores intermediários será sorteado um nó entre os p melhores, onde p é o produto de α pela quantidade de nós que ainda não foram inseridos. Por exemplo, se ainda restarem 10 nós e α for 0.3, será sorteado um nó entre os 3 melhores.

O laço entre as linhas 9 e 15 serve para determinar em qual rota e posição na rota no será inserido de forma que o acréscimo do custo seja o menor possível. O tipo de custo é definido pela variável $critCusto$ que pode considerar um entre dois critérios: minimizar a emissão dos gases poluentes - P - ou minimizar a distância percorrida - D . A variável pos_{cand} recebe a posição onde no será inserido. Até que uma quantidade igual ao número mínimo de rotas possuam ao menos um cliente, uma rota sem cliente sempre será considerada como opção de inserção no laço.

Para o critério que determina quais posições serão consideradas como candidatas na inserção do novo nó em uma rota - $critInsercao$ -, foram desenvolvidas 3 opções. O critério 1 considera apenas o fim de cada rota como posição candidata e nenhum nó, com exceção do depósito, sofre alteração do horário de visita. O critério 2 considera tanto o

fim quanto o início. Quando a inserção é feita no início, o horário de saída do depósito pode ser alterado. O critério 3 além de considerar o início e o fim, considera também inserir entre dois nós consecutivos que já estão inseridos na rota. No critério 3, o horário de visita dos nós anteriores ao nó inserido não são alterados mas os horários, velocidades e tempos de espera dos nós posteriores são recalculados. No cálculo de velocidade sempre é considerada a maior possível, independente do critérios.

Algoritmo 1: CONSTRUTIVO

Entrada: G : Grafo, $critOrd$: int, $critCusto$: int, $critInsercao$: int, α :
double

Saída: S : Solução

```

1 início
2    $numRotas \leftarrow CalculaMinRotas(G)$ ;
3    $InicializaRotas(numRotas)$ ;
4    $lista \leftarrow OrdenaNos(G, critOrd)$ ;
5   repita
6      $no \leftarrow primeiroNoLista(lista, \alpha)$ ;
7      $custo_{melhorInsercao} \leftarrow \infty$ ;
8      $rota \leftarrow 1$ ;
9     repita
10       $custo_{cand} \leftarrow$ 
11         $custoInserirNo(no, rota, critCusto, critInsercao, pos_{cand})$ ;
12      se  $custo_{cand} < custo_{melhor}$  então
13         $custo_{melhor} \leftarrow custo_{cand}$ ;  $rota_{melhor} \leftarrow rota$ ;
14         $pos_{melhor} \leftarrow pos_{cand}$ ;
15      fim
16       $rota ++$ ;
17      até  $r \leq numRotas$ ;
18      se  $custo_{cand} \neq \infty$  então
19         $S \leftarrow inserirNo(no, rota_{cand}, pos_{melhor})$ ;
20      fim
21      senão
22         $numRotas ++$ ;
23         $rota_{cand} \leftarrow numRotas$ ;
24         $S \leftarrow inserirNo(no, rota_{cand})$ ;
25      fim
26       $removeNoLista(lista, no)$ ;
27      até  $lista \neq vazia$ ;
28 fim

```

Considerando os critérios de ordenação, custo e posições candidatas, foram definidas 12 combinações de critérios. A Tabela 2 mostra essas combinações. Testes realizados em [Mendes Júnior 2017] mostram que as combinações 3, 4, 9, 10, 11 e 12 apresentam os melhores resultados na fase de construção da solução inicial, considerando como função objetivo reduzir a emissão.

Tabela 2. Valores dos critérios e suas combinações

Comb.	Inserção	Custo	Ordenação	Comb.	Inserção	Custo	Ordenação
1	Fim	Poluição	Início	7	Fim_Início	Distância	Início
2	Fim	Poluição	Diferença	8	Fim_Início	Distância	Diferença
3	Fim	Distância	Início	9	Início_Meio_Fim	Poluição	Início
4	Fim	Distância	Diferença	10	Início_Meio_Fim	Poluição	Diferença
5	Fim_Início	Poluição	Início	11	Início_Meio_Fim	Distância	Início
6	Fim_Início	Poluição	Diferença	12	Início_Meio_Fim	Distância	Diferença

3.2. Buscas locais

Busca Local pode ser entendida como uma metaheurística que substitui, a cada iteração, a solução atual por um vizinho desde que o vizinho melhore a função objetivo. A busca termina quando não é encontrado nenhum vizinho melhor que a solução corrente [Talbi 2009]. Uma solução S' é vizinha de S se S' é gerada aplicando um movimento em S . A ideia do movimento é gerar uma perturbação em S para produzir uma solução diferente de S mas que possua algumas de suas características.

Foram desenvolvidas cinco Buscas Locais, cada uma implementada considerando duas diferentes formas para o cálculo da função objetivo: minimização da emissão e minimização da distância. Em todas as Buscas Locais o movimento é executado assim que uma solução melhor que a solução corrente for encontrada. A Figura 1 ilustra os movimentos considerados.

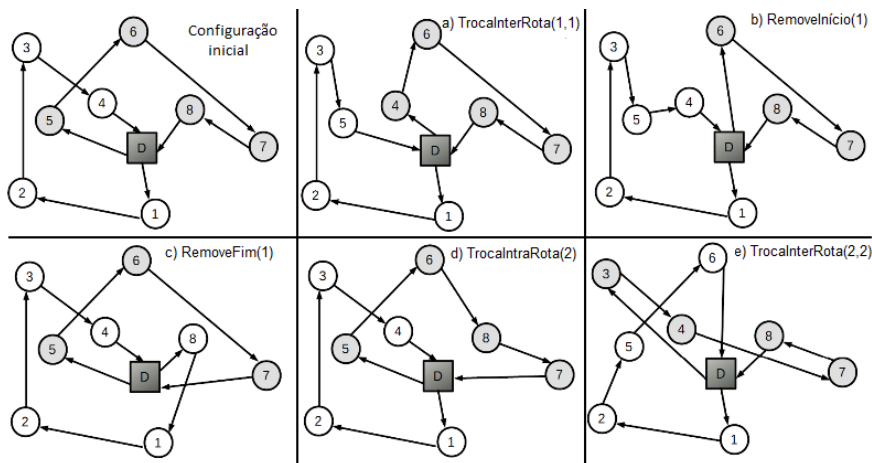


Figura 1. Exemplos dos movimentos das Busca Locais

Na Busca Local *TrocaInterRota(1,1)* o movimento realizado é a troca entre 2 nós de rotas diferentes, um de cada rota. Considerando a Figura 1 como exemplo, a troca acontece entre os nós 4 e 5. As velocidades consideradas para as arestas que ligam os nós 3 à 5 e D a 4 serão as maiores possíveis. Já a ligação de 4 a 6 deverá se adequar ao horário de visita anteriormente calculado do nó 6 para não chegar atrasado e nem adiantado. A aresta entre 5 e D não sofre do mesmo problema pois D é depósito.

Na Busca Local *RemoveInício(1)* o movimento realizado é a remoção do primeiro nó de uma rota e sua reinserção em outra rota. O critério para reinserção é o mesmo utilizado nas combinações de critérios 11 e 12. Na Figura 1 o nó removido é o 5, que é reinserido entre os nós 3 e 4. A velocidade da nova aresta que liga os nós D e 6 também

não deverá alterar nenhum horário de visita. A Busca Local *RemoveFim(1)* é semelhante a Busca Local *RemoveInício(1)*, mas o nó removido é o último nó da rota. Na Figura 1, o nó removido foi o 8 que foi reinserido entre os nós D e 1.

Na Busca Local *TrocaIntraRota(2)* o movimento realizado troca de posição de dois nós de uma mesma rota. A troca ilustrada na Figura 1 é realizada entre os nós 7 e 8, mas a troca também pode ocorrer entre nós não consecutivos. A Busca Local *TrocaInterRota(2,2)* é semelhante a Busca Local *TrocaInterRota(1,1)*, mas são considerados quatro nós de duas rotas diferentes, sendo dois nós consecutivos por rota. Na Figura 1 os nós envolvidos na troca são os nós 3 e 4 de uma rota e 5 e 6 da outra. Apenas os 4 nós terão seus horários de visitas alterados.

3.3. *Random Variable Neighborhood Descent - RVND*

O VND - *Variable Neighborhood Descent* - é um algoritmo baseado na exploração simultânea de um conjunto de buscas locais. Definido um conjunto de estruturas de vizinhança, a cada iteração uma destas estrutura é utilizada. A partir de uma solução inicial S , é realizada uma busca local utilizando uma vizinhança e , se não conseguir nenhuma melhora da solução, uma nova estrutura é utilizada na iteração seguinte. Mas, se ao final da busca local a nova solução for melhor que a corrente, então retorna-se à primeira estrutura de vizinhança para recomeçar a busca [Talbi 2009].

O Algoritmo 2 descreve o VND onde as cinco Buscas Locais foram combinadas. A cada Busca Local foi atribuído um número entre 1 e 5 para identificá-la. Um vetor de inteiros com tamanho igual a 5 guarda a ordem em que serão executadas. Desta forma, qualquer ordem de execução poderá ser facilmente definida. Quando a ordem é estabelecida de forma aleatória, tem-se o RVND - *Random Variable Neighborhood Descent*.

Algoritmo 2: VND

Entrada: S : Solução, $ordem[5]$: int, $critCusto$: int
Saída: S : Solução

```

1 início
2    $i \leftarrow 1$ ;
3   repita
4      $cAnt \leftarrow custo(S, critCusto)$ 
5      $S \leftarrow buscaLocal(S, vet[i], critCusto)$ 
6      $cDep \leftarrow custo(S, critCusto)$ 
7     se  $cAnt \geq cDep$  então
8        $i \leftarrow 1$ ;
9     fim
10    senão
11       $i++$ ;
12    fim
13  até  $i \leq 5$ ;
14 fim
```

3.4. *Iterated Local Search - ILS*

O ILS tem como ideia principal sair de ótimos locais perturbando a solução corrente e, a partir da nova solução, tenta-se chegar a um novo ótimo local. A perturbação é um

procedimento que modifica a solução mantendo algumas de suas características. Ou seja, as iterações do ILS são divididas entre a perturbação e o refinamento.

O Algoritmo 3 apresenta o algoritmo ILS proposto neste trabalho. Os parâmetros G , $porRem$, $cOrd[n]$, $cCusto[n]$, $cInsercao[n]$, $vetFichas[n]$ e n são utilizados na fase de perturbação. O $itILS$ indica o número de iterações, critério de parada do algoritmo. Na linha 2, o valor de α_{ILS} , que será utilizado durante a reconstrução da solução (linha 9), é inicializado com 0.1. Na linha 4, é feita uma cópia da solução inicial para S_{melhor} . No laço entre as linhas 5 e 17 ocorrem a perturbação e o refinamento da solução.

A perturbação ocorre entre nas linhas 7, 8 e 9. Na linha 7 é feita a remoção de uma parte dos nós definido pelo parâmetro $porRem$. Para a remoção, uma lista com os nós finais e iniciais de cada rota é construída, e é calculado o impacto da remoção de cada um, individualmente, considerando a emissão. Para cada nó é atribuído uma quantidade de fichas de modo que o nó que impacta com mais poluição receba proporcionalmente mais fichas. Então, um nó da lista é sorteado para a remoção e o novo nó, inicial ou final, entra na lista, mantendo-se a proporcionalidade de ficha com o impacto de sua emissão.

Algoritmo 3: ILS

Entrada: G : Grafo, S : Solução, $porRem$: double, $cOrd[n]$: int, $cCusto[n]$: int, $cInsercao[n]$: int, n : int, $cCustoObj$: int, $vetFichas[n]$: double, α_{In} : double, $itILS$: int

Saída: S_{melhor}

```

1 início
2    $\alpha_{ILS} \leftarrow 0.1$ ;
3    $j \leftarrow 1$ ;
4    $S_{melhor} \leftarrow copiaSolucao(S)$ ;
5   repita
6      $S_{ILS} \leftarrow copiaSolucao(S_{melhor})$ ;
7      $S_{ILS} \leftarrow remove(porRem)$ ;
8      $c \leftarrow SorteiaCombinacao(vetFichas, n)$ ;
9      $S_{ILS} \leftarrow Construtivo(G, cOrd[c], cCusto[c], cInsercao[c], \alpha_{ILS})$ ;
10     $S_{ILS}^* \leftarrow RVND(S_{ILS}, cCustoObj)$ ;
11    se  $custo(S_{ILS}^*, cCustoObj) < custo(S_{melhor}, cCustoObj)$  então
12       $S_{melhor} \leftarrow S_{ILS}^*$ ;
13    fim
14     $j++$ ;
15     $\alpha_{ILS} \leftarrow \alpha_{ILS} + \alpha_{In}$ ;
16  até  $j < it_{ILS}$ ;
17 fim
18 retorna  $S_{melhor}$ 

```

Na linha 8 é sorteada uma combinação de critérios para ser utilizada na reconstrução da solução (linha 9). O vetor $vetFichas[n]$ apresenta uma quantidade de fichas para cada combinação de critérios - seguindo o mesmo raciocínio de proporcionalidade com a emissão, descrita anteriormente para a lista de nós a serem removidos. Para definir essa quantidade, o Algoritmo Construtivo 1 é executado de forma gulosa uma vez para cada

cada combinação. Em seguida, é aplicado o VND de modo que a ordem de execução das Buscas Locais seja a mesma para todas as combinações de critérios - a ordem das Buscas Locais nessa parte muda para cada execução do ILS. A fase de refinamento acontece na linha 10, com o RVND. A solução só será aceita se o custo da nova solução for menor que o custo da solução atual. Na linha 15, o α_{ILS} é incrementado pelo valor de α_{In} .

4. Resultados

O algoritmo proposto foi implementado em C++, em um computador com sistema operacional Ubuntu 16.10 de 64 bits, 4 GB de RAM, processador Intel Core i5 de 2.60GHz. Foram adaptados 10 grupos de instâncias de dois trabalhos diferentes. Três grupos de instâncias foram adaptados de [Cordeau 2013], em um total de 56 instâncias. Os outros sete grupos foram adaptados de [Xiao and Konak 2016], sendo que cada grupo possui 20 instâncias, um total de 140. Todas as instâncias possuem demanda, janela de tempo e um depósito central. Na Seção 4.1 serão descritas as adaptações das instâncias e na Seção 4.2 os resultados computacionais dos testes.

4.1. Construção das instâncias

Por se tratarem de problemas diferentes e por não conseguir instâncias compatíveis com o problema abordado, houve a necessidade de se fazer alterações em instâncias de problemas parecidos. Para as instâncias de [Cordeau 2013] foram realizados ajustes de horário e distância, além do acréscimo de velocidades para os períodos de tempo. Primeiramente, dois ajustes foram realizados: o tempo foi convertido para valores entre 7 e 17 horas, mantendo-se a escala; a distância entre os nós foi convertida para valores entre 92.2 e 2756.5m. Estes ajustes foram realizados para tornar as instâncias mais próximas com os valores reais apresentados em [Qian and Eglese 2016].

O dia para este grupo de instâncias foi dividido em 10 períodos de uma hora cada: de 7:00 às 8:00, 8:00 às 9:00, ..., 15:00 às 16:00 e 16:00 às 17:00. O período de tempo é determinado pela hora que o veículo sai do nó, ou seja, se houver variação do período durante a viagem, de 14:59 para 15:01 por exemplo, será considerado o limite definido para o intervalo entre 14:00 e 15:00 horas.

Foram criados 5 padrões de velocidades para cada par de nós baseados em [Xiao and Konak 2015]. As velocidades foram modificadas para variar entre 6 e 90km/h já que a função de emissão dos gases apresentada em [P. G. Boulter and McCrae 2009] cobre apenas essa faixa de velocidade para o veículo de código R166, que foi o veículo considerado neste trabalho. Dos 6 grupos de velocidades apresentados em [Xiao and Konak 2015], apenas o grupo com velocidade constante não foi utilizado. Todo arco que sai do depósito foi definida com a velocidade máxima de 90km/h para qualquer horário.

Para as instâncias de [Xiao and Konak 2016], a distância também foi convertida para valores entre 92.2m e 2756.5m. Como o dia possui 5 períodos de tempo, os horários foram convertidos para valores entre 0 e 5. As velocidades maiores que 90km/h foram modificadas para 90km/h, limite máximo para $v_{i,j}$ em 1. O problema abordado em [Xiao and Konak 2016] considera frota heterogênea e, por isso, existem mais de um tipo de veículo com capacidades de carga diferentes. Como o problema abordado neste

trabalho considera frota homogênea, foi selecionado a maior capacidade de carga entre os veículos. Para mais detalhes a respeito das adaptações das instâncias consultar [Mendes Júnior 2017].

4.2. Resultados computacionais

Todos as execuções foram realizados com α_{In} igual a 0,1/300 para que ao final 300 iterações(it) α_{ILS} tenha o valor igual a 0,2. Cada instância foi executada 30 vezes.

A Tabela 3 apresenta, para cada um dos grupos de instâncias, a média da emissão, da distância percorrida e do tempo de execução para duas versões do ILS: a ILS_Pol, que minimiza a emissão, e a ILS_Dist, que minimiza a distância. É possível observar que minimizar a distância não garante baixa emissão. As soluções encontradas pelo ILS_Pol emitem, em média, 14.01% menos poluição apesar de percorrerem uma distância 10.25% maior que as soluções encontradas pelo ILS_Dist.

Tabela 3. Emissão, distância e tempo de execução médios para as duas versões do ILS

Instâncias	ILS_Pol			ILS_Dist		
	Emissão(kg)	Distância(km)	Tempo(s)	Emissão(kg)	Distância(km)	Tempo(s)
UK10	10.59	10.51	0.03	13.06	10.24	0.02
UK15	12.62	12.52	0.10	16.05	11.99	0.03
UK20	14.89	14.26	0.17	18.27	13.46	0.10
UK25	18.50	17.93	0.33	22.68	16.77	0.13
UK50	37.98	36.56	0.78	44.76	33.81	0.48
UK75	64.44	61.22	1.36	74.62	56.65	0.85
UK100	81.48	77.61	2.26	92.93	70.92	2.07
C	44.65	39.66	3.07	47.52	34.75	3.32
R	58.82	56.03	3.50	68.84	46.96	4.17
RC	59.93	57.24	4.50	70.92	48.66	3.67
Média	40.39	38.35	1.62	46.97	34.42	1.49

Para efeito de comparação, a Tabela 4 mostra a emissão e distância percorrida para duas abordagens do GRASP descritas em [Mendes Júnior 2017] para um subconjunto das instâncias: uma abordagem que minimiza a distância (GRASP_Dist) e outra que minimiza a emissão (GRASP_Pol) - do mesmo modo que foi feito para o ILS. O resultado mostra que apesar do GRASP_Dist encontrar soluções que percorrem em média distâncias menores que as encontradas pelo GRASP_Pol - 20.09% menos - a emissão das soluções do GRASP_Dist são 10.53% maiores.

Resultados apresentados em [Mendes Júnior 2017] mostram que as melhores combinações de critérios para o GRASP, quando executado com as 12 combinações para um subconjunto de instâncias (o mesmo da Tabela 4, são aquelas que consideram minimizar a distância durante a fase de construção da solução inicial e minimizar a emissão durante a fase de refinamento. Comportamento semelhante ocorre com ILS, com destaque para as combinações 11 e 12 que encontra as melhores soluções em 5.19 e 6.12 vezes de 30 execuções, respectivamente.

5. Conclusão

A ideia deste trabalho é apresentar uma abordagem heurística baseada no ILS para o Problema de Roteamento Verde com Velocidades Variáveis no Tempo com duas variações:

uma variação que tem como função objetivo minimizar a distância enquanto que a outra minimiza a emissão. E então compará-las.

Quando comparados o ILS_Pol com o ILS_Dist fica claro, como esperado, qual obteve o melhor resultado com relação a emissão, já que o objetivo do problema é minimizar a emissão dos gases poluentes e o foco do ILS_Pol é exatamente esse. Minimizar distância não garante a minimização da emissão de gases poluentes, apesar da distância estar diretamente ligada ao cálculo da emissão. Comparando os resultados encontrados pelo ILS com um outro algoritmo baseado no GRASP, e seguindo a mesma ideia desenvolvendo duas variações - uma que minimize a distância e outra a emissão -, também chega-se a mesma conclusão: minimizar a distância não garante a minimização da emissão de gases poluentes. Porém, combinações de critérios que consideram minimizar a distância na fase de criação da solução inicial e minimizar a emissão na fase de refinamento apresentam as melhores soluções.

Tabela 4. Emissão, distância e tempo de execução médios para as duas versões do GRASP e ILS para um subconjunto das instâncias

Instâncias	Min. Emissão						Min. Distância					
	Grasp			ILS			Grasp			ILS		
	Emi.(kg)	Dist.(km)	Tem.(s)	Emi.(kg)	Dist.(km)	Tem.(s)	Emi.(kg)	Dist.(km)	Tem.(s)	Emi.(kg)	Dist.(km)	Tem.(s)
c101	68.72	56.3	6.24	55.40	46.11	3.72	61.00	42.30	6.96	62.16	43.56	2.74
c102	72.17	64.28	9.33	61.91	56.03	4.39	71.09	45.79	10.54	61.09	43.77	2.98
c203	60.40	56.68	11.75	51.57	48.82	4.98	56.5	39.55	14.47	51.60	36.85	3.09
c204	50.18	48.33	13.39	49.06	47.23	5.99	54.29	36.28	16.48	49.34	34.91	3.51
r111	64.62	61.51	8.14	53.29	51.07	4.93	68.42	44.46	11.64	64.61	42.30	3.53
r112	59.13	57.04	11.44	49.12	47.85	5.78	64.43	40.48	14.02	56.90	39.01	4.00
r209	59.56	58.07	8.95	62.7	60.46	4.00	70.59	45.06	9.63	66.05	43.69	3.28
r210	56.06	55.16	7.88	55.33	54.54	4.93	68.4	47.66	9.33	66.19	45.77	3.59
rc105	80.06	75.12	7.62	66.76	63.53	4.33	92.47	58.25	10.73	82.96	55.45	3.21
rc106	73.00	68.06	8.06	58.59	55.88	4.18	73.44	49.43	11.51	68.05	46.79	3.16
rc207	65.85	63.05	9.40	57.67	56.27	4.66	74.93	46.67	10.85	66.78	44.50	3.22
rc208	60.39	57.92	10.73	50.66	49.32	6.31	57.1	37.77	11.98	54.94	38.70	3.36
UK10x5_1	10.90	10.74	0.10	10.99	10.74	0.04	11.18	10.52	0.03	11.15	10.53	0.02
UK10x5_2	9.91	9.87	0.05	9.80	9.76	0.03	13.04	8.96	0.03	9.98	8.91	0.02
UK15x5_3	12.40	11.91	0.21	12.43	12.1	0.13	15.91	11.59	0.09	16.25	11.72	0.04
UK15x5_4	12.97	12.9	0.21	13.59	13.05	0.13	17.23	12.15	0.10	16.62	12.17	0.06
UK20x5_5	15.85	15.05	0.27	16.04	14.81	0.20	21.17	14.19	0.16	19.34	14.29	0.09
UK20x5_6	7.79	7.74	0.34	7.99	7.92	0.24	10.59	7.28	0.19	10.59	7.31	0.09
UK25x5_7	20.03	19.59	0.51	20.36	19.48	0.26	25.4	17.71	0.33	24.53	17.59	0.12
UK25x5_8	22.46	21.7	0.54	21.68	21.37	0.41	27.35	19.47	0.34	25.69	19.55	0.13
UK50x5_9	45.86	44.71	1.10	44.17	42.02	0.70	54.4	39.57	1.17	53.22	38.61	0.32
UK50x5_10	36.81	36.43	1.20	35.15	34.14	0.57	44.49	32.13	1.19	42.95	31.84	0.24
UK75x5_11	54.06	53.43	2.27	49.41	48.74	1.31	65.03	46.44	3.09	59.56	44.93	0.74
UK75x5_12	66.51	65.25	2.12	62.79	59.68	1.26	81.5	56.82	2.75	73.67	55.46	0.69
UK100x5_13	89.31	87.33	3.55	85.15	81.43	1.88	106.16	76.70	5.75	97.72	74.25	1.23
UK100x5_14	97.41	95.31	3.77	92.55	87.86	1.75	116.09	82.55	5.80	108.45	79.60	1.51
Média	48.94	46.68	4.97	44.39	42.32	2.58	54.7	37.3	6.12	50.79	36.23	1.73

Para trabalhos futuros pretende-se considerar utilizar outros algoritmos construtivos, bem como tornar o algoritmo adaptativo para escolha automática dos critérios durante a construção da solução inicial. Outro ponto a ser pensado é fazer um pré-processamento dos nós e agrupá-los pela distância, dando preferência para os nós mais próximos na hora de inseri-los em uma rota. Acrescentar mais buscas locais como, por exemplo, uma que divida uma rota em duas ou uma duas rotas, aumentar o número de iterações ou os valores que α pode assumir durante a fase de construção também podem melhorar os resultados.

Referências

Cordeau (2013). Capacitated vrp with time windows instances , <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-with-time-windows-instances/>. Acessado em: 30 de Março 2017.

- Department for Environment, Food and Rural Affairs (2010). *Guide lines to Defra/DECC's green house gas conversion factors for company reporting*. Acessado em: 28 de Outubro 2014.
- Lin, C., Choy, K., Ho, G., Chung, S., and Lam, H. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, pages 1118–1138.
- Lourenço, H. R., Martin, O. C., and Stützle, T. (2010). *Handbook of Metaheuristics*, chapter Iterated Local Search: Framework and Applications. Springer New York, Editors: M. Gendreau and J.Y. Potvin.
- McKinnon, A., Cullinane, S., Browne, M., and Whiteing, A. (2010). *Green Logistics: Improving the environmental sustainability of logistics*. Kogan Page Limited.
- Mendes Júnior, D. J. (2017). Metaheurística aplicada ao problema de roteamento verde de veículos com velocidades variáveis no tempo. Trabalho de Conclusão de Curso, Ciência da Computação, Universidade Federal de Juiz de Fora.
- Molina, J. C., Eguia, I., Racero, J., and Guerrero, F. (2014). Multi-objective vehicle routing problem with cost and emission functions. *Procedia - Social and Behavioral Sciences*, pages 254–263.
- P. G. Boulter, T. J. B. and McCrae, I. S. (2009). Emissions factors 2009: Report 3 - exhaust emission factors for road vehicles in the united kingdom. Acessado em: 30 de Março 2017.
- Picelli, V. C. and Georges, M. R. R. (2011). Cadeia de suprimentos reversa e logística verde: Teoria e prática. In *Anais do XVI Encontro de Iniciação Científica da PUC-Campinas e I Encontro de Iniciação em Desenvolvimento Tecnológico e Inovação da PUC-Campinas*.
- Qian, J. and Eglese, R. (2016). Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, pages 840–848.
- Rao, S., Pachauri, S., Dentener, F., Kinney, P., Klimont, Z., Riahi, K., and Schoepp, W. (2013). Better air for better health: Forging synergies in policies for energy access, climate change and air pollution. *Elsevier Ltd*, pages 1122–1130.
- Sbihi, A. and Eglese, R. W. (2010). Combinatorial optimization and green logistics. In *Annals of Operations Research*, pages 159–1175, <https://doi.org/10.1007/s10479-009-0651-z>. Springer.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. John Wiley and Sons, Inc.
- Xiao, Y. and Konak, A. (2015). A simulating annealing algorithm to solve the green vehicle routing and scheduling problem with hierarchical objectives and weighted tardiness. *Applied Soft Computing*, pages 372–388.
- Xiao, Y. and Konak, A. (2016). The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part*, pages 146–166.