



Problema de Roteamento de Veículos voltado a minimização de emissões de dióxido de carbono

Marcos Raylan Sousa Matos, Yuri Abitbol de Menezes Frota, Luiz Satoru Ochi
Instituto de Computação - Universidade Federal Fluminense
raylan@id.uff.br, {yuri, satoru}@ic.uff.br

Av. Gal. Milton Tavares de Souza, s/nº - Instituto de Computação - 5º andar (sala 546)
São Domingos, Niterói - RJ - CEP: 24210-346
Telefones: (+055 21) 2629-5629 / 5628 / 5627 (fax).

RESUMO

O presente trabalho aborda um problema da classe do *Green Vehicle Routing and Scheduling Problem* (GVRSP). O problema abordado tem como objetivo minimizar as emissões de Dióxido de Carbono (CO_2) em sistemas de logística através de um melhor planejamento de entregas realizadas por uma frota de veículos. Neste trabalho, o GVRSP considera veículos heterogêneos, congestionamento de tráfego variável no tempo, restrições de capacidade, janelas de tempo além da entrega poder ser fracionada. O GVRSP também permite que os veículos parem nos arcos, desta forma um veículo tem permissão para percorrer um arco em múltiplos períodos de tempo. Uma formulação matemática foi proposta para descrever o GVRSP. Além disso, utilizou-se uma abordagem baseada em uma metaheurística *Multi-Start Iterated Local Search* com uma etapa de busca local inspirada na metodologia de *Variable Neighborhood Descent*. Os resultados obtidos são comparados com o estado da arte, mostrando que a abordagem proposta possui um melhor desempenho.

PALAVRAS CHAVE. Roteamento de Veículos, Emissão de CO_2 , Congestionamento de Tráfego.

Tópicos (Metaheurísticas, Otimização Combinatória, Logística e Transportes)

ABSTRACT

This paper deals with a class of Green Vehicle Routing and Scheduling Problem (GVRSP). The problem attacked aims to minimize Carbon Dioxide (CO_2) emissions in logistics systems through better planning of deliveries made by a fleet of vehicles. In this paper, the GVRSP considers heterogeneous vehicles, time-varying traffic congestion, capacity constraints, time window constraints and the delivery can be fractionated. The GVRSP also allows vehicles to stop on arcs, so vehicle is allowed to travel an arc in multiple time periods. A mathematical formulation was proposed to describe the GVRSP. Moreover, was used approach based on a Multi-Start Iterated Local Search metaheuristic with a local search step based on the Variable Neighborhood Descent methodology. The results obtained are compared with the state of the art and show that the proposed approach has better performance.

KEYWORDS. Vehicle Routing, CO_2 Emission, Traffic Congestion.

Paper topics (Metaheuristics, Combinatorial Optimization, Logistics and Transport)



1. Introdução

O Problema de Roteamento de Veículos (do termo em inglês, *Vehicle Routing Problem* - VRP) teve sua origem associada ao trabalho desenvolvido em [Dantzig e Ramser, 1959], o trabalho tratava do problema de distribuição de combustíveis e tinha como objetivo determinar o menor custo de transporte para atender um conjunto de postos de revenda. Ao contrário do VRP, variantes do *Green Vehicle Routing Problem* (GVRP) visam, além de minimizar os custos de transporte, minimizar funções que ponderam e promovem ambientes mais agradáveis e sustentáveis.

Dentre a variação do VRP, há uma variante no qual o atendimento do cliente poderá ser realizado por mais de um veículo, essa variante permite que a demanda de cada cliente seja dividida em várias frações. Uma consequência direta disso é que os clientes podem ter demandas superiores a capacidade do veículo, essa variante do VRP é referenciada na literatura como *Split Delivery Vehicle Routing Problem* (SDVRP) [Archetti e Speranza, 2008]. O SDVRP foi inicialmente proposto por [Dror e Trudeau, 1989], no trabalho os autores desenvolveram abordagens heurísticas para resolvê-lo em termos da distância percorrida e quantidade de veículos utilizados.

Recentemente, uma nova variante do VRP foi introduzida na literatura, o *Time-Dependent Vehicle Routing Problem* (TDVRP) foi proposto inicialmente no trabalho de [Chen et al., 2006]. No TDVRP, o tempo necessário para um veículo deslocar ao longo de uma estrada varia de acordo com o período de tempo em que a viagem é realizada. O horizonte de tempo é dividido em períodos, onde cada período representa um intervalo de tempo durante um dia. Por exemplo, o período 1 equivale ao período da manhã (8h às 11h), período 2 define o intervalo próximo das 11h às 13h (comumente chamado de hora do *rush*), período 3 equivale ao período da tarde (13h às 18h) e assim por diante. Para cada período, temos previamente definido uma velocidade para percorrer uma estrada, esta velocidade prefixada é baseada nas médias de velocidade obtidas naquela estrada. Essa variação de velocidades para cada período tenta simular um sistema real de trânsito, desse modo, baixas velocidades tendem a representar congestionamentos.

O tempo de partida de um vértice é usualmente considerado como uma variável de decisão, além disso, é assumido que um arco é percorrido em um único período [Chen et al., 2006]. Nesse trabalho, é permitido que o veículo pare nos arcos, assim, um veículo poderá percorrer um arco em múltiplos períodos de tempo. A decisão de roteamento dessa classe do GVRSP é semelhante ao VRP tradicional, porém é considerado em um domínio de tempo, pois trata de como agendar tarefas de viagem de forma detalhada dentro do horizonte de vários períodos de tempo. Assim, um veículo, munido de cargas, percorrendo um arco dentro de um período numa velocidade implica em uma emissão de CO_2 , i.e., um veículo carregado e em baixa velocidade tende a produzir mais CO_2 quando comparado com o mesmo veículo em velocidade superior ou vazio. Além das restrições originais, foi incluído restrições específicas do SDVRP, o modelo com *Split* torna-se interessante não apenas quando há demandas fracionáveis, mas também quando a capacidade do veículo é inferior às demandas. Logo, comparar os resultados com e sem *Split* é essencial para definir um limite inferior para o problema [Archetti e Speranza, 2008].

A utilização de metaheurísticas que encontram resultados viáveis, ou até mesmo ótimos, em tempo hábil tem despertado o interesse de grandes pesquisadores [Blum e Roli, 2003]. Assim, como alternativa aos métodos exatos, a metaheurística *Iterated Local Search* (ILS) foi empregada como técnica para o SDVRP em [Silva et al., 2015]. Versões híbridas de metaheurísticas com métodos exatos tem atraído bastante interesse ultimamente, um exemplo de hibridização foi aplicado com sucesso a uma metaheurística ILS para o VRP com frota heterogênea [Penna et al., 2013].

O presente trabalho propõe a implementação não apenas do modelo matemático, como também a metaheurística ILS para determinar o conjunto ótimo de rotas, minimizando assim, a poluição gerada pelos veículos. Desta forma, o trabalho está organizado como segue. A Seção 2 apresenta alguns dos principais trabalhos da literatura que envolvem o GVRSP, a Seção 3 faz uma breve descrição do modelo matemático assim como a definição das restrições do problema. A Seção 4 descreve a proposta heurística utilizada para resolver o problema. Os parâmetros e resultados são discutidos respectivamente nas Seções 5 e 6. Por fim, uma breve conclusão é realizada na Seção 7.



2. Revisão de Literatura

O VRP pode ser definido como um conjunto de problemas de otimização combinatória, tais problemas tem recebido muita atenção nas últimas décadas devido à sua importância prática e econômica em áreas de transporte, telecomunicações e planejamento de custos. Desde a sua introdução por [Dantzig e Ramser, 1959], o VRP tem sido extensivamente estudado na literatura. Diversas variações do VRP foram desenvolvidas para diferentes aplicações, tais como coleta e entrega, múltiplos depósitos, janelas de tempo, entrega dividida e dependente do tempo. Alguns *Surveys* das mais diversas formulações do VRP podem ser encontrados em [Laporte et al., 2000], [Toth e Vigo, 2002], [Golden et al., 2008] e [Eksioglu et al., 2009].

O Problema de Roteamento e Agendas de Veículos (do termo em inglês, *Vehicle Routing and Scheduling Problem* - VRSP) foi definido pela primeira vez por [Bodin e Golden, 1981], no VRSP os clientes têm requisitos específicos de tempo de serviço. Portanto, uma solução para o VRSP inclui tanto as rotas como os cronogramas dos veículos, onde esses cronogramas indicam os horários de chegada e partida nos clientes. Em uma abordagem exata, [Solomon, 1984] apresentou pela primeira vez uma formulação de Programação Inteira Mista para o VRSP com restrições de janela de tempo, onde alguns clientes impõem restrições de tempo em suas coletas e entregas.

Nos últimos anos, uma outra questão relacionada ao TDVRP foi proposta com o objetivo de minimizar as emissões de CO_2 em sistemas logísticos. [Kara et al., 2007] propuseram um modelo para minimizar o consumo de energia no contexto do VRP. Em outro cenário, [Jabali et al., 2012] desenvolveram um modelo para um TDVRP baseado em emissões, onde um limite de velocidade uniforme é pré-determinado para todos os arcos. Sob outra perspectiva, [Bektaş e Laporte, 2011] apresentaram o Problema de Roteamento com Poluição (do termo em inglês, *Pollution Routing Problem* - PRP) em que a velocidade de deslocamento ótima de cada arco (i, j) é determinada para minimizar uma função objetivo, o problema apresentava restrições de consumo de combustível e emissões de CO_2 . Neste mesmo trabalho, [Bektaş e Laporte, 2011] propuseram um modelo para minimizar o consumo de combustível, os autores usaram uma abordagem heurística baseada em *Simulated Annealing* (SA) para resolver o problema. Neste mesmo contexto, [Franceschetti et al., 2013] estendeu o PRP ao PRP *Time-Dependent* (TDPRP), o trabalho considerava um tráfego com congestionamento em dois períodos com velocidades prefixadas. [Xiao e Konak, 2015a] também abordaram o TDPRP, os autores propuseram uma abordagem híbrida de algoritmos genéticos e programação dinâmica para resolvê-lo.

Recentemente, [Kramer et al., 2015] propuseram uma abordagem combinando uma metaheurística com um modelo MILP (do termo em inglês, *Mixed Integer Linear Programming*) para o PRP e relataram um melhor desempenho do que o algoritmo proposto por [Maden et al., 2010] sob as mesmas instâncias. [Soysal et al., 2015] propuseram um modelo MILP para o VRP capacitado e dependentes do tempo com o objetivo de minimizar o consumo de combustível.

Os trabalhos mais recentes sobre a otimização das emissões de CO_2 no contexto do VRSP em condições de tráfego com variações de tempo são os trabalhos de [Figliozzi, 2010] e [Xiao e Konak, 2015a,b, 2016]. Nessa perspectiva, um modelo MILP foi proposto para o TDPRP com objetivos hierárquicos e uma penalidade de atraso ponderada sob um sistema de comércio de carbono [Xiao e Konak, 2015b], os autores avaliaram as emissões totais de CO_2 , tempo de chegada e o tempo total de viagem. Por fim, pode-se citar o *Survey* proposto por [Lin et al., 2014] como introdução ao GVRP e suas relações com outras versões do TDVRP. Pode-se citar também o trabalho proposto por [Cohen, 2011] como uma introdução ao tema de *Green Cities*.

Ao analisar a complexidade dos problemas de roteamento de veículos, concluíram que todos os estes problemas e suas extensões são \mathcal{NP} -Difícil, pois não são resolvidos em tempo polinomial [Lenstra e Kan, 1981]. Neste sentido, encontrar métodos que solucionem o problema de roteamento, com baixo custo computacional, tem tomado muita atenção de pesquisadores que aplicam e adaptam tais métodos a fim de encontrar os melhores resultados.



3. Problema de Roteamento de Veículos voltado a minimização de CO₂

O GVRSP é uma variante do VRP com a adição de restrições de dispersão de CO₂. O problema em questão pode ser apresentado por um grafo completo $G = (V, A)$, onde $V = \{0, 1, \dots, n\}$ é o conjunto de vértices que representam as localizações geográficas dos n clientes e $A = \{(i, j) : i, j \in V\}$ o conjunto de arcos. O conjunto dos clientes e depósito é representado por $N = \{0, \dots, n\}$, já o conjunto restrito de clientes é dado por $N' = N \setminus \{0\}$. Assim, cada cliente i possui uma demanda positiva R_i ($i \in V' \mid R_0 = 0$), além disso, o depósito assume uma demanda nula. Por último, um custo de viagem D_{ij} (assimétrico) está associado a cada arco $(i, j) \in A$.

Essa variante considera que no máximo q veículos, partindo de um único depósito, têm uma emissão de CO₂ quando utilizados. Dentre os q veículos, alguns são utilizados para servir os n clientes. O cliente i possui uma janela de tempo $[S_i, E_i]$ para atendimento e um tempo de serviço s_i para descarga da sua respectiva demanda. O horizonte de planejamento é particionado em um conjunto de períodos $K = \{[b_1, e_1], [b_2, e_2], \dots, [b_m, e_m]\}$, onde m , b_k e e_k representam respectivamente a quantidade de períodos de tempo, início e fim do período k . O arco (i, j) possui condições variáveis de tráfego, i.e., uma velocidade v_{ij}^k no trajeto $(i, j) \in A$ é prefixada em cada período $k \in K$. Por fim, cada veículo v possui uma capacidade C_v e trará consigo a fração de demanda do cliente i , demanda essa representada por y_i^v .

A emissão de CO₂ do veículo v é baseada na taxa de consumo de combustível (TCC) em função da velocidade v_{ij}^k e peso das cargas f_{ij}^v do veículo v no arco (i, j) . A função TCC, denotada pela variável g_{ijk}^v , usa coeficientes específicos para cada tipo de veículo para estimar o consumo de combustível. Foram utilizados nesse trabalho dois tipos de veículos com capacidades (C_v), coeficientes (α , β e γ), disponibilidade de tempo $[F_v, T_v]$ e capacidade de tanque de combustível (L_v) diferentes. Além disso, previamente definimos uma variável binária w_{uv} , o valor 0/1 indica se o veículo v pertence ou não ao tipo u . A descrição completa dos veículos bem como outros modelos de emissão podem ser encontrados em [Xiao e Konak, 2016]. Por fim, a função TCC pode ser melhor compreendida pela Equação 1.

$$FCR_v(v_{ij}^k, f_{i,j}^v) = \alpha_v \times (v_{ij}^k)^{-1} + \beta_v \times (v_{ij}^k)^2 + \gamma_v + \varphi_v \times f_{ij}^v \quad (1)$$

Logo abaixo temos a descrição das variáveis de decisão assim como o modelo matemático proposto para o problema. Posteriormente, é feita uma breve descrição do modelo a fim de detalhar melhor as restrições do problema. Nesse trabalho, o modelo MILP foi baseado no modelo proposto em [Xiao e Konak, 2016] e adicionado restrições específicas do SDVRP descritas no Survey de [Archetti e Speranza, 2008].

Variáveis de decisão:

$$\begin{aligned} \bullet x_{ijk}^v &= \begin{cases} 1, & \text{se o veículo } v \text{ percorre} \\ & \text{o arco } (i, j) \text{ no período } k \\ & \text{(não necessariamente todo o arco)} \\ 0, & \text{c.c.} \end{cases} \\ \bullet d_{ijk}^v &= \begin{cases} \text{Distância percorrida pelo veículo } v \\ \text{no arco } (i, j) \text{ no período } k \end{cases} \\ \bullet y_i^v &= \begin{cases} \text{Quantidade de demanda atendida} \\ \text{para o cliente } i \text{ pelo veículo } v \end{cases} \\ \bullet \bar{X}_{ij}^v &= \begin{cases} 1, & \text{se o veículo } v \text{ percorre} \\ & \text{o arco } (i, j) \text{ em algum tempo} \\ 0, & \text{c.c.} \end{cases} \\ \bullet t_{ijk}^v &= \begin{cases} \text{Tempo em que o veículo } v \\ \text{percorre no arco } (i, j) \text{ no período } k \end{cases} \\ \bullet l_i^v / \alpha_i^v &= \begin{cases} \text{Tempo de partida e chegada} \\ \text{no vértice } i \text{ pelo veículo } v \end{cases} \\ \bullet f_{ij}^v &= \begin{cases} \text{Carga do veículo } v \text{ no arco } (i, j) \end{cases} \end{aligned}$$

(MILP) Minimizar

$$\text{Min } CO_2 = \sum_{(i,j) \in A} \sum_{k \in K} \sum_{u \in U} \sum_{v \in M} w_{uv} c_v g_{ijk}^v d_{ijk}^v + \sum_{(i,j) \in A} \sum_{u \in U} \sum_{v \in M} w_{uv} c_v \varphi_v f_{ij}^v D_{ij} \quad (2)$$



Sujeito a:

$$\sum_{j \in N'} \bar{X}_{0j}^v \leq 1, \quad \forall v \in M \quad (3)$$

$$\sum_{(i,p) \in A} \bar{X}_{ip}^v - \sum_{(p,j) \in A} \bar{X}_{pj}^v = 0, \quad \forall p \in N; \quad \forall v \in M \quad (4)$$

$$x_{jpk_1}^v \leq 2 - x_{ijk_2}^v - \bar{X}_{ij}^v, \quad \forall (i,j), (j,p) \in A; \forall v \in M; \forall (k_1, k_2) \in K | k_1 < k_2 \quad (5)$$

$$x_{ijk}^v \leq \bar{X}_{ij}^v, \quad \forall (i,j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (6)$$

$$\bar{X}_{ij}^v \leq \sum_{k \in K} x_{ijk}^v, \quad \forall (i,j) \in A; \quad \forall v \in M \quad (7)$$

$$\bar{X}_{ij}^v \times D_{ij} = \sum_{k \in K} d_{ijk}^v, \quad \forall (i,j) \in A; \quad \forall v \in M \quad (8)$$

$$d_{ijk}^v \leq D_{ij} \times x_{ijk}^v, \quad \forall (i,j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (9)$$

$$x_{ijk}^v \leq \mathcal{M} \times d_{ijk}^v, \quad \forall (i,j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (10)$$

$$t_{ijk}^v = \frac{d_{ijk}^v}{v_{ijk}} \times 60, \quad \forall (i,j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (11)$$

$$\sum_{(i,j) \in A} t_{ijk}^v \leq (e_k - b_k), \quad \forall k \in K; \quad \forall v \in M \quad (12)$$

$$l_i^v \leq e_k - t_{ijk}^v + \mathbf{e}_m \times (1 - x_{ijk}^v), \quad \forall (i,j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (13)$$

$$a_j^v \geq b_k + t_{ijk}^v - \mathbf{e}_m \times (1 - x_{ijk}^v), \quad \forall (i,j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (14)$$

$$a_j^v \geq l_i^v + \sum_{k \in K} t_{ijk}^v - \mathbf{e}_m \times (1 - \bar{X}_{ij}^v), \quad \forall (i,j) \in A; \quad \forall v \in M \quad (15)$$

$$a_i^v + s_i \leq l_i^v, \quad \forall i \in N'; \quad \forall v \in M \quad (16)$$

$$S_i \leq a_i^v \leq E_i, \quad \forall i \in N'; \quad \forall v \in M \quad (17)$$

$$\sum_{u \in U} w_{uv} \times L_v \geq \sum_{(i,j) \in A} \sum_{k \in K} g_{ijk}^v \times d_{ijk}^v + \sum_{(i,j) \in A} \sum_{u \in U} w_{uv} \times \varphi_v \times f_{ij}^v \times D_{ij}, \forall v \in M \quad (18)$$

$$F_v \leq e_k - t_{0jk}^v + e_m \times (1 - x_{0jk}^v) \quad \forall j \in N'; \quad \forall k \in K; \quad \forall v \in M \quad (19)$$

$$F_v \leq a_j^v - t_{0jk}^v + e_m \times (1 - x_{0jk}^v) \quad \forall j \in N'; \quad \forall k \in K; \quad \forall v \in M \quad (20)$$

$$T_v \leq b_k - t_{i0k}^v + e_m \times (1 - x_{i0k}^v) \quad \forall i \in N'; \quad \forall k \in K; \quad \forall v \in M \quad (21)$$

$$T_v \leq l_i^v - t_{i0k}^v + e_m \times (1 - x_{i0k}^v) \quad \forall i \in N'; \quad \forall k \in K; \quad \forall v \in M \quad (22)$$

$$\sum_{(i,j) \in A} \sum_{v \in M} \bar{X}_{ij}^v \geq 1, \quad \forall j \in N \quad (23)$$

$$y_i^v \leq R_i \times \sum_{(i,j) \in A} \bar{X}_{ij}^v, \quad \forall i \in N'; \quad \forall v \in M \quad (24)$$

$$\sum_{v \in M} y_i^v = R_i, \quad \forall i \in N' \quad (25)$$

$$\sum_{i \in N'} y_i^v \leq C_v, \quad \forall v \in M \quad (26)$$

$$\sum_{i \in N} \sum_{v \in M} f_{ij}^v - \sum_{i \in N} \sum_{v \in M} f_{ji}^v = R_j, \quad \forall j \in N' \quad (27)$$

$$f_{ij}^v \leq C_v \times \bar{X}_{ij}^v, \quad \forall (i,j) \in A; \quad \forall v \in M \quad (28)$$



O lado esquerdo da função objetivo (2) minimiza os custos de emissão de carbono (medido em quilogramas - kg) em função da distância (medido em quilômetros - km) para um veículo vazio. O custo é dado pela taxa de consumo de combustível g_{ijk}^v multiplicado pelas variáveis w_{uv} e c_v ¹, o que resulta em uma emissão de CO_2 (kg), essa emissão é finalmente multiplicada pelo segmento de distância d_{ijk}^v (km), obtendo assim, um custo de CO_2 (kg/km). Já o lado direito da função objetivo é voltado para calcular a emissão extra causada pelo peso das cargas do veículo durante o trajeto, novamente temos o resultado da multiplicação de c_v por f_{ij}^v , D_{ij} e φ_v , sendo esta última variável uma taxa adicional de consumo do veículo v quando percorre com algum peso f_{ij}^v .

A restrição (3) garante que um veículo sairá no máximo uma vez do depósito. As restrições (4) e (5) estabelecem, respectivamente, a conservação de fluxo e eliminação de subciclo. As restrições (6) e (7) asseguram a consistência entre as variáveis \bar{X}_{ij}^v e x_{ijk}^v . A restrição (8) garante que se o arco (i, j) é percorrido pelo veículo v , então ele deve percorrer toda a distância D_{ij} . Já as restrições (9) e (10) asseguram a consistência entre as variáveis x_{ijk}^v e d_{ijk}^v . A restrição (10) garante que se o veículo v percorre o arco (i, j) no período k , então todo o arco deverá ser percorrido. Já a restrição (10) assegura que uma distância mínima $d(\frac{1}{M})$ deverá ser percorrida.

As próximas restrições tratam das variáveis de decisão vinculadas ao tempo. Desse modo, a restrição (11) converte a variável t_{ijk}^v em minutos, já a restrição (12) garante que montante de tempos t_{ijk}^v percorrido no período k nunca será maior que o próprio período de tempo k . A consistência das variáveis l_i^v e a_j^v são asseguradas respectivamente pelas restrições (13) e (14), já restrição (15) garante que, para o arco (i, j) , o tempo de chegada no vértice j é maior que o tempo de saída do vértice i . A restrição (16) assegura que o tempo de chegada no vértice i é menor que o tempo de serviço s_i mais o tempo de saída no mesmo vértice i . Por fim a restrição (17) garante que o tempo de chegada no vértice i está dentro da janela de tempo.

O lado esquerdo da desigualdade da restrição (18) avalia a capacidade do tanque de gasolina em função dos respectivos tipos de veículos, já lado direito da desigualdade temos duas expressões, a expressão mais a esquerda define o consumo de combustível baseado na distância total percorrida, em contrapartida, a expressão a direita define o consumo baseado nas cargas do veículo. Os pares de restrições (19, 20) e (21, 22) estabelecem, respectivamente, os tempos mínimos de partida (F_v) e máximo de volta (T_v) do veículo v . As restrições (23) até (26) tratam especificamente do *Split Delivery*. A restrição (23) permite que um cliente seja visitado por mais de um veículo. As restrições (24) e (25) tratam da alocação das demandas, já a restrição (26) assegura que a capacidade do veículo v não será excedida. As restrições (27) e (28) tratam do montante de cargas f_{ij}^v do veículo v . Assim, uma versão do GVRSP sem o *Split Delivery* exige apenas: adição da variável $\widehat{X}_{ij} \in \{0, 1\}$, remoção das restrições (23, 24, ..., 26), substituição das restrições (8) e (15) respectivamente pelas restrições (29) e (30), por fim, a adição das restrições (31), (32) e (33).

$$\widehat{X}_{ij} \times D_{ij} = \sum_{k \in K} \sum_{v \in M} d_{ijk}^v, \quad \forall (i, j) \in A \quad (29)$$

$$a_j^v \geq l_i^v + \sum_{k \in K} \sum_{h \in M} t_{ijk}^h - e_m \times (1 - \widehat{X}_{ij}), \quad \forall (i, j) \in A; \quad \forall v \in M \quad (30)$$

$$\widehat{X}_{ij} = \sum_{v \in M} \bar{X}_{ij}^v, \quad \forall (i, j) \in A \quad (31)$$

$$\sum_{j \in N} \widehat{X}_{ij} = 1, \quad \forall i \in N' \quad (32)$$

$$\sum_{i \in N} \widehat{X}_{ij} = 1, \quad \forall j \in N' \quad (33)$$

A Figura 1 representa o funcionamento do GVRSP com *Split Delivery*. O depósito, representado pelo vértice 0, armazena três veículos e quinze cargas que deverão ser distribuídas aos clientes. Para esse exemplo, os clientes estão representados por vértices numerados de 1 a 15, além disso, toda a frota de 3 veículos foi utilizada. O arco (14, 15) é percorrido pelo veículo r_3 em dois períodos, nesse exemplo, o veículo percorre parte do arco no período P_1 , para e espera iniciar o período posterior, neste caso, o P_3 para continuar o trajeto. Uma situação semelhante ocorre na aresta (5, 10), entretanto, o veículo R_1 não realiza uma parada no arco.

¹ c_v : Taxa de conversão de consumo de combustível para CO_2 do veículo v .

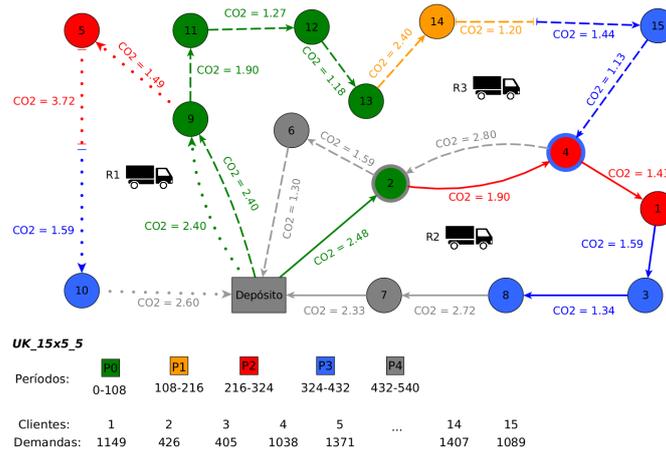


Figura 1: Solução para instância UK-15x5-5: 15 clientes, 3 veículos e emissão de 792.36 kg/km de CO₂.

O custo da solução é o resultado da distância percorrida no arco pela taxa de CO₂ dentro de um período k . Neste sentido, um veículo poderá percorrer um arco (i, j) em vários períodos de tempo, conseqüentemente, variadas distâncias dentro de um arco (i, j) , totalizando assim uma distância máxima D_{ij} . Sistemas com múltiplas paradas dentro de um arco podem tornar os custos de emissão de CO₂ menores, uma vez que, um veículo estando em um período P_i poderá continuar a percorrer o arco em um período P_{i+2} , isto por que o período P_{i+1} gera um custo maior de carbono.

4. Metodologia

Pela natureza do problema não permitir que algoritmos exatos resolvam o problema otimamente em um baixo custo computacional instâncias grandes, portanto o método heurístico foi considerado. Neste sentido, para representar a solução S na heurística, foram utilizados dois vetores: R (com q linhas e um número variável de colunas) e Y (com q linhas e n colunas). Cada linha r do vetor R representará uma rota, a qual terá um veículo associado. As colunas de cada uma dessas rotas indicarão uma permutação de clientes, $R = \{r_1 = [r_{10}, r_{12}, \dots, r_{1n}], r_2 = [r_{20}, r_{22}, \dots, r_{2n}], r_K = [r_{K0}, r_{K2}, \dots, r_{Kn}]\}$, sendo que as primeiras e últimas colunas conterão o depósito. No vetor Y , cada linha r estará associada a uma linha do vetor R . Cada uma das n colunas que compõem as linhas do vetor Y representará um cliente, sendo que Y_{rj} armazenará a fração de demanda do cliente j que é atendida pela rota r , isto se o cliente j é visitado pela rota r em R .

Com estas representações, a informação acerca da quantidade de demanda suprida por cada rota, para um cliente, é acessada rapidamente. Com o intuito de descrever a sequência de períodos ao longo de uma rota a matriz X foi definida, as agendas são representados por uma matriz X (com m linhas e um número variável de colunas), a matriz binária X_{ijk}^v recebendo 1 indica que o arco (i, j) é percorrida no período k pelo veículo v .

Nesse trabalho definimos uma velocidade fixa para cada arco (i, j) com base nas instâncias reportadas em [Xiao e Konak, 2016]. O artigo também fornece dados específicos do veículo (por exemplo, eficiência do motor, peso e capacidade) e as condições do curso do arco (por exemplo, velocidade média e distância). As taxas de emissões de CO₂ (kg/km) são calculadas para cada arco em cada período de tempo, tais emissões podem ser calculadas utilizando outros modelos de emissão da literatura, como os propostos em [Barth et al., 2005] ou [Barth e Boriboonsomsin, 2008]. Para cada arco (i, j) , a velocidade de deslocamento em um período é constante, representando a velocidade média durante o período. Este pressuposto é bem justificado no caso de estradas urbanas e suburbanas, onde a velocidade média de deslocamento depende da hora e do dia.

4.1. Método Construtivo

O procedimento de construção da solução inicial foi adaptado de métodos existentes na literatura. Uma estratégia de inserção foi adotada, a Estratégia de Inserção Paralela (EIP), onde todas as rotas são consideradas, paralelamente, durante o processo construtivo. Com a finalidade de gerar soluções iniciais diversificadas, dois critérios de inserção foram utilizados: Critério de Inserção mais Barata Viável (CIMBV) e o Critério de Inserção mais Próxima Viável (CIMPV).



O pseudocódigo **GeraSoluçãoInicial()** é apresentado no Algoritmo 1. Inicialmente, a Lista de Candidatos (LC) é preenchida com todos os clientes (linha 2). Em seguida, cada rota r é preenchida com um cliente i selecionado, aleatoriamente, da LC (linhas 4 - 7). Também aleatoriamente é selecionado o critério de inserção (linha 5). A solução inicial é gerada pela EIP (linha 9) usando o critério previamente selecionado. No critério CIMBV, o custo de inserção de um cliente não roteado $h \in LC$ em uma determinada rota, associada a um veículo k , é expresso pela Equação (34), na qual a função $g(h)$ representa o custo de inserção. O valor de $g(h)$ é calculado pela soma de dois termos, o primeiro termo computa o custo de inserção do cliente h entre cada par de clientes adjacentes i e j da rota, enquanto o segundo define a penalização usada para desestimular inserções tardias de clientes localizados distantes. O fator γ define o peso do custo de ida e volta ao depósito.

$$g(h) = (c_{ih}^k + c_{hj}^k - c_{ij}^k) - \gamma(c_{0h}^k + c_{h0}^k) \quad (34)$$

Algoritmo 1: GeraSoluçãoInicial(v)

Entrada: Lista com todos os clientes (v),
Saída: Solução S

```

1 início
2   Inicializa LC
3    $S \leftarrow \{s^1, \dots, s^v\}$  o conjunto composto por  $v$  rotas vazias
4   repita
5      $s^{v'} \leftarrow k \in LC$  selecionado aleatoriamente
6     Atualize a LC //  $LC \leftarrow LC \setminus \{k\}$ 
7   até  $v' \leftarrow v$ ;
8   CritérioInserção  $\leftarrow$  CIMBV ou CIMPV // Selecionando aleatoriamente
9    $S \leftarrow$  EstratégiaInserçãoParalela ( $s, v, LC, CritérioInserção$ )
10 retorna  $S$ 

```

O critério CIMPV calcula a distância entre um cliente $h \in LC$ e todos os clientes que já tenham sido incluídos na solução parcial, ou seja $g(h) = c_{ih}^k$ Equação (35). Assume-se que a inserção de h é sempre realizada após i . Nos dois critérios adotados o cliente i_{min} inserido é aquele associado ao menor custo, isto é, $i_{min} \leftarrow \operatorname{argmin}\{g(h) | h \in LC\}$.

$$g(h) = c_{ih}^k \quad (35)$$

4.2. Vizinhanças Intrarrotas

Para melhorar os custos de transporte em cada rota, duas estruturas de vizinhança intrarota foram aplicadas. São elas:

- **Exchange:** Permuta o posicionamento de dois clientes.
- **2-opt:** Dois arcos não adjacentes (i_1, j_1) e (i_2, j_2) são removidos. Em seguida, dois novos arcos (i_1, i_2) e (j_1, j_2) são inseridos.

4.3. Vizinhanças Inter-rotas

Com o intuito de reduzir os custos de transporte entre clientes de rotas distintas, foram implementadas duas estruturas de vizinhança inter-rotas. São elas:

- **Swap(1,1):** Permuta dois clientes de rotas distintas.
- **Shift(1,0):** Um cliente é removido de uma rota e inserido em outra.

4.4. Mecanismo de Perturbação

Uma perturbação é selecionada aleatoriamente durante a execução. Logo, a perturbação aplicada possivelmente poderá piorar a solução, porém permitindo explorar mais espaços de busca.

- **Multi-Swap(1,1) e Multi-Shift(1,0):** Múltiplos movimentos **Multi-Swap(1,1)** e **Multi-Shift(1,0)**. O número de movimentos executados é selecionado aleatoriamente no conjunto $\{1, \dots, 4\}$.
- **Merge:** Duas rotas são concatenadas em uma nova rota.

4.5. Vizinhanças para o Split Delivery

- **k-Split:** Divide a carga de um cliente entre vários veículos. Remove um cliente i de todas as rotas em que ele aparece e o realoca de tal modo que ele seja inserido na melhor posição, em cada rota que possuir capacidade residual para atender parte da sua demanda.
- **EmptyRoutes:** É utilizado para reduzir o número de rotas de uma solução.



4.6. Iterated Local Search

A metaheurística ILS, com comportamento *Multi-Start* (MS-ILS), é um método de trajetória que explora o espaço de soluções através da aplicação sucessiva de perturbações a ótimos locais [Loureço et al., 2010]. Estes ótimos locais são obtidos a partir do algoritmo de busca local, que por sua vez, tem como ponto de partida uma solução inicial proveniente do procedimento construtivo ou de uma solução perturbada. Com a finalidade de melhorar a solução corrente ao longo das iterações, o RVND (do inglês, *Randomized Variable Neighborhood Descent*) foi utilizado como método da busca local, o RVND é um método iterativo que aplica de forma aleatória um conjunto de vizinhanças. Esta abordagem do ILS com RVND, utilizada em [Subramanian et al., 2010], [Penna et al., 2013] e [Silva et al., 2015], produz, em média, resultados melhores do que a versão com ordem determinística. Por fim, o pseudocódigo do ILS é descrito no Algoritmo 2.

Algoritmo 2: $ILS(S, maxIT, msIT, maxT)$	Algoritmo 3: $BuscaLocal(S, IT)$
<pre> 1 início 2 repita 3 $S \leftarrow GeraSolucaoInicial()$ 4 $S \leftarrow BuscaLocal(S, maxIT)$ 5 se $f(S) < f(S^*)$ então 6 $S^* \leftarrow S$ $it \leftarrow 0$ 7 else 8 $it \leftarrow it + 1$ 9 $k \leftarrow k + 1$ 10 até $k \leq maxIT$ & $it \leq msIT$ & $T(maxT)$; 11 retorna S^* </pre>	<pre> 1 $S' \leftarrow S$ 2 $k \leftarrow 0$ 3 início 4 repita 5 $S \leftarrow RVND(S)$ 6 se $f(S) < f(S')$ então 7 $S' \leftarrow S$ 8 $k \leftarrow 0$ 9 $S \leftarrow Pertuba(S')$ 10 $k \leftarrow k + 1$ 11 até $k \leq IT$; 12 retorna S' </pre>

A heurística ILS descrita no Algoritmo 2 é formado por alguns procedimentos básicos, o procedimento **GeraSolucaoInicial** () (linha 3) é responsável pela construção de soluções iniciais. O segundo procedimento, **BuscaLocal** () (linha 4), é responsável por explorar o espaço de busca e encontrar bons ótimos locais. O terceiro procedimento, **Perturba**() (Algoritmo 3, linha 9), gera soluções modificadas de tal forma que o novo ótimo local obtido após a busca local seja diferente. O último elemento, identificado pelo procedimento **CriterioAceitacao**() (linha 5), é responsável por determinar se a solução proveniente da busca local será descartada ou utilizada na próxima iteração. Por fim, o processo segue até que um número máximo de iterações ($maxIT$, linha 10), iterações sem melhora ($msIT$ - linha 10) ou até um tempo limite ($maxT$ - linha 10).

5. Testes Computacionais

O algoritmo proposto foi desenvolvido na linguagem C++ utilizando o compilador g++ 4.8.2 e executado em um computador com *thread* única, processador Intel®Core™i7-870 CPU @ 2.93 GHz, 8.0 GB de memória RAM e sob a plataforma de um Sistema Operacional Linux (Ubuntu 14.04 LTS x64). A nomenclatura para as tabelas é a seguinte: **EIP*** é o valor médio da função objetivo obtida por meio do *Método Construtivo*. O ILS é dado pela sigla **ILS***, sendo este a média dos resultados obtidos para a execução de 5 casos de teste por instância. O **Gap₁** é o intervalo entre o limite superior **EIP*** e o inferior **ILS**, o **Gap₁** é dado por: $Gap_1 = \frac{100 * (ILS)}{(EIP)} - 100$, já o **Gap₂** é dado por: $Gap_2 = \frac{100 * (CPLEX)}{(ILS)} - 100$; **MSC** é a melhor solução conhecida na literatura; **Prec. ILS** e **Prec. CPLEX** representam as precisões do **ILS** e **CPLEX** em relação à **MSC**.

Os parâmetros do algoritmo **ILS** foram baseados no trabalho de [Xiao e Konak, 2016], porém a quantidade de iterações sem melhora do algoritmo **ILS** foi definida de forma empírica. Assim, definiu-se 5 casos de teste por instância, tempo limite ($maxT$) de 7200 segundos, $maxIT = n$, $msIt = 20$. Para a experimentação do método, foi utilizado a biblioteca *Mersenne Twist*² de números aleatórios, considerou-se no algoritmo, uma *seed* (semente) por execução de instância. O conjunto de instâncias incluem sete grupos de instâncias, cada grupo de instância contém vinte instâncias. Por praticidade, apenas a média por grupo de instância foram dispostas em algumas tabelas. As instâncias estão formatadas da seguinte maneira: UK-nxm-i, onde n é o número de clientes (10, 15, 25, ..., 100), m é o número de períodos, e i é o ID da instância (de 1 a 20).

² *Mersenne Twist v. 1.5*, visualizado em 23.03.2017



Vale ressaltar que, para ambas as heurísticas, o mesmo método construtivo foi utilizado. Outro ponto a se considerar é que, a versão exata sem *Split* segue a mesma modelagem proposta em [Xiao e Konak, 2016], desta forma, os resultados obtidos pelo método exato foram semelhantes, perdendo apenas em algumas instâncias devido a erros de aproximação (vide coluna **CPLEX^e**).

A Tabela 2 faz uma breve descrição das médias obtidas para os sete grupos de instâncias. Cada linha da tabela representa a média dos custos para cada grupo de instância avaliando apenas o método heurístico proposto. Em [Xiao e Konak, 2016], o problema é abordado com o uso de três heurísticas distintas. A média dos resultados das três estão traduzidas nas colunas **MSC₁** (*Nearest-Neighbor*), **MSC₂** (*Savings heuristic*) e **MSC₃** (*Insertion heuristic*).

Min CO ₂	Green Vehicle Routing and Scheduling Problem						[Xiao e Konak, 2016]						
	GVRSP			GVRSP+Split			GAP			Precisão (Prec.)			
Grupo de Instâncias	EIP*	ILS*	CPLEX	EIP*	ILS*	CPLEX	MSC ₁	MSC ₂	MSC ₃	ILS ^a	ILS ^b	CPLEX ^a	CPLEX ^b
10 × 5	962.14	597.93	557.93	962.14	541.38	522.33	635.9	637.5	767.1	-5.97%	-14.86%	-12.26%	-17.86%
15 × 5	1252.31	705	-	952.31	697.2	-	855.3	719.9	790.2	-2.07%	-3.15%	-	-
20 × 5	1142.17	1078.1	-	1142.17	1033.9	-	1173.8	1119	1084.2	-0.56%	-4.64%	-	-
25 × 5	1219.93	1054.28	-	1219.93	1017.56	-	1355.2	1066.3	1187	-1.13%	-4.57%	-	-
50 × 5	2135	2066	-	2135	2024.3	-	2584.6	2080.3	2321.2	-0.69%	-2.69%	-	-
75 × 5	3624.8	2932.3	-	3624.8	2878.06	-	3806.5	2977.4	3308.9	-1.51%	-3.34%	-	-
100 × 5	3797.25	3748.5	-	3797.25	3662.8	-	4794.3	3766.3	4311.8	-0.47%	-2.75%	-	-
Médias										-1.77%	-5.14%		

^a Precisão para a variante sem *Split*

^b Precisão para a variante com *Split*

Índice dos resultados:

prec: prec == 0.0%

prec: prec > 0.0%

prec: prec < 0.0%

Tabela 2: Resultados obtidos para 5 casos de teste do *PRPLIB* - [Xiao e Konak, 2016].

Observe que a heurística **ILS** ganha em comparação com todas as heurísticas usadas no artigo, isso ocorre tanto na versão com e sem *Split*. Todos os resultados relacionados na Tabela 2 são comparados com o melhor resultado dentre as três heurísticas do artigo. Ainda sobre os resultados reportados na Tabela 2, a heurística proposta reduziu os custos em -1.7% na sua versão sem *Split* e -5.14% com *Split*, demonstrando assim, que houve bons resultados na média geral.

7. Conclusões e Trabalhos Futuros

Este artigo apresentou uma nova abordagem para o Problema de Roteamento voltado a emissões de carbono, usando uma metodologia baseada na metaheurística *Iterated Local Search* com uma busca local intensiva através de várias vizinhanças clássicas para problemas de roteamento. Os testes realizados provaram que a metodologia proposta foi capaz de encontrar soluções de boa qualidade, em muitos casos reduzindo o custo. Avaliando a heurística sem *Split*, das 20 instâncias de tamanho 10 testadas, 15 apresentaram melhora e 5 empataram, um ganho em 75% das instâncias. Mesmo o **ILS** sem *Split* obtendo soluções próximas da média dos 7.2% em relação a melhor solução conhecida, o mesmo torna-se atrativo uma vez que as suas sucessivas iterações tendem a convergir para bons resultados com um baixo tempo computacional.

Como estudo inicial, o algoritmo provou ser promissor na exploração dessa variante do GVRSP, entretanto mais estudos são necessários. Além disso, existe a possibilidade de estender essa metodologia para outras variantes do GVRSP, por exemplo, o GVRSP definido em um grafo não completo, que são as de maior aplicação prática. Outra possibilidade seria definir a velocidade como uma variável de decisão para avaliar a melhor velocidade dentro de um arco, uma versão sem paradas nos arcos. Em outro cenário, pode-se avaliar a versão em que o peso da gasolina ao longo do trajeto é avaliado como uma variável de decisão.

Como proposta de trabalhos futuros, pretende-se trabalhar com técnicas construtivas mais eficientes, técnicas capazes de gerar boa variabilidade nas soluções sem perda de qualidade. Desta forma, é interessante realizar um estudo mais amplo do impacto dos parâmetros do algoritmo e das vizinhanças, podendo inclusive realizar adição de novas estruturas ou até mesmo a remoção de estruturas não muito eficientes para o problema. Outra proposta, é a aplicação de variações híbridas do algoritmo como recurso de pré-otimização em métodos exatos.

Agradecimentos

Os autores gentilmente agradecem o fornecimento de recursos computacionais por parte do Laboratório de Algoritmos, Grafos e Otimização (LAGO - UFF) da Universidade Federal Fluminense e o apoio fornecido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).



Referências

- Archetti, C. e Speranza, M. G. (2008). The split delivery vehicle routing problem: A survey. In *The vehicle routing problem: Latest advances and new challenges*, p. 103–122. Springer. (Citado nas páginas 2, 4)
- Barth, M. e Boriboonsomsin, K. (2008). Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2058(2058):163–171. (Citado na página 7)
- Barth, M., Younglove, T., e Scora, G. (2005). Development of a heavy-duty diesel modal emissions and fuel consumption model. *California Partners for Advanced Transit and Highways (PATH)*. (Citado na página 7)
- Bektaş, T. e Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250. (Citado na página 3)
- Blum, C. e Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308. (Citado na página 2)
- Bodin, L. e Golden, B. (1981). Classification in vehicle routing and scheduling. *Networks*, 11(2):97–108. (Citado na página 3)
- Chen, H.-K., Hsueh, C.-F., e Chang, M.-S. (2006). The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(5):383–408. (Citado na página 2)
- Cohen, N. (2011). *Green cities: An A-to-Z guide*, volume 4. Sage. (Citado na página 3)
- Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91. (Citado nas páginas 2, 3)
- Dror, M. e Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 23(2):141–145. (Citado na página 2)
- Eksioglu, B., Vural, A. V., e Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483. (Citado na página 3)
- Figliozzi, M. (2010). Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board*, 2197(2197):1–7. ISSN 03611981. (Citado na página 3)
- Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T., e Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293. (Citado na página 3)
- Golden, B. L., Raghavan, S., e Wasil, E. A. (2008). *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer Science & Business Media. (Citado na página 3)
- Jabali, O., Woensel, T., e de Kok, A. (2012). Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074. (Citado na página 3)
- Kara, I., Kara, B. Y., e Yetis, M. K. (2007). Energy minimizing vehicle routing problem. In *International Conference on Combinatorial Optimization and Applications*, p. 62–71. Springer. (Citado na página 3)
- Kramer, R., Subramanian, A., Vidal, T., e LucÁdio dos Anjos, F. C. (2015). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539. (Citado na página 3)
- Laporte, G., Gendreau, M., Potvin, J.-Y., e Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International transactions in operational research*, 7(4-5):285–300. (Citado na página 3)
- Lenstra, J. K. e Kan, A. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227. LENSTRA, J. K. and RINNOOY KAN, A. H. G. (Citado na página 3)
- Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., e Lam, H. Y. (2014). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*, 41(4 PART 1):1118–1138. ISSN 09574174. (Citado na página 3)
- Lourenço, H. R., Martin, O. C., e Stützle, T. (2010). Iterated local search: Framework and applications. In *Handbook of Metaheuristics*, p. 363–397. Springer. (Citado na página 9)
- Maden, W., Eglese, R., e Black, D. (2010). Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society*, 61(3):515–522. (Citado na página 3)
- Penna, P. H. V., Subramanian, A., e Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, p. 1–32. (Citado nas páginas 2, 9)
- Silva, M. M., Subramanian, A., e Ochi, L. S. (2015). An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research*, 53:234–249. (Citado nas páginas 2, 9)
- Solomon, M. M. (1984). Vehicle routing and scheduling with time window constraints: Models and algorithms. Technical Report 01451798, University of Pennsylvania. (Citado na página 3)
- Soysal, M., Bloemhof-Ruwaard, J. M., e Bektaş, T. (2015). The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics*, 164:366–378. (Citado na página 3)
- Subramanian, A., Drummond, L. M. d. A., Bentes, C., Ochi, L. S., e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911. (Citado na página 9)
- Toth, P. e Vigo, D. (2002). Chapter 1: An overview of vehicle routing problems. *The Vehicle Routing Problem. Society for Industrial and Applied Mathematics Philadelphia, PA, USA*. (Citado na página 3)
- Xiao, Y. e Konak, A. (2015a). Green vehicle routing problem with time-varying traffic congestion. In *Proceedings of the 14th INFORMS Computing Society Conference*, p. 134–148. (Citado na página 3)
- Xiao, Y. e Konak, A. (2015b). A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness. *Applied Soft Computing*, 34:372 – 388. ISSN 1568-4946. (Citado na página 3)
- Xiao, Y. e Konak, A. (2016). The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E: Logistics and Transportation Review*, 88:146–166. ISSN 13665545. (Citado nas páginas 3, 4, 7, 9, 10 e 11)