

Incorporando Mineração de Dados a uma Heurística *Multi-Start* ILS para o Problema de Roteamento de Veículos com Frota Heterogênea

Marcelo Rodrigues de Holanda Maia

Instituto de Computação - Universidade Federal Fluminense (UFF)
Av. General Milton Tavares de Souza, s/nº - Boa Viagem - Niterói/RJ - Brasil - CEP: 24210-346
mmaia@ic.uff.br

Instituto Brasileiro de Geografia e Estatística (IBGE)
R. General Canabarro, 706 - Maracanã - Rio de Janeiro/RJ - Brasil - CEP: 20271-205
marcelo.h.maia@ibge.gov.br

Alexandre Plastino

Instituto de Computação - Universidade Federal Fluminense (UFF)
Av. General Milton Tavares de Souza, s/nº - Boa Viagem - Niterói/RJ - Brasil - CEP: 24210-346
plastino@ic.uff.br

Puca Huachi Vaz Penna

Instituto do Noroeste Fluminense de Educação Superior - Universidade Federal Fluminense (UFF)
Av. João Jasbick, s/nº - Aeroporto - Santo Antônio de Pádua/RJ - Brasil - CEP: 28470-000
ppenna@ic.uff.br

RESUMO

Versões híbridas de metaheurísticas que incorporam técnicas de mineração de dados têm sido aplicadas a diversos problemas de otimização, alcançando resultados promissores. Neste trabalho, técnicas de mineração de dados são incorporadas a uma heurística *multi-start* para o problema de roteamento de veículos com frota heterogênea baseada na metaheurística ILS que usa um procedimento RVND na fase de busca local. Os resultados obtidos nos experimentos computacionais realizados mostram que as heurísticas híbridas propostas apresentam desempenhos superiores em relação à heurística original, alcançando soluções melhores com tempo de execução menor.

PALAVRAS CHAVE. Metaheurística Híbrida, Problema de Roteamento de Veículos com Frota Heterogênea, Mineração de Dados.

Área Principal: Metaheurísticas

ABSTRACT

Hybrid versions of metaheuristics that incorporate data mining techniques have been applied to various optimization problems, achieving promising results. In this work, data mining techniques are incorporated into a multi-start heuristic for the heterogeneous fleet vehicle routing problem based on ILS metaheuristic which uses a RVND procedure in the local search phase. The results obtained in computational experiments show the proposed hybrid heuristics present superior performance over the original heuristic, reaching better solutions with shorter run time.

KEYWORDS. Hybrid Metaheuristic. Heterogeneous Fleet Vehicle Routing Problem. Data Mining.

Main Area: Metaheuristics

1. Introdução

Nas últimas décadas, metaheurísticas têm sido propostas e aplicadas com sucesso a problemas de otimização de várias áreas, encontrando soluções satisfatórias em um tempo computacional aceitável. Técnicas de outras áreas têm sido incorporadas a metaheurísticas tradicionais com o propósito de obter resultados ainda melhores, dando origem a metaheurísticas híbridas. Um exemplo bem sucedido desse tipo de hibridização ocorreu com a metaheurística *Greedy Randomized Adaptive Search Procedures* (GRASP), proposta em (Feo e Resende, 1995), à qual foram incorporadas técnicas de mineração de dados.

A primeira proposta de hibridização do GRASP com técnicas de mineração de dados (Ribeiro *et al.*, 2006) teve como resultado uma heurística híbrida chamada *Data Mining GRASP* (DM-GRASP). A ideia básica era que padrões encontrados em boas soluções poderiam ser usados para guiar a busca, levando a uma exploração mais efetiva do espaço de soluções. Nessa versão híbrida do GRASP, após a execução de metade do número total de iterações, um procedimento de mineração de dados extrai padrões de um conjunto elite composto pelas melhores soluções encontradas até aquele momento. Esses padrões apresentam características comumente encontradas nas melhores soluções e podem ser usados para guiar a busca na metade seguinte das iterações.

Uma revisão da concepção do DM-GRASP e dos resultados de suas aplicações aos problemas de empacotamento de conjuntos, da diversidade máxima e de replicação de servidores para transmissão *multicast* confiável é apresentada em (Santos *et al.*, 2008). Essa abordagem também foi aplicada com sucesso aos problemas das *p*-medianas (Plastino *et al.*, 2011), de síntese de redes com 2-caminhos (Barbalho *et al.*, 2013) e do caixeiro viajante com coleta e entrega envolvendo um único tipo de produto (Guerine *et al.*, 2014).

Posteriormente, uma nova versão da heurística DM-GRASP, chamada *Multi DM-GRASP* (MDM-GRASP), foi proposta. A principal ideia dessa proposta, descrita em (Plastino *et al.*, 2014), é executar o procedimento de mineração de dados múltiplas vezes e de um modo adaptativo. O procedimento é executado primeiramente assim que o conjunto elite se torna estável – o que é caracterizado pela não ocorrência de alterações no conjunto elite no decorrer de um dado número de iterações – e, posteriormente, a cada vez em que o conjunto elite tiver sido alterado e novamente se tornado estável.

O MDM-GRASP também foi aplicado aos problemas das *p*-medianas (Plastino *et al.*, 2011), de síntese de redes com 2-caminhos (Barbalho *et al.*, 2013) e de replicação de servidores para transmissão *multicast* confiável (Plastino *et al.*, 2014), alcançando resultados ainda melhores que os obtidos pelo DM-GRASP, não apenas em termos de qualidade das soluções, mas também em relação ao tempo computacional.

Os resultados promissores alcançados nas diversas aplicações do DM-GRASP inspiraram a incorporação das mesmas técnicas de mineração de dados a algoritmos baseados em outras metaheurísticas. Essa abordagem foi usada para conceber versões aperfeiçoadas de um algoritmo baseado na metaheurística *Iterated Local Search* (ILS) – descrita em (Lourenço *et al.*, 2003) – para o problema de cobertura de conjuntos por pares (Yusim *et al.*, 2013) e de uma heurística *multi-start* híbrida que combina elementos de várias metaheurísticas tradicionais proposta para o problema das *p*-medianas (Martins *et al.*, 2014), ambas com bons resultados.

Neste trabalho, as técnicas de mineração de dados usadas nas heurísticas DM-GRASP e MDM-GRASP são incorporadas a um algoritmo para o problema de roteamento de veículos com frota heterogênea (PRVFH) baseado em uma heurística *multi-start* ILS (MS-ILS) que usa um procedimento *Random Variable Neighborhood Descent* (RVND) na fase de busca local (Penna *et al.*, 2013) com o objetivo de aprimorá-lo, explorando os potenciais benefícios providos por essa abordagem, demonstrados em trabalhos anteriores. Os resultados obtidos nos experimentos computacionais realizados reforçam a efetividade da abordagem, uma vez que as novas heurísticas híbridas propostas apresentam desempenhos superiores em relação à heurística original, alcançando soluções melhores com tempo de execução menor.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta o PRVFH, além de uma breve revisão dos principais trabalhos que abordam esse problema. A Seção 3 descreve os algoritmos híbridos propostos neste trabalho, explicando o processo de incorporação de mineração de dados ao algoritmo original. A Seção 4 apresenta os experimentos computacionais realizados para avaliar e comparar os desempenhos dos algoritmos, bem como os resultados obtidos. Por fim, na Seção 5, são apresentadas as conclusões deste trabalho e relacionadas ideias para trabalhos futuros.

2. Descrição do Problema

O problema de roteamento de veículos com frota heterogênea (PRVFH) é uma generalização do problema de roteamento de veículos (PRV) – portanto, pertencente à classe de problemas NP-difícil – na qual os clientes são servidos por uma frota heterogênea de veículos, com capacidades e custos distintos, ao invés de uma frota homogênea. Essa situação está mais próxima daquelas encontradas em aplicações do mundo real e o PRVFH modela esse tipo de aplicações.

O PRVFH é descrito como a seguir. Seja $G = (V, A)$ um grafo direcionado onde $V = \{0, 1, \dots, n\}$ é um conjunto composto por $n + 1$ vértices e $A = \{(i, j) : i, j \in V, i \neq j\}$ é o conjunto de arestas. O vértice 0 representa o depósito, onde se localiza a frota de veículos, enquanto o conjunto $V' = V - \{0\}$ é composto pelos vértices restantes, que representam os n clientes. Cada cliente $i \in V'$ tem uma demanda não negativa q_i . A frota é composta por m tipos diferentes de veículos, com o conjunto de tipos de veículo $M = \{1, \dots, m\}$. Para cada tipo de veículo $u \in M$, há m_u veículos disponíveis, cada um com uma capacidade Q_u e um custo fixo f_u . Finalmente, para cada combinação entre uma aresta $(i, j) \in A$ e um tipo de veículo $u \in M$ há um custo associado $c_{ij}^u = d_{ij}r_u$, onde d_{ij} é a distância entre os vértices i e j , e r_u é um custo variável por unidade de distância associado ao tipo de veículo u .

Uma rota é definida por um par (R, u) , com $R = (i_1, i_2, \dots, i_{|R|})$, $i_1 = i_{|R|} = 0$ e $\{i_2, \dots, i_{|R|-1}\} \subseteq V'$, isto é, cada rota é um circuito em G , que começa e termina no depósito, associado a um veículo de um tipo $u \in M$. Uma rota (R, u) é factível se a soma das demandas dos clientes em R não excede a capacidade Q_u do veículo associado a ela. O custo associado a uma rota é a soma do custo fixo do veículo correspondente e dos custos variáveis associados à combinação entre o tipo de veículo e cada uma das arestas percorridas. Desse modo, o PRVFH consiste em encontrar rotas factíveis de modo que cada cliente seja visitado exatamente uma vez, o número total de rotas associadas a veículos de um tipo $u \in M$ não exceda m_u e a soma dos custos associados às rotas seja minimizado.

Existem algumas variantes do PRVFH descritas na literatura. Basicamente, elas estão relacionadas à limitação da frota (limitada ou ilimitada) e aos tipos de custo considerados (fixo, variável, ou ambos). Por consistir em determinar a melhor composição da frota, além de seu melhor esquema de roteamento, a variante com frota ilimitada (Golden *et al.*, 1984) é conhecida como PRV com dimensionamento de frota heterogênea (PRVDFH). Já a variante com frota limitada (Taillard, 1999), chamada PRV com frota heterogênea fixa (PRVFHF), consiste em otimizar o roteamento de uma frota fixa disponível.

Dada a importância prática do PRVFH, ele foi intensivamente abordado nas últimas décadas, com diversos algoritmos propostos. Os melhores algoritmos conhecidos apresentados na literatura são listados a seguir. Para o PRVDFH: um algoritmo de geração de colunas baseado em uma formulação do PRVDFH como um problema de cobertura de conjuntos (Choi e Tcha, 2007); um algoritmo genético híbrido que aplica uma busca local como um método de mutação (Liu *et al.*, 2009); um algoritmo de busca tabu determinístico que usa diferentes procedimentos para gerar soluções iniciais (Brandão, 2009); uma heurística *Variable Neighborhood Search* (VNS) que usa um procedimento baseado nos algoritmos de varredura e de Dijkstra para obter uma solução inicial e várias estruturas de vizinhança na fase de busca local (Imran *et al.*, 2009); um algoritmo memético (Prins, 2009); e um algoritmo baseado na metaheurística ILS que usa um procedimento RVND na fase de busca local (Penna *et al.*, 2013). Para o PRVFHF: um algoritmo de geração

de colunas baseado em um procedimento de memória adaptativa que usa uma busca tabu embutida (Taillard, 1999); um algoritmo baseado no método *threshold accepting*, uma variante determinística da metaheurística *simulated annealing* (Tarantilis *et al.*, 2004); um algoritmo baseado em outra variante determinística de *simulated annealing* chamada *record-to-record travel* (Li *et al.*, 2007); um algoritmo memético (Prins, 2009); e um algoritmo baseado na metaheurística ILS que usa um procedimento RVND na fase de busca local (Penna *et al.*, 2013).

3. Algoritmos Propostos

Os algoritmos propostos neste trabalho, que tratam tanto o PRVDFH quanto o PRVFHF, foram concebidos usando as abordagens aplicadas previamente para incorporar mineração de dados à metaheurística GRASP, resultando na construção de suas versões híbridas DM-GRASP e MDM-GRASP. Neste caso, essas abordagens foram aplicadas ao algoritmo proposto em (Penna *et al.*, 2013) para o PRVFH, baseado na metaheurística ILS. Esse algoritmo foi escolhido para servir como base para as heurísticas híbridas propostas por ter apresentado resultados bastante competitivos em comparação com os melhores algoritmos para o PRVFH. Além disso, por se basear em uma heurística *multi-start*, ele é bem adequado às abordagens de incorporação de mineração de dados usadas para o GRASP, que também é um método *multi-start*.

Uma estratégia *multi-start* consiste em reiniciar a busca múltiplas vezes a partir de novas soluções iniciais. Heurísticas *multi-start* (Martí, 2003) são métodos iterativos onde cada iteração tem duas fases: geração e busca local. Na primeira, uma solução inicial é gerada, enquanto na segunda a solução é tipicamente (mas não necessariamente) melhorada. Cada iteração produz uma solução (usualmente um ótimo local) e a melhor dentre todas as iterações é a saída do algoritmo.

A estratégia para o uso de mineração de dados aplicada ao GRASP – que pode ser generalizada para a classe de heurísticas *multi-start* – consiste em formar, no decorrer das iterações, um conjunto elite composto pelas melhores soluções encontradas e, após satisfeito um critério de interrupção (número de iterações no DM-GRASP e estabilização do conjunto elite no MDM-GRASP), executar um procedimento de mineração de dados que identifica padrões comuns às soluções desse conjunto elite, os quais passam a ser usados na fase de geração de soluções iniciais nas iterações seguintes, de modo a guiar a busca a partir de soluções mais promissoras, com a intenção de obter soluções de melhor qualidade com um tempo de convergência mais curto.

O procedimento de mineração de dados usado nessa abordagem se baseia na formulação do problema de mineração de conjuntos de itens frequentes, uma etapa do processo de mineração de regras de associação. Esse problema consiste em identificar conjuntos de itens frequentes em uma base de dados de transações, onde cada transação é um subconjunto do universo de itens do domínio da aplicação. Um conjunto de itens é considerado frequente se ele ocorre em pelo menos um determinado número – chamado suporte mínimo – de transações da base de dados. Neste contexto, o conjunto elite é a base de dados e cada solução é uma transação.

Mais especificamente, a estratégia adotada se baseia na mineração de conjuntos de itens frequentes maximais. Um conjunto de itens frequente é chamado de maximal se ele não tem nenhum superconjunto que também seja frequente. O algoritmo FPmax* (Grahne e Zhu, 2003)¹ é usado para realizar a mineração de itens frequentes maximais nessa abordagem.

A seguir são descritas a heurística MS-ILS proposta em (Penna *et al.*, 2013) para o PRVFH, na Seção 3.1, e as heurísticas híbridas propostas neste trabalho: DM-MS-ILS e MDM-MS-ILS, nas Seções 3.2 e 3.3, respectivamente.

3.1. Heurística MS-ILS para o PRVFH

Os passos da heurística MS-ILS² proposta em (Penna *et al.*, 2013) para o PRVFH são apresentados no Algoritmo 1. Para o PRVFHF, a frota é inicializada com os veículos disponíveis definidos pela instância do problema sendo processada. Já para o PRVDFH, a frota é inicializada

¹Implementação disponível em <http://fimi.cs.helsinki.fi>

²Os autores chamam essa heurística de ILS-RVND em (Penna *et al.*, 2013)

com um veículo de cada tipo (linha 1). O número de clientes é atribuído a n (linha 2) e o de veículos a v (linha 3). A heurística *multi-start* executa $MaxIter$ iterações (linhas 5-10). Em cada iteração, uma solução inicial é gerada por um método construtivo (linha 6). $MaxIterILS$ representa o número máximo permitido de perturbações consecutivas sem melhoras na heurística ILS. Esse valor é calculado com base nos números de clientes (n) e veículos (v), e em um parâmetro β (linha 7). Na fase de busca local, a heurística ILS é usada com o objetivo de melhorar a solução inicial gerada (linha 8). Caso a solução s' retornada pela heurística ILS represente uma melhoria em relação ao custo, dado pela função f , atualiza-se a melhor solução geral s^* (linha 9). Após a execução das $MaxIter$ iterações da heurística *multi-start*, a melhor solução encontrada é retornada (linha 11).

Algoritmo 1: MS-ILS($MaxIter, \beta$)

```

1 Inicializa frota;
2  $n \leftarrow$  número total de clientes;
3  $v \leftarrow$  número total de veículos;
4  $f(s^*) \leftarrow \infty$ ;
5 para  $i \leftarrow 1$  até  $MaxIter$  faça
6    $s \leftarrow$  GeraçãoSoluçãoInicial( $v$ );
7    $MaxIterILS \leftarrow$  CalculaMaxIterILS( $n, v, \beta$ );
8    $s' \leftarrow$  ILS( $s, MaxIterILS$ );
9   se  $f(s') < f(s^*)$  então  $s^* \leftarrow s'$ ;
10 fim
11 retorna  $s^*$ ;

```

O método construtivo para geração de soluções iniciais emprega a Estratégia de Inserção Paralela (EIP), usando um critério de inserção selecionado aleatoriamente entre os dois disponíveis, chamados Critério de Inserção Mais Barata Viável Modificada (CIMBVM) e Critério de Inserção Mais Próxima Viável (CIMPV).

O pseudocódigo desse método é apresentado no Algoritmo 2. Inicialmente, a lista de candidatos (LC) é preenchida com todos os clientes (linha 1) e a solução (s) é inicializada com uma rota vazia associada a cada veículo da frota (linha 2). Cada rota é preenchida com um cliente k selecionado aleatoriamente de LC (linhas 3-6). Um critério de inserção é escolhido aleatoriamente (linha 7). Uma solução inicial é gerada usando a Estratégia de Inserção Paralela e o critério de inserção selecionado (linha 8). Se a solução s gerada não é factível, o método construtivo é reiniciado (linha 9). Se a frota é ilimitada (PRVDFH), uma rota vazia associada a cada tipo de veículo é adicionada à solução s (linha 10). Essas rotas vazias são necessárias para permitir um possível redimensionamento da frota durante a fase de busca local. Finalmente, a solução gerada é retornada (linha 11).

Algoritmo 2: GERAÇÃO SOLUÇÃO INICIAL(v)

```

1 Inicializa  $LC$ ;
2 Seja  $s = \{s^1, \dots, s^v\}$  um conjunto composto por  $v$  rotas vazias;
3 para  $v' \leftarrow 1$  até  $v$  faça
4    $s^{v'} \leftarrow k \in LC$  selecionado aleatoriamente;
5    $LC \leftarrow LC - \{k\}$ ;
6 fim
7  $CritérioInser \leftarrow c \in \{CIMBVM, CIMPV\}$  selecionado aleatoriamente;
8  $s \leftarrow$  InserçãoParalela( $v, LC, CritérioInser$ );
9 se  $\neg factível(s)$  então Retorna à linha 1;
10 se frota ilimitada então Adiciona a  $s$  uma rota vazia para cada tipo de veículo;
11 retorna  $s$ ;

```

3.2. Heurística híbrida DM-MS-ILS para o PRVFH

A primeira das heurísticas híbridas propostas neste trabalho, chamada *Data Mining MS-ILS* (DM-MS-ILS), segue uma estrutura similar à usada na proposta do DM-GRASP (Santos *et al.*, 2008). Como foi explicado previamente na Seção 3, a estratégia usada nessa abordagem requer que as soluções para o problema sejam representadas como um conjunto de itens para que o procedimento de mineração de dados identifique padrões (conjuntos de itens frequentes maximais) comuns às melhores soluções. Diversos problemas de otimização apresentam soluções naturalmente representadas como conjuntos e a maior parte das aplicações da abordagem adotada neste trabalho foi para problemas desse tipo. Entretanto, para problemas de roteamento – como o PRV e suas variantes – as soluções são compostas por sequências de itens em que a ordem é relevante.

Uma aplicação da heurística DM-GRASP ao problema do caixeiro viajante com coleta e entrega envolvendo um único tipo de produto foi proposta em (Guerine *et al.*, 2014). Como esse problema é uma generalização do problema do caixeiro viajante, uma solução consiste em uma rota que passe por todos os clientes, respeitando as restrições de demanda e de capacidade do veículo. Naturalmente, a ordem é importante para a qualidade da rota e mesmo para sua viabilidade. Uma solução para esse problema é tipicamente representada como uma lista de clientes, ordenada de acordo com a sequência de visitação definida para a rota.

Nesse trabalho, os autores propuseram uma representação alternativa de modo a permitir o uso das técnicas empregadas no DM-GRASP. Cada par de clientes consecutivos i_r e i_{r+1} , $r = 1, 2, \dots, |R| - 1$, em uma rota R é representado por uma aresta (i_r, i_{r+1}) . Desse modo, a solução pode ser representada como um conjunto de arestas, preservando a ordem dos clientes na rota. Consequentemente, cada padrão encontrado pelo procedimento de mineração de dados consiste em um conjunto de arestas frequente nas soluções do conjunto elite. Um padrão pode conter duas (ou mais) arestas consecutivas (i_r, i_{r+1}) e (i_{r+1}, i_{r+2}) , que podem ser conectadas, formando segmentos de rota maiores. Se encararmos um padrão como um subgrafo possivelmente desconexo do grafo formado pelos clientes, as componentes conexas desse subgrafo serão os segmentos de rota formados no padrão. Assim, cada padrão é composto por um ou mais desses segmentos de rota.

Uma solução para o PRVFH consiste em um conjunto de rotas. Entretanto, uma modelagem direta para o problema de mineração de conjuntos de itens frequentes não seria adequada, pois resultaria em uma busca por conjuntos de rotas inteiras frequentes. Portanto, uma decomposição da solução com uma granularidade mais fina é necessária. Neste trabalho, uma modelagem semelhante à usada em (Guerine *et al.*, 2014) é empregada. Como descrito na Seção 2, cada rota no PRVFH é representada por um par (R, u) , onde $R = (i_1, i_2, \dots, i_{|R|})$ é uma lista de clientes, ordenada de acordo com a sequência de visitação definida, e u o tipo do veículo associado à rota. Na representação alternativa adotada, para cada rota (R, u) , a lista R é decomposta em um conjunto de arestas $\{(i_1, i_2), (i_2, i_3), \dots, (i_{|R|-1}, i_{|R|})\}$ como em (Guerine *et al.*, 2014). Em seguida, cada aresta desse conjunto é associada ao tipo de veículo u , resultando em um conjunto onde cada elemento é um par formado por uma aresta (i_r, i_{r+1}) , $r = 1, 2, \dots, |R| - 1$, em R e pelo tipo de veículo u , com o formato $\{((i_1, i_2), u), ((i_2, i_3), u), \dots, ((i_{|R|-1}, i_{|R|}), u)\}$. Consequentemente, neste caso, os padrões minerados também são formados por segmentos de rota, mas cada segmento de rota possui um tipo de veículo associado a ele.

O Algoritmo 3 apresenta o pseudocódigo da heurística híbrida DM-MS-ILS proposta. A principal diferença observada em comparação com o Algoritmo 1 é que as iterações da heurística *multi-start* são divididas em duas etapas, cada uma com metade do total de iterações. A primeira (linhas 6-12) corresponde à fase de geração do conjunto elite. As iterações dessa fase são idênticas às da heurística original, exceto pela atualização do conjunto elite E (linha 10). Após o término da primeira fase, a mineração de dados é realizada sobre o conjunto elite, retornando um conjunto de padrões P (linha 13). Em seguida, tem início a segunda fase, chamada de fase híbrida (linhas 14-20). Em cada iteração dessa fase, um padrão $p \in P$ é selecionado (linha 15) e usado na geração

de uma solução inicial, realizada por um método construtivo adaptado (linha 16). Após a geração da solução inicial, o algoritmo continua como na heurística original.

Algoritmo 3: DM-MS-ILS($MaxIter, \beta$)

```

1 Inicializa frota;
2  $n \leftarrow$  número total de clientes;
3  $v \leftarrow$  número total de veículos;
4  $f(s^*) \leftarrow \infty$ ;
5  $E \leftarrow \{\}$ ;
6 para  $i \leftarrow 1$  até  $MaxIter/2$  faça
7    $s \leftarrow$  GeraçãoSoluçãoInicial( $v$ );
8    $MaxIterILS \leftarrow$  CalculaMaxIterILS( $n, v, \beta$ );
9    $s' \leftarrow$  ILS( $s, MaxIterILS$ );
10  AtualizaConjuntoElite( $E, s'$ );
11  se  $f(s') < f(s^*)$  então  $s^* \leftarrow s'$ ;
12 fim
13  $P \leftarrow$  Minera( $E$ );
14 para  $i \leftarrow 1$  até  $MaxIter/2$  faça
15    $p \leftarrow$  SelecionaPadrão( $P$ );
16    $s \leftarrow$  GeraçãoHíbridaSoluçãoInicial( $v, p$ );
17    $MaxIterILS \leftarrow$  CalculaMaxIterILS( $n, v, \beta$ );
18    $s' \leftarrow$  ILS( $s, MaxIterILS$ );
19   se  $f(s') < f(s^*)$  então  $s^* \leftarrow s'$ ;
20 fim
21 retorna  $s^*$ ;

```

O pseudocódigo do método construtivo usado na fase híbrida, que usa um dos padrões minerados na geração de uma solução inicial, é apresentado no Algoritmo 4.

Algoritmo 4: GERAÇÃOHÍBRIDASOLUÇÃOINICIAL(v, p)

```

1 Inicializa  $LC$ ;
2 Seja  $s = \{s^1, \dots, s^v\}$  um conjunto composto por  $v$  rotas vazias;
3 se frota ilimitada então
4   Remove de  $LC$  todos os clientes em  $p$ ;
5   para cada  $r' \in p$  faça  $s \leftarrow s \cup \{r'\}$ ;
6 senão
7   para cada  $r' \in p$  faça
8      $v' \leftarrow$  BuscaRotaVazia( $s, V(r')$ );
9     se  $v' \neq 0$  então
10    |  $s^{v'} \leftarrow r'$ ;
11    | Remove de  $LC$  todos os clientes em  $r'$ ;
12    fim
13  fim
14 fim
15 para  $v' \leftarrow 1$  até  $v$  faça
16    $s^{v'} \leftarrow k \in LC$  selecionado aleatoriamente;
17    $LC \leftarrow LC - \{k\}$ ;
18 fim
19  $\vdots$ 
23 retorna  $s$ ;

```

A única diferença em comparação com o Algoritmo 2 está no trecho adicionado para introduzir o padrão p na solução gerada s (linhas 3-14), imediatamente após a inicialização da lista de candidatos (LC) e de s . A introdução do padrão é realizada de maneiras diferentes dependendo da variante do PRVDFH. Para o PRVDFH, em que a frota é ilimitada, todos os clientes que ocorrem no padrão p são removidos de LC (linha 4). Em seguida, cada segmento de rota $r' \in p$ é adicionado à solução s como uma nova rota (linha 5). No caso do PRVDFH, em que a frota é fixa, para cada segmento de rota $r' \in p$, o algoritmo busca por uma rota vazia em s associada a um veículo do mesmo tipo associado a r' , dado por V (linha 8). Se houver tal rota, todo o segmento r' será incluído nela (linha 10) e todos os clientes que ocorrem nesse segmento serão removidos de LC (linha 11). Pode haver mais segmentos de rota associados a um determinado tipo de veículo que o número de veículos desse tipo disponível na frota fixa. Nesse caso, apenas os maiores segmentos de rota são adicionados diretamente à solução. Os clientes que ocorrem nos demais segmentos permanecem em LC e são adicionados à solução posteriormente. Após a introdução do padrão p na solução, o algoritmo continua como na heurística original (linhas 19-22 suprimidas).

3.3. Heurística híbrida MDM-MS-ILS para o PRVDFH

A segunda heurística híbrida proposta neste trabalho, cujo pseudocódigo é apresentado no Algoritmo 5, chama-se *Multi DM-MS-ILS* (MDM-MS-ILS). Essa heurística segue uma estrutura similar à usada na proposta do MDM-GRASP (Plastino *et al.*, 2014).

Algoritmo 5: MDM-MS-ILS($MaxIter, \beta$)

```

1 Inicializa frota;
2  $n \leftarrow$  número total de clientes;
3  $v \leftarrow$  número total de veículos;
4  $f(s^*) \leftarrow \infty$ ;
5  $E \leftarrow \{\}$ ;
6  $P \leftarrow \{\}$ ;
7 para  $i \leftarrow 1$  até  $MaxIter$  faça
8   se  $Estável(E)$  então  $P \leftarrow Minera(E)$ ;
9   se  $P = \{\}$  então  $s \leftarrow GeraçãoSoluçãoInicial(v)$ ;
10  senão
11     $p \leftarrow SeleccionaPadrão(P)$ ;
12     $s \leftarrow GeraçãoHíbridaSoluçãoInicial(v, p)$ ;
13  fim
14   $MaxIterILS \leftarrow CalculaMaxIterILS(n, v, \beta)$ ;
15   $s' \leftarrow ILS(s, MaxIterILS)$ ;
16   $AtualizaConjuntoElite(E, s')$ ;
17  se  $f(s') < f(s^*)$  então  $s^* \leftarrow s'$ ;
18 fim
19 retorna  $s^*$ ;

```

Nesse caso, a mineração de dados é realizada sempre que o conjunto elite E se torna estável, atualizando o conjunto de padrões P (linha 8). Nas primeiras iterações, quando a mineração de dados ainda não foi realizada nenhuma vez e, portanto, o conjunto de padrões P ainda está vazio, as soluções iniciais são geradas pelo Algoritmo 2 (linha 9). Após a mineração de dados ter sido realizada ao menos uma vez, as soluções iniciais são geradas pelo Algoritmo 4 usando um padrão p selecionado do atual conjunto de padrões P (linhas 10-13). Após a fase de busca local (linhas 14-15), o conjunto elite é atualizado (linha 16), assim como a melhor solução, em caso de melhora (linha 17).

4. Resultados Computacionais

Todas as heurísticas apresentadas na Seção 3 foram implementadas em C++ e testadas em um computador equipado com um processador Intel® Core™ i5 de 3,20 GHz e 1,56 GB de RAM

executando Windows 7 Pro SP1 de 32 bits.

Em todos os testes, foram considerados apenas os custos variáveis. Os parâmetros usados foram definidos como a seguir. O número de iterações *MaxIter* da heurística *multi-start* foi definido como 100. O parâmetro β , que corresponde ao nível de influência do número de veículos v no cálculo de *MaxIterILS* ($MaxIterILS = n + \beta v$), foi definido como 1. Para os parâmetros associados ao procedimento de mineração de dados, foram mantidos os mesmos valores usados em outras aplicações dessa abordagem: o tamanho máximo do conjunto elite, o suporte mínimo e o número de padrões usados foram definidos como 10, 2 e 10, respectivamente, e o número de iterações sem mudanças no conjunto elite necessário para considerá-lo estável foi definido como 5% de *MaxIter*.

Na Seção 4.1, apresentam-se os resultados para o PRVDFH, enquanto a Seção 4.2 apresenta os resultados para o PRVFHF. Em seguida, na Seção 4.3, os comportamentos da heurística original e das heurísticas híbridas propostas são analisados e comparados.

4.1. PRVDFH

Nos testes para o PRVDFH, foram usadas as mesmas instâncias dos testes realizados em (Penna *et al.*, 2013) – aquelas descritas em (Taillard, 1999), com número de clientes variando entre 20 e 100 – e um conjunto adicional de instâncias, descrito em (Brandão, 2011), com número de clientes variando entre 100 e 199. A Tabela 1 apresenta os resultados de cada heurística, reportando o custo da melhor solução, o valor médio das soluções e o tempo médio de processamento obtidos em dez execuções para cada instância. Os melhores valores na comparação entre os algoritmos são mostrados em negrito.

Tabela 1: Comparação entre MS-ILS, DM-MS-ILS e MDM-MS-ILS para o PRVDFH

Inst.	MS-ILS			DM-MS-ILS			MDM-MS-ILS		
	Melhor Solução	Solução Média	Tempo Médio	Melhor Solução	Solução Média	Tempo Médio	Melhor Solução	Solução Média	Tempo Médio
Tai/3	623,22	624,04	2,50	623,22	624,09	2,43	623,22	623,66	2,40
Tai/4	387,18	387,80	2,50	387,18	388,12	2,44	387,18	388,12	2,47
Tai/5	742,87	743,32	3,06	742,87	743,32	2,97	742,87	742,87	2,97
Tai/6	415,03	415,03	2,91	415,03	415,03	2,86	415,03	415,03	2,89
Tai/13	1495,43	1500,94	20,81	1491,86	1500,28	19,59	1491,86	1496,25	19,71
Tai/14	603,21	603,21	22,89	603,21	603,21	20,74	603,21	603,21	19,45
Tai/15	999,82	1003,25	19,67	999,82	1001,95	17,08	999,82	1001,42	16,87
Tai/16	1131,00	1135,91	17,66	1131,00	1134,10	17,19	1131,00	1132,66	16,98
Tai/17	1038,60	1043,18	73,49	1038,60	1042,96	69,07	1038,60	1040,81	74,24
Tai/18	1803,32	1814,31	80,78	1801,40	1812,01	80,90	1803,32	1816,55	83,46
Tai/19	1105,44	1112,25	165,65	1105,44	1110,99	161,26	1105,44	1109,38	154,97
Tai/20	1543,09	1549,20	151,08	1542,70	1547,86	142,22	1541,46	1548,52	131,51
Bra/N1	2236,86	2253,57	465,23	2243,76	2252,87	464,37	2245,38	2252,25	462,37
Bra/N2	2850,07	2871,72	1016,68	2838,35	2853,09	1008,30	2839,96	2852,90	1029,91
Bra/N3	2269,96	2316,61	366,85	2236,09	2282,18	362,39	2238,25	2272,86	365,76
Bra/N4	1822,78	1834,27	146,25	1823,04	1831,13	142,57	1825,82	1828,05	141,76
Bra/N5	2030,03	2043,79	626,54	2030,03	2041,84	577,19	2031,41	2040,23	560,57

A Tabela 1 mostra que as heurísticas obtiveram soluções médias iguais para duas das 17 instâncias (Tai/6 e Tai/14). Das 15 instâncias restantes, as heurísticas MDM-MS-ILS e DM-MS-ILS apresentaram melhores soluções médias para 12 e duas instâncias, respectivamente, enquanto a heurística MS-ILS obteve a melhor solução média para apenas uma (Tai/4). Considerando as maiores instâncias (aquelas com 100 clientes ou mais), o MDM-MS-ILS obteve melhores soluções médias para seis de um total de sete, enquanto a melhor solução média para a instância restante foi obtida pelo DM-MS-ILS. As heurísticas MDM-MS-ILS e DM-MS-ILS apresentaram o mesmo tempo médio para uma instância (Tai/5) e foram mais rápidos para nove e seis instâncias, respectivamente, enquanto o MS-ILS foi mais rápido para apenas uma (Tai/18).

Para avaliar a significância estatística das diferenças entre as heurísticas em relação às soluções médias, foi usado o teste t de Student. Mais especificamente, foi usado um teste pareado

por instância para cada par de heurísticas, com nível de significância $\alpha = 5\%$. Na comparação entre MS-ILS e DM-MS-ILS, houve significância estatística para três instâncias (Bra/N2, Bra/N3 e Bra/N4) – todas com 100 clientes ou mais. Na comparação entre MS-ILS e MDM-MS-ILS, houve significância estatística para três instâncias com 100 clientes ou mais (Bra/N2, Bra/N3 e Bra/N4) e para duas das instâncias menores (Tai/13 e Tai/16). Finalmente, na comparação entre DM-MS-ILS e MDM-MS-ILS, houve significância estatística para apenas uma instância (Tai/13).

4.2. PRVFHF

Nos testes para o PRVFHF foram usadas as instâncias descritas em (Brandão, 2011) e as descritas em (Li *et al.*, 2007), com número de clientes variando entre 200 e 360. A Tabela 2 apresenta os resultados obtidos por cada heurística, também em dez execuções para cada instância.

Tabela 2: Comparação entre MS-ILS, DM-MS-ILS e MDM-MS-ILS para o PRVFHF

Inst.	MS-ILS			DM-MS-ILS			MDM-MS-ILS		
	Melhor Solução	Solução Média	Tempo Médio	Melhor Solução	Solução Média	Tempo Médio	Melhor Solução	Solução Média	Tempo Médio
Bra/N1	2246,39	2261,41	633,57	2246,39	2258,65	620,09	2246,39	2258,65	605,16
Bra/N2	2898,17	2911,52	1382,33	2897,41	2910,72	1370,50	2891,17	2902,85	1327,38
Bra/N3	2378,99	2383,85	439,14	2378,99	2384,26	405,40	2378,99	2383,10	405,42
Bra/N4	1839,22	1839,23	197,03	1839,22	1839,22	176,89	1839,22	1839,42	168,55
Bra/N5	2047,81	2047,81	744,32	2047,81	2047,81	647,88	2047,81	2047,81	621,34
Li/H1	12050,39	12073,99	2330,32	12050,39	12062,22	2034,71	12050,39	12058,78	1953,79
Li/H2	10364,71	10410,56	4074,28	10342,83	10384,47	4015,73	10351,15	10378,21	4141,99
Li/H3	16270,47	16358,84	7659,96	16270,47	16305,38	7046,92	16239,48	16280,42	6815,66
Li/H4	17681,84	17882,54	13343,36	17720,66	17901,57	13058,05	17828,15	17921,21	13063,40
Li/H5	23770,01	24010,71	24568,82	23651,90	23960,35	24269,38	23640,07	23897,62	24656,98

Todas as heurísticas alcançaram a mesma solução média para uma das 10 instâncias (Bra/N5). MDM-MS-ILS e DM-MS-ILS apresentaram a mesma solução média para outra instância (Bra/N1). Das oito instâncias restantes, o MDM-MS-ILS obteve a melhor solução média para seis, enquanto DM-MS-ILS e MS-ILS para uma instância cada. Além disso, o MDM-MS-ILS foi mais rápido para seis instâncias, enquanto o DM-MS-ILS foi mais rápido para as quatro restantes.

Novamente, o teste t de Student foi usado para avaliar a significância estatística das diferenças entre as soluções médias obtidas. Na comparação entre MS-ILS e DM-MS-ILS, houve significância estatística para três instâncias (Li/H1, Li/H2 e Li/H3). Na comparação entre MS-ILS e MDM-MS-ILS, houve significância estatística para seis instâncias (Bra/N1, Bra/N2, Li/H1, Li/H2, Li/H3 e Li/H5). Finalmente, na comparação entre DM-MS-ILS e MDM-MS-ILS, houve significância estatística para duas instâncias (Bra/N2 e Li/H3).

4.3. Análise do comportamento das heurísticas

Para analisar e comparar o comportamento das heurísticas, foi realizado um teste usando a instância Bra/N4 com 1000 iterações para o PRVDFH. Na Figura 1, os gráficos da primeira linha mostram os custos das soluções obtidas nas fases de geração e busca local em cada iteração, enquanto os da segunda linha mostram visões ampliadas enquadrando as curvas correspondentes à fase de busca local. O rótulo **MD** indica as iterações que antecederam execuções do procedimento de mineração de dados nos gráficos das heurísticas híbridas.

Os gráficos mostram que todas as heurísticas apresentam o mesmo comportamento até a primeira execução do procedimento de mineração de dados. No DM-MS-ILS a mineração de dados ocorre apenas uma vez, após a iteração 500, enquanto no MDM-MS-ILS ela é realizada pela primeira vez após a iteração 328 e novamente em outras três ocasiões durante o restante da execução. Os gráficos da primeira linha mostram uma redução notável no custo das soluções geradas pelas heurísticas híbridas após cada execução do procedimento de mineração de dados, enquanto as ampliações na segunda linha revelam que também há uma redução correspondente no custo das soluções encontradas na fase de busca local. Já no MS-ILS, tanto as soluções geradas quanto as encontradas na busca local permanecem nos mesmos patamares durante toda a execução.

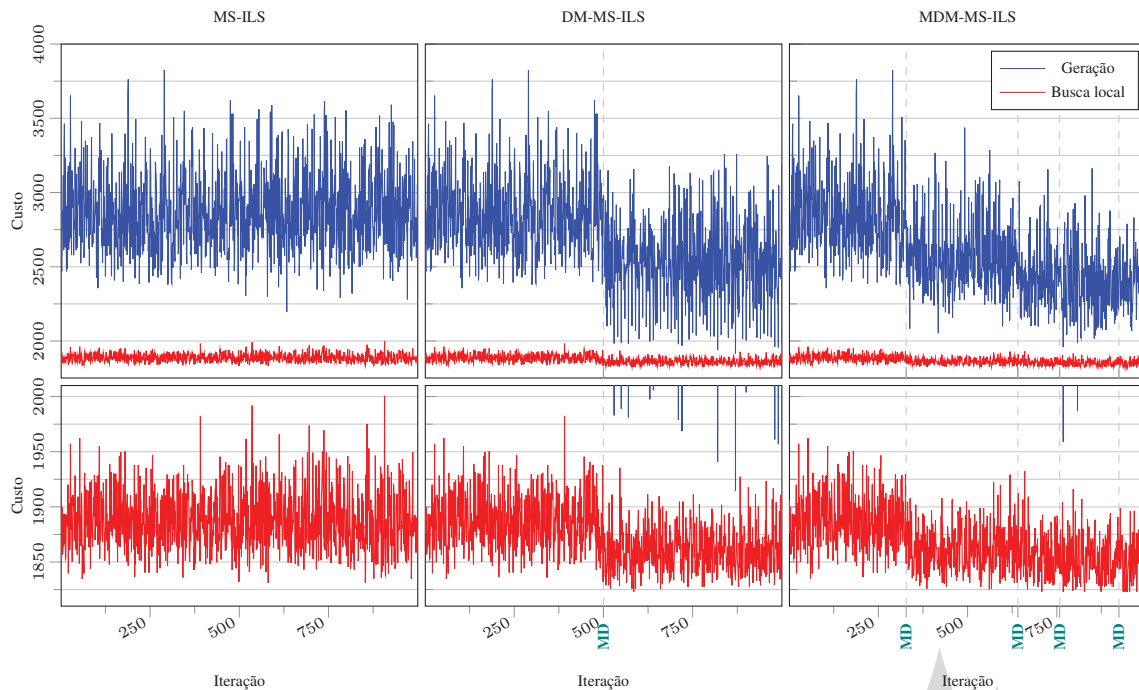


Figura 1: Gráficos de Custo x Iteração para o PRVDFH usando a instância Bra/N4 com 1000 iterações

Esses resultados ilustram o comportamento das heurísticas híbridas propostas. Após a mineração de dados, os padrões encontrados passam a ser usados na fase de geração, resultando em soluções iniciais de melhor qualidade (mais próximas de ótimos locais). Consequentemente, a busca local se beneficia de melhores pontos de partida, alcançando soluções melhores com uma diminuição do tempo de convergência que, como os resultados obtidos mostram, não apenas compensa o tempo adicional gasto com a atualização do conjunto elite e a mineração de dados como resulta em uma redução no tempo computacional total.

5. Conclusões e Trabalhos Futuros

Neste trabalho, foram propostas heurísticas híbridas construídas a partir da incorporação de uma técnica de mineração de dados a uma heurística MS-ILS de desempenho competitivo para o problema de roteamento de veículos com frota heterogênea. Foram realizados experimentos computacionais para duas variantes do problema. Para a variante com dimensionamento de frota, foram usadas 17 instâncias com número de clientes variando entre 20 e 199. Já para a variante com frota fixa, foram usadas 10 instâncias com número de clientes variando entre 100 e 360.

Os resultados alcançados reforçam a efetividade da abordagem, uma vez que as versões híbridas construídas alcançaram soluções e tempos melhores para a maioria das instâncias. Além disso, as diferenças observadas entre os desempenhos das heurísticas apresentaram significância estatística para várias instâncias. Particularmente, a diferença entre MS-ILS e MDM-MS-ILS (a melhor heurística entre as três avaliadas) se mostrou estatisticamente significativa para 62% das instâncias com mais que 100 clientes.

Como trabalhos futuros, pretende-se explorar novas formas de aplicação de técnicas de mineração de dados, com foco na melhoria da qualidade das soluções iniciais construídas na fase de geração, estratégia que se mostrou promissora neste trabalho.

Referências

Barbalho, H., Rosseti, I., Martins, S. L. e Plastino, A. (2013), A hybrid data mining GRASP with path-relinking, *Computers & Operations Research*, 40, 3159-3173.

- Brandão, J.** (2009), A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem, *European Journal of Operational Research*, 195, 716-728.
- Brandão, J.** (2011), A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, *Computers & Operations Research*, 38, 140-151.
- Choi, E. e Tcha, D.-W.** (2007), A column generation approach to the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, 34, 2080-2095.
- Feo, T. A. e Resende, M. G. C.** (1995), Greedy Randomized Adaptive Search Procedures, *Journal of Global Optimization*, 6, 109-133.
- Golden, B., Assad, A., Levy, L. e Gheysens, F.** (1984), The fleet size and mix vehicle routing problem, *Computers & Operations Research*, 11, 49-66.
- Grahne, G. e Zhu, J.** (2003), Efficiently using prefix-trees in mining frequent itemsets, *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*.
- Guerine, M., Rosseti, I. e Plastino, A.** (2014), Extending the hybridization of metaheuristics with data mining to a broader domain, *Proceedings of the 16th International Conference on Enterprise Information Systems*, 395-406.
- Imran, A., Salhi, S. e Wassan, N. A.** (2009), A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem, *European Journal of Operational Research*, 197, 509-518.
- Li, S., Golden, B. e Wasil, E.** (2007), A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, 34, 2734-2742.
- Liu, S., Huang, W. e Ma, H.** (2009), An effective genetic algorithm for the fleet size and mix vehicle routing problems, *Transportation Research Part E*, 45, 434-445.
- Lourenço, H. R., Martin, O. C. e Stützle, T.**, Iterated Local Search, em Glover, F. e Kochenberger, G. A. (Eds.), *Handbook of metaheuristics*, Kluwer Academic Publishers, Dordrecht, 321-353, 2003.
- Martí, R.**, Multi-start methods, em Glover, F. e Kochenberger, G. A. (Eds.), *Handbook of metaheuristics*, Kluwer Academic Publishers, Dordrecht, 355-368, 2003.
- Martins, D., Vianna, G. M., Rosseti, I., Martins, S. L. e Plastino, A.** (2014), Making a state-of-the-art heuristic faster with data mining, *Annals of Operations Research*, 1-22, DOI 10.1007/s10479-014-1693-4.
- 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*, 19, 201-232.
- Plastino, A., Fuchshuber, R., Martins, S. L., Freitas, A. A. e Salhi, S.** (2011), A hybrid data mining metaheuristic for the p-median problem, *Statistical Analysis and Data Mining*, 4, 313-335.
- Plastino, A., Barbalho, H., Santos, L. F. M., Fuchshuber, R. e Martins, S. L.** (2014), Adaptive and multi-mining versions of the DM-GRASP hybrid metaheuristic, *Journal of Heuristics*, 20, 39-74.
- Prins, C.** (2009), Two memetic algorithms for heterogeneous fleet vehicle routing problems, *Engineering Applications of Artificial Intelligence*, 22, 916-928.
- Ribeiro, M. H., Plastino, A. e Martins, S. L.** (2006), Hybridization of GRASP metaheuristic with data mining techniques, *Journal of Mathematical Modelling and Algorithms*, 5, 23-41.
- Santos, L. F., Martins, S. L. e Plastino, A.** (2008), Applications of the DM-GRASP heuristic: a survey, *International Transactions in Operational Research*, 15, 387-416.
- Taillard, E. D.** (1999), A heuristic column generation method for the heterogeneous fleet VRP, *RAIRO - Operations Research*, 33, 1-14.
- Tarantilis, C. D., Kiranoudis, C. T. e Vassiliadis, V. S.** (2004), A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *European Journal of Operational Research*, 152, 148-158.
- Yusim, P., Morett, R. B., Plastino, A. e Martins, S. L.** (2013), Explorando a hibridização da heurística ILS com mineração de dados, *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional*, 1927-1934.