São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

Aplicação de matheurísticas em modelos estendidos*

Francisco Ferreira L. Neto, Edna A. Hoshino, Vagner Pedrotti
Universidade Federal de Mato Grosso do Sul//Faculdade de Computação
Av. Costa e Silva,s/n. Cidade Universitária. Campo Grande-MS.
{francisco.ferreira,edna.hoshino,vagner.pedrotti}@ufms.br

RESUMO

Neste trabalho foram avaliadas matheurísticas para construir soluções viáveis para dois problemas de otimização combinatória, cujas formulações por programação linear inteira são modelos estendidos. As matheurísticas utilizadas foram a *Relax&Fix*, baseada na relaxação linear e a *Large neighborhood search*, baseada na destruição e reparo de uma solução já existente. Os modelos estendidos usados para avaliação foram originados de dois problemas, o problema da rota lucrativa capacitada e o problema de roteamento com frota particular e terceirizada, de modo que as matheurísticas foram utilizadas para resolver o subproblema de seleção de colunas após o uso da relaxação `ng-route` para resolver o *pricing*, relaxação esta a qual pode ser aplicada de forma heurística. Os resultados obtidos indicaram que o uso de `ng-route` heurístico e o uso das matheurísticas em alguns nós da árvore de enumeração de *Branch&Price* obtiveram os melhores resultados entre as abordagens propostas.

PALAVRAS CHAVE. Matheurísticas. Programação Linear Inteira. Problemas de roteamento de veículos com prêmios.

Tópicos: Programação matemática, Meta-heurísticas, Otimização combinatória.

ABSTRACT

In this paper we evaluated matheuristics to solve two combinatorial optimization problems, whose formulations for integer linear programming are extended models. The proposed matheuristics were *Relax&Fix*, based on linear relaxation, and *Large Neighborhood Search*, based on the Destruction and Reparation of an already existing solution. The extended models used for the evaluation were originated from two problems, the capacitated profitable tour problem and the vehicle routing problem with private fleet and common carrier, such that the matheuristics were used to solve the column selection subproblem after the aplication of the `ng-route` relaxation to solve *pricing*, relaxation of which can be implemented heuristically. The results indicate that the use of an heuristic `ng-route` and the use of the matheuristics in some nodes of the enumeration tree of *Branch&Price* generated the best results within the proposed approaches.

KEYWORDS. Matheuristics. Integer Linear Programming. Vehicle routing problems with profits.

Paper topics: Mathematical Programming, Metaheuristics, Combinatorial Optimization.

*Agradecemos a Capes pelo apoio financeiro ao projeto

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

1. Introdução

Problemas de otimização combinatória ocorrem em diferentes áreas de aplicação e, em geral, são classificados como NP-difíceis, para os quais não se conhece algoritmos de complexidade de tempo polinomial no tamanho da instância. O problema de roteamento de veículos é um clássico problema de otimização computacionalmente difícil que ocorre na área de redes, logística e telecomunicações. Uma abordagem bastante utilizada nesses problemas é o uso de programação linear inteira para modelá-los e resolvê-los com o uso de algoritmos exatos como o *branch-and-bound*, os quais fundamentalmente se baseiam em sucessivamente dividir o espaço de soluções e calcular limitantes duais e primais para provar a otimalidade da melhor solução encontrada. Os modelos estendidos, em geral, fornecem limitantes duais mais apertados que os modelos compactos e, desta forma, melhoram o desempenho dos algoritmos *branch-and-bound*. Por exemplo, os melhores algoritmos exatos na literatura para os problemas de roteamento de veículos, como em Pecin et al. [2017], são baseados em modelos estendidos. Os modelos estendidos se caracterizam pela presença de uma quantidade exponencial de variáveis (colunas). Para resolver a relaxação linear desses modelos, utiliza-se o método da geração de colunas. Basicamente, esse método consiste em, iterativamente, resolver dois subproblemas, um problema mestre restrito (PMR) e um problema de *pricing* (PP). O PMR equivale à relaxação linear do problema original restrito a um subconjunto menor de variáveis enquanto o PP gera novas colunas com potencial para melhorar o valor ótimo do PMR da iteração atual. Em um problema de minimização, essas colunas promissoras são aquelas cujas variáveis têm custo reduzido negativo.

Na última década, houve um aumento no uso de heurísticas baseadas em programação matemática, denominadas matheurísticas, para problemas de otimização combinatória NP-difíceis. No entanto, muitos poucos abordam o uso dessas matheurísticas em modelos estendidos. Em Ball [2011], as matheurísticas são classificadas em métodos de decomposição, métodos baseados na relaxação, métodos de melhoria e métodos baseados em técnicas de programação matemática.

Neste trabalho avaliaremos o uso de matherísticas para a resolução de modelos estendidos de programação linear inteira (PLI) para os seguintes problemas: Problema da rota lucrativa capacitada, em inglês *capacitated profitable tour problem* (CPTP), e o Problema do roteamento com frota particular e terceirizada, em inglês *vehicle routing problem with private fleet and common carrier* (VRPPFCC). O CPTP e o VRPPFCC estão em uma classe de problemas conhecida como problemas de roteamento de veículos com lucros como caracterizado por Archetti et al. [2014]. O CPTP, que foi objeto de estudo de Archetti et al. [2009], é um problema no qual se tem uma frota de veículos e um grupo de clientes com certas demandas e prêmios associados a estas, de modo que a função objetivo consiste na maximização dos prêmios coletados subtraídos dos custos das rotas utilizadas. Vale salientar que a visita dos clientes é seletiva, ou seja, nem todos os clientes precisam ser visitados. O VRPPFCC é, visto sob a descrição de Côté e Potvin [2009], um problema de minimização de custos, no qual todos os clientes precisam ser atendidos pelos veículos da frota particular ou por meio de entregas terceirizadas. Nesse problema, cada veículo da frota particular possui um custo fixo de utilização e cada cliente possui um custo, que representa o valor a ser pago caso a sua demanda seja atendida por uma entrega terceirizada. Atualmente, o melhor algoritmo exato para resolver o CPTP e o VRPPFCC foi o proposto por Pessoa et al. [2020], no qual foi desenvolvido um algoritmo exato genérico que permite resolver diversas variantes de problemas de roteamento e outros problemas de otimização combinatória. Os melhores resultados da literatura para o CPTP e VRPPFCC, do ponto de vista de heurísticas, foi proposto por Bulhoes et al. [2018].

Para a resolução do CPTP e do VRPPFCC foi implementado um algoritmo *Branch&Price* (`B&P`), similar ao desenvolvido por Archetti et al. [2009], no qual o PP foi solucionado pela relaxação

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

`ng-route` de Baldacci et al. [2011]. A implementação do `B&P` permite a resolução dos problemas de forma exata, porém para auxiliar o algoritmo a construir soluções viáveis e acelerar a convergência do algoritmo, aplicou-se as matheurísticas propostas em Alves e Hoshino [2022] nos nós da árvore de enumeração do `B&P`. Essa técnica é bastante usada para acelerar os algoritmos exatos `B&P`. Em nosso trabalho, usamos o `B&P` em um tempo limite de tempo para construir soluções aproximadas. Outrossim, como forma de reduzir o tempo utilizado para a geração de colunas do PP foi utilizada uma heurística do `ng-route`, o qual contribui para a convergência da geração de colunas, porém não garante que todas as colunas de custo reduzido negativo sejam geradas. Ou seja, um `B&P` resolvido de forma heurística também foi analisado para gerar soluções aproximadas.

Após a solução do *pricing*, é possível a interpretação do problema de seleção de rotas como um problema da mochila com restrição de conflitos, em inglês *disjunctively constrained knapsack problem* (DCKP), problema que também é NP-difícil como evidenciado pela possibilidade de reduzir o clássico problema da mochila (KP) para o DCKP, pois o KP consiste em um DCKP sem itens conflitantes. Dessa forma, foram utilizadas matheurísticas inicialmente desenvolvidas para o DCKP como forma de resolver o problema de seleção de colunas para o CPTP e o VRPPFCC.

A abordagem de um subproblema do problema mestre como um DCKP, em conjunto ao uso de matheurísticas, já foi utilizada por Lahyani et al. [2017] para resolver uma variante do problema do caixeiro viajante com lucros conhecida como problema rico de roteamento de veículos, em inglês *rich vehicle routing problem*, na qual intervalos de tempo restringem as demandas, o veículo é separado em compartimentos e certos itens apresentam conflitos com outros itens e compartimentos, de forma que uma rota deve ser determinada para o veículo com o objetivo de maximizar os lucros coletados subtraídos dos custos envolvidos na rota. O trabalho em questão interpretou a seleção dos itens a serem adicionadas no veículo como um DCKP múltiplo.

1.1. Modelos estendidos utilizados para os problemas em estudo

Dado um grafo $G = (V, E)$, os vértices em V representam o depósito e os clientes e as arestas de E os caminhos que conectam tais vértices. Considere $\mathcal{P}$ o conjunto de todas as rotas possíveis em G que respeitam a capacidade dos veículos. Cada rota $p \in \mathcal{P}$ tem um custo associado c_p e uma variável binária λ_p que representa o seu valor na solução. O tamanho da frota é dado por m. Para cada vértice i (aresta e) é dada uma constante a_i^p (a_e^p) que indica a presença do vértice i (aresta e) na rota p. A variável y_i indica a não utilização de um vértice i, de forma que $y_i = 1$ quando o vértice i não é utilizado e $y_i = 0$ caso contrário. A utilização de uma aresta e é representada por x_e, tal que $x_e = 1$ quando a aresta é utilizada e $x_e = 0$ quando a mesma aresta não entra na solução.

O custo associado a cada rota p é dado por:

$$c_p = \sum_{e \in E} c_e a_e^p - \sum_{i \in V} p_i a_i^p. \tag{1}$$

Dessa forma, o modelo estendido de PLI para o CPTP é dado por:

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

$$\text{min} \quad \sum_{p\in\mathcal{P}} c_p\lambda_p \tag{2}$$

$$\text{s.a.} \quad \sum_{p\in\mathcal{P}} \lambda_p \leq m \tag{3}$$

$$\sum_{p\in\mathcal{P}} a_i^p\lambda_p + y_i = 1 \qquad \forall i \in V\setminus\{0\} \tag{4}$$

$$\sum_{p\in\mathcal{P}} a_e^p\lambda_p = x_e \qquad \forall e \in E \tag{5}$$

$$\lambda_p \in \{0,1\} \qquad \forall p \in \mathcal{P} \tag{6}$$

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{7}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{8}$$

A função objetivo (2) visa a minimização dos custos das rotas selecionadas, conforme dado em (1). A restrição (3) garante que o número de rotas selecionadas para a solução respeite o tamanho da frota disponível. A restrição (4) não permite que duas rotas selecionadas passem pelo mesmo vértice. A restrição (5) evita que uma aresta esteja presente em mais de uma rota da solução. Ademais, as restrições (6), (7) e (8) são restrições de integralidade.

Sejam π e α as variáveis duais associadas às restrições (3), (4) e (5). O custo reduzido de uma rota p é dado pela equação:

$$\overline{c}_p = c_p - \sum_{i\in V\setminus\{0\}} \pi_i a_i^p - \sum_{e\in E} \alpha_e a_e^p - \pi_0.$$

Portanto, usando-se (1) obtém-se

$$\overline{c}_p = \sum_{(i,j)\in E} (c_{ij} - p_j - \pi_j - \alpha_{ij}) a_{ij}^p.$$

O PP associado ao modelo consiste em:

$$\min_{p\in\mathcal{P}} \overline{c}_p,$$

O modelo estendido utilizado para solucionar o VRPPFCC é similar ao aplicado ao CPTP, com diferença nos custos associados às rotas apresentado em (1), pois este deve, adicionalmente, incluir o custo fixo de uso de cada veículo da frota particular, denotado por θ. Dessa forma, para o VRPPFCC, o custo de uma rota p assume:

$$c_p = \sum_{e\in E} c_e a_e^p + \theta - \sum_{i\in V} p_i a_i^p. \tag{9}$$

O VRPPFCC pode ser resolvido de forma similar ao CPTP pois a minimização dos custos de entrega terceirizada dos clientes não atendidos pela frota particular equivale à maximização dos custos dos clientes atendidos, os quais podem ser interpretados como os prêmios associados aos clientes, como ocorre no CPTP. Adicionalmente, note que o valor da solução do VRPPFCC é obtido somando-se a constante $\sum_{i\in V} p_i$ ao valor da solução em (2).

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

1.2. Modelagem para o DCKP

Dado um grafo $G = (V, E)$, os vértices de V representam os potenciais itens a serem colocados na mochila e as arestas de E as restrições de conflito entre os itens de V, de forma que o grafo não é necessariamente completo. A capacidade da mochila é representada por C. Um item i está associado a uma variável binária x_i, de modo que $x_i = 1$ quando o item está na solução e $x_i = 0$ caso contrário. Cada item i tem valor v_i e peso p_i associados a ele.

Assim, o modelo de PLI utilizado para o DCKP consiste em:

$$\text{max} \quad \sum_{i \in V} v_i x_i \tag{10}$$

$$\text{s.a.} \quad \sum_{i \in V} p_i x_i \leq C \tag{11}$$

$$x_i + x_j \leq 1 \qquad \forall (i,j) \in E \tag{12}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{13}$$

A função objetivo (10) visa a maximização dos lucros associados aos itens adicionados a mochila. A restrição (11) garante que a solução respeita a capacidade da mochila. A restrição (12) limita a seleção de itens conflitantes. Por fim, a restrição (13) é uma restrição de integralidade.

2. Heurísticas

Como especificado anteriormente, foram utilizadas matheurísticas desenvolvidas para o DCKP para construir soluções viáveis para o CPTP e VRPPFCC, como também foi utilizada uma abordagem heurística para diminuir o tempo gasto pelo `ng-route`. Seguem portanto, na forma de tópicos, a descrição de tais implementações.

2.1. Matheurísticas

O trabalho realizado por Alves e Hoshino [2022] utilizou duas matheurísticas para a resolução do DCKP, as matheurísticas Relax&Fix (`RF`) e Busca em vizinhança em grande escala, em inglês *Large Neighborhood Search* (`LNS`). Como forma de solucionar o CPTP e o VRPPFCC, utilizamos tais matheurísticas em meio ao `B&P`, de modo que foram experimentadas diferentes periodicidades de uso das heurísticas para os nós da árvore `B&P`.

A matheurística `RF` se baseia na partição das variáveis inteiras de um problema de PLI em K partes e na resolução sucessiva de até K subproblemas de PLI. A cada iteração k, resolve-se um subproblema P_k, que consiste do problema original com a restrição de integralidade relaxada, exceto das variáveis pertencentes à parte k. Além disso, o valor das variáveis na solução ótima do P_k são fixadas para as iterações subsequentes. Em outras palavras, a `RF` decompõe o problema de PLI em K subproblemas de PLI misto com poucas variáveis inteiras. Uma vez que as variáveis do DCKP estão associadas aos itens, pode-se visualizar o passo de particionamento das variáveis do método `RF` no DCKP, como um particionamento dos itens.

O Algoritmo 1 representa a implementação utilizada para a `RF` na forma de pseudocódigo. Nesta representação, o critério *tipo* corresponde ao critério de particionamento dos itens na partição, podendo tal critério ser uma organização gulosa ou aleatória dos itens. A variável k representa a parte da partição que está sendo solucionada. O PLI utilizado no algoritmo consiste em um PLI misto, no qual a integralidade das variáveis x_i, correspondentes ao modelo utilizado para o DCKP na Seção 1.2, são relaxadas.

No passo 5 do Algoritmo 1 seleciona-se o conjunto dos itens que terão as respectivas variáveis resolvidas com a restrição de integralidade. Já no passo 6, o subproblema da iteração atual

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

Algoritmo 1 *Relax&Fix*

Entrada: Instância I, quantidade e tipo de particionamento ▷ *quantidade representada por* K
Saída: Solução S

```
1► Fixados0 ← ∅
2► Fixados1 ← ∅
3► Particiona(Partição,Itens, K, tipo)   ▷ Separa os itens entre as partes da partição pelo critério
4► para cada k de 1 a K faça
5►     ItensIntegrais ← Partição[k]
6►     S ← ResolvePLI(I, Fixados0, Fixados1, ItensIntegrais)   ▷ Solução ótima representada por S
7►     para cada item ∈ Partição[k] faça
       ▷ Fixa os itens da parte k para a próxima iteração
8►         se item ∈ S então
9►             Fixados1 ← Fixados1 ∪{ item }
10►        senão
11►            Fixados0 ← Fixados0 ∪{ item }
12►        fim se
13►    fim para
14► fim para
```

é resolvido com algumas variáveis fixadas em 0, algumas fixadas em 1 e com apenas a integralidade das variáveis da iteração atual.

A matheurística `LNS` consiste na destruição e no reparo de uma solução já existente. Assim, os itens menos interessantes, neste caso com menor valor, são removidos de uma solução e se resolve de forma exata uma versão reduzida do DCKP utilizando a capacidade disponibilizada e os itens que estão fora da mochila. Caso a nova solução seja pior que a original, o processo de remoção é revertido.

A aplicação do `LNS` é análoga a representada pelo Algoritmo 2. Nesta representação é utilizada uma solução inicial S, que neste caso foi gerada pela matheurística `RF`, a qual é copiada para S' e em seguida são removidos os $\lfloor D \cdot |S'| \rfloor$ itens de menor valor, o que caracteriza a função *Destroy*. É em seguida resolvido o PLI com os itens já dentro da mochila fixados, e a solução final é salva em S', caso seja melhor.

Algoritmo 2 *Large Neighborhood Search*

Entrada: Instância I, Solução S, porcentagem para Destruição D
Saída: Nova solução S'

```
1► S' ← S                                   ▷ Solução original é copiada
2► Destroy(S',D)                            ▷ Remove os itens de menor valor de S'
3► S" ← ResolvePLI(I, ∅, S', I)             ▷ Resolve com os itens em S' fixados
4► se o valor de S" é maior que o valor de S' então
5►     S' ← S"
6► senão
7►     S' ← S
8► fim se
```

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

2.2. Heurísticas para o `ng-route`

A fim de resolver o PP foi utilizada a relaxação `ng-route`, que permite a repetição de vértices ao longo da rota desde que eles não estejam na "memória" da rota. Essa memória é construída com base nos conjuntos $NG(i)$, que são previamente associados a cada vértice i do grafo e, em geral, se referem aos Δ vértices mais próximos de i. A repetição de um vértice i é permitida em uma rota p se existir um vértice j entre as duas ocorrências de i em p tal que $i \notin NG(j)$. Portanto, quanto maior o valor de Δ, maior é a elementariedade da rota. Como forma de diminuir o tempo gasto pela geração de colunas, foi utilizada uma heurística de *aggressive dominance* (AD), metodologia de *pricing* heurístico descrita por Costa et al. [2019], a qual determina que só serão mantidas em cada lista as rotas de melhor custo reduzido. Ao utilizar a heurística AD, a resolução do PP se torna mais rápida, com tempo de execução proporcional a $O(|V|^2 Q)$, porém perde-se a garantia que todas as colunas de custo reduzido negativo sejam geradas.

2.3. Rotas não elementares

Rotas não elementares são aquelas com repetição de vértices, que são geradas devido à relaxação `ng-route` embora sejam inválidas para uso em uma solução, como por exemplo a rota representada na Figura 1a, na qual o vértice em preto representa o depósito e é possível observar que o vértice a é visitado duas vezes. Algumas das instâncias apresentavam problemas pela falta de colunas elementares, o que dificultava que uma solução válida fosse encontrada. Dessa forma, implementaram-se dois métodos de transformação de colunas não elementares para elementares.

A primeira rotina de conversão consiste na remoção de subciclos entre dois vértices repetidos, de modo que todos os subciclos que não passam pelo depósito são removidos. Com esta abordagem, as arestas utilizadas na nova coluna estão contidas no conjunto de arestas presentes na coluna original. A Figura 1b exemplifica a utilização do método, de modo que o subciclo indicado em vermelho é removido e uma rota inicialmente inválida se torna elementar.

O segundo método se baseou na remoção de vértices repetidos, de forma que as repetições são removidas e os vértices remanescentes compõem a nova rota na mesma sequência da rota original, porém arestas não utilizadas anteriormente podem ser adicionadas a coluna neste método. Uma aplicação do segundo método é representado pela Figura 1c, tal que as arestas em vermelho indicam as que foram removidas e a em azul indica uma nova aresta foi utilizada ao transformar a rota exemplo em elementar.

Os métodos de transformação geram diversas rotas elementares com custo reduzido não negativo. Dessa forma, além das transformações foi utilizado um parâmetro binário nomeado *rc* que, se habilitado, determina que as rotas elementares de custo reduzido não negativas sejam filtradas, o que reduz o número de itens considerados pelas matheurísticas.

A implementação das abordagens propostas pode ser observada no diagrama da Figura 2.

3. Resultados Computacionais

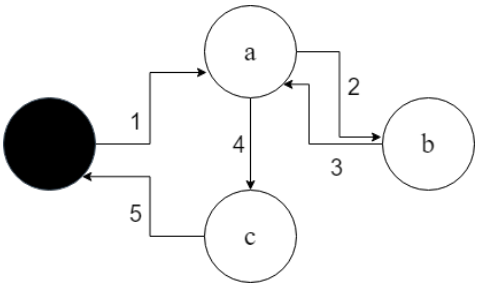
Os experimentos foram realizados com a biblioteca de otimização SCIP Gamrath et al. [2016], usando-se o resolvedor CPLEX IBM [2019] [1]. Os testes foram realizados em uma máquina com sistema operacional Linux e *multithread* do resolvedor desabilitado, equipada com processador Intel(R) Core(TM) i7-4790(3.60 GHz) e 32GB de memória RAM.

Para fins de teste, foram utilizadas as 130 instâncias [2] de Archetti et al. [2009] para o CPTP, de 51 a 200 vértices, e as 34 instâncias de Bolduc et al. [2008] para o VRPPFCC, de 51 a 484 vértices. A avaliação da qualidade dos resultados obtidos se deu pela comparação com os

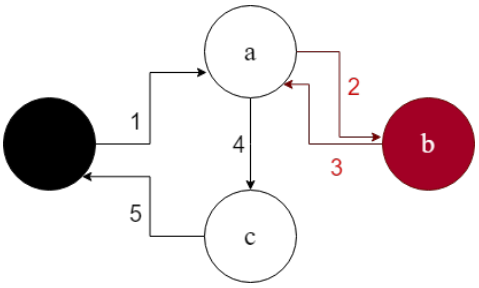
[1] Agradecimentos aos fornecedores do SCIP e do CPLEX pela disponibilização de licenças acadêmicas

[2] Instâncias disponíveis em `https://w1.cirrelt.ca/~vidalt/en/VRP-resources.html`, sob o tópico VRP-SL

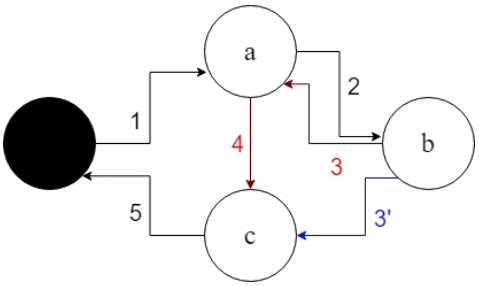
DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208



(a) Exemplo de rota não elementar



(b) Primeiro rotina de conversão



(c) Segunda rotina de conversão

Figura 1: Exemplo de aplicação das rotinas de conversão de rotas não elementares em elementares

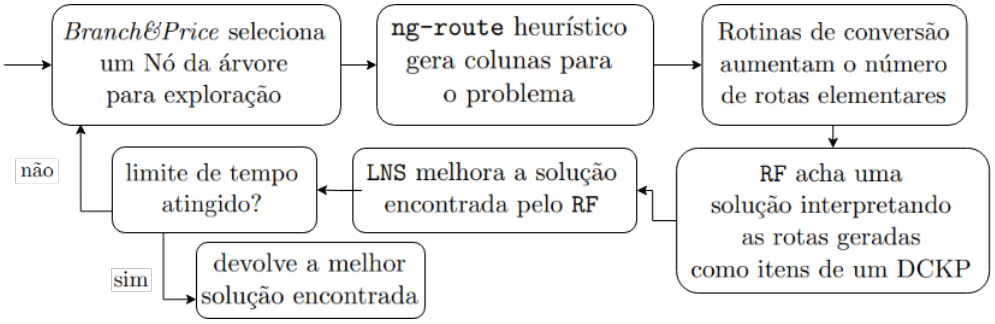


Figura 2: Representação sequencial das implementações

de Bulhoes et al. [2018], que propôs um algoritmo genético híbrido para resolver o problema de roteamento de veículos com restrições de níveis de serviço, em inglês *vehicle routing problem with service level constraints* (VRP-SL), que generaliza o CPTP e o VRPPFCC, entre outros.

Em Neto et al. [2023], avaliamos a qualidade das soluções geradas pelas matheurísticas `RF` e `LNS` executando-se o algoritmo `B&P` apenas no nó raiz, ou seja, executando-se cada uma dessas matheurísticas uma única vez considerando-se todas as rotas elementares geradas ao resolver a relaxação linear do PMR. Observamos que o uso combinado de `RF` e `LNS`, com percentual de destruição de 50%, obtiveram soluções melhores em 85% das instâncias do CPTP e em cerca de 30% das instâncias do VRPPFCC com ganho médio de 15% e 18%, respectivamente, quando comparadas às demais configurações dos parâmetros avaliados.

Adotando-se essa configuração (denotada por `RF+LNS`), realizamos testes para avaliar o ganho no uso das rotinas de conversão de rotas não elementares em elementares, descritas na Seção 2.3, e na escolha do parâmetro Δ do algoritmo `ng-route`.

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

A Tabela 1 apresenta um resumo desses experimentos realizados nas instâncias do CPTP. A coluna `B&P` indica os resultados do algoritmo `B&P` com `RF+LNS` sendo executados em todos os nós da árvore de enumeração, cujo tempo de execução foi limitado a 100 segundos e com $\Delta = 3$. As colunas `t`, `t-d5`, `t-d7` e `t-d9` apresentam os resultados quando a rotina de conversão de rotas não elementares em elementares é adotada com Δ configurado em 3, 5, 7 e 9, respectivamente. A primeira linha dessa tabela indica o total de instâncias resolvidas na otimalidade por cada um dos códigos. Já a segunda linha apresenta o gap médio, que é calculado pela média dos gaps de dualidades, considerando-se apenas as instâncias não resolvidas na otimalidade por cada código. O gap de dualidade é calculado por $(UB - LB)/UB$, no qual UB e LB denotam, respectivamente, o valor da melhor solução primal e do melhor limitante dual alcançados pelo código. A terceira linha mostra a média do tempo, em segundos, gasto por cada código apenas nas instâncias resolvidas na otimalidade e a quarta linha indica o total de instâncias em que a relaxação linear não foi resolvida no nó raiz do `B&P` dentro do limite de tempo. Por fim, a última linha apresenta o percentual das instâncias em que a melhor solução foi obtida pela matheurística `RF+LNS`.

	B&P	t	t-d5	t-d7	t-d9
otimalidade	47	49	58	67	80
gap médio	0,24	0,13	0,07	0,05	0,06
tempo médio	11,4	11,8	10,1	7,5	11,4
sem gap	2	2	6	9	16
rf+lns	0,84	0,86	0,75	0,66	0,47

Tabela 1: Uso do procedimento de elementarização de rotas e variação em Δ nas instâncias do CPTP.

Observando-se a Tabela 1, comparando-se as colunas `B&P` e `t`, nota-se que houve um ganho no uso da rotina de conversão das rotas em elementares. De fato, houve uma redução de 45% no gap médio ao usar a rotina de conversão das rotas em elementares, além de um incremento de 2 nas instâncias resolvidas na otimalidade. Avaliando-se os diferentes valores de Δ, como esperado, há um aumento significativo no total de instâncias resolvidas na otimalidade à medida em que se aumenta o valor de Δ. Por outro lado, sabe-se que quanto maior o valor de Δ mais difícil se torna o `ng-route`, o que explica o aumento no total de instâncias sem gap, ou seja, para as quais a relaxação linear no nó raiz não foi resolvida na otimalidade. Observando-se a última linha da tabela, nota-se que há um aumento no total de soluções ótimas obtidas pela relaxação linear em detrimento da matheurística à medida em que se aumenta o valor de Δ, possivelmente devido ao aumento no total de rotas elementares geradas pela `ng-route`.

Em seguida, avaliamos o impacto em usar as heurísticas de *pricing*, descritas na Seção 2.2, ao invés do *pricing* exato. Observe que isso implica que o valor da relaxação linear em cada nó deixa de ser um limitante dual válido. Vamos denotar por `B&PH` o uso do `B&P` em que o `ng-route` é executado de forma heurística e, portanto, gerando soluções sem garantia de otimalidade. Nos testes que se seguem, adotamos $\Delta = 9$ e usamos a segunda rotina de conversão de rotas elementares.

Observamos que não houve muita perda no valor da relaxação linear ao usar a heurística de dominância agressiva. Em 78% das instâncias do CPTP, não houve alteração no valor da relaxação linear no nó raiz obtido por `B&P` e `B&PH`. Nas demais instâncias, a diferença foi, na média, de apenas $0,07\%$, sendo que a maior diferença foi de $1,82\%$ (na instância `p14-2-200`). Além disso, em apenas 2 instâncias do CPTP, o `B&PH` não concluiu a relaxação linear no nó raiz.

A Tabela 2 apresenta os resultados dos experimentos para avaliar o comportamento do `B&PH`. Nessa tabela, a interpretação dos resultados em cada linha é a mesma que na Tabela 1, exceto

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

que a otimalidade e o gap de dualidade são em relação ao "limitante dual" calculado pelo B&PH, que de fato não representa um limitante dual real do problema, como mencionado no parágrafo anterior. Esse gap foi adotado pois ele dá uma medida do quão longe a heurística está para terminar de explorar o seu espaço de busca. Por exemplo, um gap alto pode indicar que o tempo limite está pequeno. A coluna B&PH+a apresenta os resultados do B&PH com a execução da matheurística RF+LNS, mesmo nas situações em que a relaxação linear do nó raiz não tenha sido calculada (por limite de tempo). Já B&PH+a+f é o B&PH+a com a diferença que a matheurística é executada apenas a cada 3 níveis da árvore de enumeração para diminuir o impacto do tempo gasto com a matheurística no tempo total gasto pelo B&PH.

	B&PH	B&PH+a	B&PH+a+f
otimalidade	83	83	86
gap médio	0,05	0,05	0,05
tempo médio	5,5	4,2	5,3
sem gap	2	2	2
sem UB	2	0	0
rf+lns	0,58	0,61	0,52

Tabela 2: Resumo comparativo do B&PH nas instâncias do CPTP.

Notadamente, o B&PH+a ajudou a reduzir o tempo médio das instâncias resolvidas na "otimalidade" e tornou possível obter uma solução primal para todas as instâncias. Já B&PH+a+f aumentou o total de instâncias resolvidas na "otimalidade", devido ao aumento no total de nós explorados pelo B&PH, propiciado pela redução no tempo total gasto com RF+LNS. Comparada com B&PH+a, houve uma redução de 23% no tempo gasto pela RF+LNS nas instâncias resolvidas por ambos e de 51% nas 3 instâncias adicionais resolvidas na "otimalidade" pelo B&PH+a+f.

A Tabela 3 apresenta um resumo comparativo da qualidade das soluções geradas pelas diferentes implementações estudadas neste trabalho. A primeira linha de resultados da tabela indica a média do gap de primalidade, dado por $(z^* - UB)/UB$, em que UB representa o valor da solução obtida por cada implementação e z^* é o melhor valor dentre todas as implementações avaliadas. Na última linha temos o total de instâncias em que cada implementação gerou a melhor solução.

	B&P	t	t-d9	B&PH	B&PH+a	B&PH+a+f
gap	0,18	0,13	0,44	0,09	0,01	0,01
total	77	80	91	104	111	103

Tabela 3: Resumo comparativo das diferentes implementações nas instâncias do CPTP.

Analisando-se a Tabela 3, podemos notar que a implementação B&PH+a gerou as melhores soluções em mais de 85% das instâncias e nos demais casos, o valor da solução gerada foi, na média, $0,01\%$ pior que o valor da melhor solução.

A partir dos dados analisados para o CPTP, avaliamos a implementação B&PH+a e a implementação B&PH+a+f para o VRPPFCC. Nos experimentos com o VRPPFCC, utilizamos o limite de tempo de 300 segundos, em conformidade com o tempo reportado na literatura para essas instâncias. A Tabela 4 apresenta os resultados deste experimento.

Observamos que ambas as estratégias não conseguiram resolver a relaxação linear no limite de tempo para 50% das instâncias, e para 3 dessas instâncias não foi possível construir uma

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

	B&PH+a	B&PH+a+f	B&PH+a+f+rc
otimalidade	0	0	0
gap médio	0,3	0,29	0,40
sem gap	17	17	16
sem UB	3	3	0
rf+lns	0,91	0,91	1

Tabela 4: Resumo comparativo do B&PH nas instâncias do VRPPFCC.

solução. Uma nova estratégia foi considerada, denotada por B&PH+a+f+rc, que consiste em incluir no PMR apenas as rotas elementares, construídas pela rotina de conversão de rotas em elementares, com custo reduzido negativo. Os resultados mostram que essa estratégia tornou possível construir soluções primais para todas as instâncias do VRPPFCC.

Por fim, avaliamos a qualidade das soluções geradas, comparando com a heurística HGS proposta em Bulhoes et al. [2018], que apresenta os melhores resultados da literatura, dentre as heurísticas propostas para o CPTP e VRPPFCC. Nas instâncias do CPTP, a implementação B&PH+a gerou soluções com o mesmo valor que HGS em 67% das instâncias e nas demais apresentou um gap de primalidade médio de $4,41\%$, com desvio padrão de $3,91\%$. O tempo médio to B&PH+a, considerando-se todas as instâncias do CPTP foi de $46,9$ segundos enquanto o tempo médio reportado pelo HGS é de $14,1$ segundos. Nota-se que o B&PH+a exige mais tempo computacional que o HGS e é superado pelo mesmo a respeito da qualidade das soluções geradas. Já para as instâncias do VRPPFCC, observamos que as matheurísticas propostas não tiveram um bom desempenho quando comparado com a literatura. A melhor estratégia avaliada, gerou soluções piores com gap de primalidade médio de 46%. Dada a dificuldade de se resolver essas instâncias, supõe-se que mais tempo de processamento é necessário para que o B&PH possa gerar soluções melhores.

Espera-se que outras heurísticas do ng-route e o uso de uma heurística de *diving* com RF+LNS, para percorrer apenas alguns "mergulhos" na árvore de enumeração do B&PH, possam acelerar a matheurística proposta neste trabalho como discutido por Sadykov et al. [2019].

4. Conclusões

Neste trabalho foi analisada a aplicação de matheurísticas em modelos estendidos. Experimentos foram realizados nos problemas CPTP e o VRPPFCC. As matheurísticas utilizadas foram a *Relax&Fix* e a *Large neighborhood search*, na qual a primeira constrói uma solução e a última é usada para melhorá-la. Essas matheurísticas foram usadas de forma combinada dentro do *Branch&Price* em alguns níveis da árvore de enumeração. Como forma de resolver o subproblema de *pricing* foi utilizada a relaxação ng-route, como também foi aplicada uma heurística para diminuir a complexidade computacional do ng-route denominada *aggressive dominance*, a qual quando aplicada ao B&P resultou no que chamamos de B&PH.

Os resultados obtidos pela abordagem indicaram que o B&PH com o uso das matheurísticas a cada três níveis da árvore de enumeração apresentou os melhores resultados dentro do tempo limite de 100 segundos para o CPTP e 300 segundos para o VRPPFCC. Porém, a abordagem proposta não superou a melhor heurística da literatura desenvolvida por Bulhoes et al. [2018]. Será avaliado em trabalhos futuros o uso de uma heurística de *diving* para permitir rápidos mergulhos dentro da árvore de B&P, como também o uso de outras heurísticas para o ng-route.

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208

Referências

Alves, A. R. e Hoshino, E. A. (2022). Matheurísticas para o problema da mochila com restrição de conflitos. *Anais do Simpósio Brasileiro de Pesquisa Operacional*, 54.

Archetti, C., Feillet, D., Hertz, A., e Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60:831–842.

Archetti, C., Speranza, M. G., e Vigo, D. (2014). Chapter 10: Vehicle routing problems with profits. In *Vehicle routing: Problems, methods, and applications, second edition*, p. 273–297. SIAM.

Baldacci, R., Mingozzi, A., e Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations research*, 59(5):1269–1283.

Ball, M. O. (2011). Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16(1):21–38.

Bolduc, M.-C., Renaud, J., Boctor, F., e Laporte, G. (2008). A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society*, 59(6):776–787.

Bulhoes, T., Ha, M. H., Martinelli, R., e Vidal, T. (2018). The vehicle routing problem with service level constraints. *European Journal of Operational Research*, 265(2):544–558.

Costa, L., Contardo, C., e Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4):946–985.

Côté, J.-F. e Potvin, J.-Y. (2009). A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research*, 198(2):464–469.

Gamrath, G., Fischer, T., Gally, T., Gleixner, A. M., Hendel, G., Koch, T., Maher, S. J., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schenker, S., Schwarz, R., Serrano, F., Shinano, Y., Vigerske, S., Weninger, D., Winkler, M., Witt, J. T., e Witzig, J. (2016). The scip optimization suite 3.2. Technical Report 15-60, ZIB, Takustr. 7, 14195 Berlin.

IBM. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual, Version 12 Release 6*. IBM Corp., 2019.

Lahyani, R., Khemakhem, M., e Semet, F. (2017). A unified matheuristic for solving multi-constrained traveling salesman problems with profits. *EURO Journal on Computational Optimization*, 5(3):393–422.

Neto, F. F. L., Pedrotti, V., e Hoshino, E. (2023). Matheurísticas para o problema de roteamento de veículos com prêmios. In *Anais do VIII Encontro de Teoria da Computação*, p. 185–189, Porto Alegre, RS, Brasil. SBC.

Pecin, D., Pessoa, A., Poggi, M., e Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100. ISSN 1867-2957.

Pessoa, A., Sadykov, R., Uchoa, E., e Vanderbeck, F. (2020). A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183:483–523.

Sadykov, R., Vanderbeck, F., Pessoa, A., Tahiri, I., e Uchoa, E. (2019). Primal heuristics for branch and price: The assets of diving methods. *INFORMS Journal on Computing*, 31(2):251–267.

DOI: 10.59254/sbpo-2023-175208
http://live-proceedings.pantheonsite.io/p/175208