

UMA ABORDAGEM MATHEURÍSTICA PARA O PROBLEMA DE MÍNIMA LATÊNCIA

Gleisson de Assis

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas 7675, 30510-000 – Nova Gameleira – Belo Horizonte – MG – Brasil
gleisson.assis@gmail.com

Elisangela Martins de Sá

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas 7675, 30510-000 – Nova Gameleira – Belo Horizonte – MG – Brasil
elisangela.martinss@gmail.com

Sérgio Ricardo de Souza

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas 7675, 30510-000 – Nova Gameleira – Belo Horizonte – MG – Brasil
sergio@dppg.cefetmg.br

RESUMO

Este artigo estuda o Problema de Mínima Latência (PML), que é uma variação do Problema do Caixeiro Viajante. Ele é resolvido por meio da implementação das matheurísticas RENS, FP e *Local Branching*. O algoritmo final proposto é uma hibridização das técnicas RENS e *Local Branching* sendo testado e comparado com problemas teste da literatura, a fim de demonstrar o uso da técnica. As implementações demonstraram resultados ótimos para problemas de até 25 elementos, sendo que para tais problemas foram obtidos resultados médios com um gap do ótimo de -5.11% que é aproximadamente 15% melhor que a execução padrão do *solver* com um tempo limite predeterminado.

PALAVRAS CHAVE. Problema de Mínima Latência, Matheurísticas, RENS, FP, *Local Branching*.

ABSTRACT

This paper studies the Minimum Latency Problem (PML), which is a variation of the Traveling Salesman Problem. It is solved through the implementation of RENS, FP and Local Branching matheuristics. The proposed algorithm is a hybridization of RENS and Local Branching techniques and was tested and compared with literature instances to demonstrate the effectiveness of the technique. The implementations found optimal results for problems up to 25 elements, and for these problems we obtained average results with an optimum gap of -5.11% which was approximately 15% better than the standard execution of the solver with a predetermined time-out.

KEYWORDS. Minimum Latency Problem, Matheuristics, RENS, FP, Local Branching.

1. Introdução

Problemas de análise combinatória estão presentes em várias áreas e são aplicáveis a inúmeros problemas do nosso cotidiano, desde uma simples escolha de qual caminho seguir para chegar a um destino, até decisões complexas, como por exemplo, a correta organização de um quadro de horários para alocação de recursos. Sendo assim, a resolução de tais problemas, obtendo um resultado ideal ou o que normalmente é chamado de resultado ótimo, pode requerer um tempo de processamento muito alto, até mesmo quando se tem poucos elementos dependendo da estrutura do problema.

A análise e estudo de técnicas que possam trazer resultados próximos de um resultado dito ideal com um tempo aceitável, se faz muito importante, visto que, em algumas áreas, a diferença de segundos pode justificar ou não o uso da informação gerada. Para os chamados problemas de otimização combinatória, o domínio é, tipicamente, finito, entretanto dada a inviabilidade computacional de se examinar todos os possíveis elementos de um domínio, algum tipo de abordagem na qual apenas uma fração dos elementos é examinada, deve ser utilizada.

O campo de pesquisa de métodos heurísticos para aplicações de problemas de otimização combinatória é amplo. Isto é devido à importância de problemas de otimização para a comunidade científica, bem como para as áreas de produção e operações [Blum e Roli, 2003]. Em sua forma geral, problemas de otimização têm como objetivo maximizar ou minimizar uma função definida sobre um determinado domínio [Gendreau e Potvin, 2005]. As heurísticas exploram uma área limitada do espaço de soluções e, em teoria, produzem respostas aceitáveis em um tempo computacional inferior ao tempo de resposta de uma solução exata [Loock e Hinnen, 2015]. Outra alternativa para encontrar boas soluções para problemas de otimização é a aplicação de metaheurísticas. De acordo com Osman e Laporte [1996] e Noronha et al. [2001], uma metaheurística é definida como um processo de geração iterativa, que conduz ao desenvolvimento de novas classes de heurísticas que melhoram consideravelmente a eficiência de algoritmos aproximados. Elas são projetadas para atacar os problemas complexos de otimização no qual os métodos heurísticos e clássicos de otimização não conseguiram ser eficazes e eficientes [Agarwal et al., 2015].

Outro meio para encontrar soluções para problemas de otimização é incorporar métodos heurísticos nas etapas de solução de problemas de programação inteira mista (*Mixed Integer Programming* – MIP), fornecendo ou melhorando soluções incumbentes com intuito de melhorar a performance do método exato. Para este fim, pode-se utilizar matheurísticas. As matheurísticas são algoritmos heurísticos desenvolvidos para mesclar técnicas de programação matemática com metaheurísticas. A principal característica que difere esta técnica de uma metaheurística é a de utilizar informações derivadas do modelo matemático do problema em questão [Boschetti et al., 2009]. Assim, o desenvolvimento de uma boa matheurística é útil tanto pela capacidade de gerar boas soluções para o problema como pela possibilidade ser integrada a um método exato baseado na resolução de problemas MIP.

Esse trabalho foca na aplicação de matheurísticas para a solução do Problema de Mínima Latência (PML), que é uma variação do Problema do Caixeiro Viajante (PCV). A partir de uma formulação matemática, serão apresentados diferentes métodos matheurísticos construtivos e de busca local para a resolução do problema, testando e comparando os resultados obtidos na resolução de problemas teste disponíveis na literatura, a fim de demonstrar a eficácia da técnica. Esta é uma pesquisa aplicada, quantitativa e no que se refere ao procedimento técnico utilizado, trata-se de uma pesquisa experimental.

Esse trabalho está organizado da seguinte forma. Na Seção 2 o problema é definido e a formulação matemática é apresentada. A Seção 3 apresenta as matheurísticas implementadas. Uma série de experimentos e seus respectivos resultados são apresentados na Seção 4 e, finalmente na Seção 5, são expostas as conclusões e considerações finais.

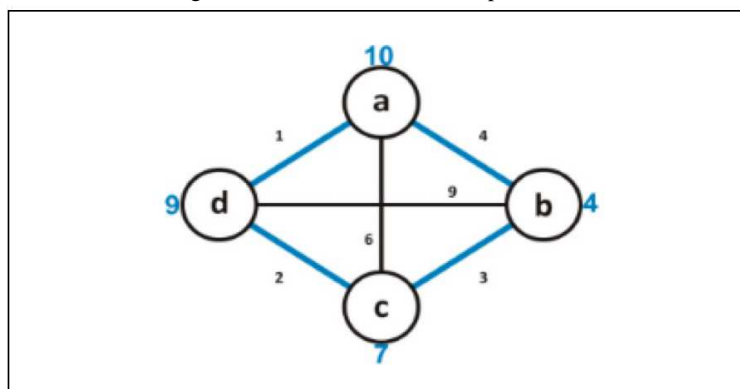
2. Definição do problema

O Problema de Mínima Latência (PML) – *Minimum Latency Problem* (MLP) – é uma variante do Problema do Caixeiro Viajante (PCV). Nesse problema, o objetivo é encontrar um circuito hamiltoniano iniciando em um único depósito, que minimize a soma dos tempos de espera (latência) dos consumidores [Sarubbi, 2008]. Esse problema também é conhecido na literatura como *Traveling Repairman Problem* [Tsitsiklis, 1992], *Delivery Man Problem* [Fischetti et al., 1993], *Cumulative Traveling Salesman Problem* [Bianco et al., 1993] e *School Bus Driver Problem* [Chaudhuri et al., 2003].

O PML pode ser aplicado, por exemplo, a um serviço de entrega de produtos a clientes em que o foco está em reduzir o tempo de espera para o recebimento e não no custo operacional da entrega. De forma geral, se diz que o PML é orientado ao cliente, enquanto o PCV é orientado ao fornecedor [Archer e Williamson, 2003]. Uma situação típica ocorre quando um distribuidor precisa fazer a entrega de várias mercadorias, para diversos clientes em localidades distintas, visando atendê-los no menor tempo de entrega possível. Este tempo de entrega é calculado da saída do depósito até o destino e o tempo de retorno para o depósito [Wu et al., 2004].

A principal diferença entre o PML e o PCV está justamente na função objetivo, pois ambos podem ser mapeados facilmente para um grafo, gerando um problema NP-Completo. Para exemplificar as diferenças entre os problemas, uma analogia é feita levando-se em consideração um grafo não-direcionado ponderado, conforme a Figura 1. Cada nó exibe o tempo total de espera e as arestas correspondem a uma unidade de tempo entre um nó e outro. O caminho destacado com a cor azul representa uma solução viável tanto para o PCV quanto para o PML, porém a função de avaliação para ambos resulta em valores diferentes. Enquanto o objetivo do PCV é minimizar o tempo total de visita dos nós, o objetivo para o PML é minimizar a soma das latências de cada nó e essa latência varia de acordo com uma dada solução.

Figura 1: Grafo não-direcionado ponderado.



O caminho p destacado ($p = (a, b, c, d, a)$) é uma solução viável para o PCV e também para o PML, porém o resultado da função objetivo para cada problema é calculado de forma diferente. Para o PCV, o resultado total t é calculado conforme a expressão (1) a seguir:

$$t = 4 + 3 + 2 + 1 \quad (1)$$

$$t = 10 \quad (2)$$

Levando-se em consideração que os valores de cada aresta significam valores de unidades de tempo para a conclusão do percurso, o caminho p visa minimizar o tempo de retorno ao nó inicial, ou depósito. Entretanto, analisando isoladamente cada nó e o seu tempo de espera, o problema toma outras proporções:

- Nó b : 4 unidades de tempo para concluir o percurso $A - B$;
- Nó c : 7 unidades de tempo (4 unidades de tempo para concluir a visita ao nó b e 3 unidades de tempo para concluir o percurso $B - C$);
- Nó d : 9 unidades de tempo (7 unidades de tempo para concluir a visita ao nó c e 2 unidades de tempo para concluir o percurso $C - D$);
- Nó a : 10 unidades de tempo (9 unidades de tempo para concluir a visita ao nó d e 1 unidade de tempo para concluir o percurso $D - A$);

Como a função objetivo do PML é a soma do tempo de espera (latência) de cada nó, tem-se o seguinte resultado (latência total l_t), na expressão (3), utilizando esse caminho:

$$\begin{aligned}
 l_t &= 4 + (4 + 3) + (4 + 3 + 2) + (4 + 3 + 2 + 1) \\
 l_t &= (4 \times 4) + (3 \times 3) + (2 \times 2) + (1 \times 1) \\
 l_t &= 30
 \end{aligned} \tag{3}$$

Através de uma observação superficial na maneira como o cálculo é realizado, pode-se tentar otimizar o caminho deslocando o nó com menor latência para o início do percurso, pois é ele que provoca o maior acúmulo de tempo. Um caminho seguindo essa visão pode ser configurado como (a, d, c, b, a) . Entretanto, esse caminho acarretará em uma latência total de 43 unidades de tempo.

O resultado desse novo caminho é maior que o resultado anterior, pois aplicamos o deslocamento apenas no primeiro elemento. Aplicando-se essa abordagem em todos os elementos teremos o caminho (a, b, c, b, a) com uma latência total de 20 unidades de tempo. Esse deslocamento não é a única técnica, e nem de longe a melhor técnica para a resolução do problema.

Sarubbi [2008] apresenta uma formulação matemática para o problema. Considere o grafo $G = (V, A)$, em que V é um conjunto de vértices e A é um conjunto de arcos. Suponha que exista um vértice-origem $v_o = 1 \in V$ e, para cada vértice $k \in V$, que inclui também o vértice 1, uma demanda única deve ser entregue por um circuito que minimize a soma dos tempos de espera de cada vértice. Um modelo de otimização linear inteira mista pode ser construído utilizando, como parâmetros de custo, os valores de tempos c_{ij} necessários para percorrer os arcos (i, j) . As variáveis de decisão x_{ij} indicam o uso ou não de um arco (i, j) , na forma:

$$x_{ij} = \begin{cases} 0, & \text{se o veículo atravessa o arco } (i, j); \\ 1, & \text{caso contrário.} \end{cases} \tag{4}$$

Além disso, define-se também as variáveis de decisão f_{ij} , que mostram o fluxo total de produtos que passam pelo arco (i, j) . Portanto, a formulação para o Problema de Mínima Latência, proposta por Sarubbi [2008], é dada por:

$$\begin{aligned}
& \min \sum_{(i,j) \in A} c_{ij} f_{ij} & (5) \\
& \text{sujeito a} \\
& \sum_{i \in V | i \neq j} x_{ij} = 1 \quad \forall j \in V & (6) \\
& \sum_{j \in V | i \neq j} x_{ij} = 1 \quad \forall i \in V & (7) \\
& \sum_{j \in V | j \neq 1} f_{1j} = n & (8) \\
& \sum_{i \in V} f_{ik} - \sum_{j \in V} = 1 \quad \forall k \in V \setminus \{1\} & (9) \\
& \sum_{i,j \in V} f_{ij} = \frac{n(n+1)}{2} & (10) \\
& f_{ij} \leq |V| x_{ij} \quad \forall i, j \in V & (11) \\
& f_{ij} \geq 0 \quad \forall i, j \in V & (12) \\
& x_{ij} \in \{0, 1\} \quad \forall i, j \in V & (13)
\end{aligned}$$

A função objetivo (5) minimiza o tempo total de espera para entrega dos produtos. As restrições (6) e (7) garantem que só chega e sai apenas um arco de cada vértice. O conjunto de restrições (8) assegura que o fluxo que sai do vértice-origem é igual ao número de vértices. As restrições (9) forçam que em cada vértice seja entregue sua demanda unitária. A restrição (10) é uma restrição redundante, que restringe o somatório de todos os fluxos na rede como sendo a cardinalidade de V . Segundo Sarubbi [2008], esta restrição é responsável por uma melhora nos *gaps* de relaxação linear da formulação. As restrições (11) e (12) garantem que não pode haver fluxo nos vértices que não foram escolhidos na solução ótima e que esse fluxo é positivo. Finalmente, as restrições (13) asseguram a integralidade da solução.

3. Métodos heurísticos utilizados

Para desenvolvimento do trabalho em questão, foram utilizados os seguintes métodos: *Relaxation Enforced Neighborhood Search* (RENS) [Berthold, 2007], *Feasibility Pump* (FP) [Fischetti et al., 2005] e *Local Branching* (LB). Os procedimentos são descritos nas seções subsequentes.

3.1. Relaxation Enforced Neighborhood Search (RENS)

O algoritmo de busca RENS é um método que integra a heurística *Large Neighborhood Search* (LNS) [Shaw, 1998], que, por sua vez, são baseadas no método *Local Search* (LS) [Berthold, 2007]. Muito utilizado na solução de problemas MIP considerados NP-difíceis, o método LNS tem, como critério de busca, a definição de uma vizinhança de tamanho consideravelmente grande em um determinado ponto e, então, com apenas uma iteração, a vizinhança é totalmente e/ou parcialmente explorada na busca da solução ótima ou de uma solução viável. Segundo [Berthold, 2007], o método RENS investiga o conjunto de todos os arredondamentos possíveis de uma solução relaxada do LP (*Linear Problem*).

Após relaxar o problema inicial, resolver o problema relaxado e de posse da solução, que pode conter valores fracionários que vão de 0 (zero) a 1 (um), a matheurística RENS fixa o valor das variáveis relaxadas que são inteiras na solução ótima do problema relaxado. Como consequência, é criado um subproblema MIP reduzido, uma vez que parte das variáveis foram fixadas, que será utilizado para realizar o processo de busca da solução ótima, viável ou em alguns casos de uma solução inicial.

3.2. Feasibility Pump (FP)

Fischetti et al. [2005] propuseram uma matheurística para construir soluções factíveis de forma geral para um MIP, chamada *Feasibility Pump* (FP). Segundo Bertacco et al. [2007], o método FP trabalha com um par de soluções (x^*, \bar{x}) com $x^* \in P$ e \bar{x} inteiro, que são atualizados iterativamente, com o objetivo de reduzir tanto quanto possível a distância entre elas, dada por $\Delta(x^*, \bar{x})$. Para ser mais específico, inicia-se com qualquer $x^* \in P$, e uma solução inteira tipicamente infactível \bar{x} como arredondamento de x^* . Em cada iteração de FP, chamada *pumping cycle*, \bar{x} é fixado e se encontra, através da otimização linear, o ponto x^* que é o mais próximo possível de \bar{x} . Se $\Delta(x^*, \bar{x}) = 0$, então é uma solução factível para o MIP, e o método finaliza. Caso contrário, \bar{x} é substituído pelo arredondamento de x^* , de modo a reduzir ainda mais $\Delta(x^*, \bar{x})$, e o processo é iterado. Caso ocorra uma ciclagem sem obtenção de uma solução viável inicial, é adotado a estratégia de fazer uma busca local na vizinhança $\Delta(x^*, x) \leq k$, para k inicialmente igual a 1. O valor de k é incrementado em uma unidade, caso a vizinhança anterior não contenha nenhuma solução viável.

3.3. Local Branching (LB)

Conforme Fischetti e Lodi [2003], a matheurística Local Branching (LB) pode ser descrita supondo uma dada solução de referência factível \bar{x} de modo que um dos objetivos seja encontrar uma solução melhorada que não esteja tão distante de \bar{x} . No método, é adicionada uma restrição de *branching* ($\Delta(x, \bar{x}) \leq k$), que visa definir a distância máxima k entre as duas soluções (x, \bar{x}) . Além disso, a heurística realiza a busca, de acordo com uma dada solução, em uma vizinhança de tamanho k pré-definido, denominada $k - OPT$ que passa a ser considerada como o conjunto de soluções que atendem à restrição adicional, sendo k um número de valor inteiro que expressa o raio de busca entre a solução \bar{x} e a solução a ser encontrada, para que então seja minimizado o tempo de busca e o esforço computacional para determinar a solução. Após realizar a busca na ramificação ($\Delta(x, \bar{x}) \leq k$), a ramificação ($\Delta(x, \bar{x}) > k$) é explorada. A exploração nesta área é feita levando-se em consideração a nova solução \bar{x} .

Para Fischetti et al. [2005], ao definir o valor do tamanho do parâmetro de vizinhança k , é possível tornar a vizinhança suficientemente pequena para que o tempo de busca seja minimizado, mas, ao mesmo tempo, grande o suficiente para provavelmente conter soluções melhores do que \bar{x} . Para esse trabalho foi adotada uma vizinha de tamanho $k = 1$.

4. Experimentos e resultados

Todos os experimentos foram realizados em uma máquina dotada de um processador Intel Core i7 de 2.9 GHz, 16GB de memória RAM, com o sistema operacional macOS High Sierra. As metaheurísticas foram implementadas utilizando a linguagem de programação C++ e foi utilizado o *solver* IBM CPLEX na versão 12.63.

Os problemas-teste utilizados para os experimentos são os mesmos utilizados por Sarubbi [2008]. Eles foram gerados obtendo custos inteiros aleatórios entre 1 e 100, armazenando-os em uma matriz de adjacência, seguindo os mesmos passos que Fischetti et al. [1993]. Os problemas-teste foram arranjados em 5 grupos, de acordo com o número de elementos, e, para cada grupo, foram gerados 10 problemas-teste. Os grupos problemas-teste analisados no trabalho possuem 10, 12, 15, 20 e 25 elementos.

Cada problema-teste foi processado usando uma construção inicial encontrada pelas matheurísticas RENS e FP, a fim de verificar qual construção inicial gerava melhores resultados. Após a construção da solução inicial, a matheurística *Local Branching* é aplicada como refinamento, com o objetivo de verificar qual construção inicial apresenta melhores resultados finais em relação ao *gap* (definido na expressão 14) do valor ótimo do problema-teste e também quanto ao tempo de execução. Para cada problema-teste, é conhecido o resultado ótimo, apresentado em Sarubbi [2008].

4.1. Avaliação dos problemas-teste

Como para cada problema-teste avaliado, o respectivo valor ótimo é conhecido, pode-se verificar o desempenho do algoritmo proposto. Ao final do processamento de cada problema-teste, foram registrados o custo da melhor solução encontrada, o tempo de execução para encontrar esta solução, a média dos valores das soluções encontradas e a média do tempo de execução. A métrica utilizada para quantificar o desempenho do resultado encontrado é a diferença percentual entre o resultado obtido e o resultado do melhor valor conhecido. Esse valor é dado em porcentagem e é calculado na forma:

$$gap(\%) = 100 \times \frac{(\text{resultado ótimo} - \text{resultado encontrado})}{\text{resultado ótimo}}$$

4.2. Experimentos

Dado o objetivo principal do trabalho, que é avaliar uma melhor combinação entre uma matheurística para geração de uma solução inicial, refiná-la e realizar a comparação com uma execução padrão do *solver* CPLEX com tempo definido, é necessário estabelecer os seguintes grupos de experimentos:

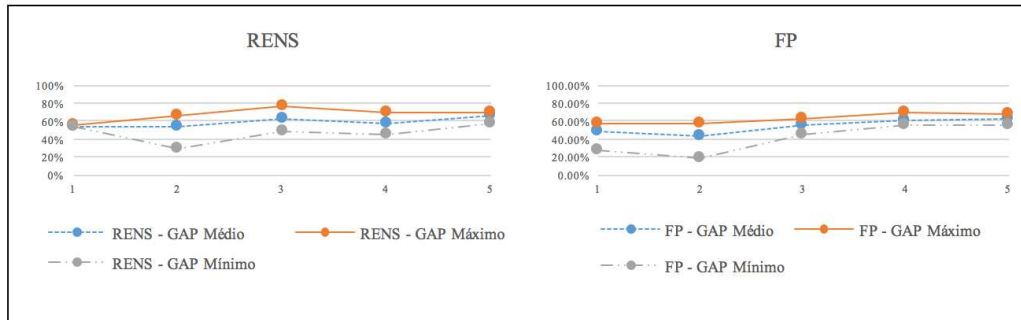
- (i) *RENS*: execução utilizando apenas o método *RENS* para geração da solução inicial;
- (ii) *FP*: execução utilizando apenas o método *FP* para geração da solução inicial;
- (iii) *RENS+LB*: execução do método *Local Branching* refinando a solução gerada através do método *RENS*, com tempo limite para resolução do MIP igual a 10 segundos;
- (iv) *FP+LB*: execução do método *Local Branching* refinando a solução gerada através do método *FP*, com tempo limite para resolução do MIP igual a 10 segundos;
- (v) *Solver Padrão*: execução padrão do solver limitando o tempo a 200 segundos.

Após realizar esses grupos de experimentos, foi possível aferir a melhor combinação, gerando assim um grupo final de experimentos para comparação de resultados com a execução padrão do *solver* limitada em 200 segundos. Dado o fato de existirem vários problemas-testes em um mesmo grupo de experimentos, foi utilizada, como métrica de desempenho de uma determinada implementação, o valor de *gap* médio para aquele conjunto de problemas-teste.

4.3. Métodos de construção da solução inicial

Conforme descrito, foram analisadas duas estratégias de construção da solução inicial: *RENS* e *FP*. Porém, dos 50 problemas-teste selecionados, o método *RENS* não conseguiu gerar uma solução inicial para 15, falhando assim em 30% dos casos. Em contrapartida, o método *FP* conseguiu gerar solução inicial para todos os problemas testes selecionados, porém com um tempo superior. Dessa forma, para realizar os comparativos entre as implementações dos métodos, os problemas-testes que não obtiveram solução inicial através do método *RENS* foram removidos dos grupos, reduzindo assim para 35 o número total de problemas-teste submetidos à matheurística de refinamento *Local Branching*.

Figura 2: Comparação entre os gaps dos resultados gerados pelos métodos construtivos RENS e FP.



Conforme destacado pela Figura 2, os *gaps* médios dos métodos foram similares, porém, o tempo gasto pelo FP é elevado em relação ao tempo gasto pelo método RENS.

4.4. Método de refinamento da solução

Os resultados apresentados nessa seção demonstram a relação dos *gaps* encontrados ao refinar as soluções iniciais geradas pelos métodos RENS e FP executando a matheurística *Local Branching*.

Figura 3: Comparação entre os *gaps* dos resultados gerados pelo método de refinamento Local Branching aos resultados apresentados pela implementação dos métodos RENS e FP.

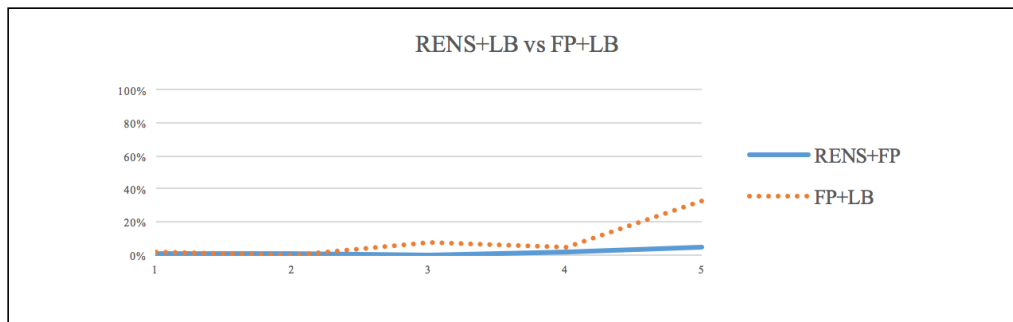


Tabela 1: Resultados comparativos entre os métodos RENS+LB e FP+LB.

Grupo	RENS+FP		FP+LB	
	GAP Médio(%)	Tempo Médio (s)	GAP Médio(%)	Tempo Médio (s)
1	-0.97%	1.99	-1.49%	3.90
2	-0.34%	4.69	-0.21%	13.80
3	0.00%	61.75	-8.11%	193.00
4	-1.53%	118.78	-5.02%	333.47
5	-5.11%	135.06	-34.43%	440.34

4.5. Comparação de resultados

Conforme destacado na seção anterior, a melhor combinação foi a do grupo de experimentos RENS+LB. Sendo assim, esse grupo foi comparado com os resultados obtidos pelo *solver* padrão, limitado ao tempo de execução de 200 segundos, para a resolução do MIP. A Tabela 2 destaca os resultados obtidos. Observa-se que apenas para o Grupo 2 o *gap* médio foi superior, contudo, com um tempo computacional consideravelmente inferior.

Tabela 2: Resultados comparativos entre o método RENS+LB e a execução do solver padrão limitado em 200 segundos.

Grupo	RENS+LB		Solver Padrão	
	GAP Médio(%)	Tempo Médio (s)	GAP Médio(%)	Tempo Médio (s)
1	-0.90%	1.99	-0.97%	2.46
2	-0.34%	4.69	-0.24%	103.78
3	0.00%	61.75	-0.10%	200.01
4	-1.43%	118.78	-3.40%	200.01
5	-5.11%	135.06	-6.01%	200.01

5. Conclusões e trabalhos futuros

A implementação da matheurística RENS apresentou resultados similares na construção de soluções iniciais em relação à implementação do FP com um tempo inferior, porém falhou em gerar resultados para 30% dos problemas testes analisados. Sugere-se utilizar a implementação FP quando houver falha no RENS. A combinação que apresentou melhores resultados para o conjunto de problemas teste foi RENS+LB.

Ao comparar a utilização da técnica proposta para o PML, observa-se resultados superiores que a execução padrão do solver em um tempo limitado em 200s.

Como trabalhos futuros, tanto as implementações do RENS quanto do FP devem ser avaliadas para se adequar melhor ao PML gerando soluções iniciais melhores. No caso do FP, outras estratégias para pós-clangem podem ser adotadas. Mais testes devem ser executados a fim de explorar outros possíveis parâmetros das matheurísticas, como tempo para resolução do MIP e soluções exploradas. Diversos trabalhos publicados utilizam problemas testes da TSPLIB como critério de comparação. Desta forma, trabalhos futuros deverão analisar estes problemas testes.

6. Agradecimentos

Os autores agradecem ao CEFET/MG, CAPES, CNPq e FAPEMIG pelo apoio ao desenvolvimento desta pesquisa.

Referências

- Agarwal, A., Colak, S., e Erenguc, S. (2015). Metaheuristic methods. In *Handbook on Project Management and Scheduling Vol. 1*, p. 57–74. Springer.
- Archer, A. e Williamson, D. P. (2003). Faster approximation algorithms for the minimum latency problem. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, p. 88–96. Society for Industrial and Applied Mathematics.
- Bertacco, L., Fischetti, M., e Lodi, A. (2007). A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1):63–76.
- Berthold, T. (2007). Rens – relaxation enforced neighborhood search. Technical Report ZIB-07-28, Konrad-Zuse-Zentrum für Informationstechnik.
- Bianco, L., Mingozzi, A., e Ricciardelli, S. (1993). The traveling salesman problem with cumulative costs. *Networks*, 23(2):81–91.
- Blum, C. e Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308.
- Boschetti, M. A., Maniezzo, V., Roffilli, M., e Röhrler, A. B. (2009). Matheuristics: Optimization, simulation and control. In *International Workshop on Hybrid Metaheuristics*, p. 171–177. Springer.

- Chaudhuri, K., Godfrey, B., Rao, S., e Talwar, K. (2003). Paths, trees, and minimum latency tours. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, p. 36–45. IEEE.
- Fischetti, M. e Lodi, A. (2003). Local branching. *Mathematical Programming*, 98(2):23–47.
- Fischetti, M., Glover, F., e Lodi, A. (2005). The feasibility pump. *Mathematical Programming*, 104(1):91–104.
- Fischetti, M., Laporte, G., e Martello, S. (1993). The delivery man problem and cumulative matroids. *Operations Research*, 41(6):1055–1064.
- Gendreau, M. e Potvin, J.-Y. (2005). Metaheuristics in combinatorial optimization. *Annals of Operations Research*, 140(1):189–213.
- Loock, M. e Hinnen, G. (2015). Heuristics in organizations: A review and a research agenda. *Journal of Business Research*, 68(9):2027–2036.
- Noronha, T., da SILVA, M., e Aloise, D. (2001). Uma abordagem sobre estratégias metaheurísticas. *Revista Eletrônica de Iniciação Científica (REIC), Ano I*, 1.
- Osman, I. H. e Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5):511–623.
- Sarubbi, J. F. M. *Problemas de roteamento com custos de carga*. Tese de Doutorado, UFMG, 2008.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In Maher, M. e Puget, J.-F., editors, *Principles and Practice of Constraint Programming — CP98*, p. 417–431, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Tsitsiklis, J. N. (1992). Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22(3):263–282.
- Wu, B. Y., Huang, Z.-N., e Zhan, F.-J. (2004). Exact algorithms for the minimum latency problem. *Information Processing Letters*, 92(6):303–309.