LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

PLANEJAMENTO E ROTEAMENTO DE TRENS UTILIZANDO A METAHEURÍSTICA VNS E TÉCNICAS EM GRAFOS

Higor D'Talles Costa, Marcia Helena Moreira Paiva, Helder Roberto de Oliveira Rocha
Universidade Federal do Espírito Santo
Departamento de Engenharia Elétrica, Vitória, Espírito Santo, Brasil
higorcosta@gmail.com, marcia.paiva@ufes.br, helder.rocha@ufes.br

RESUMO

A eficiência do transporte ferroviário pode ser aumentada realizando poucas paradas e trafegando à velocidade ótima para cada trecho de ferrovia. Com um bom planejamento esses objetivos podem ser alcançados. A solução para os problemas de planejamento e roteamento pode ser obtida pela minimização de dois objetivos: número de conflitos entre trens e tempo de parada dos trens em circulação. Foi implementado um algoritmo multiobjetivo baseado no esquema *Skewed VNS*. O algoritmo foi testado em grafos planares aleatórios de até 364 vértices, 442 arestas com circulação de 73 trens. Foram criados cenários variando o número de rotas, os pesos dos vértices e utilizando a métrica de centralidade *betweenness* no processo de roteamento. Foi construída a Fronteira de Pareto cuja qualidade foi verificada por três métricas sugeridas na literatura e uma quarta proposta neste trabalho. Os resultados obtidos foram muito bons frente aos objetivos propostos e à complexidade das topologias testadas.

PALAVRAS CHAVE. Planejamento e roteamento de trens, Operação ferroviária, Otimização multiobjetivo.

L&T – Logística e Transportes, MH – Metaheuristicas, TAG – Teoria e Algoritmos em Grafos

ABSTRACT

Railway transportation efficiency can be increased by making few stops and running at optimal speed regarding each part of the railway. With good planning these objectives can be achieved. Solution for both problems can be obtained by optimizing two objectives: number of conflicts among trains and stopping time of running trains. We implemented a multiobjective algorithm based on the Skewed VNS scheme. The algorithm was tested in random planar graphs of up to 364 vertices, 442 edges with 73 running trains. Scenarios were created varying number of routes, weight of edges and using the betweenness measure of centrality in the routing process. It was built the Pareto Front whose quality was checked by three measures suggested by some papers and one fourth proposed by this research. The results were very good when compared to proposed objectives and to the complexity of the topologies tested.

KEYWORDS. Train scheduling and routing. Railway operations. Multiobjective optimization.

L&T – Logistics and Transport, MH – Metaheuristics, TAG – Theory and Graph Algorithms

https://proceedings.science/p/106710?lang=pt-br

1. Introdução

Dada uma rede formada por nós e arestas que os conectam, o problema de roteamento pode ser definido como sendo a busca pelos pontos intermediários que conectam uma origem e um destino para que uma entidade possa trafegar ao longo de uma rede.

Quando se trata do tráfego de trens, cada trecho de ferrovia só pode ser utilizado por um trem de cada vez em um dado intervalo de tempo. Cada trem é considerado como sendo uma entidade indivisível que trafega pela ferrovia. Ao longo de sua rota um trem pode ir de uma linha para outra somente em pontos específicos que fazem o chaveamento entre as linhas disponíveis. Ao utilizar redes modeladas por grafos esses pontos de chaveamento são equivalentes aos vértices. Desde que um trecho de uma linha esteja livre, os trens podem trafegar em qualquer que seja o sentido.

O planejamento de trens está intrinsecamente ligado ao controle de fluxo na malha ferroviária. Para lidar com os desafios do problema de planejamento e roteamento de trens é proposto nesta pesquisa um método heurístico composto por três etapas: (a) geração de rotas viáveis via grafos, (b) detecção e minimização de conflitos e (c) redução de tempos de atraso. Na primeira etapa é utilizado o algoritmo proposto por [Yen, 1971]. Na segunda etapa é utilizado um algoritmo *Multiobjective Skewed Variable Neighborhood Search* (MO-SVNS) para minimizar tanto o número de conflitos quanto o tempo de atraso dos trens. Por fim, na última etapa, foi proposta uma heurística de busca e redução de tempos de atraso para otimizar os resultados obtidos na segunda etapa.

No tocante ao planejamento de trens, é fundamental que sejam evitadas as paradas indesejadas. Essa é uma questão muito importante para o transporte ferroviário já que um dos maiores custos da ferrovia é referente ao consumo de combustível [Brasil, 2018]. Nesse caso, é preciso escolher bem o local e o tempo em que um dado trem ficará parado. Surge então a necessidade de Sistemas de Suporte à Decisão (SSD), conforme mencionado por [Samà et al., 2017]. É importante que tais sistemas tenham respostas rápidas a eventos que possam ocorrer ao longo do tempo. Os SSD compreendem sistemas muito importantes para os operadores de tráfego. Quanto aos eventos que podem ocorrer, dentre eles estão os conflitos entre rotas. Assim, é necessário desenvolver algoritmos capazes de detectar e eliminar esses conflitos sem deixar de lado a necessidade de reduzir os tempos de parada dos trens.

2. Trabalhos Relacionados

Segundo [Pinedo, 2005], o planejamento e a programação são processos decisórios amplamente utilizados nos transportes. Essas atividades dependem de métodos matemáticos e heurísticos para alocar recursos limitados a atividades que precisam ser realizadas.

Para lidar com problemas de expansão de capacidade de redes a baixos custos, [Höller e Voß, 2006] fizeram uma abordagem comparando quatro técnicas de otimização: programação linear, cálculo do menor caminho pelo algoritmo de *Dijkstra*, heurística de busca *multistart* e um algoritmo com base no *Greedy Randomized Adaptive Search Procedure* (GRASP). Os três últimos são métodos heurísticos bastante utilizadas em problemas de otimização de rotas. O mesmo tipo de problema foi estudado por [Grover e Doucette, 2001]. Em todos os casos os tempos computacionais demandados pelos métodos heurísticos até a obtenção da solução foram inferiores àqueles obtidos por métodos baseados em programação linear.

Segundo [Cai et al., 1998], os problemas de roteamento podem ser modelados com ferramentas de otimização matemática bastante sofisticadas levando a resultados muito expressivos. Um dos algoritmos muito utilizados para técnicas com programação linear é o *branch-and-bound*. Esse método foi proposto pela primeira vez por [Szpigel, 1973] para resolver o problema de planejamento de trens em linha singela, isto é, linha única para passagem de trens em ambos sentidos.

Entretanto, [Cai et al., 1998] defendem que essas ferramentas devam ser restritas à solução de problemas bem reduzidos e com poucas entidades trafegando nas redes, já que se trata de um problema de difícil solução com características estocásticas e não lineares.

Nesta pesquisa a ferrovia será modelada via grafos e por isso é essencial o estudo de abordagens que reportem a essa ideia. Além dos grafos remeterem a topologias mais simples de se analisar, essa abordagem traz outras vantagens já que há várias medidas de centralidade que podem trazer bons resultados para as análises. Além disso, há muitas técnicas de roteamento que utilizam os grafos como ponto de partida.

Em relação à geração de rotas, um dos fatores que se deve levar em consideração é o custo, que pode ser analisado sob diferentes aspectos. Contudo, independentemente do aspecto considerado, em geral o que se deseja é minimizá-lo. Neste trabalho o custo está relacionado ao comprimento dos trechos, isto é, ao peso dos vértices dos grafos. Para uma determinada rota o custo é igual à soma dos comprimentos de todos os trechos que a compõem. Com o intuito de calcular os menores caminhos que conectam dois nós de uma rede, foi proposto por [Yen, 1971] um algoritmo para o cálculo dos *K Shortest Loopless Paths*. Esse algoritmo se baseia no que foi proposto por [Dijkstra, 1959], onde a partir de um determinado vértice são calculados os caminhos mais curtos para qualquer outro vértice da rede. Uma alternativa complementar para a geração de rotas ocorre pelo emprego da métrica *betweenness*. Essa métrica é proporcional à probabilidade de um nó ser intermediário na comunicação entre outros nós da rede. Então, se os pesos dos vértices dos grafos forem substituídos pelo valor da métrica *betweenness* referente a cada um dos vértices, as referências passadas para o algoritmo de Yen serão diferentes. Com isso, as regiões da rede onde havia uma probabilidade maior de congestionamento poderão deixar de ser tão disputadas no cálculo das rotas.

Um dos métodos utilizados para resolver conflitos foi proposto por [Ingolotti et al., 2005]. Ele define uma série de restrições e uma em particular contribui para o tratamento dos conflitos. Quando uma ou mais restrições são violadas por dois trens que participam de um conflito, é atribuído um atraso a um deles até que o conflito existente seja eliminado. A utilização de atrasos para resolver conflitos foi também utilizada por [Şahin, 1999], que estudou o planejamento de trens em tempo real, resolvendo os conflitos à medida em que eles ocorriam.

Em algumas pesquisas foi observado que ao eliminar todos os conflitos entre os trens em circulação nem sempre foram obtidos os menores tempos de atraso, isto é, os menores custos do processo de planejamento e roteamento. Os atrasos são gerados propositalmente para que condições seguras de ultrapassagem sejam criadas contribuindo para a eliminação dos conflitos. Entretanto, [Landex et al., 2006] propõe um método de redução desses tempos. O efeito prático é que os trens possam circular mais próximos uns dos outros sem comprometer a segurança.

3. Modelagem do Problema

Neste trabalho a métrica *betweenness* foi utilizada no processo de roteamento como alternativa para evitar grandes congestionamentos entre as rotas dos trens. Trata-se de uma métrica importante para o controle da comunicação entre os nós [Freeman, 1977]. No contexto deste trabalho o uso dessa métrica contribui para que os caminhos relativos às rotas dos trens sejam melhor distribuídos pelo grafo, evitando a sobrecarga de alguns vértices ao utilizar algoritmos para a busca de menores caminhos entre dois vértices quaisquer. O cálculo da métrica *betweenness* foi realizado com base no algoritmo de [Brandes, 2001]. Esse algoritmo é capaz de trabalhar com grafos ponderados ou não ponderados, podendo calcular os valores da métrica em função dos vértices. Utilizando o grafo ponderado para o cálculo da métrica obtém-se a *betweenness ponderada*. Um exemplo de aplicação desta última pode ser visto no trabalho de [Cousineau et al., 2015].

https://proceedings.science/p/106710?lang=pt-br

A Figura 1 mostra um modelo simplificado de uma ferrovia real em linha dupla, cujas linhas são dadas por *L1* e *L2*. Essa figura ilustra como se deu a conversão para o modelo segundo a teoria dos grafos. Os vértices do grafo representam os trechos da ferrovia e as arestas representam as conexões entre eles. Na Figura 1a as marcações com letras separam os trechos da ferrovia. As conexões entre as linhas estão numeradas de 15 a 20 na Figura 1a. No modelo em grafo, Figura 1b, os vértices estão numerados de 1 a 14 e as arestas são representadas pelas conexões entre eles. Uma vez compreendida a lógica de criação do modelo ferroviário pela teoria dos grafos, esse modelo deve ser abstraído para as redes modeladas via grafos aleatórios planares. De acordo com [Benjamin et al., 2015], um grafo é planar se ele pode ser desenhado em um plano sem que duas de suas arestas se cruzem. Esses grafos foram gerados aleatoriamente com grau médio de 2,4 para manter a proporcionalidade de vértices e arestas entre as topologias. O grau médio de um grafo é dado por *2m/n*, onde m é o número de arestas e n o número de vértices.

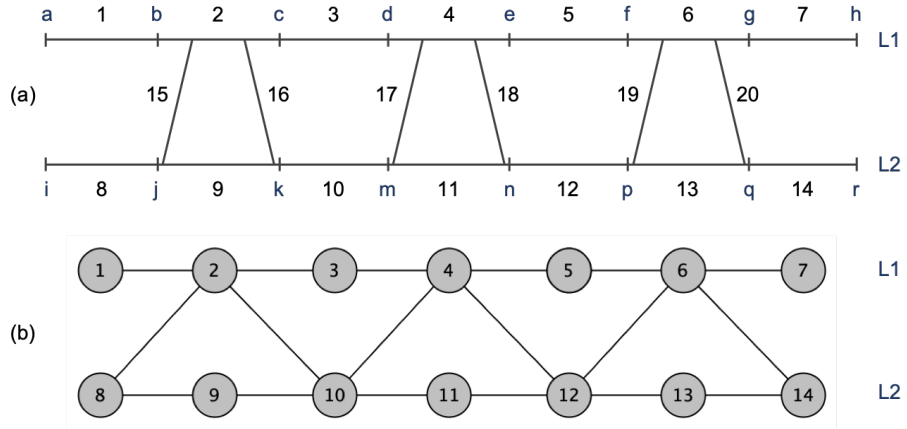


Figura 1: Equivalência entre: (a) ferrovia real e (b) ferrovia modelada pela teoria dos grafos.

As rotas para cada um dos trens foram obtidas com base no algoritmo *K Shortest Loopless Paths* proposto por [Yen, 1971]. Nesta pesquisa optou-se por realizar experimentos considerando dois cenários distintos. No primeiro deles foi considerada somente a rota de menor custo para cada um dos trens. No segundo cenário foram calculadas as três rotas de menor custo para cada um dos trens e então foi escolhida aquela que gerasse o menor número de conflitos ao inserir o trem em questão no contexto dos trens já escolhidos nas fases anteriores do processo de escolha de rotas. Para cada trecho foi estabelecido um tempo médio de cruzamento. Esse tempo é levado em consideração no processo de planejamento dos trens.

Quando a programação inicial dos trens é concluída e todas as rotas são escolhidas é necessário identificar e suprimir os conflitos existentes. A identificação dos conflitos é realizada pela comparação de todos os trechos que compõem a rota de cada um dos trens. Na primeira fase são observados os conflitos no espaço e, em seguida, é verificado se esses conflitos ocorrem em intervalos de tempo também conflitantes. Os conflitos são contabilizados por trechos e por pares de trens, conforme descrito no trabalho de [Costa et al., 2018].

4. Método de Solução

O método de solução do problema de planejamento e roteamento de trens compreende as seguintes etapas: (a) modelar a ferrovia segundo a teoria dos grafos, (b) definir regras de tráfego, (c) avaliar técnicas de planejamento e roteamento aplicáveis ao transporte ferroviário, (d) utilizar a métrica *betweenness* e verificar sua influência, (e) avaliar técnicas de detecção e solução de conflitos

de rota, (f) definir o algoritmo MO-SVNS, (g) estressar o algoritmo MO-SVNS iniciando a busca pelo pior cenário possível, (h) implementar algoritmo capaz de reduzir os tempos de atraso dos trens, (i) construir a Fronteira de Pareto para avaliar as melhores soluções encontradas segundo os objetivos que devem ser otimizados, (j) propor métrica para auxiliar na escolha da Fronteira de Pareto quando não houver referência na literatura e (k) implementar métricas capazes de mensurar a qualidade da Fronteira de Pareto.

5. Heurística de Otimização

Como heurística de otimização principal desta pesquisa foi adotada a metaheurística *Variable Neighborhood Search* (VNS). Ela explora vizinhanças gradativamente mais distantes da solução atual. Essa metaheurística permite utilizar somente uma rotina de busca local e ainda assim convergir para soluções ótimas [Hansen e Mladenović, 2001].

Na busca pela solução ótima, pode ocorrer que ela esteja em um vale ainda mais profundo, mas em uma região muito distante da vizinhança sendo explorada. Ao tentar fazer saltos muito distantes pode ocorrer a degeneração do VNS. Para tentar lidar com essa possibilidade [Hansen e Mladenović, 2001] propõem que seja adicionado um fator α que permita a aceitação de resultados piores ao avaliar a função objetivo. O parâmetro α está relacionado à distância entre a solução corrente x e a solução x" gerada após a busca local. Assim, mesmo que possam ser encontradas soluções piores do que aquelas já conhecidas até o momento, logo em seguida outras melhores poderão ser descobertas até que o ótimo global possa ser encontrado. Esse esquema foi chamado de *Skewed VNS* (SVNS).

Tomando como base o *SVNS*, foi desenvolvido um algoritmo MO-SVNS, baseado no trabalho de [Rubio-Largo et al., 2010]. Em adição ao SVNS, nesse esquema além de ser feita a otimização das duas funções objetivo que se deseja minimizar neste trabalho, também é realizada a coleta dos pontos que dão origem à Fronteira de Pareto.

Nesta pesquisa foi utilizado um algoritmo de busca local do tipo *First Improvement*, onde um movimento é realizado assim que a direção de decida é identificada. A função objetivo referente ao tempo de parada dos trens é analisada no procedimento do MO-SVNS. Esse parâmetro é utilizado pelo MO-SVNS como critério de aceitação de uma solução viável como sendo a nova solução corrente.

Quanto às estruturas de vizinhança utilizadas pelo MO-SVNS, foram definidas seis. Elas estão descritas na Tabela 1.

Tabela 1: Vizinhanças utilizadas pelo MO-SVNS.

Vizinhança	Descrição da vizinhança
Viz–1	Caso o trem fique parado em algum trecho, permutar dois dos seus tempos de parada mesmo que em um dos trechos o tempo seja nulo.
Viz–2	Onde tiver tempo de parada para um dado trem, escolher um trecho e reduzir o tempo em 10%. Em seguida, o tempo referente à redução é somado ao tempo de parada do trecho anterior.
Viz–3	Onde o trem tiver uma parada superior a 5 minutos, escolher um desses trechos e reduzir o tempo em 10%.
Viz–4	Para um trem e um trecho escolhidos ao acaso, inserir um tempo de parada padrão para criar oportunidades de ultrapassagem de trens.
Viz–5	Para o trem que estiver envolvido no maior número de conflitos, inserir um tempo de parada padrão no trecho inicial de sua rota para atrasar sua partida.
Viz–6	Para um trem escolhido ao acaso, inserir um tempo de parada padrão no trecho inicial de sua rota para atrasar sua partida.

Uma vez terminado o processo do MO-SVNS, onde todos os atrasos e cruzamentos necessários foram ajustados, é esperado que alguns atrasos sejam maiores do que o necessário. Com base nisso, foi criada uma heurística capaz de localizar e remover tempos de parada desnecessários.

https://proceedings.science/p/106710?lang=pt-br

Todas as reduções são aceitas somente se não gerarem conflitos em relação à solução encontrada pelo MO-SVNS. A Figura 2 mostra alguns exemplos de atuação dessa heurística. O $TipoI$ compreende os tempos de parada analisados para cada trem individualmente. O $TipoII$ é aquele em que é procurada uma janela de tempo em que todos os trens se encontram parados.

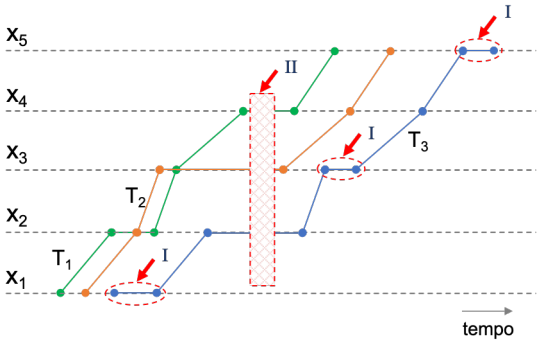


Figura 2: Paradas indesejadas $Tipo\ I$ e $Tipo\ II$.

Os problemas multiobjetivo são mais difíceis de serem resolvidos do que aqueles com um único objetivo. Isso se deve ao fato de que o processo de otimização deve encontrar a melhor relação entre os objetivos do problema. Esses pontos onde ocorrem essas relações ótimas dão origem à Fronteira de Pareto [Bagchi, 1999]. A qualidade da fronteira pode ser mensurada por algumas métricas, sendo que nesta pesquisa foram utilizadas três bem conhecidas na literatura e uma quarta proposta pelo autor. Essas métricas serão descritas a seguir.

***Hypervolume* (HV)**: Essa métrica foi proposta por [Zitzler e Thiele, 1998]. No caso da otimização biobjetivo, cada solução não dominada da Fronteira de Pareto cobre uma área retangular. A união dessas áreas constitui a área total de cobertura da Fronteira de Pareto. Essa área define a métrica HV para problemas de otimização biobjetivo.

***Maximum Spread* (MS)**: A dispersão dos valores da Fronteira de Pareto deve ser maximizada de modo que as soluções não dominadas cubram um amplo conjunto de soluções dominadas. Esse é um requisito destacado por [Zitzler, 1999] ao propor essa métrica. No caso da otimização biobjetivo essa métrica é calculada pela distância euclidiana entre os pontos mais extremos da Fronteira de Pareto, no espaço de objetivos.

***Spacing* (SP)**: Proposta por [Schott, 1995], essa métrica calcula o desvio padrão da distância de Manhattan entre as soluções vizinhas que compõem a Fronteira de Pareto. Se o valor de SP for igual a zero significa que todas as soluções estão espaçadas de modo equidistante na Fronteira de Pareto. Isso é o que se deseja em um bom algoritmo de otimização.

***Amplitude* (AM)**: Essa métrica foi proposta pelo autor. Sua concepção vem da necessidade de criar uma boa referência para a Fronteira de Pareto quando ela ainda não existe na literatura. Essa métrica usa como base os valores obtidos para HV, MS e SP de um problema que se deseja otimizar. O valor da métrica AM é dado pela expressão $AM = max_{HV} + max_{MS} - min_{SP}$.

6. Experimentos com Redes Aleatórias Planares

Foram realizados experimentos com 5 grafos distintos. As características completas de cada um deles são mostradas na Tabela 2, onde o título de cada coluna é o nome que identifica a instância.

Para cada uma das rotas e para cada um dos trechos foi estabelecida uma velocidade média de tráfego. Essa velocidade varia entre 30 e 65 km/h. Outro dado importante é o comprimento de

Tabela 2: Características das instâncias utilizadas.

Parâmetros	49v57a	94v112a	184v222a	274v332a	364v442a
Topologia	Planar	Planar	Planar	Planar	Planar
Vértices	49	94	184	274	364
Arestas	57	112	222	332	442
x	1	2	3	4	5
Trens	10	19	37	55	73
Trechos	66	138	472	650	976
Conf	8	23	152	201	319

cada trecho. Nas redes testadas ele varia entre 5 e 10 km. Nos modelos criados tanto a velocidade quanto o comprimento dos trechos foram obtidos segundo uma distribuição aleatória.

Os testes foram realizados em duas etapas. Na primeira delas foi fornecida somente uma rota por trem, fazendo com que o problema se resumisse ao planejamento de trens. Na segunda etapa o foco do experimento foi em avaliar o roteamento. Foi fornecida tanto uma quanto três rotas por trem. Essa condição permite avaliar rotas com custo um pouco maior, mas que possivelmente acarretam na geração menos conflitos. Além disso, foi testada a métrica *betweenness* tanto em grafos ponderados quanto com pesos unitários em seus vértices.

Ao longo dos experimentos foram calculados os tempos totais de viagem dos trens (TVI, TVV e TVF), os tempos totais de parada (TPI, TPV e TPF), o número total de conflitos entre os trens (CFI, CFV e CFF), o valor das métricas para a melhor Fronteira de Pareto (HV, MS, SP e AM), o tempo limite para encontrar a solução ótima ($TempoLimite$) e o tempo total de processamento ($Temp$). Nas variáveis como TVI, TVV e TVF, por exemplo, a terceira letra denota se se trata da solução inicial, solução dada pelo MO-SVNS ou solução final após a redução dos tempos de parada, respectivamente. Quando citado no texto, o prefixo $DesvPad$ diz respeito ao desvio padrão, $media$ se refere à média aritmética e $CoefVar$ se refere ao coeficiente de variação que é calculado pela relação $s/\bar{x}$, onde s é o desvio padrão e $\bar{x}$ é a média.

Na Tabela 3 são mostrados os resultados para a etapa onde o planejamento é avaliado. Se for analisada a instância *184v222a*, onde há 472 trechos e 152 conflitos e for considerado que para eliminar cada um desses conflitos bastaria adicionar um novo trecho, ao final seriam adicionados 152 novos trechos. Essa adição corresponde a um aumento de 32% em relação ao número de trechos da rede. Alternativamente, conforme proposto pelo algoritmo utilizado neste trabalho, com um aumento médio de 87% do tempo de viagem dos trens, relação entre TVF e TVI, foi possível eliminar todos os conflitos da programação dos trens. Para a próxima análise será tomada como referência a instância *274v332a*. Na relação entre TVF/TVI é possível verificar o quanto foi necessário expandir o tempo de viagem para que todos os conflitos entre as rotas pudessem ser suprimidos. Foi necessário um aumento de 69% do tempo de viagem em relação ao melhor tempo possível, TVI. E, para verificar a eficiência do algoritmo de redução dos tempos de viagem, foi avaliada a relação TVF/TVV. Pode ser verificado que ocorreu uma redução de 26% de TVF em relação a TVV. Além dessa redução, foi observado que TPF/TPV revela uma redução de 46% de TPF quando comparado a TPV. Análises similares podem ser feitas para as outras instâncias.

A Figura 3 mostra dois casos em que o coeficiente de variação atingiu valores elevados para a métrica *Spacing*. Analisando a Figura 3a, instância $94v112a$ cujo $CoefVarSP = 25\%$, é possível verificar que há uma concentração maior de soluções entre 3 e 9 conflitos. Fora dessa região as soluções da Fronteira de Pareto encontram-se bastante espalhadas. Desse modo, fica claro o porquê do valor de $CoefVarSP$ ter sido elevado. A mesma análise pode ser feita para as outras instâncias.

Como pode ser observado na Figura 4, o tempo de processamento aumenta razoavelmente

https://proceedings.science/p/106710?lang=pt-br

Tabela 3: Parâmetros calculados para cada topologia.

Grupo	Parâmetro	49v57a	94v112a	184v222a	274v332a	364v442a
Tempo	Temp (h)	0,000	0,013	9,329	20,073	156,916
	TVI (h)	9,296	22,531	72,027	101,602	149,927
	TVV (h)	10,436	32,112	196,175	230,960	528,051
	TVF (h)	9,725	27,641	140,934	171,518	307,150
	TPV (h)	1,133	9,581	124,148	129,358	338,609
	TPF (h)	0,429	5,110	68,907	69,922	157,655
Desvio Padrão	DesvPadTemp (h)	0,000	0,006	3,464	3,749	25,828
	DesvPadTVV (h)	0,435	2,324	31,789	25,195	28,659
	DesvPadTVF (h)	0,341	0,551	10,707	6,747	10,452
	DesvPadTPI (h)	0,000	0,000	0,000	0,000	0,000
	DesvPadTPV (h)	0,446	2,324	31,789	25,195	53,972
	DesvPadTPF (h)	0,341	0,551	10,707	6,745	3,650
Métrica Pareto	MediaAM	16,405	49,027	521,750	591,943	1074,800
	MediaHV	9,247	28,004	373,216	401,272	763,379
	MediaSP	8,038	2,205	5,210	11,495	9,220
	MediaMS	8,038	23,228	153,745	202,166	320,681
	DesvPadAM	0,382	1,464	33,626	16,086	22,177
	DesvPadHV	0,184	1,459	33,553	15,262	21,324
	DesvPadSP	0,288	0,551	0,509	1,284	1,576
	DesvPadMS	0,026	0,056	0,218	0,072	0,164
	CoefVarAM	0,023	0,030	0,064	0,027	0,021
	CoefVarHV	0,020	0,052	0,090	0,038	0,028
	CoefVarSP	0,036	0,250	0,098	0,112	0,171
	CoefVarMS	0,003	0,002	0,001	0,000	0,001

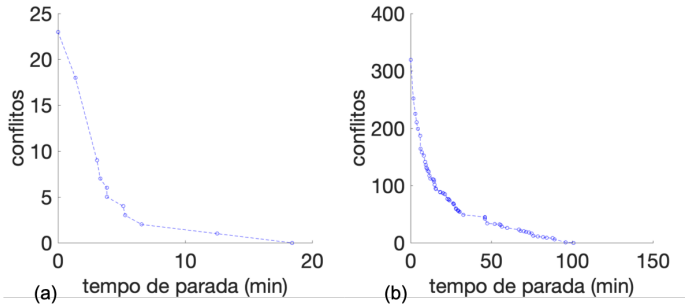


Figura 3: Coeficiente de variação para *Spacing*: (a) alto, 25% (94v112a) e (b) médio, 17% (364v442a).

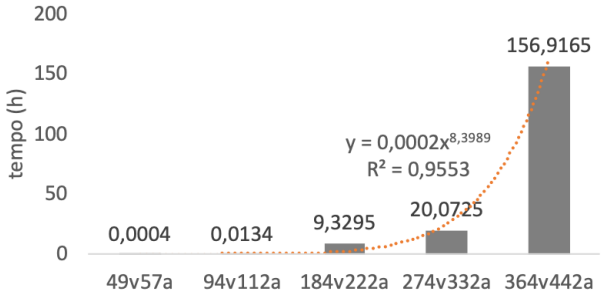


Figura 4: Tempo de processamento do MO-SVNS.

para a instância *364v442a*. Em alguns experimentos esse resultado chegou a 177 horas. Após algumas otimizações do algoritmo o tempo conseguido foi de 157 horas, isto é, foi obtida uma melhoria da ordem de 11%. Ainda assim, trata-se de um tempo muito elevado. Esse é um sinal

claro de que outras técnicas de otimização precisam ser combinadas na busca de resultados em um tempo computacional reduzido. O valor de x para as curvas de tendência pode ser encontrado na Tabela 2 para cada uma das instâncias.

Na Figura 5 é mostrado o resultado para o planejamento de rotas da instância $364v442a$. A solução inicial é mostrada na Figura 5a, onde nenhum tempo de parada foi inserido inicialmente e o número de conflitos é igual a 319. Na próxima figura, 5b, é mostrada a solução obtida pelo algoritmo MO-SVNS. Nesse caso todos os conflitos foram suprimidos, embora TVV tenha aumentado significativamente em relação a TVI. Esse efeito pode ser observado quando se compara o último vértice das Figuras 5a e b. Esses vértices foram destacados nas figuras para facilitar a visualização. Por outro lado, na Figura 5c, é notável a ação do algoritmo de redução dos tempos de parada dos trens quando se compara com a Figura 5b.

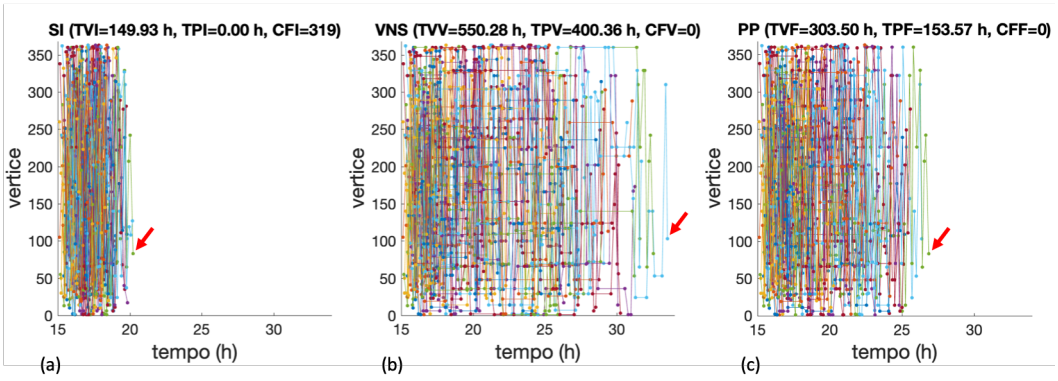


Figura 5: Supressão de conflitos para a instância *364v442a*: (a) solução inicial, (b) solução pelo MO-SVNS e (c) redução de tempos.

Além dos testes descritos anteriormente, foram elaborados outros onde tanto o planejamento quanto o roteamento dos trens pudessem ser analisados em mais detalhes. Foram elaborados 12 cenários. Todos os testes foram realizados tendo como base a instância *94v112a* e o algoritmo foi executado 5 vezes para cada um dos cenários. Nas Tabelas 4 e 5 são mostrados todos os detalhes para esses experimentos.

Tabela 4: Parâmetros calculados para os cenários 1 a 8.

Grupo	Parâmetro	Cen–1	Cen–2	Cen–3	Cen–4	Cen–5	Cen–6	Cen–7	Cen–8
Entrada	Rotas	1	1	3	3	1	1	3	3
	Pond/Btw	Pond	Pond	Pond	Pond	Btw	Btw	Btw	Btw
	Pond/Unit	Pond	Unit	Pond	Unit	Pond	Unit	Pond	Unit
	Trechos	138	137	149	152	171	156	174	169
	Conf	23	23	21	15	26	24	18	18
Tempo	Temp (h)	31,405	29,908	23,671	27,162	45,787	20,339	43,307	23,502
	TVI (h)	22,530	22,250	24,790	25,010	28,820	25,940	29,450	28,270
	TVF (h)	26,972	27,070	30,426	29,920	34,264	28,434	34,782	31,088
	TPF (h)	4,440	4,814	5,636	4,908	5,442	2,498	5,214	2,810
Desvio Padrão	DesvPadTemp (h)	11,777	12,584	10,470	9,409	18,325	5,772	26,681	20,776
	DesvPadTVF (h)	1,492	0,726	0,545	1,316	1,481	0,706	0,949	0,885
	DesvPadTPF (h)	1,495	0,721	0,545	1,315	1,481	0,704	0,748	0,884

Quando se compara os cenários *Cen–1* e *Cen–2* pode ser verificado que o uso de grafos com apenas uma rota e peso unitário leva a soluções mais rápidas, embora a qualidade da solução não seja tão boa em relação a TPF. O mesmo ocorre quando se compara os cenários *Cen–1* e *Cen-*

https://proceedings.science/p/106710?lang=pt-br

Tabela 5: Parâmetros calculados para os cenários 9 a 12.

Grupo	Parâmetro	Cen–9	Cen–10	Cen–11	Cen–12
Entrada	Rotas	3	3	3	3
	Pond/Btw	Pond	Pond	Btw	Btw
	Pond/Unit	Pond	Pond	Pond	Pond
	TempoLimite (s)	10	10	10	10
	Manter Trens	Sim	Não	Sim	Não
	Trens Início	19	19	19	19
	Trens Final	19	16	19	13
	Conf	21	21	18	18
Tempo	Temp (h)	12,571	12,377	13,345	11,886
	TVI (h)	24,790	19,936	29,450	21,474
	TVF (h)	27,270	21,894	32,502	23,038
	TPF (h)	2,480	1,956	3,056	1,566
Desvio Padrão	DesvPadTemp (h)	0,465	0,785	0,690	0,904
	DesvPadTVF (h)	1,042	2,796	0,671	2,406
	DesvPadTPF (h)	1,042	0,676	0,668	0,703

3, em que ambos grafos são ponderados. Quando se utiliza três opções de rota por trem o resultado é calculado mais rapidamente, embora a qualidade da solução piore para as variáveis TVF e TPF. E, por fim, quando se compara os cenários *Cen–3* e *Cen–4*, pode ser verificado que quando se utiliza três opções de rota e peso unitário o tempo de processamento aumenta embora o valor obtido para TPF sofra uma redução.

De modo geral foi observado que independente do uso da métrica *betweenness* no processo de roteamento ou mesmo do número de rotas por trem, na maior parte dos experimentos os resultados obtidos foram melhores quando foram utilizados grafos com pesos unitários. E quando se utiliza três rotas por trem e pesos unitários nos vértices, os resultados referentes ao TPF melhoram significativamente quando se utiliza a métrica *betweenness*.

Na Figura 6 são comparados os cenários *Cen–1*, *Cen–2*, *Cen–5* e *Cen–6*. Em todos os casos foi utilizada somente uma única rota por trem. É notável que no caso da Figura 6d, onde são utilizados pesos unitários nos vértices e a métrica *betweenness*, o processo de otimização obteve resultados muito melhores do que nos outros três cenários comparados.

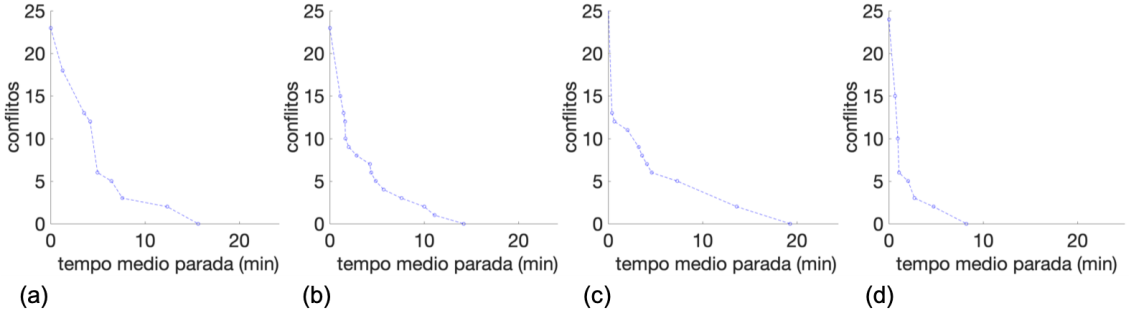


Figura 6: Resultados dos testes de roteamento: (a) *Cen–1*, (b) *Cen–2*, (c) *Cen–5* e (d) *Cen–6*.

Quando se compara os cenários *Cen–9* e *Cen–11* é verificado que além do tempo até a solução ser maior com o uso da métrica *betweenness*, os tempos TVF e TPF sofrem um aumento significativo, piorando a solução em relação a *Cen–9*. E, na comparação dos cenários *Cen–10* e *Cen–12*, no cenário *Cen–12* o valor de TVF é apenas 5% maior e o valor de TPF é 20% menor.

7. Conclusões e Trabalhos Futuros

Nesta pesquisa foi implementado um conjunto de algoritmos capazes de abstrair detalhes mais específicos de redes ferroviárias para que redes aleatórias pudessem ser utilizadas. Desse modo, a solução do problema se tornou mais genérica sendo possível reutilizar as técnicas descritas nesta pesquisa para resolver problemas de outras naturezas. O MO-SVNS mostrou-se bastante adequado na solução do problema de otimização multiobjetivo envolvendo o planejamento e o roteamento de trens. E no que tange à Fronteira de Pareto, aqui cabe um destaque especial à métrica *amplitude*, que foi proposta pelo autor para ajudar na escolha da melhor Fronteira de Pareto em problemas que não tenham correspondência exata na literatura, como é o caso deste estudo.

Quanto aos resultados obtidos para as instâncias avaliadas, eles são considerados muito bons diante do desafio da otimização biobjetivo que visa a minimização dos conflitos de rotas e a minimização dos tempos médios de parada dos trens. É válido lembrar que um dos passos da metodologia proposta previa a geração de situações com muitos conflitos para que o MO-SVNS pudesse trabalhar em cenários bastante complexos. Contudo, em redes reais é comum encontrar regras de espaçamento de entidades no espaço ou no tempo, o que pode tornar o processo de otimização bem mais rápido nessas redes. Além disso, um ponto alto da solução do problema foi a implementação do algoritmo para reduzir os tempos de viagem.

Como trabalhos futuros são sugeridas algumas linhas de pesquisa que se propõem a aumentar o alcance dos resultados obtidos neste trabalho, a saber: (a) realizar estudos comparativos com outras metaheurísticas, (b) utilizar técnicas de aprendizado de máquina para melhorar os resultados, (c) implementar o problema com base no *Job Shop Scheduling*, (d) realizar o planejamento dos trens com base na rota de um trem prioritário, (e) avaliar a ocorrência de interferências ou perda de conexões de rotas, (f) comparar resultados obtidos por programação linear e por outras metaheurísticas com aqueles obtidos nesta pesquisa.

Referências

Bagchi, T. P. (1999). *Multiobjective scheduling by genetic algorithms*. Springer Science & Business Media.

Benjamin, A., Chartrand, G., e Zhang, P. (2015). *The fascinating world of graph theory*. Princeton University Press.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

Brasil (2018). *Projeto Custos Ferroviários: Relatório Final Objeto 1*. Agência Nacional de Transportes Terrestres (ANTT), Brasılia, DF.

Cai, X., Goh, C., e Mees, A. I. (1998). Greedy heuristics for rapid scheduling of trains on a single track. *IIE transactions*, 30(5):481–493.

Costa, H. D., Paiva, M. H. M., e Rocha, H. R. d. O. (2018). Supressão de conflitos no problema de planejamento e roteamento de trens utilizando a metaheurística VNS. In *Anais do* 50º *Simposio Brasileiro de Pesquisa Operacional*, Rio de Janeiro. SOBRAPO.

Cousineau, M., Perron, S., Caporossi, G., Paiva, M., e Segatto, M. (2015). Rwa problem with geodesics in realistic otn topologies. *Optical Switching and Networking*, 15:18–28.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, p. 35–41.

Grover, W. D. e Doucette, J. (2001). A novel heuristic for topology planning and evolution of optical mesh networks. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, volume 4, p. 2169–2173. IEEE.

Hansen, P. e Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467.

Höller, H. e Voß, S. (2006). A heuristic approach for combined equipment-planning and routing in multi-layer sdh/wdm networks. *European Journal of Operational Research*, 171(3):787–796.

Ingolotti, L., Barber, F., Tormos, P., Lova, A., Salido, M. A., e Abril, M. (2005). A scheduling order-based method to solve timetabling problems. In *Conference of the Spanish Association for Artificial Intelligence*, p. 52–61. Springer.

Landex, A., Kaas, A. H., Schittenhelm, B., e Schneider-Tilli, J. (2006). Practical use of the uic 406 capacity leaflet by including timetable tools in the investigations. *WIT Transactions on The Built Environment*, 88.

Pinedo, M. (2005). *Planning and scheduling in manufacturing and services*. Springer.

Rubio-Largo, A., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., e Sánchez-Pérez, J. M. (2010). Solving the routing and wavelength assignment problem in wdm networks by using a multiobjective variable neighborhood search algorithm. In *Soft computing models in industrial and environmental applications, 5th international workshop (SOCO 2010)*, p. 47–54. Springer.

Şahin, İ. (1999). Railway traffic control and train scheduling based oninter-train conflict management. *Transportation Research Part B: Methodological*, 33(7):511–534.

Samà, M., Corman, F., Pacciarelli, D., et al. (2017). A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. *Computers & Operations Research*, 78:480–499.

Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH.

Szpigel, B. (1973). Optimal train scheduling on a single line railway.

Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *Management Science*, 17(11): 712–716.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Citeseer.

Zitzler, E. e Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, p. 292–301. Springer.

https://proceedings.science/p/106710?lang=pt-br