

PLANEJAMENTO DE MANUTENÇÃO DE ATIVOS FERROVIÁRIOS VIA SOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE MULTI OBJETIVO*

Breno Carvalho de Oliveira¹
Clayton Guimarães de Almeida²
Jesus Jonatan Souza Santos³
Eduardo Gontijo Carrano⁴
Gustavo Barros Castro⁵
Breno Delgado Silva⁶

Resumo

Neste trabalho é apresentado uma ferramenta de otimização multiobjetivo para planejamento da manutenção de ativos ferroviários. A aplicação foi modelada com base no problema do Caixeiro Viajante Multiobjetivo, considerando como objetivos a minimização da postergação da manutenção dos equipamentos e do deslocamento da equipe de manutenção. Foi utilizada uma instância real de 17 cidades que são atendidas por uma única equipe de manutenção. A ferramenta de otimização consiste na combinação do algoritmo SPEA2 (*Strength Pareto Evolutionary Algorithm*), da methaheurística GRASP (*Greedy Randomized Adaptive e Search Procedure*) e de uma heurística de busca local em vizinhança, cada uma com seu papel específico dentro da otimização. O algoritmo foi capaz de mapear soluções melhores que a atualmente adotada pela empresa, proporcionando uma economia significativa no valor total a ser gasto em um ciclo de manutenção dos ativos ferroviários.

Palavras-chave: Otimização Multiobjetivo; Caixeiro Viajante; Planejamento de Manutenção; Ativos Ferroviários.

MAINTENANCE PLANNING OF RAILWAY ASSETS VIA SOLUTION OF THE MULTIOBJECTIVE TRAVELING TRADER PROBLEM

Abstract

This manuscript presents a multiobjective optimization tool for planning maintenance of railway assets. The application was modeled based on the Multiobjective Traveling Salesman problem, considering the equipment maintenance delaying and the maintenance team displacement as objectives to be minimized. A real instance of 17 cities that are attended by the same maintenance team was considered for evaluating the algorithm. The optimization tool is a combination of SPEA2, GRASP, and a Neighborhood Local Search procedure, each one with its role in the optimization process. The algorithm was able to map solutions that are better than the one currently employed by the company, leading to a reduction of the budget spent in a railway asset maintenance cycle.

Keywords: Multiobjective optimization; Traveling salesman problem; Maintenance planning; Railway assets.

¹ Engenheiro eletricitista, Analista de processos de engenharia, Gerência de Engenharia de material rodante, VLI, Divinópolis, Minas Gerais, Brasil

² Bacharel em Sistemas de informação, gerência de tecnologia da informação, VLI, Divinópolis, MG, Brasil.

³ Engenheiro eletricitista, Supervisor de Engenharia de vagões, maquinas de via e materiais, Gerência de Engenharia de material rodante, VLI, Divinópolis, Minas Gerais, Brasil.

⁴ Doutor em Engenharia Elétrica, Professor do Departamento de Engenharia Elétrica da Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Minas Gerais, Brasil

⁵ Engenheiro mecatrônico, Analista de processos de engenharia, Gerência de Engenharia de material rodante, VLI, Divinópolis, Minas Gerais, Brasil.

⁶ Engenheiro de produção, Analista de processos de engenharia, Gerência de Engenharia de material rodante, VLI, Divinópolis, Minas Gerais, Brasil.

1 INTRODUÇÃO

A manutenção se mostra hoje como parte essencial da cadeia produtiva, requerendo a utilização de processos internos cada vez mais hábeis e efetivos. Esta nova concepção vem conduzindo a manutenção a assumir uma nova postura estratégica, não se atendo apenas a reparar equipamentos, mas também definindo planejamentos estratégicos que visam reduzir a probabilidade de uma parada de produção não planejada e custos adicionais para a empresa [1].

Baseados nestes conceitos, e diante de um cenário econômico no mínimo desafiador, a empresa sob estudo foi impulsionada a adotar uma postura diferenciada em relação aos ciclos de planejamento da manutenção dos seus ativos ferroviários, uma vez que cada frota de ativo é atendida por uma equipe exclusiva de manutenção. O planejamento deste atendimento é realizado com base no conhecimento e experiência de um funcionário, que pondera a melhor rota de atendimento que a equipe de manutenção irá realizar, diante de uma análise manual entre a relação de criticidade em que o ativo deve ser atendido e no deslocamento que a equipe de manutenção irá realizar.

Visto que no início de cada ciclo de manutenção todos os ativos estão mapeados e suas criticidades estão estabelecidas, o funcionário define qual a rota a equipe de manutenção irá realizar, levando em consideração as restrições de jornada de trabalho e limites diários da equipe, bem como buscando uma relação de custo adequado ao plano orçamentário da companhia.

O presente trabalho propõe o desenvolvimento de uma ferramenta de otimização multiobjetivo que combina o *Strength Pareto Evolutionary Algorithm* (SPEA2)[2] o *Greedy Randomized Adaptive Search Procedure* (GRASP) [3], utilizado para geração da população inicial, um procedimento de Busca Local em Vizinhança, para aprimoramento da convergência, e Programação de Compromisso [4], com finalidade de auxílio à tomada de decisão. Foram considerados como objetivos a minimização do custo total da postergação da manutenção dos equipamentos e a minimização do custo total de deslocamento da equipe de manutenção.

Este artigo possui a seguinte estrutura: a descrição do problema e o algoritmo desenvolvido é descrito na seção 2; os resultados alcançados em um cenário real são discutidos na seção 3; por fim, na seção 4 são delineadas algumas conclusões.

2 MATERIAIS E MÉTODOS

O problema proposto e sua modelagem matemática são discutidos ao longo dessa seção.

2.1 O problema do caixeiro viajante multiobjetivo

O problema do Caixeiro Viajante (PCV) é um dos mais tradicionais e conhecidos problemas de otimização combinatória. Este problema consiste em encontrar a melhor rota para um caixeiro viajante que, partindo de uma cidade origem, deve visitar todas as cidades de uma lista precisamente uma vez e então retornar à cidade de origem. O problema do Caixeiro Viajante Multiobjetivo (PCV MO) é uma

variante multiobjetivo do PCV onde são consideradas simultaneamente m funções objetivo. Como as funções são geralmente conflitantes entre si, este tipo de problema tende a apresentar mais de uma solução ótima [5].

Assim como no PCV, o PCV MO é um problema NP-difícil. Com isso, se faz necessário adotar heurísticas ou metaheurísticas para solução de instâncias de médio e grande porte desse problema. [6]

2.2 Descrição e modelagem do problema

O caso considerado foi baseado em uma empresa real do ramo ferroviário cuja equipe de manutenção atende dezesseis ativos, cada um em uma cidade diferente, partindo de uma cidade base. A equipe, para cada ciclo de manutenção, visita cada uma das cidades apenas uma vez e efetua a manutenção necessária no equipamento.

No início de cada ciclo de manutenção, já se sabe qual tipo de manutenção cada equipamento deverá ser submetido. Existem três tipos de manutenção (executadas por uma equipe de três funcionários), cada qual com suas particularidades:

1. 300h: manutenção de menor complexidade, cujo tempo de execução é de 5 horas de trabalho;
2. 600h: manutenção de maior complexidade, cujo tempo de execução é de 12 horas de trabalho;
3. Corretiva: manutenção que ocorre somente quando um equipamento falha. A duração é de 4 horas de trabalho, em média.

As manutenções preventivas (300h e 600h) são baseadas na quantidade de horas que o motor diesel dos ativos está trabalhando. Deseja-se então definir a rota ótima para execução do ciclo de manutenções em cada um dos ativos, visando minimizar o custo de deslocamento total e também minimizar o custo total de postergação da manutenção dos equipamentos, uma vez que o atraso da execução dos planos de manutenção intensifica a degradação dos ativos.

Para se encontrar o custo total de postergação da manutenção, primeiro deve-se saber qual o valor da postergação por hora e qual o valor do homem/hora parado, para os casos onde o ativo está estragado e precisará de uma corretiva. Por motivos de sigilo de informação, esses valores não serão disponibilizados neste trabalho. Todos os valores apresentados foram expressos de forma normalizada.

Estes valores são definidos por dois parâmetros de entrada: V_p e V_{hhp} . O parâmetro V_p representa o valor da postergação da manutenção por hora e o parâmetro V_{hhp} representa o valor de homem/hora parado por funcionário parado. O valor de V_{hhp} deve ser multiplicado por três, visto que cada ativo possui três funcionários trabalhando no mesmo. Com base nestas informações, é possível chegar ao cálculo do custo de postergação da manutenção para cada ativo, como mostrado na equação 1:

$$CPME_j(\vec{x}) = (Y_j - 1)[Tp_j \cdot V_p \cdot \Delta t_{j(\vec{x})}] + Y_j[3 \cdot V_{hhp} \cdot \Delta t_{j(\vec{x})}] \quad (1)$$

Em que

- $CPME_j$: custo de postergação da manutenção esperado para o ativo no vértice j
- Tp_j : taxa de postergação do tipo de manutenção do ativo no vértice j ;
- V_p : valor do custo de postergação da manutenção (por hora);
- Δt_j : tempo total de atraso de manutenção do ativo localizado no vértice j ;
- Y_j : flag que indica se o tipo de manutenção vinculada ao ativo no vértice j é uma manutenção corretiva (valor 1) ou se é uma manutenção preventiva (valor 0);
- V_{hnp} : valor do homem/hora parado.

Para o cálculo do tempo total de atraso, é preciso somar os tempos de manutenção efetuados até a finalização da manutenção do ativo corrente com o tempo de deslocamento até a cidade em que se encontra o ativo. Além disso, é necessário somar o tempo de atraso da manutenção do ativo no início do processo, conhecido a priori, que corresponde à quantidade de horas que o ativo rodou após o momento indicado para manutenção. Logo, o tempo total de atraso de um ativo \$\$\$ pode ser calculado como apresentado em (2).

$$\Delta t_j(\vec{x}) = \sum_{k \in A(j, \vec{x})} \sum_{l \in \{A(j, \vec{x}) \cup j\}} [(Td_{kl} + Tm_l)x_{kl}] + T_j^0 \quad (2)$$

Em que:

- Δt_j : tempo total de atraso do ativo no vértice j ;
- $A(j, \vec{x})$: conjunto de vértices antecedentes ao vértice j na solução \vec{x} ;
- Td_{ij} : tempo de deslocamento do vértice i até o vértice j ;
- Tm_i : tempo gasto na execução da manutenção do ativo no vértice i ;
- T_j^0 : tempo de atraso da manutenção do vértice no ativo j no início do processo.

O custo de deslocamento entre dois vértices (cidades) é definido como o produto da média de quilômetros por litro pelo custo do combustível, conforme a equação 3. A este custo deve ser adicionado o valor total das diárias, que é o produto do valor médio da diária para a equipe de três pessoas pela quantidade de dias gastos na manutenção de todos os ativos (tempo total de manutenção dividido por 8, que é a quantidade de horas trabalhadas por dia), conforme equação 4:

$$C_{ij} = \left(\frac{KM_{ij}}{K_{ml}} \right) \cdot PC_{omb} \quad (3)$$

$$CTd_{(\vec{x})} = \frac{\sum_{i=1}^n \sum_{j=2}^n (Td_{ij} + Tm_l)x_{ij}}{8} \cdot V_{diaria} \quad (4)$$

Em que:

- C_{ij} : custo de deslocamento do vértice i até o vértice j ;
- KM_{ij} : quilometragem percorrida entre os vértices i e j ;
- K_{ml} : consumo médio do veículo por km percorrido;
- PC_{omb} : preço médio do combustível;
- $CTd_{(\vec{x})}$: custo total das diárias durante a manutenção de todos os ativos;

- V_{diaria} : preço médio da diária para a equipe por dia.

Com as equações 1 a 4 é possível representar as duas funções objetivo do problema, que representam o custo total de deslocamento da equipe de manutenção e o custo total de postergação da manutenção, ambas a serem minimizadas. Estas são apresentadas, respectivamente, nas equações 5 e 6.

$$f1 = \sum_{i=1}^n \sum_{j=1}^n C_{ij} \cdot x_{ij} + CTd(\vec{x}) \quad (5)$$

$$f2 = \sum_{j=1}^n CPME_j(\vec{x}) \quad (6)$$

2.3 Algoritmo

O problema proposto apresenta uma característica semelhante ao PCV MO, no qual é necessário visitar todas as cidades apenas uma vez, partindo da cidade base e retornando à cidade base ao fim do ciclo. A região que esta equipe atende é composta por dezesseis cidades. À cada cidade foi associado um índice numérico, para facilitar a representação, e um nome fictício, para fins de sigilo, como segue: 1 - Cidade Base (base); 2 - Cidade A; 3 - Cidade B; 4 - Cidade C; 5 - Cidade D; 6 - Cidade E; 7 - Cidade F; 8 - Cidade G; 9 - Cidade H; 10 - Cidade I; 11 - Cidade J; 12 - Cidade K; 13 - Cidade L; 14 - Cidade M; 15 - Cidade N; 16 - Cidade O; 17 - Cidade P.

O algoritmo proposto é baseado no algoritmo genético multiobjetivo *Strength Pareto Evolutionary Algorithm 2* (SPEA2) [2]. Os indivíduos são representados por vetores de 18 posições, sendo que a primeira e a última são necessariamente 1 e as 16 restantes representam uma permutação válida dos índices de 2 a 17. Cada posição deste vetor representa uma cidade a ser visitada para a execução da manutenção do ativo e o conteúdo de cada posição representa o código da cidade. O fluxograma que representa os passos do algoritmo pode ser visto na Figura 1.

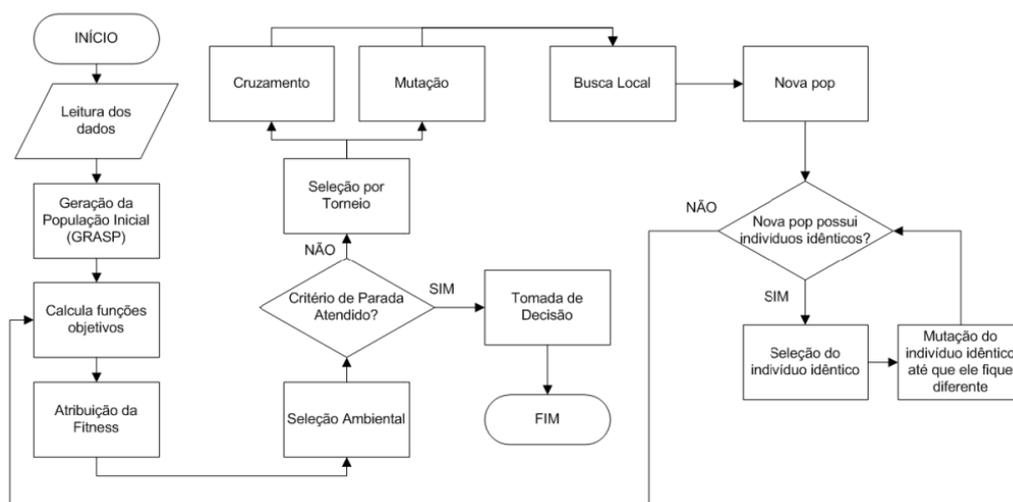


Figura 1. Fluxograma do Algoritmo SPEA2 desenvolvido

A população inicial é gerada utilizando o Greedy Randomized Adaptive e Search Procedure (GRASP)[3]. A geração de cada indivíduo é feita por meio de três etapas:

1. **Avaliação dos candidatos:** A avaliação dos candidatos foi implementada de forma que, para cada cidade a ser inserida no indivíduo, são calculadas as duas funções objetivo para todas as cidades ainda disponíveis para seleção. Após isso, é feito o somatório destes valores e a lista de candidatos é ordenada do menor valor (G_{min}) para o maior valor (G_{max}) considerando a influência deles na qualidade da solução.
2. **Geração da lista restrita de candidatos (RCL):** processo de criação da RCL é feito por meio da seleção de uma parte da lista de candidatos previamente montada. Os candidatos são selecionados buscando aqueles que possuírem os menores valores para sua função $F(x)$ previamente calculada.
3. **Escolha aleatória de um candidato:** com a lista restrita de candidatos já criada e com todos os candidatos já possuindo o seu valor de função $F_{grasp}(\vec{x})$ calculado, faz-se um sorteio para selecionar qual cidade assumirá a próxima posição na criação do indivíduo. Esta aleatoriedade, juntamente com o vetor de pesos, permite que sejam gerados indivíduos distintos.

Estas três etapas são repetidas até que todo o indivíduo esteja preenchido. Por fim, para garantir a diversidade da população gerada, foi utilizada a técnica de algoritmos celulares apresentada em [7]. O cruzamento foi feito utilizando o operador Cruzamento por Mapeamento Parcial (PMX - *Partially-Mapped Crossover*) Para mutação foi utilizado o operador Mutação por Troca (*Exchange Mutation* ou *Swap Mutation*) [8].

Com o intuito de melhorar a qualidade da nova população e acelerar a convergência dos indivíduos na direção da fronteira Pareto, foi aplicado um operador de busca local dentro do SPEA2. A busca local é aplicada a uma parte da população atual, em todas as gerações. Na busca local, o indivíduo é perturbado selecionando-se aleatoriamente um gene a ser alterado. A modificação deste é feita por meio de uma busca em uma lista de melhores vizinhos, que contém as cidades que são visitadas após o gene escolhido, em ordem crescente de distância da cidade atual. Parte dessas cidades são retornadas e, para cada uma delas, é feita a troca da mesma pela cidade atual, fazendo com que o indivíduo não se torne ineficaz. Na sequência é feita a verificação se o indivíduo diminuiu o valor das funções objetivo. Caso o novo indivíduo possua melhores valores de funções objetivo, o mesmo é armazenado. Caso contrário, a alteração é desfeita e a próxima cidade da lista é utilizada.

Após a busca local é verificado se algum dos indivíduos gerados é igual aos indivíduos do arquivo temporário. Caso seja obtido algum indivíduo já existente, o mesmo é submetido à mutação, até que ele se torne um indivíduo único.

Em seguida é criada a nova população, onde o algoritmo retorna para a atividade de cálculo das funções objetivo, iniciando um novo ciclo do SPEA2. Todos estes passos serão repetidos até que a condição de parada seja satisfeita, o que significa que o arquivo temporário estará com a melhor aproximação da fronteira Pareto encontrada.

Ao se trabalhar com uma aproximação de fronteira Pareto, tem-se o problema de possuir vários resultados que representam um conjunto de soluções ótimas (relação trade-off), dificultando a tomada de decisão para escolha do melhor indivíduo. Para tanto, foi utilizada a técnica de programação de compromisso que auxilia na escolha da solução que melhor se ajusta às preferências dos decisores. [4]

Mediante a postura conservadora da empresa estudada em relação aos seus ativos, presou-se o equilíbrio entre a confiabilidade dos equipamentos e os valores gastos nas manutenções dos mesmos. Sendo assim, assumiu-se que as duas funções objetivo terão o mesmo peso.

Com estas métricas definidas, foi utilizada a equação 7 para a escolha do melhor indivíduo via programação de compromisso.

$$\min \left[\sum_{i=1}^m w \cdot (P_i - P_j^*)^2 \right]^{\frac{1}{2}} \quad (7)$$

3 RESULTADOS E DISCUSSÃO

Visando encontrar melhores resultados, foi feita a calibração manual dos parâmetros do sistema mediante várias execuções do algoritmo. Após esse processo de calibração foram encontrados os parâmetros apresentados na Tabela 1

Tabela 1. Parâmetros de entrada do algoritmo

Parâmetro	Descrição	Valor
TotalIndividuosPopulacao	Indica, quantos indivíduos serão necessários para compor a população	100,00
TotalGeracoes	Define, quantas vezes o algoritmo deverá repetir sua execução, evoluindo a população, a cada execução	70,00
Alfa	Fator, multiplicador do GRASP	0,50
TaxaPerturbacaoDoArquivo	Percentual de indivíduos a perturbar na busca local	0,75
TaxaPerturbacaoDoIndividuo	Porcentagem que indica quantos genes do indivíduo devem ser perturbados na busca local	0,40
QuantidadePerturbacoes	Número de vezes que serão feitas perturbações em cada indivíduo durante a busca local	10,00
TaxaVizinhancaBuscaLocal	Percentual de melhores vizinhos a serem utilizados	0,35
TaxaCruzamentoPopulacao	Indica a porcentagem da população a sofrer cruzamento	0,60
TaxaMutacaoIndividuo	Indica a porcentagem de genes do indivíduo a serem alterados de forma aleatória	0,40
WCustoDeslocamento	Peso dado a função objetivo de cálculo do total do custo de deslocamento, utilizado na programação de compromisso*	0,50
WCustoManutencao	Peso dado a função objetivo de cálculo do total do custo de manutenção, utilizado na programação de compromisso*	0,50

Foram efetuadas cinco execuções do algoritmo, tendo sido encontrados resultados ligeiramente distintos, conforme sumarizado na Tabela 2. Mesmo os resultados não sendo idênticos, é possível notar que os mesmos são semelhantes, tendo poucas variações com relação ao número de indivíduos não dominados encontrados, o tempo total de execução e o custo total do ciclo de manutenções.

Tabela 2. Resultados das Simulações

Execução	No Indivíduos Não Dominados	Melhor Indivíduo Progr. de Comp. [f1, f2]	Custo Total $f(x) = f1 + f2$	Tempo Execução (seg.)
1	17	[0.84877, 0.94250]	1.79127	614
2	17	[0.84209, 0.95407]	1.79616	613
3	18	[0.83785, 0.96738]	1.80523	618
4	14	[0.84791, 0.94811]	1.79602	617
5	19	[0.84877, 0.94250]	1.79127	617

A aproximação de fronteira Pareto obtida na execução 5 é apresentada na Figura 2. Esta execução foi escolhida por ter apresentado o maior número de indivíduos não dominados e também o menor valor do custo total de manutenção na solução indicada pela solução de compromisso.

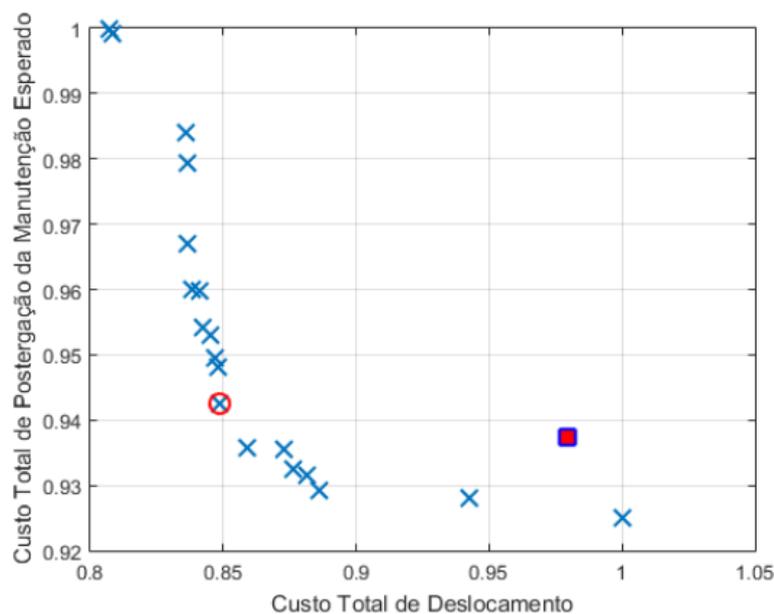


Figura 2. Gráfico da melhor aproximação da fronteira Pareto

A rota do indivíduo mais adequado obtido na execução 5 é apresentada na figura 3. Pode ser observado que, mesmo havendo cidades mais próximas a serem visitadas, o algoritmo optou por visitar algumas cidade mais distante. Isso ocorre devido ao fato de que o tipo de manutenção e o tempo inicial (T_0) de cada ativo podem gerar um aumento considerável do custo total de postergação da manutenção, caso o mesmo não seja priorizado. Pode ser observado também que as primeiras cidades a serem visitadas foram aquelas que possuem ativos com necessidade de manutenção corretiva devido ao fato de que, quanto mais os ativos com corretiva ficarem parados, maior será o valor gasto com a ociosidade dos funcionários que trabalham diretamente com o auto de linha. Logo, é possível notar que o tempo de postergação de uma manutenção corretiva é mais custoso que a postergação das manutenções de 300 horas e 600 horas

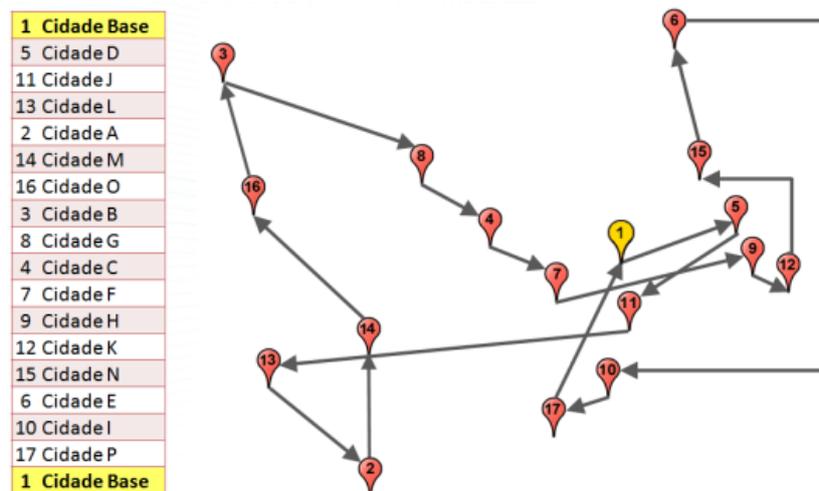


Figura 3. Mapa do melhor resultado

A solução escolhida foi comparada com a solução atual, gerada pelo funcionário que é o atual responsável por montar a rota de atendimento da equipe de manutenção da empresa. A rota montada pelo mesmo refere-se exatamente ao ciclo de manutenção utilizado neste trabalho, o qual está marcado com um quadrado vermelho de bordas azuis na Figura 2. O comparativo de valores entre o indivíduo escolhido e o indivíduo real gerado pelo funcionário pode ser observado na tabela 3

Tabela 3. Programação de compromisso x programador real

Origem	SPEA2 + Prog. Comp.	Origem	Resultado Real Prog.
Valor f1	0,84877	Valor f1	0,97924
Valor f2	0,94250	Valor f2	0,93743
Valor Total	1,79127	Valor Total	1,91667

Ao analisar os dois resultados, pode ser visto que o indivíduo encontrado pelo SPEA2 possui um valor total menor que a solução atual adotada, sendo observada uma economia total de aproximadamente 6,4%. Outro ponto importante que pode ser visto é que esta economia não foi feita em ambas as funções objetivo. Foi permitido que a função objetivo f_2 referente ao custo total de postergação da manutenção sofresse um pequeno acréscimo de 0,54%, podendo assim, minimizar o custo da função objetivo f_1 (custo total de deslocamento) em 13,32%.

Portanto, a programação de compromisso foi capaz de identificar um indivíduo que gerou uma economia no custo total de tal forma a não impactar negativamente a confiabilidade dos ativos

4 CONCLUSÃO

O objetivo deste trabalho foi minimizar o custo total de deslocamento e minimizar o custo total de postergação da manutenção de dezesseis equipamentos, cada um com suas características específicas, tipos de manutenção e localização.

Após a escolha do indivíduo mais adequado aos interesses da empresa, foi efetuada a comparação do mesmo com a solução atual adotada pela empresa. Foi visto que a solução obtida melhora o custo total de manutenção em mais de 6\%.

Com isso, pode se constatar que, mesmo o especialista possuindo grande conhecimento prático, este não conseguiu obter um menor custo total para o problema. Já o algoritmo foi capaz de mapear uma solução que, mesmo permitindo um pequeno aumento no valor do custo total de postergação da manutenção, conseguiu diminuir, em uma proporção bem maior, o custo total do deslocamento. Isso não impacta na confiabilidade dos ativos, pois o aumento de postergação foi praticamente irrelevante. Por outro lado, a economia efetuada é significativa, visto que o cenário aborda somente um ciclo de manutenção.

REFERÊNCIAS

- 1 Pinto AK, Xavier JN. Manutenção: função estratégica. 2ª Ed. Rio de Janeiro: Qualitymark; 2001.
- 2 Zietler E, Laumanns M, Thieles L. *SPEA2: Improving the strength pareto evolutionary algorithm*. Swiss Federal Institute of Technology (ETH) Zurich. 2001.
- 3 Santiago PHR. Método GRASP e ACO em Otimização. Dissertação de mestrado, UNICAMP, Campinas, São Paulo; 2015.
- 4 Bollmann HA. O uso da programação por compromisso para a estruturação de um indicador de sustentabilidade. III Encontro da ANPPAS. Brasília, Distrito Federal; 2006.
- 5 Paquete L, Chiarandini M, Stuzle T. *Evolutionary Multi-Criterion Optimization - {A Two-Phase Local Search for the Biobjective Traveling Salesman Problem*. Berlim: Springer; 2003.
- 6 Fariais MSR, Goldberg MC, Luna HPL. Aplicação de um Algoritmo Evolucionário de Busca de Soluções de Pareto para o Problema do Caixeiro Viajante Biobjetivo. Anais do XL SBPO; 2008.
- 7 Murata T, Ishibuchi H, Gen M. *Specification of Genetic Search Directions in Cellular Multiobjective Genetic Algorithms. Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization. 2001: p. 82-95.*
- 8 Malaquias NG. Usi de Algoritimos Genéticos para a Otimização de Rotas de Distribuição. Dissertação de mestrado, Universidade Federal de Uberlândia, Uberlândia, Minas Gerais; 2006