

Busca Local Iterativa aplicada ao Problema de Programação de Técnicos de Campo

Carlos Eduardo de Moraes Ferreira^{1,2}, Evelyn Michelle Henrique Braga^{1,2}, Luiz Henrique dos Santos Fernandes^{1,2}, Ramon Amorim dos Santos Delgado Ama¹, Sergio Rebouças², Antônio Augusto Chaves¹

¹Instituto de Ciência e Tecnologia - Universidade Federal de São Paulo
Av. Cesare Mansueto Giulio Lattes, nº 1.201, São José dos Campos/SP

²Instituto Tecnológico de Aeronáutica

Praça Marechal Eduardo Gomes, nº 50, São José dos Campos/SP

cem.ferreira@unifesp.br, evelynbraga@outlook.com, lhsf@ita.br,
ramon.amorim@unifesp.br, reb@ita.br, antonio.chaves@unifesp.br

RESUMO

O presente artigo propõe a aplicação de Busca Local Iterativa (*Iterated Local Search*) ao Problema de Programação de Técnicos de Campo. Este problema trata da alocação de equipes com diferentes capacidades para a execução de diferentes tarefas com custos e janelas de tempo específicas, em atendimento à demanda de serviços de clientes. A função objetivo maximiza o número de clientes atendidos no menor tempo, levando em consideração a prioridade das tarefas. Por ser um problema cuja solução tem custo computacional exponencial, o que o torna intratável para grandes instâncias, foram desenvolvidas cinco estratégias de perturbação com a reinicialização parcial da solução como busca local, no intuito de melhorar a eficiência da meta-heurística na busca de uma boa solução. O método foi implementado sobre uma base de dados da literatura constituída por diversas instâncias de teste e atingiu, em alguns casos, percentual significativo do resultado ótimo.

PALAVRAS CHAVE. Problema de Programação de Técnicos de Campo. Problema de Roteamento. Busca Local Iterativa.

Tópicos: Meta-heurísticas

ABSTRACT

The present work proposes the application of the Iterated Local Search to the Field Technician Scheduling Problem. This problem deals with the allocation of group of technicians with different skills and working hours in different tasks, with specific costs and time windows, in order to meet the costumers demand. The objective function maximizes the number of clients served in the shortest time, taking into account the priority of the tasks. Because it is a problem whose solution has exponential computational cost, which makes it intractable for large instances, five perturbation strategies were developed with the partial restart of the solution as local search, in order to improve the efficiency of the metaheuristic in the search for good solutions. The method was implemented on a database of the literature consisting of several test instances and reached significant percentage of the optimal result in some cases.

KEYWORDS. Field Technician Scheduling Problem. Routing Problem. Iterated Local Search.

Paper topics: Metaheuristics

1. Introdução

Um dos maiores desafios das grandes companhias prestadoras de serviços, principalmente as de telecomunicações, é o atendimento às demandas dos clientes em termos de serviços de instalação e manutenção de equipamentos e sistemas. Este desafio, conhecido na literatura como Problema de Programação de Técnicos de Campo - FTSP (do inglês, *Field Technician Scheduling Problem*), trata da alocação de técnicos de campo, com diferentes habilidades e jornadas de trabalho, que devem executar determinadas tarefas, com janelas de tempo, custos, tempo de deslocamento e prioridades definidas. O presente artigo propõe o uso da meta-heurística Busca Local Iterativa - ILS (do inglês, *Iterated Local Search*) [Lourenço et al., 2010] como alternativa para a solução do FTSP, tendo como base comparativa os dados apresentados por Damm et al. [2016].

Diversas abordagens deram origem ou tratam do mesmo método, mas em aplicações diferentes, como o problema de orientação de equipes com janelas de tempo – TOPTW (do inglês, *Team Orienteering Problem with Time Windows*) que é uma variante do Problema de Orientação, também chamado de Problema do Caixeiro Viajante Seletivo - STSP (do inglês, *Selective Traveling Salesman Problem*). Estas definições, que datam do início da década de 1970, estão associadas às aplicações dos problemas de patrulhamento policial e buscam a otimização de respostas às solicitações de apoio devido a ocorrências de crimes [Cardoso, 2019]. Fundamentalmente, o problema trata da alocação de recursos de diferentes tipos (policiais de diferentes especialidades) para o atendimento de demandas definidas por: custo e tempo de deslocamento; tempo de execução da tarefa demandada; e priorização das demandas, o que acrescenta ao problema de alocação as questões de roteamento e sequenciamento.

Numerosas variantes foram desenvolvidas ao longo dos últimos anos e, apesar dos autores tratarem por nomes diferentes, abordam as mesmas questões fundamentais, alterando, geralmente, apenas as funções objetivo ou algumas unidades de medida. Cortés et al. [2014] tratam do problema como Roteamento com Janelas de Tempo Flexíveis - VRPSTW (do inglês, *vehicle routing problem with soft time windows*). Tang et al. [2007] apresenta o Problema de Coleta Máxima em rotas múltiplas com bônus relacionado ao tempo - MTMCPTD (do inglês, *Multiple Tour Maximum Collection Problem with Time-Dependent rewards*). Hashimoto et al. [2011] trata do Problema de Escalonamento de Técnicos e Intervenção para Telecomunicações - TIST (do inglês, *Technicians and Interventions Scheduling Problem for Telecommunications*). Cordeau et al. [2010] e Chen et al. [2018] apresentam o Problema de Escalonamento de Técnicos e de Tarefas - TTSP (do inglês, *Technician and Task Scheduling Problem*) e sua variante de multi-período, respectivamente. Diversas denominações são encontradas em outros trabalhos, a exemplo de [Overholts II et al., 2009], [Pillac et al., 2013], [Chen et al., 2016] e [Avraham et al., 2017].

Outra denominação mais recente na literatura é o Problema de Programação de Técnicos de Campo - FTSP, [Cortés et al., 2014; Damm et al., 2016]. A abordagem trata de um problema genérico, caracterizado pela alocação de técnicos com diferentes habilidades e velocidades para execução de serviços ou tarefas, as quais devem ser realizadas em determinadas janelas de tempo, minimizando o custo de deslocamento e a quantidade de técnicos alocados. Segundo Damm et al. [2016], o FTSP não possui uma literatura extensa, sendo iniciada a partir de 2001, com crescimento significativo observado entre 2005 e 2007, quando a Sociedade Francesa de Pesquisa Operacional propôs o desafio do problema de programação de tarefas em parceria com uma empresa de telecomunicações daquele país. A empresa lidava com um elevado crescimento da demanda de serviços, enquanto dispunha de uma quantidade limitada de técnicos. As tarefas foram divididas em três níveis de prioridade e a função objetivo consistia em minimizar a soma ponderada do tempo total de conclusão de todas as tarefas de cada grupo.

O presente artigo está organizado em seis seções. A Seção 2 apresenta o modelo matemático do problema. A Seção 3 descreve a meta-heurística utilizada, o ILS, enquanto a Seção 4 explica como o ILS foi adaptado para resolver o problema estudado. A Seção 5 mostra os resultados obtidos e as respectivas discussões. Por fim, a Seção 6 conclui o presente trabalho.

2. Modelo de Programação Linear Inteira Mista

A presente pesquisa fundamenta-se no modelo de programação linear inteira mista apresentado por [Damm et al., 2016]. Seja $J = \{1, \dots, n\}$ um conjunto de tarefas independentes, $K = \{1, \dots, m\}$ um conjunto de técnicos disponíveis para executá-las e p_i o tempo de processamento da tarefa i que possui a janela de tempo representada por $[e_i, l_i]$. Os técnicos trabalham em uma janela de tempo definida por $[a_k, b_k]$. O tempo de deslocamento da tarefa i para j é dado por c_{ij} , com $i, j \in J \cup \{0\}$, onde 0 é uma tarefa *dummy* que representa a origem (sede da empresa). Cada tarefa i tem prioridade w_i . A habilidade de um técnico k para executar a tarefa i é dada por s_{ik} , o qual consiste em um parâmetro binário com valor 1 se o técnico for habilitado e 0 caso contrário. Apresenta-se abaixo as variáveis do modelo:

- y_{ik} é uma variável binária que recebe valor 1 quando a tarefa i é atribuída ao técnico k e 0 caso contrário;
- x_{ijk} é uma variável binária que recebe valor 1 quando a tarefa i precede a tarefa j na rota do técnico k e 0 caso contrário;
- t_i é a hora de início da execução da tarefa i ;
- z_k é o tempo ocioso após o técnico k retornar à origem.

O objetivo é maximizar a soma dos valores de prioridades associadas às tarefas executadas a cada dia. Busca-se, também, minimizar o tempo de espera, e o tempo de deslocamento dos colaboradores entre as tarefas. Isso é feito maximizando o tempo ocioso (tempo reservado para atividades administrativas) após o retorno à sede. A formulação do modelo utilizado neste trabalho encontra-se apresentada a seguir.

$$\max \sum_{i \in J} \sum_{\substack{k \in K \\ |s_{ik}=1}} \frac{w_i y_{ik}}{MW} + \sum_{k \in K} \frac{z_k}{MZ} \quad (1)$$

sujeito a

$$\sum_{\substack{k \in K \\ |s_{ik}=1}} y_{ik} \leq 1 \quad i \in J \quad (2)$$

$$\sum_{\substack{j \in J \cup \{0\} \\ |s_{jk}=1}} x_{ijk} = \sum_{\substack{j \in J \cup \{0\} \\ |s_{jk}=1}} x_{jik} = y_{ik} \quad k \in K, i \in J, s_{ik} = 1 \quad (3)$$

$$\sum_{\substack{i \in J \\ |s_{ik}=1}} x_{0ik} \leq 1 \quad k \in K \quad (4)$$

$$e_i \leq t_i \leq l_i - p_i \sum_{\substack{k \in K \\ |s_{ik}=1}} y_{ik} \quad i \in J \quad (5)$$

$$t_i + p_i + c_{ij} \leq t_j + M(1 - x_{ijk}) \quad k \in K, i \neq j \in J, s_{ik} = s_{jk} = 1 \quad (6)$$

$$a_k + c_{0j} \leq t_j + M(1 - x_{0jk}) \quad k \in K, j \in J, s_{jk} = 1 \quad (7)$$

$$t_i + p_i + c_{i0} \leq b_k - a_k + M(1 - x_{i0k}) \quad k \in K, i \in J, s_{ik} = 1 \quad (8)$$

$$z_k \leq b_k - a_k \quad k \in K \quad (9)$$

$$y_{ik} \in \{0, 1\} \quad i \in J, k \in K, s_{ik} = 1 \quad (10)$$

$$x_{ijk} \in \{0, 1\} \quad i, j \in J \cup \{0\}, k \in K, s_{ik} = s_{jk} = 1 \quad (11)$$

$$l_i, z_k \in \mathbb{R}_+ \quad i \in J, k \in K, \quad (12)$$

no qual

$$MW = \min_{i \in J} w_i \quad (13)$$

$$MZ = \sum_{k \in K} (b_k - a_k) \quad (14)$$

$$M = \max(\max_{i \in J} l_i, \max_{k \in K} a_k) + \max_{i, j \in J \cup \{0\}} c_{i,j} \quad (15)$$

Segundo Damm et al. [2016], o primeiro termo da função objetivo (1) tem por finalidade selecionar técnicos para as tarefas de maior prioridade, enquanto o termo secundário minimiza o tempo de deslocamento total com o aumento do tempo ocioso. A inclusão deste segundo termo permite a realização de novas tarefas externas ou de trabalhos administrativos internos ao final do dia. A restrição representada na inequação (2) garante que cada tarefa é atribuída a, no máximo, um técnico. A equação (3) garante que somente um predecessor e um sucessor serão conferidos à tarefa i atribuída ao técnico k . De acordo com a restrição (4), cada técnico deixa a origem, no máximo, uma vez. Além disso, conforme apresentado em (5), cada tarefa deve ser realizada dentro da janela de tempo da mesma, e apenas por técnicos habilitados. A restrição (6) garante que, quando duas tarefas são atribuídas ao mesmo técnico, e são consecutivamente executadas, o início da tarefa j ocorre após a conclusão de sua predecessora, i , somado ao tempo de deslocamento entre as duas locações. As restrições (7) e (8) garantem que a solução respeite a janela de tempo dos técnicos. As restrições (6), (7) e (8) também previnem sub-rotas. A restrição (9) garante que o tempo ocioso dos técnicos após o retorno à origem é menor ou igual à amplitude de sua janela de tempo. As restrições (10) e (11) definem as variáveis x_{ijk} e y_{ik} como binárias, enquanto t_i e z_k são definidas por (12) como variáveis reais não negativas.

3. Iterated Local Search

De uma maneira geral, os métodos de solução podem ser divididos em exatos e heurísticos. Os métodos exatos garantem a otimalidade da solução encontrada, enquanto os heurísticos podem gerar boas soluções, mas sem garantia de ter atingido um resultado ótimo [Talbi, 2009]. Os métodos heurísticos ainda podem ser divididos em heurísticas formuladas exclusivamente para o problema, e meta-heurísticas. Enquanto a heurística específica trata de um procedimento, ou conjunto de etapas, em busca de uma boa solução viável para um determinado problema, a meta-heurística é mais genérica, podendo ser aplicada em uma grande diversidade de problemas.

Apesar da capacidade da meta-heurística em resolver problemas complexos de programação não linear e inteira, a sua aplicação é mais usual em problemas de otimização combinatória. Hillier e Lieberman [2010] definem meta-heurística como um método genérico de solução que busca, através de processos iterativos, incrementar a qualidade da solução, utilizando processos capazes de sair de ótimos locais, garantindo uma busca robusta dentro do espaço de soluções viáveis de um problema. Diversas meta-heurísticas foram desenvolvidas ao longo das últimas décadas, sendo

as principais sumarizadas em [Talbi, 2009; Gendreau e Potvin, 2010]. A principal classificação da meta-heurística refere-se ao processo de construção do algoritmo. Se o processo parte da melhoria de uma solução inicial (s-meta-heurística), é considerado de solução única. Por outro lado, se a cada iteração, uma população de soluções é gerada (outro conjunto de soluções), o método é considerado como populacional, ou baseada em população (p-meta-heurística). O presente trabalho propõe a aplicação da meta-heurística de solução única conhecida como ILS.

O ILS é considerado um método recente, proposto por Lourenço [1995] e Martin et al. [1992] sendo consolidado no início do século (2000 a 2003). Também conhecido como Descida Iterativa, Passo Largo da Cadeia de Markov, Otimização Local Cadenciada ou Iteração Lin-Kernighan, a ideia central do ILS é focar na busca de soluções que foram geradas por algoritmo específico (*black-box*) e não em todo o espaço de busca. O comportamento do mecanismo de busca pode ser caracterizado como uma construção de soluções em cadeia desse algoritmo. Este processo parte da construção de uma solução inicial, na qual é aplicada uma busca local. Então, a cada iteração, uma perturbação é realizada sobre o resultado encontrado, seguido de uma nova busca local. A solução encontrada é aceita sob certas condições e o processo continua até atingir um critério de parada. Apesar da simplicidade do ILS, a sua iteratividade incrementa o método clássico de busca local com múltiplas inicializações, em virtude de uma nova busca ser realizada sobre a perturbação da solução anterior. O algoritmo ILS consiste das etapas descritas a seguir [Lourenço et al., 2010]:

Algoritmo 1: *Iterated Local Search*

- 1 s^0 = Gerar Solução Inicial ;
 - 2 s^* = Busca Local(s^0) ;
 - 3 **repetir**
 - 4 s' = Perturbação (s^* , histórico) ;
 - 5 $s^{*'}$ = Busca Local(s') ;
 - 6 s^* = Critério de Aceitação(s^* , $s^{*'}$, histórico) ;
 - 7 **até atingir critério de parada;**
 - 8 **Saída:** Melhor solução encontrada.
-

A qualidade da solução resultante de um método de busca local depende em grande parte da solução inicial. Esta solução pode ser construída a partir de heurísticas consagradas, ou até mesmo de um método exato. No entanto, cada tipo de problema exigirá uma função que estabeleça um custo-benefício que seja aplicável ao contexto. Esta é a primeira etapa do ILS (1).

A partir da solução inicial gerada deve-se aplicar um algoritmo de busca local. A busca local (2) pode ser uma outra s-meta-heurística ou qualquer outro algoritmo específico, sendo considerada por alguns autores como "caixa-preta". As p-meta-heurísticas não são consideradas como candidatas ao procedimento de busca local, visto que manipulam populações. No entanto, algumas p-meta-heurísticas, as que têm como característica o conceito de perturbação de uma porção da população (um pedaço da solução) como processo de diversificação, também podem ser utilizadas [Talbi, 2009].

O operador de perturbação pode ser visto como um passeio aleatório dentro da solução corrente. O método deve manter uma parte da solução intacta enquanto altera a outra parte, no intuito de encontrar uma outra solução atrativa (4). A perturbação não pode ser muito pequena (para não ficar presa em um mesmo ótimo local), nem muito grande (de forma a não perder completamente a informação da solução anterior). Para tanto, deve-se ajustar os parâmetros de intensidade mínima e máxima da perturbação, a fim de encontrar um valor para o parâmetro de modo que

permita pesquisar a vizinhança da solução sem perder informações importantes.

Após a perturbação, uma nova busca local é realizada (5) e avaliada quanto à viabilidade da nova solução (6). A nova solução gerada é então comparada à anterior e ao histórico de soluções anteriores, pode ser aceita, ou não a partir de certos critérios de seleção. O procedimento continua até ter atingido o critério de parada, que pode ser por limite de tempo, número de iterações, entre outros (7). Após o término do processo, a melhor solução é retornada pelo algoritmo (8).

4. Implementação

O ILS detalhado nesta seção foi implementado na linguagem de programação C++. Os testes computacionais foram realizados em um *notebook* com processador *Intel Core i7 4720HQ 2.60GHz*, memória RAM de *8GB* e sistema operacional *Windows 10 Pro*. A representação da solução escolhida para o problema consiste em: uma matriz na qual cada linha representa uma equipe com entradas constituindo a sequência de tarefas que serão executadas; um vetor de tamanho n com o tempo de início de cada tarefa; um vetor de tamanho m contendo o tempo ocioso de cada equipe; e um vetor contendo as tarefas não realizadas. Na Figura 1 foi exemplificada a representação de uma solução para um problema com 16 tarefas e quatro equipes de técnicos.

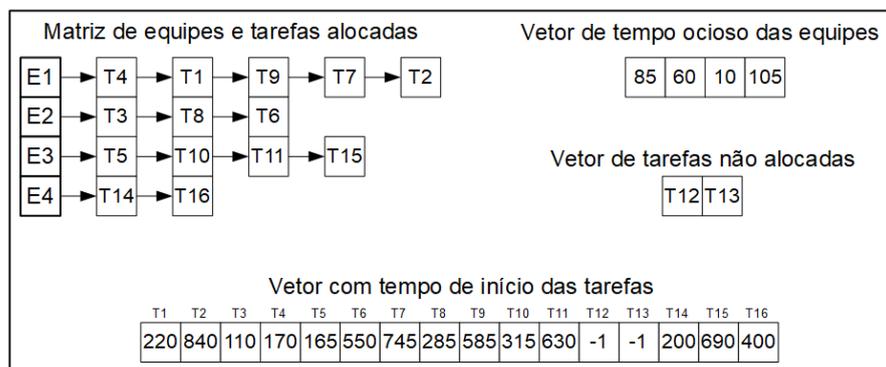


Figura 1: Exemplo de representação de solução

De acordo com a estratégia de implementação definida, a solução inicial foi montada com base em um vetor contendo todas as tarefas. Estas foram ordenadas a partir de uma heurística construtiva que estabelece uma ordem decrescente de prioridade para as tarefas e que leva em consideração também a quantidade de técnicos disponíveis e um critério de desempate. A heurística empregada segue, portanto, o critério de ordenação de tarefas sugerida em [Damm et al., 2016]:

$$\rho_i = \begin{cases} w_i - \frac{NST_i}{m+1}, & \text{se } \frac{n}{m} < 10 \\ w_i + \frac{w_i}{p_i} + \frac{p_i}{l_i - e_i}, & \text{c.c.} \end{cases}, \quad (16)$$

no qual NST_i é o número de técnicos habilitados para a tarefa i .

Assim, para instâncias com menos de 10 tarefas a serem executadas para cada equipe, foi inserida uma penalidade no cálculo referente ao número de técnicos qualificados disponíveis para determinada tarefa (dessa forma, quanto menos técnicos qualificados para uma dada tarefa, maior sua prioridade). Para instâncias com $\frac{n}{m} \geq 10$, ou seja, naquelas em que se aumenta a possibilidade de não execução de todas as tarefas apresentadas, foi inserido um bônus no cálculo, considerando o tempo de processamento das tarefas (tarefas que requerem menos tempo têm vantagem). Foi

inserido, ainda, um critério de desempate que garante que as tarefas com maior tempo de processamento e menor janela de tempo sejam tratadas com prioridade. Uma vez construído o vetor, as tarefas ordenadas são alocadas às equipes, respeitando as respectivas restrições.

O cálculo do valor função objetivo da solução considera a alocação das tarefas pelas equipes, bem como seus tempos ociosos. A alocação de tarefas visa minimizar o tempo ocioso das equipes. Assim, há uma preferência para que as mesmas sejam alocadas às equipes o mais próximo possível de seus respectivos horários de início. Como visto em (1), cada tarefa possui uma prioridade w_i , que influencia no valor final da função objetivo.

Sendo assim, o cálculo efetivo deste valor pode ser expresso pela equação:

$$FO = \sum_{m=0}^M \left(\sum_{i=0}^K \frac{w_i}{MW} \right) + \frac{Idle[m]}{MZ}, \quad (17)$$

sendo que, w_i é o valor de prioridade referente a tarefa i , MW é o menor valor de prioridade da instância em questão, K é o tamanho do vetor de tarefas alocadas para a equipe m , $Idle[m]$ o tempo ocioso da equipe m e MZ a soma das janelas de tempo de todas as equipes.

O operador de perturbação foi configurado para agir de duas formas. Primeiramente, ele remove todas as tarefas de um número k de equipes, dado $\beta_{min} \leq k \leq \beta_{max}$. É utilizado um parâmetro β que varia entre 2% e 10% do número de equipes. O valor k representa a quantidade de equipes que serão perturbadas. Os parâmetros foram calibrados a partir de testes empíricos.

Na Figura 2 é possível visualizar o efeito de uma perturbação na matriz de solução, onde duas equipes (E2 e E4) têm suas rotas limpas, transferindo suas tarefas anteriormente alocadas para o vetor de tarefas não alocadas.

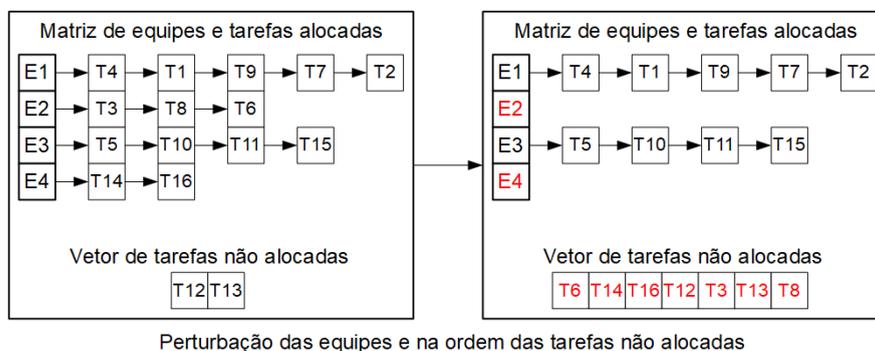


Figura 2: Exemplo de uma perturbação na matriz de equipes e rotas e no vetor de tarefas não realizadas

Em segundo lugar, o operador de perturbação altera a ordenação no vetor geral de tarefas, utilizando uma de cinco estratégias, selecionada aleatoriamente:

- organiza as tarefas em ordem decrescente por prioridade (maior prioridade primeiro);
- ordena de acordo com o número de técnicos habilitados a executar a tarefa (tarefas que podem ser executadas por mais técnicos por último);
- altera a ordenação do vetor realizando uma troca do tipo *Swap* (2,1) (duas tarefas aleatórias trocam de lugar entre si no vetor);

- segue o mesmo critério do vetor de tarefas da solução inicial;
- dispõe as tarefas em função da distância até a base (tarefas mais próximas antes).

As tarefas retiradas durante a fase de perturbação, em união com as tarefas que não eram executadas por nenhuma equipe, serão utilizadas na fase de busca local. Essa fase consiste em alocar as tarefas, que estão no vetor de tarefas não alocadas, aos técnicos. A alocação é efetuada respeitando a ordenação por prioridade efetuada no mesmo vetor, e em qual equipe, dentre as habilitadas, é capaz de iniciar a tarefa mais próximo da abertura de sua janela de tempo. A Figura 3 ilustra o processo em uma das equipes.

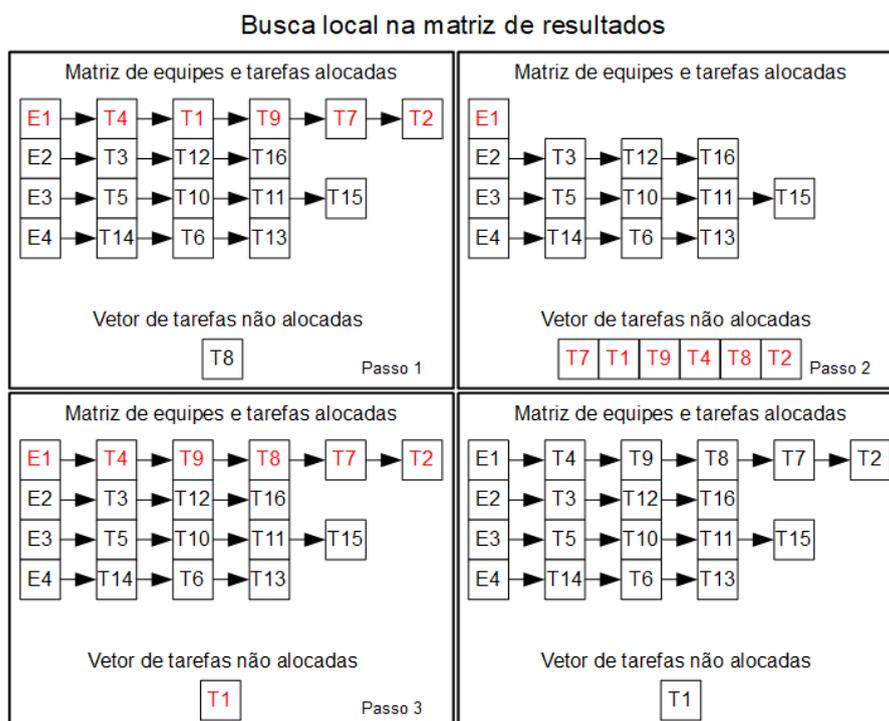


Figura 3: Exemplo da busca local "reiniciando" uma equipe por vez

Foi empregada uma estratégia de aceitação de solução pior, como forma de fugir de ótimos locais, após 50 iterações sem melhora no resultado da função objetivo. Para isso foi utilizado um contador que incrementa um a cada iteração em que o valor da função objetivo se mantém, zerando esse contador sempre que uma solução melhor é encontrada ou uma solução pior é aceita. Como critério de parada do algoritmo, foi estabelecido o valor de 2.000 iterações.

5. Resultados e Discussão

Na presente pesquisa foram empregadas instâncias geradas artificialmente e disponibilizadas por [Damm et al., 2016]. Segundo os autores, os dados levam em consideração diversos fatores como distribuição geográfica de clientes, tamanho da janela de tempo das tarefas, porcentagem de tarefas com janelas de tempo, tempo de processamento, prioridade de tarefas, número de tarefas e número de técnicos.

As regiões geográficas foram divididas em 4 categorias, de acordo com a distribuição da localização dos clientes: uniforme aleatória (R), agrupada (C), semi-agrupada (RC) e radial (RAD).

Para cada região foram criados 13 diferentes grupos de tarefas e técnicos e geradas 20 instâncias para cada grupo. Neste trabalho foram selecionados 4 grupos de instâncias, conforme discriminado na Tabela 1. Maiores detalhes sobre as instâncias podem ser consultados em [Damm et al., 2016].

Grupo	Nº de tarefas	Nº de técnicos
1	16	2
2	26	2
4	39	3
8	100	10

Tabela 1: Dimensão das instâncias selecionadas

Para cada categoria de distribuição geográfica de clientes, foram selecionadas vinte instâncias do grupo 1, vinte instâncias do grupo 2, três instâncias do grupo 4 e três instâncias do grupo 8. Com isso, foi utilizado um total de cento e oitenta e quatro instâncias. O algoritmo foi executado vinte vezes para cada instância selecionada. Como parâmetros de referência foram considerados os valores da função objetivo encontrados pela meta-heurística *Biased Random Key Genetic Algorithm* com heurística construtiva *Shortest Travel Time* (BRKGA-STT), disponíveis em [Damm et al., 2016].

Os resultados encontram-se detalhados na Tabela 2, onde podem ser verificados os melhores valores da função objetivo alcançados pelo ILS, os valores médios do ILS, os melhores valores do BRKGA-STT e os valores médios do BRKGA-STT. O *Gap 1* representa a diferença entre o melhor valor do ILS e o melhor valor do BRKGA-STT. O *Gap 2* corresponde à diferença entre a média dos resultados do ILS e a média dos resultados do BRKGA-STT. A Figura 4 ilustra a diferença percentual (%) entre a média dos resultados da função objetivo do ILS e a média dos resultados do BRKGA-STT (*Gap 2*) para cada categoria de distribuição geográfica de clientes.

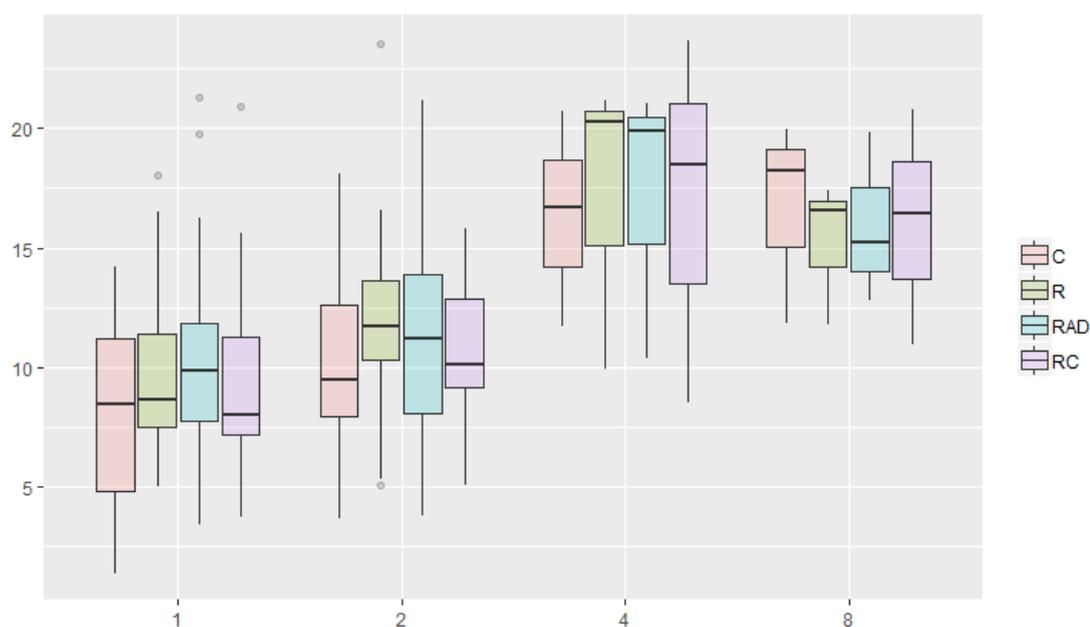


Figura 4: Diferença percentual entre a média do ILS e a média do BRKGA-STT (*Gap 2*) - %

Grupo	Distribuição agrupada (C)					
	Melhor ILS	Média ILS	Melhor BRKGA-STT	Média BRKGA-STT	Gap 1 %	Gap 2 %
1	88,08	61,91	89,07	66,73	1,11	7,22
2	107,03	82,50	112,10	92,04	4,52	10,36
4	148,00	123,8	170,03	140,57	12,96	11,93
8	414,03	369,90	479,01	431,57	13,56	14,29
Grupo	Distribuição uniforme aleatória (R)					
	Melhor ILS	Média ILS	Melhor BRKGA-STT	Média BRKGA-STT	Gap 1	Gap 2 %
1	86,06	58,66	92,04	64,76	6,50	9,42
2	95,06	70,61	104,05	79,85	8,64	11,57
4	134,03	101,27	149,01	129,88	10,05	22,03
8	386,06	360,60	460,02	426,67	16,08	15,48
Grupo	Distribuição radial (RAD)					
	Melhor ILS	Média ILS	Melhor BRKGA-STT	Média BRKGA-STT	Gap 1 %	Gap 2 %
1	77,01	51,13	81,04	56,79	4,97	9,97
2	100,06	70,33	107,02	79,03	6,50	11,01
4	153,03	104,94	168,00	124,78	8,91	15,90
8	383,03	360,90	477,01	421,47	19,70	14,37
Grupo	Distribuição semi-agrupada (RC)					
	Melhor ILS	Média ILS	Melhor BRKGA-STT	Média BRKGA-STT	Gap 1 %	Gap 2 %
1	80,09	56,61	81,05	62,19	1,18	8,97
2	109,03	75,23	116,02	84,39	6,02	10,85
4	123,04	108,00	175,01	135,28	29,69	20,16
8	414,04	371,50	469,04	425,08	11,73	12,60

Tabela 2: Resultados por tipos de distribuição geográfica de clientes

De acordo com os gráficos *boxplot* da Figura 4 e os dados da Tabela 2, conclui-se que, em relação à média de resultados, instâncias mais complexas tiveram *performance* menos favorável, o que é bastante intuitivo. Ainda no que se refere à média, os dados da tabela 2 indicam que a distribuição agrupada (C) apresentou os melhores resultados para os grupos 1, 2 e 4, enquanto a distribuição semi-agrupada (RC) obteve a melhor *performance* para o grupo 8. Por outro lado, a distribuição radial (RAD) apresentou o pior desempenho no grupo 1, e a distribuição uniforme aleatória (R) obteve a pior *performance* nos grupos 2, 4 e 8.

Por fim, em relação ao melhor resultado obtido para a função objetivo, pode-se constatar, por meio da Tabela 2, que a distribuição agrupada (C) obteve o melhor desempenho para os grupos 1 e 2. A distribuição radial (RAD) alcançou o melhor resultado para o grupo 4 e a distribuição semi-agrupada (RC) para o grupo 8. Por outro lado, a distribuição uniforme aleatória (R) apresentou os resultados menos favoráveis para os grupos 1 e 2. Para o grupo 3 a pior *performance* foi alcançada pela distribuição semi-agrupada (RC) e para o grupo 4 pela distribuição radial (RAD).

6. Conclusão

Neste trabalho revisou-se o problema proposto por [Damm et al., 2016], que trata da alocação de equipes com diferentes capacidades para execução de diferentes tarefas com custos e janelas de tempo específicas, em atendimento à demanda de serviços de clientes. Foi aplicada a meta-heurística *Iterated Local Search*, com função objetivo que maximiza o número de clientes atendidos no menor tempo, levando em consideração a prioridade das tarefas. Foram desenvolvidas cinco estratégias construtivas de solução inicial. Como busca local foi aplicada a ideia de reinicialização de equipes para exploração da vizinhança. Para contornar ótimos locais, foi empregada uma estratégia de aceitação de pior solução após certo número de iterações sem melhora da solução corrente.

Os resultados obtidos foram mais satisfatórios para instâncias menores, pois o método atingiu valores ótimos em algumas instâncias, demonstrando eficácia. A distribuição agrupada (C) apresentou, em geral, os melhores resultados para instâncias de pequeno porte, enquanto a distribuição uniforme aleatória (R) obteve os piores resultados para instâncias mais complexas. Na medida em que se aumentou a dimensão das instâncias, o *gap* percentual (%) entre o método proposto e a literatura, relativo ao melhor valor encontrado da função objetivo, elevou-se significativamente, evidenciando uma inconsistência no que diz respeito à robustez do método.

Como etapas subsequentes, pretende-se aprimorar o algoritmo por meio de estratégias alternativas, bem como pela otimização do código e ajuste dos parâmetros, buscando-se uma maior robustez na metodologia proposta neste trabalho. Projeta-se, também, o estudo de variações nos métodos de busca local e perturbação aplicados ao ILS no problema estudado, no intuito de obter melhores resultados. Após uma aproximação das soluções ótimas apresentadas na literatura, serão envidados esforços no sentido de mensurar e minimizar, se for o caso, o custo computacional da implementação da meta-heurística objeto deste trabalho. Pretende-se, ainda, implementar a meta-heurística ILS em outros problemas para analisar sua performance em relação a outros métodos da literatura.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (Processo nº 2018/15417-8) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Processo nº 303736/2018-6, Processo nº 423694/2018-9).

Referências

- Avraham, E., Raviv, T., e Khmelnitsky, E. (2017). The decentralized field service routing problem. *Transportation Research Part B: Methodological*, 104:290–316. ISSN 0191-2615. URL <https://www.sciencedirect.com/science/article/pii/S0191261517305878>.
- Cardoso, F. A. (2019). *Modelagem multiobjetivo para dimensionamento de recursos em suporte a operações de fiscalização em fronteiras*. PhD thesis, Technological Institute of Aeronautics.
- Chen, X., Hewitt, M., e Thomas, B. W. (2018). An approximate dynamic programming method for the multi-period technician scheduling problem with experience-based service times and stochastic customers. *International Journal of Production Economics*, 196:122–134. ISSN 0925-5273. URL <https://www.sciencedirect.com/science/article/pii/S0925527317303432>.
- Chen, X., Thomas, B. W., e Hewitt, M. (2016). The technician routing problem with experience-based service times. *Omega*, 61:49–61. ISSN 0305-0483. URL <https://www.sciencedirect.com/science/article/pii/S0305048315001486>.
- Cordeau, J.-F., Laporte, G., Pasin, F., e Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *J Sched*, 13:393–409. URL <https://link.springer.com/content/pdf/10.1007/s10951-010-0188-7.pdf>.
- Cortés, C. E., Gendreau, M., Rousseau, L. M., Souyris, S., e Weintraub, A. (2014). Branch-and-price and constraint programming for solving a real-life technician dispatching problem. *European Journal of Operational Research*, 238(1):300–312. ISSN 0377-2217. URL <https://www.sciencedirect.com/science/article/pii/S0377221714002203>.

- Damm, R. B., Resende, M. G., e Ronconi, D. P. (2016). A biased random key genetic algorithm for the field technician scheduling problem. *Computers & Operations Research*, 75:49–63. ISSN 03050548. URL <https://www.sciencedirect.com/science/article/pii/S0305054816301101><https://linkinghub.elsevier.com/retrieve/pii/S0305054816301101>.
- Gendreau, M. e Potvin, J.-Y. (2010). *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*. Springer US, Boston, MA, 2 edition. ISBN 978-1-4419-1663-1. URL <https://www.springer.com/us/book/9781441916631><http://link.springer.com/10.1007/978-1-4419-1665-5>.
- Hashimoto, H., Sylvain Boussier, ., Vasquez, M., Wilbaut, C., Hashimoto, H., Boussier, S., Vasquez, M., e Wilbaut, C. (2011). A GRASP-based approach for technicians and interventions scheduling for telecommunications. *Ann Oper Res*, 183:143–161. URL <http://www.g-scop.fr/ChallengeROADEF2007/>.
- Hillier, F. S. e Lieberman, G. J. (2010). *Introduction To Operations Research*. McGraw-Hill, New York, NY, 9 edition. ISBN 978-0-07-337629-5. URL <http://www.worldcat.org/isbn/0078414474>.
- Lourenço, H. R., Martin, O. C., e Stützle, T. (2010). *Iterated Local Search: Framework and Applications*, p. 363–397. Springer US, Boston, MA. ISBN 978-1-4419-1665-5. URL https://doi.org/10.1007/978-1-4419-1665-5_12.
- Lourenço, H. R. (1995). Job-shop scheduling: Computational study of local search and large-step optimization methods. *European Journal of Operational Research*, 83(2):347 – 364. EURO Summer Institute Combinatorial Optimization.
- Martin, O., Otto, S. W., e Felten, E. W. (1992). Large-step markov chains for the tsp incorporating local search heuristics. *Operations Research Letters*, 11(4):219 – 224.
- Overholts II, D. L., Bell, J. E., e Arostegui, M. A. (2009). A location analysis approach for military maintenance scheduling with geographically dispersed service areas. *Omega*, 37(4):838–852. ISSN 03050483. URL <https://linkinghub.elsevier.com/retrieve/pii/S0305048308000509>.
- Pillac, V., Guéret, C., e Medaglia, A. L. (2013). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 7(7):1525–1535. ISSN 1862-4472. URL <http://link.springer.com/10.1007/s11590-012-0567-4>.
- Talbi, E.-G. (2009). *Metaheuristic: from design to implementation*. John Wiley & Sons, Hoboken, NJ, USA. ISBN 9780470278581.
- Tang, H., Miller-Hooks, E., e Tomastik, R. (2007). Scheduling technicians for planned maintenance of geographically distributed equipment. *Transportation Research Part E: Logistics and Transportation Review*, 43(5):591–609. ISSN 1366-5545. URL <https://www.sciencedirect.com/science/article/pii/S1366554506000329>.