

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO



**PROPOSTA DE UM ALGORITMO PARA O PROBLEMA DE
AGENDAMENTO DE TÉCNICOS DE CAMPO**

ALEXANDRE KRAUSZ

MONOGRAFIA EM ENGENHARIA DE PRODUÇÃO

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
Departamento de Engenharia de Produção

ALEXANDRE KRAUSZ

Proposta de um algoritmo para o problema de agendamento de técnicos de campo

Monografia apresentada ao Departamento de Engenharia de Produção da Universidade Federal de São Carlos (UFSCar), Campus São Carlos, como parte dos requisitos para obtenção do título de bacharel em Engenharia de Produção

Orientador: Prof. Dr. Roberto Fernandes Tavares Neto

SÃO CARLOS-SP
2025

RESUMO

KRAUSZ, A. **Proposta de um algoritmo para o problema de agendamento de técnicos de campo.** 31 f. Monografia (Graduação em Engenharia de Produção) Universidade Federal de São Carlos (UFSCar), São Carlos, 2025.

O problema de agendamento de técnicos de campo é comumente enfrentado por empresas do ramo de telecomunicações e deriva do problema de roteamento de veículos, do qual se diferencia por restrições relativas às habilidades dos técnicos. O objetivo da pesquisa é alocá-los a tarefas de modo a minimizar o *makespan* do sistema e maximizar o valor de prioridade das tarefas realizadas, além de analisar o impacto da nova família de restrições no desempenho do algoritmo. Para fornecer solução ao problema, foi proposta uma heurística de programação linear inteira mista (PLIM) composta por duas etapas. Foram realizados testes computacionais com 15 instâncias. O algoritmo apresentou melhor desempenho com técnicos aptos a desenvolver uma quantidade menor de tarefas, sem prejuízo significativo às soluções encontradas.

Palavras-chave: Agendamento de técnicos, heurística PLIM, *makespan*, prioridades

ABSTRACT

KRAUSZ, A. **Proposal of an algorithm for the field technician scheduling problem.** 31 p. Undergraduate Monograph (Bachelor's in Industrial Engineering), Federal University of São Carlos (UFSCar), São Carlos, 2025.

The field technician scheduling problem is commonly faced by companies in the telecommunications sector and derives from the vehicle routing problem, differing due to constraints related to technicians' skills. The objective of this research is to allocate technicians to tasks in a way that minimizes the system's makespan and maximizes the priority value of completed tasks, while also analyzing the impact of the new family of constraints on the algorithm's performance. To provide a solution to the problem, a mixed-integer linear programming (MILP) heuristic composed of two stages was proposed. Computational tests were conducted using 15 instances. The algorithm demonstrated better performance when technicians were capable of performing fewer tasks, without significant detriment to the quality of the solutions obtained.

Keywords: Technician scheduling, MILP heuristic, makespan, priorities

SUMÁRIO

1 INTRODUÇÃO	5
1.1 CARACTERIZAÇÃO DO TEMA	5
1.2 FORMULAÇÃO DO PROBLEMA E OBJETIVO DA PESQUISA	6
1.3 JUSTIFICATIVA DA PESQUISA	9
1.4 ORGANIZAÇÃO DO TRABALHO	9
2 REVISÃO DE LITERATURA	10
2.1 PROBLEMA DE ROTEAMENTO DE VEÍCULOS (VRP)	11
2.1.1 Modelo matemático do VRP	11
2.2 PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM PRÊMIO (PCVRP)	13
2.2.1 Modelo matemático do PCVRP	13
2.3 PROBLEMA DE AGENDAMENTO DE TÉCNICOS DE CAMPO (FTSP)	14
2.3.1 Modelo matemático do FTSP	17
2.4 VIZINHO MAIS PRÓXIMO (NN)	17
3 MÉTODO DE PESQUISA	19
3.1 JUSTIFICATIVA DA ABORDAGEM E MÉTODO DE PESQUISA	19
3.2 MODELAGEM	19
3.3 PROCEDIMENTO	21
4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS	22
4.1 INSTÂNCIAS DO PROBLEMA E DETALHES DA IMPLEMENTAÇÃO	22
4.2 ANÁLISE DOS RESULTADOS	22
4.2.1 Análise de performance	23
4.2.2 Análise das soluções encontradas	26
5 CONCLUSÃO	28
5.1 LIMITAÇÕES DA PESQUISA	29
5.2 PROPOSTA DE TRABALHOS FUTUROS	29
REFERÊNCIAS	30

1 INTRODUÇÃO

Este capítulo contextualiza o tema, além de apresentar o problema de pesquisa e sua relevância acadêmica e prática. Além disso, esclarece os objetivos principais e as questões norteadoras do trabalho. Assim, oferece uma visão geral da estrutura da monografia.

1.1 CARACTERIZAÇÃO DO TEMA

Na Programação Linear Inteira Mista (PLIM), em inglês *Mixed-Integer Linear Programming* (MILP), utilizam-se variáveis inteiras e reais para modelagem de problemas de otimização. Essa técnica é amplamente utilizada em problemas de logística, como de roteamento de veículos e localização de facilidades (Arenales *et al.*, 2007).

Segundo Arenales *et al.* (2007), o Problema de Roteamento de Veículos (PRV), termo em inglês *Vehicle Routing Problem* (VRP) envolve a projeção de rotas de entrega e/ou coleta de custo mínimo, partindo de um ou mais depósitos para um número de clientes, sujeito a restrições adicionais. Além disso, o problema é NP-difícil, isto é, não é solucionável em tempo polinomial (Lenstra e Rinnooy, 1981).

Um ramo do VRP consiste no Problema de Roteamento de Veículos com Prêmios, em inglês *Prize-Collecting Vehicle Routing Problem* (PCVRP). Neste, o objetivo passa a ser a maximização do prêmio obtido, que por sua vez trata-se da combinação entre o número de clientes atendidos e suas prioridades. Diferentemente do primeiro, nesse caso, não é obrigatório atender todos os nós do problema.

Além disso, segundo Assunção e Mateus (2019), o Problema de Orientação de Equipe de Steiner, em inglês *Steiner Team Orienteering Problem* (STOP) consiste em um conjunto de veículos idênticos, cujo objetivo é maximizar a soma total da recompensa coletada, respeitando restrições relativas aos tempos de viagem de cada veículo. Adicionalmente, são fornecidos locais que, obrigatoriamente, devem ser visitados. De acordo com Tavares *et al.* (2024), é uma variação do PCVRP, com pontos obrigatórios. Por fim, o STOP também é considerado NP-difícil.

A presente pesquisa tem como foco o Problema de Agendamento de Técnicos de Campo (FTSP). Segundo Xu e Chiu (2001, p. 2), o objetivo primário do FTSP é agendar o máximo de tarefas possíveis dentro de uma janela de tempo, com restrições relativas às habilidades necessárias para realizar a tarefa e levando em consideração a prioridade das tarefas. Um

objetivo secundário é minimizar o total de horas trabalhadas, incluindo tempos de espera e viagem de todos os técnicos. Por fim, o objetivo terciário é minimizar o tempo total de viagem dos técnicos.

O problema consiste na alocação de tarefas a equipes, desde que estas possuam as habilidades necessárias para realização das primeiras. De acordo com Xu e Chiu (2001), o FTSP se assemelha ao Problema de Roteamento de Veículos com Janelas de Tempo, em inglês *Vehicle Routing Problem with Time Windows* (VRPTW), que é NP-difícil. Porém, segundo o autor, os problemas se diferem em dois aspectos: no VRPTW, o objetivo é minimizar o número de veículos (técnicos, no caso do FTSP) necessários ou a distância percorrida. Enquanto isso, o FTSP busca, entre outros objetivos, maximizar o número de tarefas completas dentro de um período de tempo. Além disso, enquanto os veículos apresentam restrições em relação à capacidade física, no FTSP as restrições estão relacionadas às habilidades requeridas pela tarefa.

1.2 FORMULAÇÃO DO PROBLEMA E OBJETIVO DA PESQUISA

O problema foi desenvolvido baseado no cenário encontrado em uma empresa de distribuição de energia elétrica, que possui tarefas obrigatórias e não obrigatórias. As primeiras devem ser realizadas no menor tempo possível e para as últimas, deve-se maximizar a combinação entre prioridades e tarefas realizadas. Ambas estão sujeitas à limitação do horário de trabalho dos técnicos e a sua habilidade de realizar determinadas tarefas. A tabela 1 e figuras 1-5 representam um exemplo simples desse problema. Neste, estão disponíveis 3 equipes para a realização de 12 tarefas, das quais 8 são obrigatórias e 4 são não obrigatórias. Inicialmente, as equipes são alocadas às primeiras e, em seguida, às últimas, partindo do último nó obrigatório. Além disso, as equipes partem da base e retornam a ela (nó zero).

Os parâmetros de entrada do problema são as ordens de serviços e os técnicos disponíveis para realizá-las. As ordens de serviço são caracterizadas pela sua localização, tempo de processamento, tempo de setup, obrigatoriedade e prioridade. Enquanto isso, os técnicos possuem habilidades distintas, que os tornam aptos ou não para a realização de uma determinada tarefa, e horários de trabalho.

Em função da problemática acima, o trabalho visa responder à questão: qual a melhor forma de alocar técnicos a tarefas obrigatórias e não obrigatórias, respeitando restrições relacionadas às suas habilidades e ao tempo disponível para realização das tarefas?

Assim, o objetivo da pesquisa é alocar técnicos a tarefas de modo a minimizar o *makespan* do sistema e maximizar o valor de prioridade das tarefas realizadas, considerando as habilidades dos técnicos e o tempo disponível. Esse objetivo pode ser desdobrado em objetivos específicos:

- minimizar o *makespan* das tarefas obrigatórias;
- maximizar a combinação entre realização de tarefas não obrigatórias e suas prioridades.

Por fim, o artigo visa analisar a influência da restrição relativa às habilidades dos técnicos nos resultados obtidos e desempenho do algoritmo.

Tabela 1 - Exemplo de obrigatoriedade das tarefas e habilidade dos técnicos

Tarefa/Equipe	Obrigatoriedade	A	B	C
1	0	1	1	0
2	0	1	0	1
3	1	1	0	1
4	1	0	1	1

Figura 1 - Mapa inicial hipotético

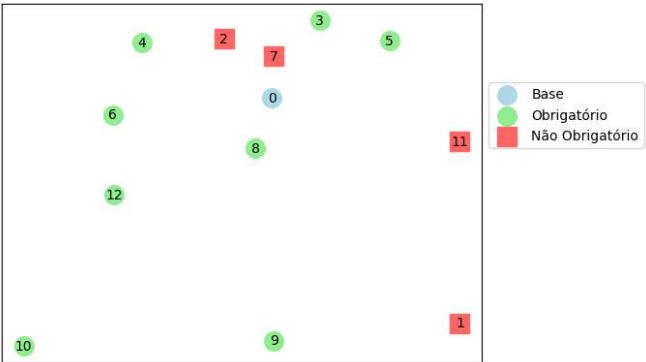


Figura 2 - Solução VRP

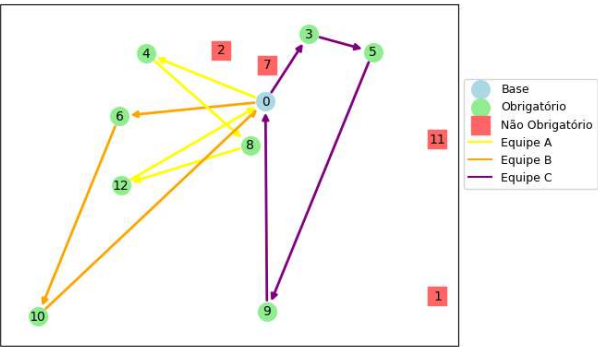


Figura 3 - Makespan VRP

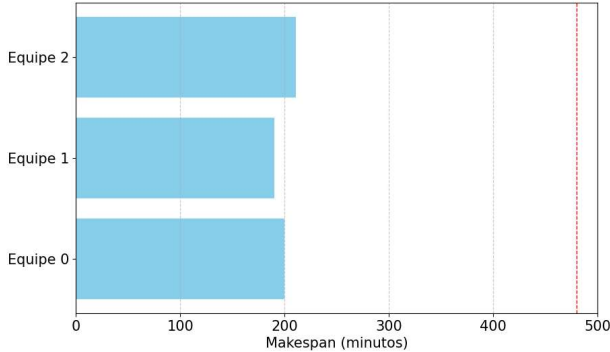


Figura 4 - Solução PCVRP

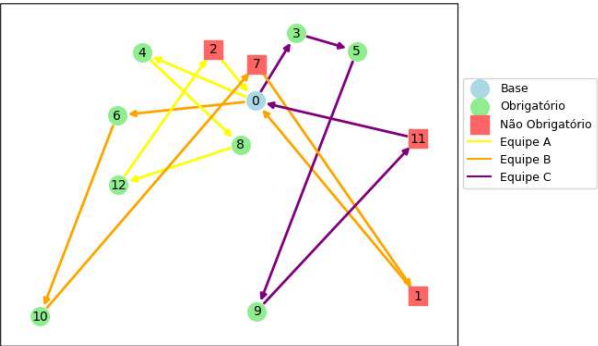
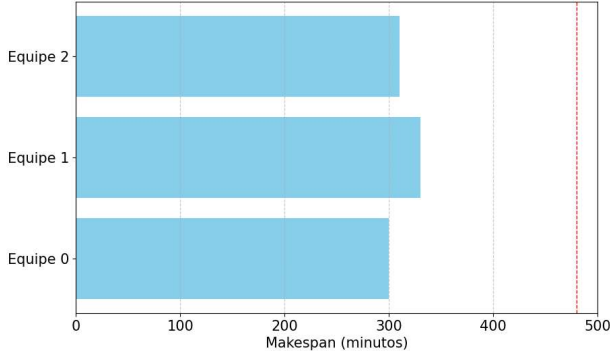


Figura 5 - Makespan PCVRP



1.3 JUSTIFICATIVA DA PESQUISA

O FTSP é um problema tipicamente enfrentado por empresas que oferecem serviços, como, por exemplo, as do ramo de telecomunicações (DAMM; RESENDE; RONCONI, 2016). Além disso, de acordo com Yahiaoui (2023), o problema tem aplicações em outras áreas, como na manutenção de infraestrutura de transportes (redes rodoviárias e ferroviárias) e instalações de esgoto.

Segundo Yahiaoui (2023), uma parte significativa dos custos de manutenção decorre de custos operacionais associados à equipe técnica. Dessa forma, a otimização do uso da força de trabalho disponível é essencial para a redução dos custos totais da empresa. Nesse sentido, destaca-se a importância do planejamento das intervenções técnicas, do balanceamento de carga de trabalho, do aumento da produtividade e do agendamento e roteirização dos técnicos.

De acordo com o mesmo autor, o FTSP é uma variação do estudo do agendamento da força de trabalho. Este, por sua vez, é relevante ao estudo de logística e transporte por sua larga aplicabilidade (Castillo-Salazar *et al.*, 2016). Isso inclui, além do problema foco da pesquisa, a alocação da força de trabalho, a roteirização de pessoal de segurança e serviços de atendimento domiciliar. Dessa forma, a partir do agendamento da força de trabalho, é possível, além de aumentar a eficiência e reduzir custos da operação, melhorar a satisfação do cliente após a entrega do produto, necessário para a manutenção e aumento do *market-share* da empresa (Khalfay *et al.*, 2017).

Portanto, o tema da pesquisa é relevante para o aumento de produtividade da empresa, redução do custo de transporte, de terceirização e de horas extras, aumento do número das tarefas realizadas e balanceamento das cargas de trabalho dos técnicos.

1.4 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado da seguinte forma. No Capítulo 2, descreve-se o referencial teórico, seguido pelo Capítulo 3, que destaca os procedimentos metodológicos adotados na pesquisa. O Capítulo 4 apresenta os resultados obtidos e o Capítulo 5 apresenta as conclusões da pesquisa.

2 REVISÃO DE LITERATURA

A modelagem apresentada consiste em uma adaptação dos problemas VRP e PCVRP a um problema de agendamento de técnicos, proposto por Tavares *et al.* (2024). Nessa abordagem, inicialmente, minimizou-se o *makespan*, isto é, o tempo entre o início da primeira tarefa e o final da última, das tarefas obrigatórias por meio da aplicação do VRP. Em seguida, por meio do PCVRP, foi maximizada a relação entre as tarefas não obrigatórias realizadas e suas prioridades, sendo valores altos relativos às tarefas mais prioritárias. Os valores das prioridades foram definidos aleatoriamente e variaram para cada instância do problema. Além disso, as equipes deveriam sair de e retornar a uma base, representada pelo nó 0.

Quadro 1 - Parâmetros do problema

Símbolo	Descrição
N	Set de tarefas
K	Set de equipes
n	Número total de tarefas = $ N $
o_i	Obrigatoriedade da tarefa i
p_i	Prioridade da tarefa i
h_{ik}	1 se o time k é capaz de executar a tarefa i , 0 caso contrário
t_{ij}	Tempo necessário para ir de i para j e realizar as tarefas em j
d_{ij}	Distância entre i e j
tt_{ij}	Tempo necessário para ir de i para j
ts_j	Tempo de setup da tarefa j
tp_j	Tempo de processamento da tarefa j

Quadro 2 - Variáveis

Símbolo	Tipo	Descrição
y_{ik}	Binária	1 se a tarefa i é executada pelo time k , 0 caso contrário
x_{ijk}	Binária	1 se o time k utiliza o arco i, j , 0 caso contrário
m_k	Contínua	<i>Makespan</i> do time k
m	Contínua	<i>Makespan</i> do sistema
u_i	Contínua	Posição relativa de visita do nó i na rota

As distâncias em quilômetros (km) foram calculadas por meio da fórmula de Haversine, a partir dos valores de latitude e longitude dos pontos. Além disso, o tempo de viagem foi calculado a partir das distâncias, com base na velocidade média de 40 km/h. Por fim, não é necessário realizar o setup de uma tarefa caso ela esteja próxima geograficamente da tarefa anterior. Assim, o cálculo de t_{ij} varia:

$$t_{ij} = \begin{cases} tt_{ij} + tp_j & \text{se } d_{ij} < 100 \\ tt_{ij} + ts_j + tp_j & \text{caso contrário} \end{cases} \quad (1)$$

2.1 PROBLMA DE ROTEAMENTO DE VEÍCULOS (VRP)

Segundo Toth e Vigo (2002), o VRP é um dos problemas mais estudados entre os problemas de otimização, por conta da sua relevância e nível considerável de dificuldade. Em inglês, o *capacitated* VRP (CVRP) consiste em um VRP em que veículos com capacidade limitada devem retirar ou entregar itens em diversas localidades. Os autores propuseram um algoritmo de ramificar e limitar, em inglês *branch and bound*, e algoritmos exatos baseados no primeiro para solucionar o VRP.

2.1.1 Modelo matemático do VRP

O modelo matemático proposto visa minimizar *makespan* para todas as equipes. Por meio dele, aloca-se times às tarefas obrigatórias. Para eliminar sub-rotas, foi proposta uma adaptação da formulação apresentada por Miller, Tucker e Zemlin (1960), conhecida como MTZ. A equação proposta (equação 7) atribui valores de "ordem" u_i a cada nó visitado e, para cada arco (i,j) percorrido pela equipe k , força u_i a ser menor que u_j somado a um termo que depende de n (número total de tarefas) e da variável binária x_{ijk} . Quando $x_{ijk} = 1$, a restrição impõe que u_j seja pelo menos 1 unidade maior que u_i . Assim, garante que a rota de cada equipe k siga uma sequência contínua.

As equações 2 e 3 compõem a função objetivo. A equação 4 relaciona as variáveis y_{ik} e x_{ijk} , de modo que a tarefa seja atribuída desde que a equipe utilize um arco à qual ela pertença. As equações 5 e 6 restringem os times a saírem e chegarem à base apenas uma vez, respectivamente. A equação 7 implica que as tarefas obrigatórias sejam realizadas exatamente uma vez, por um único time k . Além disso, a equação 8 consiste no balanceamento entre

entradas e saídas, isto é, restringe a equipe k a deixar um nó se e somente se entrar nesse nó. A equação 9 elimina sub-rotas, baseado na formulação MTZ. A equação 10 calcula os tempos de execução dos times k . A equação 11 restringe o *makespan* de cada equipe a 480 minutos, equivalente a oito 8 horas. Por fim, as equações 12 e 13 desativam a execução de tarefas não obrigatórias e arcos $i - i$, respectivamente. Dessa forma, o modelo é composto pelas equações 2 – 13.

$$\text{Min } m \quad (2)$$

$$m \geq m_k \quad \forall k \in K \quad (3)$$

$$y_{ik} = \sum_{j \in N \setminus \{i\}} x_{ijk} \quad \forall i \in N, k \in K \mid i \neq 0 \quad (4)$$

$$\sum_{j \in N \setminus \{0\}} x_{0,j,k} = 1 \quad \forall k \in K \quad (5)$$

$$\sum_{i \in N \setminus \{0\}} x_{i,0,k} = 1 \quad \forall k \in K \quad (6)$$

$$\sum_{\substack{k \in K \\ j \in N}} x_{ijk} = 1 \quad \forall i \in N \mid i \neq 0, o_i = 1 \quad (7)$$

$$\sum_{j \in N \setminus \{i\}} x_{ijk} = \sum_{j \in N \setminus \{i\}} x_{jik} \quad \forall i \in N, k \in K \mid i \neq 0 \quad (8)$$

$$u_i - u_j + (n * x_{ijk}) \leq (n - 1) \quad \forall k \in K, i, j \in N \mid i \neq j, i \neq 0, j \neq 0 \quad (9)$$

$$m_k = \sum_{\substack{i \in N \\ j \in N \setminus \{i\}}} t_{ij} \cdot x_{ijk} \quad \forall k \in K \quad (10)$$

$$m_k \leq 480 \quad \forall k \in K \quad (11)$$

$$y_{ik} = 0 \quad \forall i \in N, k \in K \mid o_i = 0 \quad (12)$$

$$x_{iik} = 0 \quad \forall i \in N, k \in K \quad (13)$$

A solução fornecida por esse modelo é armazenada nas variáveis $x1_{ijk}$ e $y1_{ik}$, utilizadas na etapa do PCVRP.

2.2 PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM PRÊMIO (PCVRP)

De acordo com Li e Tian (2016), enquanto no VRP o objetivo é voltado a minimizar custo, tempo total ou distância, no PCVRP o objetivo passa a ser maximizar o prêmio total. Além disso, no VRP existem clientes obrigatórios a serem atendidos, ao passo que no PCVRP existe uma carteira de possíveis clientes a serem atendidos. Para solucionar o PCVRP, os autores propuseram um algoritmo de busca de vizinhança variável auto-adaptativa de dois níveis, em inglês *two-level self-adaptive variable neighborhood search* (TL SAVNS).

Tavares *et al.* (2024) propuseram um modelo de Programação Linear Inteira Mista, em inglês *Mixed Integer Linear Programming* (MILP) e dois algoritmos de Busca Local Iterada (ILS). O objetivo destes foi melhorar uma solução inicial factível, por meio de operadores de substituição, destruição e inserção. Além disso, os algoritmos desenvolvidos foram aplicados com base em dados de uma empresa que realiza manutenções em sistemas de distribuição de energia.

2.2.1 Modelo matemático do PCVRP

A função objetivo do modelo abaixo (equação 14) maximiza a relação entre as tarefas realizadas e a prioridade das tarefas. A equação 15 restringe uma tarefa i a ser realizada por, no máximo, uma equipe k e sucedida por uma tarefa j . Nesse caso, diferentemente da primeira etapa, a execução de todas as tarefas não é obrigatória. Por fim, as equações 16 e 17 representam as tarefas concluídas e arcos utilizados na primeira etapa, exceto o que retorna à base. Assim, o modelo passa a ser composto pelas equações 4 – 6, 8 – 11, 13 – 17.

$$\text{Max} \sum_{\substack{i \in N \\ k \in K}} y_{ik} * p_i \quad (14)$$

$$\sum_{\substack{k \in K \\ j \in N}} x_{ijk} \leq 1 \quad \forall i \in N \mid i \neq 0 \quad (15)$$

$$x_{ijk} = 1 \quad \forall (i, j, k) \mid x1_{ijk} > 0,5, j \neq 0 \quad (16)$$

$$y_{ik} = 1 \quad \forall (i, k) \mid y_{1ik} > 0,5 \quad (17)$$

2.3 PROBLEMA DE AGENDAMENTO DE TÉCNICOS DE CAMPO (FTSP)

Tsang e Voudouris (1997) foram pioneiros em relação ao problema de agendamento de técnicos (DAMM; RESENDE; RONCONI, 2016). O problema era enfrentado pela *British Telecom*, a companhia de telecomunicações mais antiga do mundo, originada em 1846. Em sua abordagem, o objetivo era a minimização do custo total, composto pelo custo de deslocamento dos engenheiros e de horas extras. No modelo proposto, a habilidade dos técnicos era medida em termos de sua agilidade, cujo valor variava de 0 a 1 e era combinado ao tempo padrão de realização de cada tarefa. Por fim, foram desenvolvidas duas heurísticas para o problema: Busca Local Rápida, em inglês *Fast Local Search* e Busca Local Guiada, em inglês *Guided Local Search*.

Xu e Chiu (2001) propuseram procedimentos heurísticos, que incluem: uma heurística gulosa, em inglês *greedy heuristic*, um algoritmo de busca local e um Procedimento de Busca Adaptativa Gananciosa Randomizada, em inglês *Greedy Randomized Adaptive Search Procedure* (GRASP), posteriormente aplicada por Hashimoto *et al.* (2011). As soluções foram aplicadas em um problema de agendamento de técnicos de campo enfrentado por provedores de serviços na indústria de telecomunicações. Nessa abordagem, o objetivo foi maximizar o número de tarefas realizadas, levando em consideração as restrições de habilidade e urgência dos serviços.

Tang *et al.* (2006) formularam o problema de agendamento como um problema de múltiplos percursos com coleta máxima de recompensas dependentes do tempo. Além disso, consideraram a possibilidade de alocar tarefas em dias diferentes, entre os quais a prioridade de cada tarefa variava. Para encontrar solução ao problema, os autores desenvolveram uma heurística de busca Tabu com memória adaptativa.

Kovacs *et al.* (2012) foram um dos primeiros a estender o problema ao roteamento dos técnicos, visto que, antes, o problema se limitava ao agendamento. Ou seja, consideraram o custo-benefício das rotas na solução. Os autores utilizaram uma meta-heurística de Busca Adaptativa de Grandes Vizinhanças, em inglês *Adaptive Large Neighborhood Search* (ALNS), que é baseada no princípio de destruir e reparar uma solução ao longo de iterações. Assim, caracterizou-se como uma extensão do FTSP, denominado como Problema de Agendamento e Roteamento de Técnicos, em inglês *Technician Routing and Scheduling Problem* (TRSP).

Pillac, Gueret, Medaglia (2012) abordaram o TRSP dinâmico, no qual novas solicitações aparecem ao longo do tempo. Os autores propuseram uma abordagem de re-otimização rápida, baseada em uma busca paralela adaptativa de grandes vizinhanças, em inglês *parallel Adaptive Large Neighborhood Search* (pALNS).

Além disso, Damm *et al.* (2016) abordaram o FTSP considerando a atribuição de tarefas com diferentes janelas de tempo, prioridades e tempos de processamento em diferentes locais. Inicialmente, propuseram um modelo de PLIM. Em seguida, os autores apresentaram 3 heurísticas construtivas: tempo mais curto de viagem, técnico mais próximo e Cluster, isto é, áreas com concentração de tarefas. Por fim, foi proposto um Algoritmo Genético de Chaves Aleatórias Viciadas, em inglês *Biased Random Key Genetic Algorithm* (BRKGA), que consiste em uma meta-heurística evolutiva para problemas de otimização discreta e global baseada no algoritmo de chaves aleatórias de Bean (1994).

Çakirgil, Yucel, Kuyzu (2020) propuseram um modelo de programação inteira mista para encontrar soluções Pareto. Pela definição dos autores:

- uma solução s_1 domina outra solução s_2 se s_1 for pelo menos tão boa quanto s_2 em todos os objetivos e estritamente melhor que s_2 em pelo menos um objetivo.
- uma solução s é ótima ou eficiente de Pareto se não existir outra solução que domine s .

Neste estudo, objetivou-se minimizar o tempo de conclusão das tarefas prioritárias e o custo operacional total, isto é, o custo de viagem e de falta, ou seja, aquele gerado pelas tarefas não completadas dentro do limite de tempo. Além disso, cada tarefa requeria distintas habilidades e técnicos com capacidades diferentes poderiam ser agrupados em equipes para a realização dos serviços. O estudo baseou-se em um problema enfrentado por uma empresa turca de distribuição de energia.

Mathlouthi, Gendreau e Potvin (2021) apresentaram um algoritmo *branch-and-price* aplicado ao TRSP descrito por uma empresa que realiza manutenção e reparo de equipamentos eletrônicos. Nessa abordagem, foi levado em consideração os horários de descanso dos técnicos e as peças necessárias para a realização das tarefas. Cortés *et al.* (2014) também propuseram um algoritmo *branch-and-price*, aplicado em uma empresa de reparos a máquinas de escritórios. Neste estudo, os técnicos apresentavam habilidades homogêneas, ou seja, eram capazes de realizar as mesmas tarefas. Além disso, o problema considerava janelas de tempo suaves, em inglês *soft time-windows*, isto é, era permitido a realização de tarefas fora do horário de trabalho, mas que acarretava em penalidade na função objetivo. Assim, esta consistia no custo total, composto pelo tempo de viagem dos técnicos e das violações da janela de tempo.

Por fim, o estudo permitia que tarefas menos prioritárias fossem adiadas ao dia seguinte, no qual a prioridade aumentava.

Enquanto Tang et. al (2006) e Hashimoto *et al.* (2011) consideraram diretamente a prioridade das tarefas, Tsang e Voudouris (1997), Kovacs *et al.* (2012), Pillac *et al.* (2012) e Cortés *et al.* (2014) focaram nas janelas de tempo disponíveis para a realização das tarefas. Xu e Chiu (2001) e Damm *et al.* (2016) abordaram, simultaneamente, ambas as restrições.

Pourjavad e Almehdawe (2022) abordaram o FTSP levando em consideração duas novas variáveis: *overnight* e pausas para o almoço. A primeira consiste na possibilidade de, ao final de um dia de trabalho, os técnicos não retornarem a uma única base. Ou seja, é fornecido um conjunto de acomodações, nas quais os técnicos podem passar a noite. Dessa forma, os técnicos devem sair e chegar à base original somente no primeiro e último dia, respectivamente. Além disso, cada técnico possui uma pausa para almoço por dia, que ocorre no mesmo local de uma tarefa, após a conclusão desta e respeita o horário de início da tarefa seguinte.

Yahiaoui, Afifi, Allaoui (2023) consideraram restrições relativas às peças de reposição e ferramentas necessárias para realizar cada tarefa. Caso os técnicos não possuíssem o material necessário, poderiam acessar um depósito central para reabastecer os equipamentos. Para obter solução ao problema, os autores propuseram uma versão aprimorada da meta-heurística Pesquisa Local Iterada Aprimorada, denominada em inglês como *Enhanced Iterated Local Search* (eILS) e que combina operadores de pesquisa local, heurísticas de remoção, um algoritmo de melhor inserção e mecanismo de intensificação/diversificação baseado em um conjunto de soluções de elite.

Recentemente, técnicas envolvendo inteligência artificial passaram a ser adotadas para abordar o problema. Pekel (2022) utilizou aprendizado profundo, em inglês *deep learning*, aplicado em um algoritmo híbrido. Este consiste em uma combinação entre o algoritmo de Adam e um operador de mudança de corpo, em inglês *body change operator* (BCO). O desempenho da solução combinada foi comparado ao desempenho individual dos algoritmos.

Pitakaso *et al.* (2024) desenvolveram um Sistema de Inteligência Artificial Múltipla, em inglês *Artificial Multiple Intelligence System* (AMIS) para a resolução de um TRSP com múltiplos recursos, sendo eles times de técnicos e times de abastecimento.

O quadro 3 sumariza as diferentes abordagens do FTSP na literatura existente. Cada uma é classificada em relação a:

- Habilidade dos técnicos: homogêneo (não há distinção entre as tarefas realizáveis por cada técnico) ou heterogêneo (há distinção);

- Janela de tempo: *hard* (janela de trabalho fixa) ou *soft* (possibilidade de trabalho fora do expediente, com penalidade na função objetivo);
- Horário de almoço: prevê ou não horário de almoço dos técnicos;
- Horizonte de planejamento: dia único, múltiplos dias (saem e voltam à base todos os dias), *overnight* (retornam à base apenas no último dia, desde que haja acomodação);
- Objetivos: distância, prioridade, custo, tempo ou outro, como número de técnicos necessários.

Em seguida, a seção 2.3.1 apresenta a modelagem matemática da abordagem apresentada neste artigo.

Quadro 3 – Comparação entre a literatura existente

Referência	Características							Objetivos				
	Habilidade dos técnicos		Janela de Tempo		Almoço Horizonte			Distância	Prioridade	Custo	Tempo	Outro
	Homogêneo	Heterogêneo	Soft	Hard		Dia único	Múltiplos dias					
Tsang e Voudouris	1997		•	•			•					•
Xu e Chiu	2001		•		•							
Tang et al.	2006	•			•			•		•		
Hashimoto et al.	2011		•		•			•		•		
Kovacs et al.	2012		•		•		•		•			•
Pillac et al.	2012		•				•					•
Cortés et al.	2014	•		•			•			•	•	•
Damm et al.	2016		•		•		•			•		•
Çakirgil et al.	2020		•		•		•			•		•
Mathlouthi	2021		•	•		•	•		•	•		•
Pourjavad	2022	•		•		•		•	•		•	•
Pekel	2022		•		•			•			•	
Yahiaoui et al.	2023		•		•		•					•
Pitakaso et al.	2024											
Esse artigo	2025		•		•		•			•		•

Fonte: Adaptado de Pourjavad e Almehdawe (2022)

2.3.1 Modelo matemático do FTSP

A equação 18 restringe a tarefa k a ser realizada pela equipe i somente caso essa possua as habilidades necessárias para realizá-la. Essa restrição é adicionada às etapas anteriores do problema.

Dessa forma, o algoritmo é decomposto em duas fases. Na primeira, o objetivo é minimizar o *makespan* das tarefas obrigatórias. Essa etapa é composta pelas equações 2 – 13, 18. A segunda visa maximizar a soma da prioridade das tarefas não obrigatórias realizadas, partindo da solução da anterior. Essa fase, referida como PCVRP, é composta pelas equações

4 – 6, 8 – 11, 13 – 18. Ambas respeitam as restrições relativas ao tempo disponível para realização das tarefas e as habilidades dos técnicos.

$$y_{ik} \leq h_{ik} \quad \forall (i \in N, k \in K) \mid i \neq 0 \quad (18)$$

2.4 VIZINHO MAIS PRÓXIMO (NN)

A heurística do Vizinho Mais Próximo, em inglês *Nearest Neighbor* (NN), consiste em um método clássico para a construção de rotas. De acordo com Rosenkrantz, Stearns e Lewis (1977), sua complexidade de tempo é proporcional a n^2 , em que n é o número de nós. Além disso, o algoritmo consiste na criação de uma rota passo a passo, adicionando à solução corrente o nó mais próximo e ainda não visitado do último inserido. Dessa forma, foi feita uma adaptação ao problema apresentado no artigo para obtenção de uma solução inicial ao VRP.

O algoritmo identifica as tarefas obrigatórias e, para cada equipe, adiciona à rota R_k a tarefa elegível ($\in E_k$) mais próxima da anterior, com início na base. Constitui o conjunto E_k as tarefas obrigatórias ainda não alocadas e realizáveis pela equipe. O algoritmo encerra quando o comprimento do conjunto de tarefas visitadas equivale ao de tarefas obrigatórias, ou seja, quando estas estiverem todas alocadas.

Algoritmo 1 Algoritmo de Inicialização do VRP (NN)

1. Inicialize o conjunto de tarefas visitadas $V \leftarrow \{0\}$
2. Identifique as tarefas obrigatórias $O = \{i \in N \mid i \neq 0 \wedge o[i] = 1\}$
3. **para** cada equipe $k \in K$ **faça**
4. defina $R_k \leftarrow [0]$
5. **fim para**
6. **enquanto** $|V| < |O|$ **faça**
7. **para** cada equipe $k \in K$ **faça**
8. identifique a última tarefa visitada: $i \leftarrow R_k[-1]$
9. encontre as tarefas elegíveis: $E_k = \{j \in O \mid j \notin V \wedge h_{jk} = 1\}$
10. **se** $E_k \neq \emptyset$ **então**
11. escolha a tarefa mais próxima de i
12. atualize R_k
13. **senão**
14. adicione o depósito 0 ao final de R_k
15. **fim se**
16. **fim para**
17. **fim enquanto**

Fonte: Adaptado de Rosenkrantz *et al.* (1977)

3 MÉTODO DE PESQUISA

Este capítulo caracteriza a abordagem de pesquisa a ser utilizada neste trabalho, além de descrever e detalhar os procedimentos de pesquisa adotados.

3.1 JUSTIFICATIVA DA ABORDAGEM E MÉTODO DE PESQUISA

Para responder à questão de pesquisa destacada no capítulo 1, este trabalho se utilizou de uma abordagem de pesquisa quantitativa. Segundo Ganga (2011), a abordagem quantitativa é predominantemente utilizada em fenômenos naturais, que se utilizam de métodos experimentais e objetivam estabelecer relações causais entre variáveis de pesquisa. Neste caso, a pesquisa aplicará métodos quantitativos para modelar o problema e definir as variáveis envolvidas, como a disponibilidade de técnicos, localização geográfica, e demanda de serviços.

Por meio de técnicas de otimização e simulação, será analisado como diferentes estratégias de agendamento impactam a eficiência operacional, permitindo identificar a melhor solução para maximizar o desempenho do sistema, reduzir custos e melhorar a qualidade do serviço prestado.

Dentre os métodos concernentes à abordagem de pesquisa quantitativa, esta pesquisa se utilizará da Modelagem. Esta é utilizada em situações em que, segundo Ganga (2011), busca-se representar um sistema por meio de um modelo, para embasar o processo de tomada de decisão. Isso porque proporciona uma descrição simplificada das complexidades e incertezas do problema por meio de uma estrutura lógica. Especificamente, trata-se de uma modelagem axiomática prescritiva, que, segundo o mesmo autor, busca encontrar uma solução ótima para um problema de pesquisa ou comparar várias estratégias na solução de um problema específico.

Na seção a seguir, detalharemos as etapas e procedimentos metodológicos da modelagem adotada neste trabalho.

3.2 MODELAGEM

Segundo Taha (2007), as principais fases de implementação de uma modelagem de pesquisa operacional incluem:

1. Definição do problema
2. Construção do modelo
3. Solução do modelo

4. Validação do modelo
5. Implementação da solução

A definição do problema envolve definir o escopo do problema sob investigação. O objetivo é identificar os três elementos principais de um problema de decisão: descrição das alternativas de decisão, determinação do objetivo do estudo e especificação das restrições sob as quais o sistema modelado funciona.

A construção do modelo, por sua vez, envolve uma tentativa de traduzir a definição do problema em relações matemáticas. Se o problema se adequar a um dos métodos matemáticos convencionais, como a programação linear, pode-se utilizar uma ou mais soluções analíticas algorítmicas disponíveis. Alternativamente, se as relações matemáticas forem complexas demais para permitir a determinação de uma solução analítica, é possível ajustar o modelo e utilizar uma abordagem heurística ou considerar o uso de simulação. Por fim, as abordagens de simulação e heurísticas podem ser combinadas para solucionar o problema de decisão.

Além disso, a solução do modelo baseia-se na aplicação de algoritmos de otimização. Aplica-se uma análise de sensibilidade, que explora opções adicionais sobre o comportamento da solução ótima quando o modelo passa por alterações nos parâmetros. A análise de sensibilidade é especialmente necessária quando os parâmetros do modelo não podem ser estimados com precisão. Nesses casos, é essencial estimar a confiança na solução ótima através da variação dos parâmetros do problema.

Em seguida, a fase de validação do modelo garante que o que foi proposto funciona conforme previsto. Isto é, confirma se o modelo prevê de maneira adequada o comportamento do que está sendo estudado. Para isso, compara-se os resultados do modelo com a experiência prática. Assim, se a solução modelada não apresentar 'surpresas', é considerada validada. Em outras palavras, as formulações são testadas para assegurar que não haja erros. Nessa fase, o processo envolve testar rigorosamente o modelo com dados passados, para determinar se pode prever razoavelmente os comportamentos anteriores. Isso porque, de maneira geral, um modelo que apresenta bom desempenho com dados passados tende a ter um comportamento aceitável no futuro. Além disso, os modelos, em geral, não se comportam de forma confiável em eventos novos ou não testados. Por fim, se o modelo proposto representa um sistema novo, ainda não existente, há uma falta de experiência que pode comprometer a validação do modelo como uma ferramenta independente para verificar os resultados das ações modeladas.

Por fim, a implementação da solução implica na validação do modelo e tradução do mesmo em resultados ou instruções operacionais compreensíveis, que devem ser aceitas pelas pessoas que irão gerenciar o sistema.

3.3 PROCEDIMENTO

A modelagem apresentada por este artigo foi adaptada em relação à metodologia apresentada pela literatura. As etapas da pesquisa foram:

1. Implementação do modelo matemático: formulação do problema em termos matemáticos, isto é, definição de variáveis, restrições e função objetivo;
2. Validação do modelo: verificação da consistência do modelo por meio de testes com diferentes instâncias e da viabilidade das soluções obtidas;
3. Implementação do algoritmo: aplicação do método Vizinho Mais Próximo como solução inicial para a construção das rotas, permitindo uma abordagem heurística eficiente para a resolução do problema;
4. Validação do algoritmo: teste com diferentes instâncias e avaliação do desempenho da abordagem proposta, por meio da análise do tempo de execução, *gap*, número de soluções e valores de solução;
5. Escrita dos resultados: organização e análise dos resultados obtidos de forma sistemática, com uso de gráficos e tabelas para melhor visualização e interpretação dos dados;
6. Escrita da conclusão: avaliação do impacto da abordagem proposta, com destaque às contribuições, limitações e sugestões para melhorias futuras.

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Este capítulo apresenta os resultados da aplicação do algoritmo sobre as instâncias detalhadas a seguir.

4.1 INSTÂNCIAS DO PROBLEMA E DETALHES DA IMPLEMENTAÇÃO

Foram definidas 50 tarefas, sendo 25 obrigatórias e 25 não obrigatórias, localizadas na Região Metropolitana de São Paulo. Além disso, 15 instâncias para o problema, com distintos números de equipes disponíveis para a realização das tarefas, $|K| \in \{3,4,5,6,7\}$ e com a quantidade de tarefas realizáveis por equipe variando de acordo com a habilidade dos técnicos: $|h_i| \in \{[08,12]; [13,15]; [18,22]\}$. Ou seja, equipes sendo capazes de realizar de 8 a 12 (1), de 13 a 15 (2) e de 18 a 22 (3) tarefas de cada classificação, obrigatórias e não obrigatórias.

O procedimento descrito na seção 2 foi implementado no Python 3.10.11 com as bibliotecas CPLEX 22.1.1.0 e DOCPLEX 2.28.24. Cada operação foi limitada a 3.600 segundos.

4.2 ANÁLISE DOS RESULTADOS

A tabela 2 apresenta os resultados de cada instância em relação às soluções obtidas e desempenho do algoritmo, além dos parâmetros utilizados. Na tabela, são apresentados os seguintes dados:

- $|K|$: número de equipes disponíveis;
- $|h_i|$: número de tarefas realizáveis por equipe em cada fase;
- M_{NN} : *makespan* obtido pelo algoritmo de vizinho mais próximo;
- M : *makespan* do sistema após o VRP;
- N : número de soluções obtidas;
- Tempo: tempo de resolução do algoritmo;
- Status: status de resolução (101: solução ótima inteira; 102: solução ótima inteira com tolerância; 107: limite de tempo excedido);
- *Gap*: distância percentual entre solução encontrada e limitante superior (equação 19);
- Prêmio: soma de prioridades das tarefas realizadas.

Tabela 2 - Apresentação dos resultados

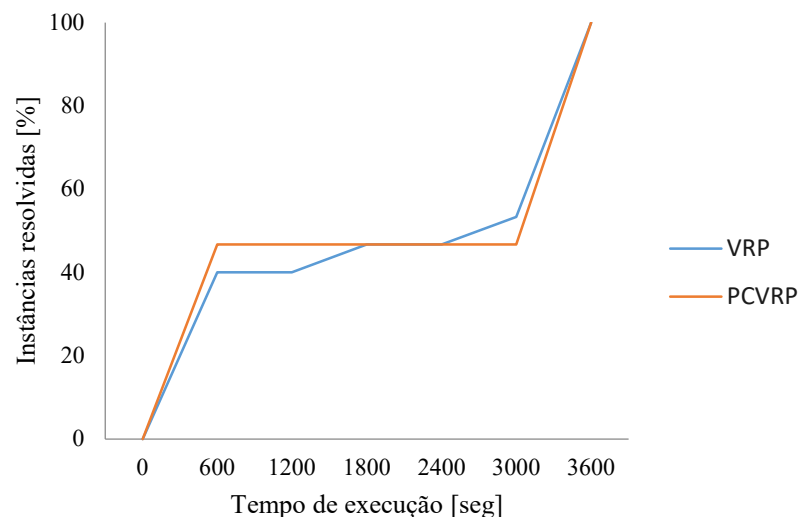
Instância			VRP							PCVRP			
#	K	h _i	M NN [min]	M [min]	N	Tempo [s]	Status	Gap [%]	Prêmio	N	Tempo [s]	Status	Gap [%]
1	3	1	636,80	539,34	3	0,82	101	0	-	-	-	-	-
2	4	1	487,96	426,91	10	319,06	102	0	152	4	20,03	101	0
3	5	1	444,52	349,59	24	77,12	101	0	157	7	67,61	101	0
4	6	1	416,71	326,33	22	45,91	102	0	185	9	48,14	101	0
5	7	1	416,71	326,33	12	35,3	102	0	185	8	53,55	101	0
6	3	2	591,64	478,97	13	188,02	102	0	124	1	63,34	102	0
7	4	2	504,87	398,91	27	3600,17	107	2,21	160	4	132,15	102	0
8	5	2	444,85	339,63	12	1522,77	102	0	193	9	3605,14	107	35,90
9	6	2	435,10	314,64	19	3384,72	102	0	173	6	3600,17	107	35,77
10	7	2	406,5	302,63	18	2984,19	102	0	194	11	3600,24	107	32,67
11	3	3	543,24	434,78	53	3600,34	107	35,68	130	7	3600,51	107	29,45
12	4	3	534,41	351,41	35	3600,21	107	41,84	174	8	3602,28	107	34,96
13	5	3	457,41	330,60	22	3602,43	107	51,73	177	4	3601,15	107	24,29
14	6	3	393,85	314,74	21	3601,18	107	56,46	168	2	3600,12	107	37,50
15	7	3	422,74	313,15	12	3604,59	107	62,19	168	3	3601,21	107	37,50

$$Gap = \frac{\text{Limitante Superior} - \text{Valor da solução}}{\text{Limitante Superior}} \quad (19)$$

4.2.1 Análise de performance

A figura 6 consiste no gráfico de performance do algoritmo. Nota-se que, em 600 segundos, em 6 instâncias retornou-se solução ótima na fase do VRP e 7 no PCVRP.

Além disso, o VRP apresentou três soluções em tempo intermediário, isto é, entre 600 e 3.600 segundos. Enquanto isso, no PCVRP, quando a solução ótima não foi obtida em 600 segundos, o algoritmo não retornou solução ótima dentro do limite de tempo.

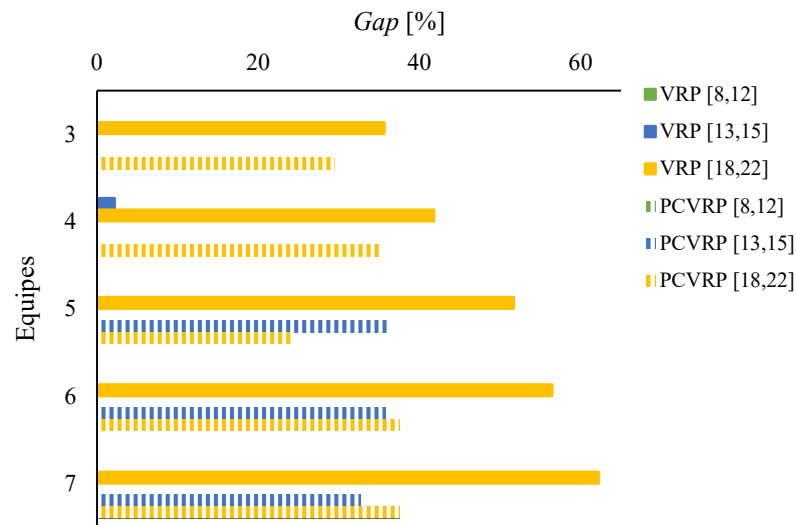
Figura 6 - Gráfico de Performance

A tabela 3 mostra que o VRP apresentou solução ótima em 9/15 instâncias, enquanto o PCVRP retornou em 7/15 instâncias.

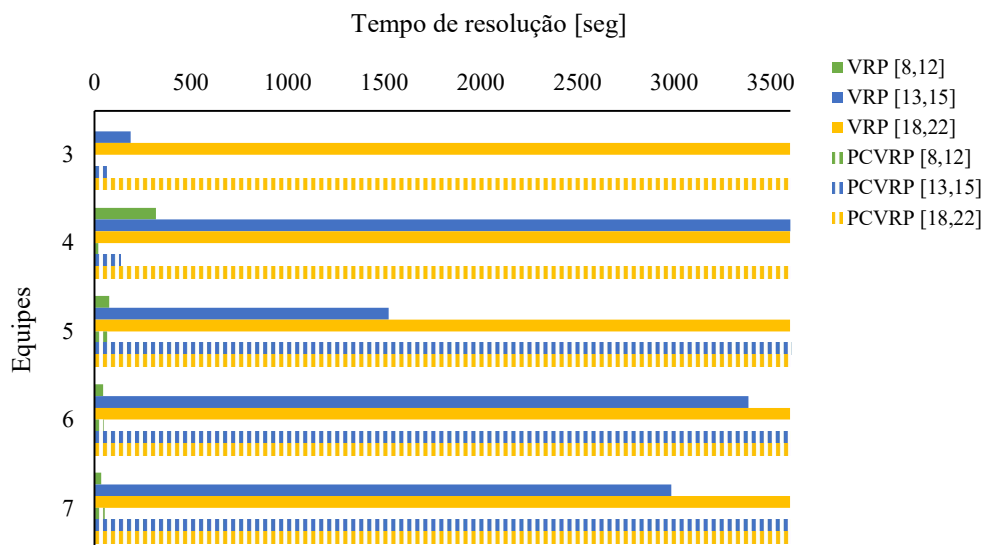
Tabela 3 - Status de solução para cada fase do problema, por número de equipes disponíveis

Equipes	Soluções Ótimas Encontradas		Limite de Tempo Atingido	
	VRP	PCVRP	VRP	PCVRP
3	2	2	1	1
4	1	2	2	1
5	2	1	1	2
6	2	1	1	2
7	2	1	1	2
Total	9	7	6	8

A figura 7 apresenta os valores de *gap* para cada instância, descrito pela fórmula 19. O *gap* foi zero, ou seja, o algoritmo alcançou solução ótima, em todas as instâncias em que o número de tarefas realizáveis foi de 8 a 12, para ambas as fases. Além disso, o desempenho do algoritmo também foi satisfatório ao aumentar o número de tarefas realizáveis para 13 a 15 na fase do VRP. Com esses parâmetros, exceto em um caso ($|K| = 4$, $gap = 2,21\%$), o *gap* foi zero. Porém, na segunda fase do problema, com número de equipes maior que 4, foram obtidos *gap*'s entre 32% e 36%, significativamente altos em relação às soluções ótimas nas instâncias anteriores. Por fim, com mais tarefas realizáveis por equipe, os *gap*'s na fase do PCVRP variaram de 29% a 38%, com uma tendência crescente, exceto em $|K| = 5$, em que o *gap* foi menor do que em relação às instancias anteriores. No VRP, o *gap* aumentou linearmente conforme o aumento do número de equipes, variando aproximadamente de 35% a 63%.

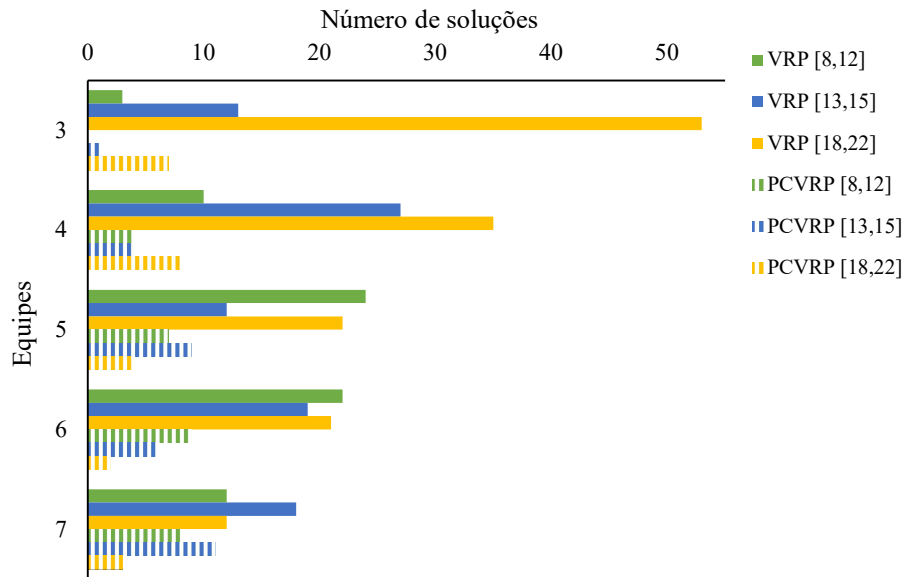
Figura 7 – Gap's

Além disso, conforme a figura 8, o tempo de resolução foi baixo para todas as instâncias em que as equipes eram capazes de realizar de 8 a 12 tarefas. Enquanto isso, com as habilidades variando de 18 a 22 tarefas, não retornou solução ótima para nenhuma instância. Por fim, para a quantidade de tarefas realizáveis intermediária, o VRP teve um tempo curto de solução apenas para $|K| = 3$. Com mais equipes, o tempo foi significativamente maior, porém apenas ultrapassou o limite de tempo em $|K| = 4$. O PCVRP, por sua vez, teve curtos tempos de resolução para os dois menores números de equipes, mas ultrapassou o limite determinado nas instâncias seguintes.

Figura 8 – Tempos de resolução

Por fim, a figura 9 apresenta o número de soluções obtidas para cada instância. Nota-se que houve maior número de soluções nos algoritmos VRP do que na segunda fase do problema. Fatores que podem ter contribuído com esse resultado foram a solução inicial provida na primeira fase e a menor restrição de tempo para a alocação das tarefas.

Figura 9 – Número de soluções obtidas



4.2.2 Análise das soluções encontradas

As figuras 10 e 11 apresentam os valores de *makespan* obtidos na fase inicial do problema. Estão representadas as soluções geradas pelo algoritmo do vizinho mais próximo e pelo algoritmo VRP, cuja solução inicial, no CPLEX *warm start*, foi fornecida pelo primeiro. Para todas as instâncias, houve redução significativa nos valores das soluções do VRP em relação ao NN.

Nota-se que, como esperado, quanto maior a quantidade de tarefas realizáveis por equipe, menor tende a ser o tempo de conclusão destas. Para $|K| = 6$ e $|K| = 7$, entretanto, o valor da solução para equipes com habilidade entre 13 e 15 tarefas foi o menor. Além disso, a diferença entre os tempos foi mais significativa com menos equipes disponíveis. Ou seja, comparar as soluções de acordo com o número de equipes e número de tarefas realizáveis, separadamente, com aumento do número de equipes, houve redução da diferença das soluções.

Adicionalmente, o comportamento da solução fornecida pelo vizinho mais próximo diminui conforme o aumento do número de equipes. Porém, o aumento da habilidade das equipes não resultou, necessariamente, na redução do *makespan* do sistema.

Por fim, não é possível concluir a influência das variáveis testadas nos prêmios coletados na segunda etapa do algoritmo, dada a natureza aleatória aos valores de prioridade de tarefas a cada instância.

Figura 10 – Soluções geradas pelo vizinho mais próximo

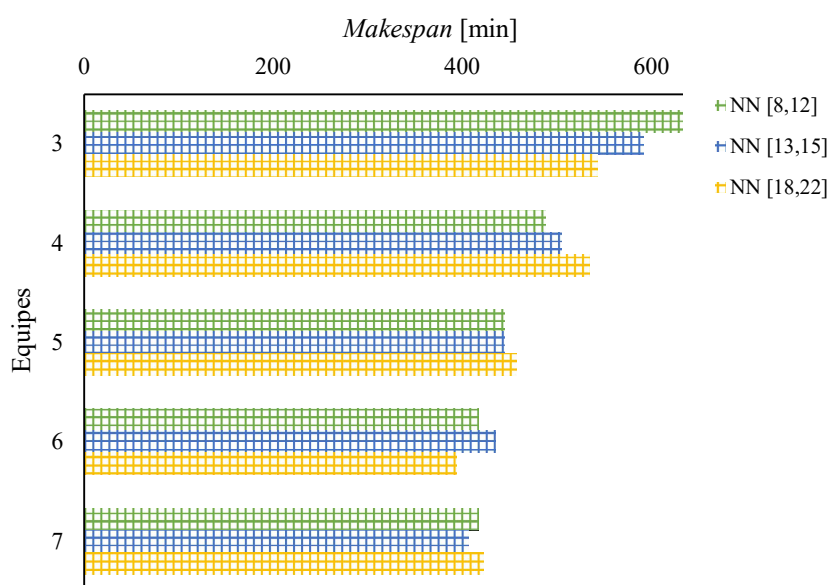
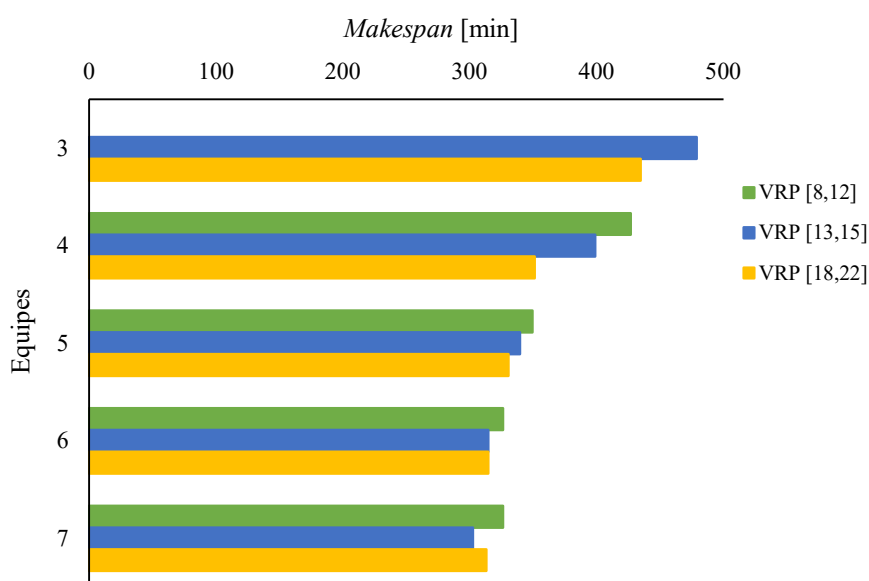


Figura 11 – Soluções geradas pelo algoritmo na fase VRP



5 CONCLUSÃO

O artigo apresentou uma heurística de PLIM ao problema de agendamento de técnicos de campo. Essa abordagem combina a modelagem matemática com técnicas heurísticas para encontrar soluções viáveis em tempo reduzido. Ao separar o problema em duas etapas distintas, é possível restringir o espaço de busca do algoritmo e obter melhores desempenhos.

O método do vizinho mais próximo demonstrou ser uma estratégia eficiente para a geração de soluções iniciais no algoritmo de agendamento de técnicos. Mesmo com um amplo espaço de busca, como nas instâncias 14 e 15, o algoritmo do VRP encontrou um número de soluções satisfatório em relação ao PCVRP. Além disso, para as instâncias 8, 9 e 10, o VRP retornou soluções ótimas dentro do limite de tempo, enquanto o PCVRP chegou ao limite de tempo com *gap's* altos. Assim, o método NN contribui com a economia de recursos computacionais, mantendo a qualidade das soluções obtidas.

Além disso, o desempenho do PCVRP com habilidades intermediárias (13 a 15) comportou-se de maneira distinta das demais. O algoritmo obteve solução ótima em curtos períodos de tempo para 3 e 4 equipes, mas ultrapassou o limite nas instâncias seguintes. Um dos fatores pode ser o resultado do VRP. Para $|K| = 4$, o *makespan* foi de 398,91, enquanto com $|K| = 5$ foi de 339,63. Com *makespan* menor, o tempo disponível para alocação de tarefas não obrigatórias é maior. Assim, aumenta-se o espaço de busca do algoritmo, o que justifica a queda no seu desempenho.

Assim, conclui-se que a variável mais relevante ao desempenho do algoritmo foi a habilidade dos técnicos em relação às tarefas. Desse modo, com essas restrições definidas, é possível aumentar o número de nós e de equipes e, ainda assim, obter resultados satisfatórios do algoritmo. Isso não é possível em propostas de VRP e PCVRP tradicionais, em que o desempenho depende do número de equipes e nós. Assim, as restrições relacionadas às habilidades artigo reduzem o espaço de busca e melhoram o desempenho do algoritmo, o que torna sua aplicabilidade mais flexível.

Por fim, é possível estabelecer uma relação entre a multifuncionalidade das equipes, o número de equipes e os resultados do algoritmo. A restrição relativa às habilidades dos técnicos é mais relevante aos cenários com menor número de equipes, uma vez que a limitação de recursos exige maior versatilidade dos técnicos para atender a uma variedade maior de tarefas. Em contrapartida, ao aumentar a disponibilidade de times, a escassez de recursos diminui, a ociosidade das equipes aumenta e há mais alternativas para a alocação das tarefas. Dessa forma, o resultado final torna-se menos influenciado pelo grau de multifuncionalidade das equipes.

5.1 LIMITAÇÕES DA PESQUISA

Uma das limitações da pesquisa foi a ausência de dados empíricos. Isto é, o modelo foi proposto considerando o cenário descrito pela empresa de distribuição de energia, mas não foi testado com valores reais. Assim, os resultados apresentados pelo artigo podem não refletir aspectos específicos dos dados da empresa.

Além disso, o artigo abordou algoritmos simples, cujos desempenhos não foram satisfatórios diante do aumento da multifuncionalidade das equipes.

5.2 PROPOSTA DE TRABALHOS FUTUROS

Como proposta de trabalhos futuros, sugere-se melhorar o algoritmo de inicialização. Quando a solução fornecida pelo vizinho mais próximo não respeitava o limite de tempo de 480 minutos, esta não era utilizada como solução inicial ao algoritmo. Além disso, sugere-se a implementação de um algoritmo de inicialização à fase do PCVRP. Uma alternativa seria normalizar as prioridades em valores iguais a um e, dessa forma, maximizar o número de nós não obrigatórios percorridos dentro do limite de tempo.

Adicionalmente, é possível complementar o problema com novas restrições. Entre elas, restrições em relação ao intervalo dentro do qual cada tarefa deve ser concluída. Por exemplo, em serviços de telecomunicação, os serviços são agendados de acordo com a disponibilidade do cliente. Além disso, abordagens de dias múltiplos permitem maior flexibilidade do algoritmo, especialmente quando as tarefas obrigatórias não puderem ser concluídas em um único dia.

Por fim, sugere-se a continuidade do estudo em relação à demanda por multifuncionalidade das equipes. Isto é, analisar o grau de multifuncionalidade necessário de acordo com a variação no número de equipes disponíveis e o impacto das alterações nos resultados obtidos pelo algoritmo.

REFERÊNCIAS

- ARENALES, M.; ARMENTANO, V. A.; MORABITO, R.; YANASSE, H. H. Pesquisa operacional. Rio de Janeiro: Campus/Elsevier, 2007. 523 p. ISBN 10-85-352-145-1454-2. <http://urlib.net/ibi/6qtX3pFwXQZGivnK2Y/Qwcqy>.
- ASSUNCAO, L., MATEUS, G.R. A cutting-plane algorithm for the Steiner team orienteering problem. *Computers & Industrial Engineering* 135, p. 922–939, 2019. <https://doi.org/10.1016/j.cie.2019.06.051>
- BEAN, JC. Generic algorithms and random keys for sequencing and optimization. *ORSA J Comput*, 6(2): 154-160, 1994.
- ÇAKIRGIL, S.; YÜCEL, E.; KUYZU, G. An integrated solution approach for multi-objective, multi-skill workforce scheduling and routing problems. *Computers & Operations Research*, v. 118, p. 104908, 2020.
- CASTILLO-SALAZAR, J.A.; LANDA-SILVA, D.; QU, R. Workforce scheduling and routing problems: Literature survey and computational study. *Annals of Operations Research*, 239(1), p. 39-67, 2016.
- CORTÉS, C. E.; GENDREAU, M.; ROUSSEAU, L. M.; *et al.* Branch-and-price and constraint programming for solving a real-life technician dispatching problem. *European Journal of Operational Research*, v. 238, n. 1, p. 300–312, 2014.
- DAMM, R. B.; RESENDE, M.G.C.; RONCONI, D.R. A Biased Random Key Genetic Algorithm for the Field Technician Scheduling Problem. *Computers & Operations Research*, vol. 75, pp. 49–63, 2016. <https://doi.org/10.1016/j.cor.2016.05.003>
- GANGA, G. M. D. Trabalho de Conclusão de Curso na Engenharia de Produção: um guia prático de conteúdo e forma. São Paulo: Atlas, 2012.
- HASHIMOTO, H.; BOUSSIER, S.; VASQUEZ, M.; *et al.* A GRASP-based approach for technicians and interventions scheduling for telecommunications. *Annals of Operations Research*, v. 183, n. 1, p. 143–161, 2011.
- KHALFAY, A.; CRISPIN, A.; CROCKETT, K. A review of technician and task scheduling problems, datasets and solution approaches. *Intelligent Systems Conference (IntelliSys)*. IEEE, pp. 288–296, 2017.
- KOVACS, A.; PARRAGH, S.N.; DOERNER, K.F.; HARTL, R.F. Adaptive large neighborhood search for service technician routing and scheduling problems. *J Sched* 15(5):579–600, 2012.
- LENSTRA, J.; RINNOOY, K.A. Complexity of vehicle routing and scheduling problems. *Networks*, 11, 221-227, 1981

- LI, K., TIAN, H. A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. *Applied Soft Computing* 43, p. 469–479, 2016. <https://doi.org/https://doi.org/10.1016/j.asoc.2016.02.040>
- MATHLOUTHI, I.; GENDREAU, M.; POTVIN, J. A metaheuristic based on tabu search for solving a technician routing and scheduling problem. *Computers & Operations Research*, v. 125, p. 105079, 2021.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM*, v. 7, n. 4, p. 326–329, 1960. <https://doi.org/10.1145/321043.321046>
- PEKEL, E. Deep Learning Approach to Technician Routing and Scheduling Problem. *Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ)*, v. 11, n. 2, p. 191–206, 2022.
- PILLAC, V.; GUÉRET, C.; MEDAGLIA, A. L. A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, v. 7, n. 7, p. 1525–1535, 2012.
- POURJAVAD, E.; ALMEHDAWE, E. Optimization of the technician routing and scheduling problem for a telecommunication industry. *Ann Oper Res* 315, 371–395, 2022. <https://doi.org/10.1007/s10479-022-04658-8>
- ROSENKRANTZ, D.J.; STEARNS, R.E.; LEWIS, P.M. An Analysis of Several Heuristics for the Traveling Salesman Problem, *SIAM Journal on Computing*, 6(3):563-581, 1977.
- TANG, H.; *et al.* Scheduling Technicians for Planned Maintenance of Geographically Distributed Equipment. *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 5, pp. 591–609, 2007. <https://doi.org/10.1016/j.tre.2006.03.004>.
- TAVARES, R.; *et al.* A MILP Heuristic to Address the Field Technician scheduling on a electric distribution maintenance sector. *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 2024.
- TAVARES, R.; NOGUEIRA, A.; BATALHA, M.O.; *et al.* Solving a shareable-setup time prize collection VRP applied to an electrical maintenance sector. *OLA'2024 International Conference on Optimization and Learning*, 2024.
- TOTH, P.; VIGO, D. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 487 – 512, 2002.
- TSANG, E.; VOUDOURIS, C; Fast local search and guided local search and their application to British telecom's workforce scheduling problem. *Oper Res Lett*, 20(3): 119-27, 1997.
- YAHIAOUI, A.; *et al.* Enhanced Iterated Local Search for the Technician Routing and Scheduling Problem. *Computers & Operations Research*, vol. 160, pp. 106385–106385, 2023. <https://doi.org/10.1016/j.cor.2023.106385>.