

Um heurística *Kernel Search* para o problema da orientação de equipes robusto com vértices de Steiner e prioridades decrescentes em função do tempo

Lucas Assunção

LITIS, ISEL, Université Le Havre Normandie
25 rue Philippe Lebon, 76600, Le Havre, França
lucas.assuncao-de-almeida@univ-lehavre.fr

Andréa Cynthia Santos

LITIS, ISEL, Université Le Havre Normandie
25 rue Philippe Lebon, 76600, Le Havre, França
andrea-cynthia.duhamel@univ-lehavre.fr

RESUMO

Apresentamos o Problema da Orientação de Equipes com vértices de Steiner e Prioridades Decrescentes para modelar operações de resgate em desastres. Os sítios que necessitam de comboio são subdivididos em obrigatórios e opcionais, sendo os últimos vinculados a valores de prioridades que diminuem com o tempo. O objetivo é encontrar rotas que maximizem a soma das prioridades coletadas, considerando tempos de serviço incertos e um limite de duração de rota. Adicionalmente, o número de sítios atendidos assumindo seus tempos de serviço máximos está sujeito a um *orçamento de incerteza*. Propomos uma formulação e a resolvemos via CPLEX e por meio de uma heurística *Kernel Search*. Os resultados mostram a dificuldade da resolução exata do problema e destacam a heurística proposta como uma alternativa promissora, capaz de encontrar, em média, em aproximadamente cinco minutos, soluções melhores ou iguais às obtidas pelo CPLEX (em até uma hora) para 83% das instâncias testadas.

PALAVRAS CHAVE. Logística humanitária, Modelo de otimização de rotas, Heurística.

L&T – Logística e Transportes, D&SP – PO em Defesa e Segurança Pública

ABSTRACT

We introduce the Robust Steiner Team Orienteering Problem with Decreasing Priorities to model disaster relief operations. Locations in need of rescue shuttles are sub-divided into mandatory and optional, being the latter linked to priorities that decrease over time. We aim to find routes that maximize the priorities collected, while considering uncertain service times and a route duration limit. Additionally, the number of serviced locations that assume their maximum service times is subject to a robust *uncertainty budget*. We propose a model for the problem and solve it via CPLEX and a *Kernel Search* heuristic. The results suggest the difficulty of the problem to be solved to optimality and highlight the proposed heuristic as a promising alternative, being able to find, within five minutes of average time execution, solutions better than or equal to the ones found by CPLEX (in up to one hour) for 83% of the instances tested.

KEYWORDS. Humanitarian logistics, Route optimization model, Heuristic.

Logistics and Transportation, OR in Defense and Public Security

1. Introdução

Operações de auxílio pós-desastres e estudos de gerenciamento de riscos têm recebido atenção significativa ao longo da última década, com o objetivo de aumentar a resiliência nos ambientes sociais e de trabalho por meio de prevenção e resposta sistemática a eventos de emergência [Baffo et al., 2017; Duhamel et al., 2016; Zheng et al., 2015]. Nesse contexto, a Pesquisa Operacional desempenha um papel fundamental na criação de métodos matemáticos para avaliar dados, orientar decisões e automatizar processos. Nas operações de auxílio em desastres e no gerenciamento de riscos (assim como em outras operações do “mundo real”), demandas, custos e disponibilidade de recursos podem variar de acordo com as circunstâncias, enquanto a qualidade das decisões tomadas é diretamente impactada por mudanças no cenário atual. Assim, considerar a incerteza dos dados é crucial para desenvolver soluções mais precisas e robustas.

Diferentes abordagens podem ser aplicadas para modelar problemas com dados incertos, como programação estocástica e Otimização Robusta (OR). Na primeira, a incerteza é modelada por meio de uma descrição probabilística e é principalmente aplicada quando que a lei de probabilidade associada aos dados incertos é conhecida *a priori*. Por outro lado, OR tem recebido atenção especial nas últimas décadas devido à sua capacidade de modelar incertezas aplicando o ferramental padrão da otimização clássica/determinística. Essa característica é especialmente vantajosa para modelar situações de auxílio em desastres, nas quais minimizar o tempo de resposta das soluções é crucial, sendo, em alguns casos, mais importante do que a otimalidade dos planos em si. Não surpreendentemente, uma tendência recente é aplicar a OR para modelar situações de crise, como decidir onde posicionar hospitais de campanha após desastres em larga escala (por exemplo, os terremotos que atingiram Katmandu, Nepal, em abril de 2015) [Coco et al., 2022], bem como a pré-posicionamento de recursos de auxílio para mitigar a escassez ou o excesso de itens utilizados [Wang et al., 2023].

Neste trabalho, introduzimos uma nova versão de OR do Problema da Orientação de Equipes com Prioridades Decrescentes (POE-PD) [Murat Afsar e Labadie, 2013], a saber, o Problema da Orientação de Equipes Robusto com vértices de Steiner e Prioridades Decrescentes (POERS-PD). No POE-PD determinístico, um conjunto de sítios passíveis de visita é fornecido, sendo que cada um é vinculado a um valor de prioridade que diminui linearmente ao longo do tempo. Assim, qualquer rota viável deve satisfazer as restrições de prazo máximo de visita dos sítios (ou seja, todos os sítios selecionados devem ser atendidos antes que suas prioridades atinjam zero), bem como um limite de tempo total de rota. O objetivo é maximizar a soma total de prioridades incorridas ao visitar os sítios, levando em conta sua diminuição ao longo do tempo.

No POERS-PD, a variação que abordamos neste trabalho, consideramos adicionalmente (i) sítios com prioridades máximas, que devem ser necessariamente visitados em alguma rota da solução, e (ii) tempos de atendimento incertos na visita dos sítios. Precisamente, tratamos os tempos de serviço como variáveis aleatórias independentes que variam de acordo com valores nominais e desvios máximos (pior caso). Além disso, um *orçamento de incerteza* [Bertsimas e Sim, 2004] é aplicado para limitar o número de sítios cujos tempos de serviço desviam para o cenário de pior caso em cada rota. Esse critério de OR foi escolhido para modelar a incerteza dos dados neste trabalho devido ao seu potencial de balancear o conservadorismo das soluções (ajustando o valor do orçamento), enquanto ainda evitando a não linearidade/intratabilidade que outros critérios (por exemplo, a *incerteza elipsoidal*) podem trazer [Bertsimas e Sim, 2004]. Essa mesma abordagem foi adotada para modelar a incerteza na versão de POERS-PD sem sítios obrigatórios, conhecido por Problema da Orientação de Equipes Robusto com Prioridades Decrescentes (POER-PD) [Yu et al., 2022].

POERS-PD é capaz de modelar operações de resgate e recuperação em emergências causadas por desastres naturais e incidentes em sítios industriais. Por exemplo, considere que, dentro de

uma região afetada por um desastre, vários locais estão potencialmente necessitando de comboios de resgate ou de suprimentos (como alimentos, água e medicamentos). Nesse cenário, POERS-PD pode ser usado para desenvolver rotas de resgate/fornecimento — usando veículos terrestres — para visitar tendas de primeiros socorros e/ou outros locais selecionados onde possa haver vítimas, levando em consideração limitações nos recursos disponíveis (por exemplo, autonomia restrita dos veículos, quantidade de recursos de auxílio, etc). Nessas situações, prioridade deve ser dada aos locais que são mais propensos a ter sobreviventes por perto, considerando também que as chances de encontrar vítimas ainda vivas dependem diretamente de uma resposta rápida, diminuindo ao longo do tempo. Valores de prioridade podem ser estimados com base em fatores como densidade populacional ou dados históricos relacionados a ocorrências semelhantes (no caso de inundações ou deslizamentos de terra, por exemplo) na região considerada ou até mesmo pela existência de pontos de encontro pré-estabelecidos em caso de incidentes em sítios industriais. Locais pré-selecionados para tendas de primeiros socorros, onde se sabe haver vítimas, são definidos como pontos de visita obrigatórios nas rotas de solução. Além disso, espera-se que o tempo de atendimento em cada local seja incerto, devido ao número variante de vítimas e/ou aos tempos de espera incertos causados por filas, por exemplo.

POERS-PD também generaliza a variação do Problema da Orientação de Equipes (POE) [Chao et al., 1996] com vértices obrigatórios, nomeado Problema da Orientação de Equipes com vértices de Steiner (POES) [Assunção e Mateus, 2019; Letchford et al., 2013]. POES (e, por consequência, POERS-PD) é particularmente desafiador, pois simplesmente encontrar uma solução viável para o problema (ou provar sua inviabilidade) é NP-difícil [Assunção e Mateus, 2019]. Essa dificuldade em gerar uma solução inicial viável adiciona um nível extra de complexidade ao problema, o que é particularmente problemática em aplicações de auxílio em desastres, onde a velocidade de resposta desempenha um papel crucial.

Até onde sabemos, POERS-PD não foi abordado na literatura anteriormente, exceto por seu caso específico, sem vértices obrigatórios (POER-PD) [Yu et al., 2022]. No trabalho, os autores propõem uma formulação compacta e outra baseada em rotas (com um número exponencial de variáveis), que é resolvida por meio de um algoritmo de *branch-and-price*. Eles também apresentam uma heurística de *busca tabu* para resolver instâncias de grande porte com até 100 vértices. Uma versão de OR do POE também foi proposta para modelar o planejamento de coleta de informações por Veículos Aéreos Não Tripulados (VANT) em missões militares [Evers et al., 2014a]. No trabalho, a incerteza é considerada tanto no consumo de recursos das rotas quanto nas prioridades dos locais a serem visitados. Os autores propõem uma formulação matemática e resolvem instâncias com até 20 vértices diretamente por meio de um resolvidor matemático comercial.

Várias outras variações não-determinísticas do POE já foram estudadas, mas modelando a incerteza por meio de programação estocástica. Nesses problemas, a incerteza é atribuída aos valores de prioridade [Angelelli et al., 2017; Song et al., 2020; İlhan et al., 2008], a parâmetros relacionados com tempo (de viagem, de serviço, etc) [Campbell et al., 2011; Evers et al., 2014b; Tang e Miller-Hooks, 2005], ou a ambos [Jin e Thomas, 2019]. Indicamos [Gunawan et al., 2016; Yu et al., 2022] para mais detalhes sobre variações clássicas e não-determinísticas do POE.

Neste trabalho, propomos uma formulação compacta para o POERS-PD, que incorpora a restrição de vértices obrigatórios ao modelo de fluxo de [Yu et al., 2022] para o POER-PD. Adicionalmente, resolvemos a formulação diretamente com CPLEX e também a partir de uma mat-heurística *Kernel Search*. Resultados computacionais sugerem a dificuldade de resolução exata do problema mesmo para instâncias relativamente pequenas (de 25 e 50 vértices). Em particular, o CPLEX é capaz de resolver na otimalidade apenas 5 das 72 instâncias testadas, e o *gap* médio das soluções obtidas é elevado (69.38%). Por outro lado, a mat-heurística proposta surge como uma

alternativa promissora, ao oferecer limites primais tão bons quanto ou melhores que os do CPLEX em 60 das 72 instâncias testadas, com uma melhoria percentual média de limites primais de 4.85% e tempo médio de execução reduzido (aproximadamente cinco minutos).

O restante deste estudo está organizado da seguinte forma. A definição formal de POERS-PD é apresentada na Seção 2, seguida da formulação matemática proposta (Seção 3) e da heurística *Kernel Search* implementada (Seção 4). A Seção 5 é dedicada aos resultados computacionais, e conclusões e ideias de trabalhos futuros são discutidas na última seção.

2. Definição do problema e notação

POERS-PD é definido em um digrafo $G = (N, A)$, onde N e A são os conjuntos de vértices e de arcos, respectivamente. Considere a origem $0 \in N$, o conjunto de vértices *obrigatórios* $S \subseteq N \setminus \{0\}$ e o conjunto de vértices *opcionais* $P \subseteq N \setminus \{0\}$, tal que $S \cap P = \emptyset$ e $N = S \cup P \cup \{0\}$. Um valor de prioridade $p_i \in \mathbb{Z}^+$ é associada a cada vértice $i \in P$, e um tempo de travessia $t_{ij} \in \mathbb{R}^+$ é atribuído a cada arco $(i, j) \in A$. É dado um limite de tempo máximo de rota T , e se supõe que os tempos de travessia dos arcos satisfazem a desigualdade triangular. Supõe-se ainda que cada prioridade $p_i, i \in P$, diminui linearmente ao longo do tempo, com uma taxa de decaimento d_i . Assim, um vértice só pode ser visitado enquanto sua prioridade existir, ou seja, antes de seu prazo final de visitaç o, dado por $D_i = p_i/d_i$ para todo $i \in P$. $D_i = T$ para todo $i \in S \cup \{0\}$.

Para cada vértice $i \in N$, o tempo de serviço nominal é dado por $\bar{s}_i$, e $\hat{s}_i$ é sua variação de pior-caso. Naturalmente, $\bar{s}_0 = \hat{s}_0 = 0$. Uma frota homogênea F é disponibilizada, sendo que cada veículo pode trabalhar por, no máximo, o limite de tempo T . POERS-PD visa encontrar até $|F|$ rotas que começam e terminam no mesmo vértice 0, de modo que todo vértice obrigatório em S pertença a exatamente uma rota e a soma total das prioridades dos vértices opcionais visitados seja maximizada. Aqui, cada vértice opcional em P também pode ser visitado por no máximo um veículo. Em POERS-PD, o espaço de soluções viáveis está sujeito a incerteza, e uma solução para o problema deve respeitar um *orçamento de incerteza* aplicado aos tempos (incertos) de serviço nos vértices, conforme detalhado a seguir.

2.1. Caracterização da incerteza

Aqui, os tempos de serviço incertos nos vértices são tratados como variáveis aleatórias independentes que assumem valores no intervalo

$$\tilde{s}_i = \bar{s}_i + \xi_i \hat{s}_i, \xi_i \in [-1, 1], \forall i \in N \setminus \{0\}, \quad (1)$$

onde ξ_i , para todo $i \in N \setminus \{0\}$, são parâmetros usados para simular a incerteza. Uma vez que o cenário de pior-caso para os tempos de serviço ocorre apenas para desvios positivos dos valores nominais, assumimos $\xi \in [0, 1]$ por simplificação. Como em [Yu et al., 2022], aplicamos a *incerteza orçamentada* [Bertsimas e Sim, 2004] como critério de OR responsável por controlar a aleatoriedade associada a $\tilde{s}$. Assim, o *conjunto de incerteza orçamentada* é definido como

$$\chi = \left\{ \tilde{s} \in \mathbb{R}^{|N \setminus \{0\}|} : \tilde{s} = \bar{s}_i + \xi_i \hat{s}_i, \xi_i \in [0, 1], \forall i \in N \setminus \{0\}, \sum_{i \in N \setminus \{0\}} \xi_i \leq \Gamma \right\}, \quad (2)$$

onde $\Gamma \in \mathbb{Z}$, $0 \leq \Gamma \leq |N \setminus \{0\}|$, é um parâmetro chamado *orçamento de incerteza*, que determina o nível de conservadorismo da solução, limitando o número de vértices cujos tempos de serviço atingem o pior cenário $\bar{s} + \hat{s}$ em uma rota. Note que, quando $\Gamma = 0$, a incerteza é completamente eliminada, e o problema determinístico correspondente surge, com $\tilde{s} = \bar{s}$.

A caracterização da incerteza descrita acima define implicitamente o conjunto de possíveis durações de rotas (não necessariamente viáveis) para o POERS-PD. Para modelar essa abstração de

forma compacta, considere as variáveis de decisão contínuas $a_{i\gamma}$, que indicam, para todo $i \in N$, $\gamma \in \{0, \dots, \Gamma\}$, o tempo de chegada em um vértice i quando até γ vértices predecessores (na mesma rota) atingiram seus tempos de serviço de pior cenário. Tais tempos de chegada podem ser calculados eficientemente por meio de um conjunto de equações recursivas baseadas em programação dinâmica [Agra et al., 2013; Lee e Lee, 2012; Munari et al., 2019; Yu et al., 2022] da seguinte forma: dada uma rota $r = (v_0 = 0, v_1, \dots, v_m, v_{m+1} = 0)$ visitando m vértices de $N \setminus \{0\}$,

$$a_{v_i\gamma} = \begin{cases} 0, & i = 0, 0 \leq \gamma \leq \Gamma, & (3a) \\ t_{0v_i}, & i = 1, 0 \leq \gamma \leq \Gamma, & (3b) \\ a_{v_{i-1}\gamma} + \bar{s}_{v_{i-1}} + t_{v_{i-1}v_i}, & i > 1, \gamma = 0, & (3c) \\ \max \left\{ a_{v_{i-1}\gamma} + \bar{s}_{v_{i-1}}, a_{v_{i-1}\gamma-1} + \bar{s}_{v_{i-1}} + \hat{s}_{v_{i-1}} \right\} + t_{v_{i-1}v_i}, & i > 1, 0 < \gamma \leq \Gamma. & (3d) \end{cases}$$

As equações (3a) e (3b) estabelecem o tempo inicial (na origem) como zero e garantem que o tempo de chegada no primeiro vértice da rota corresponde ao tempo necessário para percorrer o arco da origem até ele. Equações (3c) definem os tempos de chegada nos vértices v_i , $i > 1$, quando nenhum vértice na rota atinge o pior cenário (ou seja, $\gamma = 0$). Nesse caso, $a_{v_i\gamma} = a_{v_i0}$ é calculado simplesmente adicionando a $a_{v_{i-1}0}$ — o tempo de chegada no vértice predecessor v_{i-1} — o tempo de serviço nominal de v_{i-1} , mais o tempo de travessia do arco (v_{i-1}, v_i) . Quando pelo menos um vértice tem permissão para desviar dos valores nominais (ou seja, $0 < \gamma \leq \Gamma$), o tempo de chegada em um vértice v_i , $i > 1$, é dado pelo valor máximo dentre duas situações: (i) γ predecessores já atingiram seus tempos de serviço máximos antes de chegar a v_{i-1} , ou (ii) até $\gamma - 1$ vértices na rota alcançaram seus cenários de pior caso antes de chegar a v_{i-1} , e, portanto, ainda há orçamento para incluir v_{i-1} em seu tempo de desvio máximo $\bar{s}_{v_{i-1}} + \hat{s}_{v_{i-1}}$. Essas duas situações são descritas em (3d).

Por fim, uma rota $r = (v_0 = 0, v_1, \dots, v_m, v_{m+1} = 0)$ é dita *robusto-viável* em relação à incerteza orçamentada se, quando até Γ vértices da rota estão no pior caso, os tempos de chegada em todos os vértices opcionais são, no máximo, seus prazos finais (ou seja, $a_{v_i\Gamma} \leq D_{v_i}$, para todo $1 \leq i \leq m$), e, obviamente, o limite de tempo total da rota for respeitado (ou seja, $a_{v_{m+1}\Gamma} \leq T$, $v_{m+1} = v_0 = 0$).

3. Formulação matemática

Nesta seção, descrevemos uma formulação compacta para o POERS-PD, obtida ao incorporar a restrição de vértices obrigatórios à formulação baseada em fluxo para o POER-PD [Yu et al., 2022], em que os valores de fluxo representam os tempos de chegada nos vértices. Considere as variáveis de decisão y sobre a escolha de vértices pertencentes ou não às rotas de solução, de modo que $y_i = 1$ se o vértice $i \in N \setminus \{0\}$ for visitado por um veículo da frota, e $y_i = 0$, caso contrário. Similarmente, as variáveis binárias x identificam as rotas da solução em si: $x_{ij} = 1$ se o arco $(i, j) \in A$ for percorrido na solução; $x_{ij} = 0$, caso contrário. Além disso, as variáveis contínuas $a_{i\gamma}$, para todo $i \in N$, $\gamma \in \{0, \dots, \Gamma\}$, representam o tempo em que um veículo chega a um vértice i quando no máximo γ tempos de serviço de predecessores (na mesma rota) alcançaram seus valores de pior caso. A variável de folga φ mantém o número de veículos que não são utilizados na solução. Dado um vértice $i \in N$, definimos os conjuntos de arcos saindo e entrando em i como $\delta^+(i) = \{j \in N : (i, j) \in A\}$ e $\delta^-(i) = \{j \in N : (j, i) \in A\}$, respectivamente. A formulação compacta é definida a seguir.

$$(\mathcal{F}_1) \quad \max \quad \sum_{j \in P} (p_j y_j - d_j a_{j\Gamma}), \quad (4)$$

$$s.t. \quad y_i = 1 \quad \forall i \in S \cup \{0\}, \quad (5)$$

$$\sum_{j \in \delta^+(i)} x_{ij} = y_i \quad \forall i \in S \cup P, \quad (6)$$

$$\sum_{j \in \delta^+(0)} x_{0j} + \varphi = |F|, \quad (7)$$

$$\sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = 0 \quad \forall i \in N, \quad (8)$$

$$a_{i\gamma} \geq t_{0i}y_i \quad \forall i \in \delta^+(0), \forall \gamma \in \{0, \dots, \Gamma\}, \quad (9)$$

$$a_{i\gamma} + \bar{s}_i + t_{ij} \leq a_{j\gamma} + M'_{ij}(1 - x_{ij}) \quad \forall i \in S \cup P, j \in \delta^+(i), \forall \gamma \in \{0, \dots, \Gamma\}, \quad (10)$$

$$a_{i\gamma-1} + \bar{s}_i + \hat{s}_i + t_{ij} \leq a_{j\gamma} + M''_{ij}(1 - x_{ij}) \quad \forall i \in S \cup P, j \in \delta^+(i), \forall \gamma \in \{1, \dots, \Gamma\}, \quad (11)$$

$$a_{i\gamma} \leq D_i y_i \quad \forall i \in N, \forall \gamma \in \{0, \dots, \Gamma\}, \quad (12)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, \quad (13)$$

$$y_i \in \{0, 1\} \quad \forall i \in N, \quad (14)$$

$$a_{i\gamma} \geq 0 \quad \forall i \in N, \forall \gamma \in \{0, \dots, \Gamma\}, \quad (15)$$

$$0 \leq \varphi \leq |F|. \quad (16)$$

A função objetivo em (4) consiste da soma das prioridades incorridas ao visitar vértices opcionais. Observe que a perda nos valores de prioridade (ao longo do tempo) ligada à taxa de decaimento é levada em conta. As restrições (5) impõem que todos os vértices obrigatórios sejam selecionados, enquanto as restrições (6) garantem que cada vértice em $S \cup P$ seja visitado no máximo uma vez. As restrições (7) garantem que no máximo $|F|$ veículos saiam da origem 0, enquanto as restrições (8), juntamente com as restrições (5) e (6), garantem que, se um veículo visitar um vértice $i \in N$, então ele deve entrar e sair desse vértice exatamente uma vez.

As restrições (9)-(11) representam as equações recursivas (3b)-(3d), que calculam os tempos de chegada nos vértices ao considerar o orçamento de incerteza Γ . Em particular, (9) estão relacionadas a (3b). Aqui, a igualdade é relaxada para simplificar o modelo sem perder sua correção. Observe ainda que (3a) não foram representadas, uma vez que são implicitamente inferidas por (9). As restrições (10)-(11) representam (3c) e (3d) por meio de restrições semelhantes às de Miller-Tucker-Zemlin [Miller et al., 1960], que são ativadas/desativadas por meio de constantes suficientemente grandes $M'_{ij} = D_i + \bar{s}_i + t_{ij}$ e $M''_{ij} = D_i + \bar{s}_i + \hat{s}_i + t_{ij}$ para todo $(i, j) \in A, i \neq 0$. No modelo, o gerenciamento do fluxo associado às variáveis a também evita a existência de sub-rotas nas soluções.

As restrições (12) forçam as rotas de solução a satisfazerem os prazos máximos de visita dos vértices (nos caso dos vértices opcionais) e a duração total da rota T . Essas restrições também vinculam as variáveis de fluxo a às variáveis y , garantindo que, se o tempo de chegada em um determinado vértice for diferente de zero (independentemente da quantidade de orçamento de incerteza usado), então o vértice é selecionado na solução. As restrições restantes definem o domínio das variáveis de decisão.

4. Mat-heurística Kernel Search

Kernel Search consiste em um *framework* heurístico baseado em modelo matemático (mat-heurística) originalmente concebida para o problema da mochila multidimensional [Angelelli et al., 2010] e posteriormente generalizada para resolver qualquer problema de programação linear inteira mista [Guastaroba et al., 2017]. A heurística baseia-se na ideia de que as variáveis de decisão podem ser ordenadas de acordo com a maior ou menor probabilidade de aparecerem em uma

solução ótima. Precisamente, dado um critério de prioridade aplicado às variáveis de decisão (como seus valores na solução da relaxação linear ou seus custos reduzidos correspondentes), as “melhores” variáveis são mantidas em um *kernel*, e as demais são distribuídas em baldes (ou “*buckets*”). Então, iterativamente, o problema original (inteiro) é resolvido considerando apenas as variáveis do kernel atual e as de um dado balde candidato, um de cada vez. Dessa forma, o kernel incorpora as variáveis do balde que compõem uma nova solução incumbente sempre que uma solução de melhor qualidade for encontrada na iteração.

Neste trabalho, aplicamos *Kernel Search* para resolver a formulação $\mathcal{F}_1$ considerando as variáveis y (relacionadas à seleção de vértices) como as candidatas a compor o kernel, conforme detalhado na Figura 1. Considere os seguintes parâmetros de entrada da heurística: tamanho máximo κ dos baldes, intervalo $[\delta_{min}, \delta_{max}]$ ao qual o tempo de execução de cada sub-problema está sujeito, e a taxa μ segundo a qual o limite de tempo de execução de cada sub-problema diminui linearmente. O algoritmo começa construindo o kernel inicial K e os baldes $\lambda \in \Lambda$ de variáveis candidatas (linhas 1-4). Para isso, a relaxação linear da formulação é resolvida, e as variáveis y são ordenadas de acordo com o seguinte critério. Sejam $\bar{y}_i$ e $rc(y_i)$, para todo $i \in N$, o valor de y_i na solução da relaxação linear e seu custo reduzido correspondente, respectivamente. A ordenação é definida de tal forma que os vértices de origem, assim como os obrigatórios, recebem maior prioridade, seguidos pelos vértices cujos valores $\bar{y}$ são diferentes de zero (em ordem não-crescente) e, por último, pelos vértices restantes, em ordem não-crescente de custos reduzidos. Os custos reduzidos (também em ordem não-crescente) são usados para decidir os casos de valores de relaxação empatados. Então, considerando essa ordenação, o kernel inicial K é preenchido com os primeiros $\max\{|S| + 1, \kappa\}$ vértices, enquanto os baldes são compostos pelos demais vértices, com até κ vértices por balde. Note que o tamanho do kernel inicial é definido de forma a garantir a disponibilidade da origem e dos vértices obrigatórios em qualquer sub-problema, evitando assim a inviabilidade de suas soluções (sempre que o problema original é viável).

Em seguida, o procedimento de busca local iterativo é executado (linhas 5-13). Seja $\mathcal{F}_1(N', \nu, \delta)$ o sub-problema definido quando apenas um subconjunto de vértices $N' \subseteq N$ são elegíveis em $\mathcal{F}_1$. Além disso, nesse caso, uma solução inicial viável ν é dada, e apenas δ segundos estão disponíveis para resolver o problema. Assim, a melhor solução corrente (e possivelmente viável) é definida como ν^* e inicializada resolvendo $\mathcal{F}_1(K, \emptyset, \delta_{max})$, o sub-problema restrito às variáveis iniciais no kernel (linhas 5-7). Nesse caso, nenhuma solução inicial é considerada, e o limite de tempo de execução é dado por $\delta \leftarrow \delta_{max}$ segundos. A busca prossegue resolvendo, para cada balde $\lambda \in \Lambda$ na sequência, um sub-problema $\mathcal{F}_1(K \cup \lambda, \nu^*, \delta)$ obtido adicionando temporariamente as variáveis do balde ao kernel atual e definindo a solução atual como ponto de partida. Em cada iteração, o limite de tempo de execução é diminuído linearmente de acordo com a taxa μ , até δ_{min} (linha 10), de modo a dedicar menos esforço computacional nos últimos baldes, que têm menos probabilidade de conter vértices de interesse. Quando uma nova solução de melhor qualidade é encontrada (ao final de uma iteração), os novos vértices (se houver) que compõem essa solução são incorporados ao kernel (linha 12) e considerados nos sub-problemas subsequentes. A melhor solução encontrada ν^* — ao se buscar em todos os baldes — é retornada pela heurística.

5. Experimentos computacionais

Nos experimentos, consideramos tanto a formulação compacta $\mathcal{F}_1$ (resolvida diretamente com um resolvidor comercial) como a heurística *Kernel Search* apresentada previamente. Todos os códigos foram implementados em C++, com o resolvidor matemático ILOG CPLEX 22.11¹. Os experimentos computacionais foram realizados em uma máquina AMD EPYC-Milan de 64 bits com 2,4 GHz e 100,0 GB de RAM, sob sistema operacional Linux. A máquina possui 16 núcleos

¹<https://www.ibm.com/analytics/cplex-optimizer>

Entrada: tamanho máximo κ dos baldes, intervalo de tempo $[\delta_{min}, \delta_{max}]$ e taxa de redução μ do tempo de execução dos sub-problemas.

Saída: uma solução heurística ν^* .

1. // *Passo I. Construir o kernel e os baldes iniciais.*
2. Resolver a relaxação linear de $\mathcal{F}_1$;
3. Ordenar as variáveis y priorizando a origem e os vértices obrigatórios, e, depois, os demais vértices em ordem não-crescente dos valores de relaxação e custos reduzidos;
4. Selecionar $\max\{|S| + 1, \kappa\}$ vértices para compor o kernel K e distribuir os vértices restantes em baldes $\lambda \in \Lambda$ com até κ vértices, segundo a ordenação;
5. // *Passo II. Busca no kernel inicial.*
6. $\delta \leftarrow \delta_{max}$;
7. $\nu^* \leftarrow \mathcal{F}_1(K, \emptyset, \delta)$;
8. // *Passo III. Busca nos baldes candidatos.*
9. **para todo** ($\lambda \in \Lambda$) **fazer**
10. $\delta \leftarrow \max\{\delta_{min}, \mu\delta\}$;
11. $\nu^* \leftarrow \mathcal{F}_1(K \cup \lambda, \nu^*, \delta)$;
12. $K \leftarrow K \cup \nu^*$;
13. **fim-para**;

Figura 1: Heurística *Kernel Search* usada para resolver a formulação $\mathcal{F}_1$.

físicos, com modo *hyper-threading* desabilitado. Quando resolvendo o modelo $\mathcal{F}_1$, CPLEX foi deixado executar por até uma hora em configurações padrão. Por sua vez, a heurística foi executada com parâmetros definidos como: $[\delta_{min}, \delta_{max}] = [19, 84]$, $\kappa = 2$, $\mu = 0,96$, e com CPLEX dando ênfase em viabilidade. A escolha dos parâmetros foi realizada a partir do configurador automático Irace [López-Ibáñez et al., 2016].

5.1. Benchmark de instâncias

Em nossos experimentos, adaptamos instâncias de POER-PD [Yu et al., 2022] de $|N \setminus 0| = \{25, 50\}$ vértices, selecionando aleatoriamente um subconjunto de $\lfloor 10\%|N \setminus 0| \rfloor$ vértices como obrigatórios. As instâncias são de três tipos, a saber, C , R e RC , em que os vértices estão, respectivamente, agrupados, distribuídos aleatoriamente e em uma mistura dos dois casos anteriores. Em todas elas, o primeiro vértice é considerado a origem 0, e cada vértice recebe uma coordenada de posição bidimensional e uma prioridade. Assim, o tempo de viagem de um arco é definido como a distância euclidiana entre os vértices correspondentes. O limite de tempo das rotas T assume os valores do primeiro e terceiro quartis do conjunto de todos os prazos de entrega dos vértices de uma determinada instância. Também consideramos um valor de T definido como 80% do primeiro quartil como forma de obter limites de duração das rotas ainda mais restritos. O número de veículos $|F|$ assume os valores em $\{2, 4\}$. Por fim, o desvio de tempo de serviço de pior caso de um vértice $i \in P$ é definido como $\hat{s}_i = 50\% \bar{s}_i$, e o orçamento de incerteza Γ assume valores em $\{0, 5\}$. No total, considerando todas as possíveis realizações de parâmetros mencionadas acima, utilizamos 72 instâncias em nossos experimentos.

5.2. Resultados

Inicialmente, resolvamos a formulação compacta $\mathcal{F}_1$ diretamente com CPLEX como forma de avaliar quão desafiador o problema é na prática, além de obter limites primais (e, possivelmente, certificados de otimalidade) para as instâncias testadas. Os limites obtidos também foram utilizados para valorar a qualidade das soluções obtidas pela heurística proposta. Os resultados são descritos na Tabela 1. Nela, os conjuntos de instâncias são nomeados como $\langle t \rangle \langle |N \setminus 0| \rangle Q \langle q \rangle$, com tipo de

instância $t \in \{C, R, RC\}$, e quartil dos prazos dos vértices $q \in \{0.8, 1, 3\}$. Como especificado anteriormente, o caso 0.8 representa 80% do valor do primeiro quartil.

Instâncias	$ N \setminus \{0\} $	$\mathcal{F}_1 + \text{CPLEX}$				<i>Kernel Search</i>			
		# Ótima/Total	Tempo (s)	Gap (%)		# Melhoria/Total	Tempo (s)	Melhoria (%)	
				Média	Desvio			Média	Desvio
C25Q0.8	25	0/4	3603.65	86.10	13.41	4/4	1.19	0.58	1.16
C25Q1	25	0/4	3603.95	72.04	17.95	4/4	97.62	0.00	0.00
C25Q3	25	0/4	3607.69	38.99	11.70	2/4	517.24	0.02	0.23
R25Q0.8	25	2/4	1897.19	40.41	47.22	4/4	0.85	0.00	0.00
R25Q1	25	2/4	1903.23	34.43	41.13	2/4	222.56	-2.71	5.03
R25Q3	25	0/4	3610.03	43.80	14.66	3/4	468.32	0.17	0.24
RC25Q0.8	25	0/4	3605.72	72.09	15.26	1/4	40.91	-4.51	5.58
RC25Q1	25	0/4	3606.31	62.02	17.16	1/4	252.65	-2.80	4.16
RC25Q3	25	0/4	3611.55	41.32	15.37	3/4	519.79	1.06	1.21
Sub-total		4/36	3227.70	54.58	28.20	24/36	235.68	-0.91	3.15
C50Q0.8	50	0/4	3603.88	97.11	4.19	4/4	6.64	4.05	8.10
C50Q1	50	0/4	3603.14	93.27	6.66	4/4	99.07	18.91	21.22
C50Q3	50	0/4	3602.22	67.22	15.66	4/4	1027.27	18.46	12.70
R50Q0.8	50	1/4	3359.24	73.57	49.12	4/4	0.97	0.00	0.00
R50Q1	50	0/4	3601.85	95.11	5.94	4/4	20.20	3.01	5.46
R50Q3	50	0/4	3603.43	69.64	16.02	4/4	1037.57	12.91	9.14
RC50Q0.8	50	0/4	3602.62	97.99	4.02	4/4	11.88	3.07	6.14
RC50Q1	50	0/4	3601.98	94.73	6.35	4/4	138.15	14.62	16.93
RC50Q3	50	0/4	3605.22	69.05	14.91	4/4	1029.50	20.49	17.11
Sub-total		1/36	3575.95	84.19	21.33	36/36	374.58	10.61	13.42
Total		5/72	3401.83	69.38	28.96	60/72	305.13	4.85	11.28

Tabela 1: Resultados obtidos na resolução da formulação $\mathcal{F}_1$ diretamente com resolvidor CPLEX (limite de 3600s de execução) e por meio da mat-heurística *Kernel Search* proposta.

A primeira e segunda colunas exibem o nome de cada conjunto de instâncias e o tamanho de suas instâncias, respectivamente. Em seguida, fornecemos quatro valores referentes à resolução da formulação compacta (colunas sob “ $\mathcal{F}_1 + \text{CPLEX}$ ”). O primeiro valor corresponde ao número de instâncias resolvidas na otimalidade (ou provadas como inviáveis). O segundo corresponde ao tempo médio de processamento (em segundos) gasto na resolução do conjunto de instâncias. Observe que essa entrada leva em consideração tanto as instâncias resolvidas na otimalidade (ou provadas como inviáveis) como aquelas para as quais a execução foi pausada em 3600s. O último par de valores fornece a média e o desvio padrão dos *gaps* de otimalidade relativos obtidas pelo CPLEX. Esses *gaps* são dados por $\frac{LS-LI}{LS}$, onde LI e LS são, respectivamente, os melhores limites inferior e superior obtidos dentro de uma hora de execução. Se, para uma dada instância, nenhuma solução viável foi encontrada dentro do limite de tempo e sua inviabilidade também não foi provada, o *gap* de otimalidade correspondente é considerado 100%. Da mesma forma, o *gap* é definido como 0% sempre que a instância foi provada como inviável ou resolvida na otimalidade. Nos casos em que $LI = LS = 0$, os *gaps* de otimalidade também são definidas como 0%.

Para cada conjunto de instâncias, são apresentados também quatro valores relativos à execução da mat-heurística proposta. O primeiro valor corresponde à quantidade de instâncias para as quais foi obtido um limite tão bom quanto ou melhor que o limite primal obtido pelo CPLEX em até uma hora de execução. Em seguida, é dado o tempo médio de execução da heurística, e o último par de valores fornece a média e o desvio padrão da melhoria relativa dos limites primais

obtidos. Precisamente, esses valores correspondem a $\frac{LI_{KS}-LI}{LI}$, onde LI_{KS} e LI são, respectivamente, os melhores limites inferiores (primais) obtidos pela heurística *Kernel Search* e pelo CPLEX (dentro de uma hora de execução). Naturalmente, valores de melhoria negativos indicam os casos em que o CPLEX obteve melhores limites do que a heurística. Os resultados são resumidos na linha “sub-total” para cada tamanho de instância, e a última linha da tabela fornece os resultados gerais, considerando as 72 instâncias testadas.

Os resultados indicam que o problema é bastante desafiador na prática, uma vez que, mesmo para instâncias relativamente pequenas (de 25 e 50 vértices), o CPLEX é capaz de resolver na otimalidade apenas 5 das 72 instâncias. Além disso, o *gap* médio das soluções obtidas é elevado (69.38%). A dificuldade de resolver o problema na prática reflete a sua complexidade teórica, uma vez que achar uma solução inicial viável para o POERS-PD é NP-difícil. Por outro lado, a mat-heurística proposta surge como uma alternativa promissora, ao oferecer limites primais tão bons quanto ou melhores que os do CPLEX para 60 das 72 instâncias testadas. Em particular, ela encontrou soluções tão boas quanto ou melhores que o CPLEX para todo o subconjunto de instâncias de 50 vértices. Além disso, o tempo de execução médio da heurística é praticável (um pouco mais de cinco minutos). É interessante notar, ainda, que os tempos de execução médios da mat-heurística são maiores quanto maior o quartil de limite de tempo de rota considerado. Uma possível explicação está no fato de que, quanto maiores os limites de rota, maior é o espaço viável a ser explorado, retardando a resolução dos sub-problemas na otimalidade. Por outro lado, curiosamente, o CPLEX foi capaz de encontrar menores *gaps* médios para instâncias com maiores limites de rota T . Uma hipótese para esse comportamento também se apoia na maior disponibilidade de rotas viáveis no espaço de busca, possivelmente facilitando a construção de soluções factíveis (e limites primais) já nos primeiros nós da árvore de *branch-and-bound*.

6. Conclusões e trabalhos futuros

Neste trabalho, introduzimos uma nova versão de otimização robusta do problema de orientação de equipes com vértices obrigatórios e valores de prioridades decrescentes em função do tempo. O problema é formalmente definido utilizando a caracterização robusta conhecida por *incerteza orçamentada*, e uma formulação compacta baseada em fluxo é proposta e resolvida diretamente com CPLEX e por meio de uma mat-heurística *Kernel Search*. Os resultados evidenciam a dificuldade de resolução (exata) do problema para instância de até 50 vértices. Ademais, a heurística proposta apresenta-se como uma alternativa praticável tanto em tempo médio de execução (aproximadamente cinco minutos) como em qualidade das soluções obtidas. Em particular, a heurística foi capaz de encontrar limites tão bons quanto ou melhores que o CPLEX em 60 das 72 instâncias testadas.

Por ser um problema novo e com altas complexidades teórica e prática, diversas direções de pesquisa ainda podem ser exploradas. Em especial, primeiros passos devem ser dados em propor e testar novos modelos matemáticos possivelmente mais fortes, com ou sem desigualdades válidas, além de elaborar algoritmos exatos baseados em técnicas mais elaboradas, como planos de cortes, decomposição de Benders e geração de colunas. Além de mat-heurísticas, existem ainda os diversos *frameworks* de metaheurísticas a serem testados para o problema. Por fim, estratégias híbridas, utilizando soluções heurísticas como ponto de partida para algoritmos exatos, é também uma direção promissora de estudo futuro.

Agradecimentos

Este projeto recebeu financiamento do programa de pesquisa e inovação *European Union's Horizon 2020* no âmbito do acordo de subvenção Marie Skłodowska-Curie No 101034329. Beneficiário do Programa *WINNINGNormandy* apoiado pela Região da Normandia.

Referências

- Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L. M., Poss, M., e Requejo, C. (2013). The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3): 856–866. Transport Scheduling.
- Angelelli, E., Archetti, C., Filippi, C., e Vindigni, M. (2017). The probabilistic orienteering problem. *Computers & Operations Research*, 81:269–281.
- Angelelli, E., Mansini, R., e Grazia Speranza, M. (2010). Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11):2017–2026.
- Assunção, L. e Mateus, G. R. (2019). A cutting-plane algorithm for the Steiner team orienteering problem. *Computers & Industrial Engineering*, 135:922–939.
- Baffo, I., Carotenuto, P., e Rondine, S. (2017). An orienteering-based approach to manage emergency situation. *Transportation Research Procedia*, 22:297–304.
- Bertsimas, D. e Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.
- Campbell, A. M., Gendreau, M., e Thomas, B. W. (2011). The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, 186:61–81.
- Chao, I.-M., Golden, B. L., e Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474.
- Coco, A. A., Santos, A. C., e Noronha, T. F. (2022). Robust min-max regret covering problems. *Computational Optimization and Applications*, 83:111–141.
- Duhamel, C., Santos, A. C., Brasil, D., Châtelet, E., e Birregah, B. (2016). Connecting a population dynamic model with a multi-period location-allocation problem for post-disaster relief operations. *Annals of Operations Research*, 247:693–713.
- Evers, L., Dollevoet, T., Barros, A. I., e Monsuur, H. (2014a). Robust uav mission planning. *Annals of Operations Research*, 222:293–315.
- Evers, L., Glorie, K., van der Ster, S., Barros, A. I., e Monsuur, H. (2014b). A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research*, 43: 248–260.
- Guastaroba, G., Savelsbergh, M., e Speranza, M. (2017). Adaptive kernel search: A heuristic for solving mixed integer linear programs. *European Journal of Operational Research*, 263(3):789–804.
- Gunawan, A., Lau, H. C., e Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255 (2):315–332.
- Jin, H. e Thomas, B. W. (2019). Team orienteering with uncertain rewards and service times with an application to phlebotomist intrahospital routing. *Networks*, 73(4):453–465.
- Lee, C. e Lee, K. (2012). Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63:1294–1306.

- Letchford, A. N., Nasiri, S. D., e Theis, D. O. (2013). Compact formulations of the Steiner traveling salesman problem and related problems. *European Journal of Operational Research*, 228(1):83–92.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.
- Miller, C. E., Tucker, A. W., e Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329.
- Munari, P., Moreno, A., De La Vega, J., Alem, D., Gondzio, J., e Morabito, R. (2019). The robust vehicle routing problem with time windows: Compact formulation and branch-price-and-cut method. *Transportation Science*, 53(4):1043–1066.
- Murat Afsar, H. e Labadie, N. (2013). Team orienteering problem with decreasing profits. *Electronic Notes in Discrete Mathematics*, 41:285–293.
- Song, Y., Ulmer, M. W., Thomas, B. W., e Wallace, S. W. (2020). Building trust in home services—stochastic team-orienteering with consistency constraints. *Transportation Science*, 54(3):823–838.
- Tang, H. e Miller-Hooks, E. (2005). Algorithms for a stochastic selective travelling salesperson problem. *Journal of the Operational Research Society*, 56(4):439–452.
- Wang, D., Yang, K., Yang, L., e Dong, J. (2023). Two-stage distributionally robust optimization for disaster relief logistics under option contract and demand ambiguity. *Transportation Research Part E: Logistics and Transportation Review*, 170:103025.
- Yu, Q., Cheng, C., e Zhu, N. (2022). Robust team orienteering problem with decreasing profits. *INFORMS Journal on Computing*, 34(6):3215–3233.
- Zheng, Y.-J., Chen, S.-Y., e Ling, H.-F. (2015). Evolutionary optimization for disaster relief operations: A survey. *Applied Soft Computing*, 27:553–566.
- İlhan, T., Iravani, S., e Daskin, M. (2008). The orienteering problem with stochastic profits. *IIE Transactions*, 40:406–421.