HEURÍSTICA COLÔNIA DE FORMIGAS APLICADA À LOGÍSTICA EMERGENCIAL

João Pedro da Silva Cerqueira
Escola Politécnica, Universidade Federal da Bahia
Rua Aristides Novis, nº 2, Federação, Salvador, BA, 40210-630
joao.0602cerqueira@gmail.com

Rodrigo Freitas Sá Barretto
Escola Politécnica, Universidade Federal da Bahia
Rua Aristides Novis, nº 2, Federação, Salvador, BA, 40210-630
rodrigo.fsbarretto@gmail.com

Islame Felipe da Costa Fernandes
Instituto de Computação, Universidade Federal da Bahia
Av. Milton Santos - Ondina, Salvador - BA, 40170-110
islame.felipe@ufba.com

RESUMO

Este trabalho propõe a aplicação do Problema do Caixeiro Viajante com Bônus (PCV-B) como uma ferramenta matemática para otimizar a distribuição de recursos em contextos emergenciais. A abordagem busca estabelecer rotas eficientes, considerando a minimização da distância percorrida e a maximização da cobertura de áreas prioritárias, que possuem maiores bônus. Para isso, foi desenvolvido e adaptado o algoritmo de Colônia de Formigas (ACO) para a resolução desse problema, além da realização de testes e comparações dessa adaptação com um outro algoritmo adaptado, o GRASP, aplicado a cenários semelhantes, o qual foi superado quanto às soluções.

PALAVRAS CHAVE. Metaheurística. Distribuição. Emergencial.

Tópicos (TAG - Teoria dos Grafos e Algoritmos Relacionados, MH - Metaheurísticas)

ABSTRACT

This paper proposes the application of the Traveling Salesman Problem with Bonus (TSP-B) as a mathematical tool to optimize the distribution of resources in emergency situations. The approach seeks to establish efficient routes, considering the minimization of the traveled distance and the maximization of the coverage of priority areas, which have greater bonuses. To this end, the Ant Colony Algorithm (ACO) was developed and adapted to solve this problem, in addition to testing and comparing the adaptation with another adapted algorithm, GRASP, applied to similar scenarios, o GRASP, applied to similar scenarios, which was surpassed in terms of solutions.

KEYWORDS. Metaheuristic. Distribution. Emergency.

Paper topics (TAG - Graph Theory and Related Algorithms, MH - Metaheuristics)

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

1. Introdução

Desastres naturais, crises humanitárias e emergências climáticas são cada vez frequentes. Esses eventos impactam diretamente a vida da população, exigindo soluções imediatas e eficazes. Embora muitos desses acontecimentos não possam ser previstos com grande antecedência, é fundamental adotar medidas para mitigar os danos e garantir que a população afetada receba os recursos necessários de forma adequada e dentro de um prazo oportuno.

De acordo com a CNN Gama [2025], apenas em 2024, o Brasil registrou aproximadamente 1.690 desastres ambientais. Esses eventos afetam principalmente as áreas urbanas, em virtude da alta densidade demográfica e podem resultar no isolamento de regiões inteiras.

A discussão sobre o tema desse trabalho ganhou visibilidade recentemente, conforme indicado por Silveira et al. [2023], em razão da pandemia de coronavírus, que, além de causar danos sociais e econômicos, expôs a fragilidade do sistema global de saúde e logística.

Dessa forma, este trabalho propõe a utilização de modelos matemáticos baseados no *Problema do Caixeiro Viajante com Bônus* (PCV-B) [Feillet et al., 2005] como uma ferramenta de otimização, aplicada à definição das rotas mais eficientes para pontos de grande importância em situações emergenciais.

Pensar na logística de entrega torna-se uma questão, portanto, imprescindível. Modelos matemáticos que visam otimizar rotas de entrega ao mesmo tempo que priorizam caminhos mais urgentes, para garantir uma alocação eficaz dos recursos disponíveis, podem ser decisivos nesses contextos.

O PCV-B é fundamentado na ideia de otimização concomitantemente às escolhas ponderadas, considerando a urgência em determinados destinos mais críticos em função da intensidade e da vulnerabilidade da região afetada. No cenário abordado, o "bônus" pode ser entendido como um indicador de qual região possui maior prioridade para as entregas, levando em consideração o fator escolhido (e.g. número de pessoas, capacidade de alocação de pessoas) para indicar sua importância e a gravidade da situação no local de destino. Ademais, na literatura, há uma lacuna em relação ao uso do PCV-B em contextos emergenciais, sendo uma variação que pode representar, de melhor forma, cenários com diferentes graus de importância para os vértices, resultando em caminhos mais benéficos para a logística.

Será desenvolvida uma modelagem matemática baseada em grafos para representar as localidades envolvidas no problema. Cada localidade será representada por um vértice, ao qual será associado um bônus que reflete a gravidade da emergência naquele ponto. O objetivo deste trabalho é implementar uma variação do *Ant Colony Optimization* (ACO) para o PCV-B e avaliar o desempenho associado ao modelo proposto, que busca os melhores resultados possíveis, maximizando o bônus recolhido e limitando a distância percorrida, garantindo uma distribuição de recursos eficiente e otimizada.

Para este estudo foi seguida uma abordagem quantitativa e exploratória. A metodologia adotada é composta por três etapas principais: modelagem matemática, desenvolvimento de algoritmos e análise de resultados experimentais. A análise dos resultados será feita com base nos tempos de execução, custo total da melhor rota e o total de bônus coletado no caminho. Ademais, a análise será feita a partir da comparação de diferentes parâmetros dentro de cada algoritmo, o desempenho de um algoritmo em relação ao outro e o desempenho geral em relação a um outro tipo de heurística para o mesmo problema. Para isso, utilizar-se-á, para confrontação, o algoritmo GRASP, também adaptado para problemas com bônus, que apresentou soluções mais razoáveis com um menor custo computacional.

Em resumo, este trabalho investiga o uso do PCV-B em logísticas de situações emergenciais, propondo uma abordagem de distribuição de recursos que pondera os benefícios de cada vértice em relação ao seu custo logístico. Além disso, apresenta-se a adaptação do algoritmo de Otimização por Colônia de Formigas (ACO) para resolver o PCV-B - uma aplicação ainda não explorada na literatura.

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

2. Trabalhos Correlatos

À medida que momentos de crise humanitária se tornam recorrentes, cresce o número de estudos científicos que procuram desenvolver meios mais eficazes de como gerir a difusão de itens essenciais à sobrevivência humana, com o objetivo de mitigar os danos causados.

O estudo de Mukhairez e Maghari [2015] compara diferentes algoritmos: Ant Colony Optimization (ACO), Genetic Algorithm (GA) e Simulated Annealing (SA), e, como resultado, tem-se que o ACO encontrou um percurso mais relevante, embora tenha demorado mais.

Ramadhan et al. [2025] utilizaram uma variação do PCV, o PCV com Drone e Barco, para agir sobre situações de alagamento, utilizando-se de um caminhão, de um drone e de um barco. Para esse fim, recorreram a uma técnica aumentada de ϵ-restrito e obtiveram redução significativa do tempo de conclusão das rotas e equilíbrio entre critério de tempo e equidade na distribuição de suprimentos, além de atenderem às situações de isolamento geográficos. No entanto, além da abordagem distinta, esse trabalho foca em situações de resposta, enquanto este trabalho pode ser utilizado tanto como uma resposta, quanto uma prevenção.

Ding [2011] utiliza uma variação da Colônia de Formigas, a Fish-Swarm Ant Colony Optimization (Otimização Híbrida por Colônia de Formigas e Cardume de Peixes), que teve resultados melhores que o algoritmo clássico, mas sem as restrições de um PCV.

Wei [2025] também propõe a utilização de uma variante do ACO, a improved ant colony algorithm (IACA), para problemas PCV. Em simulações de zonas de lockdown em Pudong, Xangai, o algoritmo atingiu redução de custo de 30% em comparação ao ACO clássico, mas não leva em consideração, também, a formulação de bônus para maximizar a relação entre os benefícios da rota e seus custos. Não leva em consideração cenários onde determinada estrutura pode não ser uma boa escolha para a distribuição de recursos.

de Albuquerque et al. [2018] utilizaram diversos algoritmos para roteamento em situações de emergência, entre eles, o ACO. No estudo, esse algoritmo foi o que obteve melhor resultado, dentre os algoritmos de Christofides, Genético e Vizinho Mais Próximo. No entanto, novamente, não se formulou bônus para os vértices.

Na literatura, não se encontra trabalhos que investiguem a aplicabilidade do PCV-B para cenários de emergências, e esse trabalho vem preencher essa lacuna na literatura.

3. Descrição formal do problema

3.1. Formalização

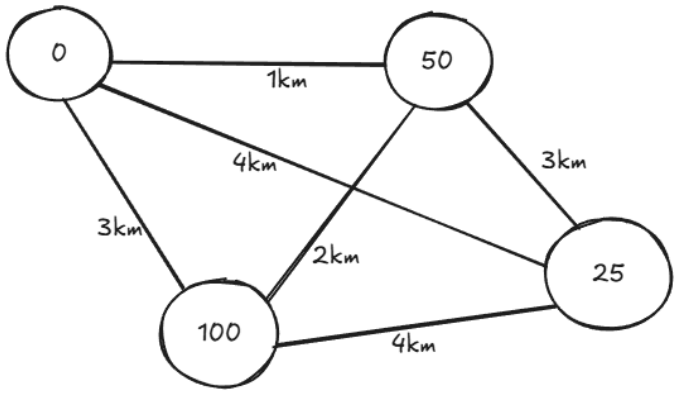


Figura 1: Exemplo de grafo para o PCV-B

O grafo G deve ser interpretado como uma representação de uma rede de respostas para emergências, na qual, temos:
Um conjunto $V = \{1, 2, ..., |V|\}$ de vértices, representando os locais que, numa situação de emergência, podem ser utilizados para logísticas de distruibuição de recursos à população. Cada

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

aresta contém um valor p_u, prêmio recebido ao ser visitado, ou γ_i, uma penalidade obtida quando o vértice i não é visitado. Além disso, temos um conjunto $E = \{1, 2, ..., |E|\}$ de arestas, que simbolizam qualquer via de comunicação que interligue dois pontos emergenciais. Para além de representar as ligações físicas, as arestas também possuem um valor c_{ij}, que representa o custo de se percorrer a aresta ij.

O objetivo deste problema é encontrar um ciclo hamiltoniano em um subgrafo induzido que maximize a diferença entre os lucros e as penalidades obtidas.

3.2. Função objetivo

Neste artigo, será implementada a variante Profitable Tour Problem (PTP) do PCV-B, definida originalmente por Dell'Amico et al. [1995]. Assim, tem-se, como função objetivo, a expressão 1 adaptada de Feillet et al. [2005], invertendo-se a função objetivo para maximizar o lucro líquido .

$$max \left(\sum_{i \in V} p_i y_i - \sum_{(i,j) \in A} c_{ij} x_{ij} \right), \tag{1}$$

Associamos uma variável binária x_{ij} $\forall$ arco (i, j), assumindo o valor de 1 quando o arco está na solução, e 0, caso contrário. Da mesma forma, definimos a variável binária y_i para os vértices. p_i representa o bônus do vértice i, c_{ij} representa o custo da aresta (i, j).

3.3. Restrições do problema

Para grafos não direcionados, impõe-se que, para cada vértice visitado, haja exatamente duas arestas incidentes na rota, conforme:

$$\sum_{v_i \in V \setminus \{v_j\}} x_{ij} = 2\, y_i, \quad \forall v_j \in V, \tag{2}$$

o que garante a formação de um ciclo que conecte os pontos emergenciais.

O vértice de partida (ponto de distribuição) deve estar presente na estrutura, conforme:

$$y_1 = 1, \tag{3}$$

estabelecendo que o local de origem dos recursos é fixo e garantido.

As definições dos domínios das variáveis são formalizadas por:

$$x_{ij} \in \{0, 1\}, \quad \forall\, (v_i, v_j) \in A, \tag{4}$$

$$y_i \in \{0, 1\}, \quad \forall\, v_i \in V. \tag{5}$$

A distância entre os pontos é o único fator considerado para definir os custos de transporte, desconsiderando-se variáveis como condições meteorológicas, estado das estradas, tráfego e direção das vias.

O vértice de partida, utilizado exclusivamente como ponto de distribuição, não necessita de atendimento emergencial e, por isso, recebe um prêmio igual a zero.

Assume-se que o problema é simétrico. Para um grafo $G = (V, E)$ e para quaisquer $u, v \in V$, se $d(u, v)$ representa a função que retorna a distância entre os vértices u e v, então:

$$d(u, v) = d(v, u).$$

O modelo restringe-se a grafos completos. Na ausência de uma conexão direta real entre dois vértices (isto é, se $(u, v) \notin E$), inclui-se uma aresta fictícia com custo extremamente elevado (tendendo a ∞), representando a inexistência prática de via direta.

Para cenários em que o grafo não é completo, a adição de arestas com valores muito altos pode gerar 2 casos extraordinários. Eles são:

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

1. Se o vértice de grau um for o ponto de partida, as opções viáveis são: não visitar nenhum outro vértice ou visitar apenas seu único vizinho, retornando posteriormente ao início.

2. Se o vértice de grau um não for o ponto de partida, ele não poderá ser visitado, pois qualquer tentativa implicaria em sair do vértice e, inevitavelmente, repetir a passagem por outro nó, violando as restrições do problema.

3.4. Modelagem em grafos

Os vértices representam pontos estratégicos de apoio logístico, como hospitais, mercados etc. As arestas representam as vias físicas que conectam essas estruturas, os bônus e as penalidades representam a capacidade de contribuição da estrutura para a coordenação e os custos exprimem os ônus associados ao transporte, decorrentes das extensões dos acessos.

4. Solução algorítmica

O Ant Colony Optimization (ACO) possui uma abordagem probabilística para explorar a maior parte das soluções possíveis, executando uma seleção que é realizada com base na quantidade de ferômonios de cada caminho. Este trabalho propôs uma adaptação do algoritmo ACO para o PCV-B, denominada ACO-PTP.

4.1. Algoritmo ACO-PTP

Algorithm 1: Algoritmo de Colonia de Formigas para o PTP

Input: grafo, numero de iteracoes (ITERATIONS), numero de formigas (FORMIGAS), taxa de evaporacao (EVAPORATION_RATE), valores Q_EDGE_VALUE e Q_VERTEX_VALUE, parametros ALPHA, BETA, GAMMA, DELTA

Output: melhor caminho, melhor distancia, melhor bonus

Função `AntColony(`*grafo*`)`**:**

```
melhor_caminho ← None
melhor_distancia ← infinito
melhor_bonus ← 0
for i ← 1 to ITERATIONS do
    distancias ← ∅
    bonus ← ∅
    caminhos_iteração ← ∅
    for j ← 1 to FORMIGAS do
        caminho_atual, distancia_atual, bonus_atual ←
          make_path(vertice_inicial)
        adicione caminho_atual à lista de caminhos_iteração
        adicione distancia_atual à lista de distancias
        adicione bonus_atual à lista de bonus
        if
          bonus_atual − distancia_atual > melhor_bonus − melhor_distancia
          then
            melhor_caminho ← caminho_atual
            melhor_distancia ← distancia_atual
            melhor_bonus ← bonus_atual
    for elemento in caminho do
        if elemento e do tipo Aresta then
            elemento.update_pheromone(distancia)
        else if elemento e do tipo Vertice then
            elemento.update_pheromone(bonus)
return melhor_caminho, melhor_distancia, melhor_bonus
```

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

O ACO-PTP estende o uso de feromônios não apenas às arestas, mas também aos vértices do grafo. A partir dessa modificação, foi desenvolvido um algoritmo voltado para o Profitable Tour Problem (PTP) [Dell'Amico et al., 1995], o qual se mostra especialmente eficaz em cenários onde a otimização da relação entre custo e benefício é crucial.

O algoritmo detalha o procedimento da meta-heurística de Colônia de Formigas adaptada ao bônus. A função **make_path** difere da versão original. Nela, obtemos o caminho escolhido por cada formiga com base na fórmula 6 que define a probabilidade de escolha de cada par de aresta e vértice, conforme definido em 4.1.2. Essa probabilidade é diretamente proporcional ao bônus do vértice, ao feromônio do vértice e ao feromônio da aresta, e inversamente proporcional ao custo da aresta. Todos os vértices adjacentes ao vértice atual, que ainda não foram visitados durante o percurso, são candidatos, incluindo o vértice inicial. Se o vértice inicial for escolhido, o ciclo foi encontrado e o caminho é interrompido. Caso contrário, fazemos o mesmo para o vértice escolhido, continuando o percurso.

4.1.1. Apresentação e explicação do pseudo-código

O algoritmo começa inicializando o melhor caminho atual como vazio, a melhor distância como infinita e o melhor bônus como 0. Com base em parâmetros definidos pelo usuário, como o grafo, o número de iterações, número de formigas, taxa de evaporação, valores Q e parâmetros alpha, beta, gamma e delta, começamos a nossa primeira iteração.

Em cada iteração, cada formiga encontrará seu próprio caminho, usando a função **make_path** explicada anteriormente. Após definir um caminho, checamos se os resultados obtidos nesse caminho, como o bônus e o custo total foram os melhores até então. Todos esses resultados são guardados conforme o seu caminho de origem. Posterior a todas as formigas, atualizamos, com base nesses valores, os feromônios depositados pelas formigs em seus caminhhos. A função **update_pheromone(distância)** utiliza a função 8 para vértices e 7 para arestas, definidas em 4.1.2, para atualizar os valores.

4.1.2. Formulação matemática do algoritmo

$$p_{i,j}^{k}(t) = \begin{cases} \frac{\tau_{i,j}^{\alpha}(t)\cdot\eta_{i,j}^{\beta}\cdot\zeta_{i}^{\gamma}(t)\cdot\sigma_{i}^{\delta}}{\sum_{j\in J^{k}}\tau_{i,j}^{\alpha}(t)\cdot\eta_{i,j}^{\beta}\cdot\zeta_{i}^{\gamma}(t)\cdot\sigma_{i}^{\delta}} & \text{se } j \in J^{k}, \\ 0 & \text{c.c.} \end{cases} \tag{6}$$

Expressão adaptada de França [2007] onde:

- $p_{i,j}^{k}(t)$ é a probabilidade da formiga k optar pela aresta (i, j) na iteração t;
- J^{k} é a lista de vértices ainda não visitados, além do vértice inicial;
- $\tau_{i,j}$ é a quantidade de feromônio na aresta (i, j);
- $\eta_{i,j}$ é a informação de qualidade dessa aresta (geralmente determinada pelo inverso da distância);
- ζ_i é a quantidade de feromônio no vértice i
- σ_i é a qualidade desse vértice, determinado pelo bônus;
- α, β, γ e δ são parâmetros que definem o grau de importância de τ, η, ζ e σ respectivamente.

A expressão 6 é usada para definir as probabilidades de cada aresta da instância de serem escolhidas pelas formigas durante a iteração, durante a execução da função **make_path**.

$$\tau_{i,j}(t+1) = (1-\rho)\cdot\tau_{i,j}(t) + \frac{Q_edge}{path_distance} \tag{7}$$

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

Onde:

- ρ é a taxa de evaporação dos feromônios;
- Q_edge é um parâmetro
- $path_distance$ é o custo total do caminho;

$$\tau_{i,j}(t+1) = (1-\rho) \cdot \tau_{i,j}(t) + Q_vertex \cdot path_bonus \tag{8}$$

Onde:

- ρ é a taxa de evaporação dos feromônios;
- Q_vertex é um parâmetro
- $path_distance$ é o custo total do caminho;

4.1.3. Parâmetros

O algoritmo desenvolvido para o ACO-PTP utiliza os parâmetros α, β, γ e δ para ajustar o comportamento da formiga durante uma iteração. O parâmetro α determina a prioridade para a escolha de uma aresta com mais feromônios, enquanto β regula a preferência por caminhos de menor custo. De forma análoga, o γ modula a precedência para os vértices com mais feromônios e δ a importância do bônus.

5. Experimentos

5.1. Metodologia

Os experimentos realizados neste trabalho tiveram como objetivo testar o funcionamento do algoritmo ACO adaptado para o PTP desenvolvido e analisar os resultados obtidos.

5.1.1. Instâncias

Para o experimento deste trabalho, foi formulado um modelo de grafo baseado em localidades de grande aglomeração populacional da cidade de Salvador, o que as posiciona como prováveis locais de refúgio para um grande número de pessoas. Dessa forma, para calcular os bônus atribuídos aos vértices, utiliza-se a seguinte fórmula:

$$p_i = \frac{\text{Área Total Construída}}{1000} \tag{9}$$

A escolha da área total construída como parâmetro para calcular o bônus baseia-se na premissa de que ela serve como um indicador da capacidade do local em alocar pessoas. Essa decisão também se justifica pela limitação de informações disponíveis no contexto analisado. Para medir as distâncias (i.e. custos) entre os pontos foi utilizado o Google Maps, de forma em que era selecionado o ponto de partida e o ponto de chegada e então, o aplicativo fornecia a distância entre os pontos (i.e. arestas do grafo).

Para compor o conjunto de testes, selecionaram-se 18 instâncias da TSPLIB95, a saber: *burma14*, *gr17*, *gr21*, *gr24*, *fri26*, *bays29*, *swiss42*, *gr48*, *hk48*, *eil51*, *berlin52*, *brazil58*, *eil76*, *pr76*, *rat99*, *kroA100*, *kroB100*, *rd100*.. Essas instâncias abrangem cenários com cardinalidades de vértices variando de 14 a 100, o que possibilita avaliar o desempenho do algoritmo em problemas de baixa e alta dimensão. Ademais, incluiu-se pares de instâncias de mesmo tamanho, mas com distribuições de custos e valores de bônus distintos, permitindo analisar o comportamento do método em cenários estruturalmente similares, porém com parâmetros de custo diferentes.

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

5.1.2. Ajuste de parâmetros

Devido à natureza probabilística do algoritmo, cada execução pode produzir resultados distintos mesmo com parâmetros idênticos. Para reduzir essa variabilidade e obter resultados estatisticamente mais confiáveis, foi realizado um procedimento de *tuning* dos parâmetros α, β, γ, δ e da taxa de evaporação (ER), detalhado na Tabela 1. Inicialmente, o algoritmo foi avaliado em 1 080 configurações de parâmetros diferentes, sendo cada combinação executada cinco vezes para o cálculo da média das métricas obtidas. Durante essa fase, comparou-se as configurações que apresentaram o maior lucro médio para formar o conjunto ótimo de parâmetros.

Parâmetro	Intervalo	Passo
ER	$0.1 - 0.3$	0.1
α	$0.5 - 2$	0.5
β	$1 - 5$	1
γ	$0.5 - 3$	0.5
δ	$1 - 3$	1

Tabela 1: Tuning

5.1.3. Testes e análises

Utilizando os parâmetros ótimos escolhidos foram submetidas mais dez execuções, a partir das quais foram calculadas as médias do lucro (bônus - custo), da distância percorrida (custo), do bônus médio e do tempo médio de execução. Durante a busca pelos parâmetros ótimos, empregaram-se 20 iterações com 10 formigas. Nas dez execuções finais com os parâmetros obtidos, utilizou-se 250 iterações e um número de formigas igual ao número de vértices.

Além disso, realizou-se uma comparação direta entre o ACO-PTP e o GRASP-PTP, inicialmente proposto por Feo e Resende [1995]. A avaliação foi utilizando-se as instâncias da TSPLIB95, variando em densidade de 14 até 100 vértices, permitindo uma análise detalhada do desempenho dos algoritmos em cenários de diferentes escalas. Para o GRASP-PTP, baseou-se em Keshtkaran e Ziarati [2016], que utilizaram o GRASP para resolver uma variante próxima, o Orienteering Problem (OP). O GRASP foi aplicado com as seguintes características: Fase de construção com um construtor guloso baseado na razão prêmio/distância, RCL estática e fase de busca local por 2-opt. Alterou-se, por fim, a função objetivo para maximizar o lucro, alinhando-se ao PTP.

Os testes foram desenvolvidos na linguagem de programação Python, versão 3.10.12, e executados em um sistema com processador Intel Core i5-1334U, 16 GB de memória RAM (3200 MT/s) e sistema operacional Ubuntu 22.04.5 LTS.

5.2. Resultados

As Tabelas 2 e 3 exibem os resultados obtidos pelos algoritmos ACO-PTP e GRASP-PTP. Observa-se que, em todas as instâncias analisadas, o ACO-PTP alcança bônus iguais ou mais elevados e custos iguais ou mais reduzidos em comparação ao GRASP-PTP, refletindo-se em lucros superiores, com destaque para *brazil58*, *ulysses22*, *berlin52*, *kroA100* e *rd100*, que obtiveram resultados significativamente superiores.

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

Tabela 2: Resultados do ACO-PTP

Instância	Lucro	Distância	Bôus	Tempo (s)
burma14	4371.50	2703.50	7075.00	0.47
gr17	6940.00	1920.00	8860.00	2.05
gr21	7936.10	2057.40	9993.50	3.08
ulysses22	3640.90	3603.50	7244.40	2.41
salvador	630.10	117.90	748.00	0.63
gr24	11037.70	1301.80	12339.50	5.59
fri26	12672.50	957.50	13630.00	5.40
bays29	14235.20	1937.80	16173.00	8.33
swiss42	19106.10	1397.90	20504.00	19.22
gr48	16885.50	5232.30	22117.80	28.46
hk48	14294.80	11852.90	26147.70	34.16
eil51	24403.90	491.60	24895.50	43.18
berlin52	19147.90	7452.10	26600.00	42.05
brazil58	14389.00	14292.30	28681.30	42.38
st70	33684.30	804.70	34489.00	112.60
eil76	33285.00	617.50	33902.50	145.60
pr76	756.00	0.00	756.00	128.76
rat99	48309.90	1504.90	49814.80	331.00
kroA100	24758.70	20660.30	45419.00	273.50
kroB100	26837.20	20587.10	47424.30	400.58
rd100	40235.90	9386.60	49622.50	308.60

Tabela 3: Resultados do GRASP-PTP

Instância	Lucro	Distância	Bônus	Tempo (s)
burma14	4222.01	2694.0	6916.0	0.011
gr17	6845.92	1513.35	8359.25	0.036
gr21	7873.03	1873.0	9746.0	0.039
ulysses22	2294.00	745.0	3039.0	0.002
salvador	627.04	117.36	744.40	0.31
gr24	10948.98	1289.7	12238.6	0.208
fri26	12550.85	993.2	13544.0	0.354
bays29	14011.34	2161.7	16173.0	0.461
swiss42	18885.41	1581.8	20467.2	4.004
gr48	16321.12	5129.6	21450.6	1.563
hk48	12710.65	9913.9	22624.55	0.31
eil51	24371.61	490.95	24862.5	9.172
berlin52	17504.42	7057.1	24561.5	0.688
brazil58	1248.05	1892.0	3140.0	0.003
st70	33500.73	944.35	34445.05	31.365
eil76	33204.35	674.85	33879.2	42.192
pr76	756.0	0.00	756.0	0.001
rat99	47921.75	1759.55	49681.3	113.48
kroA100	18580.35	13217.7	31798.05	0.58
kroB100	24219.95	16019.3	40239.25	1.72
rd100	37944.55	10242.25	48186.8	52.76

Além disso, é possível observar que algumas instâncias geraram resultados particularmente interessantes. A instância *pr76* não visitou nenhuma aresta e o único bônus coletado foi o do vértice inicial, uma vez que todos os bônus aleatórios são menores que 1000 e os custos de todas as

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

arestas adjacentes ao vértice inicial eram superiores a esse valor, resultando em um cenário em que permanecer no vértice inicial maximiza a função objetivo. Por isso, essa foi a única instância em que o GRASP-PTP não foi superado.

Ademais, apesar de *pr76* e *eil76* possuírem ambas 76 vértices, a instância *eil76* apresentou um tempo de execução bem diferente, evidenciando como a topologia do grafo influencia no resultado do algoritmo.

6. Considerações finais

Este trabalho propôs a aplicação do PCV-B, uma variação do PCV, em Logística Humanitária (LH) voltada para a distribuição de recursos em situações de emergência.

Ao longo deste estudo, adaptou-se um algoritmo de colônia de formigas para o PTP. A principal modificação em relação ao modelo tradicional consiste em ponderar as escolhas de rota pela razão entre o bônus e o custo e em propagar feromônios não apenas nas arestas, mas também nos vértices percorridos pelas formigas.

Ao término dos experimentos, constatou-se que o ACO-PTP superou ou igualou o GRASP-PTP em todos os cenários avaliados. Se necessário, ajustes nos parâmetros (número de formigas, número de iterações e taxa de evaporação) podem calibrar o algoritmo para cada situação, permitindo ampliar o tempo de execução em busca de maiores lucros ou, inversamente, restringi-lo em prol de maior eficiência computacional. Ademais, em cenários em que o tempo é um recurso crítico, o processo de tuning pode ser simplificado pela redução do número de combinações testadas. Para tentar compensar uma eventuais diminuições de lucro decorrente dessa simplificação, pode-se aumentar a quantidade de iterações e/ou formigas.

A utilização de uma infraestrutura computacional mais robusta pode resultar em ganhos expressivos de desempenho, ao possibilitar a execução de um maior número de iterações e formigas, bem como a realização de experimentos em instâncias maiores. Além disso, a aplicação de técnicas de paralelismo tende a reduzir consideravelmente o tempo de execução, o que favorece a adoção de novas estratégias de *tuning* e otimização dos parâmetros do algoritmo. Essas possibilidades são sugestões para trabalhos futuros, assim como a realizaçõ de comparações com outros algoritmos do PCV-B existentes na literatura.

Em síntese, os resultados obtidos confirmam a viabilidade e a eficácia da abordagem ACO-PTP como ferramenta de apoio à decisão em logística emergencial, evidenciando sua capacidade de maximizar lucros e reduzir custos em rotas de distribuição de recursos críticos. A flexibilidade do modelo, aliada ao potencial de ajuste fino de parâmetros e à possibilidade de exploração paralela, o torna especialmente adequado a diferentes escalas e condições operacionais.

Agradecimentos

Esta pesquisa foi apoiada pela Universidade Federal da Bahia, Fundação de Amparo à Pesquisa do Estado da Bahia, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), com o processo número 402842/2023-5.

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br

Referências

de Albuquerque, J. C., de Lucena, S. C., Campos, C. A. V., Simões, J. E., e de Souza Moura, P. N. (2018). Análise de heurísticas de tsp para planejamento de rota de vants em cenários de emergência.

Dell'Amico, M., Maffioli, F., e Värbrand, P. (1995). On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, 2(3):297–308.

Ding, H. (2011). Research of emergency logistics distribution routing optimization based on improved ant colony algorithm. In *Emerging Research in Artificial Intelligence and Computational Intelligence: International Conference, AICI 2011, Taiyuan, China, September 23-25, 2011. Proceedings 3*, p. 430–437. Springer.

Feillet, D., Dejax, P., e Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation science*, 39(2):188–205.

Feo, T. A. e Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6:109–133.

França, F. (2007). Ant colony optimization. `https://www.dca.fee.unicamp.br/~lboccato/topico_4.1_IA013_colonia_formigas.pdf`. Apresentação em slides. Unicamp.

Gama, G. (2025). Brasil bate recorde de alertas de desastres naturais em 2024. Publicado em 10/01/2025 às 17:12, atualizado em 10/01/2025 às 17:12.

Keshtkaran, M. e Ziarati, K. (2016). A novel grasp solution approach for the orienteering problem. *Journal of Heuristics*, 22:699–726.

Mukhairez, H. H. e Maghari, A. Y. (2015). Performance comparison of simulated annealing, ga and aco applied to tsp. *International Journal of Intelligent Computing Research (IJICR)*, 6(4): 647–654.

Ramadhan, F., Irawan, C. A., Salhi, S., e Cai, Z. (2025). The truck traveling salesman problem with drone and boat for humanitarian relief distribution in flood disaster: Mathematical model and solution methods. *European Journal of Operational Research*, 322(1):270–291. ISSN 0377-2217. URL `https://www.sciencedirect.com/science/article/pii/S0377221724007975`.

Silveira, S. C. T., Ferreira Filho, H. R., Pontes, A. N., Lopes, H. d. S., e Manfrini, G. C. (2023). A pandemia da covid-19 pelas lentes da logística humanitária. *Ciência & Saúde Coletiva*, 28: 749–759.

Wei, M. (2025). Optimization of emergency material logistics supply chain path based on improved ant colony algorithm. *Informatica*, 49(16).

DOI: 10.59254/sbpo-2025-212616
https://proceedings.science/p/212616?lang=pt-br