

## UM ALGORITMO ILS PARA O PROBLEMA DE LOCALIZAÇÃO DE PONTOS DE CONSOLIDAÇÃO

**Guilherme Barbosa de Almeida<sup>1</sup>, Elisangela Martins de Sá<sup>1</sup>,  
Marcene Jamilson Freitas Souza<sup>1,2</sup>, Sérgio Ricardo de Souza<sup>1</sup>**

<sup>1</sup> Programa de Pós-Graduação em Modelagem Matemática e Computacional,  
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Av. Amazonas 7675, CEP 30.510-000, Belo Horizonte (MG), Brasil

<sup>2</sup> Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)  
CEP 35.400-000, Ouro Preto (MG), Brasil

gbalmeida@cefetmg.br, elisangelamartins@cefetmg.br,  
marcane@ufop.edu.br, sergio@cefetmg.br

### RESUMO

Pontos de consolidação são instalações destinadas ao recebimento, controle, acondicionamento e armazenamento temporário dos produtos descartados pelos consumidores nos pontos de coleta. O problema de localização de pontos de consolidação consiste em determinar em que locais devem ser instalados pontos de consolidação para o atendimento de demandas de pontos de coleta na implantação de redes logísticas reversas. O objetivo deste trabalho é modelar o problema e desenvolver um algoritmo que gere boas soluções. O problema é modelado como um problema inteiro de localização de facilidades capacitadas, que é NP-Difícil. É desenvolvido um algoritmo que adapta a metaheurística *Iterated Local Search* (ILS) para resolvê-lo. Testes computacionais mostraram que o algoritmo gerou boas soluções para o problema, com média de desvios relativos médios de 0,52% com relação aos valores ótimos e tempo de execução menor que o gasto pelo *solver* CPLEX para encontrar a solução ótima.

**PALAVRAS CHAVE.** Localização de Pontos de Consolidação, ILS, Logística Reversa.

**Tópicos:** Logística e Transportes, Metaheurísticas

### ABSTRACT

Consolidation points are facilities for the reception, control, packaging, and temporary storage of products discarded by consumers at collection points. The Consolidation Centers Location Problem consists of determining the location of the consolidation centers to meet the demands of collection points in the reverse logistics networks. The objective of this work is to model the problem and develop an algorithm to approach solutions. The problem is modeled as a Discrete Capacitated Facility Location Problem, which is NP-Hard. An Iterated Local Search-based algorithm is developed to solve it. Computational tests showed that heuristics yielded good solutions for the problem, with an average GAP of 0.52% for the dataset generated and runtime smaller than the runtime taken by the CPLEX solver to find the optimal solution.

**KEYWORDS.** Consolidation Centers Location. Iterated Local Search. Reverse Logistics.

**Paper topics:** Logistics and Transport, Metaheuristics

## 1. Introdução

A forma atual com que se produz, consome e descarta o lixo eletrônico é insustentável. O relatório *Global E-Waste Monitor 2020*, abordando o lixo eletrônico no mundo, revelou o descarte de um número recorde de 53,6 milhões de toneladas em 2019. O relatório sublinha que apenas 17,4% dessa quantidade foi reciclada e que a tendência de aumento da quantidade de lixo eletrônico deve-se, principalmente, às altas taxas de consumo de equipamentos elétricos e eletrônicos, ciclos de vida curtos e poucas opções de reparo [Forti et al., 2020].

Parte fundamental para o enfrentamento desse problema é o investimento em logística reversa. De uma maneira simples, a logística reversa é o processo de retorno de produtos usados, de consumidores para produtores, com o objetivo de recapturar valores ou realizar descarte apropriado [Rogers e Tibben-Lembke, 2001]. Ela envolve uma série de atividades que visam a garantir que os produtos retornados sejam reprocessados adequadamente ou descartados de forma responsável.

Nos últimos anos, cresceram as ações legislativas que obrigam atividades de logística reversa de produtos eletroeletrônicos. Na Europa, as Diretivas de Resíduos de Equipamentos Elétricos e Eletrônicos (*Waste from Electrical and Electronic Equipment- WEEE*) abrangem um conjunto de leis que regulamentam a obrigatoriedade de atividades de logística reversa. A taxa de recolhimento mínima obrigatória desde 2019 é de 65%. As taxas mínimas de reciclagem e recuperação de produtos recolhidos variam, respectivamente, de 50% a 80% e 70% a 85%, de acordo com as categorias dos produtos recolhidos [União Européia, 2018].

No Brasil, recentemente, o Decreto 10.240, de 12 de fevereiro de 2020, estabeleceu normas para a implementação de sistema de logística reversa obrigatória de produtos eletroeletrônicos de uso doméstico e seus componentes. Ele instituiu metas crescentes de coleta de produtos a cada ano, a partir de 1% em 2021 até 17% em 2025 e apresentou quais cidades deverão implantar o sistema a cada ano. Segundo este decreto, os produtos são recebidos em pontos de coleta, enviados para pontos de consolidação, e, posteriormente, enviados para destinação final ambientalmente adequada, que consiste em reutilização, reciclagem, recuperação ou disposição final [Brasil, 2020].

A localização de pontos de consolidação é uma decisão estratégica importante para minimizar custos e é o foco deste artigo. O objetivo é modelar o problema e desenvolver uma heurística para gerar boas soluções. O problema é modelado como um problema inteiro de localização de facilidades capacitadas, que é um problema NP-Difícil [Qin et al., 2012]. É desenvolvido um algoritmo que adapta a metaheurística *Iterated Local Search* (ILS) para sua resolução. Testes computacionais são feitos para análises da qualidade das soluções geradas e convergência do algoritmo. Desta forma, espera-se que este estudo possa contribuir com as unidades gestoras de redes logísticas reversas na tomada de decisões estratégicas, bem como contribuir com a comunidade acadêmica, apresentando um modelo e um algoritmo para o problema.

A sequência deste texto é dividida em cinco seções. Na Seção 2 são apresentados a definição do problema, o modelo desenvolvido e a formulação matemática proposta. Na Seção 3 é apresentada a revisão bibliográfica dos métodos de resolução de problemas de localização de facilidades. Na Seção 4 é apresentada a metodologia para solução do problema através de uma adaptação da metaheurística ILS. Na Seção 5 são apresentados a metodologia dos testes computacionais, os resultados e análises de convergência. Na Seção 6 são apresentadas as conclusões deste estudo.

## 2. Descrição do Problema

Os pontos de consolidação são locais destinados ao recebimento, controle, acondicionamento e armazenamento temporário dos produtos descartados pelos consumidores nos pontos de coleta. O problema de determinar a quantidade e os locais para instalação de pontos de consolidação,

bem como definir fluxos de produtos recolhidos nos pontos de coleta, de forma a minimizar o custo total, é denominado Problema de Localização de Pontos de Consolidação (PLPC).

Para a modelagem do problema, considera-se que existe um conjunto de pontos de coleta, denotado por  $I$ . Cada ponto de coleta  $i \in I$  tem uma demanda  $d_i$ , que representa a quantidade de produtos recolhidos. Além disso, há um conjunto de locais potenciais disponíveis para instalação dos pontos de consolidação, denotado por  $J$ . Os pontos de consolidação são considerados com capacidade limitada  $m$ , que é igual para todas as instalações. Além disso, existe um custo  $c_{ij}$  de transportar a quantidade de produtos recolhidos em  $i \in I$  para o local potencial  $j \in J$  e um custo fixo  $F_j$  de instalação de um ponto de consolidação no local  $j \in J$ . Por fim, o problema é modelado de forma que cada ponto de coleta seja atendido por exatamente um ponto de consolidação. Esta condição segue a hipótese proposta em Coelho e Mateus [2017], que considera a indivisibilidade do fluxo de produtos em redes logísticas reversas.

A formulação para o problema usa as variáveis de decisão binárias  $x_{ij}$ , que indicam se  $i \in I$  é atendido por  $j \in J$  ( $x_{ij} = 1$ ) ou não ( $x_{ij} = 0$ ), e as variáveis de decisão binárias  $y_j$ , que indicam se um ponto de consolidação é instalado em  $j \in J$  ( $y_j = 1$ ) ou não ( $y_j = 0$ ). O problema é formulado como um Problema Inteiro de Localização de Facilidades Capacitadas (PILFC), conforme Qin et al. [2012]:

$$\min \sum_{j \in J} F_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (1)$$

$$\text{s.a.} \sum_{i \in I} d_i x_{ij} \leq m y_j, \quad \forall j \in J \quad (2)$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (3)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \quad (4)$$

$$y_j \in \{0,1\}, \quad \forall j \in J. \quad (5)$$

Neste modelo, a sentença (1) representa o objetivo de minimizar custos totais de instalação e transporte. As restrições (2) garantem que só há transporte de produtos para pontos de consolidação instalados. Além disso, estas restrições garantem que as capacidades dos pontos de consolidação são respeitadas. As restrições (3) garantem que cada ponto de coleta é atendido por exatamente um ponto de consolidação. As sentenças (4) e (5) definem como binárias as variáveis de decisão.

### 3. Revisão Bibliográfica

O PILFC faz parte da classe dos Problemas de Localização de Facilidades (PLF), que são problemas clássicos de otimização combinatória para determinar os locais em que serão instaladas facilidades e alocar a elas demandas de clientes, satisfazendo a um conjunto de restrições. Estes problemas são classificados como (i) capacitados, quando as facilidades têm capacidade de atendimento limitada, ou (ii) não capacitados, caso contrário. Nesta revisão, são considerados PLFs para os quais a quantidade de facilidades instalada não é definida *a priori*. Todos os trabalhos desta revisão formulam PLFs como Problemas Lineares Inteiros ou Problemas Lineares Mistos. Foi encontrado apenas o trabalho de Qin et al. [2012], que aborda o PLF capacitado como um Problema Linear Inteiro.

Muitos algoritmos exatos e heurísticos foram desenvolvidos nas últimas décadas para resolver PLFs. Akinc e Khumawala [1977] desenvolveram um algoritmo *Branch-and-Bound* para um PLF capacitado, buscando limitantes através de testes de abertura e fechamento de facilidades (heurísticas ADD/DROP) e mostraram sua eficiência para problemas de grande porte. Erlenkotter

[1978] apresentou um algoritmo que combina busca por ótimos duais e *Branch-and-Bound* para determinar a solução ótima para o PLF não capacitado. Jacobsen [1983] formulou as heurísticas abertura, fechamento, troca, localização-alocação alternada e método de substituição de vértices para problemas capacitados. Beasley [1988] apresentou um algoritmo utilizando Relaxação Lagrangeana e testes de redução baseados em testes de abertura e fechamento de facilidades para obter bons limitantes inferiores para o problema capacitado. Galvão e Raggi [1989] apresentaram um algoritmo exato para resolução do PLF não capacitado, utilizando uma estrutura hierárquica composta pelo Algoritmo Primal-Dual, Relaxação Lagrangeana com Método do Subgradiente e *Branch-and-Bound*. Avella et al. [2009] apresentaram uma heurística baseada em Relaxação Lagrangeana com o algoritmo *Branch-and-Cut*, para resolução de problemas capacitados de grande porte.

Nas duas últimas décadas, vários trabalhos utilizaram metaheurísticas para gerar boas soluções. Kratica et al. [2001] apresentaram um Algoritmo Genético para resolução do problema não capacitado com bom desempenho para resolução de problemas com mais de 1000 locais potenciais e clientes. Resende e Werneck [2006] apresentaram e mostraram a eficiência de uma heurística híbrida *multi-start* para o problema não capacitado, utilizando Busca Local e Reconexão por Caminhos. Hansen et al. [2007] apresentaram uma heurística *Variable Neighborhood Search* (VNS) para o PLF não capacitado e mostraram sua eficiência para problemas de grande porte. Sun [2012] desenvolveu uma heurística Busca Tabu com diferentes estruturas de memória, para guiar a busca por soluções para o PLF capacitado. Qin et al. [2012] apresentaram um algoritmo que utiliza *Simulated Annealing* (SA) para o PILFC e mostraram que é um algoritmo competitivo para resolução do problema. Díaz [2008] propõem uma heurística Busca Dispersa com *Greedy Randomized Adaptive Search procedure* (GRASP) e Busca Tabu, para gerar limitantes superiores para problemas capacitados.

#### 4. Metodologia de Solução

Neste artigo, considerando que o problema PILFC é NP-difícil [Qin et al., 2012], é desenvolvido um algoritmo que adapta a metaheurística *Iterated Local Search* (ILS) para buscar boas soluções para o problema. O ILS alterna buscas locais com perturbações para intensificar e diversificar a busca por soluções. É uma heurística simples, robusta, efetiva, que contempla em generalidade problemas combinatórios e permite adaptações para considerar especificidades dos problemas [Lourenço et al., 2003].

Inicialmente, nas Subseções 4.1 e 4.2, descreve-se como uma solução é representada e avaliada. Em sequência, nas Subseções 4.3, 4.4 e 4.5, são apresentados os procedimentos de construção de solução inicial, busca local e perturbações que são utilizados no algoritmo. Por fim, na Subseção 4.6, é apresentado o algoritmo ILS adaptado para resolução do PLPC.

##### 4.1. Representação de uma solução

Uma solução  $s$  para o problema é representada como  $s = (x^s, y^s)$  em que  $x^s = (x_{ij}^s)$  é a matriz de variáveis de decisão  $x_{ij}$  de  $s$  e  $y^s = (y_j^s)$  é o vetor de variáveis de decisão  $y_j$  de  $s$ . Por exemplo,

$$s = (x^s, y^s), \text{ com } x^s = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ e } y^s = (1, 0, 1)$$

corresponde à solução para um problema com quatro pontos de coleta e três locais potenciais, em que os pontos de coleta 1, 2 e 3 são alocados no ponto de consolidação instalado no local potencial 1

e o ponto de coleta 4 é alocado no ponto de consolidação instalado no local potencial 3. No local potencial 2 não há instalação de ponto de consolidação.

#### 4.2. Avaliação de uma solução

Para avaliar as soluções do problema, é utilizada a função de avaliação:

$$f(s) = \sum_{j \in J} F_j y_j^s + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}^s + \alpha \sum_{j \in J} \text{Max} \left( \sum_{i \in I} d_i x_{ij}^s - m, 0 \right). \quad (6)$$

As duas primeiras parcelas de  $f(\cdot)$  correspondem aos custos totais de instalação de pontos de consolidação e de transporte de produtos dos pontos de coleta para pontos de consolidação. A última parcela penaliza, para cada ponto de consolidação, o excesso de produtos enviados além da capacidade por um fator  $\alpha$ , possibilitando avaliação de soluções infactíveis. Uma solução  $s$  é considerada melhor do que uma solução  $s'$  se  $f(s) < f(s')$ .

Devido ao alto custo computacional de avaliação de soluções utilizando-se a expressão (6), se uma solução  $s'$  é obtida modificando-se uma solução  $s$ , a solução  $s'$  é avaliada somando-se, a  $f(s)$ , as diferenças entre os custos de transporte, custos de instalação de pontos de consolidação e penalizações geradas pela modificação.

#### 4.3. Construção de uma solução inicial

A solução inicial do problema é denotada por  $s_0$ . Ela é determinada como a melhor de duas soluções auxiliares  $s_1$  e  $s_2$ . A solução  $s_1$  é construída alocando cada ponto de coleta no local potencial mais próximo e instalando nele um ponto de consolidação. Já a solução  $s_2$  é construída ordenando aleatoriamente os locais potenciais e alocando neles os pontos de coleta, um a um, seguindo esta ordem. Ambas as soluções desconsideram as restrições de capacidade. A solução  $s_1$  tende a ser melhor do que a solução  $s_2$  quando for factível, pois considera redução de custos de transporte em sua construção. Por outro lado,  $s_2$  tende a ter maior chance de ser factível, pois evita alocação de muitos pontos de coleta em um mesmo local potencial.

#### 4.4. Busca Local

Neste estudo, para aplicação de buscas locais, são utilizados cinco tipos de movimentos.

O movimento 1 consiste em realocar um ponto de coleta  $i$  em um local potencial  $j$ . Caso  $j$  não tenha um ponto de consolidação instalado, é feita a instalação. Por outro lado, caso um ponto de consolidação atenda apenas um ponto de coleta e este é realocado, este ponto de consolidação é desinstalado. O movimento 2 consiste em trocar o atendimento de dois pontos de coleta. Se, na solução corrente,  $j_1 \in J$  atende  $i_1 \in I$  e  $j_2 \in J$  atende  $i_2 \in I$ , após o movimento 2,  $j_2$  atenderá  $i_1$  e  $j_1$  atenderá  $i_2$ . Esse movimento mantém o mesmo número de pontos de consolidação instalados. O movimento 3 consiste em realocar todos os pontos de coleta que estão alocados em um ponto de consolidação para um outro ponto de consolidação. Com isso, o ponto de consolidação inicial é desinstalado. O movimento 4 consiste em realocar todos os pontos de coleta que são atendidos por um ponto de consolidação para um local potencial que não possui ponto de consolidação instalado. A consequência desse movimento é a troca do local de instalação de um ponto de consolidação, preservando os pontos de coleta atendidos por ele. O movimento 5 consiste em trocar a alocação de todos os pontos de coleta que são atendidos por um ponto de consolidação com todos os pontos de coleta que são atendidos por outro ponto de consolidação. Esse movimento mantém o mesmo número de pontos de consolidação instalados.

Esses movimentos estabelecem o conjunto  $\mathcal{N}$  de cinco estruturas de vizinhança  $\mathcal{N}_k$  com  $k = 1, \dots, 5$ . Um ótimo local com relação a uma estrutura de vizinhança não necessariamente é

---

**Algoritmo 1: Pipe-VND**

---

```
1 Entrada: solução  $s$ ,  $\mathcal{N}$ 
2 Saída: solução  $s$  refinada
3  $pare \leftarrow false$ 
4 enquanto  $pare = false$  faça
5      $k \leftarrow 1$ 
6      $s' \leftarrow s$ 
7     enquanto  $k \leq |\mathcal{N}|$  faça
8          $s'' \leftarrow$  melhor vizinho de  $s'$  com relação a  $\mathcal{N}_k$ 
9         se  $f(s'') < f(s')$  então
10              $s' \leftarrow s''$ 
11         senão
12              $k \leftarrow k + 1$ 
13         fim
14     fim
15     se  $f(s') > f(s)$  então
16          $s \leftarrow s'$ 
17     senão
18          $pare \leftarrow true$ 
19     fim
20 fim
21 Retorne  $s$ 
```

---

um ótimo local com relação a outra estrutura de vizinhança. Esta é a base para o algoritmo *Variable Neighborhood Descent* (VND), que explora estruturas de vizinhanças diferentes para encontrar ótimos locais com relação a estruturas de vizinhanças diferentes [Hansen e Mladenović, 2005].

Neste trabalho, a busca local é feita usando o algoritmo VND com o procedimento *Pipe Neighborhood Change Step* para alteração das estruturas de vizinhanças (Pipe-VND). O procedimento consiste em, a partir de uma ordem de estruturas de vizinhanças, manter a busca na estrutura de vizinhança corrente, até que não seja possível melhorar a solução com a estratégia de exploração adotada. Quando isso ocorrer, passa-se para a estrutura de vizinhança seguinte. A ordem das estruturas de vizinhanças do algoritmo desenvolvido segue a ordem de movimentos de menor complexidade para os de maior complexidade, procedimento frequentemente adotado na literatura [Hansen et al., 2010]. Para exploração de vizinhanças, é utilizada a estratégia *Best Improvement*, que consiste em buscar a melhor solução vizinha da solução corrente e mover-se para ela se ela for melhor do que a solução corrente. O pseudocódigo da busca local é apresentado no Algoritmo 1.

Nas linhas 4-20 consta o ciclo principal de repetição. Nele, a partir da primeira estrutura de vizinhança (linha 5), aplica-se, à solução corrente, o procedimento de intensificação do ciclo de repetição das linhas 7-14. Se há melhora na solução (linha 15), ela é atualizada (linha 16); caso contrário, encerra-se o ciclo de repetição principal (linha 18). O procedimento no ciclo das linhas 7-14 consiste em repetir, enquanto a última estrutura de vizinhança não for explorada (condição da linha 7), a busca pela melhor solução vizinha na estrutura de vizinhança corrente (linha 8). Se há melhora, atualiza-se a solução e mantém-se a busca na estrutura de vizinhança corrente (linhas 9 e 10); caso contrário, passa-se para a próxima estrutura de vizinhança (linha 12).

#### 4.5. Perturbações

Perturbações são movimentos que visam a diversificar a busca por soluções. O procedimento ILS é composto pelas componentes: solução inicial, busca local, perturbação e critério de aceitação [Lourenço et al., 2003]. Na abordagem deste estudo, ele é adaptado com a utilização de duas perturbações.

A perturbação 1 consiste em realocar um ponto de coleta para um local potencial, que pode ou não ter um ponto de consolidação instalado, ambos escolhidos aleatoriamente. A perturbação 1

no nível  $q \in \mathcal{N}^*$ , limitado pelo número de pontos de coleta, consiste em realocar  $q$  pontos de coleta para  $q$  locais potenciais, todos escolhidos aleatoriamente. Caso, após a perturbação 1, um ponto de consolidação que estava instalado não atenda nenhum ponto de coleta, ele é desinstalado. Esta perturbação tende a abrir novos pontos de consolidação, desorganizando o atendimento dos pontos de coleta, enquanto a busca local tende a diminuir o número de pontos de consolidação instalados, reorganizando o atendimento em busca de melhoria nas soluções. Entretanto, ao se utilizar apenas buscas locais como procedimentos de fechamento, a busca pode não diminuir suficientemente o número de pontos de consolidação abertos, deixando de explorar regiões que podem conter boas soluções. Por isso, é adotada a perturbação 2, que diminui o número de pontos de consolidação abertos.

A perturbação 2 consiste em fechar o ponto de consolidação que recebe a menor quantidade de produtos e realocar os pontos de coleta atendidos por ele, um a um, em outros pontos de consolidação, seguindo uma ordem aleatória. Este procedimento pode acarretar a violação das restrições de capacidade dos pontos de consolidação que recebem pontos de coleta adicionais. Por isso, a perturbação 2 é utilizada apenas quando o algoritmo atingir um certo número de iterações sem melhora e sem a consideração de níveis.

#### 4.6. Algoritmo ILS adaptado

O algoritmo desenvolvido neste trabalho é uma adaptação da metaheurística ILS, considerando, também, as adaptações de Coelho et al. [2016] e Reinsma et al. [2018]. O pseudocódigo do algoritmo é apresentado no Algoritmo 2.

Inicialmente, na linha 3, é aplicada a busca local à solução inicial. O ciclo principal (linhas 5-31) é repetido enquanto o número de iterações sem melhora da solução não ultrapassar o parâmetro de entrada  $ilsmax$  (linha 5). Nele, a partir do nível 1, aplica-se, à solução corrente, a perturbação 1 no nível corrente, seguida de busca local (linhas 7 e 8) e, periodicamente, aplica-se, também, o ciclo de repetição das linhas 12-20. Se, após esses procedimentos, há melhora da solução (linha 22), ela é atualizada e retorna-se para o nível 1 (linha 23); caso contrário, se o número de repetições sem melhora atingiu o parâmetro de entrada  $vezesmax$  (linha 25), aumenta-se o nível de perturbação (linha 26); senão, mantém-se o nível de perturbação. O ciclo das linhas 12-20 é aplicado quando o número de iterações sem melhora for múltiplo da metade do número máximo permitido (linha 10). Nele, a perturbação 2 é aplicada à solução corrente e, em seguida, é aplicada busca local (linhas 14-15). Se, após esses procedimentos, há melhora da solução, ela é atualizada (linhas 15-16); caso contrário, encerra-se o ciclo (linhas 17-19).

### 5. Experimentos Computacionais

Esta seção apresenta a metodologia e os resultados dos experimentos computacionais realizados com o algoritmo ILS adaptado para a resolução do PLPC. Inicialmente, nas subseções 5.1 e 5.2, são descritos como as instâncias foram geradas e os parâmetros calibrados, respectivamente. Posteriormente, na subseção 5.3, são apresentados a metodologia dos testes e os resultados obtidos. Por último, na subseção 5.4, é apresentada a análise de convergência do algoritmo.

#### 5.1. Geração de Instâncias

As instâncias foram geradas adaptando as ideias de Jayaraman et al. [2003]. Os pontos de coleta e de consolidação foram distribuídos aleatoriamente em uma malha  $300 \times 300$ . Os custos de transportes foram determinados proporcionalmente às distâncias euclidianas. A demanda de cada ponto de coleta foi gerada aleatoriamente no intervalo  $[1, 5]$ , correspondendo às quantidades de 1 a 5 toneladas. A capacidade para todos os pontos de consolidação foi determinada com fatores de ajuste iguais a 1 e 5 para simular, respectivamente, estruturas pequenas e grandes. Em instâncias com fator de ajuste 5, os pontos de consolidação têm capacidade cinco vezes maior do que em instâncias com

---

**Algoritmo 2: ILS adaptado**

---

```
1 Entrada:  $s_0, \mathcal{N}, ilsm_{\max}, vezes_{\max}$ 
2 Saída: solução  $s$ 
3  $s \leftarrow \text{Pipe-VND}(s_0, \mathcal{N})$ 
4  $iter \leftarrow 0; melhoriter \leftarrow 0; \text{nível} \leftarrow 1; nvezes \leftarrow 0$ 
5 enquanto  $iter - melhoriter \leq ilsm_{\max}$  faça
6    $iter \leftarrow iter + 1$ 
7    $s' \leftarrow \text{Perturbação1}(s, \text{nível})$ 
8    $s'' \leftarrow \text{Pipe-VND}(s', \mathcal{N})$ 
9    $s' \leftarrow s$ 
10  se  $(iter - melhoriter) \bmod (ilsm_{\max}/2) = 0$  então
11     $pare \leftarrow false$ 
12    enquanto  $pare = false$  faça
13       $s' \leftarrow \text{Perturbação2}(s'')$ 
14       $s''' \leftarrow \text{Pipe-VND}(s', \mathcal{N})$ 
15      se  $f(s''') < f(s'')$  então
16         $s'' \leftarrow s'''$ 
17      senão
18         $pare \leftarrow true$ 
19      fim
20    fim
21  fim
22  se  $f(s'') < f(s)$  então
23     $s \leftarrow s''; melhoriter \leftarrow iter; \text{nível} \leftarrow 1; nvezes \leftarrow 1$ 
24  senão
25    se  $nvezes = vezes_{\max}$  então
26       $\text{nível} \leftarrow \text{nível} + 1; nvezes \leftarrow 1$ 
27    senão
28       $nvezes \leftarrow nvezes + 1$ 
29    fim
30  fim
31 fim
32 Retorne  $s$ 
```

---

mesmo número de pontos de coleta, mesmo número de locais potenciais e fator de ajuste igual a 1. Os custos fixos de instalação em locais potenciais foram determinados como proporcionais à capacidade e à quantidade de pontos de coleta dentro de um raio de 10% da área total da malha. Este procedimento simula a existência de centros urbanos em locais com grande concentração de pontos de coleta. Custos fixos de instalação em centros urbanos são maiores, conforme propõem Alumur et al. [2012].

Para os testes computacionais foram geradas dezoito instâncias com tamanhos de 100 a 800. Elas foram nomeadas de acordo com o número de pontos de coleta, número de locais potenciais e fator de ajuste de capacidade. Por exemplo, a instância 80-20-5 é uma instância com 80 pontos de coleta, 20 locais potenciais e fator de ajuste igual a 5.

## 5.2. Calibração de Parâmetros

Os valores dos parâmetros  $ilsm_{\max}$  e  $vezes_{\max}$  do algoritmo ILS adaptado foram calibrados pela ferramenta IRACE (*Iterated Racing for Automatic Algorithm Configuration*) [López-Ibáñez et al., 2016]. As instâncias para treinamento foram 80-20-1, 250-50-5 e 500-100-1 e o número máximo de experimentos para calibração foi 1000. A Tabela 1 mostra os resultados desse treinamento. Os valores retornados pela calibração constam em negrito.

O valor utilizado para o fator de penalidade  $\alpha$  da Expressão (6) foi  $10^5$ . Ele foi determinado empiricamente, de forma que qualquer solução factível seja melhor do que qualquer solução infactível.

Tabela 1: Calibração de Parâmetros

Parâmetros da Calibração	Descrição	Valores Testados
ilsmax	número máximo de iterações sem melhora	{20, <b>30</b> , 40}
vezesmax	número máximo de vezes em cada nível	{ <b>1</b> , 2, 3, 4}

Tabela 2: Resultados

Instância	CPLEX		ILS adaptado		
	Valor Ótimo	$t_{\text{CPLEX}}$ (s)	$\Delta_{\text{melhor}}$ (%)	$\Delta_{\text{médio}}$ (%)	$t_{\text{médio}}$ (s)
80-20-5	4445,9	0,5	0,00	0,00	1,94
80-20-1	4691,65	8,2	0,03	0,24	2,41
160-40-5	8950,08	35,5	0,00	0,02	5,78
160-40-1	8341,29	2205,1	0,22	0,25	21,9
250-50-5	11824,0	17,2	0,07	0,13	43,45
250-50-1	14361,4	19431,5	0,83	1,01	79,41
200-100-5	1734,39	46,44	0,16	0,61	18,37
200-100-1	1432,06	4927,1	1,10	1,78	42,41
320-80-5	14401,7	52,19	0,33	0,60	29,05
320-80-1	18590,1	11303,4	0,34	0,71	17,84
400-100-5	17933,8	2990,7	0,13	0,22	125,66
400-100-1	21494,2	6426,4	0,64	0,86	89,68
500-100-5	21709,3	380,5	0,09	0,11	295,93
500-100-1	26659,2	13422,2	0,50	0,59	151,11
550-150-5	23271,1	1423,5	0,42	0,73	201,32
550-150-1	29767,4	24241,7	0,42	0,62	179,69
650-150-5	10890,1	9150,5	0,22	0,38	354,47
650-150-1	13530,6	10432,7	0,46	0,54	739,4

### 5.3. Metodologia dos Experimentos e Resultados

Para cada instância foram realizadas dez execuções do algoritmo com os parâmetros calibrados. Foram considerados o melhor resultado obtido  $f_{\text{melhor}}$ ; a média dos resultados  $f_{\text{médio}}$ ; e o tempo médio de execução  $t_{\text{médio}}$ . Foram obtidos, também, o valor ótimo  $f^*$  e o tempo de execução  $t_{\text{CPLEX}}$ , utilizando o *solver* CPLEX versão 12.9. Para avaliar a qualidade das soluções, calculou-se os desvios relativos percentuais de  $f_{\text{melhor}}$  e  $f_{\text{médio}}$  com relação a  $f^*$  como:

$$\Delta_{\text{melhor}}(\%) = \frac{f_{\text{melhor}} - f^*}{f^*} \times 100, \quad \Delta_{\text{médio}}(\%) = \frac{f_{\text{médio}} - f^*}{f^*} \times 100.$$

Todos os algoritmos foram implementados na linguagem C++ e testados em um computador Intel Core i5-4210U 2.40 GHz, 64 bits, 8 GB de memória RAM e sistema operacional Ubuntu 18.04.5 LTS. Os resultados dos experimentos são apresentados na Tabela 2.

Os valores obtidos para  $\Delta_{\text{médio}}(\%)$  variaram entre 0 a 1,78 com média de 0,52. Isto indica que o algoritmo gerou boas soluções para o problema. Os resultados de  $\Delta_{\text{melhor}}(\%)$  mostram que, para duas instâncias, a heurística encontrou o valor ótimo para o problema em pelo menos uma execução. A média dos valores de  $t_{\text{CPLEX}}$  e  $t_{\text{médio}}$  foram, respectivamente, 5916,40 e 133,32 segundos, o que mostra que o custo computacional da heurística foi muito inferior ao do *solver* CPLEX.

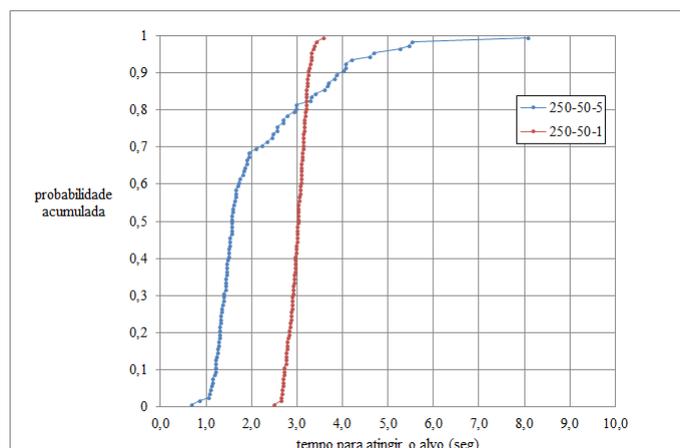


Figura 1: Gráficos de Distribuição de Probabilidade Empírica.

#### 5.4. Análise de Convergência

Para análise de convergência do algoritmo são utilizados gráficos de distribuição de probabilidades empíricas acumuladas de o algoritmo atingir valores alvos em certos tempos de execução, conforme descrito em Aiex et al. [2007]. Para esta análise, foram feitas 100 execuções do algoritmo para as instâncias 250-50-5 e 250-50-1, considerado o valor alvo de 2% acima do valor ótimo. Para cada instância, os tempos de execução para atingir o valor alvo foram ordenados de forma crescente. Posteriormente, foi associado, a cada tempo  $t_i$  de posição  $i$ , a probabilidade empírica  $p(i) = \frac{1}{100} (i - \frac{1}{2})$ , e construído o gráfico  $\{(t_i, p(i)) : i = 1, \dots, 100\}$ . A Figura 1 mostra, em um mesmo sistema de eixos, as distribuições de probabilidades empíricas acumuladas para as duas instâncias. Os gráficos mostram, para as duas instâncias, que o algoritmo tem alta probabilidade de atingir o valor alvo em poucos segundos.

#### 6. Conclusões

A resolução do PLPC exige a determinação da quantidade de pontos de consolidação, dos locais para instalação e de como as demandas devem ser alocadas. O algoritmo proposto visa a resolver essas questões, intensificando a busca de soluções por uma busca local que explora diferentes estruturas de vizinhanças e diversificando a busca com perturbações, que abrem e fecham pontos de consolidação. Os resultados mostram que o algoritmo ILS adaptado gera boas soluções para o PLPC com tempo computacional bem inferior ao do *solver* CPLEX. Isto mostra que a heurística é, também, uma boa opção para gerar uma solução inicial para o PLPC no desenvolvimento de algoritmos exatos. Além disso, a análise de convergência sinaliza que a heurística tem alta probabilidade de gerar boas soluções com baixo tempo computacional.

O próximo passo deste estudo é aprimorar o algoritmo. Para isso, propõe-se incorporar procedimentos exatos para abertura e fechamento de facilidades e realocação de clientes. Pretende-se, também, utilizar a heurística desenvolvida para gerar soluções iniciais para algoritmos exatos para o problema. Para diminuição do tempo computacional de execução, propõe-se explorar vizinhanças, analisando somente vizinhos com maior potencial de melhora de soluções. Em trabalhos futuros, pretende-se testar a heurística desenvolvida para instâncias *benchmarks* do PILFC e comparar seu desempenho com outras heurísticas da literatura.

Problemas de localização em redes logísticas reversas tendem a se tornar ainda mais complexos à medida que suas implantações avançam. Certamente, serão foco de muitos trabalhos futu-

ros, principalmente considerando a realidade brasileira, que é um país de grande extensão territorial e os sistemas estão em fase de implantação.

### Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, código de financiamento 001), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG, processo PPM CEX 0676/17), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, processo 303266/2019-8), ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) e à Universidade Federal de Ouro Preto (UFOP) pelo apoio ao desenvolvimento deste trabalho.

### Referências

- Aiex, R., Resende, M., e Ribeiro, C. (2007). Ttt plots: a perl program to create time-to-target plots. *Optimization Letters*, 1:355–366.
- Akinc, U. e Khumawala, B. M. (1977). An efficient branch and bound algorithm for the capacitated warehouse. *Management Science*, 23(6):992–1009.
- Alumur, S., Nickel, S., da Gama, F., e Verter, V. (2012). Multi-period reverse logistics network design. *Multi-period reverse logistics network design*, 220(1):67–68.
- Avella, P., Boccia, M., Sforza, A., e Vasil'ev, I. (2009). An effective heuristic for large-scale capacitated facility location problems. *Journal of Heuristics*, 15:597–615.
- Beasley, J. E. (1988). An algorithm for solving large capacitated warehouse location problems. *European Journal of Operational Research*, 33(3):314–325.
- Brasil (2020). Decreto n. 10.240, de 12 de fevereiro de 2020. <https://www.in.gov.br/en/web/dou/-/decreto-n-10.240-de-12-de-fevereiro-de-2020-243058096>. Acessado em 01/05/2021.
- Coelho, E. K. F. e Mateus, G. R. (2017). A capacitated plant location model for reverse logistics activities. *Journal of Cleaner Production*, 167:1165–1176.
- Coelho, V., Grasas, A., Ramalhinho, H., Coelho, I., Souza, M., e Cruz, R. (2016). An ils-based algorithm to solve a large-scale real heterogeneous fleet vrp with multi-trips and docking constraints. *European Journal of Operational Research*, 250(2):367–376.
- Díaz, I. A. C. . J. A. (2008). Scatter search for the single source capacitated facility location problem. *Annals of Operations Research*, 157:73–89.
- Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009.
- Forti, V., Baldé, C. P., Kuehr, R., e Bel, G. (2020). The global e-waste monitor 2020: Quantities, flows and the circular economy potential. Technical report, United Nations University (UNU)/United Nations Institute for Training and Research (UNITAR) – co-hosted SCYCLE Programme, International Telecommunication Union (ITU) & International Solid Waste Association (ISWA), Bonn/Geneva/Rotterdam. URL [http://ewastemonitor.info/wp-content/uploads/2020/12/GEM\\_2020\\_def\\_dec\\_2020-1.pdf](http://ewastemonitor.info/wp-content/uploads/2020/12/GEM_2020_def_dec_2020-1.pdf).

- Galvão, R. e Raggi, L. (1989). A method for solving to optimality uncapacitated location problems. *Annals of Operations Research*, 18:225–244.
- Hansen, P., Brimberg, J., Urošević, D., e Mladenović, N. (2007). Primal-dual variable neighborhood search for the simple plant-location problem. *INFORMS Journal on Computing*, 19(4):552–564.
- Hansen, P. e Mladenović, N. (2005). *Variable Neighborhood Search*, p. 211–238. Springer US, Boston, MA.
- Hansen, P., Mladenović, N., Brimberg, J., e Pérez, J. A. M. (2010). *Variable Neighborhood Search*, volume 146, chapter 3, p. 61–86. Springer.
- Jacobsen, S. K. (1983). Heuristics for the capacitated plant location model. *European Journal of Operational Research*, 12(3):253–261.
- Jayaraman, V., Patterson, R., e Rolland, E. (2003). The design of reverse distribution networks: models and solution procedures. *European Journal of Operational Research*, 150(1):128–149.
- Kratica, J., Tošić, D., Filipović, V., e Ljubić, I. (2001). Solving the simple plant location problem by genetic algorithm. *RAIRO Operational Research*, 35(1):127–142.
- 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.
- Lourenço, H., Martin, O., e Stützle, T. (2003). Iterated local search. In Glover, F. e Kochenberger, G., editors, *Handbook of Metaheuristics*. Springer.
- Qin, J., Ni, L., e Shi, F. (2012). Combined simulated annealing algorithm for the discrete facility location problem. *The Scientific World Journal*, 2012:576392.
- Reinsma, J. A., Penna, P. H. V., e Souza, M. J. F. (2018). Um algoritmo simples e eficiente para resolução do problema do caixeiro viajante generalizado. In *Anais do L SBPO*, Rio de Janeiro. SOBRAPO.
- Resende, M. G. C. e Werneck, R. F. (2006). A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, 174(1):54–68.
- Rogers, D. S. e Tibben-Lembke, R. (2001). An examination of reverse logistics practices. *Journal of Business Logistics*, 22(2):129–148.
- Sun, M. A. (2012). A tabu search heuristic procedure for the capacitated facility location problem. *Journal of Heuristics*, 18:91–118.
- União Europeia (2018). Waste from electrical and electronic equipment directive. Disponível em [https://ec.europa.eu/environment/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee\\_pt](https://ec.europa.eu/environment/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee_pt). Acessado em 01/05/2021.