



Uma Abordagem Híbrida para a Resolução do Problema de Localização e Roteamento com Empacotamento Bidimensional

Kamyla Maria Ferreira e Thiago Alves de Queiroz

Unidade de Matemática e Tecnologia - UFG/Regional Catalão.

Av. Dr. Lamartine Pinto de Avelar, 1120, Setor Universitário, 75704-020, Catalão-GO, Brasil.

kamylamaaria@gmail.com

taq@ufg.br

RESUMO

Este trabalho propõe uma heurística híbrida para a resolução do problema de localização e roteamento com empacotamento bidimensional, com aplicação em problemas de desastres naturais. A heurística proposta é composta pelo método de recozimento simulado e o algoritmo artificial de algas. O método de recozimento simulado é utilizado para resolver o problema de localização e roteamento na sua versão capacitada, isto é, em que os depósitos e os veículos possuem capacidade limitada. Para cada rota do problema de localização e roteamento, aplica-se o algoritmo artificial de algas para determinar a sequência em que os itens dos clientes vão ser empacotados. O problema de empacotamento estudado não considera a ordem dos clientes na rota para empacotar os itens retangulares no chão do veículo. Para analisar a heurística proposta, adapta-se um conjunto de instâncias padrão do problema de localização e roteamento. Os resultados obtidos mostram que a heurística híbrida é competitiva para grande parte das instâncias.

PALAVRAS CHAVE. Problema de Localização e Roteamento. Problema de Empacotamento Bidimensional. Recozimento Simulado.

Tópicos: L&T - Logística e Transportes. MH - Meta-heurísticas

ABSTRACT

This work proposes a hybrid heuristic for solving the location-routing problem with two-dimensional loading constraints, with application on natural disasters problems. The heuristic proposed is composed by the simulated annealing and the artificial algae algorithm. The simulated annealing method is used to solve the location-routing problem in its capacitated version, that is, the depots and vehicles have limited load capacity. For each route of the location-routing problem, it is applied the artificial algae algorithm to determine the sequence in which customers' items are packed. The packing problem under consideration does not consider the order in which customers are visited in the route when packing the rectangular items on the vehicle's floor. The analysis of the heuristic proposed is made over a set of instances adapted from the location-routing problem. The results show that the hybrid heuristic is competitive for most of the instances solved.

KEYWORDS. Location-Routing Problem. Two-Dimensional Orthogonal Packing Problem. Simulated Annealing.

Paper Topics: L&T - Logistics and Transportation. MH - Metaheuristics



1. Introdução

Os desastres naturais, como por exemplo terremotos, furações e tsunamis, provocam grande destruição. Pode-se mencionar a destruição de cidades, abalo na infraestrutura da região, desabrigo de populações, ferimentos graves nos indivíduos, propagação de doenças, fome, mortes e muitas outras consequências. Após os desastres naturais, as vítimas costumam ser alojadas em abrigos e recebem bens de socorro, tais como: comida, medicamentos, produtos de higiene, roupas, cobertores, entre outros cuidados. Assim, as autoridades precisam estabelecer rapidamente novos depósitos intermediários e um sistema de rotas para distribuir as mercadorias nos abrigos, contudo também é necessário determinar como as mercadorias vão ser empacotadas nos veículos de forma a saber o número correto de veículos necessários para suprir as demandas solicitadas.

O problema de estabelecer novos depósitos, definir as rotas e realizar o empacotamento das mercadorias é denominado por problema de localização e roteamento com restrições de empacotamento bidimensional, do inglês *Location-Routing Problem with Two-Dimensional Loading* (2L-LRP). O 2L-LRP é uma combinação de dois outros problemas, sendo o problema de localização e roteamento (*Location-Routing Problem - LRP*), e o problema de roteamento de veículos com restrições de empacotamento bidimensional (*Vehicle Routing Problem with Two-Dimensional Loading Constraints - 2L-CVRP*).

O 2L-LRP envolve os três níveis hierárquicos de decisões do sistema de gestão logística, quais sejam: (i) decisão estratégica, que está relacionada a localização das instalações; (ii) decisão tática, que busca associar os clientes aos depósitos e pode incluir a determinação das rotas; e, por fim, (iii) decisão operacional, que apesar de poder incluir o roteamento, está relacionada a forma de empacotar os itens nos veículos. Ao considerar um problema que envolve desastres naturais, o horizonte de tempo para tomar as decisões estratégicas, táticas e operacionais é pequeno, o que justifica a integração dos problemas de localização, roteamento e empacotamento.

O 2L-LRP tem como objetivo definir onde os depósitos devem ser abertos e quais as rotas de veículos devem ser estabelecidas, a partir desses depósitos, para atender a demanda de todos os clientes. Neste caso, os clientes têm uma demanda por itens, de base retangular e que não podem ser empilhados por questões de fragilidade, que devem ser empacotados na base retangular do veículo de forma viável. O objetivo do problema é minimizar o custo total associado a abertura dos depósitos e as rotas estabelecidas. No que diz respeito a complexidade, o 2L-LRP é considerado NP-difícil, uma vez que constitui-se em uma junção de dois problemas NP-difíceis [Garey e Johnson, 1979].

Na literatura, a resolução do 2L-LRP aparece no trabalho de Queiroz et al. [2016], que estudaram o problema considerando a versão capacitada e com demanda incerta, isto é, a capacidade dos veículos e depósitos é limitada e as demandas dos clientes não são conhecidas a princípio. Os autores desenvolveram um modelo de programação inteira para o problema, sendo que a validação do modelo foi realizada por meio da adaptação de um caso real sob três cenários diferentes de incertezas.

No que diz respeito a resolução do 2L-CVRP por métodos heurísticos, pode-se citar: Gendreau et al. [2008], que desenvolveram uma meta-heurística do tipo busca tabu para resolver a versão do 2L-CVRP em que os veículos tem capacidade de carga limitada. O algoritmo foi aplicado em cinco classes distintas de instâncias, sendo que cada classe é composta por 36 instâncias; Leung et al. [2010], que propuseram uma heurística baseada no recozimento simulado para resolver as versões irrestrita e sequencial. Os experimentos computacionais foram realizados em um total de 180 instâncias para cada versão tratada. Um apanhado geral sobre trabalhos envolvendo o 2L-CVRP e suas variantes pode ser encontrado em Iori e Martello [2013].

A resolução do LRP com aplicação em desastres naturais está evidenciada nos trabalhos de Song et al. [2009], Pramudita et al. [2014] e Rath e Gutjahr [2014]. Song et al. [2009] trataram o problema de evacuação das vítimas após algum tipo de desastre natural, tendo como objetivo minimizar o tempo total de evacuação. Os autores utilizaram uma heurística híbrida baseada no



algoritmo genético com redes neurais artificiais, com testes sendo realizados em um caso prático da cidade de Gulfport - Estados Unidos. Pramudita et al. [2014] utilizaram a busca tabu para resolver o problema de retirada de destroços. O problema consiste em traçar as rotas para coletar os destroços e determinar o local onde eles serão despejados. Os experimentos computacionais foram realizados em instâncias baseadas em um desastre ocorrido em Tóquio - Japão. Rath e Gutjahr [2014] estudaram a versão do LRP que busca definir o local onde os depósitos intermediários devem ser abertos e determinar as rotas que devem fornecer bens de socorro para as vítimas acidentadas. Os métodos desenvolvidos pelos autores foram aplicados em um conjunto de 21 instâncias geradas sinteticamente e em um caso real da província de Manabí - Equador.

Neste trabalho, busca-se resolver o 2L-LRP na sua versão capacitada, em que os depósitos e veículos têm capacidade limitada. Além disso, os itens não precisam ser empacotados considerando a ordem de descarregamento, ou seja, os itens podem ser remanejados durante o descarregamento. A resolução do 2L-LRP é feita por meio da aplicação de uma heurística híbrida composta pelo recozimento simulado (*Simulated Annealing* - SA) e o algoritmo artificial de algas (*Artificial Algae Algorithm* - AAA). O SA é utilizado para resolver o LRP, enquanto AAA é aplicado para obter a solução do problema de empacotamento ortogonal de itens retangulares, neste caso, sobre uma malha de *reduced raster points*.

A próxima seção define com mais detalhes o 2L-LRP, enquanto a Seção 3 descreve o método utilizado para resolver esse problema. Os experimentos computacionais são discutidos na Seção 4, enquanto que as considerações finais são apresentadas na Seção 5.

2. Caracterização do 2L-LRP

O 2L-LRP pode ser definido em um grafo não orientado $G = (V, E)$, em que V é o conjunto de vértices, composto pelo subconjunto de depósitos $I = \{1, 2, \dots, M\}$ e o subconjunto de clientes $J = \{1 + M, 2 + M, \dots, N + M\}$ e E corresponde as arestas que ligam dois vértices de V , exceto depósito com depósito, com um custo c por aresta. Cada depósito $i \in I$ tem uma capacidade W_i e custo de abertura O_i . Além disso, tem-se um conjunto de K veículos idênticos com capacidade máxima de carga Q e um custo fixo de utilização Z . O veículo tem uma base de dimensões retangulares, sendo que L é a largura e C é o comprimento. Assim, a área total do veículo é $A_t = LC$. Cada cliente $j \in J$ tem uma demanda de R_j itens retangulares, com peso total d_j e área total a_j . Ademais, cada item $r \in R$ tem largura l_{jr} , comprimento h_{jr} , área a_{jr} e peso d_{jr} .

O 2L-LRP tem as seguintes restrições: a capacidade dos depósitos e dos veículos devem ser respeitadas; cada rota tem início e fim no mesmo depósito; um veículo só pode executar uma única rota; a demanda de todos os clientes deve ser atendida; a demanda de um cliente deve ser suprida por um único veículo e em uma única visita; e, os itens devem ser empacotados na base do veículo sem sobreposição e respeitando as dimensões retangulares da base. O objetivo é minimizar o custo total proveniente da abertura de depósitos, do transporte pelas rotas e do uso de veículos.

3. Caracterização da Heurística

Propõe-se uma heurística híbrida, denominada por SA+AAA, apresentada no Algoritmo 1. A heurística híbrida é baseada nos seguintes métodos: SA, que faz uma analogia ao processo de resfriamento de um metal em estado de fusão [Kirkpatrick et al., 1983]; e AAA, que simula a vida das algas que consiste em: seu modo de mover/nadar, dado por um movimento helicoidal; adaptação a novos ambientes; e, a reprodução por divisão mitótica [Uymaz et al., 2015]. A heurística busca a cooperação entre os dois métodos, sendo que sempre que o SA consegue melhorar a solução do LRP, a mesma é considerada no AAA, que busca realizar o empacotamento dos itens nas rotas.



Algoritmo 1- Estrutura geral da heurística SA+AAA.

Entrada: I_{iter} , $N_{\text{não-melhoria}}$, T_0 , T_F , α , B , P , t_{max} ;

- 1 Gerar uma **Solução Inicial** X ;
- 2 $T \leftarrow T_0$; $N \leftarrow 0$; $F \leftarrow obj_{LRP}(X, P)$;
- 3 $F_{\acute{o}timo} \leftarrow F$; $X_{\acute{o}timo} \leftarrow X$; $S_{\acute{o}timo} \leftarrow S$; $F_{ant} \leftarrow F_{\acute{o}timo}$;
- 4 **repita**
- 5 **para** $I = 0$ até I_{iter} **faça**
- 6 $t \leftarrow$ número inteiro gerado aleatoriamente entre 1 e 5;
- 7 A partir de t escolher um **Operador de Vizinhança** e gerar Y de X ;
- 8 $F_1 \leftarrow obj_{LRP}(Y, P)$;
- 9 **se** $F_1 < F_{\acute{o}timo}$ **então** $S' \leftarrow$ para cada rota da solução Y aplique o AAA;
- 10 **senão** $S' \leftarrow$ área total da base do veículo;
- 11 $FY \leftarrow obj_{2L-LRP}(F_1, S', P)$;
- 12 **se** $FY - F \leq 0$ **então** $F \leftarrow FY$; $X \leftarrow Y$; $S \leftarrow S'$;
- 13 **senão** $r \leftarrow$ número gerado aleatoriamente entre 0 e 1;
- 14 **se** $r < e^{-(FY-F)/(B \cdot T)}$ **então** $F \leftarrow FY$; $X \leftarrow Y$; $S \leftarrow S'$;
- 15 **se** $F < F_{\acute{o}timo}$ e X for viável **então**
- 16 $X_{\acute{o}timo} \leftarrow X$; $S_{\acute{o}timo} \leftarrow S$; $F_{\acute{o}timo} \leftarrow F$; $N \leftarrow 0$;
- 17 $T \leftarrow \alpha \cdot T$; $N \leftarrow N + 1$; $X \leftarrow X_{\acute{o}timo}$; $F \leftarrow F_{\acute{o}timo}$;
- 18 **se** $F_{ant} \neq F_{\acute{o}timo}$ **então**
- 19 $F_{ant} \leftarrow F_{\acute{o}timo}$;
- 20 Aplicar a rotina de **Busca Local** em $X_{\acute{o}timo}$;
- 21 Aplicar a heurística de **Lin-Kernighan** em $X_{\acute{o}timo}$;
- 22 $X \leftarrow X_{\acute{o}timo}$;
- 23 $X \leftarrow$ Aplicar a operação de **Perturbação** em X ;
- 24 **se** $(nn \bmod 5) = 0$ **então** $X \leftarrow$ Aplicar a operação de **Diversificação** em X ;
- 25 $F_1 \leftarrow obj_{LRP}(X, P)$;
- 26 **se** $F_1 < F_{\acute{o}timo}$ **então** $S \leftarrow$ para cada rota da solução X aplique o AAA;
- 27 **senão** $S \leftarrow$ área total da base do veículo;
- 28 $F \leftarrow obj_{2L-LRP}(F_1, S, P)$;
- 29 **se** $F < F_{\acute{o}timo}$ e X for viável **então** $X_{\acute{o}timo} \leftarrow X$; $S_{\acute{o}timo} \leftarrow S$; $F_{\acute{o}timo} \leftarrow F$;
- 30 **até** $T \geq T_F$ e $N < N_{\text{não-melhoria}}$ e $time < t_{max}$;

Saída: $F_{\acute{o}timo}$

A heurística SA+AAA tem oito parâmetros de entrada, em que: I_{iter} indica o número de iterações para cada temperatura; $N_{\text{não-melhoria}}$ representa o número de reduções consecutivas que podem ser realizadas na temperatura sem que a solução $X_{\acute{o}timo}$ melhore; T_0 e T_f correspondem a temperatura inicial e final do processo, respectivamente; α é o coeficiente de resfriamento da temperatura; B é a constante usada na função de aceitação de soluções ruins; P é a penalidade aplicada quando ocorre violação da capacidade do depósito e/ou quando todos os itens não podem ser empacotados em uma rota; e, t_{max} é o tempo máximo de execução em segundos.

No início do Algoritmo 1, gera-se uma solução inicial X conforme o procedimento descrito na Subseção 3.2 e calcula-se o valor da mesma. Nesta etapa não é preciso verificar se o empacotamento é viável, uma vez que na solução inicial cada veículo atende um único cliente e, assim, todas as rotas são viáveis. Em seguida, obtém-se uma nova solução Y vizinha de X conforme a aplicação de um dos operadores descritos na Subseção 3.3. Após a geração da nova solução, verifica-se o custo associado ao problema LRP e se o custo for menor do que $F_{\acute{o}timo}$, então se verifica a viabilidade do empacotamento aplicando o AAA. Caso contrário, S' recebe a área total da base do veículo. Posteriormente, faz-se a avaliação da solução Y considerando também o empacotamento, sendo que ela é aceita caso tenha um custo menor ou se o valor da função de aceitação $e^{-(FY-F)/(B \cdot T)}$ for menor do que um número aleatório r .

No momento em que I_{iter} iterações são executadas, caso $F_{\acute{o}timo}$ seja diferente do valor



anterior, a solução $X_{ótimo}$ passa pela rotina de busca local e, em seguida, a heurística de Lin-Kernighan. Posteriormente, aplica-se uma função de Perturbação visando explorar outras regiões do espaço de busca. Além disso, após alcançar o critério $(nn \bmod 5) = 0$, a solução X passa pela função de Diversificação, também como alternativa para explorar outras regiões do espaço de busca. A heurística híbrida chega ao fim quando a temperatura T for menor ou igual a T_f , ou se a melhor solução $X_{ótimo}$ não for melhorada depois de $N_{não-melhoria}$ reduções de temperatura ou quando o tempo limite t_{max} for atingido.

3.1. Função de Avaliação

Uma solução X é avaliada conforme (1), sendo que $y_i = 1$ representa que o depósito i foi aberto $x_{ijk} = 1$ indica que o veículo k percorreu a aresta (i, j) .

$$\sum_{i \in I} O_i y_i + P \sum_{i \in I} \psi(i) + \sum_{(i,j) \in E} \sum_{k \in K} c_{ij} x_{ijk} + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} Z x_{ijk} + P \max \left\{ 0, \left(\sum_{j \in J} \sum_{r \in R_j} a_{jr} - \phi \right) \right\}^2, \quad (1)$$

A primeira parcela em (1) corresponde ao custo de abertura do depósito, enquanto que a segunda se refere a penalidade aplicada toda vez que a capacidade do depósito for ultrapassada. Para realizar o cálculo da penalidade, seja $\mu(i)$ a diferença entre a soma da demanda de todos os clientes atendidos pelo depósito i e a capacidade desse depósito. Assim, $\psi(i) = \max\{0, \mu(i)\}^2$, que usa a ideia da função de penalidade exterior. Ou seja, quando todas as restrições forem atendidas, a função pseudo-objetivo apresenta em (1) equivale à função objetivo do problema original. Contudo se alguma restrição for violada, o quadrado dessa violação é adicionado à função pseudo-objetivo como forma de impor que o método convirja para uma solução viável.

As três últimas parcelas em (1) estão relacionadas ao 2L-CVRP, em que a terceira parcela está associada ao custo de transporte, que consiste na distância percorrida por todas as rota. A quarta parcela está relacionada com o custo fixo de se utilizar veículos e, por fim, a última parcela representa a penalidade aplicada quando os itens dos clientes não podem ser empacotados. Neste caso, calcula-se a diferença entre a área total de todos os itens com a área total ϕ dos itens empacotados nas rotas (somatório de todos os S' de uma solução X) e aplica-se a ideia da penalidade exterior.

No Algoritmo 1, o valor de obj_{LRP} consiste na soma das quatro primeiras parcelas da (1). Por outro lado, o valor retornado por obj_{2L-LRP} consiste na soma de todas as parcelas de (1).

3.2. Solução Inicial

A solução inicial do 2L-LRP é gerada por meio de um procedimento guloso, que consiste em:

- **Passo 1:** Para cada depósito $i \in I$ fechado, calcula-se o valor $\Omega(i)$, que corresponde ao número de clientes ainda não atendidos e com menor custo c_{ij} .
- **Passo 2:** Selecione o depósito que vai ser aberto, com maior $\Omega(i)$, e adicione os clientes ainda não atendidos ao depósito i conforme o menor custo. Os clientes vão sendo inseridos enquanto a capacidade do depósito não for violada.
- **Passo 3:** Ao atingir a capacidade do depósito i , volte ao Passo 1 e repita os passos até que todos os clientes sejam atribuídos a algum dos depósitos.

Ao término do procedimento guloso, a solução é convertida na representação de um vetor, que é composto por uma sequência de números podendo ser os depósitos em I , os clientes em J e os zeros do N_{zero} , sendo que a primeira posição do vetor solução sempre deve ser um depósito. O N_{zero} corresponde a quantidade máxima de rotas que podem ser realizadas, que no caso é igual ao número de clientes, de modo que a capacidade e a dimensão do veículo não seja ultrapassada. Assim, no pior caso, cada veículo vai atender um único cliente.



3.3. Operadores de Vizinhança

São utilizados cinco operadores de vizinhança para explorar o espaço de soluções, os quais realizam modificações na posição dos elementos presentes no vetor solução. As operações de vizinhança realizam mudanças sob um única posição ou em uma sequência do vetor, as quais são escolhidas de forma aleatória. O tamanho da sequência pode ter no máximo a metade do tamanho do vetor solução.

As operações de vizinhança são: (1) Inserção, que consiste em inserir o elemento da posição i imediatamente antes da posição j ; (2) Inserção de uma sequência, que tenta inserir uma sequência antes da posição j ; (3) Troca, que efetua a troca dos elementos nas posições i e j ; (4) Troca de duas sequências, que busca realizar a troca entre duas sequências; e, por fim, (5) Inversão, em que há a inversão de ordem nos elementos de uma sequência.

3.4. Busca Local e Lin-Kernighan

A busca local é composta por duas operações, a Inserção e a Troca. Na busca local de inserção, realiza-se a operação de Inserção entre todas as posições $\{i, j\}$, sendo que a primeira posição do vetor solução não pode ser modificada. O procedimento termina após realizar todos os movimentos de inserção, ou quando o custo de uma das soluções for menor do que $F_{ótimo}$ e a solução for viável.

Em seguida, aplica-se a busca local de troca que consiste em aplicar a operação de Troca considerando todas as posições i e j do vetor solução. A única restrição é que a primeira posição do vetor só pode ser trocada com outra que contenha um depósito. O procedimento encerra quando uma das trocas resultar em uma solução melhor do que $F_{ótimo}$ e viável, ou até que todas as trocas possíveis sejam realizadas.

Após a aplicação da busca local, aplica-se a heurística de Lin-Kernighan, que é rápida e eficiente para melhorar o custo de uma única rota. Essa heurística é baseada na ideia de $\lambda - opt$, em que a quantidade de arcos modificados em cada passo é determinada dinamicamente e varia conforme o valor adotado para λ [Lin e Kernighan, 1973].

3.5. Perturbação e Diversificação

A função de Perturbação é aplicada cada vez que a temperatura diminui de valor. Ela considera duas rotas do vetor solução, ambas escolhidas de forma aleatória, em que a primeira rota tem início na posição r_{ini}^1 e fim em r_{fim}^1 e segunda rota começa na posição r_{ini}^2 e termina em r_{fim}^2 . Em cada uma dessas rotas, seleciona-se uma posição, por exemplo, a posição i na primeira rota e a j na segunda rota. O movimento consiste em inserir os elementos da sequência j até r_{fim}^2 imediatamente antes da posição r_{ini}^1 . Os elementos da sequência i até r_{fim}^1 são adicionados na segunda rota antes da posição que era marcada por j .

A operação de Diversificação busca aumentar ainda mais a probabilidade de modificar a solução e, assim, sair de um ótimo local. A Diversificação considera executar dois operadores consecutivamente, quais sejam: Inserção de duas sequências e Troca inversa. A Inserção de duas sequências considera duas sequências aleatórias em que a primeira é dividida de forma aleatória e tem a primeira metade inserida imediatamente antes do início da segunda sequência, ao passo que a outra metade é adicionada após o término da segunda sequência. Na operação de Troca inversa, seleciona-se aleatoriamente duas sequências que são trocadas de posição e em ordem inversa.

3.6. Algoritmo Artificial de Algas

O Algoritmo 2 apresenta a versão do AAA utilizado para resolver o problema de empacotamento ortogonal bidimensional seguindo a versão em Zhang et al. [2016]. O AAA tem como objetivo obter uma solução com a maior área empacotada possível. Para tanto, gera-se a ordem em que os itens vão ser empacotados no recipiente. A estratégia de empacotamento é uma adaptação da heurística apresentada por Wei et al. [2015].

A heurística de empacotamento busca alocar os itens de uma sequência R , em que cada item $r \in R$ é arranjado na posição que resulta na melhor aptidão para a solução, caso ainda haja



espaço no recipiente para o item ser empacotado. A aptidão de uma posição p em relação ao item r é calculada pela função (2). Após analisar todas as posições candidatas e viáveis para r , escolhe-se aquela que resulta na melhor aptidão. Vale destacar que as posições válidas são obtidas da malha de *reduced raster points* [Scheithauer e Terno, 1996].

$$f(r, p) = (\text{onlyfit}(r, p), \text{waste}(r, p), -fn(r, p), y(p), x(p)), \quad (2)$$

sendo que $\text{onlyfit}(r, p) = 1$ indica que o item r pode ser empacotado na posição p ; $\text{waste}(r, p)$ representa o desperdício gerado ao empacotar o item r em p ; $fn(r, p)$ corresponde a quantidade de lados de r que têm tamanho igual ao do espaço onde o item vai ser inserido; e, por fim, $y(p)$ e $x(p)$ correspondem a coordenada y e x da posição p , respectivamente.

Os parâmetros iniciais do Algoritmo 2 são: P_a , que corresponde a probabilidade de aplicar a operação de Adaptação; N_{iter} , que indica o número de iterações; $N_{colônias}$, que representa a quantidade de colônias de algas; e , que é o coeficiente de perda de energia; tam , que é o número de algas em uma colônia; e, f_c , que é a força de cisalhamento.

Algoritmo 2- Estrutura geral do AAA.

Entrada: $P_a, N_{iter}, N_{colônias}, e, tam, f_c$;
1 Gerar a **População Inicial** z ;
2 $T_t \leftarrow 1, \quad Ap_t \leftarrow 0, \quad t = 1, 2, \dots, N_{colônias}$;
3 $I \leftarrow 0; \quad S \leftarrow Emp(z)$;
4 **repita**
5 $E_t = \left\lceil \frac{T_t}{\max(T)} \right\rceil, \quad \omega_t = 2\pi \left(\sqrt[3]{\frac{3T_t}{4\pi}} \right)^2, \quad t = 1, \dots, N_{colônias}$;
6 $z' \leftarrow z$;
7 **para** $t = 0$ até $N_{colônias}$ **faça**
8 $nf \leftarrow True$;
9 **repita**
10 $z' \leftarrow$ Aplicar o **Movimento Helicoidal** em z'_t ;
11 $E_t \leftarrow E_t - \frac{e}{2}$;
12 **se** $Emp(z'_t) > S_t$ **então** $S_t \leftarrow Emp(z'_t); \quad z_t \leftarrow Z'_t; \quad nf \leftarrow False$;
13 **senão** $E_t \leftarrow E_t - \frac{e}{2}$;
14 $I \leftarrow I + 1$;
15 **até** $E_t \leq 0$;
16 **se** $nf = True$ **então** $Ap_t \leftarrow Ap_t + 1$;
17 Atualizar o tamanho T_t de cada colônia de algas;
18 $z \leftarrow$ Aplicar a **Operação de Reprodução** em z ;
19 $r \leftarrow$ número gerado aleatoriamente entre 0 e 1;
20 **se** $r < P_a$ **então** $z \leftarrow$ Aplicar a **Operação de Adaptação** em z ;
21 **até** $I < N_{iter}$ e número de itens empacotados for inferior a tam ;
Saída: $\max(S)$.

No começo da heurística AAA, obtém-se a solução inicial atribuindo um número aleatório entre 0 e 1 para cada alga de cada colônia, o que fica salvo na matriz de população z . O parâmetro T_t representa as condições de vida da colônia t , sendo que quanto maior o seu valor, melhor são as condições de vida e, assim, a colônia tende a se reproduzir e aumentar. O parâmetro Ap_t é a aptidão da colônia t e está relacionado com a melhora ou piora da solução. Na sequência, aplica-se a função de empacotamento $Emp(Z)$, que usa a estratégia de empacotamento discutida no início dessa seção, para assim retornar a aptidão do empacotamento, que é a área ocupada do recipiente. Em seguida, faz-se o cálculo da energia das colônias (E_t) e da superfície de atrito (ω_t). A energia de uma colônia é proporcional a seu tamanho e tem valor entre 0 e 1. A superfície de atrito representa a área da superfície de um hemisfério, uma vez que as colônias são idealizadas de forma esférica.

O próximo passo da heurística AAA consiste em aplicar o movimento helicoidal sobre cada colônia t enquanto sua respectiva energia for maior do que zero. O movimento helicoidal é



aplicado em três algas de cada colônia, as quais são escolhidas de forma aleatória. As suas posições são modificadas conforme as eqs. (3) a (5).

$$z_{ty} = z_{ty} + (z_{qy} - z_{ty})(f_c - \omega_t)\rho, \quad (3)$$

$$z_{ts} = z_{ts} + (z_{qs} - z_{ts})(f_c - \omega_t)\cos(\alpha), \quad (4)$$

$$z_{tp} = z_{tp} + (z_{qp} - z_{tp})(f_c - \omega_t)\sin(\beta), \quad (5)$$

em que y , s e p são números inteiros selecionados aleatoriamente entre $[0, tam]$; q representa uma colônia vizinha selecionada pelo método da roleta; ρ é um número real no intervalo $[-1, 1]$; e, por fim, α e β assumem valores no intervalo de $[0, 2\pi]$.

Na avaliação da nova solução, caso ela seja melhor do que a corrente, atualiza-se a solução anterior e o parâmetro nf recebe *False*. Uma vez que o critério de parada de E_t seja atingido, verifica-se o valor de nf , sendo que para $nf = True$, a alga não recebe mais energia e nutrientes e, assim, Ap_t é incrementado. Após a energia de todas as colônias serem esgotadas, a população z é submetida ao processo de reprodução. Na operação de reprodução, a colônia com menor tamanho (ou menor valor de função objetivo) tenta se mover em direção a colônia de maior tamanho, sendo este processo apresentado pelas eqs. (6) a (8).

$$alga_maior = \max_{t=1,2,\dots,N_{colônias}} \{T_t\}, \quad (6)$$

$$alga_menor = \min_{t=1,2,\dots,N_{colônias}} \{T_t\}, \quad (7)$$

$$alga_menor_l = alga_maior_l, \quad l = \text{número aleatório entre } [0, tam]. \quad (8)$$

Em seguida, aplica-se a operação de adaptação caso o valor de r seja menor do que a probabilidade de adaptação P_a . A adaptação é o processo em que uma colônia tenta se assemelhar a maior colônia da população. No AAA, o movimento de adaptação consiste em escolher a colônia com maior Ap e adaptá-la conforme as eqs. (9) e (10).

$$z_d = \max_{t=1,2,\dots,N_{colônias}} \{Ap_t\}, \quad (9)$$

$$z_{dl} = \begin{cases} z_{dl} + (alga_maior_l - z_{dl})r_2 & l = 1, 2, \dots, tam, \quad \text{se } r_1 < P_a, \\ z_{dl}, & \text{caso contrário.} \end{cases} \quad (10)$$

sendo que d é o índice da colônia que tem o maior valor de aptidão, l é o índice para cada alga, r_1 e r_2 são números reais aleatórios obtidos no intervalo $[0, 1]$.

Pode ocorrer do vetor resultante, após as operações anteriores, ser inviável, ou seja, as algas podem ter um valor fora do intervalo $[0, 1]$. Assim, uma operação de reparação é aplicada simultaneamente com os movimentos visando corrigir essa inviabilidade. O operador de reparação consiste na função *sigmoide* descrita em (11), em que τ é um parâmetro que modifica a tendência da curva e v é o valor de uma alga.

$$g(v) = sig(v) = \frac{1}{e^{-\tau v} + 1}. \quad (11)$$

4. Experimentos Computacionais

A heurística híbrida e seus procedimentos foram codificados em linguagem C++. Os testes computacionais foram realizados em um computador com processador Intel Core i7-4790K de 4 GHz, 32 GB de memória RAM e sistema operacional Linux Ubuntu 14.04 LTS. O algoritmo



foi executado 10 vezes para cada instância, sendo que o tempo limite imposto, por instância, é de 7.200 segundos. A validação da heurística ocorreu sobre 70 instâncias, as quais foram adaptadas do conjunto de instâncias de Barreto et al. [2007], uma vez que não se conhece trabalhos na literatura que resolveu o 2L-LRP, muito menos instâncias para eventuais comparações. Para tanto, utilizou-se o método proposto por Iori et al. [2007] para gerar os itens retangulares de cada cliente, para cada instância.

Os parâmetros iniciais da heurística foram calibrados pelo pacote irace de López-Ibáñez et al. [2011], exceto o valor de P . Considerou-se a penalidade P com um valor bem alto para evitar soluções que violassem a capacidade dos depósitos e/ou deixasse itens sem empacotar, buscando sempre uma solução viável. Assim, os parâmetros adotados para a heurística foram: $I_{iter} = 5357[(M + N + N_{zero})/4]$; $N_{não-melhoria} = 159$; $T_0 = 54$; $T_F = 0,04$; $\alpha = 0,97$; $B = 0,86$; $t_{max} = 7200$; $P_a = 0,43$; $N_{iter} = 10279$; $N_{colônias} = 96$; $e = 0,51$; $f_c = 2,33$; $\tau = 2$; e, $P = 500(\sum_{i \in I} O_i / \sum_{i \in I} W_i)$.

A Tabela 1 traz em cada uma de suas linhas o nome da instância com a quantidade de clientes N e depósitos M ; a quantidade de itens; melhor solução conhecida (*Best Known Solution* - BKS) até o presente na literatura do LRP, visto que a Tabela 1 corresponde ao LRP puro; a solução média alcançada nas 10 execuções do algoritmo; o GAP, em porcentagem, que corresponde a diferença entre a solução média encontrada pela heurística em relação ao BKS; o valor da melhor solução encontrado pela heurística SA+AAA nas 10 rodadas; o tempo computacional gasto em segundos para computar a solução; e, o GAP com relação a melhor solução da heurística. Os resultados em negrito indicam que a solução encontrada é igual a reportada na coluna “BKS”.

Na Tabela 1 é possível observar que a BKS foi encontrada para nove instâncias. O maior GAP, dada a melhor solução, foi de 2,12% para a instância Christ75 (75x10). O tempo computacional variou de 4,93 a 217,95 segundos, em que os maiores tempos ocorreram para as instâncias com maior número de clientes e depósitos. O tempo computacional médio foi de 65,81 segundos, enquanto que o GAP médio das melhores soluções foi de 0,36%. O GAP da solução média para as 14 instâncias foi de 1,02%, um valor relativamente pequeno.

Tabela 1: Resultado para as instâncias de Barreto et al. [2007] envolvendo o LRP puro.

Instâncias (NXM)	Itens	BKS	S _{média}	GAP (%)	S _{melhor}	Tempo (s)	GAP (%)
Christ50 (50x5)	50	565,60	575,99	1,84	565,60	20,27	0,00
Christ75 (75x10)	75	844,40	869,04	2,92	862,34	67,14	2,12
Christ100 (100x10)	100	833,40	852,64	2,31	838,76	130,95	0,64
Das88 (88x8)	88	355,78	359,11	0,94	355,78	80,92	0,00
Das150 (150x10)	150	43.919,90	44.468,66	1,25	44.207,50	217,95	0,65
Gaspelle (21x5)	21	424,90	424,90	0,00	424,90	4,98	0,00
Gaspelle2 (22x5)	22	585,10	585,11	0,00	585,11	4,93	0,00
Gaspelle3 (29x5)	29	512,10	512,41	0,06	512,10	7,89	0,00
Gaspelle4 (32x5)	32	562,20	562,22	0,00	562,22	9,03	0,00
Gaspelle5 (32x5)	32	504,30	504,32	0,00	504,30	8,95	0,00
Gaspelle6 (36x5)	36	460,37	460,37	0,00	460,37	11,15	0,00
Min27 (27x5)	27	3.062,00	3.062,02	0,00	3.062,02	6,00	0,00
Min134 (134x5)	134	5.709,00	5.853,21	2,53	5.731,98	185,01	0,40
Or117 (117x14)	117	12.290,30	12.587,64	2,42	12.437,70	166,12	1,20
Média				1,02		65,81	0,36

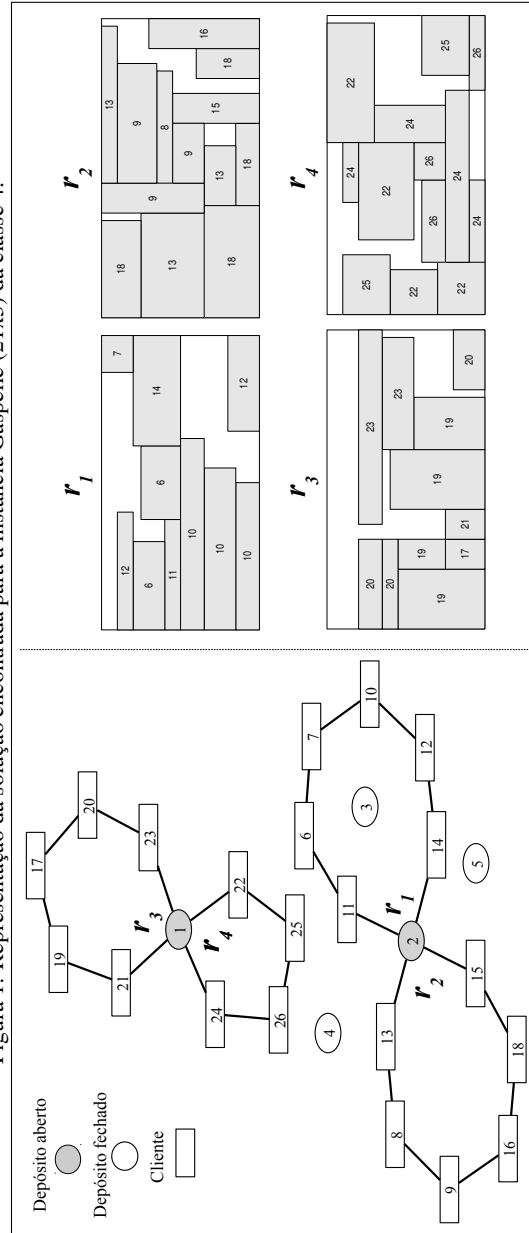
A Tabela 2 apresenta os resultados para as instâncias das classes de 2 a 5. A tabela traz o nome de cada instância, a quantidade de itens, a solução média obtida ao longo das 10 execuções pelo SA+AAA, a melhor solução obtida entre as 10 execuções da heurística (S_{Melhor}) e o seu respectivo tempo computacional em segundos.



Tabela 2: Resultado para as instâncias adaptadas de Barreto et al. [2007] para o 2L-LRP irrestrito.

Instâncias (NxM)	Classe 2				Classe 3				Classe 4				Classe 5			
	Itens	S _{média}	S _{melhor}	Tempo(s)	Itens	S _{média}	S _{melhor}	Tempo(s)	Itens	S _{média}	S _{melhor}	Tempo(s)	Itens	S _{média}	S _{melhor}	Tempo(s)
Christ50 (50x5)	77	702,24	698,47	109,67	97	724,40	723,36	95,46	125	701,01	696,85	250,46	165	696,63	694,39	128,07
Christ75 (75x10)	115	1.128,68	1.110,00	1.165,28	157	1.064,65	1.047,76	802,72	183	1.026,19	1.016,68	280,16	226	986,99	972,14	511,75
Christ100 (100x10)	148	1.273,60	1.254,22	3.498,46	201	1.250,33	1.214,56	1.250,17	252	1.203,98	1.188,53	527,99	310	1.146,13	1.132,01	1.480,09
Das88 (88x8)	133	650,92	640,88	2.238,48	170	599,52	594,30	3.846,80	216	589,80	582,82	2.392,47	251	501,30	492,17	1.387,85
Das150 (150x10)	228	63.453,20	62.109,40	7.200,00	281	59.142,53	58.845,10	7.200,00	347	60.397,94	59.186,90	5.424,79	433	57.957,07	56.438,30	5.518,15
Gaspelle (21x5)	32	453,72	453,72	28,76	41	455,22	453,06	18,59	47	424,90	424,90	7,17	62	436,12	436,12	10,85
Gaspelle2 (22x5)	33	767,24	767,24	18,45	43	737,13	737,04	18,54	58	748,45	748,45	20,84	74	712,96	585,79	47,73
Gaspelle3 (29x5)	44	602,79	602,79	50,55	65	631,79	631,79	61,53	61	578,73	578,73	15,56	87	585,79	585,79	37,78
Gaspelle4 (32x5)	49	675,26	672,80	241,17	69	715,65	714,44	101,43	64	610,51	610,51	32,70	95	629,67	629,67	30,80
Gaspelle5 (32x5)	49	664,79	664,40	149,99	69	716,27	713,54	406,01	64	584,67	584,63	36,48	95	629,67	629,67	33,61
Gaspelle6 (36x5)	55	642,22	642,12	262,91	74	566,10	563,42	54,05	97	560,00	558,59	56,50	116	541,81	540,92	62,00
Min27 (27x5)	40	3.740,82	3.740,82	51,81	61	3.731,81	3.730,05	13,86	65	3.490,76	3.490,40	27,85	98	3.570,31	3.563,10	34,65
Min134 (134x5)	204	8.923,84	8.816,35	3.043,53	254	8.439,09	8.365,60	4.713,59	341	8.759,80	8.605,98	3.683,40	411	7.955,40	7.831,73	1.141,63
Or117 (117x14)	173	20.282,47	19.780,80	7.200,00	227	19.555,98	19.363,10	4.463,83	304	20.056,96	19.607,80	3.328,85	353	19.376,77	19.222,60	3.550,91
Média				1.804,22				1.646,18				1.148,95				998,28

Figura 1: Representação da solução encontrada para a instância Gaspelle (21x5) da classe 4.





Nas instâncias da classe 2, cada cliente contém 1 ou 2 itens. Analisando os resultados, observa-se que para duas instâncias a heurística atingiu o tempo máximo, sendo que o tempo computacional médio foi de 1.804,22 segundos. As soluções mais próximas do resultado encontrado para as instâncias do LRP foram para Gaspelle (21x5), Gaspelle3 (29x5) e a Gaspelle4 (32x5), enquanto as piores soluções ocorreram para as instâncias com maior número de clientes e de itens.

Para a classe 3, os clientes têm uma demanda por itens variando de 1 a 3. Nesta classe, a solução obtida pela heurística ficou mais próxima daquela encontrada para o LRP puro, além de requerer um tempo computacional inferior (na média foi 1.646,18 segundos) ao comparar com o tempo para a classe 2. Os clientes da classe 4 contêm entre 1 a 4 itens, sendo que a heurística conseguiu obter uma solução igual a do LRP puro para a instância Gaspelle(21x5). Essa instância tem um total de 47 itens, a solução tem valor igual a 424,90 e foi preciso 7,15 segundos para a heurística achar tal solução. Além disso, em todas as 10 execuções da heurística, obteve-se a mesma solução para essa instância.

A classe 5 contém as instâncias com o maior número de itens por cliente, com valor variando entre 1 a 5 itens por clientes. As soluções obtidas nessa classe ficaram próximas da solução do LRP puro para as instâncias pequenas apenas. Destaca-se que o tempo computacional médio para resolver as instâncias da classe 5 foi de 998,28 segundos. Nota-se ainda que a tendência é a solução piorar a medida que o número da classe aumenta, pois há mais itens por cliente.

A Figura 1 traz uma ilustração da solução encontrada para a instância Gaspelle (21x5) da classe 4. A figura apresenta a solução do LRP, bem como o empacotamento dos itens nos veículos para cada rota. Percebe-se que foi necessário abrir dois depósitos e criar quatro rotas para atender a demanda dos clientes, sendo que a rota r_2 apresentou a maior ocupação da base retangular do veículo.

5. Considerações Finais

Este trabalho traz um heurística híbrida para resolver o problema de localização e roteamento com restrições de empacotamento bidimensional. Este problema ainda tem sido muito pouco explorado na literatura e a sua importância está relacionada com situações de desastres naturais, em que é preciso determinar quais depósitos devem ser abertos e definir as rotas para atender os clientes conforme a demanda por itens retangulares.

A heurística desenvolvida consiste na cooperação de duas meta-heurísticas, o recozimento simulado, aplicado sobre o problema de localização e roteamento, e o algoritmo artificial de algas, utilizado para resolver o problema de empacotamento ortogonal de itens retangulares. Os testes computacionais realizados em 70 instâncias adaptadas da literatura do LRP mostram que a heurística é competitiva. Observa-se que para as instâncias da classe 1, que corresponde ao LRP puro, a heurística obteve 9 (de 14 instâncias) com soluções iguais as apresentadas na literatura recente do LRP. No geral, o GAP médio das melhores soluções foi de 0,36%, que é um valor pequeno.

Para as instâncias das demais classes, a heurística retornou soluções satisfatórias, embora não seja conhecido limitantes ou outras soluções para efeitos de comparação. Em termos de tempo computacional, acredita-se que o tempo reportado é aceitável para as aplicações práticas, uma vez que o problema envolve os três níveis de decisão do sistema logístico.

Trabalhos futuros desta pesquisa visam estudar o 2L-LRP sequencial, além de desenvolver uma rotina de busca local para agilizar e melhorar a convergência do AAA. Outra frente de pesquisa deve ocorrer na busca por limitantes e modelos de programação linear inteira para o 2L-LRP.

Agradecimentos. Os autores agradecem o apoio financeiro recebido do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - processo 308312/2016-3), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e da Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG).



Referências

- Barreto, S., Ferreira, C., Paixão, J., e Santos, B. S. (2007). Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179:968–977.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: Freeman.
- Gendreau, M., Iori, M., Laporte, G., e Martello, S. (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18.
- Iori, M. e Martello, S. (2013). Review an annotated bibliography of combined routing and loading problems. *Journal of Operations Research*, 23(3):311–326.
- Iori, M., Salazar-González, J.-J., e Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2):253–264.
- Kirkpatrick, S., Gelatt, C. D., e Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Leung, S. C., Zheng, J., Zhang, D., e Zhou, X. (2010). Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible services and manufacturing journal*, 22(1-2):61–82.
- Lin, S. e Kernighan, B. W. (1973). An effective algorithm for the travelling-salesman problem. *Operations Research*, 21(2):498–516.
- López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., e Birattari, M. (2011). *The irace package, Iterated Race for Automatic Algorithm Configuration*. Technical report, Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium.
- Pramudita, A., Taniguchi, E., e Qureshi, A. G. (2014). Location and routing problems of debris collection operation after disasters with realistic case study. *Procedia-Social and Behavioral Sciences*, 125:445–458.
- Queiroz, T. A., Oliveira, J. F., Carravilla, M. A., e Miyazawa, F. K. (2016). Demand uncertainty for the location-routing problem with two-dimensional loading constraints. In *Computational Management Science*, p. 47–53. Springer.
- Rath, S. e Gutjahr, W. J. (2014). A math-heuristic for the warehouse location–routing problem in disaster relief. *Computers & Operations Research*, 42:25–39.
- Scheithauer, G. e Terno, J. (1996). The g4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, 47(4):511–522.
- Song, R., He, S., e Zhang, L. (2009). Optimum transit operations during the emergency evacuations. *Journal of Transportation Systems Engineering and Information Technology*, 9(6):154–160.
- Uymaz, S. A., Tezel, G., e Yel, E. (2015). Artificial algae algorithm (aaa) for nonlinear global optimization. *Applied Soft Computing*, 31:153–171.
- Wei, L., Zhang, Z., Zhang, D., e Lim, A. (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 243(3):798–814.
- Zhang, X., Wu, C., Li, J., Wang, X., Yang, Z., Lee, J.-M., e Jung, K.-H. (2016). Binary artificial algae algorithm for multidimensional knapsack problems. *Applied Soft Computing*, 43:583–595.