

## RELIGAÇÃO DE CAMINHO ADAPTATIVA E BUSCA TABU PARA O PROBLEMA DE LOCALIZAÇÃO DE FACILIDADES COM RESTRIÇÕES DE CAPACIDADE E FONTE ÚNICA

**Ramon Faganello Fachini**

Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação  
Av. Albert Einstein, 400, CEP 13083-852, Campinas, SP, Brasil  
ramonfachini@gmail.com

**Vinicius Amaral Armentano**

Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação  
Av. Albert Einstein, 400, CEP 13083-852, Campinas, SP, Brasil  
vinicius@densis.fee.unicamp.br

### RESUMO

Problemas de localização de facilidades são encontrados em diferentes áreas, tais como logística e telecomunicações. Um problema complexo é o Problema de Localização de Facilidades com Restrições de Capacidade e Fonte Única (PLFRCU), em que cada cliente é atendido por uma única facilidade. Este trabalho utiliza a Otimização por Memória Adaptativa (OMA), um dos *frameworks* mais bem-sucedidos para a solução de problemas de otimização combinatória de elevada complexidade, conforme demonstram recentes aplicações em problemas diversos, tais como o roteamento de veículos. Em particular, um novo método heurístico com memória adaptativa é proposto para resolver o PLFRCU, que consiste na combinação de Religação de Caminho Adaptativa (RCA) com o método da busca tabu. Testes computacionais com instâncias de *benchmark* comprovam que o método proposto é capaz de chegar a soluções de alta qualidade em tempos computacionais razoáveis, portanto, uma abordagem promissora para a solução do PLFRCU.

**PALAVRAS CHAVE.** Religação de Caminho Adaptativa, Busca Tabu, Problema de Localização de Facilidades com Restrições de Capacidade e Fonte Única.

**Tópicos:** MH – Metaheurísticas; OC - Otimização Combinatória.

### ABSTRACT

Facility location problems are encountered in many areas, such as logistics and telecommunications. A complex problem is the Single Source Capacitated Facility Location Problem (SSCFLP), in which each customer is served by a single facility. This article makes use of the Adaptive Memory Programming (AMP), one of the most successful frameworks for solving hard combinatorial optimisation problems, as shown by recent applications in several problems, such as the vehicle routing problem. We propose a new AMP based heuristic method for the SSCFLP, which combines the Adaptive Path Relinking (APR) technique with the tabu search method. Computational tests with benchmark instances show that the proposed method is able to find high-quality solutions in reasonable computational times, therefore, a promising approach to solve the SSCFLP.

**KEYWORDS.** Adaptive Path Relinking, Tabu Search, Single Source Capacitated Facility Location Problem.

**Topics:** MH – Metaheuristics; OC – Combinatory Optimization.

## 1. Introdução

Problemas de localização de facilidades determinam os melhores locais para abrir facilidades de modo a atender a demanda de clientes, sujeito a restrições adicionais. Localização de facilidades é parte importante da ciência da localização, que reflete a interação com outras disciplinas, tais como logística, telecomunicações, sistemas de informação geográfica e sistemas de cuidado da saúde. A pesquisa conduzida nesta área levou à criação de propriedades teóricas, modelos matemáticos e métodos de resolução. A história da ciência da localização é bem desenvolvida por [Hale e Moberg 2003] e [Daskin 2013] apresenta problemas clássicos de localização de facilidades por meio de teoria, modelos e técnicas de resolução. [Eiselt e Marianov 2011] editam um livro com diversas contribuições em ciência da localização, envolvendo modelos e métodos de otimização para diversas classes de problemas, métodos exatos e heurísticos para alguns problemas, bem como modelos desenvolvidos por geógrafos para localização de facilidades baseada em padrões de atratividade para clientes. [Eiselt e Marianov 2015] apresentam diversas aplicações divididas em três classes: negócios, serviços públicos e aquelas que tratam da aplicação da lei e resposta de serviço, como, por exemplo, localização de estações de bombeiros com 5 minutos de resposta para 80% das chamadas. [Laporte et al. 2015] editam outro livro com três temas: conceitos básicos, conceitos avançados e aplicações. Obviamente, há revisões específicas publicadas em revistas, tais como localização de facilidades e gerenciamento de cadeias de suprimento [Melo et al. 2009] e localização de redes de *hubs* [Alumur e Kara 2008].

Este trabalho aborda um problema de localização de facilidades conhecido como Problema de Localização de Facilidades com Restrições de Capacidade e Fonte Única (PLFRCU), *Single Source Capacitated Facility Location* na língua inglesa, em que cada cliente é atendido por uma única facilidade. [Cornuéjols et al. 1990] e [Alfandari 2007] demonstram que este problema é NP difícil.

Seja um conjunto  $J$  de clientes e um conjunto potencial de locais candidatos  $I$ , tal que cada cliente  $j \in J$  tem uma demanda  $d_j$  e cada facilidade aberta em um local  $i \in I$  tem capacidade  $b_i$ . A abertura de uma facilidade em um local  $i$  incorre em um custo fixo  $f_i$  e cada cliente  $j$  é atendido por uma única facilidade (fonte única) com custo  $c_{ij}$ .

Considere as variáveis binárias:

- $y_i = 1$  se uma facilidade é aberta no local,  $y_i = 0$ , caso contrário,  $i \in I$ ;
- $x_{ij} = 1$  se o cliente  $j$  é atendido pela facilidade aberta em  $i$ ,  $x_{ij} = 0$ , caso contrário,  $i \in I, j \in J$ .

O modelo matemático é apresentado a seguir:

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

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J \quad (2)$$

$$\sum_{j \in J} d_j x_{ij} \leq b_i y_i, \quad i \in I \quad (3)$$

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

A função objetivo (1) expressa a minimização do custo total de abertura de facilidades e designação de cliente a facilidades. As restrições (2) indicam que cada cliente  $j$  é atendido por uma única facilidade  $i$ . As restrições (3) impõem que a soma total das demandas de clientes atendidas por uma facilidade aberta em  $i$  é limitada pela capacidade  $b_i$ . As restrições (4) indicam o domínio das variáveis.

O modelo matemático mostra a complexidade deste problema, pois, ao se fixar as variáveis  $y_i$  em seus valores binários, o problema resultante é um Problema de Designação Generalizada (PDG), *Generalized Assignment* na língua inglesa, que também é NP difícil.

A restrição de fonte única em localização aparece em diversas situações em que um cliente requer sua designação a uma única facilidade. Na área de telecomunicações o problema de localização de concentradores no projeto de uma rede requer que cada usuário seja atendido por um único concentrador [Pirkul 1987], [Filho e Galvão 1998] e [Holmberg et al. 1999]. Outro exemplo envolve a escolha de locais para instalar plataformas para extrair petróleo de poços, que requer que cada poço pertença a uma única plataforma [Devine e Lesso 1972] e [Balas 1982].

Diversos métodos exatos e heurísticos foram propostos na literatura para a resolução do PLFRCU. [Neebe e Rao 1983] introduzem um dos primeiros métodos exatos para o problema combinando geração de colunas e *branch-and-bound*. [Holmberg et al. 1999] propõem um algoritmo de *branch-and-bound* baseado em relaxação Lagrangeana. [Díaz e Fernández 2002] desenvolvem um algoritmo de *branch-and-price* e [Yang et al. 2012] sugerem um algoritmo de *cut-and-solve*. Neste último artigo, são obtidas soluções ótimas para as maiores 20 instâncias com 30 a 80 facilidades e 200 a 400 clientes.

Entre os métodos heurísticos, predominam algoritmos que combinam relaxação Lagrangeana para a obtenção de limitantes inferiores e heurísticas construtivas ou meta-heurísticas para a obtenção de limitantes superiores [Hindi e Pienkosz 1999], [Cortinhal e Captivo 2003] e [Chen e Ting 2008]. Uma abordagem diferente é apresentada pela heurística de pareamento repetido [Ronnqvist et al. 1999], fundamentada na resolução de sucessivos problemas de pareamento até que um critério de convergência seja satisfeito. No tocante a meta-heurísticas, [Filho e Galvão 1998] propõem uma busca tabu para o PLFRCU e [Delmaire et al. 1999] apresentam algoritmos de GRASP reativo, busca tabu e variantes híbridas de tais métodos. Outras meta-heurísticas desenvolvidas foram a busca em vizinhança muito grande (*very large-scale neighborhood search*) de [Ahuja et al. 2004], *scatter search* de [Contreras e Díaz 2008] e a busca tabu iterada de [Ho 2015], recentemente proposta. Por fim, destacam-se [Guastaroba e Speranza 2014] que obtiveram excelentes resultados ao solucionar o PLFRCU por meio de uma heurística denominada *kernel search*, baseada na fixação do valor de variáveis consideradas promissoras.

Este trabalho propõe uma nova meta-heurística derivada da estrutura do método Otimização por Memória Adaptativa (OMP) [Taillard et al. 2001], *Adaptive Memory Programming* na língua inglesa, que analisa meta-heurísticas baseadas em memória, busca tabu, *scatter search*, algoritmo genético e colônias de formigas. A partir desta análise, é proposto um arcabouço (*framework*) meta-heurístico com as seguintes características: (1) um conjunto de soluções ou uma estrutura de dados que agrega as características das soluções produzidas pela busca é memorizada; (2) uma solução provisória é construída usando os dados na memória; (3) a solução provisória é melhorada por uma busca local ou uma meta-heurística; (4) a nova solução é incluída na memória ou é usada para atualizar a estrutura de dados que memoriza a história da busca. A OMP foi aplicada com sucesso a diversos problemas, tais como, roteamento de veículos [Olivera e Viera 2007], [Li et al. 2010] e [Tarantilis et al. 2013], projeto de cadeia de suprimento [Cardona-Valdés et al. 2014], programação de projetos [Paraskevopoulos et al. 2016] e otimização global [Lasdon et al. 2010].

A nova meta-heurística para o PLFRCU é baseada no procedimento de Religação de Caminho Adaptativa (RCA), *Adaptive Path Relinking* na língua inglesa, introduzida por [Tarantilis et al. 2013]. A RCA parte de um conjunto de soluções de referência (*Ref\_Set*) e contém um procedimento adaptativo de recombinação de múltiplas soluções para a criação de soluções guia temporárias (*S<sub>guide</sub>*). Essa construção de soluções provisórias é realizada por operadores de remoção e reinserção. Originalmente, tais operadores constituem a base da busca em vizinhança grande (*large neighborhood search*), apresentada por [Ropke e Pisinger 2006], em que os operadores são selecionados probabilisticamente e aplicados por meio de um procedimento adaptativo. De modo análogo, na RCA, a avaliação de atributos recorrentes nas soluções de referência controla a seleção probabilística dos operadores de remoção e reinserção para a construção de soluções guia temporárias. A partir de tais soluções, são geradas trajetórias

para as soluções do conjunto de referência e se realiza uma busca baseada em um método de religação de caminho. A escolha foi motivada pelos excelentes resultados que o método se mostrou capaz de gerar em [Tarantilis et al. 2013], além de consistir em um método que necessita de poucos parâmetros de ajuste.

Devido à alta complexidade do PLFRCU, é necessário partir de um conjunto inicial de soluções de referência de alta qualidade que são obtidas pela busca tabu, em lugar de uma busca local como originalmente proposto por [Tarantilis et al. 2013]. Os novos conjuntos de referência são obtidos a partir da RCA e o método é terminado quando um conjunto de referência não é atualizado por um número de vezes.

O restante desse artigo se encontra organizado da seguinte maneira. A seção 2 apresenta uma descrição da nova meta-heurística, enquanto a seção 3 apresenta os experimentos computacionais. A seção 4 contém conclusões e perspectivas futuras do trabalho.

## 2. Descrição de uma nova meta-heurística para o PLFRCU

Conforme mencionado na seção 1, os pontos-chave do método proposto consistem na aplicação da busca tabu para a construção do primeiro conjunto de referência de soluções para, então, aplicar o procedimento RCA ao PLFRCU. Estes são os principais componentes da nova meta-heurística que contém subcomponentes detalhados a seguir.

Para a aplicação da busca tabu é necessário que se parta de uma solução inicial factível que é obtida da seguinte forma. Obtenha uma solução por meio de uma heurística construtiva que gera as facilidades abertas e a designação de clientes. A solução inicial é a melhor solução obtida por uma busca local sobre o PDG resultante das facilidades abertas pela heurística construtiva. Durante a busca tabu, determina-se e armazena-se o primeiro conjunto de soluções de referência para iniciar a RCA proposta por [Tarantilis et al. 2013]. Esta RCA é composta de quatro procedimentos: definição de operadores de remoção e reinserção, um procedimento de religação de caminho, uma busca local para melhorar as soluções encontradas durante a religação de caminho (nessa fase aplicamos novamente uma busca local sobre o PDG) e um método de atualização do conjunto de referência [Tarantilis et al. 2013]. Este processo de RCA é repetido até atingir um limite de repetições sem que se atualize o conjunto de referência. A seguir, cada uma das etapas da meta-heurística proposta é explicada em detalhes.

### 2.1 Heurística construtiva

Existem diversas proposições de heurísticas construtivas para o PLFRCU na literatura. Neste trabalho essas heurísticas foram avaliadas e a opção que mais se adequava aos objetivos da busca tabu projetada foi escolhida.

Dentre as heurísticas construtivas encontradas na literatura do PLFRCU destacam-se as propostas por [Cortinhal e Captivo 2003], [Delmaire et al. 1999], [Hindi e Pienkosz 1999], [Ho 2015], [Holmberg et al. 1999], [Klincewicz e Luss 1986] e [Pirkul 1987].

Esses métodos foram analisados, buscando as duas alternativas que melhor explorassem a estrutura do PLFRCU durante o processo iterativo de construção das soluções. Dessa análise, foram selecionados os métodos de [Hindi e Pienkosz 1999] e [Klincewicz e Luss 1986] como os que apresentavam maior sofisticação em seu procedimento construtivo. As duas heurísticas foram implementadas e testadas para um conjunto de instâncias da literatura e obtiveram soluções factíveis para todas as instâncias testadas. Porém, pela análise dos testes, ficou evidente que a heurística de [Hindi e Pienkosz 1999] mostrou-se mais eficaz na obtenção de resultados de qualidade que a heurística de [Klincewicz e Luss 1986].

Desse modo, adotamos a heurística construtiva de [Hindi e Pienkosz 1999] em nosso trabalho. Nessa heurística, a cada estágio, um cliente é escolhido e designado a uma facilidade. Para isso, os autores definem  $\pi_{ij}$  como a penalidade decorrente da designação do cliente  $j$  à facilidade  $i$ . Essa penalidade é calculada da seguinte forma:

- Se a facilidade  $i$  já foi aberta,  $\pi_{ij} = c_{ij}$ ;

- Caso contrário,  $\pi_{ij} = c_{ij} + (f_i \cdot d_j) / b_i$ .

Uma facilidade é considerada aberta assim que um cliente é designado à mesma. Antes de se apresentarem os passos do algoritmo é importante, ainda, definir o arrependimento (*regret*) de um cliente não designado. Esse arrependimento, para um cliente  $j$  não designado, é calculado pela diferença entre o segundo menor valor e o menor valor de  $\pi_{ij}$ , tomadas todas as facilidades  $i$  cujas capacidades são maiores que a demanda do cliente  $j$ . Se um determinado cliente só pode ser designado a uma única facilidade, então lhe é atribuído um valor muito alto de arrependimento.

O algoritmo funciona pela repetição dos passos abaixo enquanto não se designarem todos os clientes:

- (1)Dentre os clientes ainda não designados, escolha o cliente com maior valor de arrependimento;
- (2)Designe o cliente escolhido à facilidade com menor valor de  $\pi_{ij}$ , dentre as facilidades com capacidade suficiente. Se nenhuma facilidade tiver capacidade suficiente, designe o cliente escolhido à facilidade com maior capacidade corrente;
- (3)Atualize os valores das penalidades e dos arrependimentos para todos os clientes não designados.

## 2.2 Busca local para o PDG

A busca local para solução do PDG parte de uma solução completa do PLFRCU, na qual já são conhecidos os valores das variáveis binárias  $y_i$  e  $x_{ij}$ . A partir dessa solução, foi necessária a definição de movimentos que pudessem induzir vizinhanças, viabilizando a execução da busca local. A partir da literatura do PDG, definiu-se a utilização de movimentos de troca e inserção. O primeiro tipo de movimento consiste na troca entre dois clientes designados a facilidades distintas, sendo um movimento de influência menor por manter a cardinalidade dos clientes designados às facilidades. O segundo tipo é o movimento de inserção, que retira um cliente designado a uma facilidade e o aloca a outra facilidade, alterando a cardinalidade dos clientes designados às facilidades.

Ainda foram necessárias outras definições do método de busca local. A primeira seria a decisão entre adotar busca local do tipo *First Improvement* ou do tipo *Best Improvement*. A segunda seria como utilizar os movimentos de troca e inserção: executar primeiro uma busca local com troca e, depois, uma busca local com movimentos de inserção; executar uma busca local com inserção e, posteriormente, uma busca local com troca; ou a cada iteração da busca avaliar o melhor dentre todos os movimentos de troca e inserção. Novamente, com base em experimentos computacionais, decidiu-se por adotar uma busca local do tipo *Best Improvement* utilizando a última das configurações de movimentos descritas, ou seja, a cada iteração escolheu-se o melhor dentre todos os movimentos de troca e inserção.

É importante destacar que essa busca local para o PDG a partir de uma configuração fixa de facilidades abertas é crítica para o bom desempenho do método proposto, já que a mesma é utilizada em vários estágios do mesmo: antes da busca tabu, dentro da busca tabu e para a melhoria das soluções factíveis encontradas pelo método de religação de caminho.

## 2.3 Busca tabu

Busca tabu [Glover e Laguna 1997] é uma meta-heurística que guia um procedimento de busca local pela utilização da solução corrente e da história da busca para explorar o espaço de soluções além da otimalidade local. Isto é feito por uma memória de curto prazo composta por atributos de soluções que são utilizados para proibir a visita a um conjunto de soluções. A memória de longo prazo armazena a frequência de atributos de soluções que indiquem quais soluções e/ou regiões foram mais ou menos visitadas. Esta informação pode ser usada para guiar a busca em processos de intensificação ou diversificação. No método aqui proposto, utilizamos somente a memória de curto prazo da busca tabu para encontrar soluções diversas e de alta

qualidade para o primeiro conjunto de soluções de referência ao qual se aplica a RCA. A memória de curto prazo é descrita a seguir.

### ***Movimentos e vizinhanças***

A busca tabu para o PLFRCU utiliza movimentos de abertura e fechamento de facilidades, e, a cada iteração, escolhe-se o melhor movimento entre todos os movimentos de abertura e de fechamento de facilidades.

Quando uma facilidade se encontra fechada para uma dada solução, avalia-se então o movimento de abertura dessa facilidade. Ao se abrir tal facilidade, a busca local do PDG é executada para obter-se um ótimo local para essa nova configuração de facilidades.

Já ao se fechar uma facilidade, é utilizada a heurística construtiva proposta por [Martello e Toth 1981] para o PDG a fim de avaliar possíveis soluções factíveis para a nova configuração de facilidades obtida. Essa heurística utiliza uma função de avaliação, que para o contexto do PLFRCU é dada por  $f(i, j) = d_j / b_i$ . Caso a heurística obtenha uma solução factível, a busca local do PDG é chamada para otimizá-la, caso contrário, abandona-se tal configuração de facilidades.

### ***Regras de ativação tabu***

A fim de se evitar uma ciclagem resultante dos movimentos de abertura e de fechamento de facilidades, definiu-se uma proibição bastante restritiva, isto é, quando o melhor movimento altera o estado da facilidade  $i$ , ficam proibidas novas alterações de estado dessa facilidade durante a duração tabu.

### ***Lista tabu***

A lista tabu foi implementada na forma de um vetor de dimensão  $|I|$  (cardinalidade do conjunto de potenciais facilidades) cujas posições representam as facilidades. Em cada posição  $i$ , o vetor armazena o número da iteração até a qual são proibidas alterações de estado daquela facilidade.

### ***Duração tabu***

A duração tabu é escolhida segundo uma distribuição uniforme em um intervalo  $[a, b]$  pré-definido, sendo  $a$  e  $b$  parâmetros do problema. Dessa forma, se em uma determinada iteração  $it$ , a alteração de estado da facilidade  $i$  passa a ser tabu, então a posição  $i$  do vetor que representa a lista tabu recebe o valor da iteração  $it$  acrescido de um valor obtido segundo distribuição uniforme no intervalo  $[a, b]$ .

### ***Critério de aspiração***

Movimentos considerados tabu são permitidos apenas quando o critério clássico de aspiração é satisfeito, isto é, se a solução resultante é melhor que a solução incumbente.

### ***Critério de parada***

O critério de parada adotado foi um número máximo de iterações, um parâmetro denotado por  $TS$ .

## 2.4 Método de atualização do conjunto de referência

Durante a busca tabu armazena-se um conjunto de  $\alpha$  soluções que compõem o primeiro conjunto de soluções de referência, em que  $\alpha$  é outro parâmetro do método.

Para que o procedimento de RCA seja eficiente quando aplicado às soluções do conjunto de referência, é importante que o mesmo armazene soluções que contenham tanto qualidade quanto diversidade. Para medir a diversidade, introduz-se uma métrica de distância entre as soluções. Define-se como distância entre duas soluções o somatório do número de facilidades que se encontram em estados diferentes nas mesmas.

Sendo  $S_{best}$  e  $S_{worst}$  as soluções de melhor e pior qualidade do conjunto de referência, respectivamente, o método de atualização do conjunto de referência permite que uma dada solução  $S$  entre no conjunto de referência se:

- Tem custo menor que  $S_{best}$ ;
- Tem custo menor  $S_{worst}$  e a distância de  $S$  às demais soluções do conjunto é maior que a distância de  $S_{worst}$  às mesmas.

A entrada de uma nova solução do conjunto de referência implica na exclusão de  $S_{worst}$ .

## 2.5 Operadores de remoção e reinserção

A partir do primeiro conjunto de referência gerado pela busca tabu são iniciadas as iterações da RCA. Para cada solução do conjunto de referência, utiliza-se um procedimento para a geração de uma solução guia temporária para a realização do religamento de caminho entre pares de soluções.

Esse procedimento ocorre pela remoção de clientes “mal designados” e pela reinserção desses clientes em facilidades “promissoras”. Tanto a remoção de clientes quanto a reinserção dos mesmos são probabilísticas, como descrito a seguir.

### Remoção de clientes

A remoção de clientes é controlada por um limiar de aceitação adaptativo  $T$ , utilizado para identificar designações promissoras. Dadas duas soluções de referência  $S_a$  e  $S_b$ , define-se uma variável binária  $I_{ij}^{S_a, S_b}$  que assume o valor 1 se a designação de um cliente  $j$  a uma facilidade aberta  $i$  está presente em ambas as soluções e o valor 0, caso contrário. O limiar  $T$  é calculado da seguinte forma:

$$T = (\sum_{S \in Ref\_Set - \{S_{best}\}} \sum_{ij} I_{ij}^{S, S_{best}}) / [(\alpha - 1) \cdot |J|] \quad (5)$$

A expressão (5) mostra que, quanto maior a similaridade média entre as soluções do conjunto de referência, maior o valor de  $T$ . Além disso, o valor do limiar  $T$  é automaticamente ajustado quando o conjunto de referência é atualizado.

Além do limiar  $T$ , o método de remoção também pondera a frequência de aparição  $o(i, j)$  de uma determinada designação  $(i, j)$ . Desse modo, para uma dada solução do conjunto de referência, cada designação  $(i, j)$  é avaliada segundo a regra:

- Se  $o(i, j) \geq T$ , o cliente  $j$  é removido da facilidade  $i$  com probabilidade  $(1 - p_c)$ ;
- Se  $o(i, j) < T$ , o cliente  $j$  é removido da facilidade  $i$  com probabilidade  $p_c$ .

Nesse caso,  $p_c$  é um parâmetro utilizado tanto pelo procedimento de remoção, quanto pelo procedimento de reinserção de clientes.

### Reinserção de clientes

Ao final da etapa de remoção, os clientes removidos têm de ser reinseridos na solução parcialmente destruída. Porém, ao invés de simplesmente realizar as realocações de modo “miope” visando unicamente à minimização de custos das designações, a etapa de reinserção também pondera a manutenção da diversidade das soluções exploradas, realizando algumas das

inserções em facilidades aleatórias. Assim, a reinserção de cada cliente removido é realizada da seguinte maneira:

- O cliente é reinserido na facilidade aberta que proporcionar o menor custo de designação com probabilidade  $(1 - p_c)$ ;
- O cliente é reinserido aleatoriamente segundo uma distribuição uniforme em uma das facilidades abertas com uma probabilidade  $p_c$ . Dois operadores de reinserção alternativos nos quais o cliente  $j$  removido era reinserido com probabilidade  $p_c$  na facilidade  $i$  que proporcionasse a maior e a menor frequência de aparição  $o(i, j)$ , respectivamente, também foram testados, porém a versão aleatória apresentou melhores resultados.

Vale observar que cada solução guia encontrada pode entrar para o conjunto de soluções de referência somente se atender as condições estabelecidas pelo método de atualização do conjunto de referência descrito na seção 2.4.

## 2.6 Procedimento de religação de caminho

A religação de caminho foi originalmente proposta por [Glover 1996] para integrar estratégias de intensificação e de diversificação em busca tabu. No contexto do algoritmo proposto nesse artigo, cada solução guia temporária é utilizada na religação de caminho com a solução de referência que a gerou.

Como as soluções ligadas têm a mesma configuração de facilidades abertas, a religação de caminho se baseia na execução de sucessivos movimentos de inserção que aproximam a solução guia da solução de referência que a gerou. É executado um movimento de inserção por iteração da religação de caminho e, a cada solução factível encontrada durante esse procedimento, aplica-se a busca local para o PDG.

Todas as soluções visitadas durante essa fase do método de solução proposto podem entrar para o conjunto de soluções de referência caso atendam as condições estabelecidas pelo método de atualização do conjunto de referência. A RCA é finalizada se nenhuma solução é adicionada ao conjunto de referência após a execução de  $k$  ciclos de aplicação de remoção e reinserção seguidos de religação de caminho, tal que  $k$  é o último parâmetro do método de solução.

O pseudocódigo do método híbrido RCA e busca tabu com todos os seus componentes é apresentado no Algoritmo 1.

---

### Algoritmo 1: Método híbrido RCA e busca tabu

---

**Entrada:**  $TS, a, b, \alpha, p_c, k$   
**Saída:** melhor solução factível encontrada  $S_{best}$

- 1  $Ref\_Set \leftarrow \emptyset$
- 2 Aplique a **heurística construtiva para o PLFRCU** para obter a solução factível inicial  $S_0$
- 3  $S_0 \leftarrow$  **busca local PDG** ( $S_0$ )
- 4  $Ref\_Set \leftarrow$  **busca tabu** ( $S_0, TS, a, b, Ref\_Set, \alpha$ )
- 5 **Repita**
- 6     **Para cada** solução  $S \in Ref\_Set$  **faça**
- 7          $S_{guide} \leftarrow$  **remoção e reinserção** ( $S, Ref\_Set, p_c$ )
- 8         **religação de caminho** ( $S_{guide}, S$ )
- 9         **Para cada** solução factível intermediária  $S_{inter}$  **faça**
- 10              $S_{inter} \leftarrow$  **busca local PDG** ( $S_{inter}$ )
- 11              $Ref\_Set \leftarrow$  **atualiza**  $Ref\_Set$  ( $S_{inter}, Ref\_Set, \alpha$ )
- 12         **Fim Para**
- 13     **Fim Para**
- 14 **Até** atingir  $k$  repetições sem que se atualize o  $Ref\_Set$

---

### 3. Experimentos computacionais

O método proposto foi implementado em linguagem C# na plataforma Microsoft Visual Studio Community 2013. Testes computacionais foram realizados utilizando um computador equipado com sistema operacional Windows 8.1, CPU Intel® Core™ i7-4790 3,6GHz e 16GB de memória RAM.

Para testar este novo método heurístico, foram realizados experimentos com um conjunto clássico de instâncias de *benchmark* da literatura, proposto por [Holmberg et al. 1999]. Esse conjunto contém 71 instâncias divididas em quatro subconjuntos que diferem entre si pelo tamanho e por suas propriedades. A Tabela 1 contém um breve descritivo desses subconjuntos, indicando o número de facilidades  $|I|$  e o número de clientes  $|J|$  considerado em suas instâncias.

**Tabela 1 – Conjunto de instâncias de [Holmberg et al. 1999]**

Subconjunto	Instâncias	$ I $	$ J $
(1)	h1 – h24	10 – 20	50
(2)	h25 – h40	30	150
(3)	h41 – h55	10 – 30	70 – 100
(4)	h56 – h71	30	200

Antes da execução dos testes propriamente ditos foi necessária a calibração do método. Conforme descrito na seção 2, o método híbrido RCA e busca tabu possui seis parâmetros: os parâmetros  $TS$ ,  $a$  e  $b$  referentes à busca tabu, o parâmetro  $\alpha$  que indica o tamanho do conjunto de referência e os parâmetros  $p_c$  e  $k$  referentes à RCA. Nesse trabalho foi realizado um processo de calibração empírico, no qual o algoritmo foi executado 5 vezes para cada uma das combinações de parâmetros descritas na Tabela 2. A configuração de melhor resultado médio foi escolhida para execução dos testes:  $TS = 50$ ;  $a = 2$ ;  $b = 6$ ;  $\alpha = 20$ ;  $p_c = 0,6$ ; e  $k = 20$ .

**Tabela 2 – Parâmetros do método de solução**

Parâmetro	Faixa de Calibração	Passo da Calibração	Valor Adotado
$TS$	[30 – 50]	10	50
$a$	[2 – 4]	1	2
$b$	[6 – 8]	1	6
$\alpha$	[10 – 20]	10	20
$p_c$	[0,6 – 0,8]	0,1	0,6
$k$	[10 – 20]	10	20

A seguir, o método foi executado 10 vezes para cada uma das 71 instâncias de teste. A Tabela 3 mostra os resultados obtidos.

**Tabela 3 – Resultados dos testes computacionais**

Subconjunto	Instâncias	Desvio Relativo Médio das 10 Execuções em Relação ao Ótimo (%)	Desvio Relativo da Melhor Solução Encontrada em Relação ao Ótimo (%)	Número de Ótimos Encontrados	Tempo Médio de Execução (s)
(1)	h1 – h24	0,14	0,04	21	3,22
(2)	h25 – h40	0,13	0,08	12	67,72
(3)	h41 – h55	0,63	0,44	8	13,69
(4)	h56 – h71	0,33	0,17	7	392,05
(1 – 4)	h1 – h71	0,28	0,16	48	107,59

A análise dos resultados mostra que o método proposto é competitivo, encontrando soluções de alta qualidade em um tempo de execução razoável (tempo médio de execução em torno de 100 segundos). Os desvios relativos médios das 10 execuções do algoritmo em relação à solução ótima são bastante reduzidos e esses valores se tornam ainda menores ao se considerar a melhor execução do algoritmo para cada instância. Além disso, 48 soluções ótimas foram encontradas para um conjunto de 71 instâncias, ou seja, aproximadamente 68% das instâncias foram resolvidas na otimalidade.

A fim de ratificar a eficiência do algoritmo proposto, os resultados obtidos foram comparados aos de dois métodos heurísticos propostos para o PLFRCU: a clássica heurística de pareamento repetido [Ronnqvist et al. 1999]; e, a mais recente das meta-heurísticas propostas, a busca tabu iterada [Ho 2015]. Essa análise comparativa se encontra na Tabela 4.

**Tabela 4 – Comparação de métodos heurísticos para o PLFRCU**

Subconjunto	Desvio Relativo da Melhor Solução Encontrada em Relação ao Ótimo (%)		
	<i>pareamento repetido</i>	<i>busca tabu + RCA</i>	<i>busca tabu iterada</i>
(1)	0,04	0,04	0,00
(2)	0,51	0,08	0,00
(3)	0,06	0,44	0,03
(4)	n/a	0,17	0,08
(1 – 3)	0,18	0,16	0,01
(1 – 4)	n/a	0,16	0,02
Outras Informações	<i>pareamento repetido</i>	<i>busca tabu + RCA</i>	<i>busca tabu iterada</i>
tempo médio solução (s) (1 – 3)	31,21	24,84	27,08
tempo médio solução (s) (1 – 4)	n/a	107,59	44,07
n° ótimos (1 – 3)	40	41	51
n° ótimos (1 – 4)	n/a	48	61

A análise comparativa dos métodos heurísticos para o PLFRCU confirma a competitividade do método híbrido. Esse algoritmo mostrou um desempenho ligeiramente superior ao da heurística de pareamento repetido e, apesar de ter obtido resultados inferiores à busca tabu iterada, evidencia-se que não existe uma grande lacuna entre seu desempenho e o da meta-heurística mais recentemente proposta para o PLFRCU na literatura. Assim, é possível confirmar que o uso do *framework* de memória adaptativa para a solução do PLFRCU consiste em uma abordagem promissora que pode evoluir, levando a obtenção de resultados estado-da-arte para o problema.

#### 4. Conclusões

Esse artigo aplicou, pela primeira vez na literatura do PLFRCU, um *framework* baseado em memória adaptativa para a solução desse problema. Para isso, a meta-heurística de RCA desenvolvida e aplicada com sucesso por [Tarantilis et al. 2013] a problemas de roteamento de veículos foi adaptada e unificada com uma meta-heurística busca tabu. A RCA parte de um conjunto inicial de soluções de referência construído pela busca tabu e aplica um procedimento adaptativo de recombinação de múltiplas soluções para a criação de soluções guia temporárias que geram trajetórias para um método de religação de caminho.

A fim de avaliar a eficácia do método, testes computacionais foram realizados com instâncias de *benchmark* e o algoritmo foi comparado com outros métodos relevantes da literatura. O método híbrido RCA e busca tabu se mostrou competitivo, revelando-se capaz de gerar soluções de alta qualidade em um tempo computacional razoável.

Melhorias ainda podem ser implementadas no algoritmo proposto visando à obtenção de resultados estado-da-arte. Para isso, sugerem-se os seguintes passos:

- Utilização de uma estrutura de dados apropriada (por exemplo, árvores binárias de busca) para armazenar e viabilizar a busca eficiente de soluções encontradas para cada configuração de facilidades durante a busca tabu, evitando que cálculos sejam refeitos para movimentos de abertura ou fechamento de facilidades;
- Utilização de memória de longo prazo para melhoria da busca tabu por meio de estratégias de intensificação e diversificação;
- Modificação do método de religação de caminho da RCA para que esse explore trajetórias adicionais entre as soluções de referência e as soluções guia geradas (na atual versão do algoritmo optou-se por explorar exclusivamente as trajetórias entre cada solução guia e a solução de referência que a gerou).

## Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil.

## Referências

- Ahuja, R. K., Orlin, J. B., Pallottino, S., Scaparra, M. P. e Scutellà, M. G. (2004). A multi-exchange heuristic for the single-source capacitated facility location problem. *Management Science*, 50(6): 749-760.
- Alfandari, L. (2007). Improved Approximation of the General Soft-Capacitated Facility Location Problem. *RAIRO Operations Research*, 41: 83-93.
- Alumur, S. e Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1): 1-21.
- Balas, E. (1982). A class of location, distribution and scheduling problems: modeling and solution methods. *Revue Belge Statistique, Inform Recherche Opl* 22: 36-65.
- Cardona-Valdés, Y., Álvarez, A. e Pacheco, J. (2014). Metaheuristic Procedure for a Bi-objective Supply Chain Design Problem with Uncertainty. *Transportation Research Part B*, 60: 66-84.
- Chen, C. H. e Ting, C. J. (2008). Combining Lagrangian heuristic and Ant Colony System to solve the Single Source Capacitated Facility Location Problem. *Transportation Research Part E*, 44: 1099-1122.
- Contreras, I. A. e Díaz, J. A. (2008). Scatter search for the single source capacitated facility location problem. *Annals of Operations Research*, 157: 73-89.
- Cornuéjols, G., Nemhauser, G. L. e Wolsey, L. A. (1990). The Uncapacitated Facility Location Problem. In Mirchandani, P. B. e Francis, R. L., editores, *Discrete Location Theory*, p. 119-171, Wiley.
- Cortinhal, M. J. e Captivo, M. E. (2003). Upper and lower bounds for the single source capacitated location problem. *European Journal of Operational Research*, 151: 333-351.
- Daskin, M. S. (2013). *Network and Discrete Location*, 2a edição, Wiley.
- Delmaire, H., Díaz, J. A., Fernández, E. e Ortega, M. (1999). Reactive GRASP and Tabu Search heuristics for the single source capacitated plant location problem. *Infor*, 37(3): 194-225.
- Devine, M. D. e Lesso, W. G. (1972). Models for the minimum cost development of offshore oil fields. *Management Science*, 18(8): 378-387.
- Díaz, J. A. e Fernández, E. (2002). A branch-and-price algorithm for the single source capacitated plant location problem. *Journal of the Operational Research Society*, 53(3): 728-740.
- Eiselt, H. A. e Marianov, V. (editores) (2011). *Foundations of location analysis*, Springer.
- Eiselt, H. A. e Marianov, V. (editores) (2015). *Applications of location analysis*, Springer.
- Filho, V. J. M. F. e Galvão, R. D. (1998). A tabu search heuristic for the concentrator location problem. *Location Science*, 6: 189-209.
- Glover, F. (1996). Tabu search and adaptive memory programming – Advances, applications and challenges. In Barr, R. S., Helgason, R. V. e Kennington, J. L., editores, *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, p. 1-75, Kluwer.
- Glover, F. e Laguna, M. (1997). *Tabu Search*, Kluwer.
- Guastaroba, G. e Speranza, M. G. (2014). A heuristic for BILP problems: The Single Source Capacitated Facility Location Problem. *European Journal of Operational Research*, 238: 438-450.

- Hale, T. S. e Moberg, C. R. (2003). Location Science Research: A Review. *Annals of Operations Research*, 123(1): 21-35.
- Hindi, K. S. e Pienkosz, K. (1999). Efficient solution of large scale, single source, capacitated plant location problems. *Journal of Operational Research Society*, 50: 268-274.
- Ho, S. C. (2015). An iterated tabu search heuristic for the Single Source Capacitated Facility Location Problem. *Applied Soft Computing*, 27: 169-178.
- Holmberg, K., Ronnqvist, M. e Yuan, D. (1999). An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113: 544-559.
- Klincewicz, J. e Luss, H. A. (1986). Lagrangian Relaxation Heuristic for Capacitated Facility Location with Single-Source Constraints. *Journal of Operational Research Society*, 37(5): 495-500.
- Laporte, G., Nickel, S. e Saldanha da Gama, F. (editores) (2015). Location Science, Springer.
- Lasdon, L., Duarte, A., Glover, F., Laguna, M. e Martí, R. (2010). Adaptive Memory Programming for Constrained Global Optimization. *Computers and Operations Research*, 37: 1500-1509.
- Li, X., Tian, P. e Aneja, Y. P. (2010). An Adaptive Memory Programming Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Transportation Research Parte E*, 46: 1111-1127.
- Martello, S. e Toth, P. (1981). An algorithm for the generalized assignment problem, In *Operational Research'81*, p. 589-603, North-Holland, Amsterdam, IFORS.
- Melo, M. T., Nickel, S. e Saldanha da Gama, F. (2009). Facility Location and Supply Chain Management – A Review. *European Journal of Operational Research*, 106: 401-412.
- Neebe, A. W. e Rao, M. R. (1983). An algorithm for the fixed-charge assigning users to sources problem, *Journal of Operational Research Society*, 34(11): 1107-1113.
- Olivera A. e Viera O. (2007). Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers and Operations Research*, 34(1): 28-47.
- Paraskevopoulos, D. C., Tarantilis, C. D. e Ioannou, G. (2016). An adaptive memory programming framework for the resource-constrained project scheduling problem. *International Journal of Production Research*. No prelo. Disponível em: <<http://dx.doi.org/10.1080/00207543.2016.1145814>>. Acesso em: 4 Maio 2016.
- Pirkul, H. (1987). Efficient Algorithms for the Capacitated Concentrator Location Problem. *Computational Operations Research*, 14(3): 197-208.
- Ronnqvist, M., Tragantalerngsak, S. e Holt, J. (1999). A repeated matching heuristic for the single-source capacitated facility location problems. *European Journal of Operational Research*, 116: 51-68.
- Ropke, S. e Pisinger, D. (2006). A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research*, 171: 750-775.
- Taillard, E. D., Gambardella, L. M., Gendreau, M. e Potvin, J-Y. (2001). Adaptive memory programming: a unified view of metaheuristics. *European Journal of Operational Research*, 135:1-16
- Tarantilis, C. D., Anagnostopoulou, A. e Repoussis, P. P. (2013). Adaptive Path Relinking for Vehicle Routing and Scheduling Problems with Product Returns. *Transportation Science*, 47(3): 356-379.
- Yang, Z., Chu, F. e Chen, H. (2012). A cut-and-solve based algorithm for the single source capacitated facility location problem. *European Journal of Operational Research*, 221(3): 521-532.