



## SOLUÇÃO DO PROBLEMA DA P-MEDIANA PARA LOCALIZAÇÃO DE ESTAÇÕES DE SERVIÇO UTILIZANDO METAHEURÍSTICA CHEMICAL REACTION OPTIMIZATION

**Danilo Cesar Azeredo Silva \***

Coordenadoria de Eletrotécnica, Campus Vitória  
Instituto Federal do Espírito Santo (IFES)  
Av. Vitória, nº 1729 - Bairro Jucutuquara, 29040-780, Vitória, ES – Brasil  
danilo@ifes.edu.br; daniloazeredo@gmail.com

**Mário Mestria \***

Coordenadoria de Engenharia Elétrica, Campus Vitória  
Instituto Federal do Espírito Santo (IFES)  
Av. Vitória, nº 1729 - Bairro Jucutuquara, 29040-780, Vitória, ES – Brasil  
mmestria@ifes.edu.br; mmestria@uol.com.br

\* PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS SUSTENTÁVEIS

Instituto Federal do Espírito Santo (IFES)  
Av. Vitória, nº 1729 - Bairro Jucutuquara, 29040-780, Vitória, ES – Brasil

### RESUMO

A Chemical Reaction Optimization (CRO) é uma metaheurística, para resolução de problemas de otimização, que imita as interações entre moléculas em uma reação química com o objetivo de atingir um estado de baixa energia estável. No presente trabalho a metaheurística CRO é utilizada para resolver de forma eficiente o problema da p-mediana para determinação da localização de estações de serviço. Resultados obtidos na resolução de 40 instâncias clássicas de teste da biblioteca *OR-Library* são comparados com os seus valores ótimos e também com aqueles obtidos da literatura na resolução por relaxação Lagrangiana, *Variable Neighborhood Search* e Algoritmo Genético. Os resultados demonstram que o CRO é bastante competitivo em relação a várias metaheurísticas de sucesso existentes, tendo superado as mesmas em alguns casos.

**PALAVRAS CHAVE.** Chemical Reaction Optimization, p-mediana, metaheurística.

**Tópicos.** MH - Metaheurísticas, L&T - Logística e Transportes, OC - Otimização Combinatória

### ABSTRACT

Chemical Reaction Optimization (CRO) is a metaheuristic, for solving optimization problems, which mimics the interactions between molecules in a chemical reaction with the purpose of achieving a stable, low-energy state. In the present work, the CRO metaheuristics is utilized for solving, in an efficient manner, the p-median problem, in order to locate service stations. Results from solving 40 classic instances of the *OR-Library* are compared to their optimal values, and to those obtained from the literature using Lagrangian relaxation, *Variable Neighborhood Search* and Genetic Algorithm. Results show that CRO is competitive with many existing successful metaheuristics, having exceeding those in some cases.

**KEYWORDS.** Chemical Reaction Optimization, p-Median, metaheuristics.

**Paper topics.** MH – Metaheuristics, L & T - Logistics and Transport, OC - Combinatorial Optimization.



## 1 Introdução

O layout e planejamento de estações de serviço é um tópico importante e que possui uma variedade de aplicações na vida real. Ambos os setores público e privado da economia são constantemente desafiados com problemas que envolvem decisões de *layout* envolvendo tais estações. Uma parte significativa dos problemas desta natureza envolve a determinação dos melhores locais para a instalação de estações de serviço, baseada em requerimentos de oferta e demanda. Tais problemas possuem muitas aplicações, incluindo a localização de estabelecimentos comerciais, escolas, hospitais, centros de ambulâncias, estações de bombeiros, caixas eletrônicos bancários, postos de gasolina, estações base de telefonia sem fio, estações de carregamento de veículos elétricos, dentre outras.

Parâmetros de projeto relativos ao problema de localização de estações são bastante diversos e podem incluir o número de estações de serviço que devem ser criadas, onde cada estação deve se situar, quão grande cada estação deve ser, além da maneira como a demanda de consumo deve ser alocada à cada estação.

A modelagem do problema de localização de estações tem sido amplamente investigada na literatura. Diferentes tipos de modelos matemáticos para solucionar o problema foram criados, incluindo os modelos *set-covering* [Toregas et al. 1971], cobertura máxima [Church e ReVelle 1974], p-centro e p-mediana [Hakimi 1964].

No presente trabalho, a metaheurística *Chemical Reaction Optimization* (CRO) proposta por [Lam e Li 2012] foi adaptada para a solução do problema da p-mediana, que é um dos modelos mais populares na literatura sobre localização de estações de serviço.

O problema da p-mediana se constitui em localizar  $p$  estações de serviço para servir  $n$  centros de demanda de forma a minimizar a distância média ponderada entre os centros de demanda e as estações selecionadas mais próximas aos mesmos. As primeiras formulações do problema foram apresentadas por [Hakimi 1964]. Tal problema é bem conhecido na literatura como sendo NP-difícil [Kariv e Hakimi 1979]. Dessa forma, vários métodos que exploram buscas em árvore foram desenvolvidos de modo a resolver o problema em tempo polinomial. O uso de técnicas de relaxação Lagrangiana e otimização por subgradientes combinadas, de um ponto de vista primal-dual, tem se mostrado como uma boa abordagem para a resolução do problema [Galvão e Raggi 1989].

Outras abordagens comumente utilizadas são as providas por metaheurísticas, as quais permitem obter rapidamente uma solução aproximada do problema da p-mediana ou, em alguns casos, uma solução ótima, porém sem que seja possível provar a sua otimalidade. Tais heurísticas incluem versões da Busca Tabu [Rolland et al. 1997], *Heuristic Concentration* [Rosing e ReVelle 1997], Algoritmo Genético [Alp et al. 2003] e *Simulated Annealing* [Al-Khedhairi 2008]. Entretanto, cada uma delas possui suas próprias limitações. Algumas são difíceis de entender e implementar, requerendo entrada e armazenamento de informações em formatos especializados. Adicionalmente, a obtenção de soluções de boa qualidade pode ser bastante sensível aos padrões de modelagem.

## 2 O problema da p-mediana

O problema da p-mediana considerado no presente trabalho visa localizar estações de serviço, a partir de um conjunto de locais candidatos a receberem tais estações e de um conjunto de centros de demanda. O modelo de programação inteira mista utilizado é mostrado abaixo:

**Variáveis de decisão:**

$$x_j = \{1 \text{ se a estação candidata } j \text{ foi selecionada; ou } 0, \text{ do contrário} \quad (1)$$

$$y_{ij} = \{1 \text{ se o centro de demanda } i \text{ é coberto pela estação } j; \text{ ou } 0, \text{ do contrário} \quad (2)$$



**Minimizar:**

$$\sum_{i=1}^m \sum_{j=1}^n d_{ij} y_{ij} \quad (3)$$

**Sujeito a:**

$$\sum_{j=1}^n y_j = p, \forall j \in J \quad (4)$$

$$\sum_{j=1}^n y_{ij} = 1, \forall i \in I \quad (5)$$

$$y_{ij} - x_j \leq 0, \forall i \in I, \forall j \in J \quad (6)$$

$$x_j \in \{0,1\}, \forall j \in J \quad (7)$$

$$y_{ij} \in \{0,1\}, \forall i \in I, \forall j \in J \quad (8)$$

**Onde:**

$J$  : conjunto de locais candidatos a receber as estações de serviço;

$I$  : conjunto de centros de demanda;

$d_{ij}$  : distância, ou custo, de um centro de demanda  $i$  a uma estação de serviço  $j$ ;

$p$  : número de estações a serem abertas (alocadas);

A função (3) representa a função de otimização. O objetivo é minimizar o custo total da demanda ponderada. A restrição (4) estabelece que o número de estações a serem abertas seja igual a  $p$ . A restrição (5) estabelece que toda a demanda do centro de demanda  $i$  deve ser satisfeita. A restrição (6) assegura que cada centro de demanda só possa ser associado a uma estação que tenha sido selecionada. Finalmente, as restrições (7) e (8) estabelecem que as variáveis de decisão devem ser binárias (0 ou 1) e não-negativas.

### 3 A metaheurística CRO adaptada ao problema da p-mediana

A metaheurística CRO é uma técnica que relaciona, de forma aproximada, reações químicas com otimização e se baseia nas duas primeiras leis da termodinâmica [Lam e Li 2012]. A primeira lei, a da conservação da energia, diz que a energia não pode ser criada ou destruída, mas apenas transformada de uma forma para outra ou transferida uma entidade para outra. A segunda lei diz que a entropia, que é a medida do grau de desordem de um sistema, tende a aumentar.

Um sistema de reação química consiste das substâncias químicas e seu ambiente. A energia do ambiente é simbolicamente representada no CRO por um reservatório de energia central (*buffer*). Uma substância química é composta de moléculas, que possuem energia potencial e cinética. Uma reação química ocorre quando o sistema é instável, no sentido de possuir energia excessiva. Todos os sistemas de reação química tendem a alcançar um estado de equilíbrio, no qual a energia potencial cai para um mínimo. O CRO simula esse fenômeno convertendo energia potencial em energia cinética, pela transferência gradual da energia das moléculas para o ambiente, através de passos consecutivos, ou sub-reações, passando por vários estados de transição, que resultam em produtos mais estáveis e com energia mínima. É um processo iterativo em busca do ponto ideal.

Uma alteração química em uma molécula é desencadeada por uma colisão. Existem dois tipos de colisões: unimoleculares e intermoleculares. No CRO, a primeira descreve a situação em que uma molécula colide com a parede de um recipiente, enquanto que a última representa os casos em que uma molécula colide com outras moléculas. Tal alteração química é chamada de reação elementar. Uma reação elementar ineficaz é aquela que resulta em uma mudança sutil da estrutura molecular. O CRO considera quatro tipos de reações elementares: colisão ineficaz com a parede, decomposição, colisão ineficaz intermolecular e síntese [Lam e Li 2012]. Com respeito às moléculas, a decomposição e a síntese estão relacionadas com a colisão ineficaz com a parede e a



colisão ineficaz intermolecular, respectivamente, mas provocam mudanças muito mais vigorosas nas estruturas moleculares. As reações moleculares elementares estão resumidas no Quadro 1 e ilustradas na Figura 1.

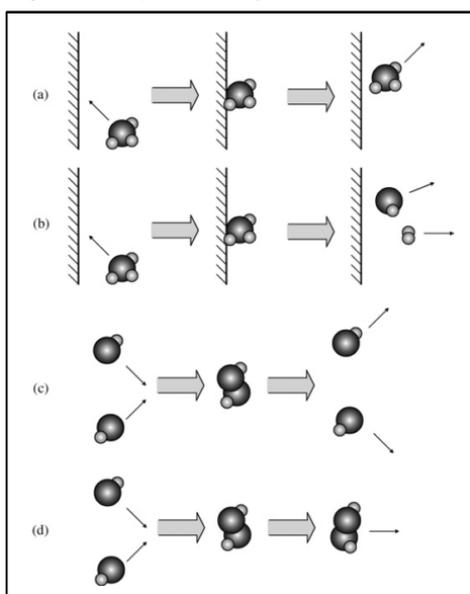
Soluções são manipuladas através de uma sequência aleatória de reações. As duas colisões ineficazes implementam busca local (intensificação), enquanto que a decomposição e a síntese dão o efeito de diversificação. Uma mistura adequada de intensificação e diversificação permite fazer uma busca efetiva do mínimo global dentro do espaço de solução. A seguir estão descritos, resumidamente, os principais componentes do CRO, bem como os detalhes particulares da presente implementação para o problema da p-mediana. Maiores informações sobre a metaheurística CRO podem ser obtidas em [Lam e Li 2012].

Quadro 1 - Características das quatro reações elementares do CRO

Características das quatro reações elementares		
Extensão da mudança	Número de moléculas envolvidas	
	Unimolecular	Intermolecular
Maior	Decomposição	Síntese
Menor	Colisão ineficaz com a parede	Colisão ineficaz intermolecular

Fonte: [Lam e Li 2012]

Figura 1 - Quatro reações elementares do CRO



Fonte: [Lam e Li 2012]

(a) Colisão ineficaz com a parede; (b) Decomposição;  
(c) Colisão ineficaz intermolecular; (d) Síntese.

O CRO é uma metaheurística populacional variável. Desta forma, o número total de moléculas em diferentes iterações pode não ser o mesmo. Nas colisões ineficazes, o número de moléculas antes e depois da alteração permanece idêntico. Já na decomposição esse número aumenta e na síntese, diminui. É possível influenciar indiretamente as frequências de decomposição e síntese alterando os parâmetros do CRO denominados  $\alpha$  e  $\beta$ , respectivamente.

As reações elementares definem as implementações das interações moleculares. No presente trabalho o CRO foi implementado usando a linguagem de programação orientada a objetos C# [ECMA 2006], dada a facilidade de modelagem das moléculas como instâncias de uma classe que contém todos os atributos necessários ao seu funcionamento. As reações elementares foram implementadas na forma de métodos de uma classe que contém os códigos dos diferentes algoritmos utilizados nas reações.



### 3.1 A molécula

O CRO é um algoritmo multiagente, cujos agentes manipulados são moléculas. A molécula é a unidade básica do CRO, e contém vários atributos requeridos para as operações do CRO:

- Estrutura molecular ( $\omega$ ): armazena uma solução do problema, ou seja, um valor para a função objetivo, além das variáveis de decisão  $x$  e  $y$  do problema da p-mediana. As variáveis  $x$  e  $y$  foram implementadas como listas encadeadas de inteiros;
- Energia potencial ( $PE$ ): é definida como o valor da função objetivo da solução correspondente, representada por  $\omega$ . Se  $f$  denota a função objetivo, então temos que:

$$PE\omega = f(\omega)$$

- Energia cinética ( $KE$ ): é um número não negativo que quantifica a tolerância do sistema em aceitar uma solução pior do que a existente;
- Número de colisões ( $NumHit$ ): número total de colisões que uma molécula já sofreu;
- Estrutura mínima ( $MinStruct$ ): é a solução ( $\omega$ ) com energia potencial ( $PE$ ) mínima, que uma molécula atingiu, até agora. Preserva a estrutura com o menor  $PE$  no histórico de reações;
- Energia potencial mínima ( $MinPE$ ): é o  $PE$  correspondente ao  $MinStruct$ ;
- Número mínimo de colisões ( $MinHit$ ): é o número da colisão em que a molécula atingiu o menor valor de  $PE$  ( $MinPE$ ).

### 3.2 As Reações Elementares

Existem quatro tipos de reações elementares que podem ocorrer a cada iteração do CRO. Elas são empregadas para manipular soluções (explorar o espaço de solução) e redistribuir energia entre as moléculas e o *buffer* de energia. Operadores são usados para modificar as soluções ou gerar novas soluções a partir das soluções correntes. No entanto, o CRO sempre garante a conservação da energia quando novas soluções são geradas através dos operadores.

Para as colisões ineficazes com a parede e as intermoleculares, foi utilizado um operador baseado no algoritmo **Fast Interchange**, proposto por [Whitaker 1983]. Nas sínteses, um operador do tipo *crossover* comumente empregado em Algoritmos Genéticos (GA) foi utilizado. Finalmente, nas decomposições foi empregado o operador **Half-Total change**. Como o nome sugere, uma nova solução é gerada a partir de uma solução existente, mantendo metade dos valores (estações selecionadas) da solução corrente e atribuindo novos valores à metade restante. A seguir estão descritas as reações elementares previstas no CRO, bem como detalhes de sua implementação no presente trabalho:

- **Colisão ineficaz com a parede**

Representa a situação na qual uma molécula colide com uma parede do recipiente e é, então, rebatida, permanecendo uma única molécula. Nessa colisão, a solução existente  $\omega$  é perturbada transformando-se em  $\omega'$ , isto é:

$$\omega \rightarrow \omega'$$

Isso é feito gerando-se um  $\omega'$  que esteja na vizinhança de  $\omega$ , através do operador *Fast Interchange* (FI). Seja  $FI(.)$  um operador de busca em vizinhança, do tipo *Fast Interchange*. Assim, temos que  $\omega' = FI(\omega)$  e  $PE\omega' = f(\omega')$ . Nesse tipo de reação, tipicamente, ocorrerá uma perda de energia potencial, ou seja,  $PE\omega'$  será menor que  $PE\omega$ , indicando que uma solução melhor foi obtida. Caso isto não ocorra e  $PE\omega'$  seja maior que  $PE\omega$ , ainda assim a solução pior poderá ser aceita desde que  $PE\omega + KE\omega \geq PE\omega'$ . No entanto, toda vez que uma reação elementar ocorre, uma certa quantidade de energia cinética ( $KE$ ) é transferida para o *buffer* de energia, diminuindo as chances de que soluções piores sejam aceitas à medida que as iterações ocorrem. A quantidade de energia cinética da molécula obtida a partir da reação ineficaz é indiretamente controlada pelo parâmetro *KElossRate*, que é um valor entre 0 e 1, inclusive, e determina a quantidade mínima de energia cinética que será aproveitada da solução original ( $\omega$ ). Por exemplo, se à *KElossRate* for atribuído o valor 0,8, um mínimo de 80% da energia cinética que a molécula possuía antes da



colisão será transferida para a molécula resultante, após a colisão. A diferença de energia cinética será, então, transferida para o *buffer* central de energia.

- **Decomposição**

Ocorre quando uma molécula ( $\omega$ ) colide com uma parede e, em seguida, quebra-se em duas partes, produzindo  $\omega'_1$  e  $\omega'_2$ , ou seja:

$$\omega \rightarrow \omega'_1 + \omega'_2$$

O objetivo da decomposição é permitir que o sistema explore outras regiões do espaço de solução, após ter efetuado considerável busca local através de colisões ineficazes. No presente trabalho, utilizou-se o algoritmo *Half-total change* para gerar novas soluções. Uma vez que são criadas mais soluções, a soma total de *PE* e *KE* da molécula original pode não ser suficiente. Em outras palavras, podemos ter  $PE\omega + KE\omega < PE\omega'_1 + PE\omega'_2$ . Como a conservação de energia não é satisfeita nestas condições, esta decomposição deve ser abortada. Para aumentar a chance de ter-se uma decomposição concluída, uma pequena porção de energia do *buffer* é retirada para apoiar a mudança.

- **Colisão ineficaz intermolecular**

Ocorre quando múltiplas moléculas colidem umas com as outras e, em seguida, se afastam. O número de moléculas permanece inalterado, ou seja:

$$\omega_1 + \omega_2 \rightarrow \omega'_1 + \omega'_2$$

Esta reação elementar é muito semelhante à colisão ineficaz unimolecular.  $\omega'_1$  e  $\omega'_2$  são obtidos através de  $\omega'_1 = FI(\omega_1)$  e  $\omega'_2 = FI(\omega_2)$ . O gerenciamento de energia é semelhante ao da colisão ineficaz com a parede, mas não envolve o *buffer*. Uma vez que mais moléculas estão envolvidas, a soma total de energia do subsistema molecular é maior do que a da colisão ineficaz com a parede. Assim, a probabilidade de as moléculas explorarem seu ambiente imediato é maior, já que as mesmas têm maior flexibilidade para serem transformadas em estruturas moleculares mais diversas. No presente trabalho utilizou-se o mesmo operador *Fast Interchange*, aplicando-se o operador a cada molécula para obter-se uma nova.

- **Síntese**

A síntese é o oposto da decomposição. Uma síntese acontece quando duas moléculas colidem uma contra a outra e se fundem, isto é:

$$\omega_1 + \omega_2 \rightarrow \omega'$$

Nessa reação, permite-se uma mudança muito maior para  $\omega'$ , em relação a  $\omega_1$  e  $\omega_2$ , bem como um aumento considerável da energia cinética da molécula resultante. Assim a mesma possui uma maior "capacidade" de explorar sua região de soluções, devido a sua energia cinética superior. A ideia por trás da síntese é a diversificação de soluções. No presente trabalho, um operador *crossover* comumente empregado em Algoritmos Genéticos (GA) foi utilizado.

- **Conservação de energia:**

Um dos pressupostos fundamentais do CRO é a conservação da energia, o que significa que a energia não pode ser criada e nem destruída. O sistema, em sua totalidade, compreende a energia de todas as moléculas definidas somada a energia contida no recipiente, que está associada ao *buffer*. A quantidade total de energia de todo o sistema é determinada pelos valores da função objetivo, isto é, a *PE* da população inicial de moléculas, cujo tamanho é *PopSize*, o *KE* inicial (*InitialKe*) atribuído às moléculas e valor inicial de energia do *buffer*.

### 3.3 Inicialização

Inicialmente, foram criadas variáveis escalares correspondentes aos parâmetros operacionais do CRO: *PopSize*, *KELossRate*, *MoleColl*, *buffer*, *InitialKE*,  $\alpha$  e  $\beta$ . A seguir, definiu-se a classe "Molécula" com seus atributos  $\omega$ , *PE*, *KE*, *NumHit*, *MinStruct*, *MinPE* e *MinHit*, o seu construtor de classe, além de quatro métodos correspondentes às reações elementares, conforme ilustrado na Figura 2. O construtor deve gerar uma solução inicial viável a partir dos dados do problema, ou seja, que não viole as restrições do modelo da p-mediana, ilustrado



no item 2. Utilizando-se do mesmo, gerou-se uma população inicial de moléculas, com o tamanho especificado por *PopSize*.

### 3.4 Colisões

Uma molécula pode atingir uma parede do recipiente ou colidir com uma outra molécula. Isto é decidido gerando-se um número aleatório  $b$  entre  $[0, 1]$ . Se  $b > MoleColl$  ou se o sistema tiver apenas uma molécula, uma colisão unimolecular ocorrerá. Caso contrário, segue-se uma colisão intermolecular. Para uma colisão unimolecular, uma molécula da população é selecionada aleatoriamente  $e$ , então, decide-se se ocorrerá com ela uma colisão ineficaz com a parede ou uma decomposição, de acordo com o critério de decomposição escolhido. O critério de decomposição foi definido como:

$$NumHit - MinHit > \alpha \quad (09)$$

De forma análoga, para uma colisão intermolecular, seleciona-se aleatoriamente duas moléculas da população  $e$ , então, determina-se se haverá uma colisão ineficaz intermolecular ou uma síntese verificando-se o critério de síntese para as moléculas escolhidas. A síntese ocorrerá se todas as moléculas atenderem o seguinte critério:

$$KE \leq \beta \quad (10)$$

As desigualdades (09) e (10) controlam o grau de diversificação através dos parâmetros  $\alpha$  e  $\beta$ . Valores adequados de  $\alpha$  e  $\beta$  promovem um equilíbrio adequado entre diversificação e intensificação.

Após a ocorrência de uma reação elementar, deve-se verificar se a condição de conservação de energia é obedecida. Caso isso não haja ocorrido a mudança deve ser descartada. A seguir, é verificada se qualquer uma das soluções produzidas naquela operação possui um valor de função objetivo mais baixo do que a melhor solução obtida até o momento. Se assim o for, registra-se tal solução como a melhor até o momento.

Se nenhum critério de parada for atingido, inicia-se uma nova iteração, após armazenar-se a melhor solução dentre as melhores soluções de toda a população, ou seja, após armazenar-se o *MinStruct* com o menor *MinPE* entre todas as moléculas.

### 3.5 Finalização do algoritmo

Se algum dos critérios de parada for atingido, segue-se para a fase final. Os critérios de parada utilizados foram o número máximo de iterações e o número máximo de iterações realizadas sem melhorias, ou seja, sem que o menor *MinPE* entre todas as moléculas tenha mudado. Na fase final, a melhor solução encontrada é exibida e o algoritmo, finalizado. O pseudocódigo principal do CRO é mostrado na Figura 3.

### 3.6 Operadores

A seguir estão descritos os operadores usados nas transformações ineficazes uni e multimoleculares, nas sínteses e nas decomposições:

#### 3.6.1 Operador *Fast Interchange*

Nas colisões ineficazes uni e multimoleculares um operador baseado no algoritmo *Fast Interchange* proposto por [Whitaker 1983] foi utilizado. No trabalho de Whitaker, três ingredientes eficientes são incorporados na heurística de troca (*interchange*) padrão:

- Avaliação de movimento, onde a melhor estação a ser removida é encontrada quando a estação a ser adicionada é conhecida;
- Atualização da primeira e da segunda estação mais próximas de cada centro de demanda;
- Estratégia restrita a primeira melhoria, onde cada estação é considerada para ser adicionada apenas uma vez. [Hansen e Mladenović 1997] implementaram uma versão modificada do algoritmo *Fast Interchange* para ser usada na metaheurística *Variable Neighborhood Search* (VNS), modificando-o para uma estratégia de obtenção da maior melhoria. Na presente implementação, o algoritmo modificado por [Hansen e Mladenović 1997] foi utilizado, sendo



também possível configurá-lo, através de um parâmetro de entrada, para terminar assim que a primeira melhoria houver sido obtida, ou executá-lo até que a melhor melhoria tenha sido alcançada.

### 3.6.2 Operador *Half-total change*

É o operador utilizado para efetuar a decomposição de uma molécula em duas outras moléculas, conforme descrito no item 3.2. Como o seu nome implica, uma nova solução é produzida a partir de uma já existente, mantendo-se metade dos valores existentes e atribuindo-se novos valores à metade restante.

Suponha que se tente produzir duas novas soluções  $\omega'_1 = [\omega'_1(i), 1 \leq i \leq n]$  e  $\omega'_2 = [\omega'_2(i), 1 \leq i \leq n]$  a partir de  $\omega = [\omega(i), 1 \leq i \leq n]$ . Para obter-se  $\omega'_1$ , primeiro copia-se  $\omega$  para  $\omega'_1$  e, em seguida, escolhe-se aleatoriamente  $\lfloor N/2 \rfloor$  elementos no vetor  $\omega'_1$ , onde  $\lfloor . \rfloor$  retorna o maior inteiro igual ou menor que o argumento. Para cada um destes elementos, por exemplo  $\omega'_1(i)$ , atribui-se um novo valor de acordo com as restrições do problema. Se  $\omega'_1$  só pode ter um valor no conjunto de estações candidatas,  $S_i$ , pode-se simplesmente selecionar aleatoriamente um elemento de  $S_i$  para  $\omega'_1(i)$ . Como os elementos são escolhidos aleatoriamente,  $\omega'_1$  e  $\omega'_2$  são bastante diferentes entre si, e também de  $\omega$ . Na presente implementação, o operador *Half-total change* assegura que os elementos de  $\omega$  estejam presentes em  $\omega'_1$  ou  $\omega'_2$ , mas não em ambos. Isto garante que todas as estações pertencentes a  $\omega$  estejam presentes, seja em  $\omega'_1$  ou  $\omega'_2$ .

Figura 2 - Classe molécula

```
"Molecule" class
class Molecule
Attributes:  $\omega$ , PE, KE, NumHit, MinStruct, MinPE, MinHit
Method:
Molecule() \constructor
{
  Randomly generate  $\omega$  in the solution space
  PE  $\leftarrow$  f( $\omega$ )
  KE  $\leftarrow$  InitialKE
  NumHit  $\leftarrow$  0 and MinStruct  $\leftarrow$   $\omega$  and MinPE  $\leftarrow$  PE and MinHit  $\leftarrow$  0
}
OnwallIneffectiveCollision()
Decomposition()
IntermolecularIneffectiveCollision()
Synthesis()
end class
```

Fonte: [Lam e Li 2012]

### 3.6.3 Operador *Distance Preserving Crossover (DPX)*

É o operador usado para efetuar a síntese de duas moléculas em uma única molécula, conforme descrito no item 3.2. O operador DPX foi utilizado por [Merz e Freisleben 1997] para resolução do problema de assinalamento quadrático, através de Algoritmo Genético, se mostrando bem adaptado ao problema da p-mediana, uma vez que se baseia unicamente na noção de distância entre soluções. Sejam  $\pi_1$  e  $\pi_2$  soluções válidas para um dado problema. A distância  $D$ , entre soluções é definida como:

$$D(\pi_1, \pi_2) = |\{i \in \{1, \dots, n\} | \pi_1(i) \neq \pi_2(i)\}| \quad (11)$$

Tal como definido em (11),  $D$  representa quantidade de estações presentes em  $\pi_1$  que não se encontram em  $\pi_2$ . O DPX tem como objetivo produzir um descendente que possua a mesma distância de cada um de seus pais, sendo esta distância igual à distância entre os próprios pais. Sejam dois pais A e B, contendo soluções viáveis. Primeiro, todas as atribuições de estações contidas em ambos os pais são copiadas para o descendente C. As posições restantes são então aleatoriamente preenchidas com estações ainda não atribuídas, assegurando-se que nenhuma atribuição que possa ser encontrada em apenas um dos pais seja inserida no descendente. Desta



forma, obtém-se um descendente  $C$ , para qual a condição  $d = D(C,A) = D(C,B) = D(A,B)$  se mantém. Tal cruzamento é altamente disruptivo, forçando as subseqüentes buscas locais a explorarem uma região diferente do espaço de solução, onde melhores soluções poderão ser encontradas.

Figura 3 - Pseudocódigo principal do CRO

```
Implementation of CRO algorithm
Input: Objective function  $f$  and the parameter values
\\ Initialization
Set PopSize, KELossRate, MoleColl, buffer, InitialKE,  $\alpha$  and  $\beta$ 
Create PopSize number of molecules
\\ Iterations
while the stopping criteria not met do
    Generate  $b$  in  $[0, 1]$ 
    if  $b > MoleColl$  then
        Randomly select one molecule  $M\omega$ 
        if Decomposition criterion (09) met then
            Trigger Decomposition
        else
            Trigger OnwallIneffectiveCollision
        end if
    else
        Randomly select two molecules  $M\omega1$  and  $M\omega2$ 
        if Synthesis criterion (10) met then
            Trigger Synthesis
        else
            Trigger IntermolecularIneffectiveCollision
        end if
    end if
    Check for any new minimum solution
end while
\\ The final stage
Output the best solution found and its objective function value
```

Fonte: [Lam e Li 2012]

#### 4 Metodologia

A metaheurística CRO para a resolução do problema da p-mediana foi implementada na forma de uma aplicação codificada na linguagem de programação *Microsoft Visual C# 2015*.

A base de dados *OR-Library* [Beasley 1990], amplamente utilizada pela comunidade científica para testes e validação de problemas de p-mediana, foi utilizada para avaliar a qualidade e desempenho da presente implementação. A base contém 40 problemas do tipo p-mediana, com tamanhos entre 100 e 900 estações/consumidores e 5 a 100 medianas.

Inicialmente, cada um dos 40 problemas da *OR-Library* foi resolvido, de forma exata, utilizando o otimizador IBM CPLEX [IBM 2017], rodando em um computador PC com Windows 10, processador Intel Core i7 de oito núcleos de 2.3 GHz e 16GB de memória RAM. A seguir, na mesma plataforma computacional, cada problema foi resolvido 20 vezes pelo aplicativo CRO, utilizando-se a mesma configuração para todos os problemas. A configuração dos parâmetros operacionais do CRO utilizadas nos problemas foi a seguinte: **PopSize = 10, KELossRate = 0,8, MoleColl = 0,2, InitialKE = 100.000,  $\alpha = 1$ ,  $\beta = 10.000$  e buffer = 0**. Como critério de parada, limitou-se o número máximo de iterações em 5.000 e o número máximo de iterações realizadas sem melhorias em 500 iterações.

Durante os testes, observou-se que os melhores resultados, em termos de desempenho, foram obtidos com populações iniciais pequenas, entre 10 e 100 moléculas. Isto permitiu que ocorresse intensificação suficiente, ou seja, que existissem colisões unimoleculares e multimoleculares ineficazes em quantidade tal que se pudesse alcançar os mínimos locais entre as vizinhanças, sem que o número de iterações aumentasse demasiadamente.



## 5 Resultados computacionais

Nos experimentos, o CRO atingiu o valor ótimo da função objetivo em 35 dos 40 problemas da biblioteca *OR-Library*, sendo que o maior *gap* foi de apenas 0,25% na resolução do problema pmed36. Os demais *gaps* ficaram abaixo de 0,06%. Neste trabalho, *gap* é definido como a diferença percentual entre o valor ótimo e o melhor resultado obtido pelo método heurístico ou exato avaliado.

Uma vez que a mesma plataforma computacional foi utilizada para resolver os problemas, de forma exata, pelo CPLEX e pelo aplicativo que implementa o CRO, uma comparação precisa de desempenho (tempos de execução) entre os dois métodos pôde ser efetuada. Na comparação com outras metaheurísticas, foi efetuada somente uma avaliação de precisão de resultados e robustez, uma vez que hardware, linguagens de programação e sistemas operacionais no qual os diferentes algoritmos foram implementados, diferem significativamente.

Na comparação com os resultados obtidos na resolução pelo otimizador CPLEX, para os problemas menores (pmed01 a pmed20), o CRO foi cerca de duas vezes mais rápido, com uma média de 12,14s contra 24,63s do CPLEX. Para os problemas de maior dimensão (pmed21 a 40), a média dos tempos de execução do CRO foi de 124,1s, contra 8157,68s do CPLEX. A média geral de tempos de execução do CRO foi de 68,14s, contra 4091,16s do CPLEX, portanto, cerca de 60 vezes mais rápida que as obtidas pelo otimizador da IBM. Entretanto, em 13 dos problemas avaliados o CPLEX apresentou um melhor desempenho, especialmente naqueles em que o tempo total para obtenção dos valores ótimos não excedeu a 3 minutos e a relação entre o número de medianas,  $p$ , e o número de vértices era baixo, entre 3 e 5, na maioria dos casos. Contudo, a maior diferença não excedeu a 101s, ficando, na média, em 21,19s.

Os resultados obtidos pelo CRO foram comparados com aqueles obtidos por [Daskin e Maass 2015] na resolução dos mesmos 40 problemas da *OR-Library*, utilizando Relaxação Lagrangiana (RL). Embora os resultados da resolução por esta técnica tenham permitido atingir valores ótimos em 39 dos 40 problemas, no problema pmed40 o *gap* foi de 19,5%, versus 0,04% do CRO. Para a resolução do problema pmed36 o tempo de resolução foi de aproximadamente 768s, contra 43s requeridos pelo CRO. O tempo médio de execução do CRO para todos os problemas foi de 68,14s, ligeiramente mais lento do que os 44,87s obtidos pela relaxação Lagrangiana. No entanto, o CRO provou ser mais rápido em 18 dos 40 problemas. Deve-se notar que esta comparação de desempenho é apenas aproximada, uma vez que as plataformas computacionais são semelhantes, mas não idênticas.

Os resultados obtidos também foram comparados com aqueles obtidos na resolução de 22 problemas da biblioteca *OR-Library*, por [Hansen e Mladenović 1997], utilizando a metaheurística *Variable Neighborhood Search* (VNS). Apenas no problema pmed30, o *gap* obtido pelo VNS foi ligeiramente menor que no CRO, ou seja, 0,15% versus 0,25%, respectivamente. Para o problema pmed40, o *gap* foi exatamente o mesmo: 0,04%.

Por fim, os resultados foram comparados com os obtidos por [Alp et al. 2003], que propuseram uma versão eficiente de Algoritmo Genético (GA) para a resolução do problema da  $p$ -mediana. Os autores tiveram sucesso em atingir o valor ótimo da função objetivo em somente 28 dos 40 problemas da *OR-Library*. Em todos os problemas, a metaheurística GA apresentou *gaps* maiores que os obtidos pelo CRO. A média dos *gaps* do CRO foi de 0,0108% contra 0,0363% do GA, portanto, 3,36 vezes maior que a obtida pelo GA. Em seu trabalho, [Alp et al. 2003] também forneceram o desvio médio para todos os problemas resolvidos por algoritmo GA. Sendo assim, foi possível comparar os resultados com aqueles obtidos pelo CRO. A somatória dos desvios médios do GA foi de 4,40% contra 2,61% do CRO.

Os resultados obtidos pelo otimizador CPLEX, bem como aqueles obtidos na resolução por CRO, RL, VNS e GA estão sumarizados na Tabela 1. Para todas as técnicas são apresentados o *gap*, exceto para o CPLEX, já que os valores da função objetivo são sempre os ótimos. Para o CRO e GA estão listados também o desvio médio para os diferentes problemas da *OR-Library*.



Tabela 1 - Comparação de métodos para resolução dos 40 problemas da *OR-Library*

<i>Pmed</i>	<i>OR-Library</i>		<i>CPLEX</i>		<i>CRO</i>		<i>RL</i>		<i>VNS</i>		<i>GA</i>	
	<i>p</i>	<i>Tam</i>	Valor ótimo	Duração(s)	Gap (%)	Desvio Médio (%)	Dur, Méd. (s)	Gap (%)	Dur, Méd. (s)	Gap (%)	Gap (%)	Desvio Médio (%)
1	5	100	5819	0,39	0,00	0,00	0,40	0,00	2,94		0,00	0,14
2	10	100	4093	0,75	0,00	0,03	0,53	0,00	8,92	0,00	0,00	0,20
3	10	100	4250	0,50	0,00	0,00	0,48	0,00	7,70		0,00	0,33
4	20	100	3034	0,33	0,00	0,03	0,77	0,00	3,06	0,00	0,00	0,29
5	33	100	1355	0,31	0,00	0,00	0,91	0,00	3,03		0,00	0,00
6	5	200	7824	6,33	0,00	0,00	1,20	0,00	15,09		0,00	0,03
7	10	200	5631	3,31	0,00	0,01	1,91	0,00	3,08	0,00	0,00	0,21
8	20	200	4445	2,61	0,00	0,00	3,37	0,00	3,09	0,00	0,00	0,00
9	40	200	2734	2,56	0,00	0,07	5,68	0,00	14,73	0,00	0,00	0,30
10	67	200	1255	2,36	0,00	0,03	6,99	0,00	5,31	0,00	0,08	0,29
11	5	300	7696	28,53	0,00	0,00	3,54	0,00	4,81		0,00	0,16
12	10	300	6634	21,97	0,00	0,00	4,41	0,00	17,30		0,00	0,03
13	30	300	4374	15,45	0,00	0,01	11,54	0,00	4,80		0,00	0,00
14	60	300	2968	18,86	0,00	0,06	21,60	0,00	8,70	0,03	0,00	0,01
15	100	300	1729	17,02	0,06	0,26	24,62	0,00	7,94	0,00	0,23	0,39
16	5	400	8162	133,77	0,00	0,00	6,34	0,00	24,55		0,00	0,03
17	10	400	6999	81,94	0,00	0,05	8,68	0,00	27,89		0,00	0,01
18	40	400	4809	57,42	0,00	0,04	28,92	0,00	6,55	0,00	0,00	0,00
19	80	400	2845	51,03	0,00	0,13	49,67	0,00	9,30	0,04	0,04	0,12
20	133	400	1789	47,19	0,00	0,11	61,31	0,00	24,50	0,00	0,17	0,22
21	5	500	9138	109,27	0,00	0,00	9,33	0,00	3,70		0,00	0,09
22	10	500	8579	3023,91	0,00	0,20	16,34	0,00	55,86	0,00	0,00	0,00
23	50	500	4619	105,33	0,00	0,02	73,30	0,00	8,64		0,00	0,05
24	100	500	2961	106,20	0,00	0,09	103,72	0,00	41,42	0,00	0,03	0,10
25	167	500	1828	106,41	0,05	0,39	123,18	0,00	72,44	0,00	0,22	0,27
26	5	600	9917	9243,08	0,00	0,00	14,58	0,00	22,25		0,00	0,01
27	10	600	8307	310,69	0,00	0,00	25,13	0,00	12,53		0,00	0,04
28	60	600	4498	181,13	0,00	0,05	137,54	0,00	12,30	0,00	0,02	0,02
29	120	600	3033	164,30	0,03	0,09	215,48	0,00	18,81	0,00	0,07	0,13
30	200	600	1989	162,77	0,25	0,60	235,73	0,00	57,55	0,15	0,40	0,48
31	5	700	10086	639,58	0,00	0,00	19,93	0,00	29,00		0,00	0,03
32	10	700	9297	15721,33	0,00	0,00	32,93	0,00	15,41	0,00	0,00	0,02
33	70	700	4700	264,66	0,00	0,08	194,94	0,00	19,88	0,00	0,00	0,04
34	140	700	3013	239,06	0,00	0,14	339,65	0,00	33,02	0,00	0,07	0,09
35	5	800	10400	31208,52	0,00	0,00	26,51	0,00	47,64		0,00	0,03
36	10	800	9934	35782,86	0,00	0,00	43,22	0,00	767,16		0,00	0,00
37	80	800	5057	352,31	0,00	0,04	310,32	0,00	97,06	0,00	0,02	0,05
38	5	900	11060	61317,58	0,00	0,00	35,98	0,00	107,78		0,00	0,09
39	10	900	9423	3595,95	0,00	0,00	50,21	0,00	136,27		0,00	0,00
40	90	900	5128	518,69	0,04	0,09	474,66	19,50	32,89	0,04	0,10	0,10

## 6 Conclusões e desenvolvimentos futuros

Neste trabalho, foi apresentada uma abordagem baseada na metaheurística *Chemical Reaction Optimization* para resolver o problema da localização de estações de serviço, baseado no modelo clássico da p-mediana. Uma aplicação escrita em linguagem C# [ECMA 2006] foi criada para implementar a solução proposta. Para validar os resultados, a biblioteca *OR-Library* contendo 40 problemas de p-mediana foi utilizada e os resultados comparados com os valores ótimos da literatura, e também com aqueles obtidos por Relaxação Lagrangiana, bem como metaheurísticas empregadas com sucesso neste tipo de problema, como a *Variable Neighborhood Search* e o Algoritmo Genético. Como parte da adaptação do CRO ao problema da p-mediana, um algoritmo de busca local de grande eficiência, denominado de *Fast Interchange*, foi utilizado nas fases de intensificação das soluções. Os algoritmos *DPX* e *Half-total change* foram utilizados na fase de síntese e diversificação, respectivamente.



Os resultados apresentados indicam que a metaheurística CRO também é aplicável ao problema da p-mediana, uma vez que o algoritmo foi capaz de chegar a soluções de alta qualidade em curto espaço de tempo. A abordagem proposta parece ser competitiva em relação aos métodos avaliados, embora não se possa afirmar que seja o melhor método para a solução do problema.

Existem várias possibilidades de pesquisas e desenvolvimentos futuros. Em primeiro lugar, poder-se-ia buscar uma melhoria nos tempos de execução, sem que houvesse perda notável de qualidade das soluções, testando-se diferentes configurações de populações e parâmetros de execução do CRO, específicas para os problemas a serem resolvidos, ao invés de utilizar-se a mesma configuração para todos os problemas. Em segundo lugar, a presente solução poderia ser modificada para lidar com problemas do tipo p-mediana capacitada, além de outras variações do mesmo problema de otimização.

## 7 Referências

- Al-Khedhairi, A. (2008). Simulated annealing metaheuristic for solving p-median problem. *Int. J. Contemp. Math. Sciences*, v. 3, n. 28, p. 1357–1365.
- Alp, O., Erkut, E. e Drezner, Z. V. I. (2003). An Efficient Genetic Algorithm for the p-Median. p. 21–42.
- Beasley, J. E. (1990). OR-LIBRARY. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>, [accessed on Fev. 25].
- Church, R. e ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association*, v. 32, n. 1, p. 101–118.
- Daskin, M. S. e Maass, K. L. (2015). The p-median problem. *Location Science*. Springer. p. 21–45.
- ECMA (2006). 334. *ISO/IEC*, v. 23270.
- Galvão, R. D. e Raggi, L. A. (1989). A method for solving to optimality uncapacitated location problems. *Annals of Operations Research*, v. 18, n. 1, p. 225–244.
- Hakimi, S. L. (1964). Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*, v. 12, n. 3, p. 450–459.
- Hansen, P. e Mladenović, N. (1997). Variable neighborhood search for the p-median. *Location Science*, v. 5, n. 4, p. 207–226.
- IBM (2017). IBM CPLEX Optimizer - United States. <https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>, [accessed on Fev. 16].
- Kariv, O. e Hakimi, S. L. (1979). An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, v. 37, n. 3, p. 539–560.
- Lam, A. Y. S. e Li, V. O. K. (2012). Chemical Reaction Optimization: A tutorial. *Memetic Computing*, v. 4, n. 1, p. 3–17.
- Merz, P. e Freisleben, B. (1997). A genetic local search approach to the quadratic assignment problem. In *7th international conference on genetic algorithms*.
- Rolland, E., Schilling, D. A. e Current, J. R. (jan 1997). An efficient tabu search procedure for the p-Median Problem. *European Journal of Operational Research*, v. 96, n. 2, p. 329–342.
- Rosing, K. E. e ReVelle, C. S. (1997). Heuristic concentration: Two stage solution construction. *European Journal of Operational Research*, v. 97, n. 1, p. 75–86.
- Toregas, C., Swain, R., ReVelle, C. e Bergman, L. (1971). The Location of Emergency Service Facilities. *Operations Research*, v. 19, n. 6, p. 1363–1373.
- Whitaker, R. A. (1983). A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *INFOR: Information Systems and Operational Research*, v. 21, n. 2, p. 95–108.