

ALGORITMO GENÉTICO PARA UM PROBLEMA DE LOCALIZAÇÃO/DISTRIBUIÇÃO MULTINÍVEL

Diogo Fernandes

Universidade Federal do Rio Grande do Norte
Campus Universitário s/n, Natal-RN, 59072-970, Brazil.
diogorobson@hotmail.com

Caroline Rocha

Universidade Federal do Rio Grande do Norte
Campus Universitário s/n, Natal-RN, 59072-970, Brazil
caroline.rocha@ect.ufrn.br

Daniel Aloise

Universidade Federal do Rio Grande do Norte
Campus Universitário s/n, Natal-RN, 59072-970, Brazil
aloise@dca.ufrn.br

Glaydston Ribeiro

Universidade Federal do Espírito Santo
Rodovia BR 101 Norte, Km 60, São Mateus-ES, 29932-540, Brazil
glaydstonribeiro@ceunes.ufes.br

Enilson Santos

Universidade Federal do Rio Grande do Norte
Campus Universitário s/n, Natal-RN, 59072-970, Brazil
emsantos@ct.ufrn.br

Allyson Silva

Universidade Federal do Rio Grande do Norte
Campus Universitário s/n, Natal-RN, 59072-970, Brazil
allysonfcs@hotmail.com

RESUMO

A área de transportes tem focado no estudo de modelos crescentes de complexidade de distribuição. Sistemas de transporte multinível surgem na presença de entidades intermediárias, chamadas satélites, entre pontos de demanda e oferta. Embora uma análise de decisão integrada seja mais eficiente, tais sistemas geralmente são resolvidos para cada nível separadamente. Neste trabalho, tratamos de um problema de distribuição em que também decidimos a localização dos pontos de oferta e satélites dentre um conjunto de pontos potenciais. O objetivo é minimizar os custos de operação desta distribuição-localização multinível enquanto respeitando restrições de demanda e capacidade em todos os níveis de transporte. Para tanto, propomos um Algoritmo Genético e apresentamos resultados computacionais comparando nosso método com o CPLEX 12. Os resultados mostram que nossa abordagem frequentemente supera o CPLEX quando ambos os algoritmos estão limitados a executar por um horizonte de tempo moderado, i.e. aproximadamente 10 minutos.

PALAVRAS CHAVE. Localização, Distribuição, Metaheurística.

Área principal Logística e Transporte

ABSTRACT

Transportation works has focused on studying more complex distribution systems. Multi-echelon transportation systems arise in the presence of intermediate transportation agents, called satellites, between demand and supply points. Although an integrated decision is more efficient, such systems are usually analysed each echelon at a time. In this work, we tackle a distribution problem where a finite set of potential distribution agents must be also located. The objective is to minimize the operation costs of the multi-echelon transportation system thereby satisfying demand and capacity constraints of its agents. For this purpose, a Genetic Algorithm is proposed and computational results are reported comparing the heuristic results with those obtained by CPLEX 12. The results show that our approach often outperforms CPLEX when both algorithms are limited to execute for moderate time horizon, i.e., approximately 10 minutes.

KEYWORDS. Location, Distribution, Metaheuristics

Main area Logistics and Transport

1. Introdução

Com a globalização dos mercados e a criação das economias de escala, as empresas têm investido no desenvolvimento de sistemas produtivos e logísticos mais eficientes para produção e distribuição de bens. Os estudos na área de transporte têm se concentrado em modelos com níveis crescentes de complexidade de distribuição (ex. Hesse, 2002; Wanga et al., 2011). Essa tendência é explicada pelo próprio interesse dos agentes desses sistemas (como atacadistas e consumidores finais) por modelos que reflitam melhor sua realidade. Neste contexto, há uma grande demanda por ferramentas de Pesquisa Operacional que permitam auxiliar o decisor/gestor da área de transportes.

Entre vários aspectos a serem estudados no planejamento do sistema logístico empresarial, dois deles são abordados neste artigo: a localização das unidades operacionais e como é realizada a distribuição na cadeia de suprimentos para clientes conhecidos. Para determinar a localização das unidades operacionais é necessário realizar um levantamento acerca das potenciais localidades nas quais as unidades operacionais irão se instalar. Sendo assim, o estudo detalhado de localização é importante não só a nível estratégico, mas também para a própria viabilidade do negócio, à medida que minimiza o impacto da tomada de decisões erradas que acarretam em custos desnecessários de instalação.

O estudo de localização não é restrito ao início das operações de determinada organização. Mesmo com uma rede de suprimentos já definida e em funcionamento, modificações no mercado podem exigir a instalação de novas unidades. Para Slack et al. (2007), essa exigência pode decorrer a partir de alterações na demanda de bens e/ou na oferta de insumos para a operação. Assim, mesmo com uma estrutura já consolidada, novas instalações podem ser necessárias e os estudos acerca dessas novas instalações devem manter – ou buscar – o equilíbrio entre os custos já existentes e os novos, assim como o nível de serviço oferecido ao cliente e os resultados financeiros da empresa. Além dos custos de instalação e dos custos de operação de cada unidade, a escolha de uma nova localização deve considerar os custos de transporte dos produtos entre as unidades operacionais.

O presente estudo tem como um de seus objetivos otimizar esse processo decisório de seleção dos locais de instalação das facilidades, bem como configurar a rede de distribuição a fim de reduzir os custos de toda a cadeia. O termo “facilidade” aqui abrange tanto as plantas de produção quanto os centros ou satélites de distribuição.

O segundo objetivo deste estudo trata da otimização da distribuição de produtos, ou seja, trata da entrega de produtos originados nas plantas com destino a clientes com demandas conhecidas a priori. Para distribuir seus produtos é importante que uma organização planeje e configure a rede de distribuição através da qual tais produtos serão transportados até os clientes. O sistema de transporte mais simples consiste em entregas de pontos de suprimentos a pontos de demanda, com os pontos de suprimentos limitados em capacidade de produção/armazenagem, além de custos de distribuição associados aos possíveis caminhos no sistema. Do ponto de vista logístico, é comum a adoção de sistemas multiníveis para a distribuição de produtos. Sistemas multiníveis são caracterizados por apresentar um (ou mais) intermediário(s) no processo de distribuição de produtos desde a fábrica até seus clientes. Tais sistemas são apresentados em diversos contextos, veja Gendron e Semet (2009), Crainic et al. (2009), Du et al. (2007), Guyonnet et al. (2009), Golden et al. (2002), Lee et al. (2007), e Van Roy (1989).

O uso de intermediários (satélites) na rede de suprimentos evita que uma fábrica tenha que atender clientes diretamente, economizando tempo para a gestão de sua cadeia produtiva. Sistemas multinível são tradicionalmente resolvidos separadamente para cada um dos níveis de transporte, o que pode, em algumas situações, levar a estratégias inviáveis de operação ou menos lucrativas (Guyonnet et al., 2009). Uma análise decisória integrada é mais eficiente caso exista uma interdependência entre o transporte feito entre cada um dos níveis do sistema. Importante frisar que para alguns destes problemas, mesmo a obtenção de uma solução viável é uma tarefa difícil dado o grande número de requisitos envolvidos. Infelizmente, embora amplamente aplicáveis, sistemas multinível de transporte carecem atualmente de ferramentas e estudos de

análise e gestão (Perboli et al., 2011).

Diante da importância destas recentes pesquisas e da demanda empresarial por soluções logísticas, este trabalho apresenta um Algoritmo Genético capaz de localizar facilidades a fim de compor uma rede de distribuição multinível ótima (ou próxima da ótima). Assim, além desta seção introdutória, a Seção 2 apresenta um modelo de programação inteira mista para o problema em questão. Na Seção 3 é apresentado o Algoritmo Genético proposto neste trabalho é com seus componentes. A Seção 4 apresenta uma análise dos resultados computacionais do algoritmo proposto para as instâncias utilizadas, baseada em comparações com o *solver* CPLEX 12.0. E por último, algumas conclusões são apresentadas na Seção 5.

2. Formulação Matemática do Problema

Um esquema de como se configura o sistema multinível de transporte de carga objetivo deste artigo é ilustrado na Figura 1, onde as plantas (conjunto I) se relacionam com os satélites (conjunto J) em um primeiro nível, que por sua vez estão relacionados aos clientes (conjunto K) no segundo nível.

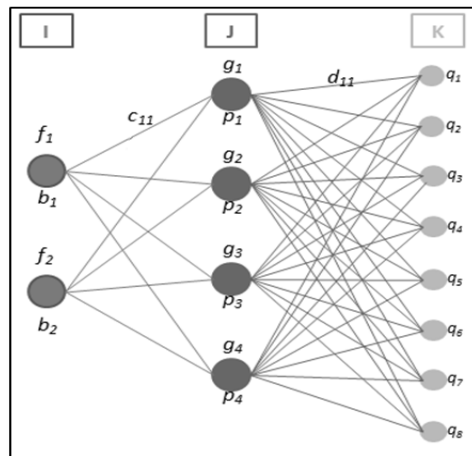


Figura 1. Ilustração de um sistema multinível de transporte com 2 plantas, 4 satélites e 8 clientes.

Os custos fixos, relativos à abertura e manutenção de plantas e satélites, são identificados pelas variáveis f_i , $i \in I$, e g_j , $j \in J$, respectivamente. As capacidades de uma planta $i \in I$ e um satélite $j \in J$ são representadas pelas variáveis b_i e p_j , respectivamente. Cada rota entre uma planta $i \in I$ e um satélite $j \in J$ possui um custo unitário de transporte c_{ij} , enquanto que, no segundo nível, o custo de transporte entre um satélite $j \in J$ e cliente $k \in K$ é representado por d_{jk} . Além disso, cada cliente $k \in K$ demanda q_k unidades do produto.

O modelo matemático apresentado a seguir é similar ao proposto por Marín e Pelegrin (1999). Sejam x_{ij} e s_{jk} as variáveis indicando a quantidade de produtos transportados da planta $i \in I$ ao satélite $j \in J$ e do satélite $j \in J$ ao cliente $k \in K$, respectivamente. Seja y_i a variável binária que recebe 1 se, e somente se, a planta $i \in I$ é selecionada para ser aberta; 0, caso contrário. Da mesma forma, seja z_j a variável que indica se o satélite $j \in J$ será aberto ($z_j = 1$) ou fechado ($z_j = 0$). O modelo matemático pode, então, ser descrito como:

$$\min \sum_{i \in I} f_i y_i + \sum_{j \in J} g_j z_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} d_{jk} s_{jk} \quad (1)$$

sujeito a:

$$\sum_{j \in J} s_{jk} = q_k \quad \forall k \in K \quad (2)$$

$$\sum_{i \in I} x_{ij} = \sum_{k \in K} s_{jk} \quad \forall j \in J \quad (3)$$

$$\sum_{j \in J} x_{ij} \leq b_i y_i \quad \forall i \in I \quad (4)$$

$$\sum_{k \in K} s_{jk} \leq p_j z_j \quad \forall j \in J \quad (5)$$

$$x_{ij} \in \mathbb{R}^+ \quad \forall i \in I, j \in J \quad (6)$$

$$s_{jk} \in \mathbb{R}^+ \quad \forall j \in J, k \in K \quad (7)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (8)$$

$$z_j \in \{0,1\} \quad \forall j \in J. \quad (9)$$

O modelo acima tem como função objetivo (1) minimizar o custo total da rede de distribuição, que é dado pelo somatório dos custos fixos das plantas e satélites selecionados, adicionado dos custos de transporte entre as plantas e os satélites e dos custos de transporte entre os satélites e os clientes. O conjunto de restrições (2) garante que as demandas dos clientes sejam atendidas. As restrições (3) obrigam que a quantidade de produto enviada aos satélites a partir do primeiro nível seja igual ao que sairá deles no segundo nível. Os conjuntos de restrições (4) e (5) garantem que a quantidade de produto enviada das plantas e satélites, respectivamente, respeitam a capacidade das instalações escolhidas. Os conjuntos de restrições (6)-(9) definem o domínio das variáveis de decisão.

Importante destacar que se as variáveis y e z forem fixadas é possível obter os valores das variáveis de transporte x e s pela resolução de um problema de fluxo a custo mínimo. A estrutura de solução do Algoritmo Genético da Seção 3 utiliza este fato, sendo representada por um vetor binário de tamanho $|I| + |J|$, com componente igual a 1 caso a planta (ou satélite) correspondente seja utilizada, e igual a 0, caso contrário.

Alguns trabalhos da literatura tratam do problema de transporte multinível apresentando variações com relação ao modelo (1)-(9). Crainic et al. (2009) utilizam variáveis para representar o grau de acesso dos meios de transporte aos locais analisados para a instalação de um satélite no sistema multinível com roteamento de veículos. Tragantalerngsak et al. (2000) propõem um método exato para resolução exata de um modelo similar onde o carregamento dos satélites é proveniente de uma única planta e a demanda de cada cliente é atendida por um único satélite. Gendron e Semet (2009) realizam análises de formulações e relaxações na resolução de problemas de localização para sistemas multiníveis de distribuição, utilizando um *solver* comercial. No entanto, o modelo tratado no referido artigo é diferente do modelo (1)-(9) apresentado, pois considera o número e a capacidade volumétrica dos veículos em cada nível, além de considerar o número e o volume de produtos, entre outras considerações. Keskin e Uster (2007) utilizaram uma heurística *scatter search* com *path relinking* para resolver um problema de transbordo, em que apenas a localização dos satélites era definida, além da simplificação utilizada ao fixar o número de satélites possíveis de serem abertos. Marín e Pelegrin (1999) e Livintchev et al. (2010) apresentam relaxações Lagrangianas para o modelo (1)-(9) obtendo limitantes inferiores.

3. Algoritmos Desenvolvidos

Este trabalho propõe uma série de heurísticas para o problema apresentado na Seção 2.

A coordenação destas heurísticas é realizada através de uma abordagem evolutiva, por meio de um Algoritmo Genético, apresentado na Seção 3.3.

3.1. Heurísticas construtivas

As heurísticas construtivas são estruturas programáveis que devem apresentar estratégias para construir uma solução viável para o problema. Neste trabalho foram desenvolvidas duas heurísticas construtivas para obtenção de soluções iniciais.

```

1 Algoritmo HC1
2    $y_i \leftarrow 0, \forall i \in I$ 
3    $z_j \leftarrow 0, \forall j \in J$ 
4   para  $\forall i \in I$  faça
5      $CBP_i \leftarrow \frac{\sum_{j \in J} c_{ij} + f_i}{b_i}$ 
6   fim-para
7   enquanto  $\sum_{i \in I} b_i y_i \leq \sum_{k \in K} q_k$ 
8     Selecione aleatoriamente um índice  $i'$  com probabilidade dada por
9      $\frac{1}{CBP_{i'}} / \frac{1}{\sum_{i: y_i=0} CBP_i}$ 
10     $y_{i'} \leftarrow 1$ 
11   fim-enquanto
12   para  $\forall j \in J$  faça
13      $CBS_j \leftarrow \frac{\sum_{k \in K} d_{jk} + g_j + \sum_{i \in I} c_{ij} y_i}{p_j}$ 
14   fim-para
15   enquanto  $\sum_{j \in J} p_j z_j \leq \sum_{k \in K} q_k$ 
16     Selecione aleatoriamente um índice  $j'$  com probabilidade dada por
17      $\frac{1}{CBS_{j'}} / \frac{1}{\sum_{j: z_j=0} CBS_j}$ 
18     $z_{j'} \leftarrow 1$ 
19   fim-enquanto
20   retorne  $(y, z)$ 

```

Figura 1. Pseudocódigo da HCl

A primeira das heurísticas construtivas propostas adota, como estratégia de construção, o critério de custo-benefício para as decisões de selecionar plantas e/ou satélites. O custo-

benefício de abertura de uma planta $i \in I$ é calculado pela razão $\frac{f_i + \sum_{j \in J} c_{ij}}{b_i}$, enquanto o custo-

benefício de abertura de um satélite $j \in J$ é calculado como $\frac{\sum_{i \in I} c_{ij} y_j + g_j + \sum_{k \in K} d_{jk}}{p_j}$ (note

que o custo-benefício dos satélites é dado em função das plantas abertas). A Figura 2 apresenta o pseudocódigo desta heurística, denominada HCl.

De acordo com a Figura 2, pode-se perceber que a mesma está dividida em dois momentos: a seleção de plantas e a seleção de satélites. O algoritmo começa com todas as plantas e satélites fechados (linhas 2 e 3). No laço das linhas 4-6, são calculadas as razões de custo-

benefício de todas as plantas, denotadas CBP_i , para $i = 1, \dots, |I|$. Em seguida, no laço das linhas 7-10, são abertas as plantas até que seja atendida a demanda dos clientes. Para a abertura das plantas é realizado o cálculo da probabilidade de abertura de cada planta; e uma delas é selecionada aleatoriamente para abertura (linhas 8 e 9). Entre as linhas 11-17 o processo acima é repetido para a seleção dos satélites. No laço entre as linhas 14 e 17 é calculada, a partir do custo-benefício de abertura dos satélites (vetor CBS), a probabilidade de abertura de cada satélite $j = 1, \dots, |J|$. Em seguida, um dos satélites é escolhido aleatoriamente para abertura na linha 16. O procedimento é repetido até que toda a demanda possa ser atendida. Finalmente, as plantas e satélites abertos pela heurística $HC1$ são retornados na linha 18.

Ao final, a solução do algoritmo é apresentada em forma de vetor binário, com componente igual a 1 caso a facilidade correspondente seja aberta, e 0, caso contrário. A segunda heurística construtiva, denominada $HC2$, parte da solução da relaxação de programação linear (PL) do modelo (1)-(9) e fixa, de maneira iterativa, em zero as variáveis binárias com valores próximos a 0, e em um as variáveis binárias com valores próximos a 1 (uma tolerância $\varepsilon = 10^{-6}$ foi utilizada). Inicialmente, a PL do modelo (1)-(9) é resolvida, e a variável binária com valor mais próximo de 1 representando uma planta, é então fixada efetivamente em um (forçando a abertura da respectiva planta). Repete-se este procedimento para garantir a abertura de um satélite. Com a fixação de duas variáveis binárias (abertura de uma planta e de um satélite), o problema relaxado é novamente resolvido, e o processo é então repetido até que a capacidade total das plantas abertas, assim como a capacidade total dos satélites abertos, seja superior à demanda total dos clientes.

3.2. Heurísticas de busca

Soluções geradas por heurísticas construtivas não são necessariamente ótimas, podendo ser frequentemente melhoradas por heurísticas de busca local. A seguir, são então descritas duas buscas locais desenvolvidas para este trabalho.

A busca local 1 (BL1) realiza movimentos isolados de abertura e fechamento de facilidades (plantas e satélites). Esses movimentos são feitos tomando-se uma facilidade, seja planta ou satélite, e complementando seu estado atual. Desta forma, uma planta/satélite fechada é aberta, e vice-versa. A vizinhança de uma solução engloba todos os vizinhos obtidos pela complementação de estados das facilidades do problema.

Para se obter o custo total da solução vizinha, incluindo os custos de transporte, é utilizado um algoritmo de fluxo a custo mínimo, proposto por Goldberg (1997) para o conjunto de plantas e satélites abertos (o algoritmo não é executado para uma solução inviável). Caso uma solução vizinha de custo inferior ao da solução atual seja encontrada, ela passa a ser a nova solução corrente. Os movimentos de busca em vizinhança são realizados iterativamente para cada planta e cada satélite da solução atual até que seja impossível encontrar-se uma solução vizinha aprimorante.

A segunda busca local (BL2) é executada após a BL1. O uso de duas buscas locais em sequência visa explorar o espaço de busca de forma abrangente (Hansen e Mladenovic, 2010). A BL2, assim como a BL1, realiza, basicamente, movimentos de abertura e fechamento de facilidades. A estratégia difere por trabalhar com pares complementares de facilidades. Entende-se por pares complementares aqueles que englobem uma facilidade aberta (*bit* 1) e outra fechada (*bit* 0).

A BL2 inicia com a busca por pares complementares, após encontrar um par complementar seus valores são complementados, abrindo-se uma facilidade que estava fechada e fechando a outra facilidade que estava aberta. Em seguida, é executado o algoritmo de fluxo a custo mínimo para obter o custo total da nova solução, a qual será comparada com a solução atual. Caso a nova solução seja menor, esta é considerada a nova solução corrente. A busca local termina quando a avaliação de todos os pares complementares não resultarem em um vizinho aprimorante.

3.3 Algoritmo Genético

De acordo com Ribeiro et al. (2011), as metaheurísticas são estruturas que coordenam os procedimentos e normas heurísticas para encontrar melhores soluções aproximadas para problemas de otimização combinatória de difícil resolução computacional. Já para Figueiredo (2007), a principal diferença entre heurísticas e metaheurísticas está na flexibilidade de aplicação entre ambas. As heurísticas são projetadas especificamente para uma determinada situação, ao passo que, as metaheurísticas podem ser utilizadas em diversas situações controlando o funcionamento de uma heurística específica.

Conforme visto, as duas heurísticas de busca local desenvolvidas na Seção 3.2 dependem diretamente da resolução de problemas de fluxo a custo mínimo. Embora de tempo polinomial, o algoritmo desenvolvido por Goldberg (1997) não calcula o custo de uma solução em tempo constante ou mesmo linear. Consequentemente, a decisão sobre qual metaheurística utilizar para coordenação das heurísticas propostas torna-se bastante relevante.

O Algoritmo Genético (AG) trabalha com um conjunto de soluções denominado *população*, e cada solução participante dessa população é denominada *indivíduo*, sendo representada por um *cromossomo*. Segundo Reeves (2003), a função do AG é recombinar os cromossomos a partir de procedimentos análogos à genética como o cruzamento e a mutação. Na Figura 3 está descrito o pseudocódigo padrão de um AG (Reeves, 2003) desde a seleção de uma população inicial de cromossomos (linha 1) até a seleção da nova população (linha 11) com o uso de seus procedimentos de cruzamento (linhas 3-6) e de mutação (linhas 7-9), atendendo as restrições de viabilidade do problema (linha 10).

Algoritmo Genético Padrão

```

1   População Inicial de Cromossomos;
2   enquanto condição de parada não satisfeita faça
3       se condição de cruzamento então
4           Selecciona os cromossomos pais;
5           Condiciona a parâmetros de cruzamento;
6           Cruzamento;
7       se condição de mutação então
8           Selecciona pontos de mutação;
9           Mutação;
10          Valida modificações;
11          Selecciona nova população;
12  fim-enquanto

```

Figura 3. Pseudocódigo de um algoritmo genético padrão

A componente estocástica dos algoritmos genético reside no mecanismo de seleção dos pais para cruzamento (linha 4) e na probabilidade de mutação (linha 7).

O tamanho da população utilizada em um algoritmo genético é um fator importante que tem influência direta na eficiência e efetividade do algoritmo. Com uma população pequena, existem poucas chances de se obter um indivíduo de boa qualidade (solução próxima do ótimo), não sendo, portanto, efetivo. Por outro lado, se a população considera um grande número de indivíduos, aumenta-se a probabilidade de obter uma solução com qualidade durante o processo evolutivo do algoritmo, mas o tempo computacional do método aumenta, tornando-o ineficiente (Reeves, 2003).

O AG aqui proposto considera, durante o seu processo evolutivo, um processo de intensificação. Quando uma nova população passa ser considerada para cruzamentos e mutações, verifica-se se o melhor indivíduo dessa população é melhor do que o indivíduo mais adaptado encontrado até o momento entre todas as populações anteriores. Em caso afirmativo, as heurísticas de busca local apresentadas na Seção 3.2 são aplicadas sequencialmente com o

objetivo de melhorar esse indivíduo elite.

A seguir é apresentada a codificação do cromossomo e os operadores de cruzamento e de mutação considerados.

3.3.1 Codificação do cromossomo

A Figura 4 exibe a representação do cromossomo utilizado. De maneira específica, todas as plantas são inicialmente representadas, seguidas de todos os satélites. Esta representação de assemelha a vetor com elementos (genes) binários. Genes iguais a um, indicam que plantas ou satélites estão abertos.

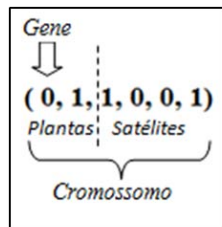


Figura 4. Estrutura de solução / Codificação do cromossomo

Cada indivíduo da população é avaliado pelo *fitness* de seu cromossomo, o que é equivalente ao custo da solução conforme a função objetivo (1).

3.3.1 População inicial

Um AG utiliza uma população inicial para então aplicar os operadores de cruzamento e mutação, durante o processo evolutivo. Neste trabalho, optou-se por utilizar as heurísticas *HC1* e *HC2* para compor esta população inicial. A heurística *HC2* foi utilizada uma única vez devido ao seu tempo computacional, enquanto a *HC1* é utilizada para compor o restante da população. De uma forma geral, o indivíduo construído por *HC2* apresenta melhor *fitness* do que os indivíduos gerados por *HC1*, servindo de guia no processo de busca no espaço de soluções.

3.3.2 Operadores de cruzamento e de mutação

Considere uma população com P indivíduos. O operador de cruzamento seleciona aleatoriamente dois indivíduos e realiza uma mistura genética, produzindo dois filhos. O gene dos pais é transmitido com igual probabilidade (50%) aos filhos. Este procedimento diversifica a população, pois os dois filhos gerados substituem os pais na geração futura.

O operador de mutação é aplicado aos filhos gerados. Um gene tem probabilidade igual a 3,00% de sofrer mutação. A mutação indica a troca do valor genético entre zero e um. Este procedimento também ajuda a diversificar a população.

4. Resultados Computacionais

Os testes computacionais consideraram seis classes de problemas com cinco instâncias cada, totalizando 30 instâncias. A Tabela 1 apresenta, para cada classe de problema, os intervalos de valores utilizados na geração das instâncias, onde $B = \sum_{k \in K} q_k / |I|$ e $P = \sum_{k \in K} q_k / |J|$. O valor de cada parâmetro é obtido a partir de uma distribuição uniforme no intervalo correspondente. Todas as instâncias geradas possuem 50 plantas, 100 satélites e 200 clientes.

Tabela 1. Intervalos de parâmetros para geração de instâncias

Parâmetros	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
------------	----------	----------	----------	----------	----------	----------

b_i	[2B 5B]	[15B 25B]	[5B 10B]	[15B 25B]	[5B 10B]	[5B 10B]
f_i	[20000 30000]	[20000 30000]	[20000 30000]	[20000 30000]	[20000 30000]	[20000 30000]
c_{ij}	[35 45]	[35 45]	[35 45]	[35 45]	[50 100]	[35 45]
p_i	[2P 5P]	[15P 25P]	[5P 10P]	[15P 25P]	[5P 10P]	[5P 10P]
g_j	[8000 12000]	[8000 12000]	[8000 12000]	[8000 12000]	[8000 12000]	[8000 12000]
d_{ik}	[55 65]	[55 65]	[55 65]	[800 1000]	[50 100]	[800 1000]
q_k	[10 20]	[10 20]	[10 20]	[10 20]	[10 20]	[10 20]

Tabela 2. Resultados médios do AG e do solver CPLEX nas 30 instâncias geradas

Classe	Instância	AG	tempo _{AG}	CPLEX	tempo _{cpex}	GAP
1	1	1485211	1256,11	1477398	4817,16	0,53%
	2	1478191	1245,12	1464441	618,86	0,93%
	3	1497213	1315,83	1494399	4471,84	0,19%
	4	1466731	1243,9	1462309	4778,32	0,30%
	5	1502200	1042,68	1492462	866,81	0,65%
2	1	725039,8	449,77	721718	3961,08	0,46%
	2	725034,8	441,35	721293	7634,7	0,52%
	3	717274,6	432,27	716443	3360,56	0,12%
	4	727676	482,53	725412	16325,05	0,31%
	5	721656	481,53	718536	8471,34	0,43%
3	1	984952,8	555,27	973482	4559,45	1,16%
	2	975119,6	554,58	968617	2349,54	0,67%
	3	997462,8	618,51	978690	6509,59	1,88%
	4	982250	531,98	973559	70191,6*	0,88%
	5	978946	552,92	975602	4251,74	0,34%
4	1	5299738,6	465,23	5299973	37123,1*	0,00%
	2	5280308	467,06	5279599	84569,03*	0,01%
	3	5228131,4	472,76	5227517	72888,28*	0,01%
	4	5317938	439,63	5320811	70554,11*	-0,05%
	5	5253489	472,15	5251871	118384,97*	0,03%
5	1	1060799	483,63	1058360	12374,4	0,23%
	2	1053325	484,18	1050553	51553,18	0,26%
	3	1071790	500,93	1057271	105701,01	1,35%
	4	1054638	474,99	1052324	84407,1	0,22%
	5	1060621	467,8	1059397	7018,5	0,12%
6	1	5513397	565,27	5506970	92689,58*	0,12%
	2	5488644	611,7	5482791	84964,27*	0,11%
	3	5466694	592,21	5446763	66923,16*	0,36%
	4	5529502,8	637,65	5517165	70300,67*	0,22%
	5	5468482	550,89	5463695	54553,18*	0,09%

As instâncias são resolvidas utilizando o AG da Seção 3, implementado em C++ e compilado pelo g++ 4.2. O solver CPLEX 12.0 foi utilizado nas comparações. Os algoritmos foram executados em um Genuine Intel 6 com 2 GB de memória RAM e 2.3 GHz.

O AG foi executado com os seguintes parâmetros: (i) tamanho da população = 100; (ii) número de gerações = 400 (utilizado como condição de parada do algoritmo); (iii) taxa de mutação = 3% . Além disso, as heurísticas de busca local da Seção 3.2 são aplicadas em sequência a cada 50 iterações sob as cinco melhores soluções da população.

Tabela 3. Resultados médios do AG para obtenção de sua melhor solução nas das 10 execuções efetuadas e do *solver* CPLEX abortado no tempo médio de execução do AG

Classe	Instância	AG	tempo _{melhorAG}	CPLEX _{limitado}	GAP
1	1	1485211	19,3	1494933	-0,65%
	2	1478191	189,75	1476778	0,10%
	3	1497213	197,61	1512828	-1,04%
	4	1466731	20,48	1466731	0,00%
	5	1502200	17,54	1499009	0,21%
2	1	725039,8	414	735404	-1,43%
	2	725034,8	420,22	727379	-0,32%
	3	717274,6	308,82	724032	-0,94%
	4	727676	409,17	734047	-0,88%
	5	721656	425,74	739901	-2,53%
3	1	984952,8	474,06	973556	1,16%
	2	975119,6	205,52	971742	0,35%
	3	997462,8	238,25	986188	1,13%
	4	982250	18,97	981267	0,10%
	5	978946	19,36	1028756	-5,09%
4	1	5299738,6	325,73	5323572	-0,45%
	2	5280308	410,69	5332402	-0,99%
	3	5228131,4	325,9	5269097	-0,78%
	4	5317938	414,71	5364659	-0,88%
	5	5253489	299,32	6383828	-21,52%
5	1	1060799	20,45	1316534	-24,11%
	2	1053325	85,76	1059422	-0,58%
	3	1071790	18,48	1258629	-17,43%
	4	1054638	17,91	1272402	-20,65%
	5	1060621	17,26	1310700	-23,58%
6	1	5513397	90,28	5521581	-0,15%
	2	5488644	24,65	6096633	-11,08%
	3	5466694	337,05	5829176	-6,63%
	4	5529502,8	146,42	6041048	-9,25%
	5	5468482	92,15	5995891	-9,64%

A Tabela 2 apresenta os resultados médios das execuções do AG para dez sementes distintas. A primeira coluna se refere à classe de problemas e a segunda à instância particular (1 a 5). A terceira coluna apresenta o valor médio do melhor indivíduo (solução) obtido pelo AG em cada instância e a quarta apresenta o tempo médio de execução do algoritmo. A quinta e sexta colunas se referem aos valores de solução ótima e aos tempos de execução, respectivamente, referentes à utilização do CPLEX. Os valores marcados com asterisco na sexta coluna se referem ao tempo total de execução do CPLEX antes do mesmo ser abortado por falta de memória. Neste caso, a solução reportada na quinta coluna se refere a melhor solução obtida até então. Finalmente, a sétima coluna (GAP) se refere às diferenças relativas entre o CPLEX e os valores médios das soluções obtidas pelo AG.

Da Tabela 2, verifica-se que as diferenças relativas entre as soluções do AG e do CPLEX são sempre inferiores a 2%. Em particular, para uma instância da classe 4, o resultado médio do AG é melhor do que o do CPLEX antes de sua finalização. Além disso, nota-se que o tempo computacional médio utilizado pelo AG é, com exceção de uma única instância da classe 1, inferior ou muito inferior ao do CPLEX.

Com o objetivo de fornecer uma análise mais detalhada desta última observação, a

Tabela 3 apresenta em suas terceira e quarta colunas os valores médios das melhores soluções obtidas pelo AG em 10 execuções e os tempos médios de execução necessários para obtê-las (note que estes tempos são inferiores aos tempos médios totais de execução do AG apresentados na Tabela 2). A quinta coluna (CPLEX_{limitado}) apresenta o valor da melhor solução viável obtida pelo CPLEX, tendo como limite máximo de tempo de execução os tempos médios utilizados pelo AG para encontrar a sua melhor solução. Finalmente, a sexta coluna apresenta as diferenças relativas entre os valores obtidos na quarta e sexta colunas.

Os resultados apresentados na Tabela 3 mostram que frequentemente, para o tempo limite dado, os resultados do AG são superiores aos obtidos pelo CPLEX. Em quatro instâncias testadas, os resultados médios do AG foram mais do que 20% superiores. Isto demonstra que para aplicações onde o tempo é uma restrição importante para o processo de tomada de decisão, o AG proposto neste artigo é uma boa opção como ferramenta de suporte. Além disso, em aplicações onde a incerteza dos dados é uma característica inerente ao seu processo de aquisição, a aplicação do AG permite avaliar diferentes parâmetros/cenários do problema em um tempo computacional reduzido.

5. Conclusões

Este artigo apresentou um Algoritmo Genético (AG) para o problema de localização/distribuição multinível. O AG utiliza duas heurísticas construtivas (HC1 e HC2) para compor a sua população inicial. A HC1 possui aleatoriedade o que permite gerar soluções distintas para o problema, e a HC2 gera uma única solução a partir da relaxação de programação linear do modelo matemático do problema.

Os resultados computacionais mostraram que o AG proposto é capaz de gerar soluções de boa qualidade, principalmente quando o tempo computacional é uma variável importante no processo de solução.

Referências Bibliográficas

- Crainic, T.G., Mancini, S., Perboli, G. e Tadei, R.** (2010), Two-echelon vehicle routing problem: A satellite location analysis, *Procedia Social and Behavioral Sciences*, 2, 5944-5955.
- Du, T., Wang, F.K. e Lu, P.-Y.** (2007), A real-time vehicle-dispatching system for consolidating milk runs. *Transportation Research Part E*, 43, 565-577.
- Figueiredo, J.L.C.T.** (2007), *Optimização da localização de armazéns de redistribuição*. Dissertação de mestrado. Universidade Técnica de Lisboa. Lisboa, Portugal. 82 p.
- Gendron, B. e Semet, F.** (2009), Formulations and relaxations for a multi-echelon capacitated location–distribution problem. *Computers & Operations Research*, 36, 1335-1355.
- Goldberg, A.** (1997), An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm. *Journal of Algorithms*, 22, 1-29.
- Golden, B.L., Assad, A.A. e Wasil, E.A.**, Routing Vehicles in the Real World: Applications in the Solid Waste, Beverage, Food, Dairy and Newspaper Industries, em Toth, P e Vigo, D. (Eds.), *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 245-286, 2002.
- Guyonnet, P., Grant, F.K. e Bagajewicz, M.J.** (2009), Integrated model for refinery planning, oil procuring, and product distribution. *Industrial & Engineering Chemistry Research*, 48, 463-482.
- Hansen, P. e Mladenovic, N.** (2001), Variable Neighborhood Search: Principles and Applications. *European Journal of Operations Research*. 130, 449-467.
- Hesse, M.** (2002). Shipping News: the implications of electronic commerce for logistics and freight transport. *Resources, Conservation and Recycling*, 36, 211-240.
- ILOG CPLEX 12.1 User's Manual**. Disponível em <http://coin-or.blogspot.com.br/2011/04/ibm-ilog-cplex-121-user-manual.html>. 15/05/2012.

- Keskin, B.B. e Uster, H.** (2007), A scatter search-based heuristic to locate capacitated transshipment points, *Computers & Operations Research*, 34, 3112-3125.
- Lee, B.K., Kang, K.H. e Lee, Y.H.** (2008), Decomposition heuristic to minimize total cost in a multi-level supply chain network. *Computers & Industrial Engineering*, 54, 945-959.
- Litvinchev, I., Ozuna, E.L. e Mata, M.** (2010), Cotas lagrangianas para el problema de localización capacitado en dos etapas. *Anais do XXXIII CNMAC*, 3, 1114-1115.
- Marín, A. e Pelegrin, B.** (1999), Applying lagrangian relaxation to the resolution of two-stage location problems. *Annals of Operations Research*, 86, 179-198.
- Perboli, G., Tadei, R. e Vigo, D.** (2011), The two-echelon capacitated vehicle routing problem. *Transportation Science*, 45, 364-380.
- Reeves, C.**, Genetic Algorithms, em Glover, F. e Kochenberger, G.A. (Eds.), *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.
- Ribeiro, C. C., Rosseti, I. e Souza, R. C.** (2011), Effective probabilistic stopping rules for randomized metaheuristics: GRASP implementations. *LNCS 6683*. 146-160.
- Slack, N., Chambers, S. e Johnston R.** *Administração da Produção*. 2ª ed. Atlas, 2007.
- Tragantalerngsak, S., Holt, J. e Rönnqvist, M.** (2000), An exact method for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, 123, 473-489.
- van Roy, T.J.**, (1989), Multi-Level Production and Distribution Planning with Transportation Fleet Optimization, *Management Science*, 35, 1443-1453.
- Wanga, K.-J., Makond, B. e Liu, S.-Y.** (2011), Location and allocation decisions in a two-echelon supply chain with stochastic demand – A genetic-algorithm based solution. *Expert Systems with Applications*, 38, 6125-6131.