Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 2, N. 1, 2014.
Trabalho apresentado no CMAC-Sul, Curitiba-PR, 2014.

Algoritmo Evolução Diferencial Adaptado para o Problema das P-Medianas

Danielle Durski Figueiredo
Universidade Tecnológica Federal do Paraná – Departamento de Matemática
80230-901, Campus Curitiba, Curitiba, PR
E-mail: danidurski@gmail.com

Luzia Vidal de Souza, Aline Santos de Araújo,
Universidade Federal do Paraná – PPGMNE
81531-970, Campus III, Curitiba, PR
E-mails: luzia@ufpr.br, prof.alinearaujo@gmail.com

Resumo: *O estudo dos problemas de p-medianas se relaciona diretamente com problemas organizacionais da sociedade, como, por exemplo, a localização de escolas, postos de saúde, etc. Os algoritmos de Evolução Diferencial (ED) são poderosos algoritmos de otimização evolucionária, propostos inicialmente, para problemas em espaços contínuos. Recentemente, tem sido propostas adaptações ao seu mecanismo de mutação diferencial para aplicação em problemas combinatórios. Neste trabalho, uma proposta deste tipo de adaptação é apresentada para este algoritmo, para resolver o problema das p-medianas em um espaço de busca discreto. Testes computacionais foram realizados com instâncias geradas aleatoriamente e os resultados encontrados sugerem que a técnica proposta é promissora e apropriada para a resolução do problema das p-medianas.*

Palavras Chave*: Otimização, Evolução Diferencial Discreta, Problema das P-Medianas*.

1 Introdução

Os problemas de otimização classificados como *NP-hard* podem apresentar dificuldades quando são abordados com métodos exatos, especialmente nos casos de dimensões elevadas, uma vez que o tempo computacional necessário para a obtenção do valor ótimo global cresce exponencialmente à medida que os dados de entrada aumentam.

Os algoritmos de Evolução Diferencial (ED) são poderosos algoritmos de otimização evolucionária, propostos inicialmente na década de 1990, para a otimização de sistemas com variáveis contínuas. Recentemente, adaptações têm sido propostas ao seu mecanismo de mutação diferencial para otimização de problemas combinatórios.

Este trabalho tem como objetivo apresentar uma proposta de adaptação na operação de mutação diferencial em um algoritmo de Evolução Diferencial para aplicação em problemas de otimização discreta, especificamente problemas de *p*-medianas.

Este artigo é organizado como segue: na Seção 2 aborda-se o problema de localização de instalações, onde é apresentada a formulação matemática para o problema das *p*-medianas; na Seção 3 apresenta-se o algoritmo ED clássico no domínio contínuo; na Seção 4 a versão proposta neste trabalho para o ED no domínio discreto para aplicação no problema das *p*-medianas. Na Seção 5 encontram-se os resultados e as considerações finais.

2. Problema de localização de instalações

Algoritmos de otimização que abordam problemas de localização de instalações tratam do problema de identificar as melhores localizações, em uma área específica, para instalações de serviços. Segundo Minieka (1978) os problemas de localização dividem-se em dois tipos básicos: problemas de localização de centros, que buscam minimizar a maior distância a ser percorrida e

DOI: 10.5540/03.2014.002.01.0105

problemas de localização de medianas (*p*-medianas). O problema das *p*-medianas é tratado detalhadamente na seção seguinte.

2.1 Problema das p-medianas (PPM)

O problema das *p*-medianas é um problema clássico de otimização combinatória que tem como finalidade definir a melhor localização para *p* instalações (medianas) em uma região, minimizando a soma de todas as distâncias de cada ponto de demanda a sua mediana mais próxima. Métodos que abordam o PPM têm sido apresentados na literatura, podendo-se citar o estudo de Freitas *et al*. (2012) que se utilizou dos resultados da teoria espectral na construção de um algoritmo híbrido baseado no método de Teitz & Bart. O estudo do PPM tem grande importância, pois contribui diretamente em problemas da vida em sociedade como, por exemplo, a determinação da localização de pontos de coleta e transmissão de dados do processo eleitoral brasileiro [Correia *et al*., 2010].

Segundo Christofides (1975), o problema das *p*-medianas pode ser formulado como um problema de programação linear inteira binária, da seguinte forma:

$$Z = \min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \cdot x_{ij} \tag{1}$$

sujeito a:

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j \in N \tag{2}$$

$$\sum_{i=1}^{n} x_{ii} = p \tag{3}$$

$$x_{ij} \leq x_{ii} \qquad i, j \in N \tag{4}$$

$$x_{ij} \in \{0,1\} \quad i, j \in N \tag{5}$$

Onde: $[\, d_{ij} \,]_{n \times n}$ é uma matriz simétrica de distâncias ponderadas; $[\, x_{ij} \,]_{n \times n}$ é a matriz de alocação, com $x_{ij} = 1$ se o vértice *i* é alocado ao vértice *j* e $x_{ij} = 0$, caso contrário; $x_{ii} = 1$ se o vértice *i* é uma mediana e $x_{ii} = 0$, caso contrário; *p* é o número de medianas (facilidades) a serem localizadas e *n* é o número de vértices na rede, e $N = \{1, ..., n\}$.

A função objetivo (1) minimiza a soma das distâncias ponderadas d_{ij}, onde d_{ij} é o produto da distância entre os vértices v_i e v_j ($d(v_i, v_j)$) pelo peso w_j, sendo o peso w_j a demanda de cada vértice v_j. Assim $d_{ij} = w_j \cdot d(v_i, v_j)$.

As restrições (2) e (4) definem que cada vértice *j* é alocado a somente um vértice *i*, que deve ser uma mediana. A restrição (3) determina o número *p* de medianas a ser localizado e a restrição (5) corresponde às condições de integralidade.

A seguir apresenta-se o algoritmo de Evolução Diferencial utilizado neste trabalho.

3 Algoritmo de Evolução Diferencial para espaços contínuos

Price e Storn (1995) desenvolveram o algoritmo Evolução Diferencial (ED) com a intenção de resolver o problema de ajuste polinomial de Chebychev.

Inicialmente um conjunto populacional $G=\{V_1, V_2, ..., V_n\}$ de indivíduos (vetores) é gerado aleatoriamente e deve cobrir todo o espaço de busca. Na ausência de qualquer conhecimento acerca do espaço de busca (regiões promissoras ou mesmo soluções parciais), utiliza-se uma distribuição uniforme para a população inicial. Numa operação definida como mutação, o algoritmo ED gera novos vetores de parâmetros, chamados de vetores doadores, através da adição da diferença ponderada entre dois vetores a um terceiro indivíduo, conforme operação a seguir:

DOI: 10.5540/03.2014.002.01.0105

$$V_d = V_i + F(V_j - V_k) \text{ ou } V_d = V_{melhor} + F(V_j - V_k) \quad (6)$$

Onde V_d é o vetor doador, F é um peso escalar aplicado ao vetor diferença, V_{melhor} é o vetor da população que apresenta o melhor *fitness* e V_i , V_j e V_k representam indivíduos aleatórios e mutuamente distintos, escolhido da população. Outros operadores propostos de mutação podem ser encontrados em Price e Storn (1995).

Após realizada a mutação, escolhe-se aleatoriamente outro vetor, denominado vetor alvo (V_{alvo}), cujas componentes são misturadas com as componentes do vetor doador, resultando no vetor chamado experimental. Este processo é conhecido por cruzamento. Na versão básica do algoritmo ED um cruzamento binomial com probabilidade de cruzamento $CR \in [0,1]$ é apresentado conforme a seguinte regra:

$$e_i = \begin{cases} a_i, & \text{se } r_i \le CR \\ d_i, & \text{se } r_i > CR \end{cases} \quad i = 1, 2, ..., n \quad (7)$$

Sendo r_i números aleatórios pertencentes ao intervalo [0,1], e_i , d_i e a_i são os respectivos componentes dos vetores: experimental, doador e alvo.

A seleção entre os vetores alvo e experimental é realizada de forma gulosa, ou seja, se o valor da função objetivo aplicada ao vetor experimental for melhor do que o valor dela aplicada ao vetor alvo, então o vetor experimental substitui o vetor alvo na geração seguinte, caso contrário, o vetor alvo é mantido na próxima geração.

A operação mutação aplicada no ED é simples, porém, a maneira como está definida torna-a impraticável para problemas de otimização discreta [Deng *et al.,* 2009]. Surgem então pesquisas que propõem modificações na operação mutação para o ED a fim de inserir o algoritmo operacional no domínio discreto.

4 Evolução Diferencial para otimização discreta

Problemas de otimização combinatória discreta, abordados pelo algoritmo de Evolução Diferencial Discreta (EDD), geram novas propostas de melhoria para o método EDD. As propostas, em geral, se referem a adaptações nos operadores genéticos, sobretudo no operador de mutação, visando gerar soluções factíveis e maior velocidade de convergência para a solução ótima. Uma abordagem metaheurística para a Evolução Diferencial é proposta por Prado *et al.* (2010) para a otimização discreta, definindo a diferença entre duas soluções candidatas como uma lista de movimentos no espaço de busca, e assim preservando o mecanismo de busca em domínios discretos. O método foi aplicado em problemas do caixeiro viajante e no problema das N-Rainhas.

Neste artigo, é proposta uma nova forma de aplicar o operador de mutação na metaheurística Evolução Diferencial Discreta, que aborda problemas de *p*-medianas. O operador é descrito a seguir.

4.1 Algoritmo de Evolução Diferencial Discreto proposto para PPM (EDDppm)

As adaptações do algoritmo de ED apresentadas na literatura para otimização combinatória basicamente não abordam problemas de *p*-medianas. Uma nova e genérica versão para o operador de mutação do algoritmo ED para resolver o problema das *p*-medianas é proposta.

As variáveis envolvidas no operador de mutação proposto são: $G_{máx}$ o número máximo de iterações do algoritmo, N_p o número de vetores da população, N_{med} o número de medianas a ser determinada, N_v o número total de vértices do problema, (X_{ij}, Y_{ij}) a coordenada cartesiana da *i*-ésima mediana do *j*-ésimo vetor da população $G = \{V_1, V_2, ..., V_{Np}\}$, os valores reais F e λ e a taxa de cruzamento CR compreendidos entre 0 e 1. Considera-se ainda N um número inteiro positivo, onde N indica o número de medianas do vetor que apresenta melhor *fitness* da geração G (V_{melhor}) que irão sofrer a mutação. Seguem os passos do algoritmo proposto:

P1: Atribuir valores para os parâmetros $G_{máx}$, N_p, N e CR.

DOI: 10.5540/03.2014.002.01.0105

P2: Após a geração da população inicial, definir o V_{alvo} e selecionar o V_{melhor}.

P3: A seguir sorteiam-se:
- dois vetores V_1 e V_2 da população, de modo que V_1, V_2, V_{alvo} e V_{melhor} sejam distintos entre si;
- N medianas do V_{melhor}, que serão substituídas;
-uma componente do vetor V_1 e uma do vetor V_2 cujas coordenadas cartesianas são respectivamente (X_{n1}, Y_{n1}) e (X_{m2}, Y_{m2}).

P4: Obter aleatoriamente as coordenadas cartesianas (X_0 ,Y_0) pertencentes ao segmento de reta formado pelos pontos cujas coordenadas cartesianas sejam (X_{n1}, Y_{n1}) e (X_{m2}, Y_{m2}).

P5: Obter as coordenadas cartesianas (X_1 ,Y_1) pertencentes ao um ponto P_1 do espaço contínuo da seguinte forma: $P_1 = F\ (X_0, Y_0)$.

P6: Determinar o vértice P do problema mais próximo do ponto P_1 tal que o número de vezes que P tenha sido escolhido, nas iterações anteriores, para a mutação seja menor que $(G_{máx} / N_p)$ e $P \notin G$ (população).

P7: Substituir uma das N medianas selecionadas do V_{melhor} pelo vértice P. FIM

Obtém-se o vetor doador (V_{doador}) após as N medianas do V_{melhor} sofrerem a mutação.

Após a operação de mutação realiza-se o cruzamento binomial, descrito anteriormente (ver Seção 3), entre o V_{doador} e o V_{alvo}, com uma probabilidade de cruzamento $CR \in [0,1]$, resultando no vetor experimental (V_{exp}) e aplica-se a seleção gulosa, conforme descrito no algoritmo Evolução Diferencial contínuo, obtendo assim uma nova população.

No operador de mutação, as operações entre vetores, codificados com números inteiros, geralmente não geram soluções viáveis ao se multiplicar, por exemplo, um vetor por um escalar F ou λ compreendidos entre 0 e 1. A proposta de se utilizar as coordenadas cartesianas dos vértices neste operador tem o propósito de gerar novos vértices, no espaço contínuo, que estejam localizados no espaço de busca. O passo P6, do algoritmo proposto, retorna ao espaço discreto ao identificar o vértice do problema mais próximo do gerado no espaço contínuo, apresentando assim soluções viáveis.

A seguir apresentam-se os resultados obtidos utilizando o algoritmo EDD com as adaptações propostas para o problema das p-medianas.

5 Resultados e conclusões

Os algoritmos abordados foram programados em Visual Basic, versão 2010, e para o desenvolvimento computacional foi utilizado um computador Sony com processador Intel Core i5 com 640 GB de HD, 6 GB de RAM e sistema operacional Windows 7.

Diante da dificuldade de se encontrar instâncias na literatura que forneçam as coordenadas cartesianas dos vértices, foram geradas aleatoriamente 30 (trinta) instâncias, com no mínimo 100 e máximo 600 vértices, disponíveis em: <http://paginapessoal.utfpr.edu.br/durski>. Com o propósito de avaliar o desempenho do algoritmo proposto neste trabalho, em relação à qualidade das soluções obtidas e o tempo computacional necessário à obtenção da solução ótima, compararam-se os resultados obtidos com as soluções ótimas adquiridas através do software Lingo 13.0. Efetuaram-se 10 (dez) simulações para cada problema teste, adotando-se como critério de parada o número máximo de iterações.

Na Tabela 1 são apresentados os resultados da implementação do algoritmo proposto. Destacando-se nas duas primeiras colunas, respectivamente, o número n de vértices e o número p de medianas; a seguir são apresentadas as soluções ótimas (*SÓtimo*) e os tempos computacionais (em segundos) do software Lingo 13.0; nas colunas seguintes encontram-se as melhores e as piores soluções (*SMelhor* e *SPior*), a frequência (*Freq.*) de *SMelhor*, entre as 10 simulações efetuadas, e os desvios percentuais (D_p) apresentados pelo EDDppm que foram determinados da seguinte forma: $D_p = 100.(\ SMelhor - SÓtima)/\ SÓtima$.

DOI: 10.5540/03.2014.002.01.0105

Inst.	N	P	Lingo		EDD Proposto				
			SÓtimo	Tempo	SMelhor	SPior	Freq.	Dp	Tempo
1	100	5	351,46	18	351,46	351,46	10	0	< 1
2		10	219,23	15	219,23	219,23	10	0	< 1
3		20	130,97	16	130,97	131,46	8	0	< 1
4		33	86,88	15	86,88	87,70	5	0	1
5	200	5	3315,98	47	3315,98	3320,355	6	0	< 1
6		10	2173,28	39	2173,28	2173,28	10	0	2
7		20	1361,58	42	1361,58	1364,30	7	0	4
8		30	959,34	44	959,34	1029,60	4	0	7
9		40	836,3	42	838,73	843,92	3	0,291	11
10		67	524,27	46	524,27	525,92	3	0	18
11	300	5	15205,3	260	15205,3	15209,08	8	0	2
12		10	10404	256	10404	10455,03	7	0	4
13		30	5265,75	272	5276,21	5332,76	4	0,199	13
14		60	3095,3	265	3144,86	3171,56	3	1,601	14
15		100	1980,49	259	2003,99	2022,69	3	1,187	34
16	400	5	67035,20	455	67035,20	67035,20	10	0	3
17		10	46454	481	46454	46557,82	6	0	4
18		40	20009,9	519	20146,00	20503,37	4	0,680	17
19		80	12003,5	521	12162,82	12300,97	3	1,327	52
20		133	7679,36	511	7771,03	7840,12	2	1,194	88
21	500	5	85382,4	901	85382,4	85785,38	7	0	3
22		10	58590,9	848	58590,9	58904,86	6	0	6
23		50	22136,2	856	22379,18	22575,93	5	1,098	32
24		100	13301,6	886	13598,47	13808,22	4	2,232	64
25		167	8425,24	877	8612,28	8734,35	2	2,220	148
26	600	5	102849	1336	102849	102879,9	6	0	5
27		10	71249,4	1362	71249,4	71627,84	6	0	11
28		60	24210,7	1311	24402,6	24497,19	3	0,793	77
29		120	14364,2	1345	14647,88	14850,08	2	1,975	168
30		200	9074,87	1340	9187,14	9321,63	2	1,237	356

Tabela 1: Resultados dos testes computacionais

Com as informações da Tabela 1, constatou-se que o algoritmo EDDppm apresentou em 56,7% das instâncias soluções ótimas. O tempo computacional do algoritmo proposto é crescente à medida que se aumentam os dados de entrada. Obteve-se em 100% das instâncias tempos computacionais menores que o do obtido pelo software Lingo 13.0. A instância 30 apresentou o maior tempo

DOI: 10.5540/03.2014.002.01.0105

computacional, 356 segundos, que representa aproximadamente 26,57% do tempo apresentado pelo software Lingo 13.0.

Pode-se perceber que os valores dos desvios percentuais do EDDppm são pequenos, apresentando um valor máximo 2,232 na instância 24, demonstrando a efetividade do algoritmo de *p*-medianas para as instâncias abordadas.

Para trabalhos futuros, pretende-se investigar novas instâncias, com quantidades maiores de vértices e a combinação dessa metodologia com outras metaheurísticas encontradas na literatura.

Referências

[1] Correia, J.H.; Sousa, G.F.; Nascimento, I.Q.; Formiga, L.A.C.; Nascimento, R.Q. **Otimização e implementação de um sistema de alocação ótima de serviços públicos utilizando a metaheurística Grasp.** XLII Simpósio Brasileiro de Pesquisa Operacional, Bento Gonçalves, Brasil, Setembro de 2010.

[2] Christofides, N. **Graf theory – An algorithmic approach**. New York: ed. Academic Press, 1975.

[3] Deng, C.; Zhao, B.; Yang, Y.; Deng, A. **Novel Binary Differential Evolution Algorithm for Discrete Optimization**. Fifth International Conference on Natural Computation, p.346-349, 2009.

[4] Freitas, C.R.; Machado, C.M.S.; Retamoso, M.R. **Um método baseado na substituição de vértices e teoria espectral para problemas de p-medianas.** Claio, Simpósio Brasileiro de Pesquisa Operacional, Rio de Janeiro RJ, Brasil, setembro de 2012.

[5] Minieka, E. **Optimization algorithms for networks and graphs**. New York: ed. Marcel Dekker, INC. 1978.

[6] Prado, R. S.; Silva, R. C. P.; Guimarães, F. G.;Magela, O. **Uma nova abordagem para a evolução diferencial em otimização discreta.** XVIII Congresso Brasileiro de Automática. Bonito-MS, 2010.

[7] Price, K. V.; Storn, R. M. **Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuos Spaces**. Berkeley: Intenational Computer Science Institute, 1995.

