

**Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**

---

# Problema do Caixeiro Viajante abordado pelo Algoritmo de Evolução Diferencial

Danielle Durski Figueiredo<sup>1</sup>

Universidade Tecnológica Federal do Paraná, Departamento de Matemática, Curitiba-PR, Brasil

Carlos Ropelatto Fernandes<sup>2</sup>

Universidade Estadual do Paraná, Colegiado de Matemática, Paranavaí-PR, Brasil

Alessandra Beatriz Pachas Zavala<sup>3</sup>

Instituto Federal do Paraná, Departamento de Matemática, Pinhais-PR, Brasil

**Resumo.** 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, para problemas em espaços contínuos. Recentemente, tem sido propostas adaptações ao seu mecanismo de mutação diferencial para aplicação em problemas combinatoriais. Neste trabalho, uma proposta deste tipo de adaptação é apresentada para este algoritmo, para resolver o Problema do Caixeiro Viajante (PCV). O estudo dos PCV's se relaciona diretamente com problemas organizacionais da sociedade. Testes computacionais foram realizados com instâncias da literatura e os resultados encontrados sugerem que a técnica proposta é promissora e apropriada para a resolução dos problemas abordados.

**Palavras-chave.** Otimização, Evolução Diferencial Discreta, Problema do Caixeiro Viajante.

## 1 Introdução

O Problema do Caixeiro Viajante (PCV), é um dos mais tradicionais e frequentemente estudado em otimização combinatória [1]. O objetivo do PCV é encontrar, em um grafo  $G = (V, A)$ , o circuito hamiltoniano de menor custo. De forma simples, pode-se definir que um grafo é um conjunto de Vértices e Arestas. Os vértices (ou nós) são pontos que podem representar, por exemplo, cidades ou postos de atendimento. Já as arestas são linhas que conectam os vértices e podem representar, numa aplicação real, estradas por exemplo. Um circuito hamiltoniano é um passeio que percorre todos os vértices de um grafo e retorna ao vértice origem (início do passeio), passando por cada vértice apenas uma vez [3].

---

<sup>1</sup>durski@utfpr.edu.br

<sup>2</sup>cropelatto@yahoo.com.br

<sup>3</sup>alessandra.zavala@ifpr.edu.br

Sua descrição, embora simples, revela um problema de classe *NP*-difícil [5], o que o torna intratável na obtenção de soluções exatas. Desta forma, é comum a busca de abordagens heurísticas uma vez que não pode ser tratado por algoritmos polinomiais.

O foco deste trabalho é apresentar uma proposta de adaptação na operação de mutação diferencial em um algoritmo de Evolução Diferencial para aplicação em problemas de otimização discreta, especificamente para o problema do Caixeiro Viajante.

Este artigo está organizado em seis seções. Na seção 1 encontra-se uma introdução sobre o trabalho. Na seção 2, é apresentado o modelo matemático do PCV. Na seção 3, é descrito o algoritmo de Evolução Diferencial para Espaços Contínuos. O algoritmo de Evolução Diferencial Discreto para Espaços Discretos e o algoritmo proposto neste trabalho são apresentados na seção 4. Na seção 5 são discutidos e comentados os resultados da presente pesquisa e, finalmente, na seção 6 são feitas as conclusões.

## 2 Problema do Caixeiro Viajante

Em 1954, Dantzig-Fulkerson-Johnson, para resolver um problema com 49 cidades, formularam o PCV como um problema de programação 0-1 sobre um grafo  $G = (N, A)$ . Como as condições do problema exigem, essa formulação possui restrições tanto de controle de fluxo nos nós, como de eliminação de possíveis sub-rotas. Segundo [4], a forma geral da formulação matemática de Dantzig-Fulkerson-Johnson é dada por:

$$\text{Minimizar } Z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (1)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (4)$$

onde:

$x_{ij} = 1$ , se o arco for escolhido para integrar a solução;  $x_{ij} = 0$ , caso contrário;  $S$  é um subgrafo de  $G$ ;  $|S|$ : número de vértices do subgrafo  $S$ . Nessa formulação assume-se que:  $x_{ii}$  não existe, pois é sem aceção sair do nó e retornar para ele mesmo;  $n$  é o número de variáveis.

## 3 Algoritmo de Evolução Diferencial para Espaços Contínuos

O algoritmo de Evolução Diferencial (ED) foi desenvolvido por [8] com a intenção de resolver o problema de ajuste polinomial de Chebychev.

Inicialmente um conjunto populacional  $G = \{V_1, V_2, \dots, V_n\}$  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:

$$V_d = V_i + F(V_j - V_k) \quad \text{ou} \quad V_d = V_{melhor} + F(V_j - V_k) \quad (5)$$

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 [8].

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:

$$e_i = \begin{cases} a_i, & \text{se } \theta_i \leq CR \\ d_i, & \text{se } \theta_i > CR \end{cases} \quad i = 1, 2, \dots, n \quad (6)$$

Sendo  $\theta_i$  números aleatórios pertencentes ao intervalo  $[0, 1]$ ,  $e_i$ ,  $d_i$  e  $a_i$  são as respectivas 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 [2]. A seguir são apresentados alguns trabalhos 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

Nota-se que cada novo problema de otimização combinatória discreta, ao ser resolvido utilizando o algoritmo da Evolução Diferencial Discreta (EDD) gera novas propostas de melhoria para o método ED. 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 otimizada. A seguir são apresentados alguns trabalhos que propõem modificações na operação mutação para o ED a fim de inserir o algoritmo operacional no domínio discreto.

Uma abordagem meta-heurística para a Evolução Diferencial em otimização discreta é proposta por [7], 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. Em [12] também é apresentada uma adaptação do algoritmo EDD para o Problema do Caixeiro Viajante.

Outras aplicações do EDD no domínio discreto podem ser citadas: *scheduling flowshop* [6], [9], [13]; problema de atraso de máquina [11].

Neste artigo, é proposta uma nova forma de aplicar o operador de mutação na meta-heurística Evolução Diferencial Discreta, que aborda problemas de roteirização. O operador é descrito a seguir.

#### 4.1 Algoritmo de Evolução Diferencial Discreto proposto para PCV (EDDpcv)

Uma nova e genérica versão para o operador de mutação do algoritmo EDD para resolver o PCV é proposta. Seguem os passos do algoritmo proposto:

**P0:** Definem-se primeiramente as variáveis envolvidas no operador de mutação proposto. Seja  $G_{max}$  o número máximo de iterações do algoritmo,  $N_p$  o número de vetores da população,  $N_v$  o número total de vértices do problema,  $N_m$  o número de vértices do vetor que irá sofrer a mutação,  $(X_{ij}, Y_{ij})$  a  $i$ -ésima coordenada cartesiana do  $j$ -ésimo vetor da população  $G = \{V_1, V_2, \dots, V_{N_p}\}$  e os valores reais  $F$ ,  $\lambda$  e a taxa de cruzamento  $CR$  compreendidos entre 0 e 1. Considere ainda  $V_{melhor}$  o vetor que apresenta melhor *fitness* da geração  $G$ .

**P1:** Atribuir valores para os parâmetros  $G_{max}$ ,  $N_p$ ,  $N_m$  e  $CR$ ;

**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;

- de forma randômica escolher um vértice do  $V_{melhor}$ , que será substituído;

- 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 as coordenadas cartesianas  $(X, Y)$  pertencentes a um ponto  $P$  do espaço contínuo da seguinte forma:

$$X = F((1 - \lambda) \cdot X_{n1} + \lambda \cdot X_{m2}) \quad (7)$$

$$Y = F((1 - \lambda) \cdot Y_{n1} + \lambda \cdot Y_{m2}) \quad (8)$$

**P5:** Determinar o vértice  $P_0$  do problema mais próximo do ponto  $P$  tal que o número de vezes que  $P_0$  tenha sido escolhido, nas iterações anteriores, para a mutação seja menor que  $(G_{max}/N_p)$  e  $P_0 \notin V_{melhor}$ ;

**P6:** Substituir o vértice selecionado do  $V_{melhor}$  pelo vértice  $P_0$ . **FIM.**

Obtém-se o vetor doador ( $V_{doador}$ ) após os  $N_m$  vértices do  $V_{melhor}$  sofrerem a mutação.

A Figura 1 traz um exemplo bidimensional ilustrando os vetores que participam da geração de uma componente do vetor doador ( $V_{doador}$ ).

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.

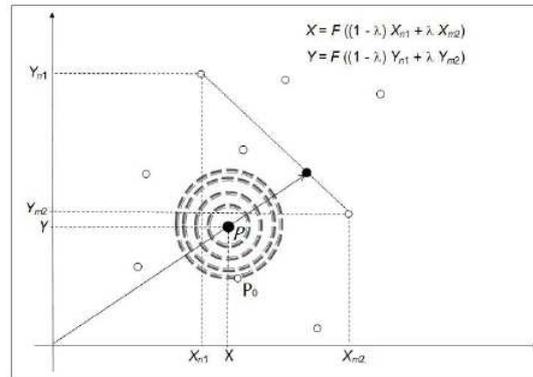


Figura 1: Processo de geração de uma componente do vetor doador ( $V_{doador}$ ).

Fonte: Autores.

A seguir apresentam-se os resultados obtidos utilizando o algoritmo proposto com as adaptações propostas para o PCV. Tais resultados foram comparados com instâncias encontradas na literatura.

## 5 Resultados

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.

Foram utilizadas 10 (dez) instâncias, com 48, 52, 70, 76, 99 e 100 vértices da base TS-PLIB. 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. Efetuou-se 10 (dez) simulações para cada problema teste, adotando-se como critério de parada o número máximo de iterações, determinando-se os desvios percentuais (erros), indicados por [10], da seguinte forma:  $Dp = 100(Media - Otima)/Otima$ .

Onde *Media* é a média entre as 10 (dez) simulações efetuadas em cada instância no algoritmo proposto e *Otima* é a solução ótima global dos problemas testados.

Na Tabela 1 são apresentados os resultados da implementação do algoritmo utilizado neste trabalho. Destacando-se nas duas primeiras colunas, respectivamente, o nome das instâncias com o número de vértices e as respectivas soluções ótimas. Nas colunas seguintes são apresentadas as soluções obtidas pelo algoritmo proposto, os desvios percentuais e os tempos médios computacionais (em segundos).

Com as informações da Tabela 1, constatou-se que o algoritmo proposto apresentou em 70% das instâncias soluções ótimas. O tempo computacional do algoritmo proposto é crescente à medida que se aumentam os dados de entrada.

Em 30% das instâncias testadas não se atingiu a solução ótima, mas os desvios apresentados pelo algoritmo proposto foram menores que 1%, mostrando que o mesmo apresenta

boa qualidade de solução.

Tabela 1: Resultados obtidos pelo algoritmo proposto (EDDpcv).

<i>Instâncias</i>	<i>Ótimo</i>	<i>Média</i>	$D_p$	<i>Tempo(s)</i>
att48	10628	10628	0,00	4,32
berlin52	7542	7543	0,01	4,91
kroA100	21282	21282	0,00	38,73
kroB100	22141	22210	0,31	35,20
kroC100	20749	20954	0,99	66,35
kroD100	21294	21294	0,00	41,63
kroE100	22068	22068	0,00	79,87
pr76	108159	108159	0,00	15,72
rat99	1211	1211	0,00	31,39
st70	675	675	0,00	6,60

O maior tempo computacional, 79,87 segundos, deu-se na instância kroE100, onde se atingiu a solução ótima em todas as simulações efetuadas.

A pesquisa ainda encontra-se em andamento e segue para verificação de seus resultados com outros métodos.

## 6 Conclusões

Neste trabalho apresentou-se uma nova abordagem heurística para o problema de roteirização, Caixeiro Viajante, apresentando uma proposta de adaptação do operador de mutação diferencial do algoritmo Evolução Diferencial, desenvolvido inicialmente para problemas onde o espaço solução é contínuo. Os testes foram realizados com 10 (dez) instâncias, onde soluções ótimas foram utilizadas para comparação.

O algoritmo proposto obteve, na maioria das instâncias testadas, soluções ótimas e considera-se aceitáveis os tempos computacionais necessários para atingir os melhores resultados.

Problemas de roteirização requerem, geralmente, que o algoritmo seja utilizado de forma constante. Desta forma, torna-se preferível que o algoritmo apresente um tempo computacional adequado com a oferta na qualidade da solução. Como exemplo pode-se citar o planejamento de um transporte escolar onde a economia no percurso percorrido é um benefício diário aos usuários deste recurso. Desta forma, o algoritmo proposto é viável na aplicação aos tipos de problemas abordados neste trabalho.

## Agradecimentos

Nossos agradecimentos a Universidade Tecnológica Federal do Paraná (UTFPR), a Universidade Estadual do Paraná (UNESPAR) campus de Paranavaí e ao Instituto Federal do Paraná (IFPR), Pinhais, pelo apoio durante a realização deste trabalho.

## Referências

- [1] D. L. Applegate, R. E. Bixby, V. Chvátal, W. J. Cook. The traveling salesman problem: a computational study. In *Princeton Series in Applied Mathematics*. Princeton University Press, Princeton, 2006. ISBN 9781400841103.
- [2] C. Deng, B. Zhao, Y. Yang, A. Deng. Novel Binary Differential Evolution Algorithm for Discrete Optimization. *Fifth International Conference on Natural Computation*, p. 346-349, 2009.
- [3] M. Ganhoto. O Problema do Caixeiro Viajante. *Sicorp News*, Ano I, nº 6, Brasil, julho de 2003.
- [4] M. C. Goldbarg, H. P. L. Luna. *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*. Campus, 2. ed., Rio de Janeiro, 2005.
- [5] A. C. B. Guedes, J. N. F. Leite, D. J. Aloise. Um algoritmo genético com infecção viral para o problema do caixeiro viajante. *XXVII Encontro de Engenharia de Produção*. Foz do Iguaçu, Brasil, 2007.
- [6] Q. K. Pan, L. Wang, B. Qian. A Novel Differential Evolution Algorithm for Bi-Criteria No-Wait Flowshop Scheduling Problems. *Computers & Operations Research*, v. 36, p. 2498-2511, 2009.
- [7] R. S. Prado, R. C. P. Silva, F. G. Guimarães, O. Magela. Uma nova abordagem para a evolução diferencial em otimização discreta. *XVIII Congresso Brasileiro de Automática*. Bonito-MS, 2010.
- [8] K. V. Price, R. M. Storn. Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341-359, 1997.
- [9] B. Qian, L. Wang, D. X. Huang, W. L. Wang, X. Wang. An Effective Hybrid DE-Based Algorithm for Multi-Objective Flow Shop Scheduling with Limited Buffers. *Computers and Operations Research*, v. 36, n. 1, p. 209-233, 2009.
- [10] E. L. F. Senne, L. A. N. Lorena. Abordagens Complementares para problemas de p-medianas. *Produção*, v. 13, n. 3, São Paulo, 2003.
- [11] M. F. Tasgetiren, Q. K. Pan, T. C. Liang. A Discrete Differential Evolution Algorithm for the Single Machine Total Weighted Tardiness Problem with Sequence Dependent Setup Times. *Computers & Operations Research*, v. 36, p. 1900-1915, 2009.
- [12] M. F. Tasgetiren, P. N. Suganthan, Q. K. Pan. An Ensemble of Discrete Differential Evolution Algorithms for Solving the Generalized Traveling Salesman Problem. *Applied Mathematics and Computation*, v. 215, p. 3356-3368, 2010.
- [13] L. Wang, Q. K. Pan, P. N. Suganthan,; W. H. Wang, Y. M. Wang. A Novel Hybrid Discrete Differential Evolution Algorithm for Blocking Flow Shop Scheduling Problems. *Computers & Operations Research*, v. 37, p. 509-520, 2010.