



## EVOLUÇÃO DIFERENCIAL APLICADA AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO EM SISTEMAS *FLOW SHOP* PERMUTACIONAL

### **Marco Antonio Reichert Boaretto**

Programa de Pós-Graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal do Paraná (UFPR), Rua Cel. Francisco Heráclito dos Santos, 100, Cep: 81531-980, Curitiba, PR, Brasil  
marco.boaretto@hotmail.com

### **Márcia de Fátima Moraes**

Universidade Estadual do Paraná  
Avenida Comendador Norberto Marcondes, 733, Campo Mourão, Paraná  
marciafmoraes@yahoo.com.br

### **Leandro dos Santos Coelho**

Programa de Pós-Graduação em Engenharia Industrial e Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Imaculada Conceição, 1155, Cep: 80215-901, Curitiba, PR, Brasil  
Programa de Pós-Graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal do Paraná (UFPR), Rua Cel. Francisco Heráclito dos Santos, 100, Cep: 81531-980, Curitiba, PR, Brasil  
lscoelho2009@gmail.com

### **RESUMO**

O algoritmo de Evolução Diferencial (ED) é uma metaheurística baseada em mecanismos evolutivos populacionais, cujas experiências em aplicações revelaram promissor seu uso para a solução de Problemas de Programação da Produção (PPP) em sistemas *Flow Shop* Permutacional (FSP). Embora diversas variações do algoritmo de ED para o PPP em FSP tenham sido propostas, a influência das estratégias de ED e da configuração dos parâmetros no desempenho do algoritmo não tem sido objeto de muitas investigações. Neste contexto, o propósito deste estudo foi avaliar o desempenho do algoritmo com diferentes estratégias de ED e diferentes valores para os parâmetros do algoritmo de ED aplicado ao PPP em sistemas FSP. Foram avaliadas 95 combinações de estratégias de ED com diferentes configurações de valores para os parâmetros e os resultados mostraram que as melhores estratégias de ED e os melhores valores para os parâmetros variam em função da dimensão do problema testado.

**PALAVRAS CHAVE.** Estratégias de Evolução Diferencial, Parâmetros do Algoritmo de Evolução Diferencial, Desempenho do Algoritmo.

### **ABSTRACT**

The Differential Evolution (DE) algorithm is a metaheuristic based on population evolutionary mechanisms, whose experiences on applications have shown promise its use in the solution of scheduling problems in Permutational Flow Shop (PFS) systems. Although several variations of the DE algorithm for scheduling problem in PFS have been proposed, the influence of DE strategies and parameter configuration on algorithm performance has not been the subject of many studies. In this context, the purpose of this study was to evaluate the performance of the algorithm with different DE strategies and different values for the parameters of the DE algorithm applied to scheduling problem in PFS systems. We evaluated 95 combinations of DE strategies with different configurations for the values of the parameters and the results showed that the best DE strategies and the best values for the parameters vary according to the dimension of the problem tested.

**KEYWORDS.** Differential Evolution Strategies. Parameters of the Differential Evolution Algorithm, Performance of the Algorithm.



## 1. Introdução

O acentuado desenvolvimento dos Algoritmos Evolutivos (AEs) nas últimas décadas tem melhorado sua eficiência e aplicabilidade na solução de problemas de otimização nas diferentes áreas da Engenharia e Ciência da Computação [Eiben e Smith 2015]. Os AEs tradicionalmente incluem Algoritmos Genéticos, Programação Evolutiva, Estratégias Evolutivas e Programação Genética [Brownlee 2011]. Algoritmos de Estimativa de Distribuição, Evolução Diferencial, entre outros mais recentemente desenvolvidos, também são categorizados como AEs [Simon 2013].

O algoritmo de Evolução Diferencial (ED), foco deste estudo, foi desenvolvido por [Storn e Price 1995, 1997] para solução de problemas complexos de otimização irrestrita e com variáveis contínuas. O algoritmo de ED é uma metaheurística baseada em mecanismos evolutivos populacionais que utiliza operadores genéticos de mutação, cruzamento e seleção para evoluir a população de soluções potenciais do problema a ser resolvido em direção ao ótimo global.

O Algoritmo de ED, diferente de outros algoritmos evolutivos, não usa funções de distribuição de probabilidade a fim de adicionar variações na população, ao invés disto, usa a diferença de vetores aleatoriamente selecionados como fonte de variações aleatórias [Price *et al.* 2005]. Uma importante característica da ED é a pequena quantidade de parâmetros utilizados, sendo eles a quantidade de indivíduos/vetores mantidos na população ( $Np$ ), o fator de mutação ( $F$ ), o fator de cruzamento ( $Cr$ ) e o número de gerações realizadas durante o processo (iterações) [Silva 2009].

Desde sua introdução, devido a sua efetividade, seus conceitos simples, sua facilidade de implementação e rapidez na convergência, muitos avanços têm sido obtidos no sentido de tornar viável a aplicação da abordagem de ED em diversas áreas da Engenharia [Tonge e Kulkarni 2012]. Embora o algoritmo de ED tenha sido originalmente projetado para otimização de variáveis contínuas [Ramos e Tupia 2013] nos últimos anos diversas versões discretas do algoritmo de ED têm sido propostas na literatura.

Experiências em aplicações do algoritmo de ED revelaram ser promissor o seu uso para a solução de Problemas de Programação da Produção (PPP) [Sun *et al.* 2011], [Tonge e Kulkarni 2012], [Jarboui, Siarry e Teghen 2013] e [Allahverdi 2015]. O PPP tratado neste estudo é o *Flow Shop* Permutacional (FSP), um caso especial do *Flow Shop* (FS).

O FSP é um típico problema de otimização combinatória, o qual determina a sequência de processamento das tarefas nas máquinas para satisfazer algum objetivo ou critério de desempenho [Mokhtari *et al.* 2011] e [Lin *et al.* 2015]. No FSP a ordem de processamento de todas as tarefas nas máquinas é a mesma em todas as máquinas do FS [Taillard 1993].

Embora diversas variações do algoritmo de ED tenham sido propostas para o PPP em sistemas FSP, a influência das estratégias de ED e da configuração dos parâmetros no desempenho do algoritmo não tem sido alvo de muitas investigações. No estudo realizado por [Carvalho *et al.* 2016], cujo propósito foi identificar e analisar as variantes adotadas nos algoritmos de ED desenvolvidos para PPPs em sistemas FSP, não foram identificados padrões específicos para a configuração dos parâmetros dos algoritmos de ED desenvolvidos para os PPPs.

Assim, diante do exposto, o principal propósito deste estudo foi avaliar o desempenho do algoritmo com diferentes estratégias de ED e diferentes valores para os parâmetros do algoritmo de ED aplicado aos PPPs em sistemas FSP. Para o alcance de tal objetivo, a versão original do algoritmo de ED [Storn e Price 1995, 1997] foi adaptada para o caso de minimização do *Makespan* em sistemas FSP.

Este artigo encontra-se estruturado em cinco seções. Após a contextualização e ambientalização do estudo mencionado nesta seção, o referencial teórico referente ao algoritmo de ED e ao Método de Taguchi para seleção de parâmetros é exposto na seção 2. O delineamento da experimentação é apresentado na terceira seção. A quarta seção, denominada resultados e discussões, contempla e discute os resultados obtidos na experimentação computacional. Por fim, as considerações finais são apresentadas.

## 2. Evolução Diferencial



A Evolução Diferencial (ED) é um método de busca paralela direta que tem como conceito básico a utilização de uma população composta por  $Np$  indivíduos (vetores), que representam soluções potenciais (soluções candidatas) dentro do espaço de busca do problema a ser resolvido [Brownlee 2011].

No algoritmo de ED a população de soluções candidatas é submetida a operações de mutação, cruzamento, avaliação e seleção. As operações de mutação, cruzamento, avaliação e seleção são repetidas geração após geração até que algum critério de parada seja satisfeito [Storn e Price 1997] e [Brownlee 2011].

O algoritmo original da ED baseado em [Storn e Price, 1997], [Price *et al.* 2005] e [Simon 2013] pode ser definido pelas seguintes etapas:

### 1) Inicialização

Nesta etapa são determinados os valores dos limites superiores ( $b_{j,U}$ ) e inferiores ( $b_{j,L}$ ) para definir o domínio em que os valores serão escolhidos. A população inicial de vetores é formada aleatoriamente e uniformemente distribuída entre os limites superiores e inferiores do problema e deve cobrir a totalidade do espaço de busca. A ED usa a diferença de vetores aleatoriamente selecionados como fonte de variações aleatórias para gerar o vetor alvo  $x_{j,i,0}$  da geração inicial ( $g=0$ ) da  $j$ -ésima dimensão da  $i$ -ésima população, como pode ser vista na equação (1) a seguir:

$$x_{j,i,0} = rand_j(0,1).(b_{j,U} - b_{j,L}) + b_{j,L} \quad (1)$$

onde  $rand_j$  é a função geradora de números aleatórios uniformemente distribuídos entre 0 e 1. O subscrito  $j$  indica que a cada parâmetro é gerado um valor aleatório novo.

### 2) Mutação

No processo de mutação, a modificação da população inicia-se pela geração de novos indivíduos denominados vetores modificados ou doadores. Para cada indivíduo da população, denominado vetor alvo, um vetor doador é gerado. Na versão clássica do algoritmo ED, os vetores doadores são gerados pela adição da diferença ponderada entre dois indivíduos da população escolhidos aleatoriamente e uniformemente a um terceiro indivíduo da população também escolhido aleatoriamente e uniformemente. Para cada vetor alvo  $x_{i,G}$ ,  $i=0,1,\dots, N_p - 1$  em uma geração  $G$ , um vetor mutante  $v_{i,G}$ , resultado da combinação de três diferentes vetores, escolhidos aleatoriamente, é gerado de acordo com a Equação (2):

$$v_{i,G} = x_{r_0,G} + F.(x_{r_1,G} - x_{r_2,G}) \quad (2)$$

com índices aleatórios,  $r_0, r_1, r_2 \in \{1, 2, \dots, Np\}$  inteiros, mutuamente diferentes e  $F > 0$ .  $F$  é uma constante de mutação  $\in [0, 2]$  o qual controla a amplificação da variação diferencial  $(x_{r_1,G} - x_{r_2,G})$ .

### 3) Cruzamento

A operação de cruzamento, também chamada reprodução, é introduzida após a etapa de mutação para aumentar a diversidade de vetores parâmetros perturbados. A ED emprega cruzamento uniforme, também referenciado como recombinação discreta, que constrói vetores de teste de valores de parâmetros que foram copiados a partir de dois vetores diferentes. O algoritmo de ED cruza cada vetor com um vetor mutante, conforme equação (3):

$$u_{i,G} = u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{se } (rand_{i,j}[0,1] \leq CR) \text{ ou } j = j_{rand} \\ x_{j,i,G} & \text{caso contrário} \end{cases} \quad (3)$$

onde  $rand_{i,j}[0,1]$  é um número aleatório uniformemente distribuído, o qual é chamado um novo para cada  $j$ -ésima componente do  $i$ -ésimo vetor parâmetro.  $Cr$  é uma constante de cruzamento



$\in [0,1]$  especificada pelo usuário.  $j_{rand} \in 1,2,\dots,D$  é um índice escolhido aleatoriamente, o qual garante que  $u_{i,G}$  receba pelo menos um parâmetro de  $v_{i,G}$ .

#### 4) Seleção

Para decidir se deve ou não tornar-se um membro da geração  $G+1$ , o vetor de ensaio  $u_{i,G}$  é comparado com o vetor alvo  $x_{i,G}$  usando um critério ganancioso. Se o vetor experimental produzir um valor menor ou igual que a função de custo do vetor alvo, o vetor de ensaio substitui o vetor alvo na geração seguinte. Caso contrário, o vetor alvo permanece na população para a próxima geração. Ou seja, se o vetor  $u_{i,G}$  produz um valor menor do que a função de custo de  $x_{i,G}$ , então  $x_{i,G+1}$  é definido como  $u_{i,G+1}$ , caso contrário, o antigo valor de  $x_{i,G}$  é retido, conforme equação (4):

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{se } f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G} & \text{caso contrário} \end{cases} \quad (4)$$

Esta última operação é chamada de seleção. Cada vetor população tem que servir uma vez como o vetor alvo para que as  $Np$  competições tenham lugar em uma geração.

## 2.1 Evolução Diferencial Discreta

Conforme mencionado na seção introdutória deste artigo, o algoritmo de ED foi originalmente projetado para otimização de variáveis contínuas [Ramos e Tupia 2013]. No entanto, versões discretas do ED propostas na literatura tem mostrado que o algoritmo é capaz de produzir bons resultados quando aplicado a diferentes tipos de problemas.

A única fase em que os domínios discretos causam problemas em um algoritmo de ED é na geração do vetor mutante [Simon 2013]. A estratégia mais simples para o uso do algoritmo de ED para a solução de problemas de otimização com variáveis no domínio do conjunto de números inteiros consiste na aproximação dos valores fracionários para o inteiro mais próximo quando necessário [Silva 2014].

No entanto, a literatura apresenta diversas abordagens para tratar a questão da conversão dos domínios nos algoritmos de ED, dentre as quais são destacadas: Abordagem por Matriz de Permutação; Abordagem por Matriz de Adjacência; Abordagem de Índice de Posição Relativa; Abordagem de Posição de Menor Valor; Abordagem de Transformação *Forward/Backward*.

Neste estudo foi utilizada para conversão dos domínios contínuos em discretos, a Abordagem de Índice de Posição Relativa, em que permutações são obtidas por determinação dos tamanhos relativos dos diferentes parâmetros que definem uma instância [Onwubolu e Davendra 2009].

Na Abordagem de Índice de Posição Relativa, após a aplicação dos operadores de mutação e recombinação, os vetores gerados provavelmente possuirão valores fracionários. Assim, antes das soluções serem avaliadas pela função objetivo, elas são transformadas em vetores de permutação de inteiros válidos [Silva 2014].

O processo de conversão dos vetores, de acordo com a abordagem de Índice de Posição Relativa ocorre da seguinte forma: dado o vetor  $u_{i,G}$ , obtido após a mutação e recombinação, e considerando que uma solução válida é uma permutação de rótulos numéricos de 1 a  $n$ , o rótulo 1 é alocado na mesma posição onde se encontra o menor valor em  $u_{i,G}$ , o rótulo 2 na posição do segundo menor valor em  $u_{i,G}$ , e assim por diante [Silva 2014].

## 2.2 Estratégias de Evolução Diferencial



O esquema de mutação ED/*rand*/*1*/*bin* mostrado na seção anterior não é a única variante da Evolução Diferencial que provou ser útil. O vetor mutante pode geralmente ser gerado usando cinco diferentes esquemas de mutação e cruzamento, propostos por [Storn e Price 1995, 1997], que dão origem a dez variantes de ED, conhecidas na literatura como Estratégias de Evolução Diferencial.

As estratégias de ED são representadas como ED/*x*/*y*/*z*, sendo que: *x* determina o vetor que será perturbado, podendo ser “*rand*” que significa que um vetor da população foi escolhido aleatoriamente ou “*best*” que significa o vetor de menor custo da população foi escolhido; *y* determina a quantidade de diferenças ponderadas utilizadas para a perturbação de *x*, podendo ser “*1*” ou “*2*”; e *z* especifica qual o tipo de cruzamento, podendo ser “*exp*” que significa exponencial ou “*bin*” que significa binomial.

As dez estratégias de ED, desenvolvidas por [Storn e Price 1995, 1997] que podem ser adotadas nos algoritmos de ED são apresentadas no Quadro 1 a seguir.

Estratégia	Equação para geração do vetor mutante ( $v_{i,G}$ )
1) ED/ <i>best</i> / <i>1</i> / <i>exp</i>	$v_{i,G} = x_{best,G} + F.(x_{r0,G} - x_{r1,G})$
2) ED/ <i>rand</i> / <i>1</i> / <i>exp</i>	$v_{i,G} = x_{r0,G} + F.(x_{r1,G} - x_{r2,G})$
3) ED/ <i>rand-to-best</i> / <i>1</i> / <i>exp</i>	$v_{i,G} = x_{i,G} + F.(x_{best,G} - x_{i,G}) + F.(x_{r0,G} - x_{r1,G})$
4) ED/ <i>best</i> / <i>2</i> / <i>exp</i>	$v_{i,G} = x_{best,G} + F.(x_{r0,G} - x_{r1,G}) + F.(x_{r2,G} - x_{r3,G})$
5) ED/ <i>rand</i> / <i>2</i> / <i>exp</i>	$v_{i,G} = x_{r0,G} + F.(x_{r1,G} - x_{r2,G}) + F.(x_{r3,G} - x_{r4,G})$
6) ED/ <i>best</i> / <i>1</i> / <i>bin</i>	$v_{i,G} = x_{best,G} + F.(x_{r0,G} - x_{r1,G})$
7) ED/ <i>rand</i> / <i>1</i> / <i>bin</i>	$v_{i,G} = x_{r0,G} + F.(x_{r1,G} - x_{r2,G})$
8) ED/ <i>rand-to-best</i> / <i>1</i> / <i>bin</i>	$v_{i,G} = x_{i,G} + F.(x_{best,G} - x_{i,G}) + F.(x_{r0,G} - x_{r1,G})$
9) ED/ <i>best</i> / <i>2</i> / <i>bin</i>	$v_{i,G} = x_{best,G} + F.(x_{r0,G} - x_{r1,G}) + F.(x_{r2,G} - x_{r3,G})$
10) ED/ <i>rand</i> / <i>2</i> / <i>bin</i>	$v_{i,G} = x_{r0,G} + F.(x_{r1,G} - x_{r2,G}) + F.(x_{r3,G} - x_{r4,G})$

Quadro 1 – Estratégias de Evolução Diferencial

Fonte: [Storn e Price 1995, 1997]

Ressalta-se que a estratégia de Evolução Diferencial adotada no desenvolvimento dos métodos de solução é uma questão muito importante, pois o algoritmo pode apresentar diferentes comportamentos baseados na estratégia selecionada.

## 2.3. Parâmetros do Algoritmo de ED

### 2.3.1. Tamanho da População ( $Np$ )

A população inicial deve ser gerada o mais próximo possível da superfície da função objetivo [Storn e Price 1997]. Em algoritmos de ED,  $Np$  deve ser maior ou igual a 4 [Storn e Price 1997], sendo sugerido pelos mesmos autores que  $Np$  esteja entre  $5D$  e  $10D$ . [Gamperle *et al.* 2002] propõem que  $Np$  seja escolhido entre  $3D$  e  $8D$ , enquanto [Rönkkönen *et al.* 2005] sugerem que  $Np$  seja escolhido entre  $2D$  e  $40D$ .

Para [Fachin 2014] aumentando-se o tamanho da população, aumenta-se a diversidade do universo de potenciais vetores experimentais e minimiza-se, portanto, o risco de estagnação devido a uma convergência prematura. Por outro lado, grandes valores de  $Np$  aumentam o tempo computacional. Portanto a escolha de  $Np$  é um compromisso entre tempo de cálculo e convergência.



### 2.3.2. Fator de Mutação ( $F$ )

A literatura apresenta diversas formas de definir o parâmetro  $F$ , não havendo consenso sobre tal questão, sendo sugerido por [Simon 2013] o intervalo  $0,4 < F < 0,9$  e por [Das e Suganthan 2011] o intervalo  $0,4 < F < 1$ . De acordo com [Storn e Price 1997] iniciar com  $F = 0.5$  é geralmente uma boa escolha e a utilização de valores menores que 0,4 e maiores que 1 pode ocasionar uma degradação do desempenho do algoritmo. De acordo com [Rönkkönen *et al.* 2005] típicos valores para o fator de mutação  $F$  seriam  $0,4 < F < 0,95$  sendo  $F = 0,9$  um bom compromisso entre velocidade e probabilidade de convergência. Todavia, [Gamperle *et al.* 2002] consideram  $F = 0,6$  uma boa escolha inicial.

[Gamperle *et al.* 2002] destacam que embora valores elevados de  $F$  aumentem a probabilidade de o algoritmo escapar de um mínimo local, valores  $F > 1$  diminuem consideravelmente a velocidade de convergência. [Price *et al.* 2005] estabelecem que o fator de mutação  $F$  deve possuir um limite superior e inferior, e ressaltam que para evitar uma convergência prematura,  $F$  deve ser suficientemente grande para neutralizar o efeito da seleção. Além disto, valores pequenos de  $F$  reduzem a chance de se escapar de um mínimo local.

### 2.3.1. Fator de Cruzamento ( $Cr$ )

Frequentemente a probabilidade de cruzamento  $Cr \in [0; 1]$  deve ser considerada menor do que 1. Caso não ocorra convergência, uma  $Cr \in [0,8; 1]$  pode ajudar [Storn e Price 1997].

Quanto menor o valor de  $Cr$ , menor a probabilidade de o vetor experimental receber componentes do vetor doador [Fachin 2014]. Para [Gamperle *et al.* 2002] valores elevados para  $Cr$  podem levar a convergência prematura ou a diminuição da velocidade de convergência, devendo ser escolhido um valor entre 0,3 e 0,9. [Rönkkönen *et al.* 2005] sugerem que a taxa de cruzamento deve ser escolhida dentre da faixa  $0 \leq Cr \leq 0,2$  para funções separáveis ou  $0,9 \leq Cr \leq 1$  para funções não separáveis e multimodais

## 2.4 Método Taguchi para Combinação de Parâmetros

A ED como qualquer metaheurística, possui parâmetros a serem definidos como  $F$ ,  $Cr$  e  $Np$  que ao serem variados podem retornar uma resposta muito diferente para cada diferente combinação de valores. E dependendo do experimento a ser trabalhado, a sintonia dos parâmetros pode ser um tanto trabalhosa quando for comparado um número alto de combinações de parâmetros.

A fim de diminuir o trabalho de combinar cada um dos possíveis valores de parâmetros e otimizar o tempo dos experimentos, foram desenvolvidos métodos de combinação de parâmetros como método Taguchi [Taguchi 1986] que ficou popular nos anos 80 por ser uma robusta ferramenta de design de parâmetros.

O Método Taguchi utiliza um vetor ortogonal para classificar os resultados do experimento [Mozdgir *et al.* 2013], o conceito chave deste método consiste na maximização da taxa de sinal-para-ruído que é utilizado como uma medida de performance dos experimentos [Tsai 2015]. A taxa de sinal ruído, equação 5, representa a quantidade de variação presente na variável de resposta [Mozdgir *et al.* 2013].

$$\text{taxa} \frac{S}{N} = -10 \log_{10}(\text{função objetivo})^2 \quad (5)$$

onde,  $S$  é o termo sinal que indica o valor desejado,  $N$  o ruído que indica o valor indesejado, e *função objetivo* se refere ao custo da mesma.

Para gerar a matriz de valores dos parâmetros o método Taguchi utiliza o número de graus de liberdade para cada variável [Mozdgir *et al.* 2013]. Após terem sido feitos os testes com um número de combinações reduzidas é calculado a taxa sinal-para-ruído de cada um dos valores das funções objetivos resultantes. A média dos valores da taxa sinal-para-ruído das soluções geradas





para cada parâmetro é calculado e então os parâmetros que apresentam o maior valor da média são definidos como os melhores parâmetros para o problema aplicado.

### 3. Delineamento da Experimentação

Para analisar a influência de diferentes configurações dos parâmetros no desempenho do algoritmo de ED, a versão clássica do algoritmo [Storn e Price 1995, 1997], disponível em [ICSI 2017], foi adaptada para o problema de minimização do *Makespan* em ambientes *Flow Shop* Permutacional e codificada em Matlab.

O algoritmo de ED adaptado para o problema em questão foi testado no *benchmark* de Heller, composto por dois casos. A Tabela 1 apresenta os casos testes, com suas respectivas configurações de tarefas e máquinas, bem como seus limitantes superiores ( $C_{max}$ ).

Tabela 1 – *Benchmark* de Heller

Caso Teste	Número de Tarefas	Número de Máquinas	Limitante Superior ( $C_{max}$ )
Hel_001_20_10	20	10	136
Hel_002_100_10	100	10	516

Fonte, OR *Library*, 2017 e Ancão, 2012.

Para cada uma das diferentes combinações de parâmetros investigadas neste estudo, para cada caso teste foram realizados 50 experimentos (*runs*), sendo utilizado como critério de parada o número máximo de 2000 gerações.

Inicialmente foram testadas todas as dez estratégias de ED [Storn e Price 1997], sendo definido  $Np = 50$  e 7 diferentes combinações de valores para os parâmetros  $F$  e  $Cr$ . Nesta etapa do estudo, para cada caso teste, foram avaliadas 70 combinações de parâmetros, ou seja, 7 combinações de  $F$  e  $Cr$  multiplicado por 10 estratégias de ED (Quadro 1 – referencial teórico). A Tabela 2 ilustra as combinações de valores para os parâmetros  $F$  e  $Cr$  selecionadas para testes.

Tabela 2 – Combinações de valores para os parâmetros  $F$  e  $Cr$

Combinações de Valores para os Parâmetros $F$ e $Cr$													
1		2		3		4		5		6		7	
$F$	$Cr$	$F$	$Cr$	$F$	$Cr$	$F$	$Cr$	$F$	$Cr$	$F$	$Cr$	$F$	$Cr$
0,2	0,1	0,4	0,2	0,4	0,3	0,5	0,5	0,8	0,8	0,9	0,1	0,9	0,9

Fonte, o autor 2017.

Posteriormente, foram geradas 25 novas combinações de valores para os parâmetros  $F$ ,  $Cr$  e  $Np$ , bem como a estratégia de ED a ser empregada na execução do algoritmo. Para cada caso teste, estas 25 novas combinações foram avaliadas utilizando o Método de Taguchi. A Tabela 3 ilustra as combinações de estratégias de ED e de valores dos parâmetros  $F$ ,  $Cr$  e  $Np$ .

Tabela 3 – Combinações de estratégias de ED e valores para os parâmetros  $F$ ,  $Cr$  e  $Np$

Combinações Estratégias de ED ( $E$ ) e Valores para os Parâmetros $F$ , $Cr$ e $Np$																			
1				2				3				4				5			
$E$	$F$	$Cr$	$Np$	$E$	$F$	$Cr$	$Np$	$E$	$F$	$Cr$	$Np$	$E$	$F$	$Cr$	$Np$	$E$	$F$	$Cr$	$Np$
4	0,1	0,1	30	2	0,1	0,2	70	6	0,1	0,5	100	8	0,1	0,5	50	1	0,1	0,9	90
6				7				8				9				10			
6	0,2	0,1	90	10	0,2	0,2	100	1	0,2	0,5	30	7	0,2	0,8	50	3	0,2	0,9	70
11				12				13				14				15			
9	0,5	0,1	70	6	0,5	0,2	50	7	0,5	0,5	90	8	0,5	0,8	100	5	0,5	0,9	30
16				17				18				19				20			
5	0,8	0,1	100	8	0,8	0,2	70	9	0,8	0,2	90	3	0,8	0,8	30	10	0,8	0,8	70
21				22				23				24				25			
2	0,9	0,9	50	3	0,9	0,1	50	4	0,9	0,5	70	5	0,9	0,8	90	7	0,9	0,9	30

Fonte, o autor 2017.

Todas as combinações de valores para  $F$ ,  $Cr$  e  $Np$  foram selecionadas pelos autores, tendo como base os trabalhos no referencial teórico, bem como diversos outros trabalhos orientados aos



problemas de programação da produção em ambientes *Flow Shop*, reportados na literatura especializada e sumarizados em [Carvalho *et al.* 2016].

#### 4. Resultados e Discussões

##### 4.1. Resultados Obtidos

Para o primeiro conjunto de testes, todas as 70 combinações de parâmetros foram analisadas tomando como base as estatísticas resultantes, ou seja, em termos de valores mínimo, médio e máximo, bem como em termos de desvio padrão, tempo computacional e convergência.

Para os dois casos testes, todas as estratégias de ED foram capazes de obter soluções melhores que os limitantes superiores disponíveis na literatura especializada. Assim, para definição dos melhores valores para os parâmetros  $F$  e  $Cr$ , tomamos como referência a convergência do método, ou seja, a iteração em que o método convergiu para a melhor solução. As tabelas 4 e 5 ilustram os melhores valores obtidos para cada estratégia de ED, para os casos testes, Hel\_001\_20\_10 e Hel\_002\_100\_10, respectivamente.

Tabela 4 – Melhores valores obtidos para cada estratégia de ED: Caso teste\_Hel\_001\_20\_10

Estratégia	$F$	$Cr$	Min	Média	Max	Desvio Padrão	Tempo (s)	Convergência
1	0,9	0,1	135	136,88	139	0,5584	1,66134	1793
2	0,5	0,5	135	136,92	137	0,3405	2,17038	762
3	0,4	0,3	135	137,28	139	0,783503	11,05248	314
4	0,4	0,2	135	136,94	138	0,424264	9,08054	905
5	0,2	0,1	135	135,94	137	0,6197	1,6174	1165
6	0,9	0,1	135	136,88	139	0,5584	1,81724	1261
7	0,2	0,1	135	135,84	137	0,7103	1,60958	858
8	0,5	0,5	135	138,3	141	1,4604	1,76288	280
9	0,2	0,1	135	136,04	137	0,6376	1,90456	1475
10	0,2	0,1	135	136,04	137	0,6376	1,90456	1475

Fonte, o autor 2017.

Tabela 5 – Melhores valores obtidos para cada estratégia de ED: Caso teste\_Hel\_002\_100\_10

Estratégia	$F$	$Cr$	Min	Média	Max	Desvio Padrão	Tempo (s)	Convergência
1	0,2	0,1	515	520,1	526	2,7939	25,8182	720
2	0,2	0,1	515	515,98	520	0,8449	24,08228	1600
3	0,4	0,3	515	521,7	527	2,873169	19,51416	1149
4	0,2	0,1	515	519,2	524	2,733018	27,75638	1879
5	0,2	0,1	515	516,38	518	0,635353	20,6304	1483
6	0,2	0,1	515	519,44	524	2,921839	25,2105	607
7	0,2	0,1	515	515,74	517	0,527218	25,77904	1596
8	0,4	0,3	515	519,56	524	2,572698	20,95454	995
9	0,2	0,1	515	517,82	524	2,446906	27,23632	963
10	0,2	0,1	515	515,98	517	0,318799	16,39993	1239

Fonte, o autor 2017.

As Figuras 1 e 2 representam a distribuição dos melhores valores de todos os 50 experimentos para as melhores combinações de  $F$  e  $Cr$  de cada uma das 10 estratégias para os casos





testes, Hel\_001\_20\_10 e Hel\_002\_100\_10, respectivamente. Ressaltamos que cada estratégia foi executada com 7 diferentes combinações de valores para  $F$  e  $Cr$ .

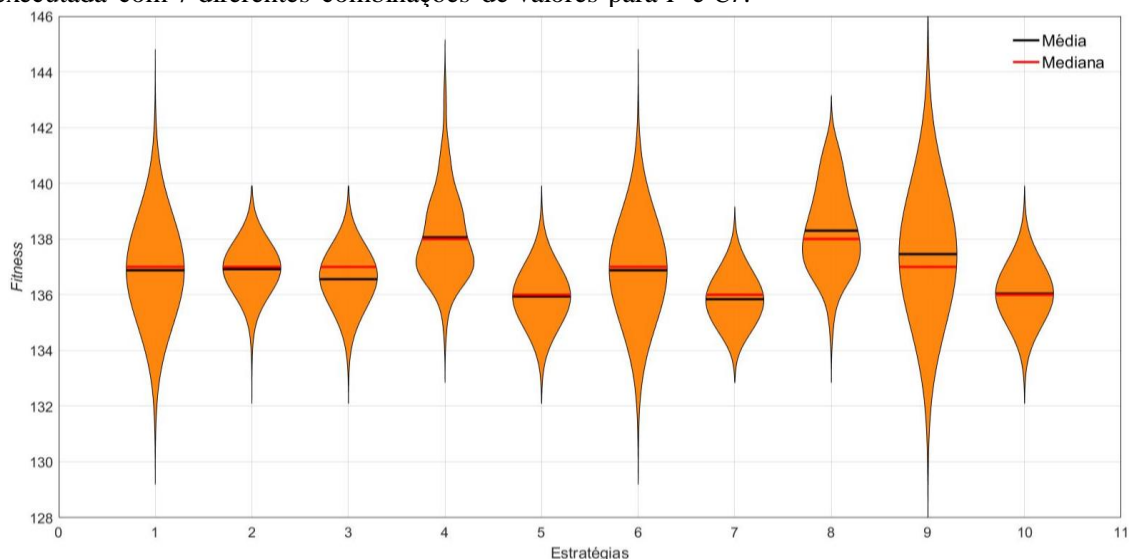


Figura 1 – Distribuição dos resultados dos 50 experimentos das melhores combinações de  $F$  e  $Cr$  de cada uma das 10 estratégias para o caso teste\_Hel\_001\_20\_10.

Fonte, o autor 2017

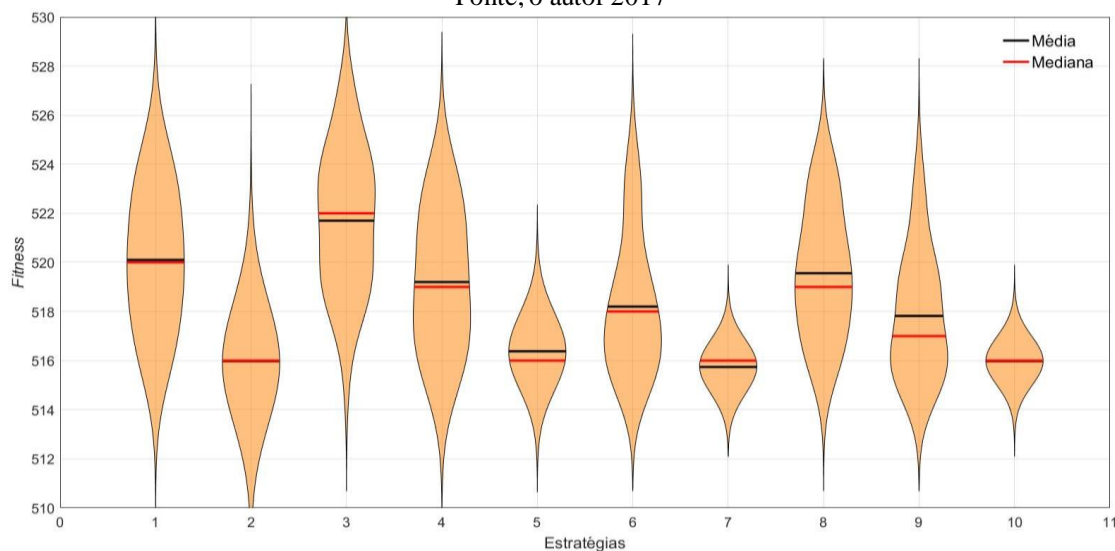


Figura 2 – Distribuição dos resultados dos 50 experimentos das melhores combinações de  $F$  e  $Cr$  de cada uma das 10 estratégias para o caso teste\_Hel\_002\_100\_10.

Fonte, o autor 2017

Para o segundo conjunto de testes, todas as 25 combinações de parâmetros foram analisadas através do Método de Taguchi. A Tabela 6 ilustra os melhores resultados fornecidos pelo Método de Taguchi, para os casos testes, Hel\_001\_20\_10 e Hel\_002\_100\_10 e Caso teste\_Hel\_002\_100\_10.

Tabela 6 – Melhores resultados obtidos pelo Método de Taguchi: Caso teste\_Hel\_001\_20\_10

Caso teste	Estratégia	F	Cr	NP	S/N	Min	Média	Max	Desvio	Tempo(s)	Convergência
Hel_001_20_10	9	0,5	0,2	50	-42,61	135	137,76	137	0,5175	1,65746	1376
Hel_002_100_10	9	0,8	0,1	50	-54,25	516	518,56	521	1,2643	16,4285	1745

Fonte, o autor 2017.

#### 4.2 Discussão dos Resultados



Utilizando o método das Estatísticas para análise das configurações (Tabelas 4 e 5), para ambos os casos testes, todas as estratégias de ED foram capazes de obter soluções melhores que os limitantes superiores dos problemas, no entanto, em termos de convergência para o caso teste Hel\_001\_20\_10 a estratégia 8 (ED/and-to-best/1/bin), executada com a configuração de parâmetros  $Np=50$ ,  $F=0,5$  e  $Cr=0,5$  obteve melhor desempenho em termos de convergência, enquanto que para o caso teste Hel\_002\_100\_10 a estratégia 6 (ED/best/1/bin), executada com a configuração de parâmetros  $Np=50$ ,  $F=0,2$  e  $Cr=0,1$  obteve melhor desempenho em termos de convergência.

Através do Método de Taguchi (Tabelas 6) para o caso teste Hel\_001\_20\_10 a estratégia 9 (ED/best/2/bin), executada com a configuração de parâmetros  $Np=50$ ,  $F=0,5$  e  $Cr=0,2$  apresenta melhor desempenho, enquanto que para o caso teste Hel\_002\_100\_10 a estratégia 9 (ED/best/2/bin), executada com a configuração de parâmetros  $Np=50$ ,  $F=0,8$  e  $Cr=0,1$  apresenta melhor desempenho.

A Tabela 7 mostra os melhores resultados obtidos com o método de análise das Estatísticas e com o Método de Taguchi.

Tabela 7 - Resultados em comparação com o método Taguchi

Caso teste Hel_001_20_10									
Método	F	Cr	Estratégia	NP	Min	Média	Max	Desvio	Convergência
Estatísticas	0.5	0.5	8	50	135	138.3	141	1.4604	280
Taguchi	0.5	0.2	9	50	135	137.76	137	0.5175	1376
Caso teste Hel_002_100_10									
Método	F	Cr	Estratégia	NP	Min	Média	Max	Desvio	Convergência
Estatísticas	0.2	0.1	6	50	515	519.44	524	2.921839	607
Taguchi	0.8	0.1	9	50	516	518.56	521	1.2643	1745

Fonte, o autor 2017.

Como pode ser visualizado na Tabela 8, o método de seleção dos parâmetros via análise das estatísticas e o método de Taguchi, para ambos os casos testes, divergem no que diz respeito a melhor estratégia de ED a ser adotada e também no que diz respeito a combinação de melhores valores para a configuração dos parâmetros  $F$  e  $Cr$ .

Como pode ser visto, nos resultados apresentados, para ambos os casos testes, embora os parâmetros definidos pelo Método Taguchi (melhores configurações) também conseguiram alcançar o limitante da função objetivo, nas configurações determinadas pelo Método das Estatísticas, o número de gerações em que o valor convergiu foi bem menor.

Quando analisadas variações no tamanho de  $Np$ , verificou através do Método de Taguchi que o tamanho de  $Np$  não variou em função da dimensão do problema, sendo verificado que para ambos os casos, Hel\_001\_20\_10 e Hel\_002\_100\_10 com dimensões  $D=20$  e  $D=100$ , respectivamente, uma população de  $Np=50$  pode ser adequada.

## 5. Considerações Finais

Neste artigo, análises do desempenho do algoritmo de ED com diferentes estratégias de ED e diferentes configurações de parâmetros foram apresentadas.

Neste estudo, dois diferentes tipos de análises foram realizadas para definição das melhores estratégias de ED, bem como dos melhores valores para a configuração dos parâmetros, para dois casos testes de diferentes dimensões. No primeiro tipo de análise, 70 diferentes combinações de estratégias de ED, com diferentes valores de  $F$  e  $Cr$ , para  $Np=50$ , foram avaliadas para um conjunto por meio de estatísticas fornecidas pelo método. No segundo tipo de análise, 25 novas combinações de estratégias de ED, com diferentes valores para  $Np$ ,  $F$  e  $Cr$ , o método de Taguchi foi aplicado para identificar as melhores configurações de parâmetros para os casos testados.



Para o Caso teste\_Hel\_001\_20\_10, através dos métodos das Estatísticas e de Taguchi, as melhores estratégias identificadas foram a 8 (ED/*rand-to-best/1/bin*) e a 9 (ED/*best/2/bin*), respectivamente, enquanto que as combinações de parâmetros foram  $F = 0,5$  e  $Cr = 0,5$  para o método das Estatísticas e  $F = 0,5$  e  $Cr = 0,5$  para o método de Taguchi. Para o Caso teste\_Hel\_002\_100\_10, através dos métodos das Estatísticas e de Taguchi, as melhores estratégias identificadas foram a 6 (ED/*best/1/bin*) e a 9 (ED/*best/2/bin*), respectivamente, enquanto que as combinações de parâmetros foram  $F = 0,2$  e  $Cr = 0,1$  para o método das Estatísticas e  $F = 0,8$  e  $Cr = 0,1$  para o método de Taguchi.

Sugere-se para trabalhos futuros analisar o desempenho das configurações, aqui identificadas como melhores, em um número maior de casos testes. Como não foram identificados as mesmas configurações de parâmetros para ambos os casos testes, que apresentam dimensões diferentes, sugere-se também para trabalhos futuros, a investigação de abordagens de ED adaptativas, uma vez que estas podem gerar melhores resultados uma vez que consideram, caso a caso, as melhores configurações de parâmetros.

## Referências

- Allahverdi, A. (2015) The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operation Research*, 246: 345-378.
- Ancău, M. (2012) On solving flowshop scheduling problems. *Proceedings of The Romanian Academy, Series A*, 13, 1: 71–79.
- Brownlee, J. (2011) *Clever Algorithms: Nature-Inspired Programming Recipes*. 1th edition. Clever Algorithms, Australia.
- Carvalho, L., Morais, M. F., Coelho, L. S., Rocha, R. P. e Beline, E. L. (2016) Evolução Diferencial: Características dos Métodos de Solução para a Programação da Produção em Ambientes Flow Shop Permutacional. In *Anais do XXXVI ENEGEP*, João Pessoa/PB. ABEPRO.
- Das, S. e Suganthan, P. N. (2011) Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15, 1: 4–31.
- Eiben, A. E. & Smith, J. E. (2015) *Introduction to Evolutionary Computing*. 2<sup>th</sup> edition. Springer-Verlag Berlin Heidelberg, Berlin, Alemanha.
- Fachin, J. M. (2014) Abordagem de evolução diferencial híbrida auto adaptativa e aplicação na realização de calibração automática de funções de softwares de motores automotivos. 2014. 261 f. Dissertação (Mestrado). Engenharia Elétrica, Universidade Federal do Paraná. Curitiba, PR.
- Gamperle, R., Muller, S. D. e Koumoutsakos, P. (2002) A parameter study for differential evolution, In: GRMELA, A.; MASTORAKIS, N. E. *Advances in intelligent systems, fuzzy systems, evolutionary computation*. WSEAS Press, Cambridge, Reino Unido.
- ICSI – International Computer Science Institute (2017). Differential Evolution (DE) for Continuous Function Optimization (an algorithm by Kenneth Price and Rainer Storn). Web page. <http://www1.icsi.berkeley.edu/~storn/code.html>. Acessado em 2017-02-02.
- Jarbouï, B., Siarry, P. e Teghem, J. (2013) *Metaheuristics for Production Scheduling*. ISTE Ltda., Londres, Reino Unido.
- Lin, Q., Gao, L., Li, X. e Zhang, C. (2015) A hybrid backtracking search algorithm for permutation flow-shop scheduling problem. *Computers & Industrial Engineering*, 85 437-446.
- Mokhtari, H., Abadi, I. N. K. e Cheraghali, A. (2011) A multi-objective flow shop scheduling with resource-dependent processing times: trade-off between Makespan and cost of resources. *International Journal of Production Research*, 49, 19: 5851–5875.



- Mozdgir, A., Mahdavi, I., Badeleh, I. S., & Solimanpur, M. (2013). Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing. *Mathematical and Computer Modelling*, 57, 1–2: 137–151.
- Onwubolu, G. C. e Davendra, D. (2009) *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*. Studies in Computational Intelligence, 175. Springer-Verlag Berlin Heidelberg, Berlin, Alemanha.
- OR Library - Operations Research Library (2017). Web page: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/flowshop1.txt>. Acessado em 2017-02-02.
- Price, K. V., Storn, R. M. e Lampinen, J. (2005) *A. Differential evolution - A practical approach to global optimization*. Springer-Verlag Berlin Heidelberg, Berlin, Alemanha.
- Ramos, A. e Tupia, M. (2013) A systematic for solving flow shop scheduling problem using differential evolution algorithm. *International Journal of Applied Science and Technology*, 03, 07: 64-74.
- Rönkkönen, J., Kukkonen, S. e Price, K. V. (2005) Real parameter optimization with differential evolution. In *Anais IEEE Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, Scotland.
- Silva, A. L. M. (2014) Algoritmo baseado em evolução diferencial para solução de problemas de otimização combinatória. 2014. 86 f. Dissertação (Mestrado). Engenharia Elétrica. Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, MG.
- Silva, R. C. P. (2010) Um estudo sobre a auto adaptação de parâmetros na evolução diferencial. 2010. 66f. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Ouro Preto, Ouro Preto.
- Simon, D. (2013) *Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence*. John Wiley & Sons, Inc., New Jersey, Estados Unidos da América.
- Storn, R. e Price, K. (1995) Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA.
- Storn, R. e Price, K. (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization*, 11: 341-359.
- Sun, Y., Zhang, C., Gao, L. e Wang, X. (2011) Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. *The International Journal of Advanced Manufacturing*, 55: 723-739.
- Taguchi, G. (1986). *Introduction to quality engineering*, White Plains: Asian Productivity Organization/UNIPUB.
- Taillard, E. (1993) Benchmarks for basic scheduling problems. *European Journal of Operations Research*, 64: 278-285.
- Tonge, V. G. e Kulkarni, P. S. (2012) Permutation flowshop scheduling problem using DE: A survey. *International Journal of Societal Applications of Computer Science*, 01, 01: 39-41.
- Tsai, J. T. (2015). Improved differential evolution algorithm for nonlinear programming and engineering design problems. *Neurocomputing*, 148: 628–640.