REVISTA INTERDISCIPLINAR DE PESQUISA EM ENGENHARIA

CILAMCE 2016

XXXVII IBERIAN LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

# PROBLEMA DE ALOCAÇÃO DE CONFIABILIDADE-REDUNDÂNCIA USANDO EVOLUÇÃO DIFERENCIAL AUTO-ADAPTATIVA

**Hellen Cristina Spengler**

**Gustavo Valentim Loch**

hspengler.mat@gmail.com

gustavo.gvalentim@gmail.com

Universidade Federal do Paraná

Campus III - Centro Politécnico - Av. Cel. Francisco H. dos Santos - 210 - Jardim das Américas - Curitiba, 81531-970, Paraná, Brasil

***Resumo.*** *Esse trabalho considera a otimização não-linear inteira-mista do problema de alocação de confiabilidade-redundância, determinando simultaneamente a confiabilidade e redundância dos componentes de um sistema em série. Grande número de métodos (exatos e meta-heurísticos) já abordaram esse problema, apresentando soluções satisfatórias. Nesse trabalho é utilizado um algoritmo de evolução diferencial auto-adaptativa (SaMDE) junto a um algoritmo de busca local. O exemplo numérico resolvido indica que o método proposto possui boa performance, embora ainda instável, comparado aos resultados anteriormente conhecidos.*

***Palavras-chave:*** *Otimização de confiabilidade, Redundância, Evolução diferencial, Auto-adaptação.*

# 1 INTRODUÇÃO

Com a evolução da tecnologia e a atenção contra falhas, o problema de aumento da confiabilidade de sistemas (série, série-paralelo p.ex.) torna-se importante. O design de tais sistemas influência diretamente em sua confiabilidade, tornando-os mais competitivos. Existem diversas abordagens para o problema, porém será tratado pela alocação de confiabilidade-redundância, em que se procura aumentar a confiabilidade dos componentes e usar componentes redundantes (Valian et al., 2013).

Nesse trabalho o problema será formulado pela maximização da confiabilidade do sistema, dada por uma função objetivo, e sujeito a múltiplas restrições não-lineares. Assim, pode-se entendê-lo por um problema de programação não-linear inteiro-misto (Hikita, 1992). Segue a sua forma:

$$
\begin{aligned}
&\textit{Maximize } R_s = f\left(\mathbf{r}, \mathbf{n}\right), \\
&\textit{sujeito a} \qquad g\left(\mathbf{r}, \mathbf{n}\right) \leq l \\
&0 \leq r_i \leq 1, \;\; r_i \in \mathbb{R}, \;\; n_i \in \mathbb{Z}^+, \;\; 1 \leq i \leq m.
\end{aligned}
$$

Em que $R_s$ é a confiabilidade do sistema, $g$ representa o conjunto de restrições (usualmente interpretadas como peso, volume e custo), $\mathbf{r} = (r_1, r_2, ..., r_m)$ o vetor da confiabilidade dos componentes do sistema e $\mathbf{n} = (n_1, n_2, ..., n_m)$ o vetor dos componentes redundantes do sistema. Logo, $r_i$ e $n_i$ são a confiabilidade e a redundância de cada i-ésimo subsistema, respectivamente. O vetor $l$ representa os recursos do problema e $m$ o número de subsistemas do sistema.

A seguir, será apresentada uma metaheurística de evolução diferencial com auto-adaptação (SaMDE) junto a um algoritmo simples de busca local, para resolução do exemplo de um sistema em série. Também será comparado com resultados anteriores na literatura.

# 2 EVOLUÇÃO DIFERENCIAL COM AUTO-ADAPTAÇÃO

O Algoritmo de Evolução Diferencial (DE), proposto em (Storn and Price, 1995), é baseado no processo de evolução natural. Embora traga tal estrutura, seus operadores não são baseados neste processo, tornando-se um simples e eficiente método de busca.

Porém, ainda que eficaz possui deficiências. Seja pelo número limitado de combinações do espaço de busca ou pelos parâmetros fixos, que levam a disperdiar a convergência no processo final. Além disso, a escolha de seus parâmetros influência diretamente no resultado obtido (Eiben et al, 1999). Portanto, a auto-adaptação dos parâmetros traz uma solução a estas deficiências. Diversos métodos já foram desenvolvidos utilizando estes conceitos, tais como o jDE, SaDE, DESAP e SaMDE (Prado, 2010).

## 2.1 Algoritmo SaMDE e busca local

O algoritmo SaMDE baseia-se na auto-adaptação dos parâmetros para múltiplas estratégias de mutação. Apresentando rápida convergência comparado ao algoritmo DE. Porém, por vezes torna-se prematura tal convergência, diminuindo sua capacidade de diversificação e estacionando em ótimos locais (Prado, 2010).

**Algorithm 1** Algoritmo SaMDE

$t \leftarrow 1$
Inicializar população $X_t = (x_{t,i}; i = 1, 2, ..., NP)$
Avaliar $X_t$
**while** condição de parada **do**
    **for** $i = 1 : NP$ **do**
        Selecione aleatoriamente $r_1, ..., r_5 \in 1, ..., NP$
        Selecione aleatoriamente $\delta_i \in 1, ..., n$
        Selecione aleatoriamente $F^{'} \in [0.8, 1]$
        **for** k=1:Número de estratégias **do**
            $V_i^k = V_{r_1}^k + F^{'}\left(V_{r_2}^k - V_{r_3}^k\right)$
        **end for**
        $w \leftarrow$ Estratégia escolhida pela roleta
        $F_i^w = F_{r_1}^w + F^{'}\left(F_{r_2}^w - F_{r_3}^w\right)$
        $CR_i^w = CR_{r_1}^w + F^{'}\left(CR_{r_2}^w - CR_{r_3}^w\right)$
        **for** $j = 1 : n$ **do**
            **if** $U_{[0,1]} \leq CR_i^w \vee j = \delta_i$ **then**
                **if** $w = 1$ **then**
                    $u_{t,i,j} \leftarrow x_{t,r_1,j} + F_i^w\left(x_{t,r_2,j} - x_{t,r_3,j}\right)$
                **else if** $w = 2$ **then**
                    $u_{t,i,j} \leftarrow x_{t,best,j} + F_i^w\left(x_{t,r_1,j} - x_{t,r_2,j}\right)$
                **else if** $w = 3$ **then**
                    $u_{t,i,j} \leftarrow x_{t,r_1,j} + F_i^w\left(x_{t,r_2,j} - x_{t,r_3,j}\right) + F_i^w\left(x_{t,r_4,j} - x_{t,r_5,j}\right)$
                **else if** $w = 4$ **then**
                    $u_{t,i,j} \leftarrow x_{t,i,j} + F_i^w\left(x_{t,r_1,j} - x_{t,i,j}\right) + F_i^w\left(x_{t,r_2,j} - x_{t,r_3,j}\right)$
                **end if**
            **else**
                $u_{t,i,j} \leftarrow x_{t,i,j}$
            **end if**
        **end for**
        **for** $i = 1 : NP$ **do**
            **if** $f\left(u_{t,i}\right) \leq f\left(x_{t,i}\right)$ **then**
                $x_{t+1,i} \leftarrow u_{t,i}$
            **else**
                $x_{t+1,i} \leftarrow x_{t,i}$
            **end if**
        **end for**
    **end for**
    $t \leftarrow t + 1$
**end while**

Em um segundo momento, é aplicado um pós-processamento utilizando uma busca local. Promovendo pequenas alterações nas posições reais da solução encontrada pela metaheurística. Caso seja factível, é aceita a nova solução e retorna-se a incrementar a confiabilidade dos componentes.

# 3 EXEMPLOS NUMÉRICOS

Alguns exemplos já são costumeiramente adotados na área (Chen, 2005). Portanto, traz-se uma breve aplicação ao sistema em série, que está descrito logo abaixo.

$$
\begin{aligned}
&\textit{Maximize } R_s = \prod_{1}^{m} R_i\left(n_i\right) \\
&\textit{sujeito a} \\
&g_1\left(\mathbf{r},\mathbf{n}\right) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 \leq V, \\
&g_2\left(\mathbf{r},\mathbf{n}\right) = \sum_{i=1}^{m} \alpha_i \left(-\frac{1000}{\ln r_i}\right)^{\beta_i} \left[n_i + \exp\left(\frac{n_i}{4}\right)\right] \leq C, \\
&g_3\left(\mathbf{r},\mathbf{n}\right) = \sum_{i=1}^{m} w_i n_i \exp\left(\frac{n_i}{4}\right) \leq W, \\
&0 \leq r_i \leq 1,\ r_i \in \mathbb{R},\ n_i \in \mathbb{Z}^+,\ 1 \leq i \leq m.
\end{aligned}
$$

A seguir, é apresentada a melhor solução encontrada pelo método proposto. Nota-se que em diversas vezes não se obteve tão bom desempenho. Tornado-se um método satisfatório mas instável.

**Table 1: Dados usados no sistema em série.**

| Estado | $10^5_{\alpha_i}$ | $\beta_i$ | $w_i v_i^2$ | $w_i$ | V | C | W |
|---|---|---|---|---|---|---|---|
| 1 | 2.330 | 1.5 | 1 | 7 | 110 | 175 | 200 |
| 2 | 1.450 | 1.5 | 2 | 8 | | | |
| 3 | 0.541 | 1.5 | 2 | 8 | | | |
| 4 | 8.050 | 1.5 | 4 | 6 | | | |
| 5 | 1.950 | 1.5 | 2 | 9 | | | |

Comparando com as soluções apresentadas na literatura, nota-se também que a abordagem traz uma solução tão quanto eficiente as anteriores.

**Table 2: Comparação do melhor resultado para o sistema em série com outros resultados apresentados na literatura.**

| Parâmetros | Hikita et al. | Chen | Valian et al. | Spengler et al. |
|---|---|---|---|---|
| $R_s$ | 0.931451 | 0.931678 | 0.931682387 | 0.9316823845 |
| $n$ | $(3, 2, 2, 3, 3)$ | $(3, 2, 2, 3, 3)$ | $(3, 2, 2, 3, 3)$ | $(3, 2, 2, 3, 3)$ |
| $r$ | 0.774887 | 0.779266 | 0.779416938 | 0.7793899301 |
| | 0.870065 | 0.872513 | 0.871833278 | 0.8718308464 |
| | 0.898549 | 0.902634 | 0.902885082 | 0.9028872228 |
| | 0.716524 | 0.710648 | 0.711393868 | 0.7113899926 |
| | 0.791368 | 0.788406 | 0.787803712 | 0.7878417664 |

# 4 CONCLUSÕES

A metaheurística apresentada, associada ao pós-processamento por uma busca local, possibilitou um bom desempenho, porém por muitas vezes instável. De forma que apenas o melhor resultado é apresentado. Devendo-se tal instabilidade a convergência ainda pré-matura da metaheurística. Desse modo, demonstrando um trabalho ainda a ser estudado e melhorado, seja pela de hibridação com outras heurísticas ou métodos exatos, que abranjam tais deficiências.

Além disso, ainda se faz necessária a validação para outros exemplos já considerados na literatura. Permitindo uma melhor comparação de desempenho.

## *AGRADECIMENTOS*

Agradecimentos ao apoio financeiro do Cnpq e paciência do prof. Gustavo V. Loch.

# REFERÊNCIAS

Chen, T., & You, P., 2005. Immune algorithms-based approach for redundant reliability problems with multiple component choices. *Computers in Industry*, n. 56, pp. 195–205.

Eiben, A. E., Michalewicz, Z., Schoenauer, M., & Smith, J. E., 1999. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 124–141.

Hikita, M., Nakagawa, Y., & Harihisa, H., 1992. Reliability optimization of systems by a surrogate constraints algorithm. *IEEE Trans. Reliab*, n. 41, pp. 473–480.

Prado, R. S., 2010. *Um estudo sobre a autoadaptação de parâmetros na evolução diferencial*. PhD thesis, Universidade Federal de Ouro Preto/Minas Gerais.

Storn, R., & Price, K., 1995. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report.*, vol. 17, n. 1, pp. 200–212.

Valian, E., Tavakoli, S., Mohanna, S. & Haghi, A., 2013. Improved cuckoo search for reliability optimization problems. *Comput. Ind. Eng*, n. 64, pp. 459–468.