

Heurística híbrida para o problema de alocação de confiabilidade-redundância de sistemas em série

Hellen Cristina Spengler
Universidade Federal do Paraná
PPGMNE
Curitiba, Brasil

Gustavo Valentim Loch
Universidade Federal do Paraná
PPGMNE
Curitiba, 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 heurísticos) já abordaram esse problema, apresentando soluções satisfatórias. Nesse trabalho é apresentada uma nova heurística híbrida, utilizando o algoritmo de evolução diferencial auto-adaptativa (SaMDE, do inglês Self adaptive Mutation Differential Evolution) e a otimização por enxame de partículas (PSO, do inglês particle swarm optimization). Os exemplos numéricos resolvidos indicam que o método proposto possui bom desempenho, comparado aos resultados anteriormente conhecidos na literatura.

Palavras-chave—Otimização de confiabilidade, Redundância, Evolução diferencial, Heurística híbrida.

I. INTRODUÇÃO

As aplicações de otimização, atualmente, se apresentam em grande número. Sendo diversas dessas provindas de processos na indústria. Normalmente possuem grande complexidade e múltiplos objetivos a serem abrangidos. Tais como a minimização de variáveis de custo, peso e volume, e a maximização da confiabilidade, rentabilidade e qualidade [1]. Dentre estes objetivos, destaca-se a confiabilidade.

Uma definição para a confiabilidade, foi proposta em 1952 pelo Advisory Group on the Reliability of Electronic Equipment: Confiabilidade indica a probabilidade de execução de desempenho em função de produtos específicos e conseguir atingir os objetivos dentro de um cronograma de tempo sob determinado ambiente.

Na produção, seja de produtos, sistemas ou estruturas, a confiabilidade influencia diretamente na rentabilidade e segurança. Projetos de itens confiáveis, levam a processos mais robustos e não dispendem intervenções futuras do fabricante, que causam vultosos custos. Na área de sistemas, a modelagem de seus designs está relacionada diretamente a confiabilidade apresentada pelo sistema. E também está relacionada diretamente as variáveis de custo, peso e volume. Tornando-se um problema de múltiplos objetivos. Em que se maximiza a confiabilidade pela função objetivo e minimiza o custo, peso e volume pelas funções de restrição [1].

Buscando pela maximização da confiabilidade desses sistemas, pode-se tomá-los por duas abordagens: acrescer a confiabilidade dos componentes do sistema; usar componentes redundantes nos subsistemas. A primeira abordagem, leva a poucas opções de solução e ao esgotamento da tecnologia perante a auto nível de confiabilidade exigido pelos componentes. Já a segunda abordagem, traz grandes acréscimos nas restrições, levando a infactibilidade.

Na tentativa de contornar tais limitações, unem-se ambas as abordagens. Podendo acrescer a confiabilidade dos componentes do sistema, e usar componentes redundantes nos subsistemas. Assim, tem-se elementos intercambiáveis que trazem novas formas factíveis de melhoria. Tal abordagem chama-se Problema de Alocação de Confiabilidade-Redundância [7].

Pela estrutura da abordagem tem-se que existem variáveis inteiras (componentes redundantes) e reais (confiabilidade dos componentes). Além disso a suas restrições, referentes a custo, peso e volume, são não lineares. Logo, pode-se entender por um problema de otimização não linear inteiro-misto. Sua formulação está a seguir.

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

Em que R_s é a confiabilidade do sistema, g representa o conjunto de restrições (usualmente interpretadas como peso, volume e custo), $\mathbf{r} = (r_1, r_2, \dots, r_m)$ é o vetor da confiabilidade dos componentes do sistema e $\mathbf{n} = (n_1, n_2, \dots, n_m)$ é 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.

Muitos trabalhos já buscaram resolver esse problema utilizando diversas abordagens: métodos exatos, algoritmos metaheurísticos e suas respectivas melhorias e combinações [5]. Assim, nesse trabalho é apresentada uma nova heurística híbrida, baseada na evolução diferencial com auto-adaptação (SaMDE) junto a otimização por enxame de partículas, para

resolução do exemplo de um sistema em série. Também será comparado com resultados anteriores da literatura .

II. EVOLUÇÃO DIFERENCIAL COM MUTAÇÃO AUTOADAPTATIVA - SAMDE

Os algoritmos evolutivos (AEs) são baseados no processo de evolução natural, em que a competição (por comida ou acasalamento) destaca e promove indivíduos geneticamente mais desenvolvidos. Assim, as novas gerações possuem maior probabilidade de receber características desses indivíduos. Dentre os algoritmos evolutivos, destaca-se o Algoritmo de Evolução Diferencial (DE), proposto por Storn e Price em 1995. Embora, traga a estrutura de um algoritmo evolucionário, seus operadores não são inspirados em um processo natural. Nem sendo baseado em uma modelagem de processos de busca naturais, como o algoritmo do morcego (BA) ou o algoritmo do sistema imune artificial (IA), tornando-se um simples e eficiente método de busca [6].

Entretanto, ainda que eficaz possua deficiências. Seja pelo número limitado de combinações do espaço de busca ou pelos parâmetros fixos, que levam a desperdiçar a convergência no processo final. Além disso, a escolha de seus parâmetros influencia diretamente no resultado obtido. Sendo sua busca manual um processo demasiadamente complicado, pois para cada problema adequa-se diferentes valores, que apenas especialistas com experiência poderiam descrevê-los [2].

Portanto, a auto-adaptação dos parâmetros traz uma solução a estas deficiências, proporcionando maior número de combinações do espaço de busca. E também facilitando na adaptação dos parâmetros a cada diferente problema, evitando o desperdício de tempo e trabalho nas atividades de determinar quais os melhores parâmetros para cada problema a ser tratado. Diversos métodos já foram desenvolvidos utilizando estes parâmetros como base, tais como o jDE, SaDE, DESAP e SamDE [6].

O algoritmo SamDE faz a auto-adaptação dos parâmetros para múltiplas estratégias de mutação. Permitindo o uso dessas estratégias em diferentes momentos, tornando-se um algoritmo que maximiza seu desempenho comparado aos algoritmos com uma única estratégia. No entanto, mesmo que apresente rápida convergência, por vezes torna-se prematura, diminuindo sua capacidade de diversificação, e tendo soluções com convergência para ótimos locais [6]. Desse modo, o processo de hibridação pretende trazer maior diversificação, e, por consequência, melhorar seu desempenho de convergência, permitindo uma busca em âmbito global.

III. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

O Algoritmo de otimização por enxame de partículas (PSO) tem uma abordagem populacional baseada na inteligência coletiva. De maneira diferente dos algoritmos evolutivos, tem-se uma abordagem baseada na estrutura de convívio em grupos, levando em consideração diversos aspectos [4].

Logo, a cada partícula é atribuída uma velocidade. Tal velocidade é utilizada para atualizar as posições, levando em

Algorithm 1 Evolução Diferencial com Mutaço Autoadaptativa - SamDE

```

1:  $t \leftarrow 1$ 
2: Inicializar população  $X_t = (x_{t,i}; i = 1, 2, \dots, NP)$ 
3: Avalia  $X_t$ 
4: while condição de parada do
5:   for  $i = 1 : NP$  do
6:     Selecione aleatoriamente  $r_1, \dots, r_5 \in 1, \dots, NP$ 
7:     Selecione aleatoriamente  $\delta_i \in 1, \dots, n$ 
8:     Selecione aleatoriamente  $F' \in [0.8, 1]$ 
9:     for  $k=1$ :Número de estratégias do
10:       $V_{r_1}^k = V_{r_1}^k + F' (V_{r_2}^k - V_{r_3}^k)$ 
11:    end for
12:     $w \leftarrow$  Estratégia escolhida pela roleta
13:     $F_i^w = F_{r_1}^w + F' (F_{r_2}^w - F_{r_3}^w)$ 
14:     $CR_{r_1}^w = CR_{r_1}^w + F' (CR_{r_2}^w - CR_{r_3}^w)$ 
15:    for  $j=1:n$  do
16:      if  $U_{[0,1]} \leq CR_{r_1}^w \vee j = \delta_i$  then
17:        if  $w = 1$  then
18:           $u_{t,i,j} \leftarrow x_{t,r_1,j} + F_i^w (x_{t,r_2,j} - x_{t,r_3,j})$ 
19:        else if  $w = 2$  then
20:           $u_{t,i,j} \leftarrow x_{t,best,j} + F_i^w (x_{t,r_1,j} - x_{t,r_2,j})$ 
21:        else if  $w = 3$  then
22:           $u_{t,i,j} \leftarrow x_{t,r_1,j} + F_i^w (x_{t,r_2,j} - x_{t,r_3,j}) + F_i^w (x_{t,r_4,j} - x_{t,r_5,j})$ 
23:        else if  $w = 4$  then
24:           $u_{t,i,j} \leftarrow x_{t,i,j} + F_i^w (x_{t,r_1,j} - x_{t,i,j}) + F_i^w (x_{t,r_2,j} - x_{t,r_3,j})$ 
25:        end if
26:      else
27:         $u_{t,i,j} \leftarrow x_{t,i,j}$ 
28:      end if
29:    end for
30:    for  $i = 1 : NP$  do
31:      if  $f(u_{t,i}) \leq f(x_{t,i})$  then
32:         $x_{t+1,i} \leftarrow u_{t,i}$ 
33:      else
34:         $x_{t+1,i} \leftarrow x_{t,i}$ 
35:      end if
36:    end for
37:  end for
38:   $t \leftarrow t + 1$ 
39: end while

```

conta o melhor valor da função aptidão encontrada pela partícula (conhecimento local) e o melhor valor da função aptidão encontrado pela população (conhecimento global). Sendo uma modelagem matemática de quanto a partícula confia em si mesma e quanto ela confia na população, para a tomada de decisão de que direção seguir. Proporcionando então uma ponderação dessas variáveis para obter um conhecimento prévio para o próximo passo. A seguir é descrito o procedimento:

$$v_{k+1}^i = wv_k^i + c_1r_1(p^i - x_k^i) + c_2r_2(p_k^s - x_k^i).$$

Em que v_{k+1}^i é a velocidade da i -ésima partícula na $k+1$ iteração, x_k^i é a posição da i -ésima partícula na k iteração, r_1 e r_2 são números aleatórios entre $[0; 1]$, p^i é a melhor posição encontrada pela i -ésima partícula, p_k^s é a melhor posição dentre todas as partículas na iteração k , w é o parâmetro de inércia no intervalo $[0, 4; 1, 4]$ e c_1 o parâmetro de confiança em si mesmo e c_2 de confiança na população [4]. Logo, em um seguinte momento tem-se a atualização das posições da população, que é dada por:

$$x_{k+1}^i = x_k^i + v_{k+1}^i.$$

Com esse procedimento torna-se possível uma exploração diferenciada pelo espaço de busca. Em que as partículas se aproximam parcialmente das melhores soluções encontradas, avaliando sua vizinhança com possíveis candidatos a nova solução com melhor incremento. Tal propriedade agrega uma opção de busca para uma heurística híbrida com o algoritmo SaMDE. Que viabiliza novos movimentos a população, e permite uma boa convergência.

IV. HEURÍSTICA HÍBRIDA

Utilizando o método de busca do algoritmo de otimização por enxame de partículas juntamente com o algoritmo SaMDE, traz-se a vantagem de percorrer o espaço de busca na vizinhança das melhores soluções encontradas. Tais vizinhanças tem possíveis candidatos a novas soluções com melhor aptidão. E também existindo indivíduos menos competitivos e que são uma opção para desacelerar a velocidade de convergência. O que se apresenta como uma característica eficiente, dado que o algoritmo SaMDE possui convergência prematura.

Assim, quando o algoritmo SaMDE escolhe aleatoriamente cinco indivíduos da população, que serão utilizados no processo de mutação, aplica-se o método de busca do algoritmo de otimização por enxame de partículas, por um dado número de vezes. Além disso, é construído certo número de vizinhos relacionados a cada um dos cinco indivíduos escolhido. Tais vizinhos são descartados posteriormente da busca terminada. O critério adotado para construção dos vizinhos é pela modificação de dois pontos, escolhidos aleatoriamente. Fazendo pequenas modificações, também aleatórias, mas restritas a sexta casa decimal. Levando em consideração apenas as posições reais das soluções, isto é, as r_i confiabilidades de cada componente dos subsistemas. Lembrando que cada solução é composta pelo vetor da confiabilidade dos componentes do sistema, $\mathbf{r} = (r_1, r_2, \dots, r_m)$, e pelo vetor dos componentes redundantes do sistema, $\mathbf{n} = (n_1, n_2, \dots, n_m)$.

Então, após as soluções serem acrescidas com o termo de velocidade, retomou-se a mutação original do algoritmo SaMDE. Por meio de experimentação, foram modificados os valores usuais para os parâmetros. Logo, foram tomados $w = 0,4$, $c_1 = 0,1$ e $c_2 = 0,15$. Representando que a partícula possui maior confiança na população do que em si mesma. E, portanto seu movimento tende preferencialmente em direção do ótimo global. Pesquisando seus entornos mais próximos.

V. EXEMPLO NUMÉRICO

Alguns exemplos já são costumeiramente adotados na área [7]. Portanto, traz-se uma breve aplicação ao sistema em série, que está descrito logo abaixo.

$$\begin{aligned} \text{Maximize } R_s &= \prod_1^m R_i(n_i) \\ \text{sujeito a} \\ g_1(\mathbf{r}, \mathbf{n}) &= \sum_{i=1}^m w_i v_i^2 n_i^2 \leq V, \\ g_2(\mathbf{r}, \mathbf{n}) &= \sum_{i=1}^m \alpha_i \left(-\frac{1000}{\ln r_i} \right)^{\beta_i} [n_i + \exp\left(\frac{n_i}{4}\right)] \leq C, \\ g_3(\mathbf{r}, \mathbf{n}) &= \sum_{i=1}^m w_i n_i \exp\left(\frac{n_i}{4}\right) \leq W, \\ 0 \leq r_i \leq 1, r_i &\in \mathbb{R}, n_i \in \mathbb{Z}^+, 1 \leq i \leq m. \end{aligned}$$

Em que m é o número de subsistemas do sistema, n_i é o número de componentes no subsistema i , r_i é a confiabilidade de cada componente no subsistema i , $R_i(n_i)$ é a confiabilidade de cada subsistema, R_s é a confiabilidade do sistema, g são as funções de restrição. As variáveis w_i , v_i e c_i são o peso, volume e custo de cada componente no subsistema i , respectivamente. As constantes V, C e W , são os limites superiores de volume, custo e peso das restrições do problema, respectivamente.

Os parâmetros utilizados para implementação dessas instâncias foram adotados de acordo com Hikita et al. [5], Chen [7], Valian et al. [3]. Nota-se que diversas outras literaturas também abordam estes problemas e suas resoluções. Sendo neste trabalho comparado a uma pequena parcela da literatura, tentando trazer os melhores resultados já apresentados.

Observa-se que existem diversos outros exemplos que também são usualmente adotados para comparação. Porém, para não delongar a escrita desse trabalho, optou-se por trazer apenas um exemplo para comparação. Enfim, os valores dos parâmetros estão apresentados nas tabelas a seguir.

Tabela I
DADOS USADOS NO SISTEMA EM SÉRIE

| Estado | $10^5 \alpha_i$ | β_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 | | | |

Logo, percebe-se que foram tomados cinco subsistemas para compor o sistema. Nota-se que não necessita ser fixamente esse valor adotado. E sujeito a três restrições não-lineares, que dizem respeito a variáveis de volume, custo e peso do problema. Quanto ao cálculo da função objetivo, foi feito relacionado a estrutura dos sistemas considerados. Tornando-se necessários novos cálculos para diferentes valores de subsistemas e estruturas do sistema.

Além disso, foram tomados os seguintes valores para os parâmetros relacionados com a heurística híbrida: tamanho da população = 30, vizinhos por indivíduo selecionado = 10 (passo híbrido com PSO), número máximo de gerações = 500.

Tabela II

COMPARAÇÃO DO MELHOR RESULTADO PARA O SISTEMA EM SÉRIE COM OUTROS RESULTADOS APRESENTADOS NA LITERATURA

| | Hikita et al. | Chen | Valian et al. | Proposta |
|-------|-----------------|-----------------|-----------------|-----------------|
| R_s | 0,931451 | 0,931678 | 0,931682387 | 0,931665055 |
| 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 | 0,7816224505 |
| | 0,870065 | 0,872513 | 0,871833278 | 0,8725123656 |
| | 0,898549 | 0,902634 | 0,902885082 | 0,9013686430 |
| | 0,716524 | 0,710648 | 0,711393868 | 0,7109117217 |
| | 0,791368 | 0,788406 | 0,787803712 | 0,7871177084 |

Observa-se que o resultado apresentado é a melhor solução encontrada após vinte repetições da heurística para o mesmo problema. Embora, observando a Tabela II o resultado encontra-se próximo dos valores obtidos por outros métodos, ou até superior a alguns, ainda não foi capaz de ultrapassar as melhores soluções conhecidas, como em Valian et al. [3]. Tendo então ainda a necessidade de um incremento a heurística, para torná-la além de eficiente também competitiva.

VI. CONCLUSÕES

O trabalho apresenta uma nova heurística híbrida para resolução do problema de alocação de confiabilidade-redundância para sistemas em série. Sendo um problema de programação não-linear inteiro-misto sujeito a múltiplas restrições não-lineares.

A heurística híbrida trouxe solução aos problemas de convergência pré-matura, do algoritmo de evolução diferencial com mutação auto-adaptativa (SaMDE). Possibilitando novos movimentos ao algoritmo no espaço de busca, que permitiu uma eficaz convergência.

Sendo comparadas as aplicações a um exemplo clássico da literatura, para o sistema em série. Mostrou soluções eficientes em relação as já conhecidas na literatura. Porém, ainda que eficiente, não se mostrou competitiva. E não alcançando valores de confiabilidade maior do que os melhores conhecidos.

Desse modo, faz-se necessária ainda novas abordagens de reestruturação da heurística. Sendo essa por um pós-processamento das soluções obtidas, relacionado a uma busca local capaz de atuar nas partes reais da solução (ou seja, nas confiabilidades dos subsistemas). Dado que sua parte inteira já se encontra estável, e também é de fácil determinação para as instâncias consideradas nos exemplos propostos.

Além disso, outras formas de hibridações também se fazem de interesse para estudos futuros. Utilizando a hibridação com métodos exatos sobre um espaço de busca restrito. Obtendo uma eficiente busca local para regiões contínuas restritas.

Por final, destaca-se ainda a necessidade de maiores testes para validação da estabilidade de convergência da heurística proposta. E também, para melhor determinação de seus parâmetros.

AGRADECIMENTOS

Agradecimentos a Capes, pelo apoio financeiro que permitiu o desenvolvimento desta pesquisa, e ao prof. Gustavo V. Loch pela dedicação e paciência.

REFERÊNCIAS

- [1] A. K. Dingra, *Optimal Apportionment of Reliability & Redundancy in Series Systems Under Multiple Objectives*. *IEEE Transactions on Reliability*. vol. 41, n. 4., pp. 576-582, 1992.
- [2] A. E. Eiben, Z. Michalewicz, M. Schoenauer and J. E. Smith. *Parameter control in evolutionary algorithms*. *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 124-141, 1999.
- [3] E. Valian, S. Tavakoli, S. Mohanna and A. Haghi, *Improved cuckoo search for reliability optimization problems*. *Comput. Ind. Eng.* n. 64, pp. 459-468, 2013.
- [4] L.S. Coelho *An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications*. *Reliability Engineering and System Safety*. n. 94, pp. 830-837, 2009.
- [5] M. Hikita, Y. Nakagawa and H. Harihsa, *Reliability optimization of systems by a surrogate constraints algorithm*. *IEEE Trans. Reliab.* n. 41, pp. 473-480, 1992.
- [6] R. S. Prado, *Um estudo sobre a autoadaptação de parâmetros na evolução diferencial*. PhD thesis, Universidade Federal de Ouro Preto/Minas Gerais, 2010.
- [7] T. Chen and P. You *Immune algorithms-based approach for redundant reliability problems with multiple component choices*. *Computers in Industry*. n. 56, pp. 195-205, 2013.