

# Otimização Estrutural com Algoritmo de Evolução Diferencial Auto-adaptativo (SaDE)



Yago Pereira dos Anjos Santos

*Programa de Pós-graduação em Modelagem Computacional*

*Universidade Federal de Juiz de Fora*

Juiz de Fora, Brasil

yago.pereira@estudante.ufjf.br

**Resumo**—Neste trabalho, inspirado em um algoritmo de evolução diferencial auto-adaptativo, são resolvidos problemas de otimização estruturais multi-objetivo, cujos objetivos se mostram conflitantes. O trabalho [12] resolve problemas de otimização estrutural através do algoritmo GDE3 com uma estratégia de penalização adaptativa APM acoplada, comparando os resultados obtidos com algoritmos clássicos bem conhecidos na literatura, como o NSGA-II e o próprio GDE3 com sua estratégia de tratamento de restrição nativo. Diante disso, o objetivo do presente trabalho será resolver alguns problemas estruturais aplicando a abordagem adaptativa SaDE proposta em [9], que elimina a necessidade de definir os parâmetros de controle ótimos CR (taxa de recombinação) e F (fator de mutação) para a obtenção das soluções não dominadas que constituem a fronteira de Pareto.

**Palavras-chave**—Evolução diferencial, Otimização estrutural multi-objetivo, Método de penalização adaptativo, SaDE

## I. INTRODUÇÃO

O presente trabalho tem como um de seus objetivos o estudo do problema comum em construção civil do projeto de estruturas de treliça. A depender do tipo de projeto, o engenheiro pode se deparar com uma demanda por uma estrutura que seja leve e rígida, isto é, usar o mínimo de material possível porém sem comprometer a integridade estrutural da construção, de modo que a mesma não entre em colapso. Desta forma, estamos diante de um clássico problema de otimização estrutural multi-objetivo, onde os objetivos em questão são conflitantes. Neste caso, deseja-se minimizar o peso de uma estrutura e, simultaneamente, o deslocamento máximo dos seus nós, isto é, há uma tensão limite que deve ser respeitada. Segundo [12], o algoritmo de evolução diferencial (DE) é uma meta-heurística popular para resolver-se problemas de otimização estrutural em espaços de busca contínuos. Isso se deve à sua simples implementação e à sua eficiência quando aplicado para a solução de problemas do mundo real. Além disso, existe também o NSGA-II (Non-Dominated Sorting Genetic Algorithm - II), proposto por [4],

a meta-heurística mais popular no contexto de otimização estrutural, de acordo com [13]. Este trabalho será focado na utilização do denominado GDE3 (the third generation step of Generalized Differential Evolution), proposto por [8], para a solução dos problemas estruturais. Segundo [13], o algoritmo GDE3 foi adotado em [11] para resolver alguns problemas de otimização estrutural multiobjetivos, a saber, os problemas de treliças com 10, 25, 60, 72 e 942 barras. Neste trabalho, serão resolvidos os problemas de treliças com 10, 25, 60 e 72 barras com o algoritmo GDE3 acoplado com o método APM (Adaptive Penalty Method) para tratamento de restrições, conforme [12]. O principal objetivo do trabalho será aplicar a técnica conhecida por SaDE, proposta em [9], para a adaptabilidade dos parâmetros de controle do algoritmo DE, a saber, a taxa de recombinação CR e o fator de mutação F, eliminando a necessidade de pre-determiná-los, uma vez que esses parâmetros são sensíveis ao tipo de problema tratado.

## II. DESCRIÇÃO E FORMULAÇÃO DO PROBLEMA

Conforme descrito na seção anterior, faz-se necessário um ponto de vista multi-objetivo para a solução dos problemas estruturais que serão tratados aqui. Desta maneira, possibilita-se ao tomador de decisões a escolha mais adequada de acordo com a preferência do projeto em questão. A formulação do problema estrutural resolvido aqui é descrito da seguinte forma: encontrar um conjunto de variáveis de decisão  $\mathbf{x} = (x_1, \dots, x_n)$ , correspondentes às áreas de seção transversal das barras ( $A_1, \dots, A_n$ ) da treliça, que minimiza simultaneamente o peso da estrutura e o deslocamento máximo de seus nós. Assim, temos a seguinte formulação:

$$\begin{aligned} \min_{s.t.} f_1(\mathbf{x}) &= \sum_{j=1}^n \rho A_j L_j & e & \quad f_2(\mathbf{x}) = \max(|u_{il}|) \\ & & & |s_{jl}| \leq s_{adm} \\ j &= 1, \dots, n & i &= 1, \dots, M \quad l = 1, \dots, N_L \end{aligned}$$

Para este trabalho os problemas serão resolvidos para o caso discreto, em que o vetor  $\mathbf{x} = (x_1, \dots, x_n)$  refere-se à áreas de seções transversais  $(A_1, \dots, A_n)$  da seguinte maneira: a  $j$ -ésima área de seção transversal é  $A_{[x_j]}$ , onde  $[x_j] = \text{round}(x_j)$ ,  $x \in [1, |AP|]^n$ ,  $j = 1, \dots, n$  é um índice para um elemento do conjunto de possíveis valores  $AP = \{A_1, \dots, A_{|AP|}\}$ , e  $|AP|$  é o número total de elementos em  $AP$ . Além disso,  $s_{jl}(\mathbf{x}) = E \varepsilon_{jl}(\mathbf{u}(\mathbf{x}))$  são os limites de estresse (restrições) e os parâmetros restantes são como a seguir:  $E$  é o módulo de Young;  $n$  é o número total de barras na estrutura de treliça;  $M$  é o número de graus de liberdade;  $N_L$  é o número de casos de carregamento aplicado à estrutura;  $L_j$  é o comprimento da  $j$ -ésima barra;  $\rho$  é a densidade do material;  $u_{il}$  é o deslocamento nodal do  $i$ -ésimo grau de liberdade;  $s_{jl}$  é o estresse da  $j$ -ésima barra, todos no  $l$ -ésimo caso de carregamento;  $s_{adm}$  é estresse permitido; e  $\varepsilon_{jl}(\mathbf{u}(\mathbf{x}))$  representa a mudança unitária no comprimento da barra. Conforme mencionado na introdução, serão resolvidos aqui os problemas de treliças de 10, 25, 60 e 72 barras.

### III. TRABALHOS RELACIONADOS

Embora o algoritmo GDE3 tenha sido desenvolvido com o seu próprio mecanismo de dominação restritiva para tratamento de restrições, Vargas et al. [11] propuseram o acoplamento da técnica de penalização adaptativa APM com o GDE3. O método APM, proposto por [2], não requer um parâmetro de penalidade pré-definido pelo usuário. Ele funciona por meio da utilização da informação da população, tais como média da função objetivo e o nível de violação de cada restrição durante a evolução. Vargas et al. [11] mostraram que o algoritmo proposto GDE3+APM performa melhor do que o algoritmo de colônia de formigas adotado em [1], sem no entanto investigar a contribuição da técnica APM quando comparada com a regra de tratamento de restrições originalmente usada no GDE3. Diante disso, Vargas et al. [12] compararam ambas as técnicas para tratamento de restrições constatando por meio de experimentos computacionais que o algoritmo GDE3+APM obteve melhores resultados para a maioria das métricas consideradas. Conforme apontado na literatura [10], [5], a performance do algoritmo original DE é altamente dependente não somente das estratégias de aprendizado adotadas mas também da configuração dos parâmetros de controle críticos, a saber, a taxa de recombinação  $CR$  e o fator de mutação  $F$ . Diante disso, [9] propõe o algoritmo SaDE (Self-adaptive Differential Evolution), onde os parâmetros supracitados,  $CR$  e  $F$ , não precisam ser pré-especificados. Logo, o algoritmo DE proposto visa automaticamente adaptar os parâmetros durante a evolução com base em artigos da literatura [6] e [7]. O algoritmo SaDE será usado neste trabalho para automatizar a obtenção dos parâmetros da taxa de recombinação e do fator de mutação do algoritmo GDE3.

### IV. MÉTODOS UTILIZADOS

No contexto de otimização multiobjetivo, o resultado final é um conjunto de soluções ditas não-dominadas, conceito que será melhor explicado pelo conceito de dominância de Pareto.

Sejam  $\mathbf{x}, \mathbf{y} \in D \subset \mathbb{R}^n$ . Dizemos que um vetor  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  domina outro vetor  $\mathbf{y} = (y_1, y_2, \dots, y_n)$  no sentido de Pareto, denotado por  $\mathbf{x} \succ \mathbf{y}$ , se e somente se, para todo  $i = 1, \dots, P$ ,  $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$  e existe  $1 \leq j \leq P$  tal que  $f_j(\mathbf{x}) < f_j(\mathbf{y})$ . Desta forma, dadas duas soluções  $\mathbf{x}, \mathbf{y} \in D$ , uma e apenas uma das seguintes condições é válida:  $\mathbf{x} \succ \mathbf{y}$ ,  $\mathbf{y} \succ \mathbf{x}$ , ou  $\mathbf{x}$  e  $\mathbf{y}$  são soluções não-dominadas. Uma solução  $\mathbf{x} \in D$  é dita Pareto ótima se e somente se, não existe solução  $\mathbf{y} \in D$  tal que  $\mathbf{y} \succ \mathbf{x}$ . O verdadeiro conjunto ótimo de Pareto ( $PS_{true}$ ) é definido como o conjunto de todas as soluções Pareto ótimas, e o conjunto de todos os valores de funções objetivo correspondentes às soluções em  $PS_{true}$  é chamado de frente de Pareto ótima ( $PF_{true}$ ). Diante disso, no contexto de meta-heurística multiobjetivo, a saída do problema é considerada como um conjunto de soluções mutuamente não-dominadas que aproxima  $PS_{true}$ . Este é o denominado conjunto de Pareto ótimo obtido ( $PS_{obtained}$ ) e o conjunto das imagens das soluções em  $PS_{obtained}$  é a frente de Pareto ótima obtida ( $PF_{obtained}$ ).

#### A. O Algoritmo de Evolução Diferencial Generalizado Acooplado com um Método de Penalização Adaptativa (GDE3 + APM)

O esquema de penalização adaptativa (APM) acoplada ao algoritmo de evolução diferencial generalizado utilizado aqui é inspirado no artigo [12]. O algoritmo GDE3 gera aleatoriamente uma população inicial e a evolui através de seleção e os operadores de mutação e recombinação, que utilizam um fator de mutação ( $F \in \mathbb{R}$ ) e uma taxa de recombinação  $CR \in [0, 1]$  além de um parâmetro  $N$  para o tamanho da população.

Seja  $P_G$  uma população de  $N$  vetores de decisão  $\mathbf{x}_{i,G}$  na geração  $G$ , onde  $i \in \{1, \dots, N\}$  é o índice do vetor. Cada  $\mathbf{x}_{i,G}$  da população na geração  $G$  é um vetor  $n$ -dimensional, e  $\mathbf{x}_{j,i,G}$  são suas  $j$ -ésimas coordenadas, com  $j \in \{1, \dots, n\}$ . Então, a partir de um vetor decisão  $\mathbf{x}_{i,G}$  cria-se um vetor teste (trial)  $\mathbf{u}_{i,G}$  através das operações de mutação e recombinação conforme mostrado no Algoritmo 1 [12]. Neste algoritmo, três vetores  $\mathbf{x}_{r_1,G}$ ,  $\mathbf{x}_{r_2,G}$  e  $\mathbf{x}_{r_3,G}$  são aleatoriamente selecionados na população, são mutuamente diferentes e diferentes de  $\mathbf{x}_{i,G}$ .

Após as operações de mutação e recombinação, o vetor teste  $\mathbf{u}_{i,G}$  é comparado com o vetor decisão  $\mathbf{x}_{i,G}$  usando o conceito de dominação por restrição. Um vetor  $\mathbf{x}$  domina um vetor  $\mathbf{y}$  no espaço de restrição (fato denotado por  $\mathbf{x} \succeq_c \mathbf{y}$ ) se uma, e somente uma, das seguintes condições são verdadeiras:

- 1) Ambos são inactiváveis e  $\mathbf{x} \succ \mathbf{y}$  no espaço de funções de violação de restrições.
- 2)  $\mathbf{x}$  é factível e  $\mathbf{y}$  é inactivável.
- 3)  $\mathbf{x}$  e  $\mathbf{y}$  são factíveis e  $\mathbf{x} \succ \mathbf{y}$  no espaço de funções objetivo.

Consequentemente, o vetor teste  $\mathbf{u}_{i,G}$  é selecionado de modo a substituir o vetor decisão  $\mathbf{x}_{i,G}$  para a próxima geração  $P_{G+1}$  se  $\mathbf{u}_{i,G} \succeq_c \mathbf{x}_{i,G}$ . Caso  $\mathbf{x}_{i,G} \succeq_c \mathbf{u}_{i,G}$ ,  $\mathbf{u}_{i,G}$  é descartado e  $\mathbf{x}_{i,G}$  é mantido na população, do contrário, ambos são incluídos em  $P_{G+1}$ . Além disso, o tamanho da população é reduzido usando um método de elitismo quando o tamanho  $P_{G+1}$  é maior do que  $N$ . Este método é baseado no ranking não-dominado (Non-dominated Ranking) e na distância

---

**Algoritmo 1:** Operações de Mutação e Recombinação no DE.

---

```
1 Randomly select  $J \in \{1, \dots, n\}$ 
2 for  $j = 1$  to  $n$  do
3   Randomly select  $rand \in [0, 1]$ 
4   if  $rand < CR$  or  $j = J$  then
5      $\mathbf{u}_{j,i,G} \leftarrow \mathbf{x}_{j,r_1,G} + F(\mathbf{x}_{j,r_2,G} - \mathbf{x}_{j,r_3,G})$ 
6   end
7   else
8      $\mathbf{u}_{j,i,G} \leftarrow \mathbf{x}_{j,r_1,G}$ 
9   end
10 end
```

---

de aglomeração (Crowding Distance), dois mecanismos bem conhecidos apresentados no algoritmo NSGA-II.

O método de penalização adaptativo (APM) tem por objetivo a configuração automática dos coeficientes de penalização para a função de penalização usada para avaliar soluções candidatas. Essa configuração automática é feita usando feedback do processo de busca. A ideia é observar a violação de cada restrição e definir um coeficiente mais alto para aquelas restrições que parecem ser as mais difíceis. As quantidades incluídas na definição dos valores de penalidade são a média dos valores da função objetivo das soluções e a média da violação da  $j$ -ésima restrição, ambos para um dado conjunto de soluções candidatas. Diante disso, considerando cada objetivo separadamente e sua minimização sujeita a  $m$  restrições, a aptidão  $f_{fitness}(\mathbf{x})$  de um dado indivíduo pode ser escrita como:

$$f_{fitness}(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } \mathbf{x} \text{ é factível,} \\ \bar{f}(\mathbf{x}) + \sum_{j=1}^J k_j \nu_j(\mathbf{x}), & \text{caso contrário,} \end{cases}$$

onde

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle, \\ \langle f(\mathbf{x}) \rangle, & \text{caso contrário,} \end{cases}$$

e  $\langle f(\mathbf{x}) \rangle$  é a média dos valores da função objetivo das soluções na população atual. O coeficiente de penalidade  $k_j$ , correspondente à  $j$ -ésima restrição é definido a cada geração por:

$$k_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle \nu_j(\mathbf{x}) \rangle}{\sum_{l=1}^J [\langle \nu_l(\mathbf{x}) \rangle]^2},$$

onde  $\nu_j(\mathbf{x})$  é a violação da  $j$ -ésima restrição tomando a média da população atual.

Enfim, o algoritmo GDE3+APM acopla o algoritmo GDE3 com a técnica de tratamento de restrições APM, isto é, ele substitui as funções objetivo  $f_1, f_2, \dots, f_n$  no GDE3 pelos valores de função objetivo  $f_{i,fitness}$ ,  $i = 1, \dots, n$ , penalizados pela técnica APM. Em outras palavras, o GDE3+APM funciona assim como o GDE3 com uma população de soluções todas factíveis. Além disso, devido à natureza estocástica envolvida na técnica em questão, um arquivo externo é usado para armazenar as soluções não-dominadas encontradas durante o processo de busca. Caso contrário, não se tem

a garantia de que as soluções desejáveis permaneçam na população geracional [3]. Uma representação esquemática do algoritmo GDE3+APM pode ser encontrada em [12] em que o Algoritmo 1 é utilizado.

### B. Self-adaptive Differential Evolution (SaDE)

O algoritmo Self-adaptive Differential Evolution (SaDE) aborda o desafio do ajuste manual dos parâmetros de mutação  $F$  e taxa de recombinação  $CR$  no DE tradicional. Em vez de usar valores fixos, o SaDE adapta automaticamente esses parâmetros durante o processo de otimização com base em seu desempenho histórico.

O SaDE mantém dois arrays de memória  $M_{CR}$  e  $M_F$  de tamanho  $H$  (tipicamente  $H = 5$ ) que armazenam valores de parâmetros bem-sucedidos. Desta forma, para cada indivíduo na população:  $CR_i$  é gerado a partir de uma distribuição normal:  $CR_i \sim \mathcal{N}(M_{CR}[k], 0, 1)$ , truncado para  $[CR_{min}, CR_{max}]$ ;  $F_i$  é gerado a partir de uma distribuição de Cauchy:  $F_i \sim \text{Cauchy}(M_F[k], 0, 1)$ , truncado para  $[F_{min}, F_{max}]$ ; o índice  $k$  é selecionado aleatoriamente de  $\{1, 2, \dots, H\}$ .

A adaptação ocorre a cada  $LP$  gerações (período de aprendizado, tipicamente 50 gerações). Os valores bem-sucedidos de  $CR$  são calculados pela média:  $M_{CR}^{novo} = \text{média}(CR_{bem-sucedidos})$ . Os valores bem-sucedidos de  $F$  usam a média de Lehmer:  $M_F^{novo} = \sum F^2 / \sum F$ , favorecendo valores bem-sucedidos maiores. As atualizações de memória usam substituição circular (round-robin) para manter a diversidade.

Na estrutura GDE3+SaDE+APM implementada, cada vetor tentativa (trial) usa valores  $CR_i$  e  $F_i$  gerados individualmente. O sucesso dos parâmetros é rastreado com base na seleção na comparação de aptidão APM. O arquivo externo preserva soluções com seus parâmetros associados. A adaptação continua ao longo de toda a otimização, permitindo a exploração dinâmica do espaço de busca.

Desta maneira, o mecanismo SaDE elimina a necessidade de ajuste manual de parâmetros enquanto mantém um desempenho robusto em diversos problemas de otimização de treliças, desde estruturas simples de 10 barras até estruturas espaciais complexas de 72 barras.

## V. EXPERIMENTOS E RESULTADOS

Conforme mencionado anteriormente, os problemas que serão resolvidos aqui são as estruturas de treliças de 10, 25, 60 e 72 barras, conforme [12]. Para o algoritmo GDE3+APM foram usados os parâmetros para a taxa de recombinação CR e fator de mutação F conforme o artigo [12] e podem ser conferidos na Tabela I juntamente com o tamanho da população *Pop* e o número de gerações *Gen*.

Para a versão SaDE do algoritmo inicialmente foram adotados  $CR = 0.5$  e  $F = 0.5$  para todas as instâncias tratadas aqui, e os parâmetros se ajustam ao longo das gerações. Além disso, para este caso o tamanho da população é igual a 50 e o número de gerações é igual a 1000 assim como no caso anterior.

Tabela I  
PARÂMETROS GDE3+APM

Problem	Pop	Gen	CR	F
10-bar	50	1000	0.9	0.4
25-bar	50	1000	0.9	0.2
60-bar	50	1000	0.1	0.7
72-bar	50	1000	0.9	0.3

### A. A treliça de 10 barras

Para o problema da estrutura de treliça de 10 barras, o material possui densidade  $\rho = 0.11b/in^3$  e o módulo de Young é  $E = 10^4 ksi$ . Cargas verticais de  $100kips$  são aplicadas nos nós 2 e 4 e o estresse em cada barra é limitado por  $\pm 25ksi$ . Os valores para as áreas de seção transversal ( $in^2$ ) são escolhidos do conjunto: 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, e 33.50; resultando em 42 opções. A Figura 1 mostra uma representação esquemática para este problema.

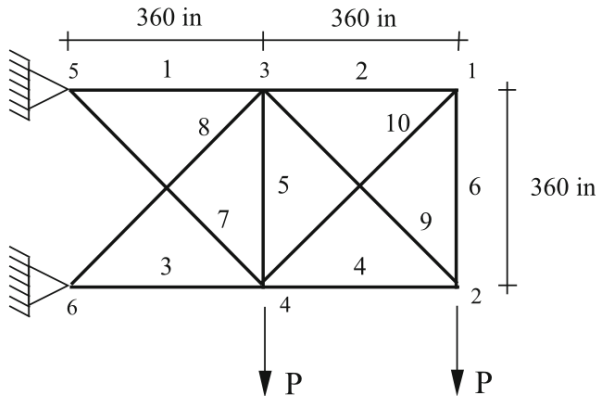


Figura 1. Estrutura da treliça de 10 barras adaptado de [12].

A Figura 2 mostra as soluções não dominadas obtidas pelos algoritmos GDE3+APM e SaDE.

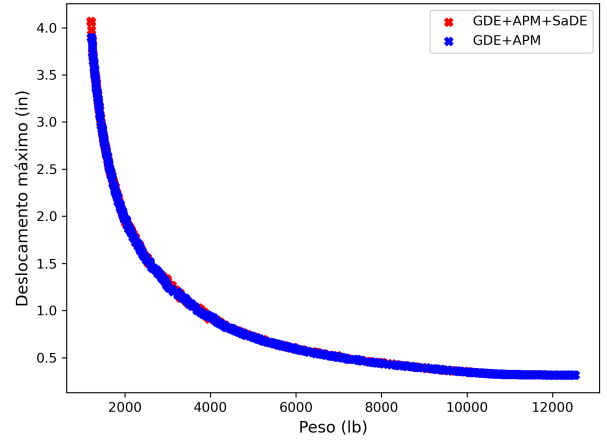


Figura 2. Frente de Pareto obtida para a treliça de 10 barras simulados pelo GDE3+APM e SaDE.

### B. A treliça de 25 barras

Na treliça de 25 barras, o material tem densidade  $\rho = 0.11b/in^3$  e o módulo de elasticidade é  $E = 10^4 ksi$ . A tensão em cada membro é limitada por  $\pm 40ksi$ . Os valores das áreas das seções transversais ( $in^2$ ) são escolhidas do conjunto: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4; resultando em 30 opções. Os dados de carregamento e agrupamento de membros para as variáveis são conforme o artigo [12]. A Figura 3 é uma representação esquemática para esta estrutura.

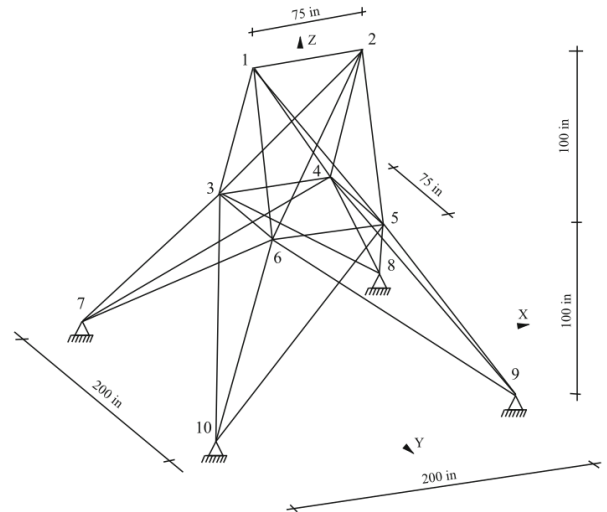


Figura 3. Estrutura da treliça de 25 barras adaptada de [12].

A Figura 4 mostra as soluções não dominadas obtidas pelo algoritmo GDE3+APM e pelo algoritmo adaptativo SaDE.

### C. A treliça de 60 barras

Para o problema da treliça de 60 barras é uma estrutura de anel, conforme pode ser visto na Figura 5, o material possui

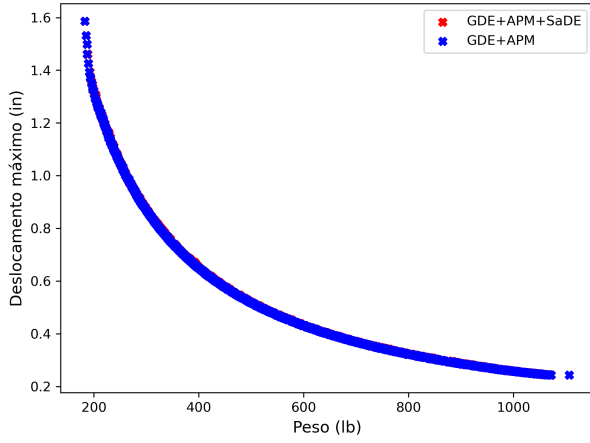


Figura 4. Frente de Pareto obtido para a treliça de 25 barras simulados pelo GDE3+APM e SaDE.

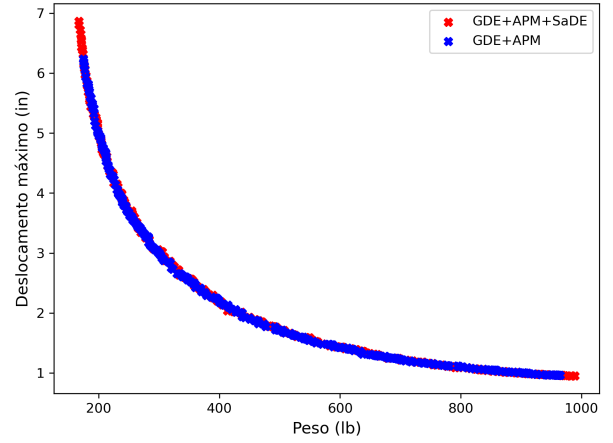


Figura 6. Frente de Pareto obtida para a estrutura de treliça de 60 barras simulados pelo GDE3+APM e SaDE.

densidade  $\rho = 0.1lb/in^3$  e o módulo de Young  $E = 10^4ksi$ . O raio externo do anel é  $100in$  e o raio interno é  $90in$ . A tensão em cada barra é limitada em  $60ksi$ . Os valores das áreas das seções transversais são escolhidas do conjunto: 0.5, 0.6, 0.7, ..., 4.7, 4.8, 4.9; resultando em 45 opções. Os dados de carregamento e agrupamento dos membros para este problema são conforme referência [12].

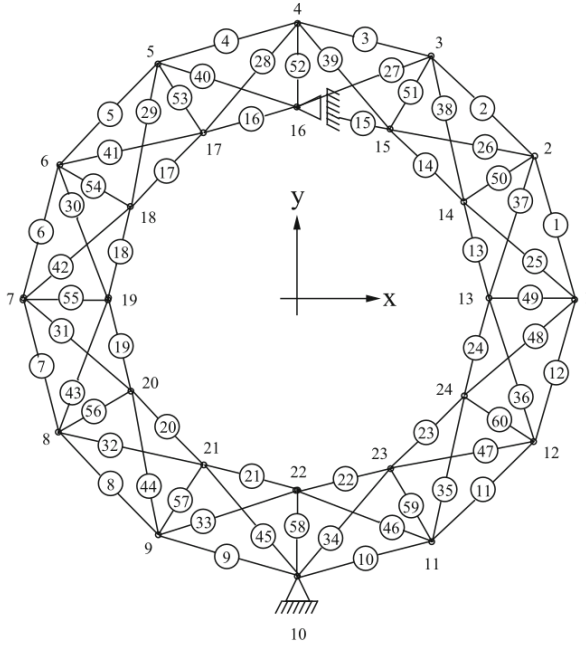


Figura 5. Estrutura da treliça de 60 barras adaptado de [12].

Na Figura 6 são apresentadas as soluções não dominadas obtidas para a treliça de 60 barras com o algoritmo GDE3+APM e com o algoritmo adaptativo SaDE.

#### D. A treliça de 72 barras

Para a estrutura da treliça de 72 barras também temos  $\rho = 0.1lb/in^3$  e  $E = 10^4ksi$ . A tensão em cada membro

é limitada por  $\pm 25ksi$ . E os valores das áreas das seções transversais são escolhidos do conjunto: 0.1, 0.2, 0.3, ..., 2.3, 2.4, 2.5; resultando em 25 opções. Os dados de carregamento e o agrupamento dos membros são conforme referência [12].

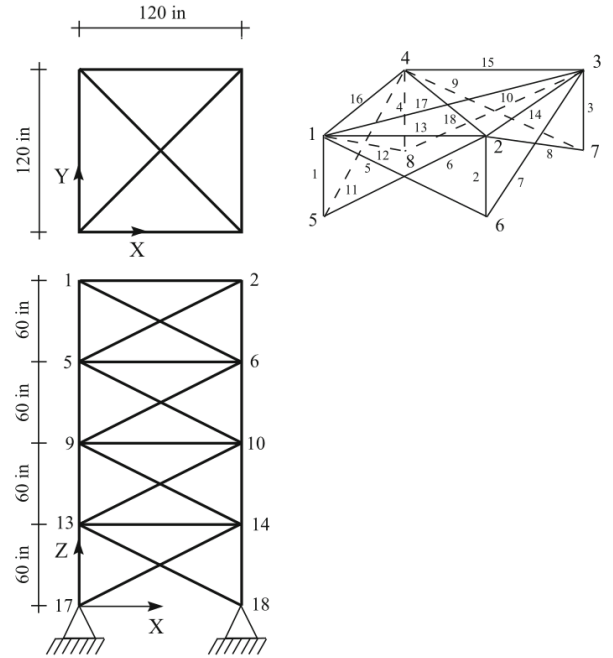


Figura 7. Estrutura da treliça de 72 barras adaptado de [12].

A Figura 8 mostra os resultados obtidos, isto é, as soluções não dominadas obtidas por ambos os algoritmos, GDE3+APM e SaDE.

## VI. CONCLUSÕES

A partir dos resultados obtidos pelos experimentos computacionais, vistos na seção anterior, constata-se que as frentes de Pareto para o caso do algoritmo GDE3+APM ficam próximos dos observados na literatura, vide [12]. Apesar de não se

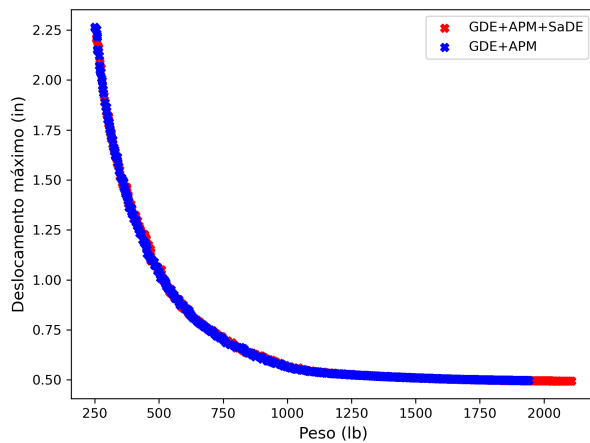


Figura 8. Frente de Pareto obtida para a treliça de 72 barras simulados com GDE3+APM e SaDE.

conhecer a semente utilizada para realizar os experimentos observados na literatura, foram utilizados os mesmos parâmetros para CR, a taxa de recombinação, e F, o fato de mutação, para a produção dos experimentos no caso GDE3+APM. Além disso, o algoritmo adaptativo SaDE, baseado nas ideias de [9], mostrou-se eficiente na obtenção das soluções não dominadas para cada um dos problemas tratados, obtendo fronteiras de Pareto próximas das obtidas para os Parâmetros ótimos do caso do algoritmo GDE3+APM.

## REFERÊNCIAS

- [1] ANGELO, J. S., BERNARDINO, H., AND BARBOSA, H. Multi-objective ant colony approaches for structural optimization problems. In *Proceedings of the eleventh international conference on computational structures technology* (2012), vol. 99.
- [2] BARBOSA, H. J., AND LEMONGE, A. C. An adaptive penalty scheme in genetic algorithms for constrained optimization problems. In *Proceedings of the 4th annual conference on genetic and evolutionary computation* (2002), pp. 287–294.
- [3] COELLO, C. A. C., LAMONT, G. B., AND VELDHIJZEN, D. A. V. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.
- [4] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
- [5] GAMPERLE, R. Parameter study for differential evolution. *Proceedings of the WSEAS NNA-FSFS-EC 2002* (2002), 293.
- [6] GÓMEZ, J., DASGUPTA, D., AND GONZÁLEZ, F. Using adaptive operators in genetic search. In *Genetic and evolutionary computation conference* (2003), Springer, pp. 1580–1581.
- [7] JULSTROM, B. A. What have you done for me lately? adapting operator probabilities in a steady-state genetic algorithm. In *Proceedings of the 6th International Conference on Genetic Algorithms* (1995), pp. 81–87.
- [8] KUKKONEN, S., AND LAMPINEN, J. Gde3: The third evolution step of generalized differential evolution. In *2005 IEEE congress on evolutionary computation* (2005), vol. 1, IEEE, pp. 443–450.
- [9] QIN, A. K., AND SUGANTHAN, P. N. Self-adaptive differential evolution algorithm for numerical optimization. In *2005 IEEE congress on evolutionary computation* (2005), vol. 2, IEEE, pp. 1785–1791.
- [10] ROGALSKY, T., KOCABIYIK, S., AND DERKSEN, R. Differential evolution in aerodynamic optimization. *Canadian Aeronautics and Space Journal* 46, 4 (2000), 183–190.
- [11] VARGAS, D., LEMONGE, A., BARBOSA, H., AND BERNARDINO, H. An algorithm based on differential evolution for structural multiobjective optimization problems with constraints (in portuguese). *Rev Int Métodos Num Cálculo Eng* 32, 2 (2016), 91–99.
- [12] VARGAS, D. E., LEMONGE, A. C., BARBOSA, H. J., AND BERNARDINO, H. S. Differential evolution with the adaptive penalty method for structural multi-objective optimization. *Optimization and Engineering* 20, 1 (2019), 65–88.
- [13] ZAVALA, G. R., NEBRO, A. J., LUNA, F., AND COELLO COELLO, C. A. A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization* 49, 4 (2014), 537–558.