

Técnicas de Evolução Diferencial com Conjuntos Embaralhados

Autores:

Geovana Trindade da Silva Oliveira
Milena Almeida Leite Brandão

Orientadora:

Sezimária de Fátima Pereira Saramago

Resumo: Neste trabalho são apresentados alguns métodos de otimização estocásticos tais como, Evolução Diferencial, Evolução com Conjuntos Embaralhados e Evolução com Conjuntos Competitivos. Como estes algoritmos de otimização foram desenvolvidos para resolver problemas irrestritos, considerando apenas os limites laterais das variáveis, torna-se necessário utilizar algum artifício para que os problemas com restrições também se apliquem aos métodos de otimização irrestrita. Por isso, é dada uma breve descrição do Método de Penalidade, o qual transforma um problema restrito em um problema irrestrito equivalente. Para ilustração, três problemas práticos de otimização são resolvidos usando cada um dos métodos abordados. É feita uma análise dos resultados obtidos e são apresentadas comparações entre os métodos.

Palavras-chave: Evolução Diferencial, Evolução com Conjuntos Embaralhados, Evolução com Conjuntos Competitivos, Método de Penalidade.

1. Introdução

Segundo Pirsig (1974) a “Otimização matemática é o título formal dada ao ramo da ciência computacional, que procura responder à pergunta Qual é o melhor? para os problemas em que a qualidade de qualquer resposta pode ser expressa como um valor numérico. Tais problemas surgem em todas as áreas da matemática, das ciências físicas, químicas e biológicas, engenharia, arquitetura, economia e gestão, e a gama de técnicas disponíveis para resolvê-los é quase tão grande quanto”.

A otimização é uma ferramenta importante para tomada de decisão durante a análise e o projeto de sistemas físicos. Após a formulação do modelo, um algoritmo de otimização é utilizado para obter a solução do problema, usualmente com a ajuda de códigos computacionais. Não existe um algoritmo universal, mas um conjunto de métodos, entre os quais alguns são mais apropriados para determinadas aplicações específicas. A escolha do método de otimização depende do usuário e pode determinar a eficácia ou falha para a solução do problema.

Existem diversos métodos de otimização e desta forma vários autores já apresentaram diferentes propostas para classificá-los, não sendo possível escolher uma proposta universalmente aceita. Os métodos dependem do tipo do problema em estudo (programação linear ou não-linear, problemas restritos ou irrestritos), das variáveis de projeto (contínuas ou discretas), do tipo de resposta desejada (otimização local ou global; projeto ótimo ou controle ótimo), das técnicas empregadas (determinísticas, estocástica ou híbrida). Além disso, os problemas podem ser classificados segundo o número de variáveis (pequenos ou grandes) ou segundo a suavidade das funções (diferenciáveis ou não-diferenciáveis).

Na otimização estocástica pode-se trabalhar com “incertezas” de algumas variáveis e produzir soluções que otimizam a “performance esperada” do modelo. Pode-se garantir que as variáveis x satisfaçam às restrições para alguma probabilidade especificada anteriormente (modelos probabilísticos). Estes métodos também são conhecidos como métodos heurísticos. O termo heurístico tem sua origem na palavra grega “heuriskein” que significa “encontrar ou descobrir”. Estes métodos utilizam apenas informações da função a ser otimizada, que pode ser de difícil representação gráfica, não-diferenciável, descontínua, não-linear, multimodal. Alguns exemplos dos métodos heurísticos são os métodos naturais, que tentam simular os processos usados na natureza para resolver problemas

complexos. Entre estes métodos pode-se citar: Busca Tabu, *Simulated Annealing* e métodos baseados em população (Algoritmos Genéticos, Evolução Diferencial, Otimização por Colônia de Formigas, Otimização por Bando de Pássaros, etc). Estas técnicas trabalham com um conjunto de pontos, necessitam de muitas avaliações da função objetivo e sua maior desvantagem é o alto custo computacional.

Neste estudo serão apresentados alguns métodos de otimização estocástica, a saber, Evolução Diferencial, Evolução com Conjuntos Embaralhados, Evolução com Conjuntos Competitivos e Evolução Diferencial Melhorada.

A Evolução Diferencial (ED) foi criada após Ken Price tentar usá-la para resolver o problema do polinômio de Chebychev o qual foi apresentado por Rainer Storn. A descoberta aconteceu quando Price teve a idéia de utilizar diferentes vetores para recriar o vetor população criando uma abordagem diferente das encontradas nas estratégias de evolução que tratam de problemas de otimização. Desde então foram feitos vários testes e aperfeiçoamentos os quais tornaram o algoritmo da ED versátil e robusto.

O algoritmo é iniciado criando uma população inicial de N_p indivíduos, escolhida aleatoriamente e devendo cobrir todo o espaço de busca. Geralmente, é criada por uma distribuição de probabilidade uniforme, quando não há nenhum conhecimento sobre o problema.

Cada indivíduo, chamado de *vetor*, possui n componentes representadas por valores reais, sendo n o número de variáveis de projeto. Assim, a população segue uma evolução natural, em que o número de indivíduos é, geralmente, constante durante todas as gerações.

A idéia principal da evolução diferencial é gerar novos indivíduos, denotados vetores modificados ou doadores, pela adição da diferença ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Esta operação é chamada *mutação*.

As componentes do indivíduo doador são misturadas com as componentes de um indivíduo escolhido aleatoriamente (denotado vetor alvo), para resultar o chamado vetor tentativa, ou vetor experimental. Tal processo é referido como *cruzamento*. Se o vetor experimental resultar um valor da função objetivo menor que o vetor alvo, então o vetor experimental substitui o vetor alvo na geração seguinte. Esta última operação é chamada *seleção*. O procedimento é finalizado quando algum critério de parada é obedecido.

Apesar de vários aspectos positivos, tem-se observado que a Evolução Diferencial (ED) às vezes não apresenta uma performance tão boa quanto se espera. A análise empírica da ED tem mostrado que o algoritmo pode deixar de prosseguir rumo a um ótimo global tendendo a um estado de estagnação, no qual os indivíduos ficam muito parecidos entre si, isto é, a população fica homogênea e o algoritmo não apresenta qualquer melhora, apesar de aceitar novos indivíduos na população. Além disso, ED também sofre com o problema da convergência prematura. Esta situação surge quando há uma perda da diversidade da população. Como resultado, a população converge para um ponto que pode não ser uma solução ótima local. Isso geralmente ocorre quando a função objetivo é multi modal, tendo vários ótimos locais e global.

Assim como outros algoritmos evolutivos, o desempenho da ED se deteriora com o aumento da dimensionalidade da função objetivo. Várias modificações foram feitas na estrutura deste algoritmo para melhorar seu desempenho. No presente estudo, propõem-se duas modificações no esquema básico da ED. A primeira modificação é o conceito de dividir a população em diversos conjuntos (ou subpopulações) e a segunda modificação é a utilização de uma única matriz em ED, em contraste com as duas matrizes utilizadas em ED básica proposta por Storn e Price.

2. Evolução Diferencial

A Evolução Diferencial (ED) foi criada após Ken Price tentar usá-la para resolver o problema do polinômio de Chebychev o qual foi apresentado por Rainer Storn. A descoberta aconteceu quando Price teve a idéia de utilizar diferentes vetores para recriar o vetor população criando uma abordagem diferente das encontradas nas estratégias de evolução que tratam de problemas de otimização. Desde então foram feitos vários testes e aperfeiçoamentos os quais tornaram o algoritmo ED versátil e robusto.

O algoritmo é iniciado criando uma população inicial de N_p indivíduos, escolhida aleatoriamente e devendo cobrir todo o espaço de busca. Geralmente, é criada por uma distribuição de probabilidade

uniforme, quando não há nenhum conhecimento sobre o problema.

Cada indivíduo, chamado de *vetor*, possui n componentes representadas por valores reais, sendo n o número de variáveis de projeto. Assim, a população segue uma evolução natural, em que o número de indivíduos é constante durante todas as gerações.

A idéia principal da evolução diferencial é gerar novos indivíduos, denotados vetores modificados ou doadores, pela adição da diferença ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Esta operação é chamada *mutação*.

As componentes do indivíduo doador são misturadas com as componentes de um indivíduo escolhido aleatoriamente (denotado vetor alvo), para resultar o chamado vetor tentativa, ou vetor experimental. O processo de misturar os parâmetros é referido como *cruzamento*. Se o vetor experimental resultar um valor da função objetivo menor que o vetor alvo, então o vetor experimental substitui o vetor alvo na geração seguinte. Esta última operação é chamada *seleção*. O procedimento é finalizado quando algum critério de parada é obedecido.

2.1 Operadores da Evolução Diferencial

Assim como o Algoritmo Genético, a Evolução Diferencial apresenta algumas operações a serem seguidas durante cada geração para manter a convergência não-prematura do método, garantindo, então, a diversidade da população e a obtenção da solução ótima, ou próxima desta.

Considere o seguinte problema:

$$\begin{cases} \min f(X), & X \in \mathbb{R}^n \\ \text{sujeito a } C_j(X) \geq 0, & j = 1, \dots, m \end{cases} \quad (1)$$

onde, $f(X)$ representa a função objetivo, também chamada de função custo ou função adaptação, C_j as restrições de desigualdade, assumindo valores reais, e sejam $X^L = (x_1^L, x_2^L, \dots, x_n^L)^T$ e $X^U = (x_1^U, x_2^U, \dots, x_n^U)^T$ as restrições laterais inferior e superior respectivamente.

A população de indivíduos durante a q -ésima geração é definida como:

$$X_d^{(q)} = (x_{d,1}, x_{d,2}, \dots, x_{d,n})^T, \quad i = 1, \dots, n \quad e \quad d = 1, \dots, N_p \quad (2)$$

A população inicial é escolhida aleatoriamente, dentro do espaço de busca, definido pelas restrições laterais, fornecidas pelo usuário:

$$x_{d,i}^{(0)} = x_i^L + r(x_i^U - x_i^L), \quad i = 1, \dots, n \quad e \quad d = 1, \dots, N_p \quad (3)$$

sendo que r é um número gerado randomicamente entre 0 e 1.

Agora com a população estabelecida, é necessário a criação de novos indivíduos, isto se dará através dos operadores que serão apresentados em seguida.

2.1.1 Mutação

Sejam os vetores X_α , X_β e X_γ escolhidos aleatoriamente e distintos entre si. Na geração q um par de vetores (X_β, X_γ) define uma diferença $X_\beta - X_\gamma$. Esta diferença é multiplicada por $F > 0$, sendo denotada por diferença ponderada, e é usada para perturbar o terceiro vetor X_α ou o melhor vetor X_{best} da população. Este processo, que resulta no vetor doador $V^{(q+1)}$, pode ser escrito matematicamente como:

$$\begin{aligned} V^{(q+1)} &= X_\alpha^{(q)} + F \left(X_\beta^{(q)} - X_\gamma^{(q)} \right) \\ &\quad \text{ou} \\ V^{(q+1)} &= X_{best}^{(q)} + F \left(X_\beta^{(q)} - X_\gamma^{(q)} \right) \end{aligned} \quad (4)$$

sendo que os índices aleatórios $\alpha, \beta, \gamma \in \{1, \dots, N_p\}$ são inteiros distintos entre si e diferentes do índice d . O número de indivíduos da população, N_p , deve ser maior ou igual a 4. O fator de escala, ou taxa de perturbação, F , é um fator real, positivo e constante variando entre 0 e 2, cujo objetivo é controlar a amplitude da diferença ponderada. A Figura 1 mostra um exemplo bidimensional que ilustra os diferentes vetores que participam da geração do vetor doador $V^{(q+1)}$.

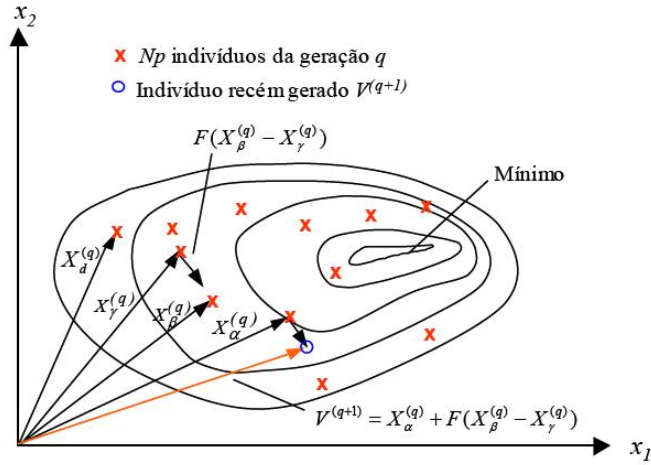


Figura 1: Processo de gerar o vetor doador $V^{(q+1)}$ para uma função objetivo bidimensional

Se o número de indivíduos da população é grande o suficiente, a diversidade da população pode ser melhorada usando duas diferenças ponderadas para perturbar um vetor existente, ou seja, cinco vetores distintos são escolhidos aleatoriamente na população atual. O vetor diferença ponderada usa dois pares de diferenças ponderadas e é usado para perturbar o quinto vetor ou o melhor vetor da população atual. Este processo pode ser dado por:

$$\begin{aligned}
 V^{(q+1)} &= X_{\alpha}^{(q)} + F \left(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)} \right) \\
 \text{ou} \\
 V^{(q+1)} &= X_{best}^{(q)} + F \left(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)} \right)
 \end{aligned} \tag{5}$$

onde os índices aleatórios $\alpha, \beta, \delta, \gamma \in \{1, \dots, N_p\}$, são inteiros mutuamente distintos e diferentes do índice d , tais que $N_p \geq 6$.

Existem outras maneiras, que serão apresentadas adiante, de realizar a operação de mutação, o que diferencia as diversas estratégias que podem ser utilizadas pelo método da Evolução Diferencial.

2.1.2 Cruzamento

Na operação cruzamento, cujo objetivo é aumentar a diversidade dos indivíduos que sofreram a mutação, as componentes do vetor doador, $V^{(q+1)}$, são misturadas com as componentes de outro indivíduo denominado vetor alvo. A escolha do vetor alvo, que deve ser diferente dos vetores já usados anteriormente, é aleatório, segundo uma probabilidade de cruzamento CR . Desta forma, obtém-se o vetor tentativa ou experimental, $U^{(q+1)}$, definido como:

$$u_i^{(q)} = \begin{cases} v_i^{(q+1)}, & \text{se } r_i \leq CR \\ x_{d,i}^{(q+1)}, & \text{se } r_i > CR, \quad i = 1, \dots, n. \end{cases} \tag{6}$$

onde r_i é um número randômico entre 0 e 1 e $x_{d,i}$ são as componentes do vetor alvo $X_d^{(q)}$. A probabilidade do cruzamento ocorrer, CR , representa a probabilidade do vetor experimental herdar os valores das variáveis do vetor doador, e está compreendida entre 0 e 1, sendo fornecida pelo usuário. Quando $CR = 1$, por exemplo, todas as componentes do vetor experimental virão do vetor doador $V^{(q+1)}$. Por outro lado, se $CR = 0$, todas as componentes do vetor experimental virão do vetor alvo $X_d^{(q)}$.

Se após o cruzamento uma ou mais componentes do vetor experimental estiver fora da região de busca, fazem-se as correções:

$$\begin{cases} \text{Se } u_i < x_i^L \text{ então } u_i = x_i^L \\ \text{Se } u_i > x_i^U \text{ então } u_i = x_i^U \end{cases}$$

O cruzamento dado pela Equação 6 é denominado de *cruzamento binomial* e pode ser observado na Figura 2.

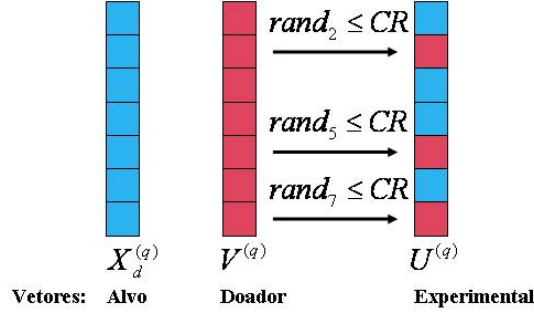


Figura 2: Cruzamento binomial

Outra forma de realizar o cruzamento, denominado *cruzamento exponencial*, é definida segundo a Figura 3. As componentes do vetor experimental são dadas pelas componentes do vetor doador enquanto o número randômico for menor ou igual à probabilidade de cruzamento CR .

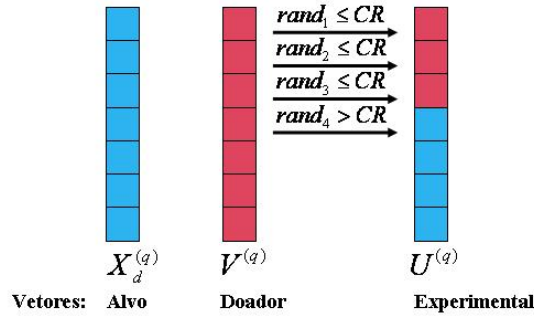


Figura 3: Cruzamento exponencial

Após determinado o vetor experimental $U^{(q)}$, torna-se necessário selecionar os melhores descendentes. Esta operação será vista na próxima seção.

2.1.3 Seleção

A seleção é o processo de produzir melhores filhos. Diferentemente de outros algoritmos evolutivos, a evolução diferencial não usa hierarquia (elitismo) nem seleção proporcional. Em vez disso, o custo do vetor experimental $U^{(q+1)}$ é calculado e comparado com o custo do vetor alvo $X_d^{(q)}$. Se o custo do vetor alvo for menor que o custo do vetor experimental, o vetor alvo é permitido avançar para a próxima geração. Caso contrário, o vetor experimental substitui o vetor alvo na geração seguinte. Em outras palavras, este processo pode ser escrito como:

$$\begin{cases} \text{Se } f(U^{(q+1)}) \leq f(X_d^{(q)}), \text{ então } X_d^{(q+1)} = U^{(q+1)} \\ \text{Se } f(U^{(q+1)}) > f(X_d^{(q)}), \text{ então } X_d^{(q+1)} = X_d^{(q)} \end{cases} \quad (7)$$

O processo iterativo continua até que seja alcançado algum critério de parada, sendo que um número máximo de gerações deve ser estabelecido. Para problemas com restrição, um critério pode ser a não violação das restrições ou o melhor indivíduo ter encontrado um valor dentro de uma precisão pré-estabelecida. Assim, os seguintes critérios de parada podem ser estabelecidos:

$$\begin{cases} q < \max \\ |f_{\min}^{(q+1)} - f_{\min}^{(q)}| < \epsilon \end{cases} \quad (8)$$

2.2 Estratégias da Evolução Diferencial

Vale ressaltar que a Evolução Diferencial apresenta diferentes estratégias obtidas a partir da forma com que os operadores de mutação e cruzamento trabalham, ou seja, as estratégias da evolução diferencial podem variar de acordo com o tipo de indivíduo a ser modificado na formação do vetor doador, o número de indivíduos considerados para a perturbação e o tipo de cruzamento a ser utilizado, podendo ser escritas como: ED/a/b/c, sendo que:

a $\rightarrow$ especifica o vetor a ser perturbado, podendo ser *rand* (um vetor da população escolhido aleatoriamente) ou *best* (o vetor de menor custo da população);

b $\rightarrow$ determina o número de diferenças ponderadas usadas para a perturbação de a;

c $\rightarrow$ denota o tipo de cruzamento (exp: exponencial; bin: binomial).

Em 1995, Storn and Price deram o princípio de trabalho da estratégia básica usando apenas o operador cruzamento binomial (devido aos experimentos binomiais independentes), onde o cruzamento é executado em cada variável sempre que um número $r \in [0; 1]$ aleatório for menor que a probabilidade de cruzamento CR .

Alguns anos mais tarde, Storn and Price (1997) desenvolveram mais estratégias usando o operador cruzamento exponencial, em que o cruzamento é executado nas variáveis em um laço até que esteja dentro do limite de CR . A primeira vez que um número $r \in [0; 1]$ aleatório ultrapassa o valor de CR , nenhum cruzamento é executado e as variáveis restantes são deixadas intactas. Resumidamente, as dez estratégias podem ser descritas de acordo com a Tabela 1.

No entanto, uma estratégia que funciona bem para um dado problema pode não funcionar bem quando aplicada a outro problema. A estratégia a ser adotada para um problema é determinada por tentativa e erro.

Tabela 1: Estratégias do método Evolução Diferencial

Estratégia	Operador Mutação	Notação
1	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/rand/1/bin
2	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/best/1/bin
3	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand/2/bin
4	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/best/2/bin
5	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand-to-best/2/bin
6	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/rand/1/exp
7	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/best/1/exp
8	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand/2/exp
9	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/best/2/exp
10	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand-to-best/2/exp

3. Evolução com Conjuntos Embaralhados

A ideia principal da abordagem da Evolução com Conjuntos Embaralhados (ECE) é tratar a procura global como um processo de evolução natural. Neste estudo será aplicado esse conceito para o algoritmo da ED. Os N_p pontos de amostragem, constituem uma população que é dividida em diversos

conjuntos (sub-populações), tais que cada conjunto tenha o mesmo número de indivíduos. A cada um dos conjuntos é permitido evoluir de forma independente (ou seja, explorar o espaço de busca em direções diferentes). Depois de um certo número de gerações, os conjuntos são forçados a se misturar e novos conjuntos são formados através de um processo de embaralhamento. Este procedimento aumenta a capacidade de sobrevivência por um intercâmbio de informações sobre o espaço de busca adquirida de forma independente por cada conjunto.

Cada membro de um conjunto é um pai com a capacidade de participar de um processo de reprodução. Por fim, cada novo descendente substitui o seu ponto de destino correspondente no complexo atual ao invés do ponto de toda a população. Isto garante que cada pai receba pelo menos uma chance de contribuir para o processo de reprodução antes de serem substituídos ou eliminados. Assim, nenhuma das informações contidas na amostra é ignorada. O método ECE é projetado para melhorar características do método da ED, incorporando em si o poderoso conceito de conjuntos embaralhados.

Conjuntos embaralhados ajudam a garantir que as informações contidas na amostra são eficientes e completamente exploradas. Também ajudam a garantir que o conjunto de informações não se degenere. Estas vantagens do método ECE que motivam a aplicá-lo sobre o algoritmo da ED, com a esperança de se obter melhores propriedades de convergência global sobre uma ampla gama de problemas. Em outras palavras, dado um número pré-especificado de avaliações da função objetivo (nível fixo de eficiência), espera-se que o método ECE tenha uma maior probabilidade de sucesso em seu objetivo de encontrar o ótimo global em comparação à ED.

O método ECE é baseado em uma síntese de quatro conceitos que se revelaram bem sucedidas para a otimização global (Holland, 1975): (a) combinação de abordagens aleatórias e determinísticas, (b) o conceito de computação paralela, (c) o conceito de uma evolução sistemática de um conjunto de pontos que exploram o espaço, no sentido da melhoria global, (d) o conceito de evolução competitiva.

Uma breve discussão destes conceitos é apresentada aqui. O uso de estratégias determinísticas permite que o algoritmo ECE faça uso efetivo das informações da superfície de resposta para orientar a pesquisa, enquanto a inclusão de elementos aleatórios ajuda a tornar o algoritmo robusto e flexível. A pesquisa começa com um conjunto de pontos selecionado ao acaso abrangendo todo o espaço de busca Ω . Um grande, porém suficiente, número de pontos ajuda a garantir que o conjunto contenha informações quanto ao número, localização e tamanho das principais regiões viáveis.

A implementação de uma estratégia de agrupamento implícito ajuda a concentrar a busca na mais promissora das regiões identificadas pelo conjunto inicial. O uso de uma estratégia de evolução sistemática ajuda a garantir que a busca seja relativamente robusta e guiada pela estrutura da função objetivo. A robustez é resultado do fato de que a estrutura de conjuntos é capaz de lidar muito bem com superfícies da função objetivo rudes, insensíveis e altamente não convexas e é relativamente pouca afetada pelos mínimos locais pequenos que são encontrados na busca pela solução global. Além disso, não são necessárias informações de derivadas. A implementação de uma estratégia de evolução competitiva, conforme descrita por Holland (1975), tem sido feita por ser útil na melhoria da eficiência da convergência global.

3.1 O Método Evolução com Conjuntos Embaralhados - ECE

Em seguida será apresentado um algoritmo para o método Evolução com Conjuntos Embaralhados conforme descrito na trabalho de Duan *et al.* (1993). Um fluxograma para este método é apresentado na Figura 4.

Passo 1: *Inicialização.* Selecione $p \geq 1$ e $m \geq n + 1$, onde p é o número de conjuntos e m é o número de pontos em cada conjunto. Calcule o tamanho da amostra $s = p \times m$.

Passo 2: *Geração da amostra.* Obtenha a amostra de s pontos $x_1, \dots, x_s$ pertence à região viável $\Omega \subset \mathbb{R}^n$. Calcule o valor da função f_i em cada ponto x_i . Na ausência de informação prévia, use uma distribuição uniforme de amostragem.

Passo 3: *Rank points.* Organize os s pontos na ordem crescente do valor da função. Guarde-os

em uma matriz $D = \{x_i, f_i; i = 1, \dots, s\}$, de modo que $i = 1$ represente o ponto com o menor valor da função.

Passo 4: *Particionando em conjuntos.* Particione D em p conjuntos $A^1, \dots, A^p$, cada um contendo m pontos, tais que:

$$A^k = \{x_j^k, f_j^k; x_j^k = x_{k+p(j-1)}, f_j^k = f_{k+p(j-1)}, j = 1, \dots, m\}.$$

Passo 5: *Evoluir cada conjunto.* Evoluir cada conjunto $A^k, k = 1, \dots, p$ de acordo com o algoritmo da evolução com conjuntos competitivos (ECC) descrito separadamente.

Passo 6: *Conjuntos embaralhados.* Substitua $A^1, \dots, A^p$ em D , tal que $D = \{A^k, k = 1, \dots, p\}$. Ordene D em ordem de crescimento do valor da função.

Passo 7: *Testando a convergência.* Se os critérios de convergência foram satisfeitos, pare. Caso contrário, retorne ao passo 4.

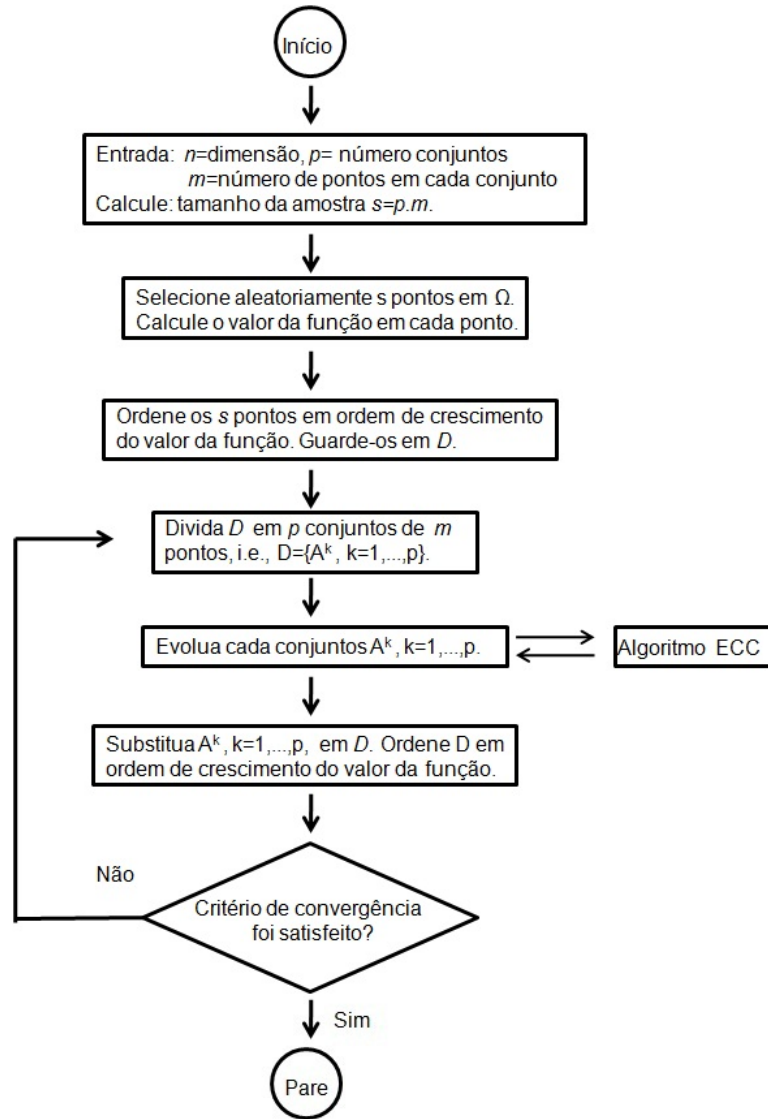


Figura 4: Fluxograma do Método Evolução com Conjuntos Embaralhados(fonte: Duan *et al.* (1993), adaptado pelas autoras).

A filosofia por trás da abordagem ECE é tratar a busca global como um processo de evolução natural. Os s pontos amostrados, constituem uma população. A população é dividida em várias comunidades (conjuntos), sendo que cada uma delas é permitido evoluir de forma independente, ou seja, busca o espaço em diferentes direções. Depois de um certo número de gerações, as comunidades são forçadas a se misturar e as novas comunidades são formadas por um processo de embaralhamento. Este procedimento aumenta a capacidade de sobrevivência por um intercâmbio de informações (sobre o espaço de busca) obtido de forma independente por cada sub-população.

Cada membro de um conjunto é um pai em potencial com a capacidade de participar de um processo de reprodução. Um subconjunto selecionado a partir do conjunto é como um par de pais, salvo que um subconjunto pode ser composto por mais de dois membros. Para garantir que o processo de evolução seja competitivo, é necessário que a probabilidade de que os melhores pais contribuam para a geração de descendentes seja maior que a dos piores pais. O uso de uma distribuição de probabilidade triangular assegura esta competitividade. O procedimento de Nelder e Mead (1965) é aplicada a cada subconjunto para gerar a maior parte dos descendentes. Esta estratégia utiliza as informações contidas no subconjunto para conduzir a evolução para uma direção de melhora. Além disso, os descendentes são introduzidos em locais aleatórios do espaço viável sob determinadas condições, a fim de garantir que o processo de evolução não fique preso em regiões pouco promissoras. Isso é um pouco semelhante ao processo da mutação em resposta à convergência prematura que pode ocorrer na evolução biológica. Cada mutação também ajuda a aumentar a quantidade de informação armazenada na amostra. Por fim, cada descendente novo substitui o pior ponto do subconjunto atual, ao invés de o pior ponto da população. Isso garante que cada pai receba pelo menos uma chance de contribuir para o processo de reprodução, antes de ser substituído ou descartado. Assim, nenhuma das informações contidas na amostra é ignorada. A arquitetura do método ECE é ilustrado na Figura 5.

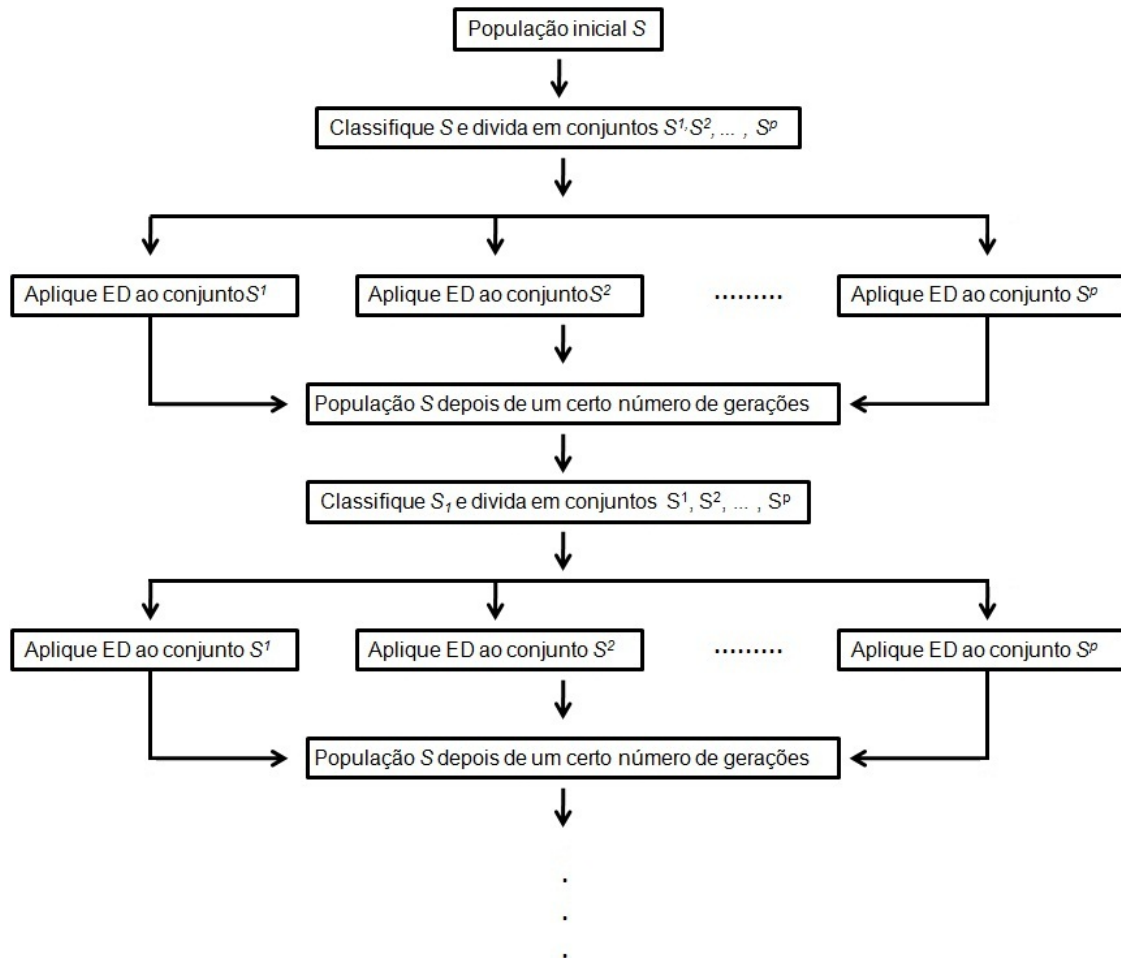


Figura 5: Arquitetura do Método Evolução com Conjuntos Embaralhados (fonte: Duan *et al.* (1993), adaptado pelas autoras).

3.2 O Método Evolução com Conjuntos Competitivos - ECC

O algoritmo da Evolução com Conjuntos Competitivos (ECC) necessário para a atualização de cada conjunto no Passo 5 do método ECE é apresentado a seguir e é ilustrado na Figura 6.

Passo 1: *Inicialização.* Selecione q, α , e β , onde $2 \leq q \leq m, \alpha \geq 1$, e $\beta \geq 1$.

Passo 2: *Atribuir pesos.* Atribuir uma distribuição de probabilidade triangular para A^k , isto é,

$$p_i = 2(m+1-i)/m(m+1), \quad i = 1, \dots, m. \quad (9)$$

Observando a Equação (9), verifica-se que o ponto x_1^k ($i = 1$) tem a maior probabilidade, $p_1 = 2/(m+1)$. O ponto x_m^k ($i = m$) tem a menor probabilidade, $p_m = 2/m(m+1)$.

Passo 3: *Seleção dos pais.* Escolher aleatoriamente q pontos distintos $u_1, \dots, u_q$ de A^k de acordo com a distribuição de probabilidade acima especificada (os q pontos definem um sub-conjunto). Armazene-os na matriz $B = \{u_i, v_i, i = 1, \dots, q\}$ onde v_j é o valor da função associado ao ponto u_j . Armazene em L as localizações de A^k que são usadas para construir B .

Passo 4: *Geração de descendentes.*

(a) Ordenar B e L de modo que os q pontos estejam dispostos em ordem crescente do valor da função. Calcule o centróide g , utilizando a expressão:

$$g = [1/(q-1)] \sum_{j=1}^{q-1} u_j. \quad (10)$$

(b) Calcule o novo ponto (etapa de reflexão)

$$r = 2g - u_q. \quad (11)$$

(c) Se $r \in \Omega$, calcule o valor da função f e vá para a etapa (d), caso contrário, calcule o menor hipercubo $H \subset \mathbb{R}^n$ que contenha A^k , gere aleatoriamente um ponto $z \in H$, calcule f_z , defina $r = z$ e $f_r = f_z$ (etapa mutação).

(d) Se $f_r < f_q$, substitua u_q por r e vá para a etapa (f); caso contrário, (etapa contração) calcule

$$c = (q + u_q)/2 \quad e \quad f_c. \quad (12)$$

(e) Se $f_c < f_q$, substitua u_q por c e vá para a etapa (f); caso contrário, gere aleatoriamente um ponto $z \in H$ e calcule f_z (etapa mutação). Substitua u_q por z .

(f) Repita α vezes as etapas de (a) até (e), sendo que $\alpha \geq 1$ é um parâmetro especificado pelo usuário.

Passo 5: *Substituição dos pais pelos descendentes.* Substitua B por A^k usando a localização original armazenada em L . Ordene A^k segundo o crescimento do valor da função.

Passo 6: *Iteração.* Repita β vezes os passos de (1) até (4), onde $\beta \geq 1$ é um parâmetro especificado pelo usuário que determina quantos filhos deverão ser gerados (o quanto cada conjunto deverá evoluir).

4. Método de Penalidade Aplicado às Técnicas Evolutivas

Muitos algoritmos de otimização foram desenvolvidos para resolver problemas irrestritos, considerando apenas os limites laterais das variáveis, como no caso da Evolução Diferencial, enquanto apenas

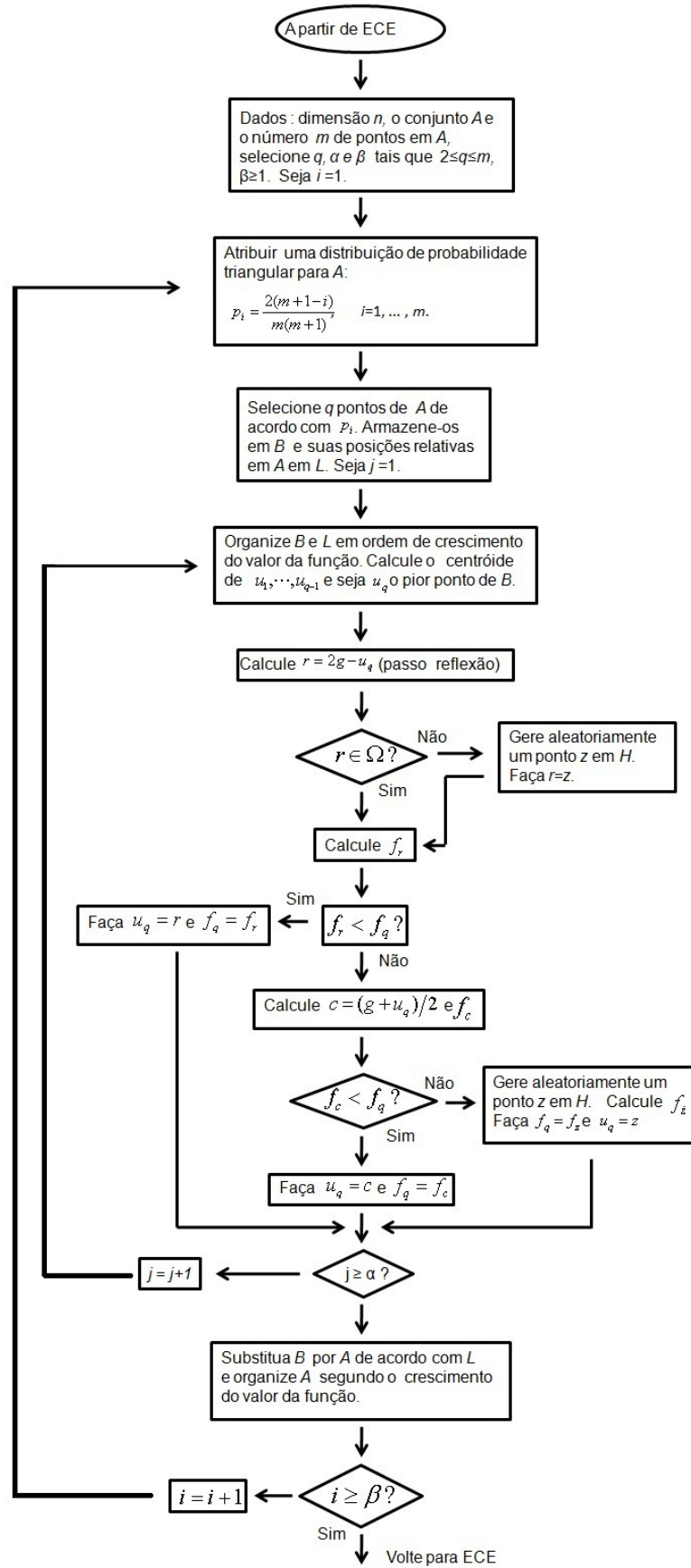


Figura 6: Fluxograma do Método Evolução com Conjuntos Competitivos (fonte: Duan *et al.* (1993), adaptado pelas autoras).

alguns algoritmos processam as restrições de desigualdade e igualdade do problema para limitar a região viável. Assim, torna-se necessário utilizar algum artifício para que os problemas com restrições também se apliquem aos métodos de otimização irrestrita.

Uma das abordagens fundamentais para a otimização restrita é substituir o problema original por uma função de penalidade que é composta da função objetivo original do problema de otimização com restrições somada a um termo adicional para cada restrição, o qual é positivo quando o ponto atual X viola essa restrição e zero caso contrário.

A maioria das abordagens definem uma sequência de funções de penalidade, na qual os termos penalizados por violarem as restrições são multiplicados por um coeficiente positivo. Ao fazer este coeficiente crescer, penalizam-se as violações das restrições mais severamente, forçando o ponto mínimo da função de penalidade se aproximar cada vez mais da região viável do problema restrito.

Essas abordagens são conhecidas como Método da Penalidade Exterior, porque o termo penalizado por cada restrição é diferente de zero somente quando X é inviável em relação a tal restrição. Muitas vezes, o ponto de mínimo das funções de penalidade são inviáveis com relação ao problema original, e aproximam-se da viabilidade apenas no limite em que os parâmetros de penalidade se tornam cada vez maiores.

Considere o seguinte problema:

$$\begin{aligned} \min f(X) \quad \text{sujeito a} \quad & g_j(X) \geq 0, \quad \text{restrições de desigualdade} \\ & X \in \mathbb{R}^n \quad h_l(X) = 0, \quad \text{restrições de igualdade} \end{aligned} \quad (13)$$

Afim de que os problemas restritos 13 sejam transformados em problemas irrestritos será utilizado, neste estudo, o Método da Penalidade Exterior. Se o objetivo da otimização é minimizar a função objetivo $f(X)$, a função de penalidade $P(X)$ é somada a função principal $f(X)$ de modo a aumentar o valor da função nos pontos que estão fora da região viável, por outro lado, se o objetivo é maximizar, a função de penalidade é subtraída da função principal se o ponto não obedece às restrições.

Esta nova função objetivo, chamada *pseudo objetivo*, é penalizada de acordo com um *fator de penalidade* r_p toda vez que encontrar uma restrição ativa. Assim, este escalar amplia a penalidade, e quanto maior for seu valor, maior será a eficiência do método para obedecer às restrições. O modelo matemático da função pseudo objetivo $\Phi(X)$ utilizada para minimizar uma função $f(X)$ é dada por:

$$\Phi(X) = f(X) + r_p P(X), \quad (14)$$

e a função de penalidade $P(X)$ dada por:

$$P(X) = \sum_{j=1}^m \left(\max [0, g_j(X)]^2 \right) + \sum_{k=1}^l h_k(X)^2, \quad (15)$$

sendo $g_j(X)$ as restrições de desigualdade, $h_l(X)$ as restrições de igualdade e m e l o número de restrições de desigualdade e igualdade respectivamente.

Deste modo, os pontos fora do espaço viável apresentarão valores muito ruins para a otimização e serão facilmente desconsiderados pelos algoritmos de otimização irrestrita.

5. Simulações Numéricas

Para testar e avaliar os algoritmos Evolução Diferencial e Evolução com Conjuntos Embaralhados, serão propostos, a seguir, alguns problemas-teste, necessários para comparar as técnicas apresentadas neste estudo. O código foi escrito na forma de um toolbox para *MATLAB*®.

5.1 Projeto de um recipiente de pressão

O objetivo deste problema é o de minimizar o custo total do desenho de um recipiente de pressão conforme a Figura 7.

Existem quatro variáveis no projeto: x_1 (T_s , espessura do invólucro), x_2 (T_h , espessura da tampa), x_3 (R , raio interno) e x_4 (L , comprimento da seção cilíndrica do recipiente), sendo que R e L são contínuos. O problema pode ser descrito conforme segue:

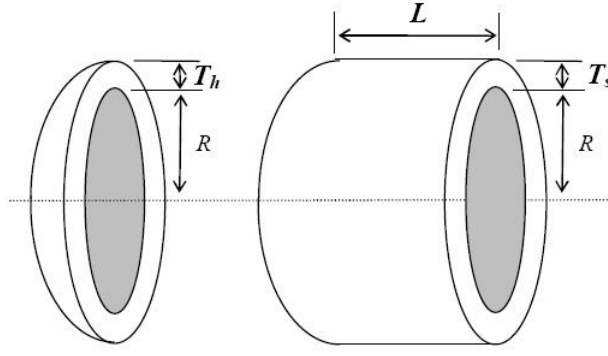


Figura 7: Projeto de um recipiente de pressão

Tabela 2: Comparação de alguns resultados do problema do recipiente de pressão

Variáveis de projeto	Hu <i>et al.</i> (2003)	Coello (2000)	Deb (1997)	Evolução Diferencial
x_1	0,8125	0,8125	0,9375	0,2233
x_2	0,4375	0,4375	0,5000	0,9611
x_3	42,09845	40,3239	48,3290	4 53,7741
x_4	176,6366	200,0000	112,6790	175,9351
$g_1(x)$	0,0	-0,034324	-0,004750	0,8146
$g_2(x)$	-0,03588	-0,052847	-0,038941	-0,4481
$g_3(x)$	-5,820e-11	-27,10584	-3652,876	$-9 * 10^5$
$g_4(x)$	-63,3624	-40,0000	-127,3210	-64,0649
$f(x)$	6059,1312	6288,7445	6410,3811	6343,9254

$$\min f(x) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \quad (16)$$

Sujeito a:

$$g_1(x) = -x_1 + 0,0193x_3 \leq 0 \quad (17)$$

$$g_2(x) = -x_2 + 0,00954x_3 \leq 0 \quad (18)$$

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (19)$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (20)$$

Os seguintes limites laterais das variáveis $x = (x_1, x_2, x_3, x_4)$ serão utilizados:

$$0 \leq x_1 \leq 10, 0 \leq x_2 \leq 10, 10 \leq x_3 \leq 100, 100 \leq x_4 \leq 200.$$

Na Tabela 2 são apresentados alguns valores aproximados dos resultados encontrados na literatura e os resultados aproximados obtidos utilizando Evolução Diferencial para a resolução do problema de recipiente de pressão. Na Tabela 3 são apresentados os resultados obtidos com o método Evolução com Conjuntos Embaralhados em quantidades diferentes de conjuntos, sendo que t representa o tempo de processamento em segundos.

5.2 Projeto de uma viga engastada

O objetivo é minimizar o custo de uma viga com as limitações de tensão de cisalhamento, tensão de dobramento na viga, esforço de carga na barra, reflexão final da viga e restrições laterais, conforme expresso na Figura 8.

Tabela 3: Resultados do problema de um recipiente de pressão para ECE.

Variáveis de projeto	2 Conjuntos t=5,2570s	4 Conjuntos t=5,7250s	10 Conjuntos t=5,5380s	20 Conjuntos t=7,4560s
x_1	0,8631	0,8555	0,7787	0,7784
x_2	0,4267	0,4229	0,3850	0,3842
x_3	44,6980	44,3239	40,3499	40,3275
x_4	146,9913	150,8823	199,5795	199,8933
$g_1(x)$	-0,0004	-9,15e-6	-6,72e-7	-4,0e-5
$g_2(x)$	-0,0003	-6,14e-5	-8,26e-5	-1,0e-4
$g_3(x)$	-679,6647	-3,3706	-7,3757	-13,1134
$g_4(x)$	-93,0087	-89,1177	-40,4205	-40,1067
$f(x)$	6052,4298	6031,2781	5886,6091	5886,2353

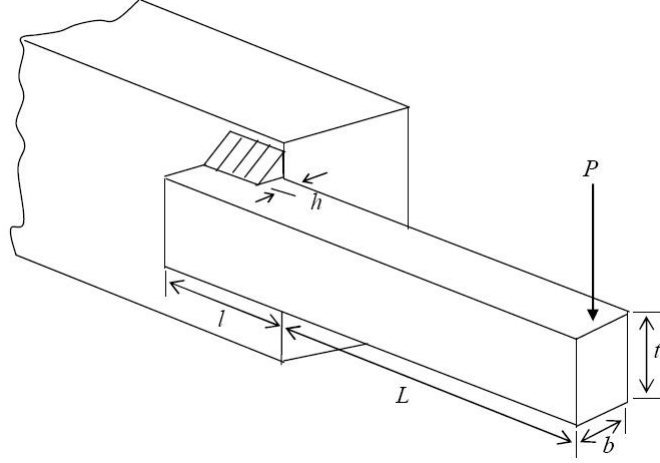


Figura 8: Projeto de uma viga engastada

O problema pode ser expresso como:

$$\min f(x) = 1,10471x_1^2x_2 + 0,04811x_3x_4(14,0 + x_2) \quad (21)$$

Sujeito a:

$$g_1(x) = \tau(x) - \tau_{max} \leq 0 \quad (22)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \quad (23)$$

$$g_3(x) = x_1 - x_4 \leq 0 \quad (24)$$

$$g_4(x) = 0,10471x_1^2 + 0,04811x_3x_4(14,0 + x_2) - 5 \leq 0 \quad (25)$$

$$g_5(x) = 0,125 - x_1 \leq 0 \quad (26)$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0 \quad (27)$$

$$g_7(x) = P - P_c(x) \leq 0 \quad (28)$$

sendo que,

$$\tau(x) = \sqrt{\tau'^2 + 2\tau'\tau''\frac{x_2}{2R} + \tau''^2} \quad (29)$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2} \quad (30)$$

$$\tau'' = \frac{MR}{J} \quad (31)$$

$$M = P(L + \frac{x_2}{2}) \quad (32)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad (33)$$

$$J = 2 \left[\sqrt{2}x_1x_2 \left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right) \right] \quad (34)$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2} \quad (35)$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3x_4} \quad (36)$$

$$P_c(x) = \frac{4,013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}} \right) \quad (37)$$

onde:

$P = 6000 \text{ lb}$, $L = 14 \text{ in}$, $E = 30 \cdot 10^6 \text{ psi}$, $G = 12 \cdot 10^6 \text{ psi}$, $\tau_{max} = 13600 \text{ psi}$, $\sigma_{max} = 30000 \text{ psi}$
e $\delta_{max} = 0,25 \text{ in}$.

Os seguintes limites laterais das variáveis $x = (x_1, x_2, x_3, x_4)$ serão utilizados:

$0,1 \leq x_1 \leq 2$, $0,1 \leq x_2 \leq 10$, $0,1 \leq x_3 \leq 10$, $0,1 \leq x_4 \leq 2$.

Na tabela 4 são apresentados os valores aproximados dos resultados encontrados na literatura e obtidos com a Evolução Diferencial. Na tabela 5 são apresentados os resultados para o método Evolução com Conjuntos Embaralhados, na qual t representa o tempo de processamento em segundos.

Tabela 4: Comparação de alguns resultados para o projeto de um viga engastada

Variáveis de projeto	Hu <i>et al.</i> (2003)	Coello (2000)	Deb (1997)	Evolução Diferencial
x_1	0,2057	0,2088	0,2489	0,1907
x_2	3,4705	3,4205	6,1730	4,2191
x_3	0,0366	8,9975	8,1739	8,4662
x_4	0,2057	0,2100	0,2533	0,2408
$g_1(x)$	0,0	-0,3378	$-5 * 10^3$	-299,4210
$g_2(x)$	0,0	-353,9026	-255,5769	-803,1146
$g_3(x)$	$-5 * 10^{-17}$	-0,0012	-0,0044	-0,0502
$g_4(x)$	-3,4330	-3,4119	-2,9829	-3,2090
$g_5(x)$	-0,0807	-0,0838	-0,1239	-0,0657
$g_6(x)$	-0,2355	-0,2356	-0,2347	-0,2350
$g_7(x)$	$-9 * 10^{-13}$	-363,2323	$-4 * 10^3$	$-3 * 10^3$
$f(x)$	1,7248	1,7483	2,4331	1,9566

Tabela 5: Resultados do problema de uma viga engastada para ECE.

Variáveis de projeto	2 Conjuntos t=7,41s	4 Conjuntos t=4,8830s	10 Conjuntos t=6,7230s	20 Conjuntos t=7,9880s
x_1	0,2794	0,2055	0,2057	0,2058
x_2	2,8417	3,4753	3,4724	3,4703
x_3	7,4943	9,0393	9,0366	9,0358
x_4	0,2991	0,2057	0,2058	0,2058
$g_1(x)$	-0,5225	-2,1166	-3,9511	-1,1344
$g_2(x)$	-0,8868	-17,8126	-4,2678	-0,4141
$g_3(x)$	-0,0197	-0,0002	-0,0001	0
$g_4(x)$	-3,1754	-3,4321	-3,4326	-3,4328
$g_5(x)$	-0,1544	-0,0805	-0,0807	-0,0808
$g_5(x)$	-0,2326	-0,2355	-0,2355	-0,2355
$g_7(x)$	-1089,7967	-1,1995	-2,6463	-3,1720
$f(x)$	2,0614	1,7256	1,7253	1,7251

Os resultados obtidos mostram que os algoritmos utilizados, Evolução Diferencial e Evolução com Conjuntos Embaralhados, foram eficientes em encontrar um valor ótimo para os problemas apresentados. Embora os problemas estudados sejam relativamente simples de serem resolvidos, enquanto a Evolução Diferencial levou cerca de 20 segundos para obter o ótimo, o método Evolução com Conjuntos Embaralhados precisou, em média, de um tempo computacional três vezes menor.

5.3 Projeto ótimo de um manipulador 3R ortogonal

Neste trabalho, deseja-se estudar os robôs manipuladores com três juntas rotacionais, cuja abreviação é dada por 3R, conforme Figura 9.

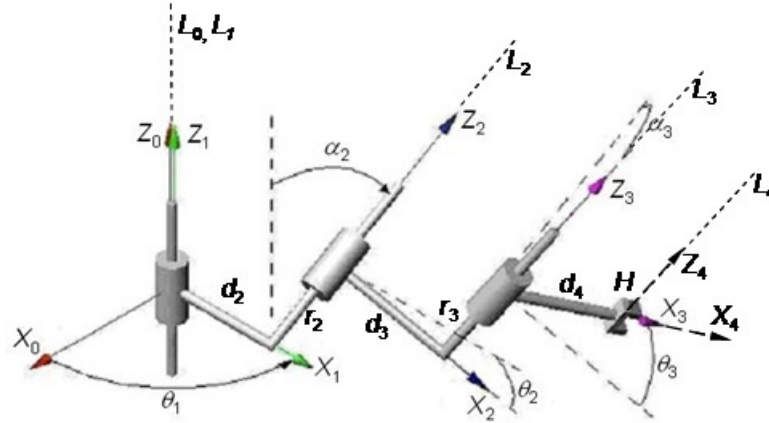


Figura 9: Esquema cinemático de um robô manipulador 3R com seus parâmetros de projeto

O mecanismo de um manipulador em série é um conjunto cinemático constituído de uma sucessão de corpos rígidos ligados entre si por juntas rotacionais e/ou prismáticas. Uma forma de representação cinemática comum é a utilização dos parâmetros de Denavit-Hartenberg (DH). Dois sistemas de referência R_{j-1} e R_j são relacionados pela matriz de transformação T_j^{j-1} , dada por:

$$T_j^{j-1} = \begin{pmatrix} c_j & -s_j & 0 & d_j \\ c\alpha_j s_j & c\alpha_j c_j & -s\alpha_j & -r_j s\alpha_j \\ s\alpha_j s_j & s\alpha_j c_j & c\alpha_j & r_j c\alpha_j \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

sendo $c_j = \cos(\theta_j)$, $s_j = \sin(\theta_j)$, $c\alpha_j = \cos(\alpha_j)$ e $s\alpha_j = \sin(\alpha_j)$, $j = 1, \dots, n + 1$.

Assim, ao efetuador de um robô de n graus de liberdade é associado um sistema de coordenadas, T_n^0 , que é uma matriz de transformação homogênea $T_n^0 = T_1^0 \cdot T_2^1 \cdots T_j^{j-1} T_{j+1}^j \cdots T_n^{n-1}$.

Trabalhando com estas matrizes, obtém-se o Modelo Geométrico Direto (MGD) de um manipulador 3R, dado por:

$$\begin{aligned} x &= [d_2 + (r_3 s \alpha_3 - d_4 c \alpha_3 s_3) s_2 + (d_3 + d_4 c_3) c_2] c_1 + \{s \alpha_2 (r_2 + r_3 c \alpha_3 + d_4 s \alpha_3 s_3) \\ &\quad + c \alpha_2 [(r_3 s \alpha_3 - d_4 c \alpha_3 s_3) c_2 - (d_3 + d_4 c_3) s_2]\} s_1 \\ y &= [d_2 + (r_3 s \alpha_3 - d_4 c \alpha_3 s_3) s_2 + (d_3 + d_4 c_3) c_2] s_1 - \{s \alpha_2 (r_2 + r_3 c \alpha_3 + d_4 s \alpha_3 s_3) \\ &\quad + c \alpha_2 [(r_3 s \alpha_3 - d_4 c \alpha_3 s_3) c_2 - (d_3 + d_4 c_3) s_2]\} c_1 \\ z &= c \alpha_2 (r_2 + r_3 c \alpha_3 + d_4 s \alpha_3 s_3) - s \alpha_2 [(r_3 s \alpha_3 - d_4 c \alpha_3 s_3) c_2 - (d_3 + d_4 c_3) s_2] \end{aligned} \quad (38)$$

Cuja matriz Jacobiana J obtida é dada abaixo:

$$J = \begin{pmatrix} -s_3 c_2 d_4 - c_2 r_2 & r_3 & -s_3 d_4 \\ s_3 s_2 d_4 + s_2 r_2 & d_3 + c_3 d_4 & 0 \\ c_2 d_3 + c_2 c_3 d_4 + d_2 + s_2 r_3 & 0 & c_3 d_4 \end{pmatrix} \quad (39)$$

5.3.1 Formulação do Problema de Otimização

O trabalho proposto consiste na síntese dimensional de um manipulador 3R ortogonal ($\alpha_2 = -90^\circ$ e $\alpha_3 = 90^\circ$), onde o projeto ótimo corresponde àquele que otimiza algumas características desejáveis para a melhor performance do robô a executar tarefas. Seja a síntese dimensional deste manipulador formulada pelo seguinte problema de otimização:

$$\text{Maximizar } F(X) = [V \quad -D \quad R] \quad (40)$$

onde

$$X = (1, d_3, d_4, r_2, r_3)^T, \quad 0, 1 \leq d_3, d_4, r_2, r_3 \leq 3, 0 \quad (41)$$

sendo que V é o volume do espaço de trabalho, R é a rigidez da estrutura do mecanismo serial, e D é o índice de isotropia que representa a destreza do manipulador.

5.3.2 Volume do Espaço de Trabalho

O volume do espaço de trabalho, V , é o volume do sólido de revolução obtido pela rotação da seção radial em torno do eixo z . Assim, usando o Teorema de Pappus-Guldin, conforme apresentado na Figura 10, o volume é dado através da equação:

$$V = 2\pi r_g A, \quad (42)$$

sendo A a área da seção radial plana que é coberta pela família de curvas.

Esta pesquisa propõe uma formulação numérica para aproximar o cálculo da área da seção radial através de sua discretização em uma malha retangular (Saramago et al., 2007). Inicialmente, devem-se obter os valores extremos dos vetores r e z , ou seja,

$$\begin{aligned} r_{min} &= \min\{r\} \quad e \quad r_{max} = \max\{r\} \\ z_{min} &= \min\{z\} \quad e \quad z_{max} = \max\{z\} \end{aligned} \quad (43)$$

Adotando-se o número de subintervalos desejados para a discretização ao longo de r e z (n_r e n_z), pode-se calcular as dimensões das áreas elementares da malha através das seguintes expressões:

$$\Delta_r = \frac{r_{max} - r_{min}}{n_r} \quad e \quad \Delta_z = \frac{z_{max} - z_{min}}{n_z} \quad (44)$$

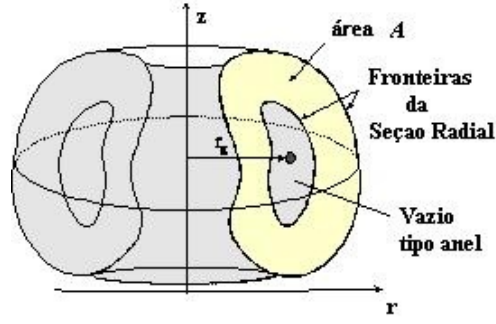


Figura 10: Cálculo do volume do espaço de trabalho de manipuladores 3R

Utilizando as equações de 38, do modelo geométrico direto do manipulador, calcula-se todos os pontos da família de curvas que compõem a seção radial do espaço de trabalho. Dado um determinado ponto (r, z) , determina-se sua posição dentro da malha de discretização, através do seguinte controle de índices:

$$i = \text{int} \left(\frac{r - r_{\min}}{\Delta_r} \right) + 1 \quad e \quad j = \text{int} \left(\frac{z - z_{\min}}{\Delta_z} \right) + 1 \quad (45)$$

Conforme mostrado no esquema da Figura 11, o ponto da malha que pertence ao espaço de trabalho será identificado como $P_{ij} = 1$, caso contrário terá valor nulo, ou seja:

$$P_{ij} = \begin{cases} 0, & \text{se } P_{ij} \notin W(H) \\ 1, & \text{se } P_{ij} \in W(H) \end{cases} \quad (46)$$

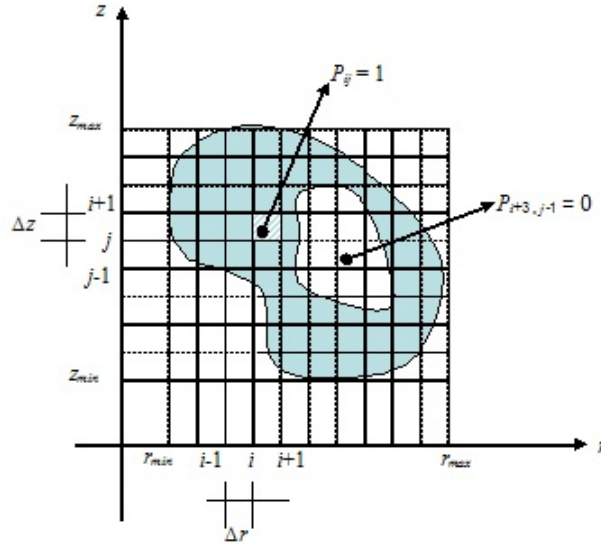


Figura 11: Discretização da seção radial usando malha retangular

Desta forma, a área total é obtida pela soma de todas as áreas elementares da malha que estão contidas, totalmente ou parcialmente, na seção radial, conforme Equação 47. Observa-se que apenas os pontos pertencentes ao espaço de trabalho contribuem para o cálculo da área:

$$A = \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta_r \Delta_z) \quad (47)$$

A coordenada do baricentro é calculada considerando a soma dos baricentros de cada área elementar, dividida pela área total, dada por:

$$r_g = \frac{\sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta_r \Delta_z) \left((i-1) \Delta_r + \frac{\Delta_r}{2} + r_{\min} \right)}{A} \quad (48)$$

Finalmente, conhecendo-se os valores da área e do baricentro da seção radial, dados pelas Equações 47 e 48, pode-se calcular o volume do espaço de trabalho do manipulador usando a Equação 42. Desta forma, a Equação 42 representa a função objetivo a ser maximizada para obter o maior volume do espaço de trabalho possível.

5.3.3 Rigidez do Mecanismo Serial

Pode-se definir rigidez como sendo a capacidade de um sistema mecânico de suportar cargas sem grandes mudanças em sua geometria. Assim, a rigidez é uma característica mecânica que descreve o comportamento de uma estrutura sujeita às forças estáticas em termos da deflexão elástica. Desta forma, o estudo da rigidez de um sistema equivale a obter a matriz de rigidez, K , da estrutura analisada, que representa a medida da capacidade da estrutura de resistir às deformações devido à ação de esforços externos. Neste trabalho, o modelo de rigidez da estrutura é obtido a partir da matriz Jacobiana do mecanismo serial, considerando os elementos como sendo molas. A matriz de rigidez do sistema de juntas do mecanismo no espaço cartesiano é dada por:

$$K = [J]^T K_j [J] \quad (49)$$

onde K_j é a matriz diagonal $n \times n$, para o caso estudado, mecanismo serial com 3 juntas rotacionais, tem-se $K_j = \text{diag}[k_1, k_2, k_3]$ e J a matriz Jacobiana, dada na Equação 39.

Além disso, os elementos da diagonal da matriz de rigidez são utilizados como os valores da rigidez do sistema. Estes elementos representam a rigidez pura em cada direção, e refletem a rigidez/inflexibilidade das ferramentas da máquina de forma mais clara e direta. Assim, a função objetivo para otimizar a rigidez do sistema pode ser escrita conforme a Equação 50. Neste caso, a rigidez R deve ser maximizada:

$$R = n_1 K_{11} + n_2 K_{22} + n_3 K_{33} \quad (50)$$

onde $K_{ii}, i = 1, 2, 3$, representa os elementos da diagonal da matriz de rigidez do mecanismo, n_i é o fator de peso para cada rigidez direcional que caracteriza a prioridade da rigidez nesta direção.

5.3.4 Destreza

O índice de desempenho de um sistema mecânico robótico é uma quantidade escalar que mede o quão satisfatório o sistema se comporta com relação à transmissão de movimento e força. Este índice pode ser definido para todos os tipos de sistemas mecânicos robóticos, particularmente, os manipuladores seriais. Neste trabalho, será adotado o índice de isotropia, $k(J)$, que é definido como o índice de condicionamento da matriz Jacobiana J . Tal índice pode ser escrito como a relação entre o maior e o menor valor singular de J :

$$k(J) = \frac{|\lambda_{\max}(J)|}{|\lambda_{\min}(J)|} \quad (51)$$

sendo que $\lambda_{\max}$ e $\lambda_{\min}$ significam, respectivamente, os valores singulares máximo e mínimo de J .

Na robótica, o índice de isotropia reflete a precisão resultante das velocidades operacionais de entrada a partir das velocidades articulares calculadas usando a inversa de J . Este índice pode alcançar valores de 1 a infinito.

O valor 1 significa uma isotropia: o elipsóide das velocidades toma a forma de uma esfera. Fisicamente, quando o manipulador está em uma posição isotrópica, o efetuador terá a mesma facilidade para se deslocar em todas as direções. Assim, manipuladores isotrópicos são aqueles cuja Jacobiana pode alcançar valores isotrópicos, neste caso, $k(J) = 1$. Quanto mais próximo de 1, maior a agilidade para executar tarefas no espaço de trabalho.

5.3.5 Otimização Multi-Objetivo

Em muitas situações, várias funções objetivo precisam ser minimizadas e/ou maximizadas simultaneamente, e neste caso, o problema é denominado problema de otimização multi-objetivo (ou multicritério).

A função objetivo do problema pode ser reescrita como um vetor $f(x)$, cujas componentes são k funções objetivos que se deseja maximizar e/ou minimizar. Esta função vetorial pode ser escrita como:

$$\text{Maximizar ou Minimizar } f(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T \quad (52)$$

Existem vários métodos para resolver este tipo de problema (Oliveira e Saramago, 2010). Neste trabalho será utilizado o Método do Critério Global. Esta técnica consiste em transformar o vetor de funções objetivo em uma função escalar.

Neste método, a solução ótima é um vetor de variáveis de projeto que minimiza algum critério global. A função que descreve este critério deve ser definida pelo projetista de modo que se obtenha uma solução mais próxima possível da solução ideal. Tal função global pode ser escrita como uma família de Métricas- L_p , definida como:

$$L_p = \left[\sum_{i=1}^k |f_i^0 - f_i(x)|^s \right]^{1/s}, \quad 1 \leq s \leq \infty \quad (53)$$

Em vez de trabalhar com a distância no sentido absoluto, é recomendado o uso de distâncias relativas, assim a Equação 53 pode ser reescrita como:

$$L_p = \left[\sum_{i=1}^k \left| \frac{f_i^0 - f_i(x)}{f_i^0} \right|^s \right]^{1/s}, \quad 1 \leq s \leq \infty \quad (54)$$

Utilizando a métrica- L_2 relativa, têm-se as funções escalares dadas, respectivamente, por:

$$L_{2R} = \left(\left(\frac{f_1^0 - f_1(x)}{f_1^0} \right)^2 + \dots + \left(\frac{f_k^0 - f_k(x)}{f_k^0} \right)^2 \right)^{1/2} \quad (55)$$

Assim, na Equação 55, a função vetorial foi normalizada pelo uso do conceito de solução ideal f_k^0 . Esta solução é determinada obtendo-se separadamente o ótimo viável, para todas as funções objetivo. Em outras palavras:

$$f_k^0 = \min f_k(x), \quad k = 1, \dots, K \quad (56)$$

5.3.6 Resultados

Utilizando o método de otimização Evolução Diferencial para resolver o problema do manipulador 3R ortogonal foi encontrado, após 29,61 minutos de execução do programa, o valor ótimo $X = [1 \ 3.0 \ 3.0 \ 3.0 \ 0.1]^T$, sendo que $V = 1896,66$, $D = 1.0413$ e $R = 105.6675$, donde $F(X) = [1896,66 \ -1.0413 \ 105.6675]$. A Figura 12 apresenta o gráfico da área da seção transversal do espaço de trabalho do robô manipulador.

Para o método Evolução com Conjuntos Embaralhados foram feitas simulações para 2, 4 e 10 conjuntos. A tabela 6 apresenta os valores obtidos, sendo que t representa o tempo de processamento em minutos e a Figura 13 os gráficos para a área da seção transversal do espaço de trabalho do robô manipulador para os três casos.

Comparando os dados obtidos com ED e ECE nota-se que os resultados são muito parecidos, porém para este problema o método da ED alcançou o ótimo mais rápido, o que não significa que este método é melhor, uma vez que a diversidade da população do ED é menor o que pode levar o algoritmo à um mínimo local, o que não ocorreu neste caso. Sabendo que os valores ideais do volume V , da destreza D e da rigidez R são, respectivamente, $V_{id}=2382,7412$; $D_{id}=1,0256$ e $R_{id}=125,0769$ conclui-se que os métodos foram eficientes em resolver o problema já que os resultados encontrados se aproximam dos ideais. Visto que o ECE é um algoritmo mais robusto que o ED é natural que para

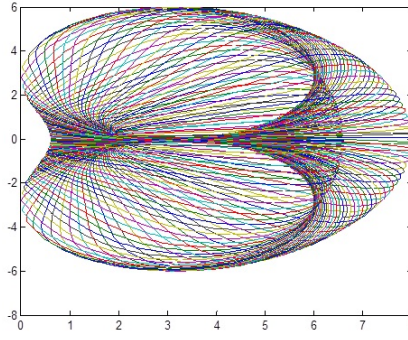


Figura 12: Área da seção transversal do espaço de trabalho do robô manipulador

Variáveis de projeto	2 Conjuntos t=46 min	4 Conjuntos t=45 min	10 Conjuntos t=111 min
d_3	2,9999	3	3
d_4	3	3	3
r_2	2,9999	2,9997	2,9999
r_3	0,1434	0,14445	0,14258
$f(X)$	V=1902,23	V= 1902,59	V=1902,63
	D= 1,0526	D= 1,0530	D= 1,0524
	R= 105,6678	R=105,6760	R=105,6771

Tabela 6: Resultados do problema do manipulador 3R ortogonal para ECE.

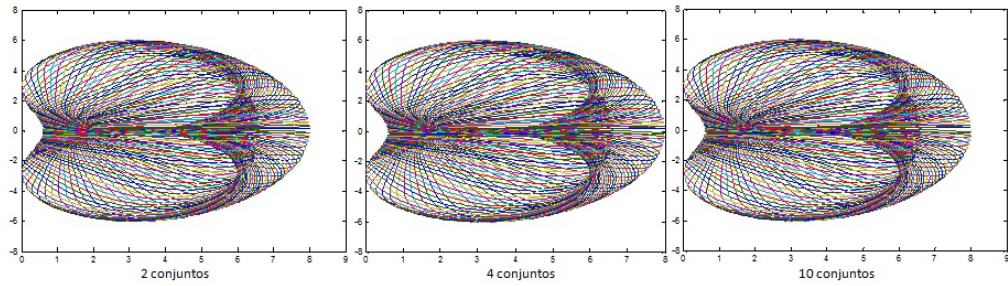


Figura 13: Área da seção transversal do espaço de trabalho do robô manipulador

problemas “maiores” aquele precise de mais tempo ou de mais esforço computacional que este.

6. Conclusão

Em um mundo em que tempo é igual a dinheiro, quanto mais rápido o algoritmo conseguir resolver um problema melhor. Por isso, pretende-se modificar o algoritmo básico de Evolução Diferencial desenvolvendo o método denominado Evolução Diferencial Melhorada (EDM) - Improved Differential Evolution - que usa o conceito da Evolução com Conjuntos Embaralhados (ECE) - Shuffled Complex Evolution - (Duan 1993). Em EDM a população inicial de indivíduos é particionada em várias subpopulações, e então o algoritmo da ED será aplicado a cada subpopulação de forma independente. Após um determinado número de gerações, a população total é embaralhada e, em seguida, os pontos são redistribuídos para as subpopulações. Esta alteração ajuda a aumentar a exploração das capacidades do algoritmo ED.

Assim o objetivo deste estudo é combinar o que há de melhor nos métodos da Evolução Diferencial e da Evolução com Conjuntos Embaralhados e implementá-los em paralelo, pois assim, cada conjunto do método ECE evoluirá em um processador diferente e ao mesmo tempo, seguindo o princípio de que grandes problemas geralmente podem ser divididos em problemas menores, que então são resolvidos concorrentemente (em paralelo).

A idéia de se dividir tarefas de programas por vários processadores é antiga mas só recentemente vem se tornando viável devido ao rápido avanço de *hardware* e *software*. As máquinas evoluíram muito em velocidade e com o aparecimento de máquinas com vários processadores; a velocidade de comunicação vem apresentando uma grande evolução permitindo que estes processadores troquem informações com rapidez; e os programas que possibilitam esta comunicação vêm mostrando grande aumento de eficiência. Dessa forma, pretende-se diminuir o tempo de processamento durante a execução do algoritmo melhorado, para permitir sua aplicação em problemas complexos.

7. Agradecimentos: À CAPES e à FAPEMIG pelo auxílio/bolsa concedidos durante a realização deste trabalho e das diversas pesquisas do grupo.

Referências

- [1] BRANDÃO, M.A.L., SARAMAGO, S.F.P., 2011, Métodos Estocásticos de Otimização: Algoritmos Genéticos e Evolução Diferencial. Notas em Matemática Aplicada-SBMAC.
- [2] COELLO C. A. C. (2000) *Use of a self-adaptive penalty approach for engineering optimization problems*. Computers in Industry, vol. 41, no. 2, pp. 113-127.
- [3] DEB, K. (1997) *GenaAS: a robust optimal design technique for mechanical component design*, Dasgupta, D. and Michalewicz 2 (eds). Evolutionary Algorithms in Engineering Applications Berlin, pp. 497-514.
- [4] DUAN, Q. A., GUPTA, V. K. and SOROOSHIAN, S., *Shuffled Complex Evolution Approach for Effective and Efficient Global Minimization*, Journal of Optimization Theory and Applications: Vol. 76, N° 3, March 1993.
- [5] HOLLAND, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.
- [6] Hu, X.; Eberhart, R.; Shi, Y. (2003) *Engineering optimization with particle swarm*, IEEE Conference on Swarm Intelligence, Indianapolis, Indiana.
- [7] NELDER, J. A., and MEAD, R., *A Simplex Method for Function Minimization*, Computer Journal, Vol. 7, pp. 308-313, 1965.
- [8] OLIVEIRA, L. S., SARAMAGO, S. F. P., 2010, Multiobjective Optimization Techniques Applied to Engineering Problems. Journal of the Brazilian Society of Mechanical Sciences and Engineering (Impresso). , v.XXXII, p.94 - 104.
- [9] PIRSIG, R.M., *Zen and the Art of Motorcycle Maintenance: An Inquiry into Values*, New York: Quill. 25th Anniversary Edition, 1974.
- [10] PRICE, W. L., *A Controlled Random Search Procedure for Global Optimization*, Toward Global Optimization 2, Edited by L. C. W. Dixon and G. P. Szeg6, North-Holland, Amsterdam, Holland, pp. 71-84, 1978.
- [11] PRICE, W. L., *Global Optimization by Controlled Random Search*, Journal of Optimization Theory and Applications, Vol. 40, pp. 333-348, 1983.
- [12] PRICE, W. L., *Global Optimization Algorithms for a CAD Workstation*, Journal of Optimization Theory and Applications, Vol. 55, pp. 133-146, 1987.
- [13] SARAMAGO, S. F. P., BERGAMASCHI, P. R., OLIVEIRA, G. T. S., 2007, Optimization of the Workspace Volume of 3R Manipulators Using Hybrid Methodologies In: 19th International Congress of Mechanical Engineering, Cobem 2007. ABCM, v.1. p.1 - 10.