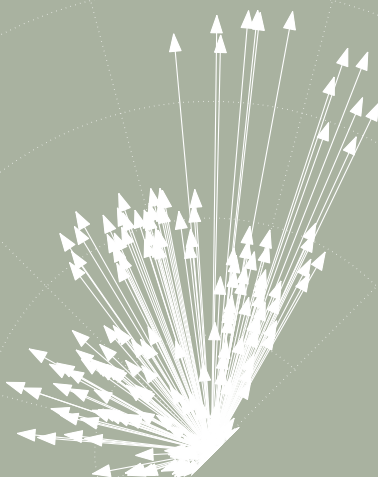


MANUAL DE
**COMPUTAÇÃO
EVOLUTIVA
E META
HEURÍSTICA**

ANTÓNIO GASPAR-CUNHA
RICARDO TAKAHASHI
CARLOS HENGGELER ANTUNES
COORDENADORES



IMPrensa DA
UNIVERSIDADE
DE COIMBRA

COIMBRA
UNIVERSITY
PRESS

(EDITORAufmg)

CAPÍTULO 3

Estratégias Evolutivas

*Lino Costa **

*Pedro Oliveira ***

**Departamento de Produção e Sistemas
Universidade do Minho*

***Instituto de Ciências Biomédicas Abel Salazar
Universidade do Porto*

Na natureza, de acordo com a Teoria da Evolução de Darwin, a evolução dos seres vivos ocorre devido à selecção natural e à adaptação ao ambiente. A selecção natural faz com que os seres vivos mais aptos, em relação ao ambiente, tenham maior probabilidade de sobrevivência. Por outro lado, para que seja possível a adaptação dos seres vivos a um ambiente continuamente em mudança, surgiram mecanismos na natureza que tornam possível a diversidade, isto é, a existência de seres vivos com características próprias que, potencialmente, os podem tornar mais aptos. A contínua evolução das espécies pode ser vista como um processo de optimização consistindo na adaptação dos seres vivos ao seu meio ambiente. Este modelo biológico inspirou o desenvolvimento de diversos Algoritmos Evolucionários¹, na década de 60 do século XX, nomeadamente, na Alemanha, as Estratégias Evolutivas (Rechenberg, 1973; Schwefel, 1995) e, nos Estados Unidos da América, os Algoritmos Genéticos (Holland, 1975;

¹ O termo Algoritmo Evolucionário é utilizado, genericamente, na descrição de sistemas computacionais para a resolução de problemas que utilizam como elementos chave na sua implementação, modelos computacionais baseados em mecanismos de evolução. De referir que estes mecanismos de evolução correspondem ou relacionam-se com processos evolutivos biológicos.

Goldberg, 1989) para serem aplicados a problemas de optimização².

As Estratégias Evolutivas (EEs) foram desenvolvidas com o objectivo de serem aplicadas em optimização numérica, revelando-se como algoritmos de optimização robustos e eficientes. As EEs são algoritmos iterativos que pesquisam com base em populações de indivíduos, que representam potenciais soluções para o problema de optimização. Cada iteração é, por analogia com os sistemas biológicos, chamada de *geração*. Em cada geração, os operadores genéticos (*recombinação* e *mutação*) actuam sobre a população de indivíduos permitindo a exploração do espaço de procura e controlar a diversidade dos indivíduos da população. O princípio de sobrevivência do mais apto é implementado pela *selecção* que garante a sobrevivência dos indivíduos que representam soluções de maior qualidade para o problema. As condições que se devem verificar para terminar o processo iterativo (e.g., a obtenção de uma solução de qualidade suficiente para a resolução do problema) constituem os chamados critérios de paragem. Uma boa introdução às EEs, focando, com alguma profundidade, os fundamentos teóricos, é feita por Beyer e Schwefel (2002).

1. Optimização em Espaços Contínuos

Um grande número de aplicações de EEs a problemas de optimização com espaços de procura discretos e contínuos encontram-se descritas na literatura (Rechenberg, 1964, 1994; Schwefel, 1995; Bäck, 1996). No entanto, as EEs serão aqui apresentadas como algoritmos para a resolução de problemas de programação não linear em espaços de procura contínuos (Nocedal e Wright, 1999):

$$\min f(\mathbf{x}) \text{ onde } \mathbf{x} \in \Omega \quad (3.1)$$

sujeito a

$$g_j(\mathbf{x}) \geq 0 \text{ com } j = 1, \dots, m$$

$$h_i(\mathbf{x}) = 0 \text{ com } i = m + 1, \dots, m + p$$

onde \mathbf{x} é o vector das *variáveis de decisão*; $f(\mathbf{x})$ é a *função objectivo*, i.e., a função que se pretende minimizar³; $\mathbf{g}(\mathbf{x})$ é o vector das *restrições do tipo desigualdade* que devem ser satisfeitas; $\mathbf{h}(\mathbf{x})$ é o vector das *restrições do tipo igualdade* que devem ser satisfeitas.

Na formulação apresentada existem n variáveis reais, m restrições do tipo desigualdade e p restrições do tipo igualdade (o número total de restrições é $m + p$). O *espaço das variáveis*, $\Omega \subseteq \mathbb{R}^n$, corresponde ao conjunto de todos os valores possíveis para as variáveis de decisão. A procura da solução de um problema de optimização, o ponto *ótimo* \mathbf{x}^* , é feita no espaço das variáveis⁴. As restrições de desigualdade são expressas em termos de desigualdades do tipo maior ou igual (\geq)⁵. Muitas vezes, poderão existir restrições de desigualdade especificando limites inferiores (\underline{x}_k) e superiores (\overline{x}_k) para as n variáveis x_k , i.e., $\underline{x}_k \leq x_k \leq \overline{x}_k$, para $k = 1, \dots, n$.

Qualquer ponto $\mathbf{x} \in \Omega$ diz-se satisfazer uma restrição se, para essa restrição, o lado esquerdo da expressão calculado nesse ponto está de acordo com o lado direito, em termos do operador relacional.

² De notar que estas duas abordagens não são as únicas inspiradas na natureza para resolver problemas de optimização, existindo outras, tais como, a Programação Evolucionária (Fogel et al., 1966), a Programação Genética (Koza, 1992), os algoritmos de Colónia de Formigas (Dorigo et al., 1996) e os algoritmos de Enxames de Partículas (Kennedy e Eberhart, 1995).

³ De notar que qualquer problema formulado em termos de maximização de uma função objectivo $f(\mathbf{x})$ pode ser reformulado da seguinte forma: $\max f(\mathbf{x}) = -\min(-f(\mathbf{x}))$.

⁴ Em geral, os problemas de optimização são abordados pressupondo-se que o ponto ótimo \mathbf{x}^* existe, é único, e pode ser localizado utilizando um algoritmo de optimização. Apesar de muitas vezes este ser o caso, existem situações em que tal não se verifica: se $f(\mathbf{x})$ não é limitada inferiormente, então \mathbf{x}^* não existe ou, para certas funções objectivo, \mathbf{x}^* poderá não ser único.

⁵ Qualquer restrição expressa em termos da desigualdade menor ou igual pode ser facilmente transformada em termos da desigualdade maior ou igual bastando para isso multiplicá-la por -1 .

Um ponto é dito *ponto admissível* se todas as restrições do tipo desigualdade e restrições do tipo igualdade forem satisfeitas nesse ponto. O conjunto de todos os pontos admissíveis constitui a *região admissível* e pode ser definido da seguinte forma:

$$A = \{\mathbf{x} \in \Omega : \mathbf{g}(\mathbf{x}) \geq 0 \wedge \mathbf{h}(\mathbf{x}) = 0\}.$$

Todos os pontos que não pertencem ao conjunto A são pontos não admissíveis. O óptimo \mathbf{x}^* , a solução do problema, pertence necessariamente à região admissível⁶.

2. Características Gerais

Existem diferenças importantes relativamente à forma como as EEs e os algoritmos de optimização tradicionais (não evolucionários) resolvem os problemas de optimização contínua formulados em (3.1). Tal como as EEs, os algoritmos de optimização tradicionais são iterativos. Em geral, iniciam a procura a partir de uma aproximação inicial ao óptimo que se pretende encontrar. A partir da aproximação inicial são geradas, sucessivamente, novas estimativas do óptimo. O processo iterativo é repetido até que os critérios de paragem sejam verificados. Os diversos algoritmos de optimização distinguem-se pela forma como são calculadas as novas estimativas ao longo da procura. Quase todas as abordagens utilizam os valores da função objectivo, das restrições e, muitas vezes, das primeiras e segundas derivadas destas funções. Alguns algoritmos, para calcular uma nova aproximação ao óptimo, utilizam apenas a informação relativa à aproximação actual, enquanto que outros consideram, também, a informação recolhida durante as iterações passadas. Os algoritmos de optimização tradicionais podem ser caracterizados, em grande parte, pela descrição anterior. No entanto, convém separar os algoritmos tradicionais em dois grupos principais: os métodos directos e os métodos baseados em gradientes. Os métodos directos guiam a procura com base em informação relacionada com a função objectivo e/ou restrições. Enquanto que os métodos baseados em gradientes, para além desta informação, necessitam, também, das primeiras e/ou segundas derivadas. Por este motivo, em geral, os métodos baseados em gradientes não são eficientes quando, num problema, não se verificam as condições de diferenciabilidade e/ou continuidade das funções. De salientar que, nestes métodos, os resultados obtidos dependem grandemente das aproximações ao óptimo consideradas no início do processo iterativo e não são eficientes na resolução de problemas com espaços de procura de natureza discreta.

Em contraste com estas abordagens ditas tradicionais, as EEs:

- iniciam a procura a partir de uma população de potenciais soluções geradas de forma aleatória (caso seja conhecida podem partir de uma aproximação inicial);
- trabalham, ao longo das gerações, com populações de potenciais soluções, em vez de uma única aproximação ao óptimo por iteração;
- não utilizam nenhuma informação relativa às primeiras e/ou segundas derivadas da função objectivo e/ou restrições;
- não exigem nenhuma condição relativa à continuidade e convexidade do espaço de procura;
- podem utilizar mecanismos que permitem encontrar múltiplos óptimos locais (se existirem) numa única execução;

⁶ Quando um ponto satisfaz uma restrição j do tipo desigualdade, duas situações podem ocorrer: o ponto está no limite da região admissível da restrição, i.e., $g_j(\mathbf{x}) = 0$ e neste caso a restrição j diz-se *activa*; ou o ponto está no interior da região admissível da restrição, i.e., $g_j(\mathbf{x}) > 0$ e neste caso a restrição j diz-se *inactiva*. Para qualquer ponto admissível \mathbf{x} , todas as p restrições do tipo igualdade estão activas.

- guiam a procura com base em mecanismos e/ou regras probabilísticas (estocásticas).

Estas características das EEs fazem com que sejam particularmente eficientes na resolução de problemas reais, onde os espaços de procura podem não ser convexos e, por isso, muitas vezes contêm um grande número de óptimos locais e/ou mais do que uma solução óptima. Por outro lado, muitos problemas reais apresentam espaços de procura de natureza discreta onde não é possível garantir as condições de diferenciabilidade e continuidade desejáveis para muitos dos algoritmos tradicionais.

3. Nomenclatura

As EEs foram desenvolvidas, inicialmente, por Rechenberg (1973), com o objectivo de resolver problemas de optimização, tendo como base as estruturas e os processos de optimização que ocorrem na natureza. Mais tarde, Schwefel (1981) desenvolveu novos esquemas evolutivos com base nos mesmos princípios.

Nas EEs é utilizada uma população com um determinado número de indivíduos. Em cada geração, a partir dos indivíduos presentes na população, os *progenitores*, são gerados novos indivíduos, os *descendentes*. Desta forma, ao longo das gerações, novas populações são sucessivamente geradas. As EEs trabalham directamente com a representação real das variáveis de decisão, pelo que cada indivíduo é um vector de números reais representando uma potencial solução para o problema de optimização.

Desde o seu aparecimento, tem vindo a ser utilizada uma nomenclatura própria para designar as diferentes EEs. Esta nomenclatura é baseada no número de progenitores, no número de descendentes e no tipo de selecção considerado. O número de progenitores é designado por μ e o número de descendentes por λ . Por outro lado, dois tipos de selecção foram originalmente descritos e designados por selecção '+' e selecção ','. A primeira e mais simples EE, desenvolvida por Rechenberg (1973), onde a selecção era feita sobre uma população de dois membros, i.e., $\mu + \lambda = 1 + 1$, é designada, na nomenclatura atrás apresentada, por EE-(1 + 1). Posteriormente, o mesmo autor desenvolveu uma estratégia multimembros mais complexa, onde a selecção era feita sobre uma população de $\mu > 1$ indivíduos e um descendente; i.e., $\mu + \lambda = \mu + 1$, que é designada por EE-($\mu + 1$).

De uma forma mais genérica, numa EE-($\mu + \lambda$), numa determinada geração, existe uma população de μ progenitores que gera λ descendentes por mutação (Figura 3.1). Em seguida, os $\mu + \lambda$ indivíduos são avaliados e ordenados de acordo com os seus valores da função objectivo⁷. Finalmente, o processo de selecção faz com que apenas os μ melhores de todos os $\mu + \lambda$ indivíduos se tornem os progenitores da geração seguinte, i.e., a selecção é feita a partir dos $\mu + \lambda$ indivíduos. De referir que este mecanismo de selecção é determinístico, uma vez que apenas os melhores indivíduos são seleccionados para formarem a população de progenitores da geração seguinte.

Um outro modelo conceptual pode ser definido onde, numa determinada geração, uma população de μ progenitores gera, por mutação, λ descendentes (assumindo que $\lambda > \mu$). Em seguida, os λ descendentes são avaliados e ordenados de acordo com os seus valores da função objectivo. Então, os μ melhores dos λ descendentes gerados tornam-se os progenitores da próxima geração, i.e., os μ progenitores não são incluídos no processo de selecção. Esta EE que utiliza a selecção ',' é designada por EE-(μ, λ) (Figura 3.2). Os dois tipos de EEs atrás descritas, a EE-($\mu + \lambda$) e a EE-(μ, λ), diferem basicamente no procedimento de selecção.

Originalmente, as EEs baseavam-se num único operador, a mutação, para gerar novos indivíduos. Posteriormente, foi introduzido um outro operador, a recombinação, que era aplicado juntamente com a mutação (Schwefel, 1995). Mais adiante, nesta mesma secção, descrever-se-á detalhadamente este operador.

⁷ Assumindo que os indivíduos representam pontos admissíveis. Mais tarde, serão abordados diversos mecanismos de tratamento de restrições.

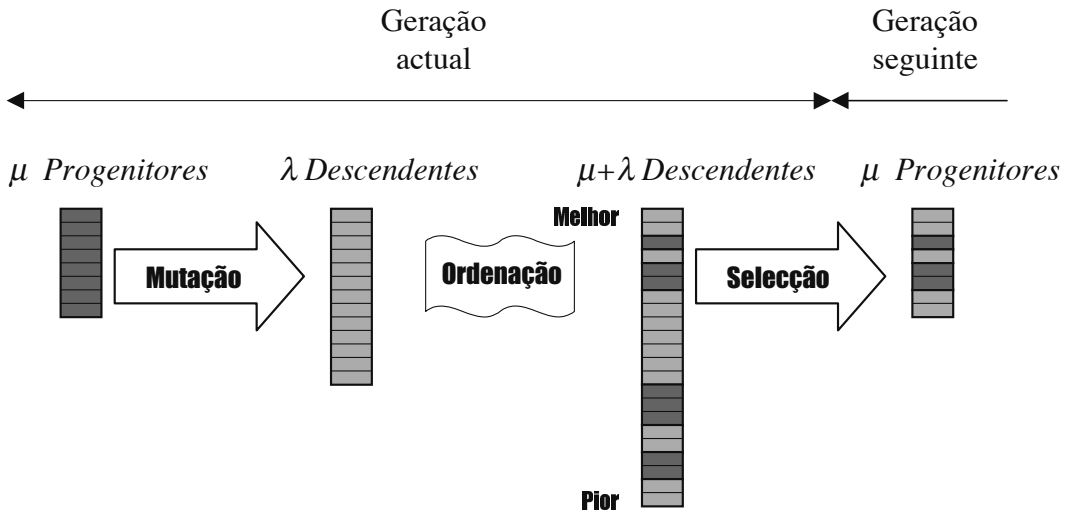


Figura 3.1: Aspecto Geral da Estratégia Evolutiva $(\mu + \lambda)$

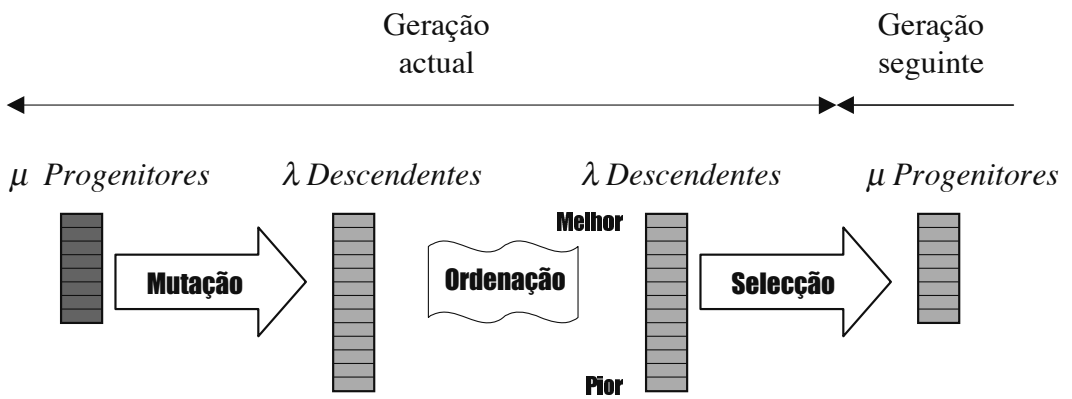


Figura 3.2: Aspecto Geral da Estratégia Evolutiva (μ, λ)

Em seguida, cada uma das EEs atrás referidas irá ser descrita com maior detalhe. Todos os algoritmos apresentados são formulados para problemas de minimização sem restrições. Mais adiante, ir-se-á focar diversos esquemas de tratamento de restrições quer do tipo desigualdade quer do tipo igualdade.

4. Estratégia Evolutiva (1+1)

Este esquema evolutivo com apenas 2 membros, designado por EE-(1 + 1), foi proposto por (Rechenberg, 1964) para a resolução de problemas de optimização. Numa determinada geração, existe apenas um progenitor ($\mu = 1$) e um único descendente ($\lambda = 1$), e a selecção tem lugar apenas entre estes dois membros.

O funcionamento da EE-(1 + 1) pode ser descrito pelo Algoritmo 1. A procura inicia-se a partir de um ponto inicial \mathbf{x}_0 (uma aproximação ao óptimo). Em seguida, um novo ponto \mathbf{x}_N é gerado por mutação através da adição de uma quantidade aleatória normal com média 0 e variância σ^2 . O desvio padrão σ está relacionado com o tamanho do passo e vai sendo adaptado ao longo da procura. Depois, os dois pontos são comparados e o melhor (com menor valor da função objectivo e satisfazendo todas as restrições) é seleccionado para progenitor da próxima geração. Este processo é repetido até que o critério de paragem seja verificado.

De seguida, vai-se analisar com mais detalhe cada um dos passos envolvidos neste algoritmo.

Algoritmo 1 Estratégia Evolutiva (1 + 1)

Require: $\mathbf{x}_0, \Delta x$

```

1:  $t \leftarrow 0$ 
2:  $\mathbf{x}^{(t)} \leftarrow \mathbf{x}_0$  [ onde  $\mathbf{x}^{(t)} = (x_1, x_2, \dots, x_n)^{(t)}$  ]
3:  $\sigma \leftarrow \frac{|\Delta x|}{\sqrt{n}}$ 
4: para  $i = 1$  to  $10n$  faça
5:    $\text{sucessos}[i] \leftarrow 0$ 
6: fim para
7: enquanto CP falso faça [ // é o resto da divisão inteira ]
8:   se  $(t//n = 0 \wedge t \geq 10n)$  então [ Regra de 1/5 sucessos ]
9:      $P_s \leftarrow \frac{\sum_{i=1}^{10n} \text{sucessos}[i]}{10n}$ 
10:     $\sigma \leftarrow \begin{cases} \sigma/c & \text{se } P_s > 1/5 \\ c\sigma & \text{se } P_s < 1/5 \\ \sigma & \text{se } P_s = 1/5 \end{cases}$  com  $c = 0.85$ 
11:   fim se
12:    $\mathbf{x}_N \leftarrow \mathbf{x}^{(t)} + \mathbf{N}(0, \sigma^2)$  [ Mutação ]
13:   se  $(f(\mathbf{x}_N) < f(\mathbf{x}^{(t)}))$  então [ Selecção ]
14:      $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}_N$ 
15:      $\text{sucessos}[(t+1)//10n] \leftarrow 1$ 
16:   senão
17:      $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)}$ 
18:      $\text{sucessos}[(t+1)//10n] \leftarrow 0$ 
19:   fim se
20:    $t \leftarrow t + 1$ 
21: fim enquanto
22:  $\mathbf{x}_{final} \leftarrow \mathbf{x}^{(t)}$ 
23: retorno  $\mathbf{x}_{final}$ 

```

Aproximação Inicial

Como já foi dito, para iniciar a procura é necessária uma aproximação inicial ao ótimo \mathbf{x}_0 . Mas, para além disso, é necessário escolher os valores iniciais para o desvio padrão σ . O valor inicial típico para o desvio padrão σ pode ser expresso pela seguinte equação:

$$\sigma = \frac{\Delta x}{\sqrt{n}},$$

onde Δx é uma medida aproximada da distância esperada da aproximação inicial \mathbf{x}_0 ao ótimo e n é a dimensão do problema (o número de variáveis de decisão).

Ao longo da procura, devem-se garantir as seguintes duas condições para os valores dos tamanhos do passo σ :

1. $\sigma > 0$ para $i = 1, \dots, n$;
2. os valores de σ devem ser suficientemente grandes para que pelo menos o dígito menos significativo da variável x_i seja alterado.

As condições anteriores podem ser escritas em termos dos seguintes limites inferiores para os tamanhos do passo: $\sigma \geq \varepsilon_1$ e $\sigma \geq \varepsilon_2 |x_i|$ para $i = 1, \dots, n$, onde ε_1 e ε_2 dependem da precisão do computador utilizado, sendo $\varepsilon_1 > 0$ e $\varepsilon_2 > 0$.

Mutação

O operador de mutação consiste em gerar um novo ponto \mathbf{x}_N através da adição de uma quantidade aleatória \mathbf{z} . A quantidade aleatória \mathbf{z} traduz o tamanho do passo (ou comprimento do deslocamento), e a sua escolha deve ser feita de tal forma que pequenos deslocamentos ocorram frequentemente e grandes deslocamentos ocorram raramente. Com este propósito, em geral, as quantidades aleatórias \mathbf{z} são geradas de acordo com uma distribuição Normal. Para além disso, as seguintes condições devem ser impostas à distribuição dos tamanhos do passo:

1. O valor esperado dos componentes z_i de \mathbf{z} , com $i = 1, \dots, n$, deve ser nulo, i.e., $E[z_i] = 0$;
2. As variâncias σ^2 , com $i = 1, \dots, n$, devem ter valores pequenos.

Logo, os componentes aleatórios z_i podem ser calculados de acordo com uma distribuição Normal com média nula e variância σ^2 , i.e., $z_i \sim N(0, \sigma^2)$. Para determinar números aleatórios normais a partir de número aleatórios uniformes, podem ser utilizadas as regras de transformação de Box e Muller (1958)⁸.

Controlo do Tamanho do Passo

Para que o processo de optimização seja eficiente, os deslocamentos devem ser continuamente modificados. Se os deslocamentos forem demasiado pequenos, o número de iterações do processo de procura é desnecessariamente grande; pelo contrário, se forem demasiado grandes, poderá não se obter uma boa aproximação ao ótimo, ou mesmo, o processo poderá não convergir. Por isso, em todas as estratégias de optimização, o controlo do tamanho do passo é uma das componentes mais importantes no processo de procura.

⁸ De acordo com estas regras, dois números aleatórios independentemente normalmente distribuídos com média nula e variância unitária podem ser calculados partir de quaisquer dois números aleatórios gerados uniformemente no intervalo $[0, 1]$ da seguinte forma: $x_1 = \sqrt{-2 \ln(y_1)} \sin(2\pi y_2)$ e $x_2 = \sqrt{-2 \ln(y_1)} \cos(2\pi y_2)$ onde y_1 e y_2 são os dois números aleatórios gerados uniformemente no intervalo $[0, 1]$ e, x_1 e x_2 são os dois números aleatórios normalmente distribuídos com média nula e variância unitária. Para se obter dois números aleatórios z_1 e z_2 normalmente distribuídos com média nula e variância σ^2 , basta multiplicar x_1 e x_2 pelo desvio padrão σ , i.e., $z_1 = \sigma x_1$ e $z_2 = \sigma x_2$.

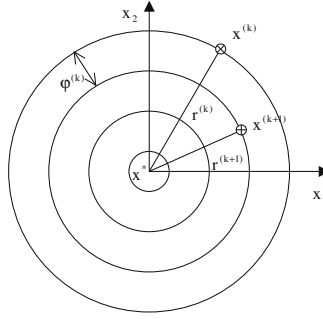


Figura 3.3: Razão de Progresso

Nas EEs, a grandeza das quantidades aleatórias z_i depende do desvio padrão σ . Quanto maior for o desvio padrão σ , maiores serão as quantidades aleatórias z_i e, conseqüentemente, os tamanhos do passo. Rechenberg (1973), baseando-se na aplicação da EE-(1 + 1) a alguns tipos de problemas de otimização, formulou a chamada "Regra de 1/5 de Sucessos" para controlar os tamanhos do passo. Esta regra pode ser descrita da seguinte forma:

"De tempos a tempos, ao longo do processo de procura, calcule-se a frequência de sucessos, i.e., a razão entre o número de sucessos e o número de iterações. Se a razão for maior que 1/5, aumente-se a variância, se for menor que 1/5, diminua-se a variância."

Assumindo-se que esta regra é aplicada periodicamente todas as Δ_t iterações, a sua expressão pode ser definida da seguinte forma para uma iteração k :

$$\sigma^{(k+1)} = \begin{cases} c_{inc} \sigma^{(k)} & \text{se } P_s(\Delta_t) > 1/5 \\ c_{dec} \sigma^{(k)} & \text{se } P_s(\Delta_t) < 1/5 \\ \sigma^{(k)} & \text{se } P_s(\Delta_t) = 1/5 \end{cases},$$

onde $P_s(\Delta_t)$ é a proporção de sucessos nas últimas Δ_t iterações e, $c_{dec} < 1$ e $c_{inc} > 1$ são, respectivamente, os coeficientes de diminuição e de aumento do desvio padrão σ .

Defina-se a razão de progresso como o valor esperado da diferença radial após uma iteração k (Figura 3.3):

$$\varphi^{(k)} = E \left[\left\| \mathbf{x}^* - \mathbf{x}^{(k)} \right\| - \left\| \mathbf{x}^* - \mathbf{x}^{(k+1)} \right\| \right] = E \left[r^{(k)} - r^{(k+1)} \right],$$

onde \mathbf{x}^* representa o ótimo e, $\mathbf{x}^{(k)}$ e $r^{(k)}$ representam, respectivamente, a aproximação ao ótimo e a distância ao ótimo na iteração k .

Verifica-se que, para muitos problemas, a regra de 1/5 de sucessos mostra-se extremamente eficiente na manutenção, aproximadamente, da maior razão de progresso possível para o ótimo. No entanto, interessa definir qual a frequência com que o critério de sucesso deve ser testado (o valor do parâmetro Δ_t) e qual o factor de diminuição ou de aumento dos desvios padrão mais eficiente (os valores dos coeficientes c_{dec} e c_{inc}). Para tentar responder a esta questão, podem-se utilizar os resultados obtidos por Rechenberg (1973). A máxima razão de progresso é dada por:

$$\varphi_{\max} = c_1 \frac{r^{(k)}}{n} \text{ sendo } c_1 \cong 0.2025,$$

com variância comum σ^2 , cujo valor σ_{opt}^2 ótimo é dado por:

$$\sigma_{opt}^2 = \left(c_2 \frac{r^{(k)}}{n} \right)^2 \text{ com } c_2 \cong 1.224,$$

para todos os componentes $z_i^{(k)}$ do vector aleatório $\mathbf{z}^{(k)}$. Nestas expressões, $r^{(k)}$ é a distância actual ao óptimo e n é o número de variáveis. Destas equações pode-se obter a relação para as variações nos comprimentos dos deslocamentos após uma geração, supondo a condição de máxima razão de convergência:

$$\frac{\sigma_{opt}^{(k+1)}}{\sigma_{opt}^{(k)}} = \frac{r^{(k+1)}}{r^{(k)}} = \frac{r^{(k)} - \varphi_{\max}}{r^{(k)}} = 1 - \frac{c_1}{n},$$

ou após n gerações:

$$\frac{\sigma_{opt}^{(k+n)}}{\sigma_{opt}^{(k)}} = \left(1 - \frac{c_1}{n}\right)^n.$$

Quando n for muito maior que 1, o factor do comprimento do deslocamento tende para uma constante:

$$\lim_{n \rightarrow +\infty} \left(1 - \frac{c_1}{n}\right)^n = e^{-c_1} \cong 0.817 \cong \frac{1}{1.224}.$$

Este resultado aplica-se ao caso limite em que existem muitas variáveis (n grande) e expressa que a razão de progresso é inversamente proporcional ao número de variáveis. O facto da razão de progresso perto do seu máximo ser bastante insensível a pequenas variações das variâncias, juntamente com o facto da probabilidade de sucesso só poder ser determinada pela média de diversas gerações, conduz à seguinte formulação, mais precisa, da regra de 1/5 de sucessos onde n é o número de variáveis de decisão do problema de optimização:

”Após cada n gerações, verifique-se quantos sucessos ocorreram nas últimas $10n$ gerações.

Se este número for inferior a $2n$, multipliquem-se os comprimentos dos deslocamentos pelo factor 0.85; senão, dividam-se os comprimentos dos deslocamentos pelo factor 0.85.”

Esta regra permite que os comprimentos dos deslocamentos ou as variâncias dos deslocamentos aleatórios sejam controlados, fixando-se os seguintes valores $c_{dec} = 0.85$, $c_{inc} = 1/0.85$ e $\Delta_t = 10n$. No entanto, a probabilidade de sucesso não fornece nenhuma indicação de quão apropriadas são as razões das variâncias σ_i^2 relativamente umas às outras (para cada variável), pelo que os comprimentos dos deslocamentos só podem ser todos reduzidos em conjunto ou todos aumentados em conjunto. Por este motivo, em geral, considera-se que $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_n^2 = \sigma^2$.

Critério de Paragem

O critério de paragem determina quando o processo de procura deve ser terminado. Em geral, são utilizadas as condições:

$$\left|f(\mathbf{x}^{(k-\Delta_k)}) - f(\mathbf{x}^{(k)})\right| \leq \varepsilon_3 \text{ ou } \frac{|f(\mathbf{x}^{(k-\Delta_k)}) - f(\mathbf{x}^{(k)})|}{|f(\mathbf{x}^{(k)})|} \leq \varepsilon_4,$$

e $\Delta_k \geq 20n$, onde ε_3 e ε_4 dependem da precisão do computador utilizado, sendo $\varepsilon_3 > 0$ e $\varepsilon_4 > 0$. A condição $\Delta_k \geq 20n$ assegura que, no caso extremo, entre testes da regra de 1/5 de sucessos, os comprimentos dos deslocamentos são reduzidos ou aumentados, pelo menos pelo factor $(0.85)^{\pm 20} \cong (25)^{\mp 1}$, o que evita que a procura seja terminada apenas porque os comprimentos dos deslocamentos são forçados a variar muito frequentemente. Por outro lado, é evidente que o critério de convergência não precisa de ser verificado em todas as gerações. O procedimento usual é testá-lo apenas cada $20n$ gerações.

5. Estratégias Evolutivas Multimembros

Nestas EEs multimembros existem, em cada geração, μ progenitores e λ descendentes. A EE- (μ, λ) distingue-se da EE- $(\mu + \lambda)$ apenas na selecção que tem lugar, no primeiro caso, entre os λ descendentes enquanto que, no segundo caso, se faz a partir dos $\mu + \lambda$ indivíduos (Schwefel, 1995).

Caso seja conhecida uma aproximação ao óptimo, a procura pode ser iniciada gerando μ pontos a partir desse ponto inicial. Caso contrário, a população inicial pode ser gerada de forma aleatória. Em cada geração, λ pontos são gerados por mutação, como resultado da adição de números aleatórios normais \mathbf{z} . O desvio padrão da distribuição normal utilizada na geração das quantidades aleatórias normais tem um papel crucial numa vez que está relacionado com os deslocamentos ocorridos ao longo da procura. Várias regras podem ser adoptadas com o propósito de adaptar estes desvios padrão. Depois, os λ pontos gerados por mutação são ordenados de acordo com os seus valores da função objectivo. Finalmente, os μ melhores pontos são seleccionados para serem os progenitores na geração seguinte. Este processo é repetido até que o critério de paragem seja satisfeito.

De seguida, são apresentados mais detalhes relativos a cada um dos passos do procedimento atrás descrito. Serão, também, apresentados vários algoritmos correspondentes a diferentes variantes de estratégias evolutivas multimembros.

Aproximação inicial

De forma semelhante à EE- $(1+1)$, a procura pode iniciar-se a partir de uma aproximação ao óptimo \mathbf{x}_0 , que permitirá gerar os μ pontos da população inicial⁹. Caso, não seja conhecida nenhuma aproximação ao óptimo, a população inicial pode ser gerada de forma aleatória. Para além disso, é necessário escolher os valores iniciais para parâmetros associados ao esquema de auto-adaptação considerado¹⁰. Caso se conheça uma estimativa da distância ao óptimo, os valores iniciais típicos para o desvio ou os desvios padrão podem ser expressos pela mesma equação indicada para EE- $(1+1)$.

Mutação

O operador de mutação consiste em gerar novos pontos pela adição de quantidades aleatórias, tal como acontecia na EE- $(1+1)$. Contudo, nas EEs multimembros, cada progenitor produz, em média, λ/μ descendentes, de tal forma que λ descendentes sejam gerados. As mesmas condições acerca da normalidade das quantidades aleatórias \mathbf{z} são consideradas neste tipo de estratégias. No entanto, dependendo do tipo de controlo de passo usado para fazer a auto-adaptação, as quantidades aleatórias normais são calculadas de diferentes formas como é descrito mais adiante.

Recombinação

Schwefel (1995) notou uma aceleração considerável na procura, bem como a facilitação da adaptação dos desvios padrão pela introdução do operador de recombinação nas EEs. Basicamente, a recombinação consiste em, antes da mutação, recombinar um conjunto de progenitores por forma a encontrar uma nova solução. Seja ρ o número de progenitores que participam na recombinação ($1 \leq \rho \leq \mu$). Estes ρ progenitores são escolhidos aleatoriamente de entre os μ indivíduos da população. Quando $\rho = 1$ então não existe recombinação. Logo, a nomenclatura das EEs pode ser estendida, e as EEs com recombinação são, em geral, referidas por EE- $(\mu/\rho + \lambda)$ ou EE- $(\mu/\rho, \lambda)$. Existem dois tipos de recombinação principais:

⁹ Pode-se utilizar a aproximação ao óptimo como a média da distribuição de probabilidade para gerar os restantes pontos da população inicial, e.g., a média da distribuição normal.

¹⁰ Estes parâmetros poderão ser o desvio padrão inicial σ comum a todas as variáveis, ou os desvios padrão iniciais σ_i distintos para cada variável ou ainda as rotações de direcções de procura relacionadas com a matriz de covariâncias.

- a Recombinação Intermédia, e
- a Recombinação Discreta.

Estes tipos de recombinação podem ser aplicados quer às variáveis de decisão quer aos desvios padrão σ . Em seguida, estes tipos de recombinação são descritos em termos das variáveis de decisão. A sua aplicação aos desvios padrão σ faz-se de forma similar.

Recombinação Intermédia

Neste tipo de recombinação, os componentes dos descendentes são obtidos através da média dos componentes dos progenitores correspondentes (escolhidos aleatoriamente a partir da população). Logo, considerando ρ progenitores escolhidos aleatoriamente, os descendentes \mathbf{x}_D serão dados por:

$$\mathbf{x}_D = \frac{1}{\rho} \sum_{m=1}^{\rho} \mathbf{x}_m. \quad (3.2)$$

Se $\rho = \mu$, este esquema tende a gerar descendentes próximos do centróide da população.

Recombinação Discreta

Neste tipo de recombinação, cada componente dos descendentes é escolhido aleatoriamente a partir de um dos ρ progenitores. Logo, supondo ρ progenitores escolhidos aleatoriamente, os descendentes \mathbf{x}_D serão dados por:

$$\mathbf{x}_D = (x_{u_1,1}, \dots, x_{u_n,n}) \text{ com } u_1 \in U(0, \rho), \dots, u_n \in U(0, \rho), \quad (3.3)$$

onde u_i , para $i = 1, \dots, n$, são índices gerados aleatoriamente (uniformemente), indicando a partir de que progenitor (dos ρ progenitores) o valor da variável de decisão é copiado para o descendente. Este procedimento permite obter diferentes combinações dos valores das variáveis de decisão a partir das soluções existentes na população.

Controlo do Tamanho do Passo

Os desvios padrão podem ser actualizados de diversas formas. Este processo consiste na adaptação dos parâmetros da estratégia durante a procura (Schwefel, 1995). As implementações mais conhecidas são:

- a Adaptação Isotrópica,
- a Adaptação Não Isotrópica, e
- a Adaptação Não Isotrópica com Rotação

Em seguida, descreve-se sucintamente cada um destes esquemas de actualização dos desvios padrão.

Adaptação Isotrópica

Neste esquema, tal como na EE-(1 + 1), é considerada uma variância σ^2 comum a todas as variáveis, i.e., $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_n^2$. O desvio padrão comum σ são agora actualizados pela equação:

$$\sigma^{(k+1)} = \sigma^{(k)} e^z \text{ com } z \sim N(0, \Delta_\sigma^2), \quad (3.4)$$

onde z é determinado de acordo com uma distribuição Normal com média zero e variância Δ_σ^2 , onde Δ_σ é um parâmetro do algoritmo.

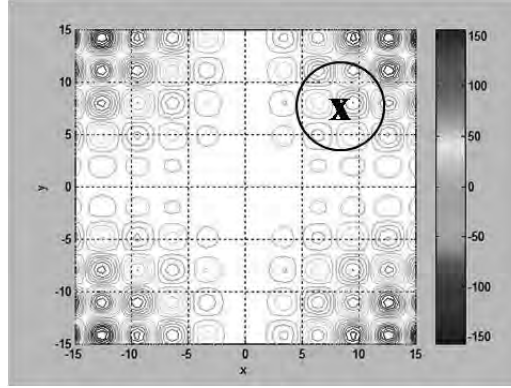


Figura 3.4: Adaptação Isotrópica

A Figura 3.4 ilustra o funcionamento deste esquema de adaptação para um problema com duas variáveis de decisão. Nesta figura, todos os pontos pertencentes à circunferência centrada no ponto x correspondem a pontos com igual probabilidade de serem gerados por mutação.

Em geral, o valor de $\Delta\sigma$ é fixado em $1/\sqrt{n}$. Uma vez que apenas existe uma única variância comum σ^2 , cada indivíduo da população possui a seguinte estrutura:

$$(\mathbf{x}; \sigma) = (x_1, x_2, \dots, x_n; \sigma).$$

Por sua vez, a mutação é dada por:

$$x'_i = x_i + z_i \text{ com } z_i \sim N(0, \sigma). \quad (3.5)$$

Algoritmo 2 Estratégia Evolutiva $(\mu/\rho, \lambda)$ com Adaptação Isotrópica

Require: $\mu, \rho, \lambda, \Delta\sigma$

- 1: $t \leftarrow 0$
 - 2: $\mathbf{P}^{(t)} \leftarrow \left\{ (\mathbf{x}_m^{(t)}; \sigma^{(t)}; f(\mathbf{x}_m^{(t)})) : m = 1, \dots, \mu \right\}$ [onde $\mathbf{x}_m^{(t)} = (x_1, x_2, \dots, x_n)^{(t)}$]
 - 3: **enquanto** CP falso **faça**
 - 4: **para** $l = 1$ to λ **faça**
 - 5: $\mathbf{g}_l \leftarrow \text{agrupar}(\mathbf{P}^{(t)}, \rho)$
 - 6: $\sigma'_l \leftarrow \text{recombinar}_s(\mathbf{g}_l)$
 - 7: $\mathbf{x}'_l \leftarrow \text{recombinar}_x(\mathbf{g}_l)$
 - 8: $\sigma''_l \leftarrow \text{mutar}_s(\sigma'_l)$
 - 9: $\mathbf{x}''_l \leftarrow \text{mutar}_x(\mathbf{x}'_l, \sigma''_l)$
 - 10: **fim para**
 - 11: $\mathbf{D}^{(t)} \leftarrow \left\{ (\mathbf{x}''_m^{(t)}; \sigma''_m^{(t)}; f(\mathbf{x}''_m^{(t)})) : m = 1, \dots, \mu \right\}$
 - 12: $\mathbf{P}^{(t+1)} \leftarrow \begin{cases} \text{selecção}(\mathbf{D}^{(t)}) & \text{se } EE - (\mu/\rho, \lambda) \\ \text{selecção}(\mathbf{D}^{(t)} \cup \mathbf{P}^{(t)}) & \text{se } EE - (\mu/\rho + \lambda) \end{cases}$
 - 13: $t \leftarrow t + 1$
 - 14: **fim enquanto**
 - 15: $\mathbf{x}_{final} \leftarrow \text{melhor}(\mathbf{P}^{(t)})$
 - 16: **retorno** \mathbf{x}_{final}
-

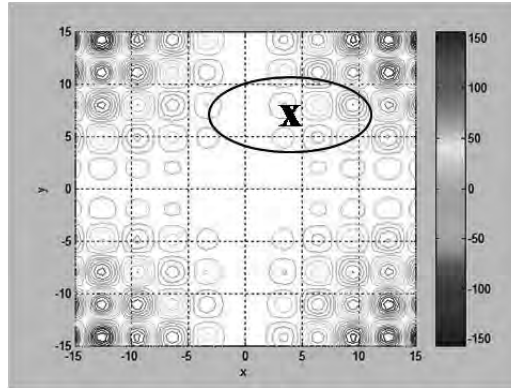


Figura 3.5: Adaptação Não Isotrópica

O Algoritmo 2 é o algoritmo da EE- $(\mu/\rho^+, \lambda)$ com adaptação isotrópica. Neste algoritmo, a função *agrupar* selecciona aleatoriamente ρ indivíduos da população de progenitores \mathbf{P} . As funções *recombinar_s* (para os desvios padrão) e *recombinar_x* (para as variáveis de decisão) recombina, respectivamente, os ρ indivíduos de acordo com a equação (3.2) ou a equação (3.3). A função *mutar_s* muda os desvios padrão aplicando a regra isotrópica expressa na equação (3.4). Finalmente, a função *mutar_x* corresponde à mutação das variáveis de decisão como é descrita na equação (3.5). O algoritmo devolve a solução correspondente ao indivíduo da população final com menor valor da função objectivo (a melhor aproximação ao óptimo).

Adaptação Não Isotrópica

Neste esquema, os desvios padrão σ_i são actualizados (um para cada variável de decisão) de acordo com a equação:

$$\sigma_i^{(k+1)} = \sigma_i^{(k)} e^{z_i} e^z, \quad (3.6)$$

onde $z_i \sim N(0, \Delta_\sigma^2)$, $\mathbf{z} \sim N(0, \Delta_{\sigma^c}^2)$ e, Δ_σ e Δ_{σ^c} são parâmetros do algoritmo. Em geral, os valores de Δ_σ e Δ_{σ^c} são fixados, respectivamente, em $1/\sqrt{2\sqrt{n}}$ e $1/\sqrt{2n}$. Este esquema não isotrópico, ao contrário do anterior, permite adaptar os desvios padrão independentemente para cada variável de decisão.

A Figura 3.5 ilustra o funcionamento deste esquema de adaptação para um problema com duas variáveis de decisão. Neste figura, os pontos pertencentes à elipse são os descendentes com igual probabilidade de serem gerados a partir de um ponto x .

Neste esquema existe agora uma variância associada a cada variável σ_i^2 , pelo que cada indivíduo da população passa a possuir a seguinte estrutura:

$$(\mathbf{x}; \sigma) = (x_1, \dots, x_n; \sigma_1, \dots, \sigma_n).$$

Para fazer a mutação, aplica-se a seguinte equação:

$$x'_i = x_i + z_i \text{ com } z_i \sim N(0, \sigma_i). \quad (3.7)$$

O Algoritmo 3 é o algoritmo da EE- $(\mu/\rho^+, \lambda)$ com adaptação não isotrópica. Tal como anteriormente, neste algoritmo, as funções *agrupar*, *recombinar_s* e *recombinar_x* permitem seleccionar aleatoriamente ρ indivíduos da população de progenitores \mathbf{P} e recombiná-los de acordo com a equação

Algoritmo 3 Estratégia Evolutiva $(\mu/\rho, \lambda)$ com Adaptação Não Isotrópica**Require:** $\mu, \rho, \lambda, \Delta_\sigma, \Delta_{\sigma'}$

```

1:  $t \leftarrow 0$ 
2:  $\mathbf{P}^{(t)} \leftarrow \left\{ (\mathbf{x}_m^{(t)}; \mathbf{s}_m^{(t)}; f(\mathbf{x}_m^{(t)})) : m = 1, \dots, \mu \right\}$  [ onde  $\mathbf{x}_m^{(t)} = (x_1, x_2, \dots, x_n)$  e  $\mathbf{s}_m^{(t)} = (\sigma_1, \sigma_2, \dots, \sigma_n)$  ]
3: enquanto CP falso faça
4:   para  $l = 1$  to  $\lambda$  faça
5:      $\mathbf{g}_l \leftarrow \text{agrupar}(\mathbf{P}^{(t)}, \rho)$ 
6:      $\mathbf{s}'_l \leftarrow \text{recombinar}_s(\mathbf{g}_l)$ 
7:      $\mathbf{x}'_l \leftarrow \text{recombinar}_x(\mathbf{g}_l)$ 
8:      $\mathbf{s}''_l \leftarrow \text{mutar}_s(\mathbf{s}'_l)$ 
9:      $\mathbf{x}''_l \leftarrow \text{mutar}_x(\mathbf{x}'_l, \sigma''_l)$ 
10:  fim para
11:   $\mathbf{D}^{(t)} \leftarrow \left\{ (\mathbf{x}''_m^{(t)}; \mathbf{s}''_m^{(t)}; f(\mathbf{x}''_m^{(t)})) : m = 1, \dots, \mu \right\}$ 
12:   $\mathbf{P}^{(t+1)} \leftarrow \begin{cases} \text{selecção}(\mathbf{D}^{(t)}) & \text{se } EE - (\mu/\rho, \lambda) \\ \text{selecção}(\mathbf{D}^{(t)} \cup \mathbf{P}^{(t)}) & \text{se } EE - (\mu/\rho + \lambda) \end{cases}$ 
13:   $t \leftarrow t + 1$ 
14: fim enquanto
15:  $\mathbf{x}_{final} \leftarrow \text{melhor}(\mathbf{P}^{(t)})$ 
16: retorno  $\mathbf{x}_{final}$ 

```

(3.2) ou a equação (3.3). Contudo, a função mutar_s muda os desvios padrão aplicando agora a regra não isotrópica expressa na equação (3.6). Finalmente, a função mutar_x corresponde à mutação das variáveis de decisão tal como é descrita na equação (3.7) de forma independente para variável de decisão.

Adaptação Não Isotrópica com Rotação

Neste esquema, para modelar a possível correlação entre as variáveis de decisão são consideradas $n(n-1)/2$ rotações. Os desvios padrão σ_i e as rotações α_j são actualizadas de acordo com a equação:

$$\begin{aligned} \sigma_i^{(k+1)} &= \sigma_i^{(k)} e^{z_i} \\ \alpha_j^{(k+1)} &= \alpha_j^{(k)} + \beta N(0, 1), \end{aligned} \quad (3.8)$$

onde $z_i \sim N(0, \Delta_\sigma^2)$, $\mathbf{z} \sim N(0, \Delta_{\sigma'}^2)$ e Δ_σ e $\Delta_{\sigma'}$, e β são parâmetros do algoritmo. Em geral, os valores de Δ_σ e $\Delta_{\sigma'}$ são fixados, respectivamente, em $1/\sqrt{2\sqrt{n}}$, $1/\sqrt{2n}$ e $\beta = 5$. Este esquema não isotrópico com rotações, ao contrário do anterior, permite adaptar os desvios padrão tendo em conta a possível existência de correlações entre as variáveis de decisão. Neste esquema existe agora uma variância associada a cada variável σ_i^2 e rotações α_j , pelo que cada indivíduo da população passa a possuir a seguinte estrutura:

$$(\mathbf{x}; \sigma; \alpha) = (x_1, \dots, x_n; \sigma_1, \dots, \sigma_n; \alpha_1, \dots, \alpha_r)$$

com $r = n(n-1)/2$.

A Figura 3.6 ilustra o funcionamento deste esquema de adaptação para um problema com duas variáveis de decisão. Neste figura, os pontos pertencentes à elipse sujeita a rotação são os descendentes com igual probabilidade de serem gerados a partir de um ponto x .

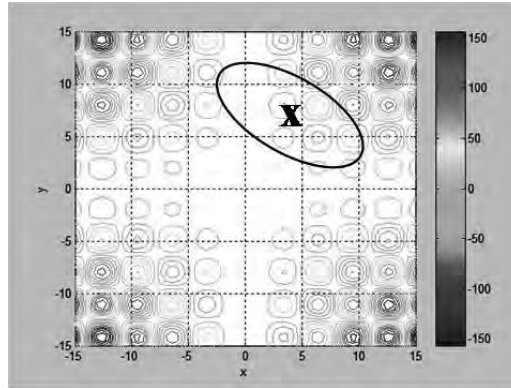


Figura 3.6: Adaptação Não Isotrópica com Rotação

A mutação é feita de acordo com a seguinte equação:

$$\begin{aligned} \mathbf{x}' &= \mathbf{x} + \mathbf{z} \text{ com } \mathbf{z} \sim N(0, \mathbf{C}') \\ \mathbf{C}' &= \begin{cases} \sigma_i^2 & \text{se } i = j \\ \frac{1}{2}(\sigma_i^2 - \sigma_j^2) \tan(2\alpha_{ij}) & \text{se } i \text{ e } j \text{ correlacionados} \\ 0 & \text{se } i \text{ e } j \text{ não correlacionados} \end{cases} \end{aligned} \quad (3.9)$$

O Algoritmo 4 é o algoritmo da EE- $(\mu/\rho, \lambda)$ com adaptação não isotrópica e com rotação. As funções *agrupar*, *recombinar_s* e *recombinar_x* são semelhantes às dos algoritmos anteriores. No entanto, a função *mutar_s* muda os desvios padrão aplicando agora a regra não isotrópica com rotações expressa na equação (3.8). Finalmente, a função *mutar_x* corresponde à mutação das variáveis de decisão tal como é descrita na equação (3.9) tendo em conta as covariâncias, i.e., as associações entre variáveis de decisão.

CrITÉrio de Paragem

O critério de convergência permite definir o término do processo de procura. Em geral, o critério de convergência adoptado para as estratégias evolutivas multimembros é terminar a procura quando as seguintes condições são verificadas:

$$\left| f_{\max}^{(k)} - f_{\min}^{(k)} \right| \leq \varepsilon_3 \text{ ou } \frac{\left| f_{\max}^{(k)} - f_{\min}^{(k)} \right|}{\left| \bar{f}^{(k)} \right|} \leq \varepsilon_4,$$

onde $f_{\max}^{(k)}$, $f_{\min}^{(k)}$ e $\bar{f}^{(k)}$ são, respectivamente, o maior e o menor valor da função objectivo e a média dos valores da função objectivo da população de μ progenitores, ε_3 e ε_4 dependem da precisão do computador utilizado, sendo $\varepsilon_3 > 0$ e $\varepsilon_4 > 0$.

6. Tratamento das Restrições

Os esquemas evolutivos, tal como foram apresentados, permitem resolver problemas de optimização sem restrições. É possível tratar problemas restrições de desigualdade do tipo $g_j(\mathbf{x}) \geq 0$ com $j = 1, \dots, m$, utilizando um mecanismo de eliminação das soluções não admissíveis, i.e., em cada geração,

Algoritmo 4 Estratégia Evolutiva $(\mu/\rho, +\lambda)$ com Adaptação Não Isotrópica com Rotação**Require:** $\mu, \rho, \lambda, \Delta_\sigma, \Delta_{\sigma'}, \beta$

```

1:  $t \leftarrow 0$ 
2:  $\mathbf{P}^{(t)} \leftarrow \left\{ (\mathbf{x}_m^{(t)}; \mathbf{s}_m^{(t)}; \mathbf{a}_m^{(t)}; f(\mathbf{x}_m^{(t)})) : m = 1, \dots, \mu \right\}$  [ onde  $\mathbf{x}_m^{(t)} = (x_1, x_2, \dots, x_n)$ ,  $\mathbf{s}_m^{(t)} = (\sigma_1, \sigma_2, \dots, \sigma_n)$  e  $\mathbf{a}_m^{(t)} = (\alpha_1, \alpha_2, \dots, \alpha_k)$  com  $k = n(n-1)/2$  ]
3: enquanto CP falso faça
4:   para  $l = 1$  to  $\lambda$  faça
5:      $\mathbf{g}_l \leftarrow \text{agrupar}(\mathbf{P}^{(t)}, \rho)$ 
6:      $(\mathbf{s}_l, \mathbf{a}_l) \leftarrow \text{recombinar}_s(\mathbf{g}_l)$ 
7:      $\mathbf{x}'_l \leftarrow \text{recombinar}_x(\mathbf{g}_l)$ 
8:      $(\mathbf{s}'_l, \mathbf{a}'_l) \leftarrow \text{mutar}_s(\mathbf{s}_l, \mathbf{a}_l)$ 
9:      $\mathbf{x}''_l \leftarrow \text{mutar}_x(\mathbf{x}'_l, \mathbf{s}'_l, \mathbf{a}'_l)$ 
10:   fim para
11:    $\mathbf{D}^{(t)} \leftarrow \left\{ (\mathbf{x}''_m{}^{(t)}; \mathbf{s}''_m{}^{(t)}; \mathbf{a}''_m{}^{(t)}; f(\mathbf{x}''_m{}^{(t)})) : m = 1, \dots, \mu \right\}$ 
12:    $\mathbf{P}^{(t+1)} \leftarrow \begin{cases} \text{selecção}(\mathbf{D}^{(t)}) & \text{se } EE - (\mu/\rho, \lambda) \\ \text{selecção}(\mathbf{D}^{(t)} \cup \mathbf{P}^{(t)}) & \text{se } EE - (\mu/\rho + \lambda) \end{cases}$ 
13:    $t \leftarrow t + 1$ 
14: fim enquanto
15:  $\mathbf{x}_{final} \leftarrow \text{melhor}(\mathbf{P}^{(t)})$ 
16: retorno  $\mathbf{x}_{final}$ 

```

se a mutação ou recombinação gerarem um ponto que não satisfaz as restrições, este não é aceite. Desta forma, a procura restringe-se ao interior da região admissível. No entanto, é muitas vezes difícil especificar um vector inicial \mathbf{x}_0 que seja uma aproximação ao óptimo e que satisfaça todas as restrições (se for utilizado um vector inicial que não satisfaça as restrições poderá demorar muito tempo até que o processo de procura determine um ponto da região admissível). Um dos processos para a determinação de um vector inicial na região admissível é o descrito por Box (1965). Neste processo, uma função objectivo auxiliar $r(\mathbf{x})$ representando a soma dos valores das funções das restrições violadas é construída:

$$r(\mathbf{x}) = \sum_{j=1}^m g_j(\mathbf{x}) \delta_j(\mathbf{x}),$$

onde

$$\delta_j(\mathbf{x}) = \begin{cases} -1 & \text{se } g_j(\mathbf{x}) < 0 \\ 0 & \text{se } g_j(\mathbf{x}) \geq 0 \end{cases}.$$

Um decréscimo no valor da função $r(\mathbf{x})$ representa uma aproximação à região admissível. Assim, quando $r(\mathbf{x}) = 0$ então \mathbf{x} satisfaz todas as restrições e pode ser utilizado como vector inicial. Enquanto não é encontrado um ponto admissível, a procura é feita com base na função $r(\mathbf{x})$ definida anteriormente.

O esquema atrás descrito permite tratar apenas restrições do tipo desigualdade. As restrições do tipo igualdade $h_i(\mathbf{x}) = 0$ podem ser reformuladas como restrições de desigualdade da seguinte forma:

$$h_i(\mathbf{x}) \geq 0 \wedge -h_i(\mathbf{x}) \geq 0.$$

Para além disso, podem ser consideradas quantidades positivas e pequenas, $\underline{\varepsilon}$ e $\bar{\varepsilon}$ de tal forma que $-\underline{\varepsilon} \leq h_i(x) \leq \bar{\varepsilon}$, i.e.:

$$h_i(\mathbf{x}) + \underline{\varepsilon} \geq 0 \wedge -h_i(\mathbf{x}) + \bar{\varepsilon} \geq 0.$$

No entanto, o tratamento de restrições do tipo igualdade pode criar dificuldades às EEs, dado que encontrar novos pontos que sejam admissíveis pode traduzir-se num comportamento oscilatório.

Outras técnicas de tratamento de restrições podem ser implementadas. Coello Coello (2002) faz uma revisão, no contexto dos algoritmos evolucionários, de diversas técnicas para o tratamento de restrições do tipo desigualdade e igualdade.

7. Estratégias Evolutivas Avançadas

Nesta secção são referidas algumas abordagens baseadas em EEs que utilizam técnicas avançadas quer para a adaptação dos parâmetros de procura quer pela utilização de informação de procura não local.

Uma dessas abordagens são as Meta-EEs propostas por Rechenberg (1994) (também, chamadas de *Nested Evolution Strategies*). As Meta-EEs utilizam informação não local na procura. Para isso, a procura é hierarquicamente organizada em diversas EEs que fazem procura local por um período de γ gerações. Há uma EE externa cujas populações são constituídas por diversas EEs internas. Assim, a notação usada para descrever uma Meta-EE passa a ser $EE-[\mu'/\rho', +\lambda'(\mu/\rho, +\lambda)^\gamma]$ onde os parêntesis rectos dizem respeito à EE externa e os curvos às EEs internas. Neste algoritmo, há μ' populações de progenitores de $EE-(\mu/\rho, +\lambda)$ que geram λ' $EE-(\mu/\rho, +\lambda)$ descendentes. Após γ gerações, a selecção correspondente à EE externa é feita para determinar as melhores μ' das $EE-(\mu/\rho, +\lambda)$ que servirão de base para a próxima população de λ' $EE-(\mu/\rho, +\lambda)$ descendentes. As Meta-EEs podem ser utilizadas em optimização global, multi-local, problemas inteiros mistos e, também, na optimização dos próprios parâmetros de procura das EEs internas.

Outras abordagens, são as propostas por Ostermeier et al. (1994) and Hansen e Ostermeier (2001), que utilizam informação do progresso ao longo das gerações para fazer a auto-adaptação dos parâmetros de procura. No algoritmo proposto por Ostermeier et al. (1994), designado por CSA-ES (*Cumulative Step-size Adaptation Evolution Strategy*), informação relativa ao percurso evolutivo percorrido ao longo das gerações é utilizada para adaptar os tamanhos do passo. No algoritmo CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*) desenvolvido por Hansen e Ostermeier (2001), o percurso evolutivo cumulativo é também utilizado para adaptar a matriz de covariâncias que permite a aplicação de mutação correlacionada. O estudo e a descrição aprofundada destas estratégias evolutivas avançadas sai fora do âmbito deste capítulo, podendo o leitor encontrar os detalhes destas propostas nas referências bibliográficas atrás indicadas.