**3.4 Estratégias de Evolução**  
Estratégias de Evolução, Estratégia Evolutiva, Estratégias Evolutivas, ES.

### **3.4.1 Taxonomia**

Estratégias de Evolução é um algoritmo de otimização global e é uma instância de um Algoritmo Evolutivo do campo da Computação Evolutiva. Estratégias de Evolução é uma técnica irmã de outros Algoritmos Evolutivos, como Algoritmos Genéticos (Seção 3.2), Programação Genética (Seção 3.3), Sistemas de Classificação de Aprendizagem (Seção 3.9) e Programação Evolutiva (Seção 3.6). Um descendente popular do algoritmo de Estratégias de Evolução é o Covariance Matrix Adaptation Evolution Strategies (CMA-ES).

### **3.4.2 Inspiração**

Estratégias de Evolução é inspirada na teoria da evolução por meio da seleção natural. Especificamente, a técnica é inspirada no processo macro ou no nível de espécie da evolução (fenótipo, hereditariedade, variação) e não se preocupa com os mecanismos genéticos da evolução (genoma, cromossomos, genes, alelos).

### **3.4.3 Estratégia**

O objetivo do algoritmo de Estratégias de Evolução é maximizar a adequação da coleção de soluções candidatas no contexto de uma função objetivo de um domínio. O objetivo era, classicamente, alcançado por meio da adoção de variação dinâmica, um substituto para a descendência com modificação, onde a quantidade de variação era adaptada dinamicamente com heurísticas baseadas em desempenho. Abordagens contemporâneas adaptam parâmetros que controlam a quantidade e o viés da variação com as soluções candidatas.

### **3.4.4 Procedimento**

Instâncias dos algoritmos de Estratégias de Evolução podem ser descritas de forma concisa com uma terminologia personalizada na forma (μ, λ) − ES, onde μ é o número de soluções candidatas na geração parental, e λ é o número de soluções candidatas geradas a partir da geração parental. Nesta configuração, os melhores μ são mantidos se λ > μ, onde λ deve ser maior ou igual a μ. Além do chamado algoritmo de seleção por vírgula das Estratégias de Evolução, pode-se definir uma variação de seleção por adição (μ + λ) − ES, onde os melhores membros da união das gerações μ e λ competem com base na aptidão objetiva para uma posição na próxima geração. A configuração mais simples é o (1 + 1) − ES, que é um tipo de algoritmo de escalada de colina ganancioso. O Algoritmo 3.4.1 fornece uma listagem de pseudocódigo do algoritmo (μ, λ) − ES para minimizar uma função de custo. O algoritmo mostra a adaptação de soluções candidatas que coadaptam seus próprios parâmetros de estratégia que influenciam a quantidade de mutação aplicada aos descendentes das soluções candidatas.

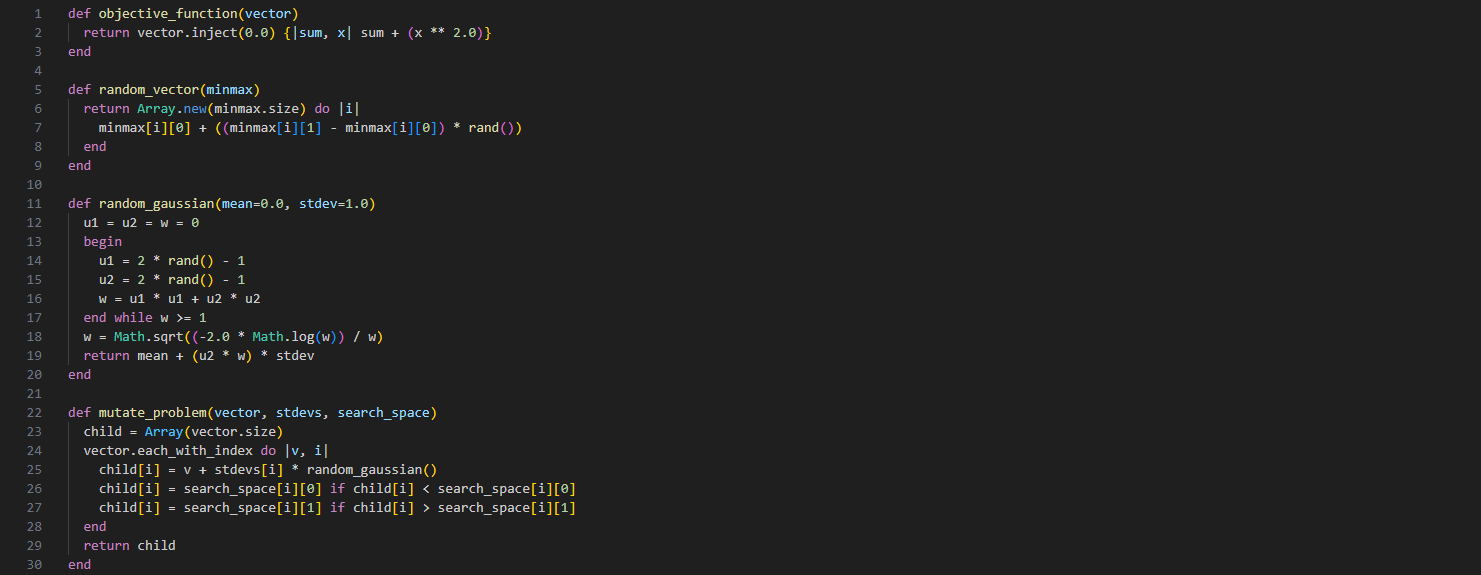
**Algoritmo 3.4.1**: Pseudocódigo para Estratégias de Evolução (μ, λ).  
Entrada: μ, λ, TamanhoDoProblema  
Saída: Sbest  
População ← InicializarPopulação(μ, TamanhoDoProblema);  
AvaliarPopulação(População);  
Sbest ← ObterMelhor(População, 1);  
enquanto ¬CondiçãoDeParada() faça  
 Filhos ← ∅;  
 para i = 0 até λ faça  
  Parenti ← ObterPai(População, i);  
  Si ← ∅;  
  Siproblema ← Mutar(P iproblema, P iestratégia);  
  Siestratégia ← Mutar(P iestratégia);  
  Filhos ← Si;  
 fim  
AvaliarPopulação(Filhos);  
Sbest ← ObterMelhor(Filhos + Sbest, 1);  
População ← SelecionarMelhores(População, Filhos, μ);  
fim  
retornar Sbest;

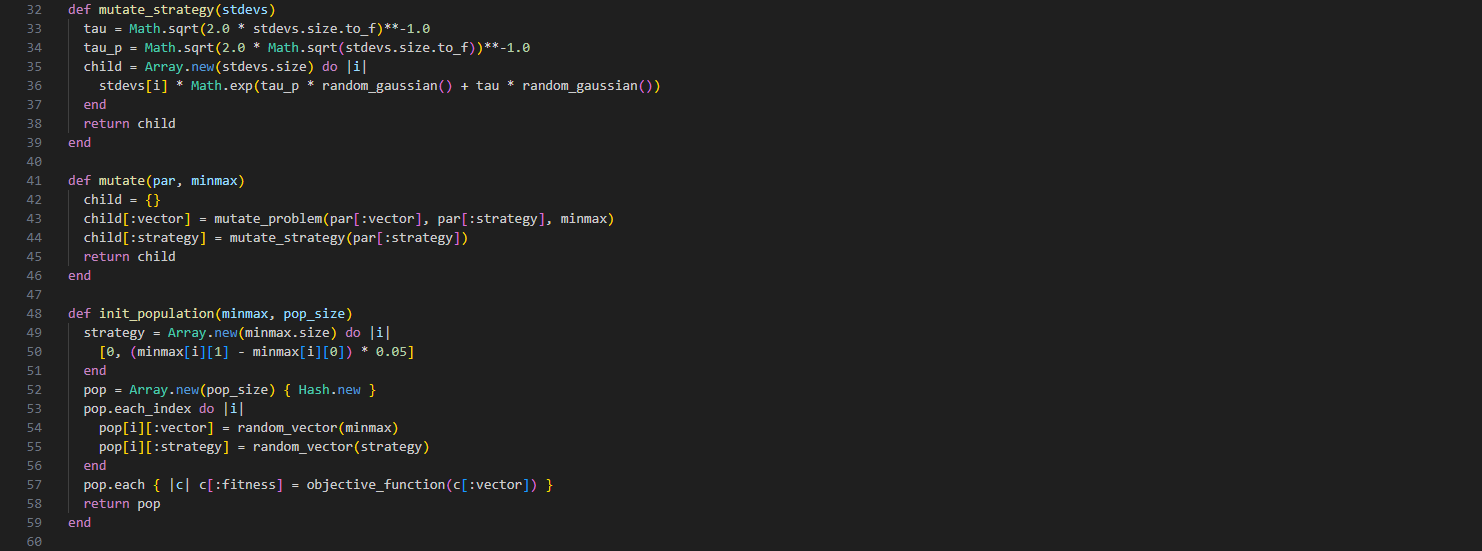
### **3.4.5 Heurísticas**

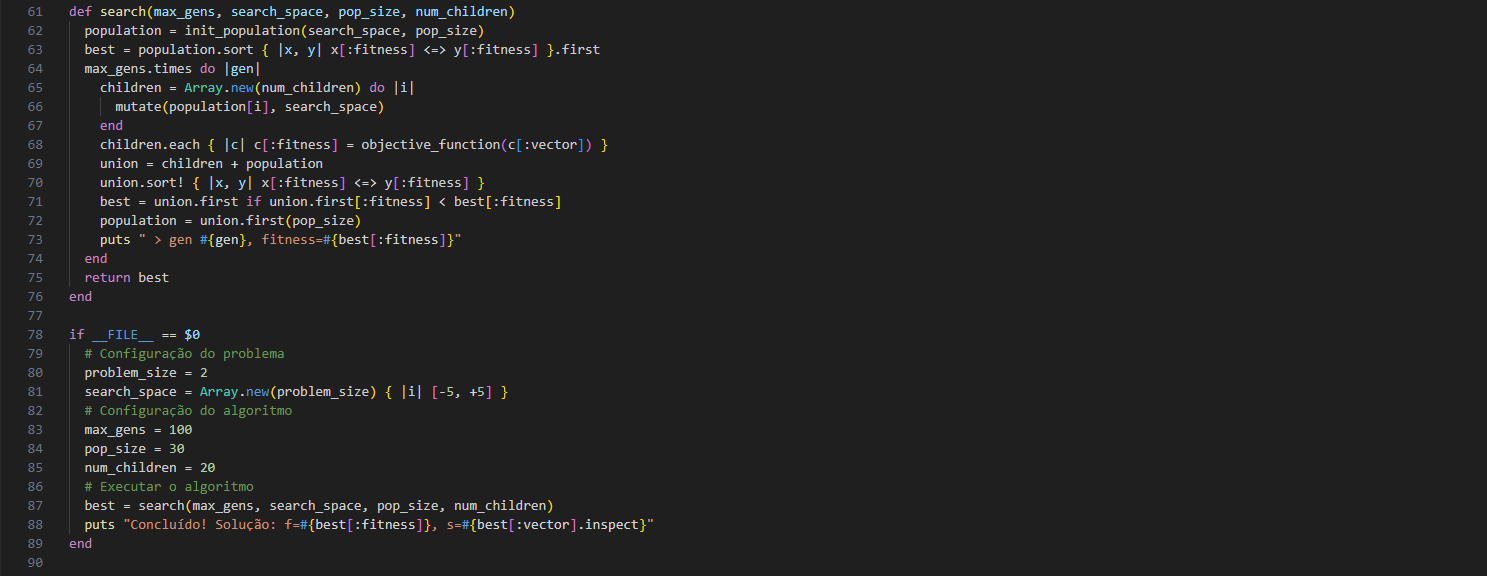
* Estratégias de Evolução utilizam representações específicas do problema, como valores reais para otimização de funções contínuas.
* O algoritmo é comumente configurado de forma que 1 ≤ μ ≤ λ.
* A razão de μ para λ influencia a quantidade de pressão de seleção (ganância) exercida pelo algoritmo.
* Uma atualização contemporânea na notação dos algoritmos inclui um ρ como (μ/ρ, λ) − ES, que especifica o número de pais que contribuirão para cada nova solução candidata usando um operador de recombinação.
* Uma regra clássica usada para governar a quantidade de mutação (desvio padrão utilizado na mutação para otimização de função contínua) era a regra de 1/5, onde a razão de mutações bem-sucedidas deveria ser 1/5 de todas as mutações. Se for maior, a variância é aumentada; caso contrário, se a razão for menor, a variância é diminuída.
* A variação de seleção por vírgula do algoritmo pode ser boa para instâncias de problemas dinâmicos, dada a sua capacidade de exploração contínua do espaço de busca, enquanto a variação de seleção por adição pode ser boa para refinamento e convergência.

### **3.4.6 Listagem de Código**

A Listagem 3.3 fornece um exemplo do algoritmo de Estratégias de Evolução implementado na linguagem de programação Ruby. O problema demonstrado é uma instância de otimização de função contínua que busca minimizar f(x), onde f = ∑n i=1 x² i, −5,0 ≤ xi ≤ 5,0 e n = 2. A solução ótima para essa função é (v0, . . . , vn−1) = 0,0. O algoritmo é uma implementação de Estratégias de Evolução baseada na versão simples descrita por Bäck e Schwefel, que também foi usada como base para um estudo empírico detalhado. O algoritmo é um (30 + 20) − ES que adapta tanto as variáveis do problema quanto as da estratégia (desvios padrão).







### **3.4.7 Referências**

#### **Fontes Primárias**

As Estratégias Evolutivas foram desenvolvidas por três estudantes (Bienert, Rechenberg, Schwefel) na Universidade Técnica de Berlim em 1964, como um esforço para otimizar automaticamente um problema de design aerodinâmico. O trabalho seminal nas Estratégias Evolutivas foi a tese de doutorado de Rechenberg [5], que mais tarde foi publicada como um livro, ambos em alemão. Muitos relatórios técnicos e artigos foram publicados por Schwefel e Rechenberg, embora o artigo seminal publicado em inglês tenha sido escrito por Klockgether e Schwefel sobre o problema de design do bocal em duas fases.

#### **Saiba Mais**

Schwefel publicou sua dissertação de doutorado logo após Rechenberg, que também foi lançada como um livro, ambos em alemão. O livro de Schwefel foi posteriormente traduzido para o inglês e é uma referência clássica para a técnica. Bäck et al. oferecem uma introdução clássica à técnica, abordando a história, o desenvolvimento do algoritmo e os passos que o levaram ao ponto em que estava em 1991. Beyer e Schwefel fornecem uma introdução contemporânea ao campo, que inclui uma história detalhada da abordagem, os desenvolvimentos e melhorias desde sua criação, além de uma visão geral dos achados teóricos realizados.