CSE/ECE 848 Introduction to Evolutionary Computation

Module 2, Lecture 9, Part 3

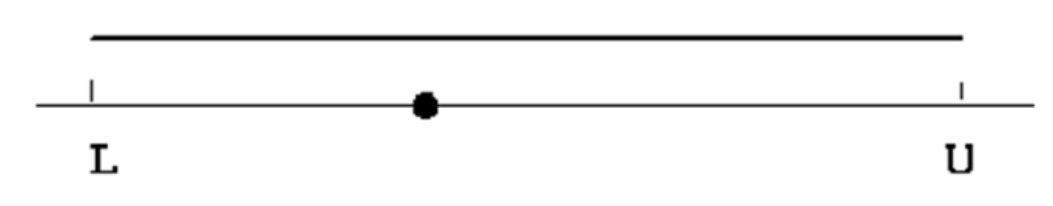
Variants of BGAs and RGAs

Kalyanmoy Deb, ECE

Koenig Endowed Chair Professor

Real-parameter Mutation Operators

- Variable-wise mutation operators
- Every variable is mutated with prob. p_m $(\approx 1/n)$
- x_i is mutated to x'_i (perturbed in neighborhood)
- Uniform Mutation: (Not recommended)



- Gaussian Mutation
- Polynomial Mutation
- Mutation clock (effective for bit-wise mutation)
  - Estimate the next bit (variable) to be mutated

Reference: Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. Chichester, London: Wiley

Gaussian Mutation Operator

- Use a truncated Gaussian distribution with σ:
- Create a random number $u_i \in [0,1]$ for every var.

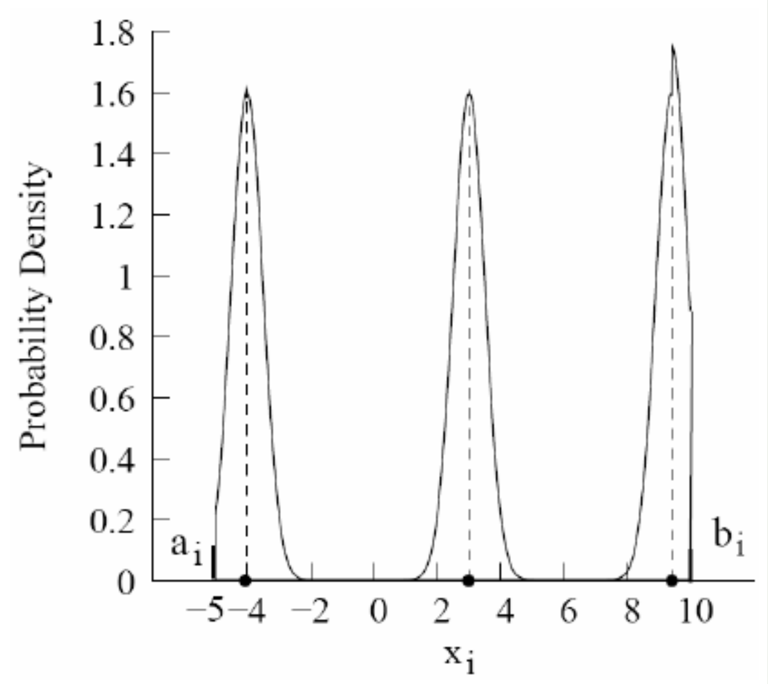
$$x_i' = x_i + \sqrt{2}\sigma(b_i - a_i)\operatorname{erf}^{-1}(u_i'),$$

erf() is the Error function

$$u_i' = \begin{cases} 2u_L(1-2u_i), & \text{if } u_i \le 0.5, \\ 2u_R(2u_i - 1), & \text{otherwise,} \end{cases}$$

$$u_L = 0.5\left(\operatorname{erf}\left(\frac{a_i - x_i}{\sqrt{2}(b_i - a_i)\sigma}\right) + 1\right)$$

$$u_R = 0.5\left(\operatorname{erf}\left(\frac{b_i - x_i}{\sqrt{2}(b_i - a_i)\sigma}\right) + 1\right)$$

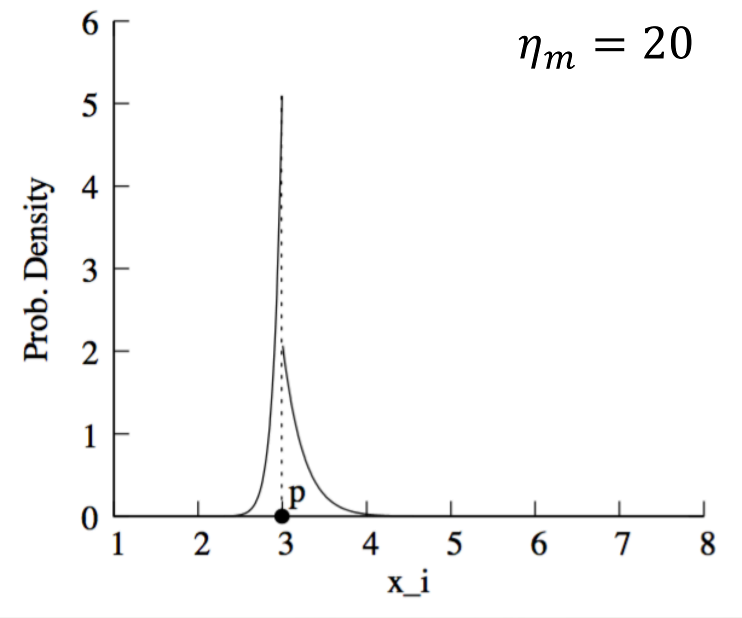


Boundary handling included

- Vector-wise Gaussian mutation possible

Polynomial Mutation Operator

- For the chosen variable, parent is $p \in [a, b]$
- Choose a random number $u \in [0,1]$
- If $u \le ½$, calculate $\delta_L = (2u)^{\frac{1}{1+\eta_m}} - 1$; $\eta_m \in [20,100]$
  else calculate $\delta_R = 1 - (2(1-u))^{\frac{1}{1+\eta_m}}$
- Calculate mutated child:
  - $p' = p + \delta_L(p-a)$, if $u \le ½$;
    $p' = p + \delta_R(b-p)$, otherwise
- Illustration: $p = 3.0 \in [1,8]$



Reference: Deb, K. and Deb, D. (2014). Analysing mutation schemes for real-parameter genetic algorithms. Int. J. Artificial Intelligence and Soft Computing, 4(1), Inderscience Enterprises Ltd., 1–28.

Boundary Handling in Real-parameter EAs

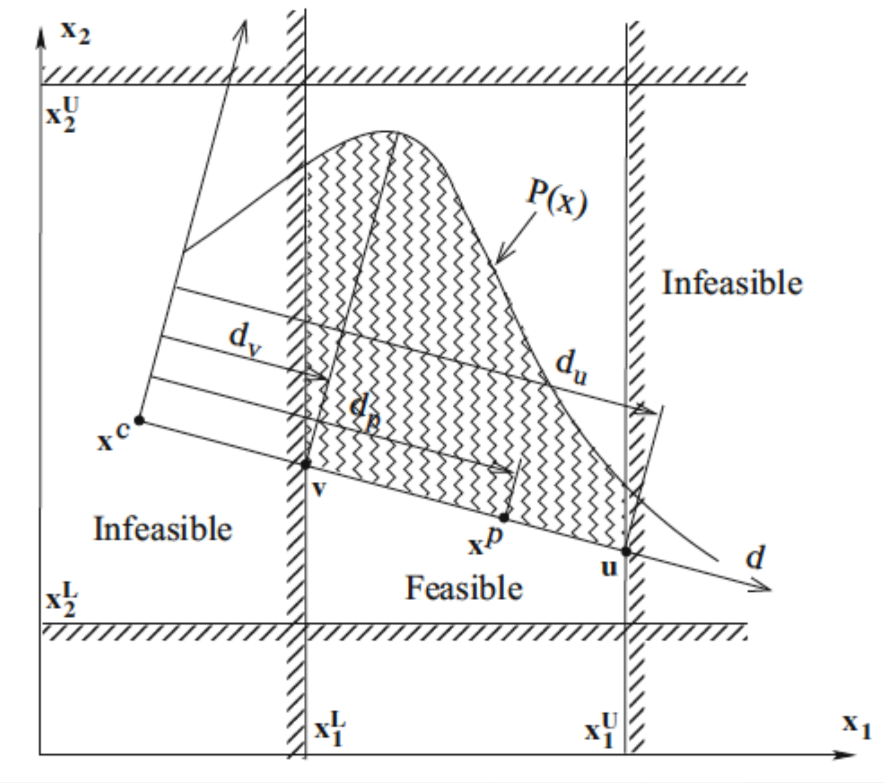
- When a boundary-less search creates a child outside variable bounds, bring it back, but how?
- Inverse Parabolic penalty:

Repair child x^c to create $\vec{y}$:

$$d' = d_v + \alpha d_v \tan\left(r \tan^{-1} \frac{a - d_v}{\alpha d_v}\right)$$

$$\vec{y} = \vec{x}^c + d'(\vec{x}^p - \vec{x}^c)$$

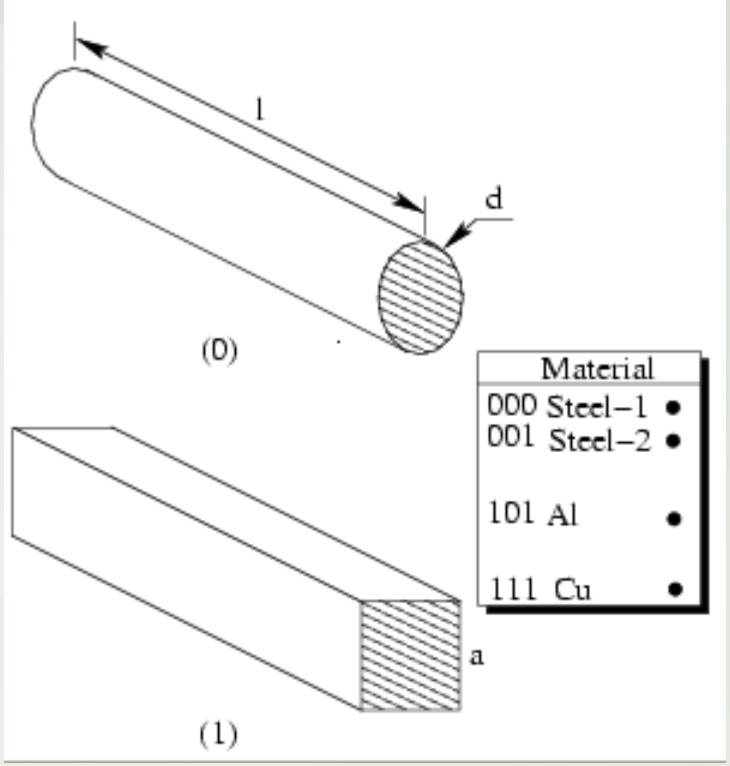
$r \in [0,1]$
$\alpha = 1.2$
$a = d_u$



Reference: Padhye, N., Mittal, P., and Deb, K. (2015). Feasibility preserving constraint handling strategies for real-parameter evolutionary optimization. Computational Optimization and Applications, 62(3), 851–890.

Mixed Variable Handling

- EAs are excellent for handling mixed variables
- A mixed representation: [(1) 14 23.457 (101)]
  - (1): circular or square cross-section
  - 14 mm: diameter/side
  - 23.457 mm: length
  - (101): material
- Otherwise, many opt. runs
- Recomb & mut. on similar var.
- Permutation + real + discrete



Reference: Deb, K., and Goyal. M. (1996). A combined genetic adaptive search (GeneAS) for engi-neering design. Computer Science and Informatics, 26(4), 30–45.

End of Module 2, Lecture 9, Part 3

- Real-parameter mutation creates a perturbed point
  - Gaussian mutation
  - Polynomial mutation
- Boundary handling is required (with RGAs, DE, PSO, etc.)
- Handling mixed variables relatively easy with EAs