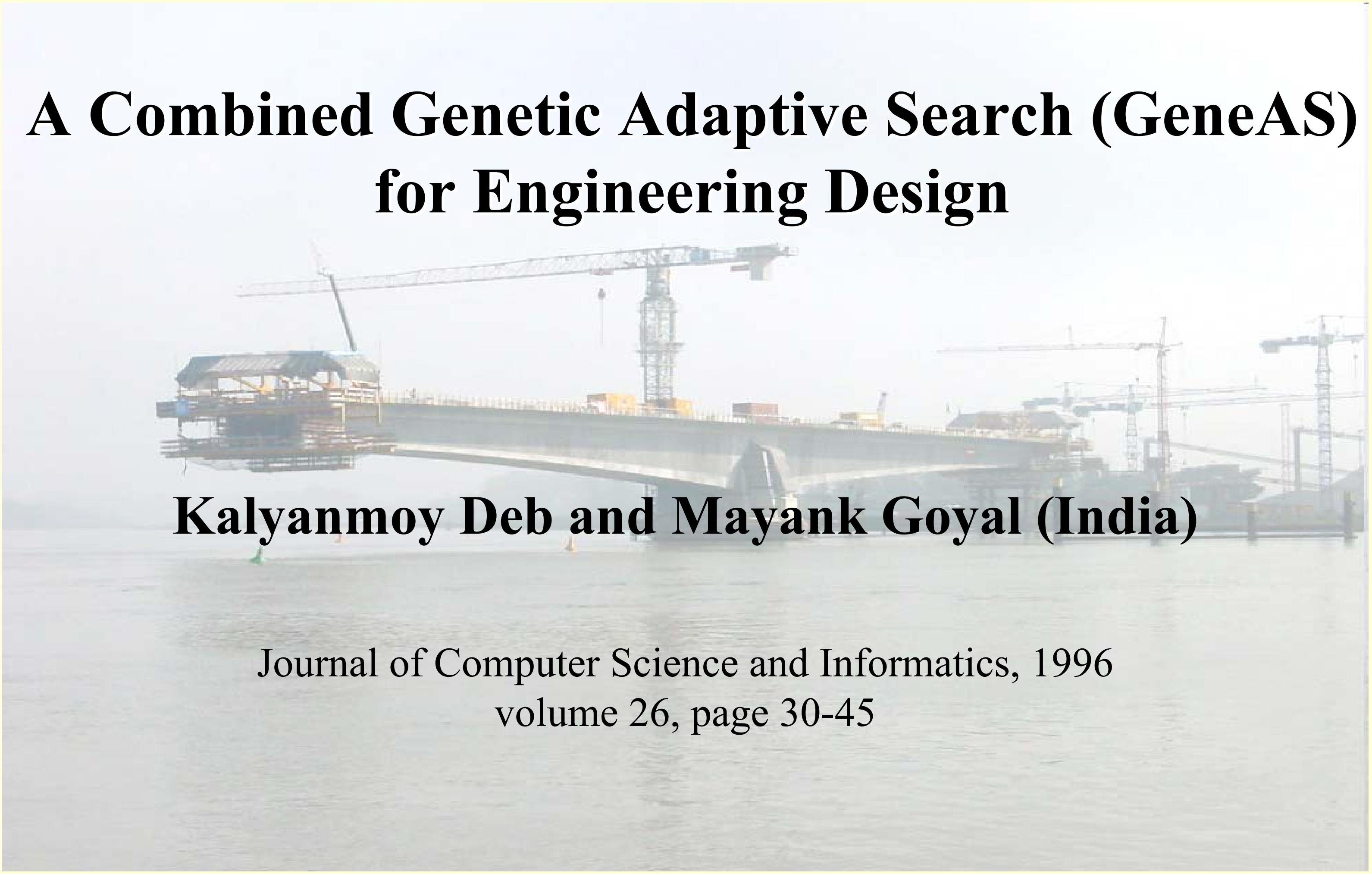


A Combined Genetic Adaptive Search (GeneAS) for Engineering Design

Kalyanmoy Deb and Mayank Goyal (India)

Journal of Computer Science and Informatics, 1996
volume 26, page 30-45

Reviewed by

Paskorn Champrasert
paskorn@cs.umb.edu
http://dssg.cs.umb.edu

Outline

- Introduction
- Genetic Operations
  - Crossover (Binary Crossover, SBX Crossover)
  - Mutation
- GeneAS Algorithms
- Experiment Results
- Conclusion

Introduction

- The aim in an engineering design problem is to minimize or maximize a design objective and satisfy constraints.
- The optimization problem can be expressed in the following form:

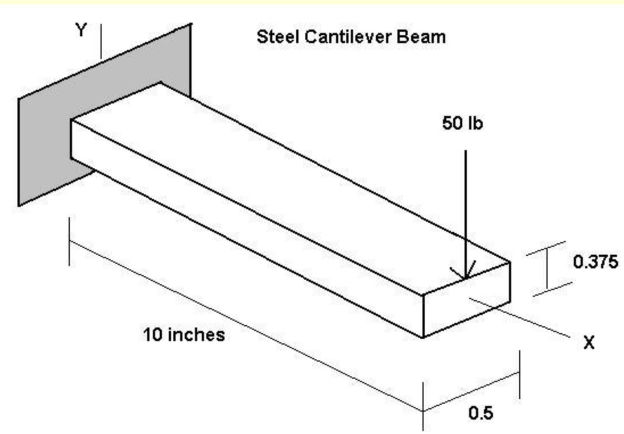
$$
\begin{aligned}
&\text{Minimize} \quad f(\boldsymbol{x}) \\
&\text{subject to} \\
&g_j(\boldsymbol{x}) \geq 0, \qquad j = 1, 2, \ldots, J; \\
&h_k(\boldsymbol{x}) = 0, \qquad k = 1, 2, \ldots, K; \\
&x_i^{(L)} \leq x_i \leq x_i^{(U)}, \qquad i = 1, 2, \ldots, N.
\end{aligned}
$$

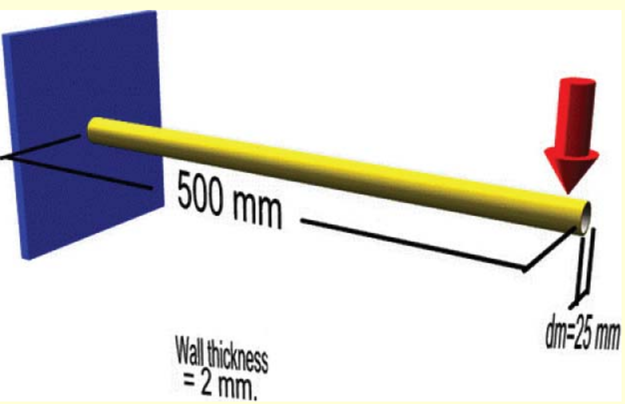
- The variable vector ***X*** represents a set of variables.

$$x_i, i = 1, 2, , N$$

A Simple Cantilever beam design problem:

- A cantilever beam is a beam that is rigidly supported at one end and carries a certain load on the other end.





Design possibilities:

- **Shape (circular or square)**
  → <u>binary design variable</u> (0/1)
- **Size**
  Diameter of circular shape
  Side of square shape
  → <u>discrete variable</u>
  because size are not available in any sizes
- **Length**
  → <u>real variable</u>
  because we can cut beams at any length
- **Material**
  → **discrete variable**
  (cast iron, aluminum, steel, brass)

Research Issues

- The optimization problem is involving with different types of variables.
  E.g.,
  - For the material variable
    if the steel is represented as 1 and cast iron is represented as 2,
    no intermediate value (e.g., 1.4) is allowed.
    → binary-coded variable should be used.
  - For the beam length variable (0-100)
    if the length is represented by using 4 bit binary-coded variable (0000-1111). The precision of the variable is only 100/16 = 6.25
    no more precise value (e.g., 2.5) is allowed
    → real-coded variable should be used.
- Designers must rely on an optimization algorithms which is flexible enough to handle a mixed type of variables

GeneAS

- The algorithm restricts its search only to the permissible values of the design variables.

- GeneAS uses a combination of binary-coded and real-coded GA, depending on the nature of the design variables.
  - Variables are coded differently
  - Genetic Operations (crossover and mutation) are performed differently for each variable.

    E.g.,
    - Beam length is real-coded variable
      real value crossover and real value mutation are used.
    - Beam diameter is binary-coded variable
      binary value crossover and real value mutation are used.

Binary-Coded GA

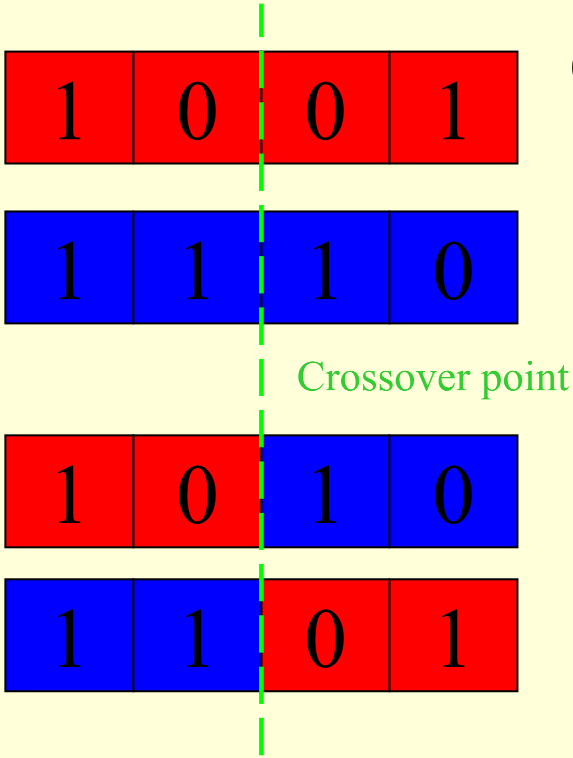
- In a binary-coded GA, the variables are represented in binary strings formed with 1 and 0.
- Variables can be expressed as :

$$(b_1b_2b_3\ldots b_n) ,\ \ b_i \text{ is ether 1 or 0}$$

- E.g., 7-bit string representing the diameter in the range of (3,130) mm, the precision in the solution is 1 mm

$$(130\text{-}3)/(1111111_B - 0000000_B)$$
$$= 127/127 = 1\text{mm}$$

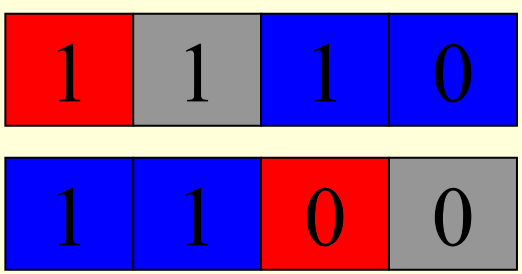
Binary Genetic Operations



One-point Crossover

1) A single crossover point on both parents string is selected.
2) All data beyond that crossover point in either parent is swapped between the two parent organisms.

The resulting organisms are the offspring:

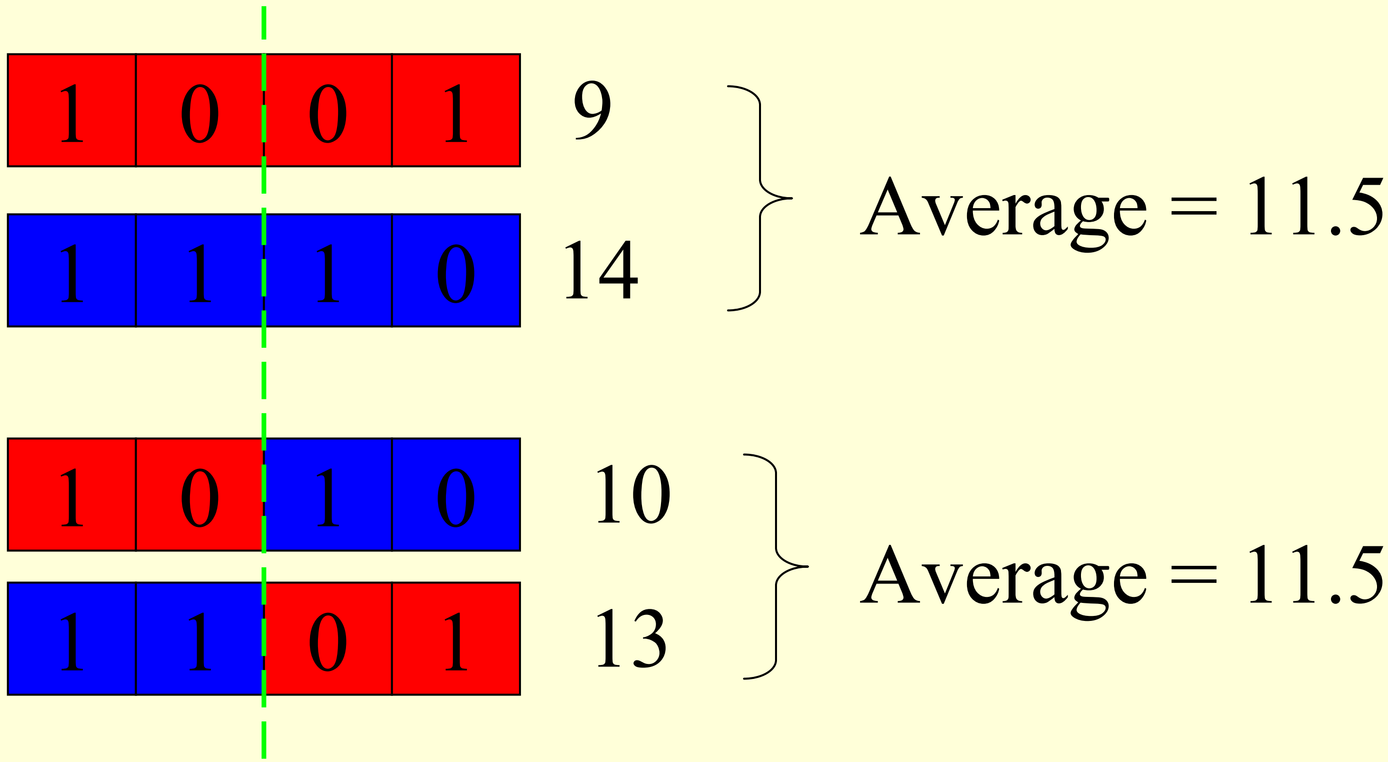


Binary Mutation

Each bit of offspring has a probability (mutation rate) to flip (0 →1 or 1→0)

Important Property of One-Point Crossover

1) The average of the decoded parameter values is the same before and after the crossover operation.



$$
\begin{array}{cccc|ccccc}
 & B_1 & & & & A_1 & & & \mathrm{DV} \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & & 85 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & & 51 \\
\cline{9-9}
 & B_2 & & & & A_2 & & \text{Avg.} & 68
\end{array}
\quad \Longrightarrow \quad
\begin{array}{cccc|ccccc}
 & B_1 & & & & A_2 & & & \mathrm{DV} \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & & 83 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & & 53 \\
\cline{9-9}
 & B_2 & & & & A_1 & & \text{Avg.} & 68
\end{array}
$$

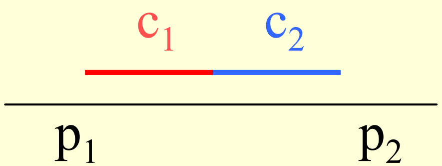
Figure 1: The action of single-point crossover on two random strings are shown. DV stands for the decoded parameter value. Notice that the average of the decoded parameter values is the same before and after the crossover operation.

Important Property of One-Point Crossover

2) Spread factor:

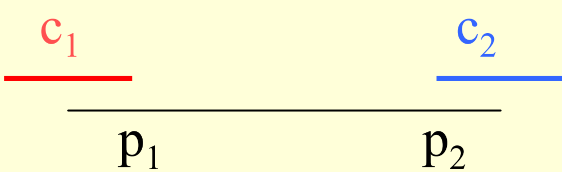
Spread factor β is defined as the ratio of the spread of offspring points to that of the parent points:

$$\beta = \left| \frac{c_1 - c_2}{p_1 - p_2} \right|$$



- **Contracting Crossover** $\beta < 1$

  The offspring points are enclosed by the parent points.

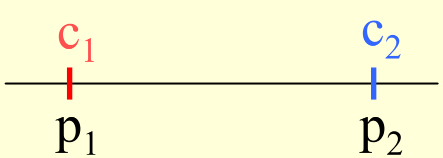


- **Expanding Crossover** $\beta > 1$

  The offspring points enclose the parent points.

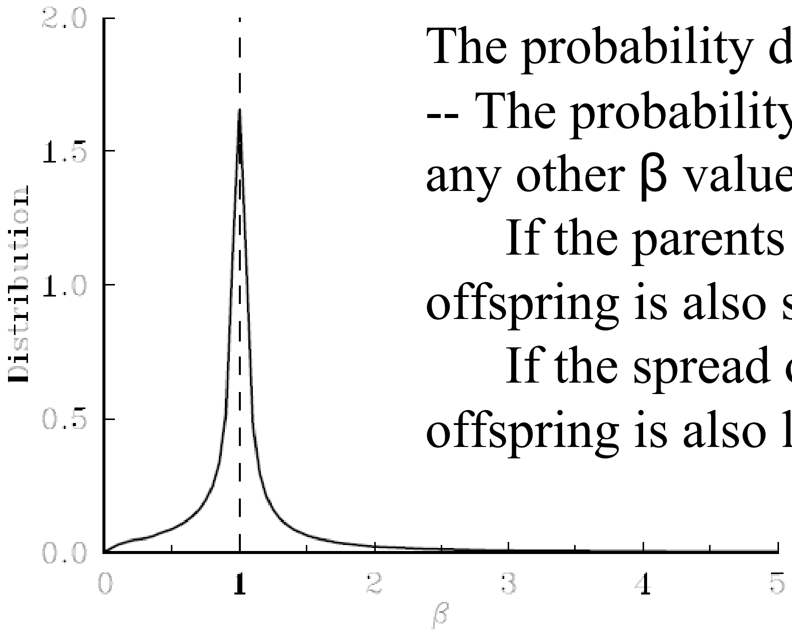
- **Stationary Crossover** $\beta = 1$

  The offspring points are the same as parent points



Spread Factor Property

- Assume parents strings are two strings:
  $p1 = (a_1 a_2 a_3 \ldots a_{15})$
  $p2 = (b_1 b_2 b_3 \ldots b_{15})$.
- The crossover point is varied from (0,15)



The probability distribution:
-- The probability of occurrence of $\beta \approx 1$ is more likely than any other β value.

If the parents are closer, the spread of the two likely offspring is also smaller.

If the spread of parents is more, the spread of likely offspring is also large.

Figure 4: Probability distributions of contracting and expanding crossovers on all pairs of random binary strings of length 15 are shown.

Real-Coded GA

- In a real-coded GA, the variables are represented in continuous variables.
- Variables can be expressed as :

  X , X is any number in $[X^L, X^U]$

  X^L is lower bound , X^U is upper bound

- The binary-coded genetic operations cannot be used for real-coded variables.

- The genetic operations have to be modified in order to handle real-coded variables
  However, the proposed real-coded genetic operations should reserve the binary-coded genetic operation properties.

SBX
Simulated Binary Crossover

- Simulated Binary Crossover (SBX) was proposed in 1995 by Deb and Agrawal.

- SBX was designed with respect to the one-point crossover properties in binary-coded GA.
  - **Average Property:**
    The average of the decoded parameter values is the same before and after the crossover operation.
  - **Spread Factor Property:**
    The probability of occurrence of spread factor $\beta \approx 1$ is more likely than any other β value.

SBX – Offspring Value

To reserve the average property, the offspring values are calculated as:

Offspring:

$$c_1 = \bar{x} - \frac{1}{2}\beta(p_2 - p_1)$$

$$c_2 = \bar{x} + \frac{1}{2}\beta(p_2 - p_1)$$

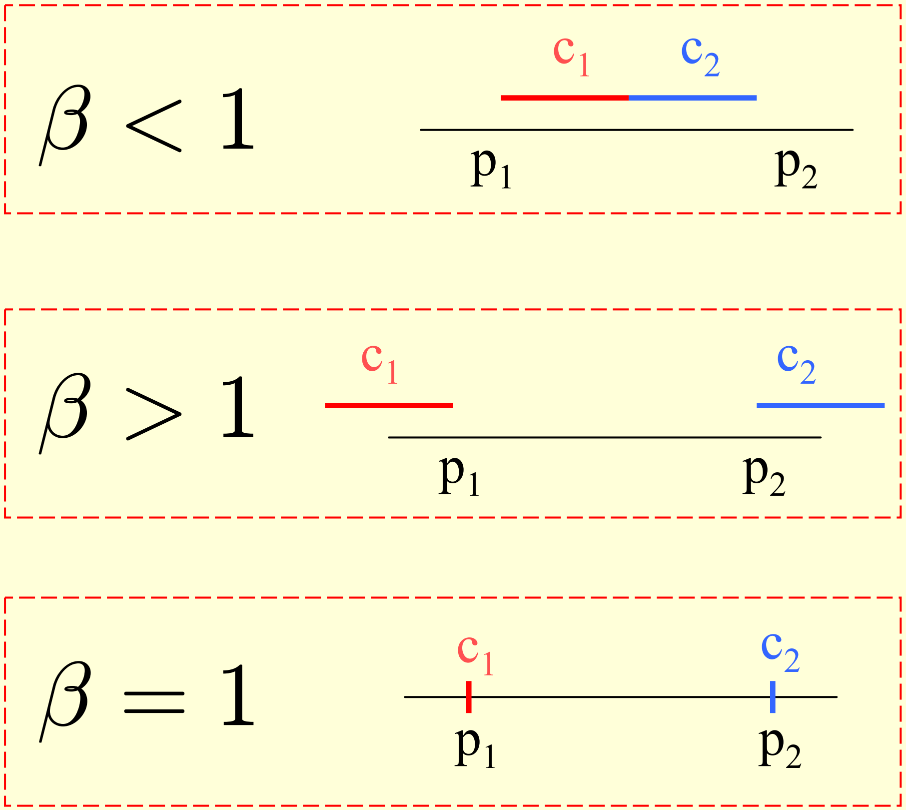
By,

$$\bar{x} = \frac{1}{2}(p_1 + p_2)$$

$$p_2 > p_1$$

So,

$$\bar{c} = \bar{x}$$

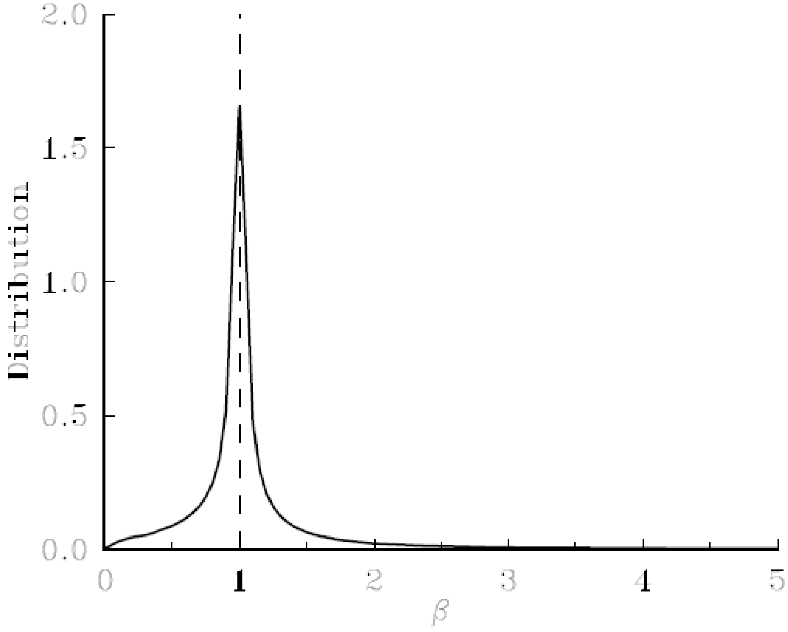


Spread Factor (β) in SBX

- The probability distribution of β in SBX should be similar (e.g. same shape) to the probability distribution of β in Binary-coded crossover.

- This paper proposed probability distribution function as:

$$\begin{cases} c(\beta) = 0.5(n+1)\beta^n, \beta \leq 1 \\ c(\beta) = 0.5(n+1)\frac{1}{\beta^{n+2}}, \beta > 1 \end{cases}$$



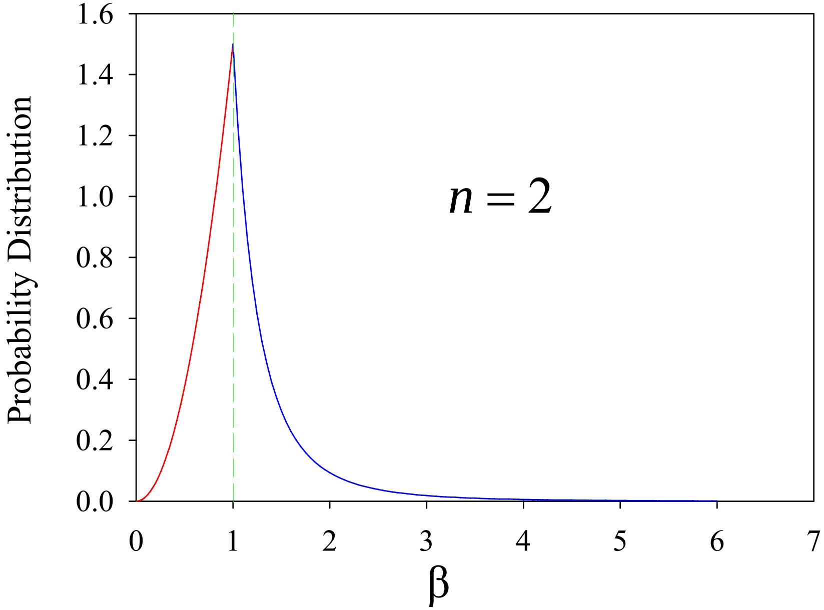


Figure 4: Probability distributions of contracting and expanding crossovers on all pairs of random binary strings of length 15 are shown.

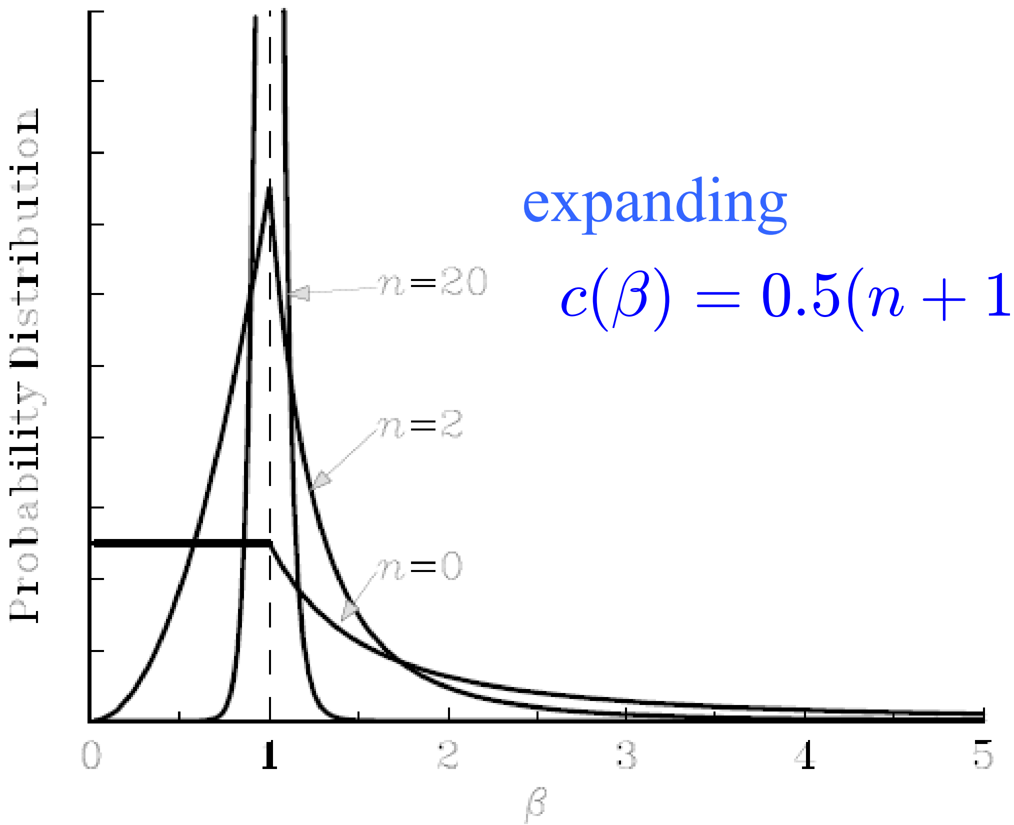
A large value of n gives a higher probability for creating 'near-parent' solutions.

contracting

$$c(\beta) = 0.5(n+1)\beta^n, \beta \leq 1$$

expanding

$$c(\beta) = 0.5(n+1)\frac{1}{\beta^{n+2}}, \beta > 1$$

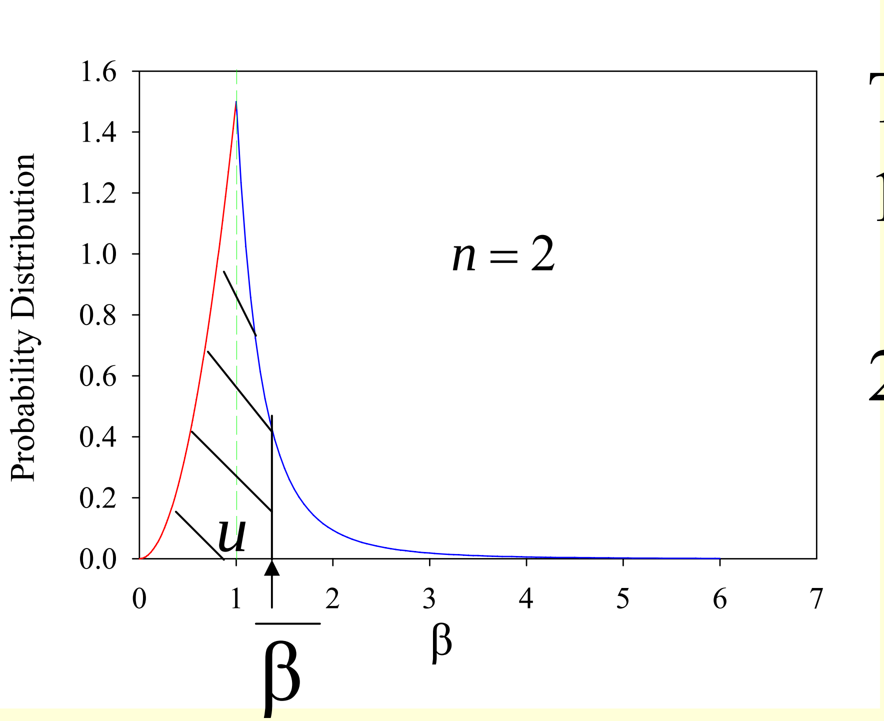


Mostly, n=2 to 5

Figure 5: Probability distributions of contracting and expanding crossovers for the proposed polynomial model are shown.

Spread Factor (β) as a random number

- $\overline{\beta}$ is a random number that follows the proposed probability distribution function:



To get a $\overline{\beta}$ value,

1) A unified random number u is generated [0,1]
2) Get β value that makes the area under the curve = u

Offspring:

$$c_1 = \bar{x} - \frac{1}{2}\overline{\beta}(p_2 - p_1)$$

$$c_2 = \bar{x} + \frac{1}{2}\overline{\beta}(p_2 - p_1)$$

Real Mutation

- Mutation probability (p_m) in (0,1) is applied to each variable.
- For each variable
  If a unified random number u_m in (0,1) < p_m
  mutation operator is performed for the variable.

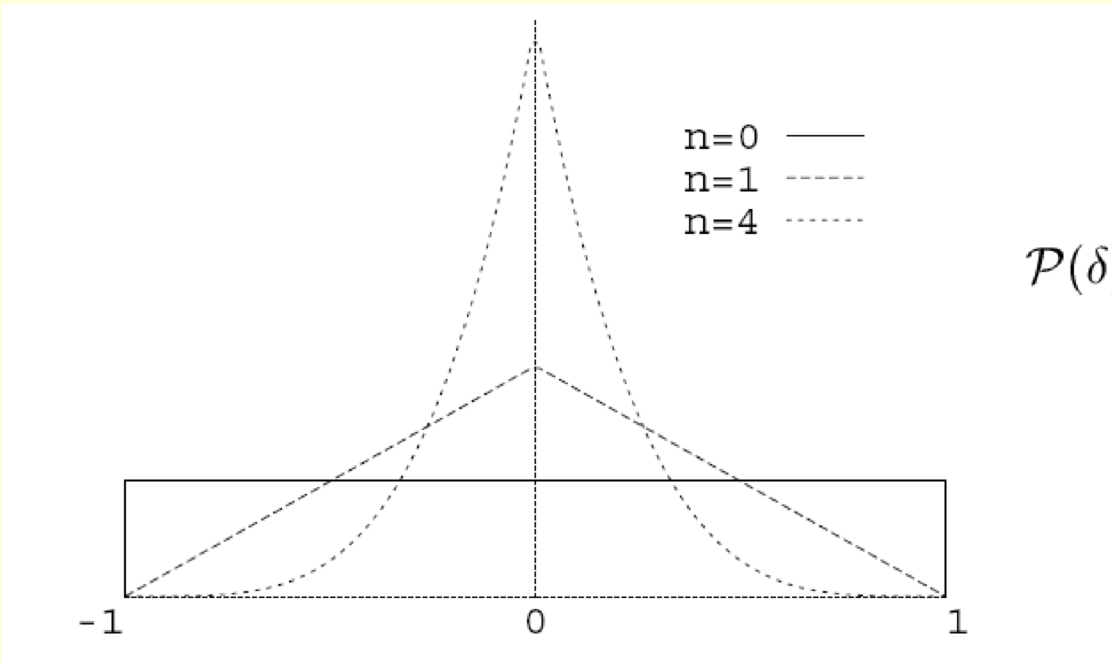
$$c = p + \bar{\delta}\Delta_{\max}.$$

p: parent
c: offspring

$\Delta_{\max}$ is a fixed quantity, It represent the maximum permissible change in the parent

$\bar{\delta}$ is a random number. Its probability distribution function is:

$$\mathcal{P}(\delta) = 0.5(n+1)(1-|\delta|)^n.$$ $\delta \in (-1,1).$



$$\mathcal{P}(\delta) = 0.5(n+1)(1-|\delta|)^n.$$

$$\delta \in (-1, 1).$$

$\bar{\pm}$ is generated the same process as the $\overline{\beta}$

Combined GA (GeneAS)

- GeneAS
  - Each variable is coded depending on its nature.
  - The crossover operator and mutation operator have to be applied variable by variable.
    - If it is a binary-coded variable, the binary-genetic operations are performed
    - If it is a real-coded variable, the real-genetic operations are performed.
  - The fitness function and constraint function are calculated as normal.

```
/* Choose a coding for handling mixed variables */
for i = 1 to population_size
        old_solution[i] = random_solution;
        old_solution[i].fitness = fitness(old_solution[i]);
generation = 0;
repeat
        generation = generation + 1;
        for i = 1 to population_size step 2
                parent1 = reproduction(old_solution);
                parent2 = reproduction(old_solution);

                (child1, child2) = crossover(parent1, parent2);

                new_solution[i] = mutation(child1);
                new_solution[i+1] = mutation(child2);

                new_solution[i].fitness = fitness(new_solution[i]);
                new_solution[i+1].fitness = fitness(new_solution[i+1]);

        for i = 1 to population_size
                old_solution[i] = new_solution[i];
until (termination);
end.
```

*reproduction = selection mechanism

Experiments

- Two different problems are solved using GeneAS.
- Two different problems were solved using different traditional optimization algorithms.
  - Lagrange multiplier method
    by Kannan and Kramer 1993
  - Branch and Bound method
    by Sandgren 1988
- GeneAS
  Crossover Probability =0.9
  Mutation Probability =0
  Selection mechanism = ?
  Population size =50
  Max generation =?

Gear Train Design

A compound gear train is to be designed to achieve a specific gear ratio between the driver and driven shafts (Figure 5). The objective of the gear train design is to find the number of teeth in

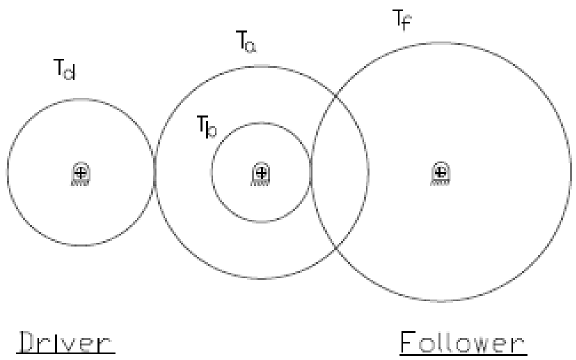


Figure 5: A compound gear train

each of the four gears so as to minimize the error between the obtained gear ratio and a required gear ratio of 1/6.931 (Kannan and Kramer, 1993). Since the number of teeth must be integers, all four variables are discrete. By denoting the variable vector $\boldsymbol{x} = (x_1, x_2, x_3, x_4) = (T_d, T_b, T_a, T_f)$., we write the NLP problem:

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = \left[\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right]^2 \\ \text{Subject to} \quad & 12 \le x_1, x_2, x_3, x_4 \le 60, \\ & \text{all } x_i\text{'s are integers.} \end{aligned}$$

- Since, all the four variables are discrete, each variable is coded in six-bit binary string (64 values), so that variables take values between 12 and 75.
- Four constraints are added to the objective function.

$$(60 - x_i \geq 0, \text{ for } i = 1, 2, 3, 4)$$

Result

Table 1: Gear train design

Design variables	Optimal solution (GeneAS)	(Kannan and Kramer)	(Sandgren)
x_1 (T_d)	17	13	18
x_2 (T_b)	14	15	22
x_3 (T_a)	33	33	45
x_4 (T_f)	50	41	60
$f(\boldsymbol{x})$	$1.362(10^{-09})$	$2.146(10^{-08})$	$5.712(10^{-06})$
Gear Ratio	0.144242	0.144124	0.146667
Error	0.026%	0.11%	1.65%

GeneAS have found much better solutions than the previously known optimal solutions.

Welded Beam Design

- A rectangular beam needs to be welded as a cantilever beam to carry a certain load.
- The objective of the design of the welded beam is to minimize the overall cost of the fabrication which involves the cost of beam material and the cost of weld deposit.
- Two different welding configurations can be applied.
  (1 is used for four-sided welding and 0 is used for two-sided welding) : X1

- One of the four materials can be used
  - (00-steel, 01-cast iron, 10-aluminum, 11-brass): X2
- h : X3
- t : X4
- b : X5
- l : X6

$$\underbrace{(0)}_{\text{weld type}} \quad \underbrace{(10)}_{\text{material}} \quad \underbrace{0.0625}_{h} \quad \underbrace{0.625}_{t} \quad \underbrace{0.0625}_{b} \quad \underbrace{1.563}_{\ell}$$

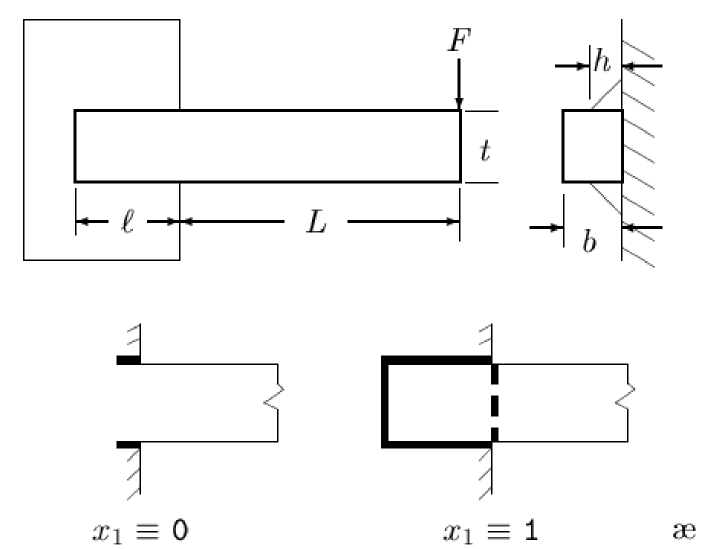


Figure 7: Welded beam problem

$$\begin{aligned}
\text{Minimize} \quad & f(\boldsymbol{x}) = (1+c_1)x_3^2(x_6 + x_1x_4) + c_2x_4x_5(L + x_6) \\
\text{Subject to} \quad & g_1(\boldsymbol{x}) = S - \sigma(\boldsymbol{x}) \geq 0, \\
& g_2(\boldsymbol{x}) = P_c(\boldsymbol{x}) - F \geq 0, \\
& g_3(\boldsymbol{x}) = \delta_{\max} - \delta(\boldsymbol{x}) \geq 0, \\
& g_4(\boldsymbol{x}) = 0.577S - \tau(\boldsymbol{x}) \geq 0,
\end{aligned}$$

$$\underbrace{(0)}_{\text{weld type}} \quad \underbrace{(10)}_{\text{material}} \quad \underbrace{0.0625}_{h} \quad \underbrace{0.625}_{t} \quad \underbrace{0.0625}_{b} \quad \underbrace{1.563}_{\ell}$$

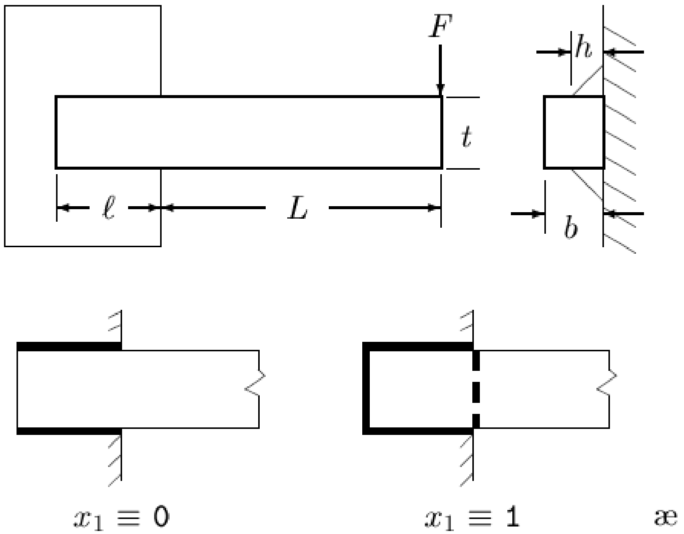


Figure 7: Welded beam problem

$$\begin{cases}
x_1 = 0, \quad A = 1.414x_3x_6, & J = 1.414x_3x_6\left[\frac{(x_3+x_4)^2}{4} + \frac{x_6^2}{12}\right], \quad \text{if } x_1 \text{ is } 0; \\
x_1 = 1, \quad A = 1.414x_3(x_4 + x_6), & J = 1.414x_3\frac{(x_3+x_4+x_6)^3}{12}, \quad \text{otherwise.}
\end{cases}$$

$$\begin{cases}
S = 30(10^3), \quad E = 30(10^6), \quad G = 12(10^6), \quad c_1 = 0.1047, \quad c_2 = 0.0481, & \text{if } x_2 \text{ is } 00 \text{ (Steel)}; \\
S = 8(10^3), \quad E = 14(10^6), \quad G = 6(10^6), \quad c_1 = 0.0489, \quad c_2 = 0.0224, & \text{if } x_2 \text{ is } 01 \text{ (CI)}; \\
S = 5(10^3), \quad E = 10(10^6), \quad G = 4(10^6), \quad c_1 = 0.5235, \quad c_2 = 0.2405, & \text{if } x_2 \text{ is } 10 \text{ (Al)}; \\
S = 8(10^3), \quad E = 16(10^6), \quad G = 6(10^6), \quad c_1 = 0.5584, \quad c_2 = 0.2566, & \text{if } x_2 \text{ is } 11 \text{ (Brass)}.
\end{cases}$$

Results

X1 = Four-sided X2 =Steel

X3 =0.1875 X4=8.2500 X5 =0.2500 X6 =1.6849

g1(x) =380.1660

g2(x) =402.0473

g3(x) = 0.2346

g4(x) = 0.1621

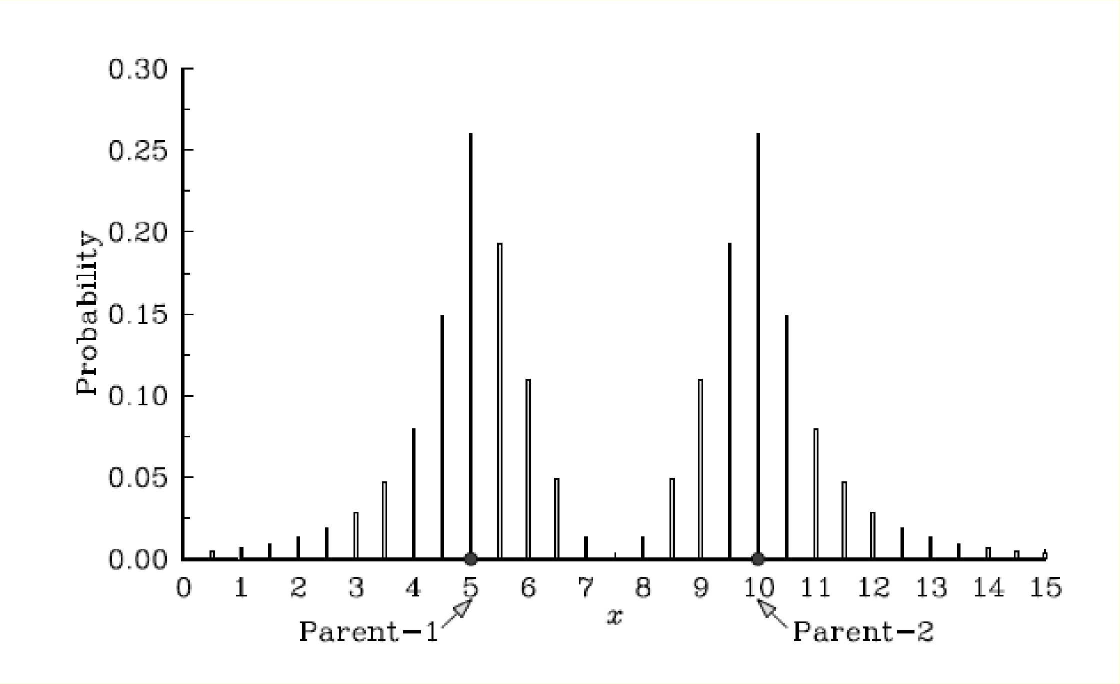
F(x) =1.9622

The above solution is likely to be very near to the true optimal solution, as one of the constraints (g_4) is very close to zero, meaning that the material is well utilized to have a shear strength of the weld equal to the allowable limit (In fact, the shear stress is only $9.4(10^{-4})\%$ smaller than the allowable limit). The bending stress is 1.3% smaller than the allowable limit and the critical buckling load is 6.7% higher than the applied load.

Conclusions

- This paper presents a proposed genetic algorithms for engineering optimization problems.
- The type variable can be different.
- SBX and real-mutation is proposed.
- The results show that GeneAS can find very good optimal solutions.

Discrete Variables



SBX

$$\mathcal{C}(\beta) = \begin{cases} 0.5(n+1)\beta^n, & \text{if } \beta \leq 1; \\ 0.5(n+1)\frac{1}{\beta^{n+2}}, & \text{otherwise.} \end{cases}$$

$$x_i^{(1,t+1)} = 0.5\left[(1+\bar{\beta})x_i^{(1,t)} + (1-\bar{\beta})x_i^{(2,t)}\right], \quad (5)$$

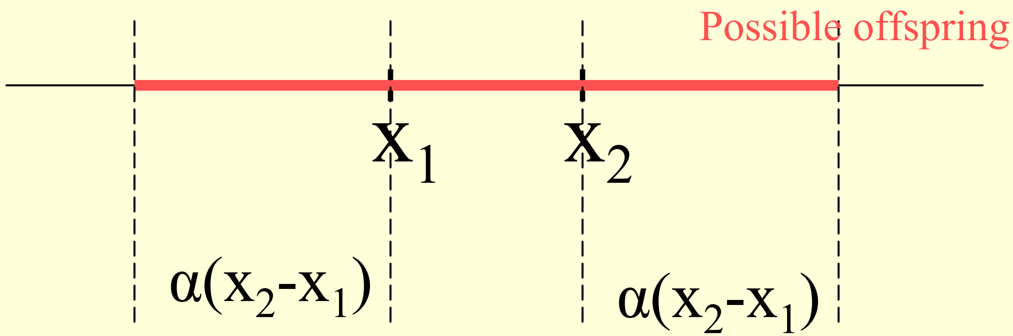
$$x_i^{(2,t+1)} = 0.5\left[(1-\bar{\beta})x_i^{(1,t)} + (1+\bar{\beta})x_i^{(2,t)}\right]. \quad (6)$$

```
float get_beta(u)
float u;
{
   float beta;

   if (1.0 - u < EPSILON ) u = 1.0 - EPSILON;
   if (u < 0.0) u = 0.0;
   if (u < 0.5) beta = pow(2.0*u,        (1.0/(n+1.0)));
   else         beta = pow((0.5/(1.0-u)),(1.0/(n+1.0)));
   return beta;
}
```

BLX-α

- Blend Crossover (BLX-α) was proposed by Eshelman and Schaffer (1993)
- BLX-α operator is used for real-parameter GAs.
- α represents a constant. (0,1)
- For two parents x_1 and x_2 (assume $x_1 < x_2$)
- BLX-α randomly picks a solution in the range $[x_1 - \alpha(x_2-x_1),\ x_2 + \alpha(x_2-x_1)]$



- The investigators have reported that BLX-0.5 (α=0.5) performs better than BLX operators with other α value.