SIMULATED BINARY CROSSOVER (SBX)

Cómputo Evolutivo

Arturo Márquez Flores

26 Marzo 2019

Maestría en Inteligencia Artificial
Universidad Veracruzana
CIIA – Centro de Investigación en Inteligencia Artificial
Sebastián Camacho No 5, Xalapa, Ver., México 91000
https://github.com/arturomf94/ce-mia

CONTENIDO

- Motivación
- SBX
- Ejemplos

MOTIVACIÓN

INTRODUCCIÓN

- La cruza es el operador principal.
- La cruza sirve para explorar nuevas y mejores alternativas.

INTRODUCCIÓN

- La cruza es el operador principal.
- La cruza sirve para explorar nuevas y mejores alternativas.
- **La codificación importa.**

ESPACIOS DISCRETOS

- Representación binaria.
- Cruza de un punto.

CRUZA BINARIA

B_1				A_1		
1	0	1	0	1	0	1
0	1	1	0	0	1	1
B_2				A_2		

Figura: Padres

B_1				A_2		
1	0	1	0	0	1	1
0	1	1	0	1	0	1
B_2				A_1		

Figura: Hijos

PROPIEDAD IMPORTANTE I

En general:

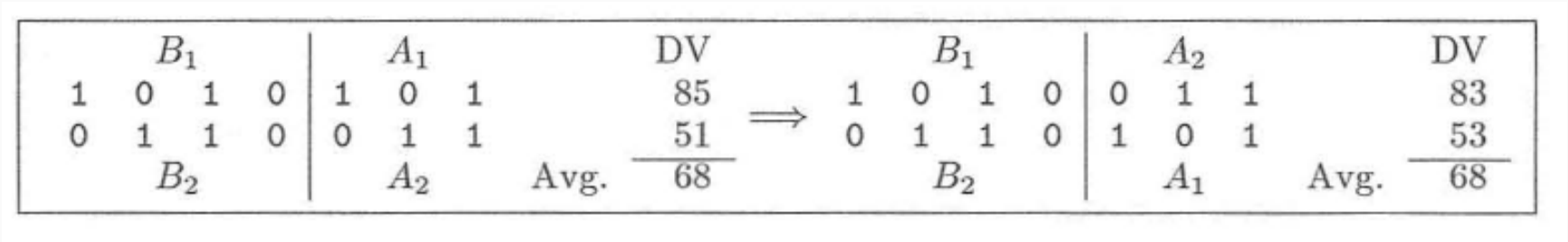
$$x_i = B_i 2^k + A_i$$

$$y_i = B_i 2^k + A_{-i}$$

Y

$$\frac{x_1 + x_2}{2} = \frac{y_1 + y_2}{2}$$

PROPIEDAD IMPORTANTE I



B_1				A_1				DV
1	0	1	0	1	0	1		85
0	1	1	0	0	1	1		51
B_2				A_2			Avg.	68

$\Longrightarrow$

B_1				A_2				DV
1	0	1	0	0	1	1		83
0	1	1	0	1	0	1		53
B_2				A_1			Avg.	68

Figura: Promedio se Mantiene

PROPIEDAD IMPORTANTE I

Esto se cumple incluso con una transformación lineal como:

$$p_i = mx_i + t$$

$$c = my_i + t$$

FACTOR DE DISPERSIÓN

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

PROPIEDAD IMPORTANTE II

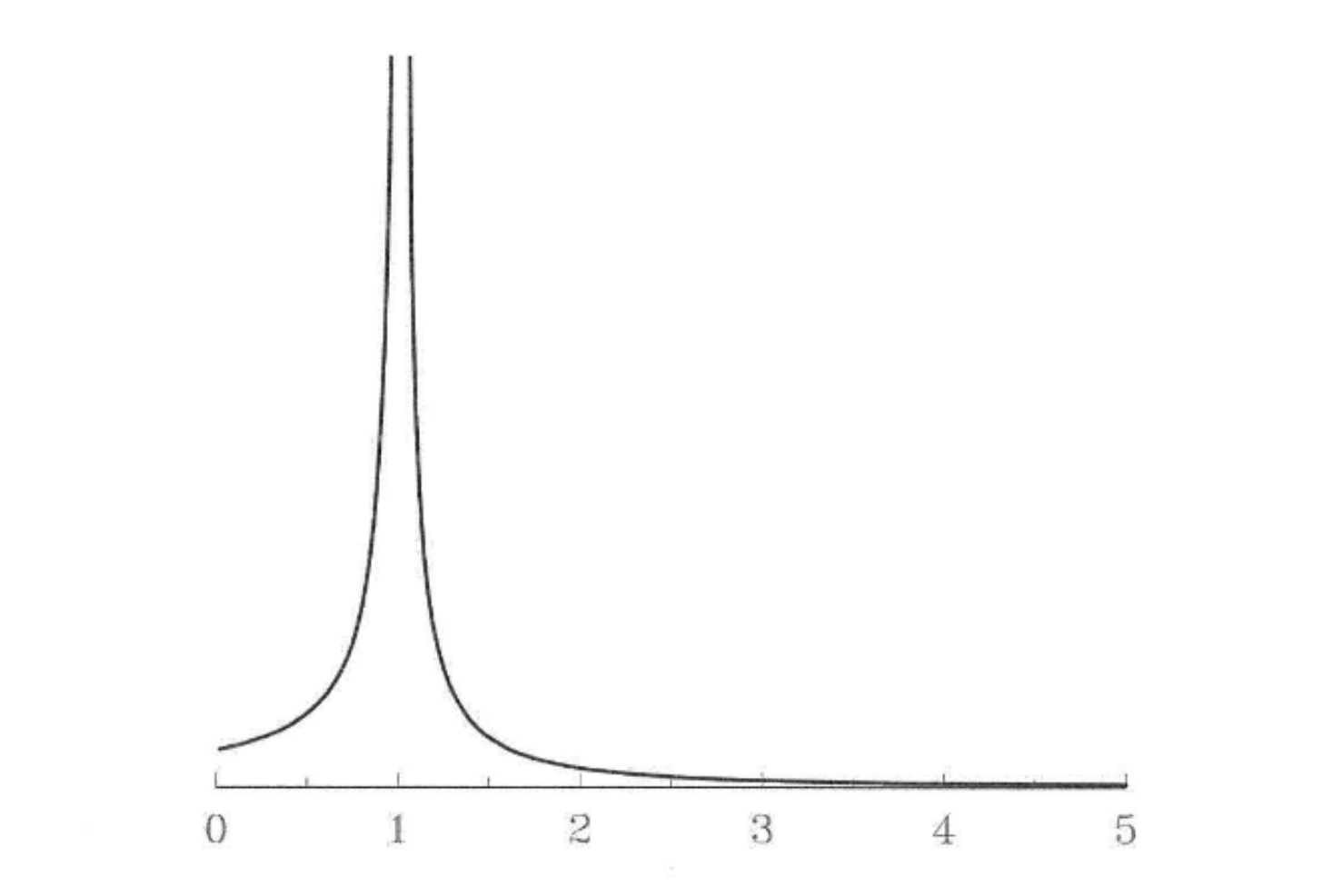


Figura: Distribución de β

PROPIEDAD IMPORTANTE II

Intuición:

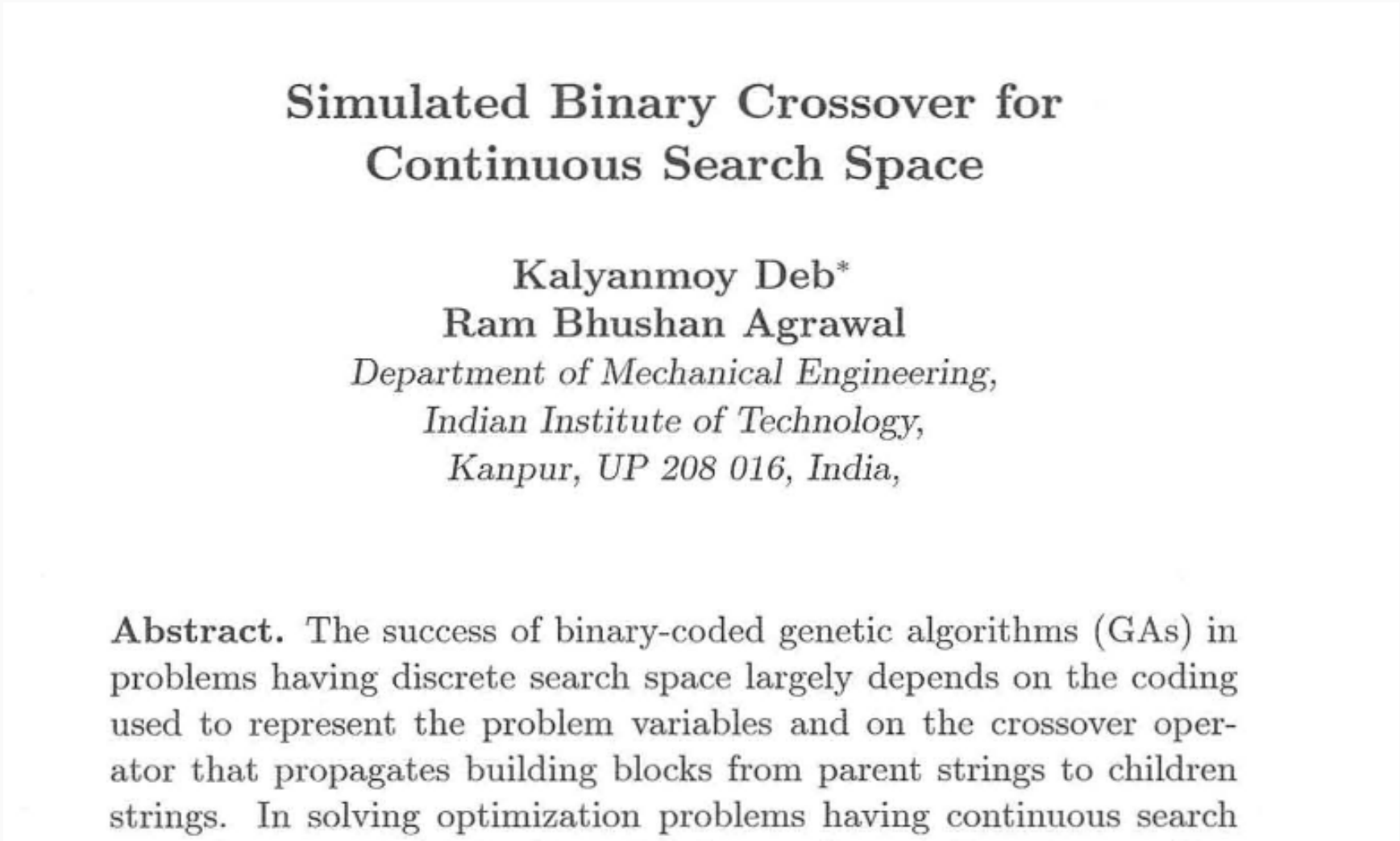
La dispersión de los padres es heredada a los hijos.

ESPACIOS CONTINUOS

Dificultades:

- Hamming Cliffs.
- Precisión.

SBX



Simulated Binary Crossover for Continuous Search Space

Kalyanmoy Deb*
Ram Bhushan Agrawal
Department of Mechanical Engineering,
Indian Institute of Technology,
Kanpur, UP 208 016, India,

Abstract. The success of binary-coded genetic algorithms (GAs) in problems having discrete search space largely depends on the coding used to represent the problem variables and on the crossover operator that propagates building blocks from parent strings to children strings. In solving optimization problems having continuous search

Figura: SBX

IDEA CENTRAL

- Importante replicar el poder de búsqueda de la representación binaria.
- Diseño de SBX simula este poder de búsqueda.

IDEA CENTRAL

- Importante replicar el poder de búsqueda de la representación binaria.
- Diseño de SBX simula este poder de búsqueda.
- **Preservar las propiedades importantes.**

SIMULACIÓN

$$C(\beta) = 0{,}5(n+1)\beta^{n}$$

$$E(\beta) = 0{,}5(n+1)\frac{1}{\beta^{n+2}}$$

Para valores altos de n hay menos varianza en la distribución. Los hijos pueden tener más dispersión que los padres.

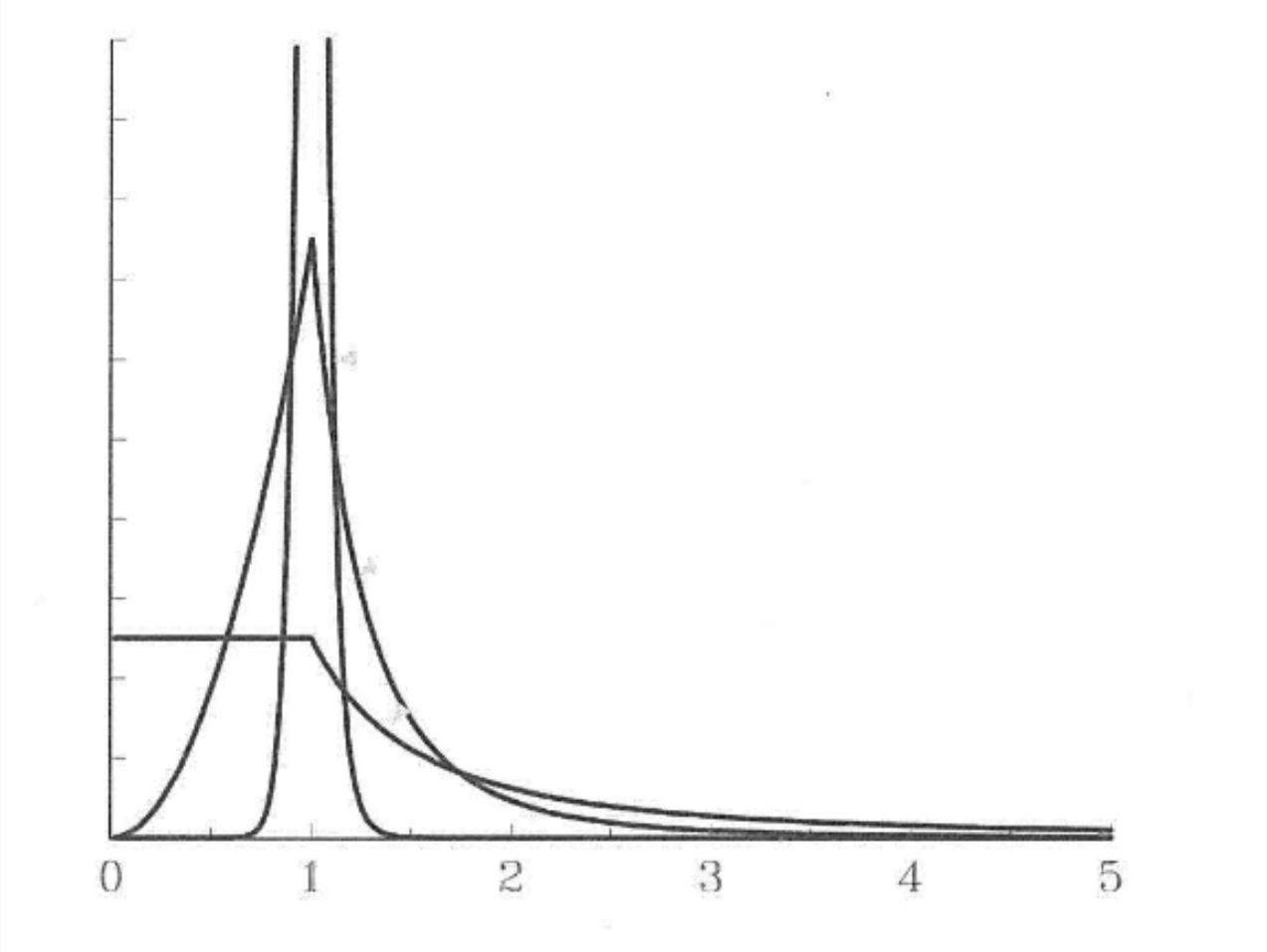


Figura: Distribución de β para diferentes n

Para ajustar una $\hat{\beta}$ basta con generar un número u aleatorio en el intervalo $[0, 1]$ y tomar la β tal que $\Pr(x <= \beta) = u$.

Luego los hijos se generan de la siguiente manera:

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

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

EJEMPLOS

EJEMPLOS

En **este colab** podemos experimentar con el operador SBX.