GENERALIZACION DEL OPERADOR DE CRUCE EN ALGORITMOS GENETICOS

Barrios, D., Ríos, J., Segovia, J.

Facultad de Informática de Madrid (U.P.M.)

Departamento de Inteligencia Artificial

Campus de Montegancedo, 28660 Madrid

Abstract

Cuando se utilizan algoritmos genéticos en problemas de optimización, existen ciertas limitaciones respecto de la convergencia. En este trabajo se plantea una estrategia para evitarlas, consistente en la construcción de un nuevo operador, el Cruce Generalizado, que capacita al algoritmo para acceder a mayor diversidad de puntos. Para ello, se parte de propiedades extraídas del cruce ordinario. A partir de ellas, y dado el fundamento teórico que proporcionan, se efectúa la mencionada generalización.

1. Introducción

Con el método de optimización proporcionado por los Algoritmos Genéticos se parte de una muestra aleatoriamente elegida en el dominio de búsqueda, llamada población, que se transforma de generación en generación [4]. Cuando se llega a un estado estacionario, se entiende que la población está constituida por las soluciones proporcionadas por el algoritmo. La transformación se realiza con la aplicación de diversos operadores, y el algoritmo actúa sobre una codificación del espacio de búsqueda.

La relación que existe entre progenitores y descendientes, mediante los operadores de cruce conocidos, tiene como consecuencia restricciones sobre los puntos a obtener. Por otra parte, el teorema de Holland [6] enuncia que la probabilidad de cada esquema de sobrevivir al cruce aumenta cuando la longitud y el orden de éste, con las definiciones habituales, disminuye. Si bien el orden no presenta problemas, ya que se puede pensar en la solución como intersección de esquemas de orden 1, la condición sobre la longitud resulta indeseable.

La solución que se propone en este trabajo consiste en la transformación del cruce, de modo que éste sea capaz de efectuar una búsqueda más exhaustiva, al tiempo que se logre una independencia absoluta entre la longitud de los patrones y su capacidad de sobrevivir.

2. Propiedades del cruce ordinario

Las propiedades y definiciones que se van a estudiar en esta sección establecen las bases necesarias para la generalización del operador de cruce.

Propiedad 2.1:

Sean x_1 y x_2 son dos elementos cualesquiera del espacio de búsqueda. Sean x'_1,x'_2 el resultado del cruce genético de los elementos anteriores. Entonces se tiene que:

$$x_1+x_2=x'_1+x'_2$$

En lo que sigue, x_1, x_2 son dos elementos del espacio de búsqueda, de representación binaria $a_0a_1...a_n$ y $b_0b_1...b_n$ respectivamente. La representación decimal de dichos elementos será $a_0+a_1.2+...+a_n.2^n$ y $b_0+b_1.2+...+b_n.2^n$ respectivamente.

Definición 2.1:

*Se llama "x_1 ó x_2" al punto que resulta como consecuencia de aplicar la función lógica **or** a las respectivas codificaciones de x_1 y x_2. Dicho elemento de escribirá como:*

$$x_1 \vee x_2$$

y su expresión en binario será $d_0d_1...d_n$, siendo d_i=Máx $\{a_i, b_i\}$. Se llama "cadena máxima" de x_1 y x_2 a esta última cadena.

Definición 2.2:

*Se define "x_1 y x_2" como el punto que resulta de aplicar la función lógica **and** a las respectivas codificaciones de x_1 y x_2. Dicho elemento se escribirá como:*

$$x_1 \wedge x_2$$

y su expresión en binario será $d_0d_1...d_n$, siendo d_i=Mín $\{a_i, b_i\}$. Se llama "cadena mínima" de x_1 y x_2 a esta última cadena.

Propiedad 2.2:

Dados los puntos x_1 y x_2 del espacio de búsqueda, suponiendo que se cruzan y que el resultado del cruce son los puntos x'_1 y x'_2, se tiene:

$$i)\quad x_1 \wedge x_2 \le x_1, x_2 \le x_1 \vee x_2$$
$$ii)\quad x_1 \wedge x_2 \le x'_1, x'_2 \le x_1 \vee x_2$$

Es decir, tanto los los puntos que se cruzan, como los descendientes de éstos mediante dicho cruce, están limitados al mismo intervalo, llamado en adelante *"intervalo de cruce"*, determinado por los extremos correspondientes a las definiciones 2.1 y 2.2.

3. Definición y análisis del operador de Cruce Generalizado

Con el nuevo operador de cruce que se propone en este trabajo, llamado *Cruce Generalizado* en adelante, se podrá acceder a cualquier punto del intervalo de cruce, y únicamente a ellos. Para ello, si dos elementos cualesquiera de la población resultan elegidos para la aplicación del operador, el resultado serán los elementos x'_1 y x'_2 tales que:

$$x'_1 \in [x_1 \wedge x_2,\ x_1 \vee x_2],\ x'_2 = (x_1 + x_2) - x'_1$$

siendo x'_1 un punto, elegido aleatoriamente en el intervalo, siempre que los valores alcanzados por la función objetivo en éstos puntos supere, o al menos iguale, la que se tenía antes de hallar x'_1 y x'_2. En caso contrario, se tomará como producto del cruce aquellos dos puntos, elegidos entre los progenitores y los descendientes, cuyas imágenes sean las mayores.

En el siguiente lema se prueba que el resultado de aplicar el nuevo operador a dos puntos se mantiene dentro del intervalo de cruce.

Lema 3.1:

Sea $\boldsymbol{y=(x_1+x_2)-x}$, tal que:

$$x \in [x_1 \wedge x_2 \, , \, x_1 \vee x_2]$$

Entonces se tiene:

$$y \in [x_1 \wedge x_2 \, , \, x_1 \vee x_2]$$

Demostración:

Basta tener en cuenta la propiedad:

$$x_1 + x_2 - (x_1 \wedge x_2) = x_1 \vee x_2$$

ya que,por ser $y=(x_1+x_2)-x$, se tienen las desigualdades siguientes:

$$(x_1 + x_2) - (x_1 \vee x_2) \leq y \leq (x_1 + x_2) - (x_1 \wedge x_2)$$

de donde:

$$x_1 \wedge x_2 \leq y \leq x_1 \vee x_2$$

El siguiente resultado formaliza la idea de que el cruce generalizado conserva e! concepto de "herencia" [2], transmitida en forma de parte común entre cadenas, y establecido el otros operadores de cruce. El significado de esto es que, si bien es posibl(que los descendientes se alejen entre si, existe cierto control entre la distancia quє guardan.

Supóngase cierta generación formada por los puntos $x_1, x_2, \ldots x_p$ del espacio de búsqueda. Supóngase elegidos dos puntos, x_1 y x_2 sin pérdida de generalidad, para el cruce generalizado. Sea **I** el intervalo de cruce para estos puntos. Si las cadenas correspondientes a x_1 y x_2 son respectivamente:

$$x^1{}_1 x^1{}_2 \ldots x^1{}_m x^1{}_{m+1} \ldots x^1{}_n$$
$$x^2{}_1 x^2{}_2 \ldots x^2{}_m x^2{}_{m+1} \ldots x^2{}_n$$

y se tiene:

$$x^1{}_i = x^2{}_i \, , \; \forall i \geq m+1$$

siendo **m** el máximo entero en estas condiciones, entonces la longitud del intervalo **I** está dada por:

$$\rho(I)=\sum_{x^1{}_i\neq x^2{}_i,\, i\leq m} 2^i$$

Lema 3.2:

$$\rho(I)<2^{m+1} \Leftrightarrow \forall y\in[x_1\wedge x_2\,,\,x_1\vee x_2]\,,\ si\ y=\sum_{i=1}^{l} y_i 2^{i-1}\,,\ y_i\in\{0,1\},$$

$$se\ tiene\ y_i=x^1{}_i=x^2{}_i\ ,\forall i\geq m+1$$

Demostración:

i) Suponiendo la longitud del intervalo acotada por 2^{m+1}, por poderse expresar dicha longitud como

$$\rho(I)=\sum_{\substack{x^1{}_i\neq x^2{}_i\\ i=1,\ldots,n}} 2^i<2^{m+1}$$

se tiene que

$$x^1{}_i\neq x^2{}_i,\forall i\geq m+1$$

Por tanto, cualquier punto y contenido en el intervalo, si se expresa en la forma

$$y=\sum_{i=1}^{n} y_i 2^{i-1},\ y_i\in\{0,1\}$$

como debe ser

$$Min\{x^1{}_i,x^2{}_i\}\leq y_i\leq Max\{x^1{}_i,x^2{}_i\},\forall i$$

se tiene que

$$y_i=x^1{}_i=x^2{}_i,\forall i\geq m+1$$

ii) Recíprocamente, si existe al menos un punto y en el intervalo que verifica

$$y_i=x^1{}_i=x^2{}_i$$

para todo i≥m+1, se tiene que la cadena mínima y la cadena máxima, para x_1 y x_2, coincide en las posiciones correspondientes a i≥m+1. Por tanto, la longitud del intervalo será

$$\rho(I)=\sum_{\substack{x^1{}_i\neq x^2{}_i\\ i\leq m}} 2^i$$

evidentemente acotado por 2^{m+1}.

4. Resultados de orden práctico

En general quedó probada una mejor convergencia usando el nuevo operador. Incluso con ejemplos donde existen dificultades para los algoritmos genéticos convencionales, con el nuevo operador de cruce mejora la convergencia. En otros casos, en los que se probó funciones que conducen a problemas paradójicos con el operador ordinario, se llegó al resultado deseado con el cruce generalizado.

En todos los casos, se utilizó codificación en binario, correspondiendo cada cadena de longitud 10 a la expresión en base dos del punto correspondiente, y siendo el intervalo de búsqueda el [0 , 1023]. Todos los parámetros, que aparecen en la tabla 1, se mantuvieron constantes durante los experimentos.

Tamaño de población	Probab. cruce	Probab. mutación
50	.5	.01

-Tabla 1: Parámetros-

En la tabla 2, la segunda columna corresponde al óptimo buscado, la tercera al alcanzado usando cruce generalizado, y la última al alcanzado con cruce ordinario. En las dos últimas columnas se indica, entre paréntesis, el número de generaciones de convergencia en cada caso.

$f_1(x)=(128/3)(x-500)$, $500< x <512$ $f_1=100$ en el resto.	512	512 (4)	781 (20)
$f_2(x)=x$	1023	1023 (9)	944 (19)
$f_3(x)=(x-1023)^2$	0	0 (9)	2 (20)
$f_4(x)=1/(x+1)$	0	0 (5)	4 (4)
$f_5(x)=x$, $x < 1023/2$ $f_5(x)=-(x-1023)$,resto	511	511 (10)	404 (8)

-Tabla 2: Funciones test-

5. Conclusiones

Se ha definido un operador de cruce, con el cual es posible una búsqueda más exhaustiva en todo el dominio, resolviendo problemas de convergencia subóptima que, de otro modo, pueden presentarse, tanto debido a problemas paradójicos, como a escaso tamaño de la población o a otras dificultades intrínsecas de cada problema.

Con el cruce generalizado se produce una mayor diversidad de población en cada generación, con lo cual se evita la convergencia prematura. El teorema de Holland se modifica, resultando independiente la probabilidad de supervivencia de cada patrón de su longitud. Por tanto, se obtiene una base teórica que permite deducir la mejor convergencia cuando se utiliza este operador.

Referencias

[1] J. D. Bagley, The behavior of adaptative systems which employ genetic and correlation algorithms, Doctoral dissertation, University of Michigan (1976).

[2] J. E. Baker, Adaptive selection methods for genetic algorithms, Proceeding of an International Conference on Genetic Algorithms and Their Applications, 101-111,

(1985).

[3] K. A. De Jong, Analysis of Behavior of a Class of Genetic Adaptive Systems, Ph. D. dissertation, University of Michigan (1975).

[4] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning (Addison-Wesley Publishing Company, Inc. 1989)

[5] J. H. Holland, Adaptation in natural and artificial systems (The University of Michigan Press, 1975).

[6] G. Sywerda, Uniform Crossever in Genetic Algorithms, submited to the 1989 International Conference on Genetic Algorithms.

[7] D. Whitley, Using Reproductive Evaluation to Improve Genetic Search and Heuristic Discovery, Genetic Algorithms and their Applications: Proceeding of the Second International Conference on Genetic Algorithms (1987) 108-115.

[8] D. Barrios, Operador de Cruce Generalizado en Algoritmos Genéticos, Tesis Doctoral. Octubre 1991.

[9] D. Barrios, A. Pazos, J. Ríos, J. Segovia, Oscillanting Problems in Genetics Algorithms, enviado y recibido por Complex Systems, USA.