

Evolución diferencial

Introducción

Para resolver problemas computacionales complejos, se ha buscando inspiración en la naturaleza.

La optimización está en el corazón de muchos procesos naturales como la propia evolución darwiniana.

A través de millones de años, todas las especies tuvieron que adaptar sus estructuras físicas para adaptarse a los ambientes que eran.

La computación evolutiva utiliza progreso iterativo, como el crecimiento o el desarrollo de una población. Donde la población es “filtrada” como parte de una búsqueda guiada para lograr el fin deseado .

Introducción

El paradigma de las técnicas de computación evolutiva se remonta a principios de 1950, con la idea de usar los principios darwinianos para resolver problemas de forma automatizada.

Fue en los años sesenta cuando tres interpretaciones diferentes de esta idea comenzaron a desarrollarse.

Programación evolutiva (EP) (Lawrence J. Fogel, EU).

Estrategias evolutivas (ES) (I. Rechenberg y H.-P. Schwefel, Alemania).

Algoritmos genéticos (John Henry Holland, Ann Arbor).

Estas áreas se desarrollaron por separado durante unos 15 años. En los 90's se unificaron como diferentes representantes de la llamada computación evolutiva.

A principios de los noventa, emerge una cuarta corriente con las mismas ideas generales, Programación genética (GP).

Introducción

Actualmente el campo de metaheurísticas inspiradas en la naturaleza está constituida por:

- Algoritmos evolutivos: GA, EP, ES, GP, diferencial evolución (DE) , etc.
- Algoritmos de inteligencia colectiva: optimización de colonia de hormigas (ACO) , optimización por enjambre de partículas (PSO) , colonia de abejas, optimización de forrajeo bacteriana (OFB) .

También el campo se extiende en un sentido más amplio para incluir sistemas auto-organizativos, vida artificial (organismos digitales), algoritmos meméticos y culturales [S15] , búsqueda armónica, sistemas inmunes artificiales.

Evolución diferencial

Evolución diferencial (DE) surge como una versión competitiva de la computación evolutiva.

El primer artículo escrito en DE aparece como un informe técnico por R. Storn y KV Price en 1995.

DE adquiere prestigio en diferentes concursos internacionales.

- En el Primer Concurso Internacional de Optimización Evolutiva en mayo de 1996 que se celebró conjuntamente con la Conferencia IEEE 1996 Internacional sobre Computación Evolutiva (CEC).
- Termina en tercer lugar en el Primer Concurso Internacional de Optimización Evolutiva (primera ICEO), que se celebró en Nagoya, Japón.
- El mejor algoritmo evolutivo para resolver la suite de prueba de la primera ICEO (los dos primeros lugares se les dio a algoritmos no evolutivos, que no son de aplicación universal, pero resuelven los problemas de prueba más rápido que DE) .

Evolución diferencial

Ventajas de DE


En comparación con otros AEs , DE es mucho más simple y sencilla de implementar.


- El cuerpo principal del algoritmo toma de cuatro a cinco líneas de código en cualquier lenguaje de programación.


Independientemente de su simplicidad, DE ha demostrado un mejor desempeño que otras técnicas de optimización.



El número de parámetros en DE es muy pequeño, y sus efectos sobre el rendimiento del algoritmo se han estudiado a profundidad.

Evolución diferencial



JournalsBooksSign in 

Advanced search

Search results: 305,306 results found. [See image results](#)  Save search alert  RSS






Refine filters


Year


☐ 2017 (20)
☐ 2016 (11,822)
☐ 2015 (18,548)
☐ 2014 (16,710)
☐ 2013 (15,971)
[View more >>](#)


Publication title


☐ Physics Letters A (2,884)
☐ Physica D: Nonlinear Phenomena (2,631)
☐ Journal of Computational Physics (2,602)
☐ Physica A: Statistical Mechanics and its Applic... (2,548)
☐ Journal of Mathematical Analysis and Applicatio... (2,199)
[View more >>](#)

  Download PDFs  Export  Relevance  All access types

☐ **An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization** Original Research Article
Information Sciences, Volume 348, 20 June 2016, Pages 124-141
Xianpeng Wang, Lixin Tang
[Abstract](#)  PDF (1130 K)

☐ **Quasi-oppositional differential evolution for optimal reactive power dispatch** Original Research Article
International Journal of Electrical Power & Energy Systems, Volume 78, June 2016, Pages 29-40
M. Basu
[Abstract](#) [Research highlights](#)  PDF (915 K)

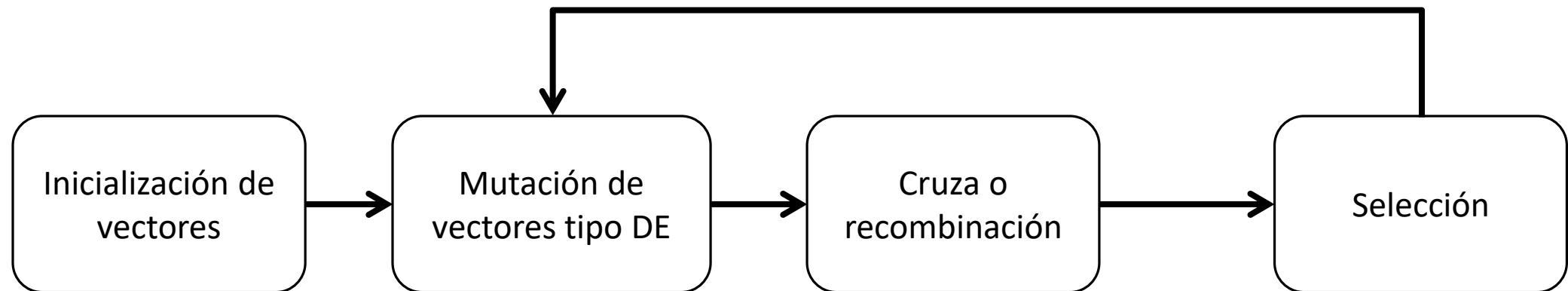
☐ **Mathematical approach assisted differential evolution for generator maintenance scheduling** Original Research Article
International Journal of Electrical Power & Energy Systems, Volume 82, November 2016, Pages 508-518
G. Balaji, R. Balamurugan, L. Lakshminarasimman
[Abstract](#) [Research highlights](#)  PDF (982 K)

☐ **A two-step approach for damage detection in laminated composite structures using modal strain energy method and an improved differential evolution algorithm** Original Research Article
Composite Structures, Volume 147, 1 July 2016, Pages 42-53
T. Vo-Duy, V. Ho-Huu, H. Dang-Trung, T. Nguyen-Thoi
[Abstract](#)  PDF (1284 K)

Conceptos básicos y formulación en un espacio real continuo

DE es un algoritmo de optimización continua.

Trabaja mediante un ciclo que se resume en la siguiente figura



Inicialización de vectores

Inicialización de vectores

DE busca un óptimo en un espacio D-dimensional, en \mathbb{R}^D .

Inicia con una población aleatoria de NP vectores en \mathbb{R}^D .

Cada vector, genoma/cromosoma, es una solución para el problema de optimización multidimensional.

Cada generación en DE se denota como $G = 0, 1, \dots, G_{\max}$.

Como cada vector cambia durante las diferentes generaciones, se representan como:

$$X_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}]$$

Inicialización de vectores

Para inicializar los vectores se deben tener en cuenta el rango de las variables, denotados como

$$X_{\min} = \{x_{1,\min}, x_{2,\min}, \dots, x_{D,\min}\}$$

$$X_{\max} = \{x_{1,\max}, x_{2,\max}, \dots, x_{D,\max}\}.$$

De esta forma, la j-ésima componente del i-ésimo vector será

$$x_{j,i,0} = x_{j,\min} + rand_{i,j}[0, 1] \cdot (x_{j,\max} - x_{j,\min})$$

Mutación con diferencia de vectores

Mutación con diferencia de vectores

Biológicamente, "mutación" significa un cambio repentino en las características genéticas de un cromosoma.

En computación evolutiva, mutación es visto como un cambio o perturbación con un elemento aleatorio.

Terminología

- Un vector padre de la generación actual se llama vector objetivo.
- Un vector mutante obtenido a través de la operación de mutación diferencial se conoce como vector donante.
- Una descendencia formada por recombinar el donante con el vector objetivo se denomina vector de prueba .

Mutación con diferencia de vectores

Una de las formas más simples de DE-mutación es la siguiente

Para cada individuo $X_{i,G}$ en la población se genera un vector donante.

Se elijen tres vectores distintos al vector objetivo, $X_{r_1}^i$, $X_{r_2}^i$, y $X_{r_3}^i$.

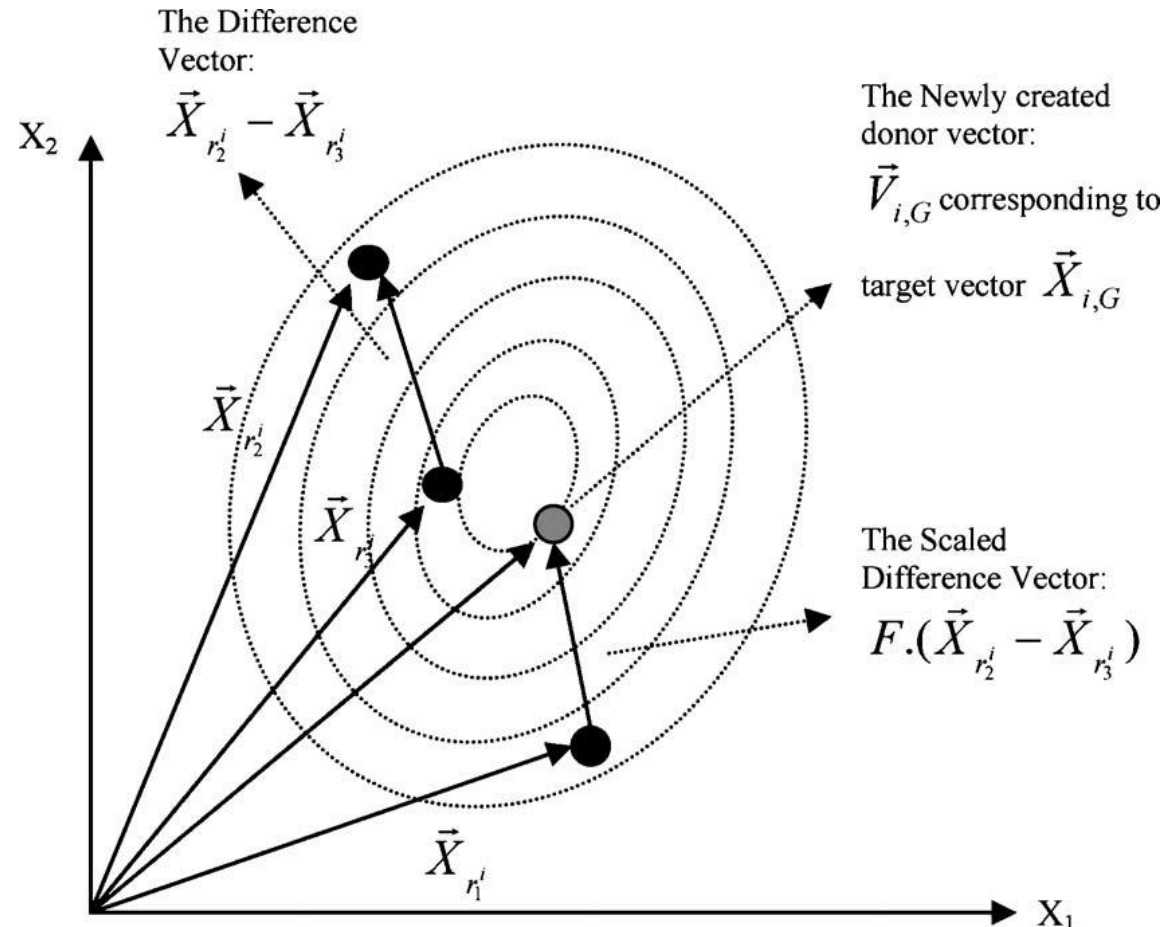
r_1^i , r_2^i , r_3^i son generados con una distribución uniforme en $[1, NP]$.

El vector donante es generado de la siguiente forma:

$$V_{ri,G} = X_{ri1,G} + F \cdot (X_{ri2,G} - X_{ri3,G})$$

F, es un parámetro de control de DE

Mutación con diferencia de vectores



Cruza

Cruza

Para mejorar la diversidad de la población, se realiza una cruce después de generar al vector donante.

El vector donante intercambia sus componentes con el vector de destino, para generar el vector de prueba

$$U_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \dots, u_{D,i,G}].$$

Existen dos formas de realizar la cruce

- Exponencial
- Binomial

Cruza exponencial

Se elige aleatoriamente un entero n en $[1, D]$.

Este entero indica desde donde se inicia el cruce o cambio de componentes entre el vector objetivo y el vector donante.

Se elige otro entero L en el intervalo $[1, D]$, que denota el número de componentes que el vector donante aporta al vector objetivo.

Después de elegir n y L el vector prueba se obtiene como

$$u_{j,i,G} = v_{j,i,G} \text{ for } j = [n]_D, [n + 1]_D, \dots, [L - 1]_D$$

$$u_{j,i,G} = x_{j,i,G} \text{ for all other } j \in [1, D]$$

$[n + 1]_D$, representa la función módulo D .

Cruza exponencial

El entero L es generado en el intervalo $[1, D]$ de acuerdo al siguiente ciclo

```
 $L = 0;$   
DO  
{  
     $L ++;$   
} WHILE ( $(rand(0, 1) \leq Cr)$  AND  $(L \leq D)$ ).
```

Cruza exponencial

Cr se llama la tasa de cruce y es un parámetro de control de la DE como F .

Para cada vector donante , nuevos valores de n y L deben seleccionarse.

Cruza binomial

Por otro lado, la cruce binomial se realiza en cada una de las D variables cada vez que un número generado en $[0, 1]$ es menor o igual que Cr .

En este caso, el número de parámetros heredado del donante tiene (casi) una distribución binomial.

El esquema puede ser descrito como

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } (rand_{i,j}[0, 1] \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,G} & \text{otherwise} \end{cases}$$

Cruza binomial

$\text{rand}_{i,j}[0, 1]$, es un número aleatorio distribuido de manera uniforme, nuevo para cada j -ésima componente del i -ésimo vector.

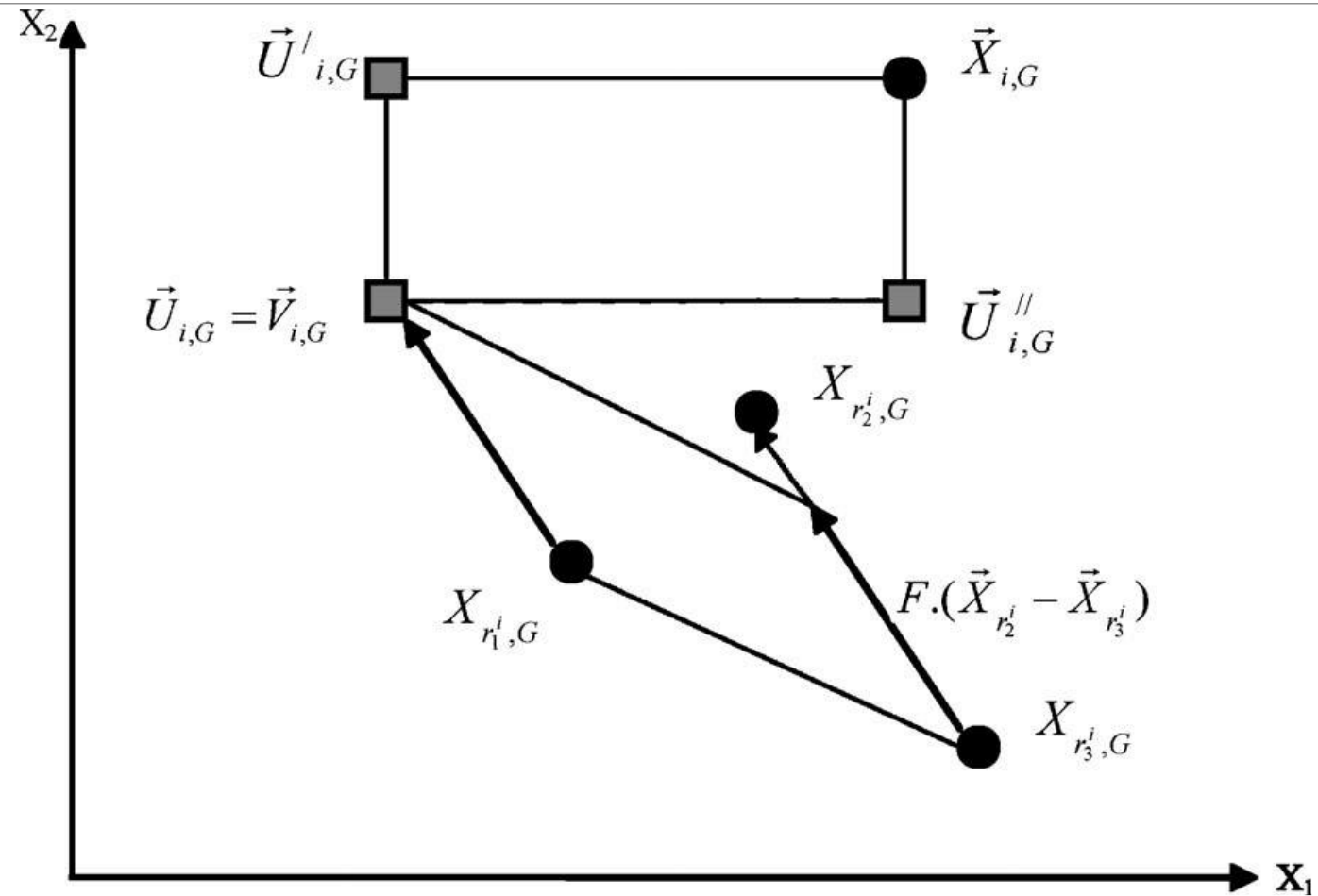
$j_{rand} \in [1, 2, \dots, D]$ es un índice elegido al azar, que asegura que $U_{i,G}$ tiene al menos un componente de $V_{i,G}$. En este caso, se crea una vez por cada vector por generación.

Cruza binomial

En dos dimensiones se pueden obtener 3 posibles resultados

- 1) $U_{i,G} = V_{i,G}$, las dos componentes de $U_{i,G}$ son heredadas de $V_{i,G}$.
- 2) $U_{i,G}$, la primer componente ($j = 1$) viene de $V_{i,G}$ y la segunda ($j = 2$) de $X_{i,G}$.
- 3) $U_{i,G}$, la primer componente ($j = 1$) viene de $X_{i,G}$ y la segunda ($j = 2$) de $V_{i,G}$.

Cruza binomial



Selección

Selección

Para mantener el tamaño de la población constante, se requiere de un proceso de selección que determine si el vector objetivo o el vector de prueba sobrevivirá a la siguiente generación.

La operación de selección se describe como

$$\begin{aligned}\vec{X}_{i,G+1} &= \vec{U}_{i,G} && \text{if } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ &= \vec{X}_{i,G} && \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G})\end{aligned}$$

Selección

Por lo tanto, si el vector de prueba produce un valor igual o inferior de la función objetivo, sustituye al vector objetivo correspondiente.

En otro caso se retiene el objetivo en la población.

De esta forma, la población mejora o permanece en el mismo estado de fitness, pero nunca se deteriora.

Nótese que en el vector objetivo se sustituye por el vector prueba incluso si ambos dan el mismo valor de la función objetivo. Esto permite que los vectores se puedan mover incluso sobre superficies planas (flat fitness landscapes).

DE iteración y criterio de paro

Iteración y criterio de paro

Consiste de 4 pasos:

- Inicializar una población
- Mutación
- Cruza o recombinación
- Selección

Se puede detener el algoritmo cuando:

- Se alcanza un máximo número de generaciones, G_{max}
- Cuando no hay cambios importantes en la función objetivo
- Cuando se ha alcanzado un valor predefinido en la función objetivo

Terminología

Terminología

Hasta ahora hemos visto los pasos básicos de DE.

El esquema de mutación se basa en la suma de un vector base X_1 , más la diferencia ponderada de dos vectores diferentes $F(X_2 - X_3)$.

Este esquema se conoce como DE/rand/1.

Cuando se usa junto con una cruce binaria se representa como DE/rand/1/bin.

En general se usa una representación del tipo DE/x/y/z, donde:

- x representa al vector base
- y es el número de diferencias entre vectores consideradas
- z es el tipo de cruce considerada

Familia de Storn y Price

Hay 4 esquemas de mutación propuestos por Storn y Price

- 1) “*DE/best/1* :” $\vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G})$
- 2) “*DE/target – to – best/1* :” $\vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G})$

Familia de Storn y Price

3) “*DE/best/2* :” $\vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G})$
 $+ F \cdot (\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i,G})$

4) “*DE/rand/2* :” $\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G})$
 $+ F \cdot (\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i,G}).$ (

Familia de Storn y Price

Storn y Price propusieron diez estrategias diferentes para DE y algunas pautas en la aplicación de estas estrategias.

Estas estrategias se derivaron de los cinco sistemas diferentes de mutación DE descritos anteriormente.

Cada estrategia de mutación se combina con el cruce exponencial o binomial.

Dando un total de $5 \times 2 = 10$ DE estrategias.

En general, ningún método de mutación ha resultado ser mejor para todos los problemas.

Esquemas de mutación

No obstante diversos esquemas de mutación necesitan más investigación para determinar en qué circunstancias se desempeñan bien y en qué tipo de problemas tienen resultados pobres.

Mezura-Montes et al. presentaron un trabajo en esta dirección.

Empíricamente compararon ocho esquemas diferentes en un conjunto de 13 problemas de referencia.

Los autores tuvieron en cuenta un esquema de mutación conocido como DE / rand / 2 / dir, que incorpora la información de la función objetivo para guiar la dirección de los vectores de los donantes de la siguiente manera:

Esquemas de mutación

$$\vec{V}_{i,G} = \vec{X}_{r_1,G} + \frac{F}{2} \cdot (\vec{X}_{r_1,G} - \vec{X}_{r_2,G} - \vec{X}_{r_3,G})$$

Donde $X_{r_1,G}$, $X_{r_2,G}$, and $X_{r_3,G}$ son elementos diferentes de la población tales que $f(X_{r_1,G}) \leq \{f(X_{r_2,G}), f(X_{r_3,G})\}$.

Los experimentos realizados por Mezura-Montes et al. indican que el DE/best/1/bin es el esquema más competitivo independientemente de las características del problema, con base en la precisión final y robustez de los resultados .

Diferencias y similitudes con EA

Diferencias y similitudes. Mutación

En el contexto de GAs y EA, la mutación es un cambio al azar de algún parámetro.

EA normalmente simulan la mutación con incrementos acumulativos generados aleatoriamente con una distribución de probabilidad predefinida y fija.

DE muta los vectores base (padres secundarios) con diferencias ponderadas.

A medida que pasan las generaciones, estas diferencias tienden a adaptarse a la escala natural de la problema. Por ejemplo, si la población se hace compacta en una variable pero sigue siendo ampliamente dispersa en otra, la diferencia de los vectores mantendrán esta característica.

Esta adaptación automática mejora significativamente la convergencia del algoritmo .

Diferencias y similitudes. Mutación

Uno de los aspectos más importantes de este tipo de mutación es que las perturbaciones vectoriales se generan a partir de $NP \cdot (NP - 1)$ diferencia de vectores de la población.

Esto nos lleva a una de las principales ventajas de DE: *contour matching*.

Contour matching, se refiere a la adaptación de la población de tal manera que las regiones prometedoras se investigan de forma automática una vez que se detectan.

Una de las ventajas de la resta entre vectores es que tanto el tamaño de la mutación como su orientación se adaptan automáticamente al paisaje de la función objetivo.

Diferencias y similitudes. Mutación

Price *et al.* mencionan que contour matching también induce a la transferencia de una cuenca a otra, donde los puntos de búsqueda pueden pasar de una cuenca de atracción, es decir, un mínimo local, a otro.

Diferencias y similitudes. Cruza

Tanto DE y ES emplean cruza para crear un único vector de prueba.

La mayoría GAs recombinan dos vectores para producir dos vectores de prueba a menudo por un punto de cruce.

Una de las técnicas de cruza para GAs real es la cruza en n puntos en la cual se dividen aleatoriamente en $(n+1)$ bloques y las secciones en particiones adyacentes se heredan de padres diferentes.

Estudios indican que un número par de puntos de cruza reduce el sesgo de representación.

Diferencias y similitudes. Cruza

La cruza exponencial de DE emplea uno y dos puntos de cruza con el objetivo de reducir sus sesgos individuales.

El sesgo de representación inherente a la cruza en n puntos puede ser eliminado si los donantes se determinan por D ensayos aleatorios independientes.

Diferencias y similitudes. Selección

La selección se aplica a un proceso evolutivo en dos diferentes etapas

- La selección de padres para decidir qué vectores se someterán a recombinación
- La selección del superviviente entre padres e hijos.

A diferencia de GAs, tanto ES y DE dan a todos los individuos las mismas oportunidades de ser seleccionados como padres, sin considerar su fitness.

Cuando sólo avanzan los vectores descendencia se puede perder la mejor solución encontrada.

Retener la mejor solución, elitismo, juega un papel importante para lograr la convergencia del algoritmo al óptimo global.

Por esta razón y la mejora de la velocidad, la mayoría de las EA incluyendo DE, EP, y algunas versiones de ES tienen en cuenta la población actual, para determinar a los miembros de la siguiente generación.

Parámetros de control

Parámetros de control

Hay tres principales parámetros de control para DE:

- El factor de escala mutación F
- La constante de cruza C_r ,
- El tamaño de la población NP

Storn y Price proponen:

- Para NP podría ser elegido un valor entre el $5 \cdot D$ y $10 \cdot D$ (D es la dimensión del problema)
- Una opción inicial de F de 0.5
- El rango efectivo de F es por lo general entre 0.4 y 1 .

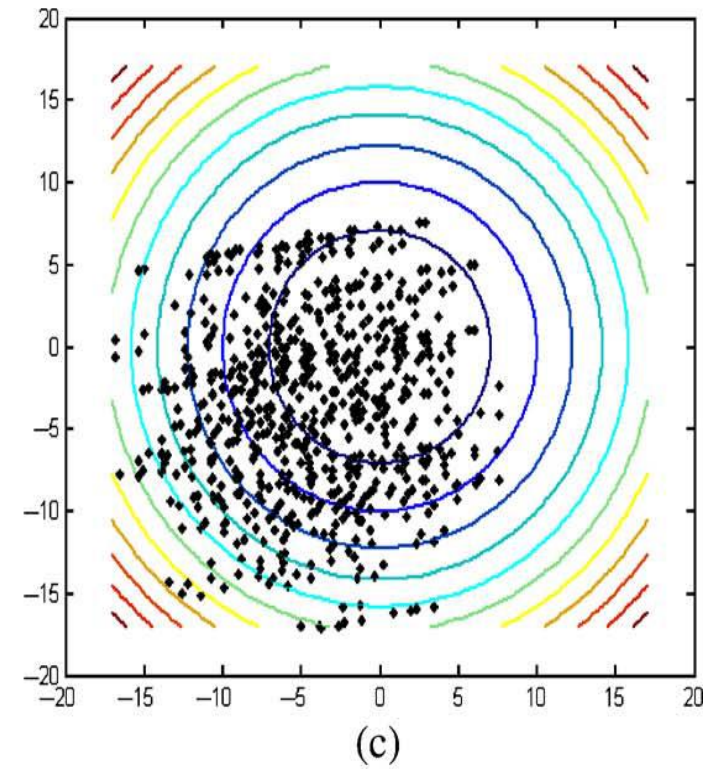
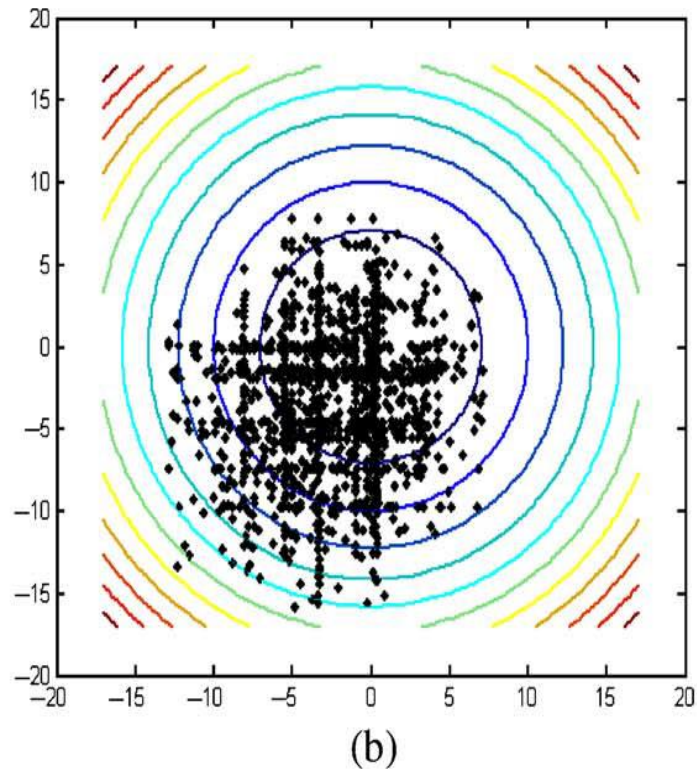
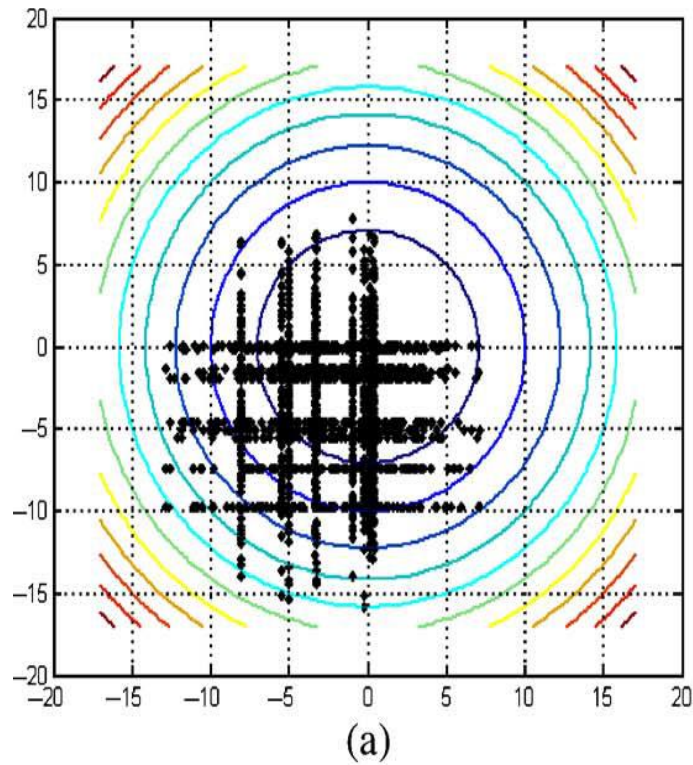
Parámetros de control

El parámetro Cr controla cuántos parámetros son cambiados en un individuo.

Para un valor bajo de Cr , se cambian un pequeño número de parámetros en cada generación, y el movimiento tiende a ser ortogonal al eje de coordenadas actual.

Para valores altos de Cr (cerca de 1) se provoca que la mayoría de las direcciones del vector mutante sean heredadas, prohibiendo la generación de movimientos ortogonales.

Parámetros de control



Parámetros de control

Gämperle et al. evaluaron diferentes ajustes de parámetros en la Esfera, Rosenbrock, y Rastrigin.

Sus resultados revelaron que la capacidad de búsqueda y la velocidad de convergencia son muy sensibles a la elección de los parámetros de control de NP , F, y Cr.

Además, una opción de parámetros es:

- NP entre 3-D y 8-D
- F = 0.6
- Cr entre [0.3 , 0.9]

Otros autores afirman que $0.4 < F < 0.95$ con F = 0.9 pueden servir como una buena primera opción.

Auto-adaptación

Debido a la complejidad y delicadeza en la selección de los parámetros, muchos autores han recurrido a técnicas de auto-adaptación.

Normalmente se aplica para determinar los valores de F y Cr .

Por ejemplo, F , es aproximado por una distribución normal con media 0.5 y desviación estándar 0.3.

Se generan un conjunto de valores para F , y se aplican a cada vector objetivo en la población. De esta forma, se mantiene intensificación, con valores pequeños, y exploración con valores grandes.

Auto-adaptación

Evaluación del fitness de F

$$F = \begin{cases} \max \left\{ l_{\min}, 1 - \left| \frac{f_{\max}}{f_{\min}} \right| \right\} & \text{if } \left| \frac{f_{\max}}{f_{\min}} \right| < 1 \\ \max \left\{ l_{\min}, 1 - \left| \frac{f_{\min}}{f_{\max}} \right| \right\} & \text{otherwise} \end{cases}$$

Donde $l_{\min} = 0.4$ es el límite inferior de F , f_{\min} y f_{\max} son los valores máximo y mínimo de la función objetivo de los individuos en una generación.

Auto-adaptación

El ajuste de Cr puede hacerse en base a los resultados obtenidos previamente.

Se puede suponer que sigue una distribución normal, con media Cr_m , y desviación estándar 0.1 para no salirse de $[0,1]$.

El valor de Cr_m puede inicializarse en 0.5 y modificarse conforme el algoritmo avanza.

Observe que surgen otros parámetros a calibrar, pero su ajuste suele ser menos sensible.

Auto-adaptación

Se asigna a cada individuo parámetros F y Cr , junto con una probabilidad de heredarlos.

Los sobrevivientes tendrán mejores costos y propagarán sus parámetros.

$$F_{i,G+1} = \left. \begin{array}{ll} F_l + rand_1 * F_u & \text{with probability } \tau_1 \\ = F_{i,G} & \text{else} \end{array} \right\}$$

Donde F_l y F_u son

$$\text{and } Cr_{i,G+1} = \left. \begin{array}{ll} rand_3 & \text{with probability } \tau_2 \\ = Cr_{i,G} & \text{else} \end{array} \right\} \quad] .$$

Auto-adaptación

Se ha propuesto que F varíe aleatoriamente entre 0.5 y 1 para cada vector.

Los mismos autores sugieren que F decrezca de forma lineal de 1 hasta 0.5. De esta forma, al principio se promueve la exploración, y al final se realiza un proceso de intensificación.

Parámetros

Aunque existen menos estudios sobre el tamaño de la población, se han propuesto técnicas para disminuir el número de individuos empleados, de tal forma que se mejore la velocidad y robustez del algoritmo.

También se han propuesto poblaciones de tamaño auto-adaptativo.