

Algoritmos Genéticos

Introducción

- Computación evolutiva (CE) engloba los procesos de adaptación inspirados en mecanismos de la selección natural, usando operadores genéticos mantienen a una población en constante evolución.
- La computación evolutiva tiene un dominio muy amplio y extenso la cual va desde la optimización numérica de una variable hasta la optimización multiobjetivo pudiendo involucrar diferentes variables en el proceso.

Introducción

- En la actualidad la CE esta bien afianzada dentro de las ciencias de la computación debido a ciertas características:
 - No necesitan conocimiento específico sobre el problema que intentan resolver, resolviendo inclusive problemas para los cuales no se conoce solución alguna.
 - Son conceptualmente muy simples y tienen amplia aplicabilidad.
 - Pueden explorar fácilmente las arquitecturas en paralelo, en caso de estar disponibles.

Introducción

- Pueden ser agrupados en :
 - Programación evolutiva
 - Estrategias evolutivas
 - Algoritmos genéticos
- Existe también una variación del algoritmo genético que esta siendo mas popular estos tiempos que es la programación genética.

Programación Evolutiva

- La programación evolutiva se fundamenta en las ideas de Lawrence J. Fogel quien concibió el uso de la evolución simulada en la solución de problemas por los años 60. En esta técnica la inteligencia se ve como un comportamiento adaptativo y se enfatiza los nexos de comportamiento entre padres e hijos.
- Puede considerarse como una abstracción al nivel de especies por lo que no requiere el uso de un operador de recombinación. Asimismo usa selección probabilística.

Programación Evolutiva

El algoritmo básico de la programación evolutiva es:

Iniciar

Generar aleatoriamente la población inicial

Repetir

Aplicar mutación

Calcular la aptitud de cada hijo

Seleccionar las soluciones que permanecerán

Reemplazar la población actual por la seleccionada

Hasta que cierta condición se satisfaga

Fin

Estrategias Evolutivas

- A mediados de los 60 fue desarrollado un método de ajustes discretos aleatorios inspirados en el mecanismo de mutación que ocurre en la naturaleza. La técnica creada fue denominada “**estrategias evolutivas**”. Fueron usadas inicialmente para resolver problemas hidrodinámicos de alto grado de complejidad.
- La versión original usaba solo un padre y con él se generaba un solo hijo. Ese hijo se mantenía si era mejor que el padre de lo contrario se eliminaba.

Estrategias Evolutivas

Iniciar

Generar aleatoriamente la población inicial, $G(0)$

Sea $t = 0$;

Evaluar $G(t)$

Repetir

Sea $t = t + 1$,

Seleccionar $G1(t)$ de $G(t)$

Recombinar $G1(t)$ para generar $G(t+1)$

Mutar

Evaluar

Supervivientes

Hasta que cierta condición se satisfaga

Fin

Algoritmos Genéticos

Definición

Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual (cruzamiento). Cada uno de estos objetos matemáticos suele ser una cadena de caracteres de longitud fija, que ajusta el modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

Algoritmos Genéticos

Terminología

Genética: es la ciencia que estudia los mecanismos de transmisión de características de una especie de una generación para otra.

Población: conjunto de individuos los cuales corresponde a un conjunto de soluciones iniciales.

Individuos: también es llamado de cromosoma y corresponden a una posible solución, el cual es asociado a un valor que es llamado de aptitud.

Función de aptitud: grado de aptitud de un individuo, y mide la capacidad de la solución(individuo) para la solución de determinado problema.

Flujo grama de AG



Flujo grama de AG

Características de los AG

- Los AG trabajan sobre un conjunto de parámetros codificados es decir efectúan las operaciones sobre los cromosomas y no sobre el problema.
- Realizan la búsqueda bajo un conjunto de puntos (paralelismo implícito)
- Usan la información de la función de aptitud, no requiere otros conocimientos.
- Las reglas de transición son probabilísticas no determinísticas.

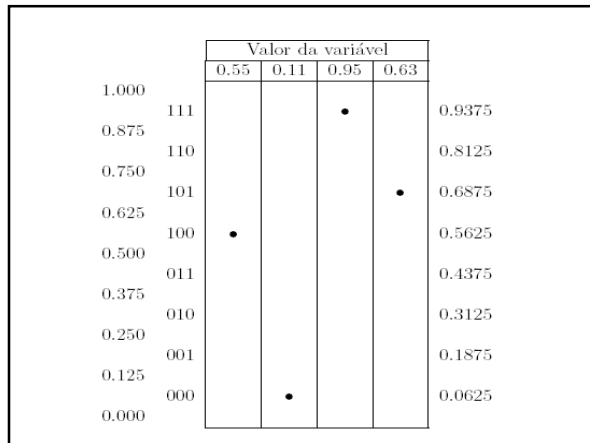
Representación de soluciones

- Cada individuo de una población generada por el AG, representa una posible solución del problema que se desea resolver. Así las variables de la función objetivo deben ser representadas por un individuo.
 - Por ejemplo considere la siguiente función objetivo $f(x)=x^2$. Todos los individuos de cualquier población deben representar la variable x . Una posible representación es una simple binarización de los números, otra solución podría ser usar los mismos valores enteros. Escoger una adecuada representación es un área de interés para los AG.

Representación de soluciones

Representación binaria

- Una de las representaciones mas utilizadas en los AG es la representación binaria ideal para problemas donde las variables son discretas. Por lo tanto si las variables son continuas una conversión es necesaria.
- Por ejemplo si tenemos valores en el intervalo $[0,1]$ que debe ser representado como cadenas binarias de tamaño 3



Representación de soluciones

Representación continua

- Para casos donde el error ocasionado por la representación binaria es critica los AG pueden usar directamente los valores continuos. En este caso ninguna conversión es necesaria. No en tanto se debe implementar operadores genéticos de cruzamiento y mutación adecuados a la representación.

Operadores genéticos

Selección para la reproducción

El objetivo principal del operador de selección es copiar las mejores soluciones eliminando las soluciones de baja aptitud, mientras el tamaño de la solución es constante. Esto es realizado siguiendo los siguientes pasos.

1. Identificar las mejores soluciones de la población
2. Realizar múltiples copias de las mejores soluciones
3. Eliminar soluciones de baja aptitud lo que permite que varias copias de las mejores soluciones puedan ser insertadas en la población.

Operadores genéticos

Selección proporcional

$$P_i = \frac{F_i}{\sum_{i=1}^N F_i}$$

Siendo F_i la aptitud de la solución, N es el tamaño de la población. Por tanto la probabilidad de ser escogidos será, 14.4%, 49.2%, 5.5% y 30.9%, que puede ser representado como una ruleta



Operadores genéticos

Selección por torneo

En la selección por el torneo son realizadas varias competiciones entre dos soluciones y la mejor solución es copiada en la lista de soluciones. Este proceso es repetido hasta llenar la lista. Goldber demostró que este método posee una convergencia igual o mejor que las otras estrategias de selección además de poseer una complejidad menor que las otras. Para obtener el número de copias esperado, es necesario obtener la probabilidad de cada solución P_i .

El número de copias en la lista de soluciones es calculado por $C_i = P_i N$. O sea, las soluciones con mejor valor de aptitud tendrán más copias en la lista de soluciones.

Operadores genéticos

Selección por ranking.

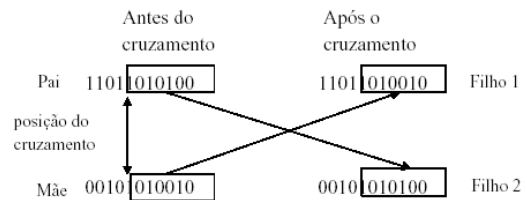
Esta solución ordena la población por el valor de su aptitud, desde la peor solución 1 hasta la mejor N siendo el número de copias proporcional a su ranking.

Operadores genéticos

Cruzamiento: Este operador genera nuevas soluciones a partir de las soluciones escogidas de la lista de soluciones. El operador de cruzamiento posee diferentes variaciones, muchas de ellas específicas a un determinado problema. La forma más simple de cruzamiento es conocida como cruzamiento de un punto, que consiste en:

1. Escoger arbitrariamente dos individuos de la lista de soluciones
2. Escoger dentro de la cadena del individuo una posición k llamada posición de cruzamiento
3. Crear nuevos descendientes cambiando las cadenas parciales de cada uno de los individuos.

Operadores genéticos



Operadores genéticos

• Mutación

La mutación es un operador que produce una alteración aleatoria en una posición de un pequeño número de individuos. La mutación es la segunda manera de los AGs explorar el espacio de búsqueda. Esta pequeña alteración impide que el algoritmo genético tenga convergencia muy rápida, evitando su estabilización en regiones de mínimos locales.

• Elitismo

El operador de elitismo mantiene las mejores soluciones encontradas previamente en las generaciones siguientes.

Algoritmos genéticos

Pasos para implementar un AG

- Para resolver un determinado problema utilizando AG los siguientes pasos deben ser considerados:
 1. Definir una representación a ser usada para cada individuo de manera que una solución completa puede ser representada
 2. Definir las estrategias de sustitución, selección, cruzamiento y mutación
 3. Definir la función de aptitud
 4. Ajustar los siguientes parámetros:
 - Tamaño de la población
 - Probabilidad de cruzamiento
 - Probabilidad de mutación
 - Número de generaciones

Ejemplo

Considere el problema de maximizar la función
 $f(x) = x^2$ en un intervalo de $[0, 31]$

Nº	Población inicial (generada aleatoriamente)	Valor x	$F(x) = x^2$	Pseleci $\frac{f_i}{\sum f}$	Valor Esperado $\frac{f_i}{f}$	Valor Actual
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Suma			1170	1.00	4.00	4.0
Media			293	0.25	1.00	1.0
Máximo			576	0.49	1.97	2.0

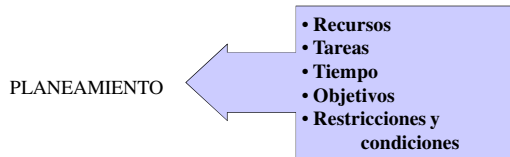
Ejemplo

Segunda generación

Cadenas seleccionadas para cruzamiento	Pareja Para cruce	Posición de cruz.	Nueva Población	Valor x	$F(x) = x^2$
0 1 1 0 1	2	4	0 1 1 0 0	12	144
1 1 0 0 0	1	4	1 1 0 0 1	25	625
1 1 0 0 0	4	2	1 1 0 1 1	27	729
1 0 0 1 1	3	2	1 0 0 0 0	16	256
Suma					1754
Media					439
Máximo					729

Algoritmos Genéticos en Optimización de Planeamiento

- En todas las actividades hay planeamiento que involucra:



- Optimizar:
 - “Asignar, en el tiempo los recursos para ejecutar tareas, respetando las restricciones y condicionamientos tales que, se alcancen los objetivos del problema.”

27

Ejemplos

- Problema del vendedor
- Optimización de a Distribución/Recoleccion (Ruta de vehículos)
- Asignación de Espacio Físico y Recursos: hospitales, aulas, etc
- Planeamiento de vuelos de aerolíneas
- Planeamiento Operación (embarque, generación, etc)
- Planeamiento Financiero
- Optimización de la Producción Industrial

28

Variables Típicas en Planeamiento

- Restricciones de Recursos**
 - número de instancias de cada recurso/máquina;
 - diferencia entre las instancias (velocidad, tiempo máximo de operación, capacidad, etc).
 - paradas de manutención
- Restricciones Temporales**
 - horario preferencial de funcionamiento;
 - tiempo de transporte de material entre máquinas
- Reajuste de las Máquinas**
 - tareas exigen preparación de las máquinas (automático ou manual).

29

Variables Típicas en Planeamiento

- Prioridad**
 - tareas con prioridades diferentes (plazo de entrega, emergencia, mantenimiento, tipo de cliente, etc.)
- Stock**
 - materia prima e de consumo: disponibilidad, orden de almacenamiento, etc.
- Reprogramación**
 - reprogramación de los recursos bajo contingencias
- Precedencia**
 - Algunas tareas no se pueden programar antes de otras ya finalizadas.

30

Características del Planeamiento

- Existen muchas **condiciones** y **restricciones** que no se pueden expresar matemáticamente;
- Los métodos de búsqueda fallan por requisitos de tiempo;
- Reducir el espacio de búsqueda disminuye el tiempo de ejecución pero limita la performance;
- Las Heurísticas son útiles** para acelerar la búsqueda;
- GA es una técnica que se adapta a problemas poco estructurados como los de planeamiento

31

Ejemplo simple de Planeamiento de Producción

- 90 tareas:** (a, b, c, d, e, f.....ab, ac.....)
- cada tarea tiene un peso por su importancia (lucro, prioridad, beneficio etc)
- 30 recursos:** solo una instancia por cada recurso
- Tareas con tiempo de ejecución de **1 a 3 horas**
- Programar para un periodo de **40 horas de producción**
- Algunas tareas tienen **restricciones**
- Objetivo:** maximizar la suma de los pesos de las tareas planeadas en las primeras 40 horas (1 semana)

32

Ejemplo de Tareas:

a..u

Tiempo de ejecución de cada tarea: 1, 2 6 3 horas

Recursos requeridos por cada tarea: “.” = recurso no requerido

“a” = recurso requerido

Tareas	Tiempo Ejecución	Recursos Utilizados por las Tareas
a	3	..a.....aa.....aa.....a.....
b	1	..b.....b.....b.....b.....
c	3	..c.....c.....c.....c.....
d	3	..d.....d.....d.....d.....
e	3	..e.....e.....e.....e.....
f	1	..f.....f.....f.....f.....
g	3	g.....g.....g.....g.....
h	2	h.....h.....h.....h.....
i	1i.....i.....i.....i.....
j	1j.....j.....j.....j.....
k	1k.....k.....k.....k.....
l	1l.....l.....l.....l.....
m	3m.....m.....m.....m.....
n	2n.....n.....n.....n.....
o	3o.....o.....o.....o.....
p	1p.....p.....p.....p.....
q	3q.....q.....q.....q.....
r	1r.....r.....r.....r.....
s	3s.....s.....s.....s.....
t	1t.....t.....t.....t.....

Ejemplo de Planeamiento

- En orden de recepción de las tareas (alfabética)
- Planeamiento es una matriz 40x3, cuyos elementos pertenecen a un alfabeto de 90 símbolos (tareas)
- EB= 90^{1200} , muchas soluciones son planeamientos inválidos

Tempo

Hora	1	2	3	4	5	6	7	8	9	10	11
1	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
2	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
3	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
4	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
5	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
6	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
7	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
8	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
9	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
10	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....
11	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....	..a.....c.....c.....a.....a.....c.....c.....c.....a.....c.....c.....

Modelo de Algoritmo Genético

- Problema
- Representação
- Decodificação
- Avaliação
- Operadores
- Técnicas
- Parâmetros

35

Representação

- Cromosoma \equiv permutación (lista) de tareas
- EB= 90!

$$P_1 = (a, b, c, d, e, \dots, t)$$

$$P_2 = (d, s, e, h, g, \dots, i)$$

- El cromosoma codifica el **orden y la vez** (posición) en los cuales se planearán las (colocadas en la matriz 40x30)
- Se exige un **constructor** de planeamientos válidos

36

Construir soluciones válidas

Concentra todo el conocimiento del problema:
restricciones, recursos, horarios, etc.

Regla Principal:

“Si una tarea está planeada para la hora t , no se puede colocar ningún otra en t , excepto cuando no haya intersección de recursos solicitados”

Decodificador del Problema de Planeamiento

- 1 Elige la primera tarea de la lista;
- 2 Coloca la tarea en el planeamiento a partir de $t=1$;
- 3 Elige la próxima tarea e intenta colocarla en el planeamiento considerando las restricciones presentes, a partir de $t=1$ até $t=40$ horas;
- 4 Vuelve al paso 3 mientras no se acabe la lista.

$$P = (\underline{t}, c, s, a)$$
$$P = (\underline{t}, \underline{c}, s, a)$$
40

$$P = (\underline{t}, \underline{c}, \underline{s}, a)$$

$P = (\underline{t}, \underline{c}, \underline{s}, \underline{a}) \sqsubseteq 7 \text{ horas}$

42

[illegible]

t	a	-	-	-	-	a	a	-	-	a	a	t	-	-	t	t	a	-	-	-	-	-	-	-
-	a	-	-	-	-	a	a	-	-	a	a	-	-	-	a	-	a	-	-	-	-	-	-	-
-	a	-	-	-	-	a	a	-	-	a	a	-	-	-	a	-	a	-	-	-	-	-	-	-

1
2
3
4
5
6
7
8
9
10
11

1	.	a	.	s	s	.	a	a	s	s	.	a	a	.	t	s	.	t	t	.	a	s	.	.	.	s	.	.	.	1
2	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	2
3	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	3
4	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	4
5	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	5
6	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	6
7	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	7
8	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	8
9	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	9
10	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	10
11	.	a	.	s	s	.	a	a	s	s	.	a	a	.	s	.	s	.	.	a	s	.	.	.	s	11

[illegible]

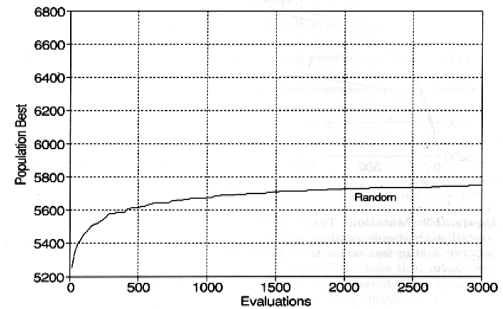
8

Evaluación de Performance

- Búsqueda Aleatoria para comparar resultados
 - crear una lista de tareas y evaluar
- Espacio de Representación = $90! = 10^{138}$
- Media de 50 Experimentos de 3000 Individuos: media de los mejores en cada instante

49

Búsqueda Aleatoria



Mutación Basada en Posición

- Intercambia las posiciones de dos tareas seleccionadas aleatoriamente.
- Ejemplos:

(a b c d e f)

↓

(e b c d a f)

(a b c d e f)

↓

(a d c b e f)
- Este operador no mantiene el orden relativo de las posiciones seleccionadas con respecto

51

Mutación Basada en Orden

- Se seleccionan dos tareas aleatoriamente y la última se pone adelante de la primera
- Ejemplos:

(a b c d e f)

↓

(e a b c d f)

(a b c d e f)

↓

(a d b c e f)
- Este operador mantiene el orden relativo de parte del cromosoma

52

Mutación de desorden

- Desordena una sub-lista seleccionada aleatoriamente

(a | b c d | e f)

↓

(a | c d b | e f)

e l)

(a b c | d e f)

↓

(a b c | f d e)
- Este operador tiene más fuerza de

Disminución de la mutación

Crossover de Orden

- Posiciones escogidas al azar
- Posiciones de las tareas en un progenitor se imponen en las tareas correspondientes en el otro progenitor

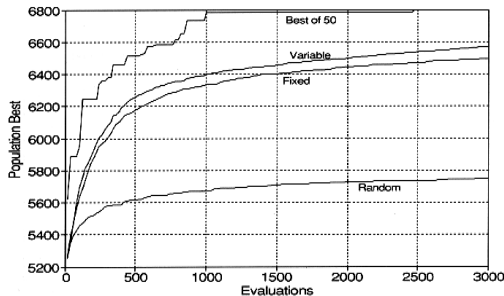
$P_1 = a \ b \ c \ d \ e \ f \ g \ h \ i \ j$
 $P_2 = e \ i \ b \ d \ f \ a \ j \ g \ c \ h$
 Patrón 0 1 1 0 1 0 0 1 0 0

$F_1 = _ \ i \ b \ _ \ f \ _ \ g \ _ \$
 $F_2 = _ \ b \ c \ _ \ e \ _ \ h \ _ \$

$F_1 = a \ i \ b \ c \ f \ d \ e \ g \ h \ j$
 $F_2 = i \ b \ c \ d \ e \ f \ a \ h \ j \ g$

54

Variando Probabilidades de Crossover + Mutación



Algoritmos Evolutivos para Optimización Multi-Objetivo

Problemas Multi-Objetivos

- Aplicaciones reales tienen múltiples criterios para evaluarse: **múltiples objetivos**
- Ejemplos:
 - Un producto comercial, minimizar el **costo** y maximizar la **calidad**
 - Un circuito eléctrico, minimizar el **área ocupada** y **potencia** disipada, maximizando la **ganancia**
 - Un portafolio de inversiones: maximizar la **rentabilidad** y reducir el **riesgo**

Problemas Multi-Objetivos

- Optimizar cada objetivo **por separado** no da buenos resultados.
- Si se optimiza un solo objetivo, **los demás pueden no ser satisfechos**
- Técnicas tradicionales de optimización, como *Simulated Annealing*, *Gradient Descent* y **Algoritmos Genéticos** se basan en una **evaluación global única**

¿Cómo comparar soluciones?

- **Solución 1**
 - Objetivo 1: 50% satisfecho
 - Objetivo 2: 30% satisfecho
- **Solución 2**
 - Objetivo 1: 40% satisfecho
 - Objetivo 2: 45% satisfecho
- **¿Cuál solución es mejor?**
 - Depende de la importancia de los objetivos

¿Cómo comparar soluciones?

- Una estrategia es **combinar las evaluaciones** de cada objetivo para tener una evaluación global por cada solución:

$$\text{Aptitud} = F(f_1, f_2, \dots, f_n)$$

- Normalmente sumatorias o productorias

Métodos de Evaluación

- Agregación de Objetivos
- Estrategia de Pareto
- Minimización de la Distancia al Objetivo
- Minimización de Energía (ICA)

Agregación de Objetivos

- Suma ponderada de los objetivos

$$F = \sum_{i=1}^n w_i f_i$$

donde w_i representa el peso atribuido a cada objetivo

Agregación de Objetivos

Ejemplo:

- *Solución 1*
 - Objetivo 1: 50% satisfecho
 - Objetivo 2: 30% satisfecho
- *Solución 2*
 - Objetivo 1: 40% satisfecho
 - Objetivo 2: 45% satisfecho
- Ej: para pesos $w_i=1$:
 - Solución 1: $50+30 = 80$
 - Solución 2: $40+45 = 85$
- Para $w_1=1$ e $w_2=0,5$:
 - Solución 1: $50+15=65$
 - Solución 2: $40+22,5=62,5$

Agregación de Objetivos

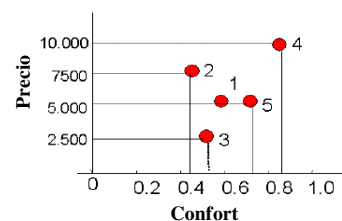
- Ventajas
 - Simple para implementar
 - Puede usarse en técnicas tradicionales de optimización
- Desventajas
 - Performance baja en la práctica: exige aplicar **conocimientos específicos** del problema para la especificación óptima de los pesos w_i
 - Exige que se **normalicen las evaluaciones** para sumar de forma comparable entre sí

Formalización de la Optimización Multi-objetivo

$$\left. \begin{array}{ll} \text{maximizar/minimizar} & f_m(\mathbf{x}), \quad m = 1, 2, \dots, M \\ \text{restrita a} & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, J; \\ & h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n. \end{array} \right\}$$

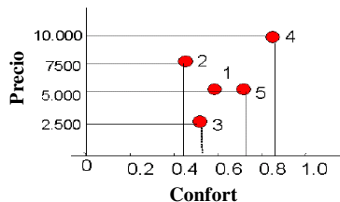
donde: Cada f_m es una función objetivo
 $\mathbf{x} = [x_1, x_2, \dots, x_n]$ son las variables del problema
 g_j e h_k son las restricciones
 $x_i^{(L)}, x_i^{(U)}$ son los límites superior e inferior para x_i

Ejemplo



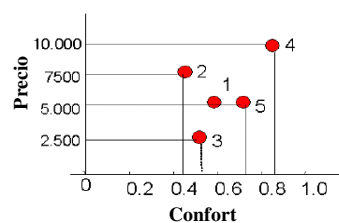
cuál es la mejor opción?

Ejemplo: continuación



- La opción 5 es mejor que la opción 1 e 2
- La opción 3 es la más barata (y mejor que 2)
- La opción 4 tiene el mejor valor para confort

Ejemplo: continuación



- Existen tres soluciones “óptimas” (3, 4, y 5)
- No se puede afirmar superioridad entre ellas

Técnicas basadas en Óptimos de Pareto

Tratan de encontrar en un sólo proceso genético varias soluciones de la Frontera de Pareto

En varias iteraciones realiza una mejora progresiva del conjunto No Dominado

Comparación de soluciones

- Para optimizar un problema es necesario poder comparar cual solución es mejor que otra.
- En MOOP se utiliza el concepto de dominancia de Pareto:
 - Una solución domina a otra si es mejor en al menos un objetivo e por lo menos igual en los demas.

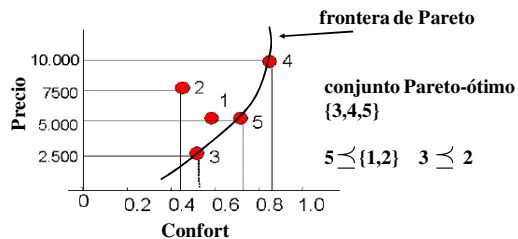
Dominancia de Pareto

- x domina a y , si
 - x no es superado por y en algun objetivo
 $f_j(y) \not\leq f_j(x), j = 1, \dots, M$
 - x es mejor al menos en un objetivo,
 $f_i(x) < f_i(y)$
- $x \preceq y$
 - x domina a y
 - x no es dominado por y
 - x es superior a y

Optimalidad

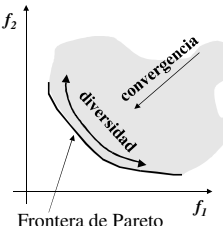
- El **conjunto Pareto-óximo** está formado por todas las soluciones no dominadas.
- La **frontera de Pareto** está formada por los valores de las funciones objetivos $f(x) = [f_1(x), f_2(x), \dots, f_m(x)]$ para cada x del conjunto Pareto-óximo.

Ejemplo: continuación



Metas en optimización multi-objetivo

- Encontrar el conjunto de soluciones mas cercano a la frontera de Pareto.
- Garantizar la mayor diversidad posible a lo largo de la frontera de Pareto



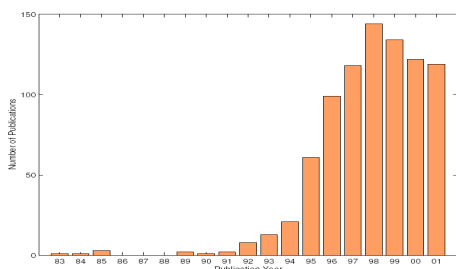
Técnicas para resolver MOOPs

- Técnicas Matemáticas
 - Pesos ponderados
 - Vector de restricciones ϵ
 - Programacion de metas
- Computación Evolutiva
 - Algoritmos Genéticos
 - Estrategias Evolutivas

Por qué Algoritmos Evolutivos?

- Permiten encontrar um conjunto de soluciones Pareto-óptimas en una sola ejecución del algoritmo.
- Son menos susceptibles a la naturaleza de la frontera de Pareto: concavidad, discontinuidades.

Publicaciones por año (Coello, 2001)



Algoritmos Genéticos

- Método de búsqueda/optimización
- Inspirado en la selección natural de Darwin
- Trabaja sobre una población de soluciones (paralelismo implícito)
- Evalua las soluciones usando la función objetivo o una función de aptitud
- Obtiene nuevas soluciones combinando/alterando soluciones de la poblacion anterior
- Es un proceso iterativo que converge a una solucion o satisface un criterio de parada

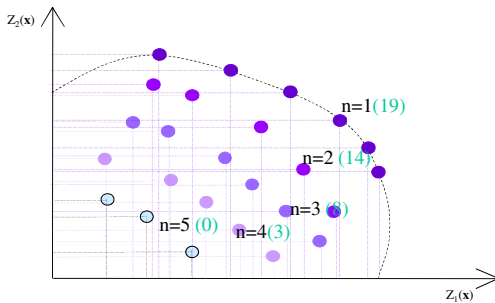
Algoritmos genéticos para optimización multi-objetivo (MOEA)

- Surgieron a partir de mediados de los años 80.
- Principales diferencias respecto a los Algoritmos Evolutivos tradicionales:
 - Usan el concepto de dominancia de Pareto
 - Hallan un conjunto de soluciones optimas

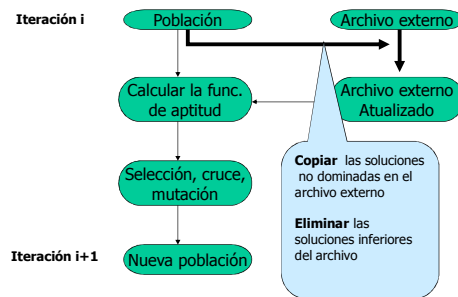
Métodos de Ordenamiento

En cada población, a cada individuo se le asigna un orden según su dominación sobre los otros individuos de la población:

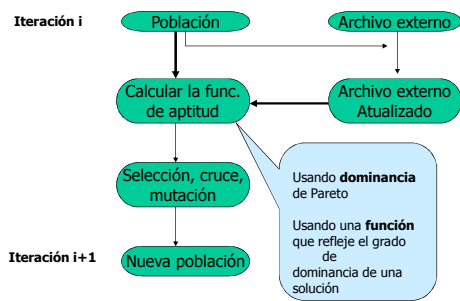
1. La aptitud de cada cadena está asociada a su orden:
 $f(x)$ depende de n_x Ó
2. La aptitud de cada cadena está asociada al número de individuos que domina $f(x)$ depende de $\#_x$



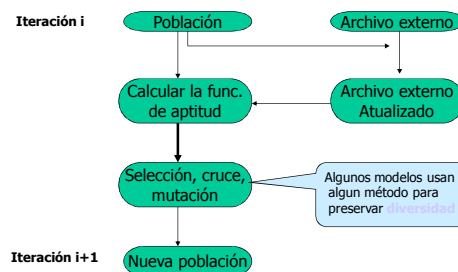
Funcionamiento básico MOEA



Funcionamiento básico MOEA



Funcionamiento básico MOEA



Minimización de la Distancia al Objetivo

- Se evalúa una solución, calculando la **distancia entre el vector de medidas y las especificaciones del usuario** para cada objetivo

$$F = \left(\sum_{i=1}^N |user_i - f_i|^p \right)^{1/p} \quad p \geq 1$$

- Presión por compromiso aumenta con p que hace crecer la penalidad a soluciones con poca performance en un objetivo:
 - $p=1$, distancia Manhattan
 - $p=2$, distancia Euclidiana
 - $p=\infty$, criterio MinMax: $F = \max(|user_i - f_i|) \quad 1 \leq i \leq n$

Minimización de Energía

- Método **adaptativo** para agregar objetivos
- Actualiza los pesos dinámicamente** durante el proceso evolutivo
- Establece automáticamente **más peso a objetivos menos satisfechos** por la población en total
- El **objetivo es minimizar todos los pesos** (medida de energía), que equivale a satisfacer los objetivos
- Incorpora especificaciones del usuario**

Minimización de Energía

- Cálculo de la aptitud:

$$F = \sum_{i=1}^n w_i Fnorm_i$$

donde n es el número de objetivos, w_i el peso asociado al objetivo i y $Fnorm_i$ el vector normalizado de aptitudes dado por la ecuación:

$$Fnorm_i = \frac{f_i}{\bar{f}_i}$$

donde el denominador representa el **promedio de las evaluaciones** alcanzadas por soluciones de la población con relación al objetivo i

Minimización de Energía

- Actualización de los pesos:
 - Basada en la ecuación de actualización de pesos del algoritmo *backpropagation* de redes neurales

$$\omega_{i,t+1} = k_1 \cdot \alpha \cdot \omega_{i,t} + k_2 \cdot (1 - \alpha) \cdot e_{i,t}$$

- k_1, k_2 : constantes de normalización
- α : constante análoga al término de momento usado en redes neuronales: de la característica de memoria, aumentando la estabilidad del sistema
- $e_{i,t}$: error porcentual entre la especificación de usuario para el objetivo i y la aptitud promedio de la población para el objetivo i en el instante t

Minimización de Energía

- Cálculo del error:
 - diferencia entre la especificación de usuario para el objetivo i y la evaluación media en el tiempo t

$$e_{i,t} = \left| \frac{user_i - \bar{f}_{i,t}}{\bar{f}_i} \right|$$

- Cálculo de la energía E del sistema:
 - Disminuye a medida que la evaluación se aproxima de la especificación

$$E = \sum_{i=1}^n w_i^2$$

Cálculo de las Constantes

- El objetivo de las constantes k_1 e k_2 es ayudar a que la **suma de los valores de los pesos $S_{w,t}$** , para un determinado instante t , dé una medida del estado de convergencia del sistema (E) con relación a las especificaciones del usuario.
- E es la energía del sistema:

$$E = \sum_{i=1}^n w_i^2$$

- la **minimización de la energía** equivale al proceso de satisfacción de múltiples objetivos. Por eso el nombre de *Metodología de Minimización de Energía*.

Minimización de Energía

- Ventajas
 - Implementación simple, si la comparamos con los métodos basados en estrategias de Pareto
 - Orientación de la evolución se adapta a los problemas encontrados durante el proceso
 - Buenos resultados en la síntesis de circuitos electrónicos usando AG

Minimização de Energia

- Desventajas
 - Exige que las evaluaciones estén normalizadas para llegar a valores comparables entre si
 - Puede generar *speciation*, que significa el surgimiento de diferentes grupos (especies) especializados en satisfacer cada objetivo
 - Requiere la especificación de un valor *target* para cada objetivo *i* (*user specification*)
 - El gráfico de performance muestra oscilaciones: el mejor individuo en la generación *t* puede no ser el mejor en la generación *t+1* por cambios de los valores en w_i

Tendencias actuales en Investigación

- 1) Desarrollo de técnicas eficientes para la reducción del chequeo de no dominancia
- 2) Uso del posicionamiento geográfico como mecanismo para mantener diversidad en la frontera de Pareto.
- 3) Desarrollo de formas de elitismo para obtener mejor calidad de soluciones y menor costo computacional.
- 4) Manejo de las restricciones
- 5) Métricas para comparar técnicas
- 6) Consolidación teórica
- 7) Desarrollo de Funciones de aptitud MO
- 8) Resolución de Problemas bajo incertidumbre
- 9) Mezclas con otros métodos clásicos y metaheurísticas