

BIOINFORMÁTICA

2013 - 2014

PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- **Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo**
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados

BIOINFORMÁTICA

TEMA 10. ALGORITMOS EVOLUTIVOS PARA PROBLEMAS MULTIOBJETIVO

- 1. PROBLEMAS MULTIOBJETIVO**
- 2. EVOLUCIÓN EN PROBLEMAS MULTIOBJETIVO**
- 3. ELITISMO EN LA BÚSQUEDA EVOLUTIVA MULTIOBJETIVO**
- 4. NSGA-II (ELITISMO DENTRO DE LA POBLACIÓN)**
- 5. MÉTRICAS DE COMPARACIÓN**

K. Deb, Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, 2001.

C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Pub., 2002. (Second Edition, 2007).

1. Problemas Multiobjetivo

- Muchos problemas reales se caracterizan por la existencia de **múltiples medidas de actuación**, las cuales deberían ser optimizadas, o al menos ser satisfechas simultáneamente.

Ejemplo: Diseño de un sistema de control de aire acondicionado, optimizando el conjunto de parámetros de un sistema de control:

- Minimizar el consumo de energía
- Maximizar el confort de los usuarios
- Maximizar la estabilidad del sistema de control,

Concepto de Dominancia

Un **Problema Multiobjetivo** consiste en:

$$\text{Max o Min } z = f(x) = (f_1(x), f_2(x), \dots, f_K(x))$$

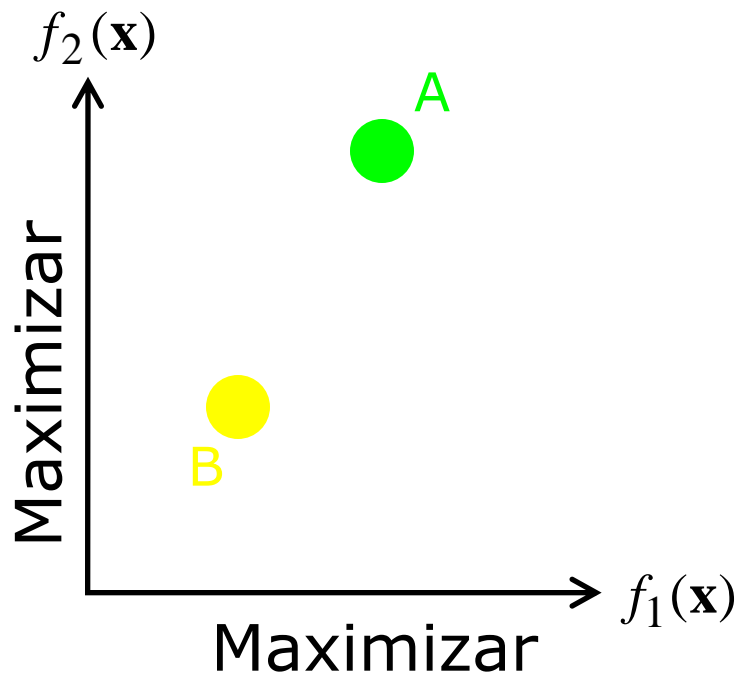
Soluciones pareto-optimales o no-dominadas: Se dice que un vector **a** domina a otro **b** (se nota como $\mathbf{a} \prec = \mathbf{b}$) si, y sólo si (suponiendo maximización):

$$\forall i \in \{1, 2, \dots, K\} \mid f_i(\mathbf{a}) \geq f_i(\mathbf{b}) \wedge \exists j \in \{1, 2, \dots, K\} \mid f_j(\mathbf{a}) > f_j(\mathbf{b})$$

Es decir, una solución domina a otra si es mejor o igual en todos los objetivos y al menos mejor en uno de ellos

Concepto de Dominancia

Maximizar $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$



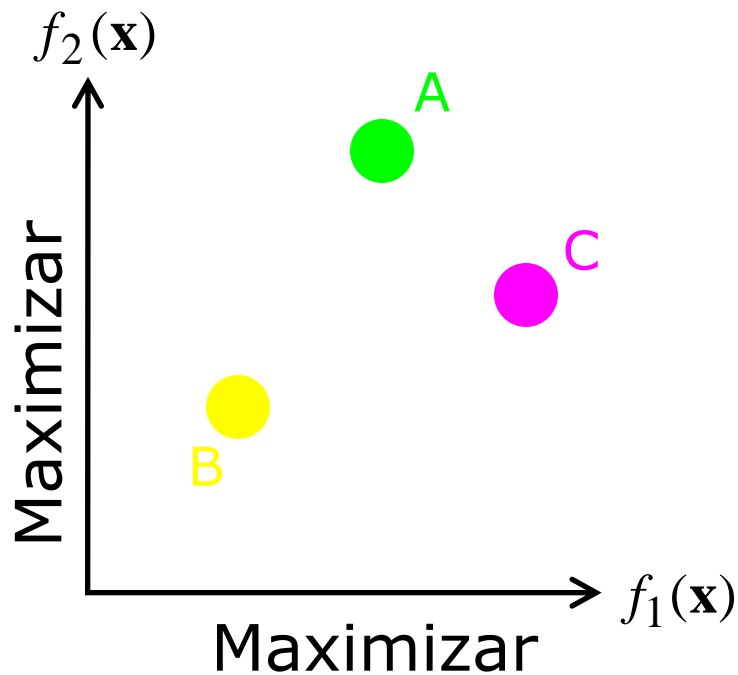
A domina a B

B es dominada por A

(A es mejor que B)

Concepto de Dominancia

Maximizar $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$



A y C son no dominadas entre sí
(ninguna domina a la otra)

Las dos dominan a B

Concepto de Dominancia

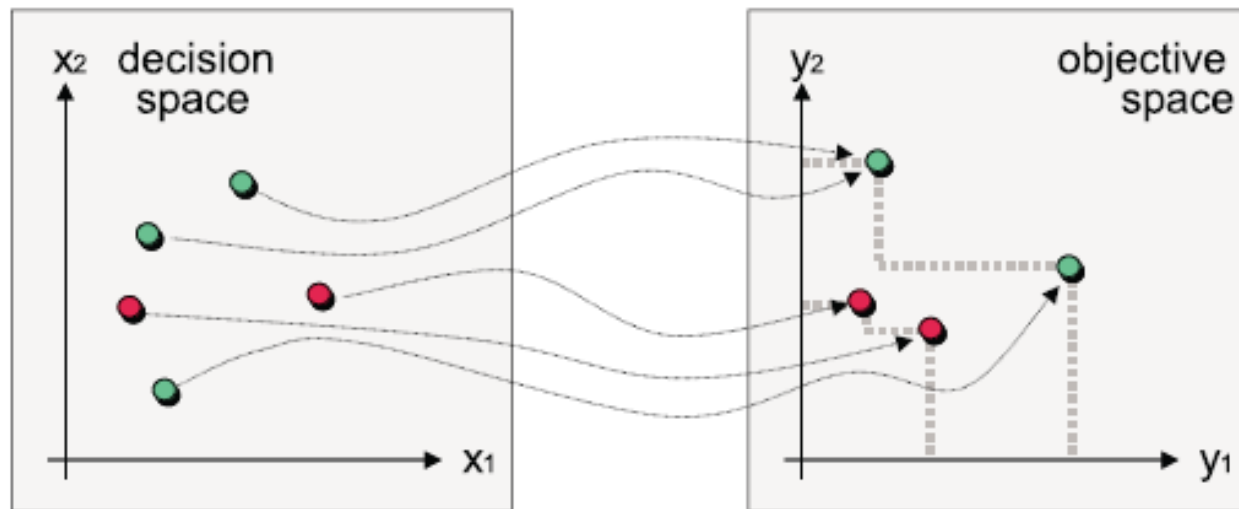
Una solución es **Pareto-optimal** si no es dominada por ninguna otra solución del espacio

El conjunto de todas las soluciones no dominadas $X^* \subset X$ es el **conjunto Pareto-optimal** y compone la **solución óptima del problema multiobjetivo**

Los vectores de valores de las funciones objetivo de los elementos del conjunto Pareto-optimal $z(X^*) \subset Y$ forman la **frontera (o el frente) de Pareto**

Ejemplo: Conjuntos de Pareto en \mathbb{R}^2

Pareto set ● Pareto front
Pareto set approximation ● Pareto front approximation



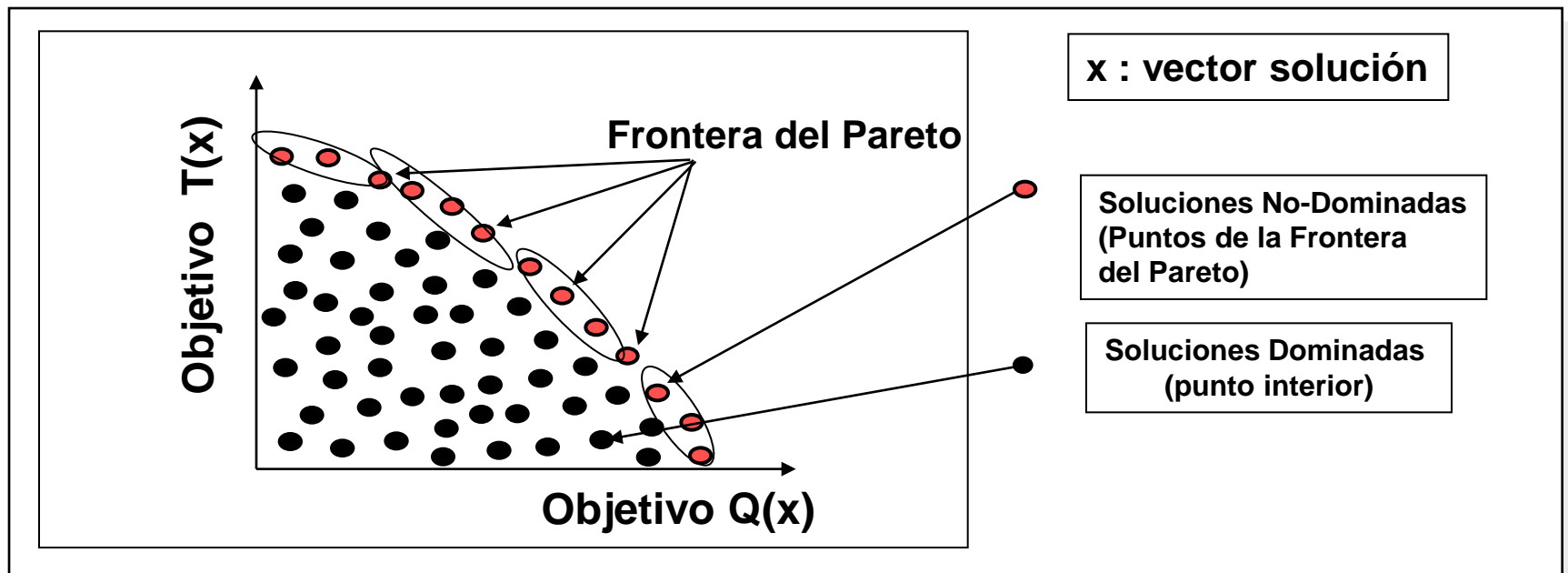
$$(x_1, x_2, \dots, x_n) \xrightarrow{\quad} \mathbf{f} \xrightarrow{\quad} (y_1, y_2, \dots, y_k)$$

search

evaluation

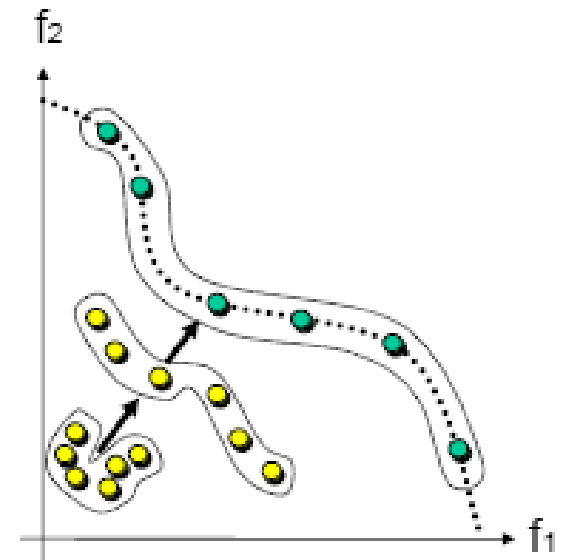
Problemas Multiobjetivo (2)

- No suele existir una única solución optimal, existe un conjunto (a veces infinito) de soluciones No-Dominadas que forma la Frontera del Pareto
 - **Ejemplo:**
Identificar la frontera del Pareto para $[Max Q(x), Max T(x)]$



Objetivo de la Optimización Multiobjetivo

- El objetivo es encontrar una **aproximación del frente de Pareto** de la mayor calidad posible
 - Debe estar tan cerca del frente de Pareto óptimo como sea posible
 - Las soluciones deben estar uniformemente distribuidas sobre el frente
 - La aproximación debe capturar todo el frente del Pareto, incluyendo los extremos

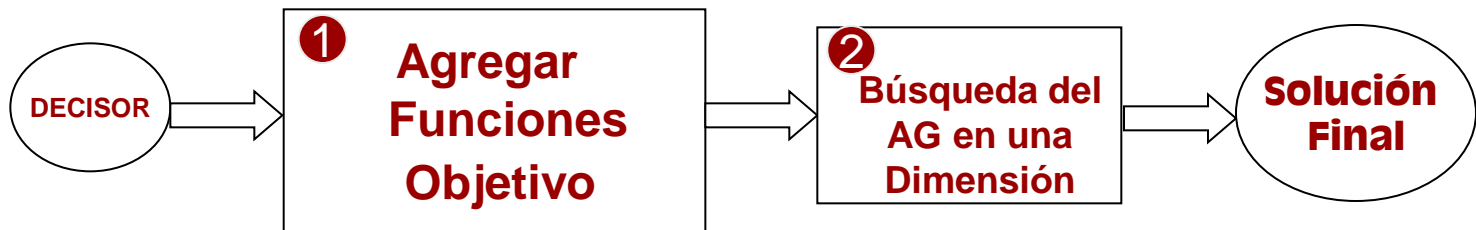


Problemas Multiobjetivo (3)

¿Qué necesitamos para resolver este problema?:

- *Un método de búsqueda basado en los múltiples objetivos.*
- *Una política de equilibrio entre los objetivos.*
- *Un orden para este proceso de optimización.*

Vamos a considerar 2 posibilidades: a) Agregación + búsqueda



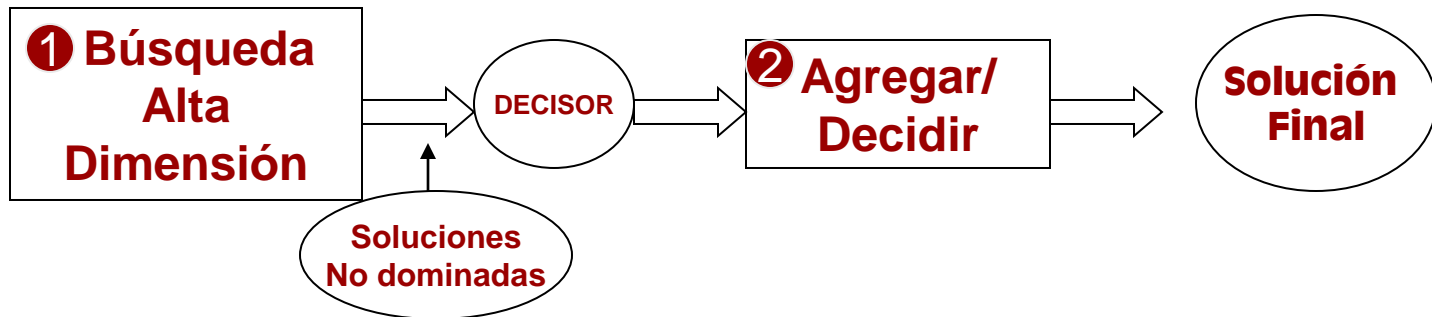
Primero se agregan los objetivos dando lugar a una función de fitness para el AG que se aplica a continuación.

Problemas Multiobjetivo (4)

¿Qué necesitamos para resolver este problema?:

- *Un método de búsqueda basado en los múltiples objetivos,*
- *Una política de equilibrio entre los objetivos,*
- *Un orden para este proceso de optimización.*

Vamos a considerar 2 posibilidades: b) Búsqueda + agregar/decidir



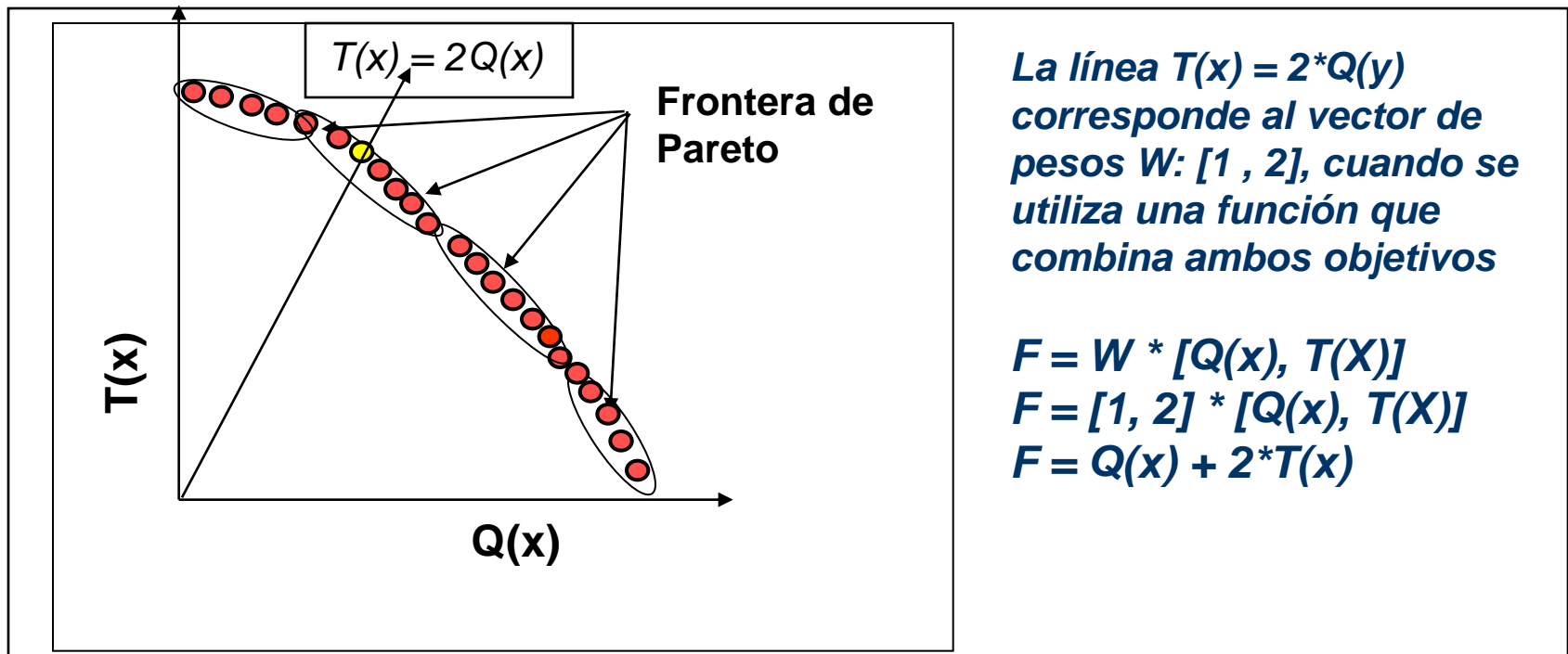
Nota: Se puede considerar una tercera posibilidad híbrida, combinando búsqueda en alta dimensión con búsquedas en dimensiones menores vía agregación parcial de objetivos, como modelos interactivos.

2. Evolución en Problemas Multiobjetivo

- Se evoluciona una población de soluciones al problema.
- Se aplican mecanismos que mantengan diversidad en la población para conseguir un conjunto de soluciones no dominadas lo más grande posible.
- Dos tipos de modelos de acuerdo a las tipologías a) y b):
 - **Modelos evolutivos utilizando pesos para la agregación de los objetivos.**
 - **Modelos evolutivos que generan poblaciones de soluciones no dominadas.**

Modelos Evolutivos utilizando Pesos

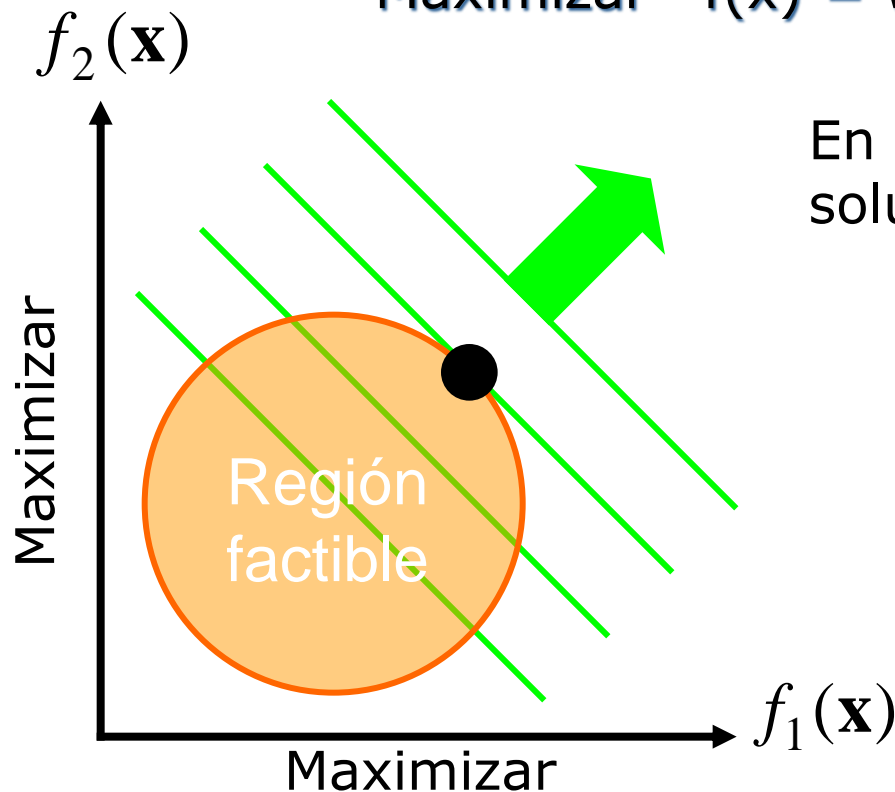
- La agregación de los objetivos conduce a la obtención de un único punto de equilibrio en la frontera.
- Ejemplo: $[Max Q(x), Max T(x)]$
*Dar a $T(x)$ dos veces la importancia de $Q(x)$, ej: $T(x) = 2*Q(x)$*



Modelos Evolutivos utilizando Pesos (2)

- Presentan los problemas habituales de un optimizador MO basado en agregación de los objetivos usando pesos:

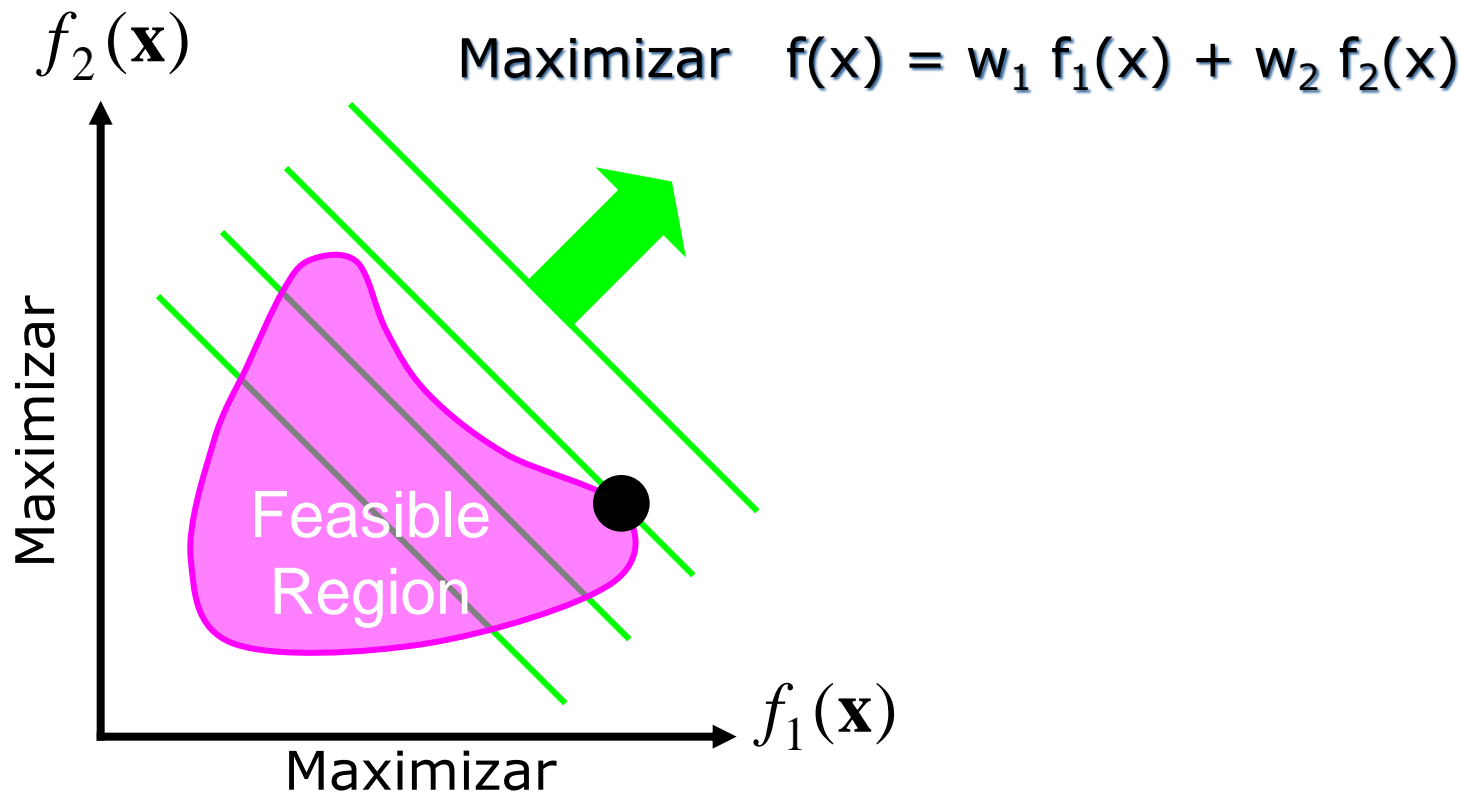
$$\text{Maximizar } f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x})$$



En principio, se obtiene una única solución en cada ejecución

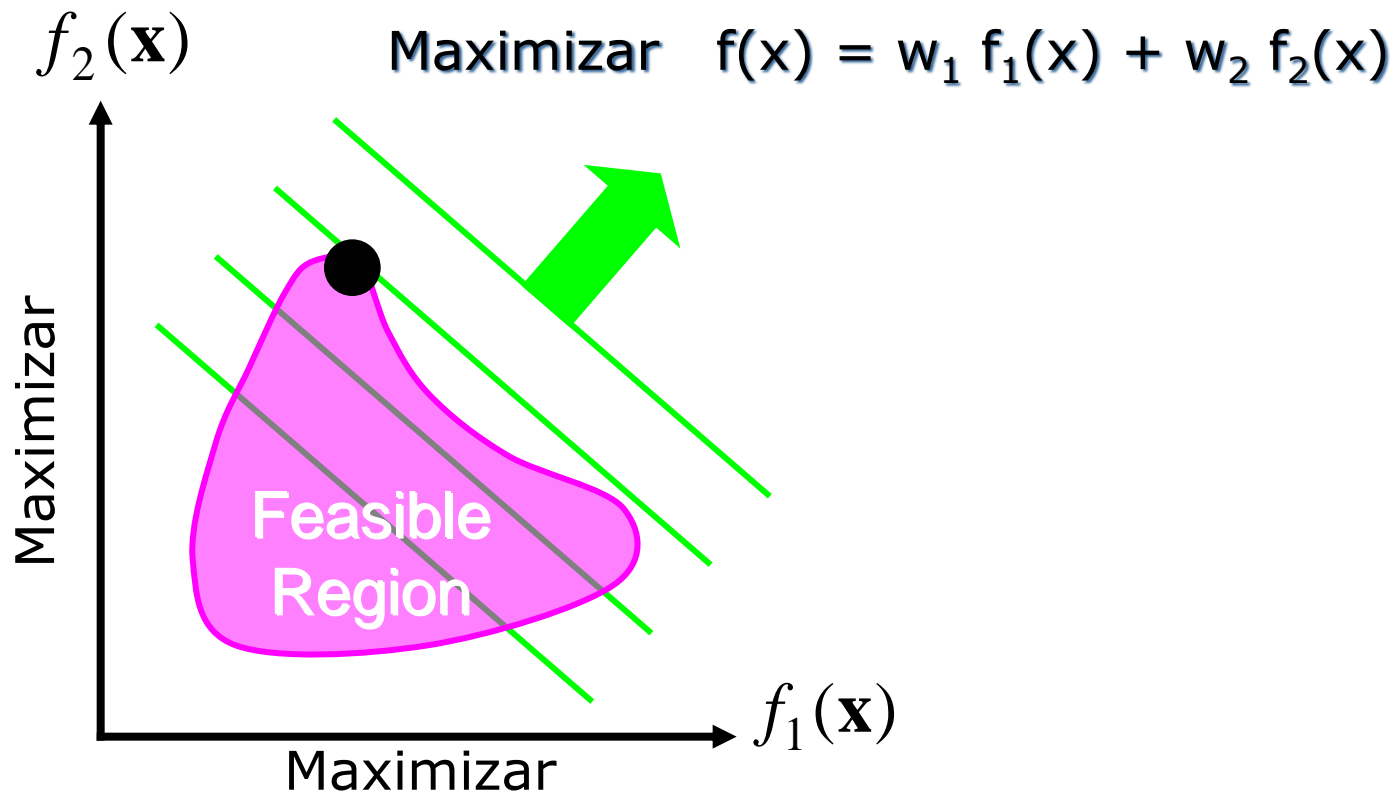
Modelos Evolutivos utilizando Pesos (3)

- El enfoque es muy sensible a la especificación de los pesos
- No puede encontrar soluciones en regiones cóncavas del frente de Pareto



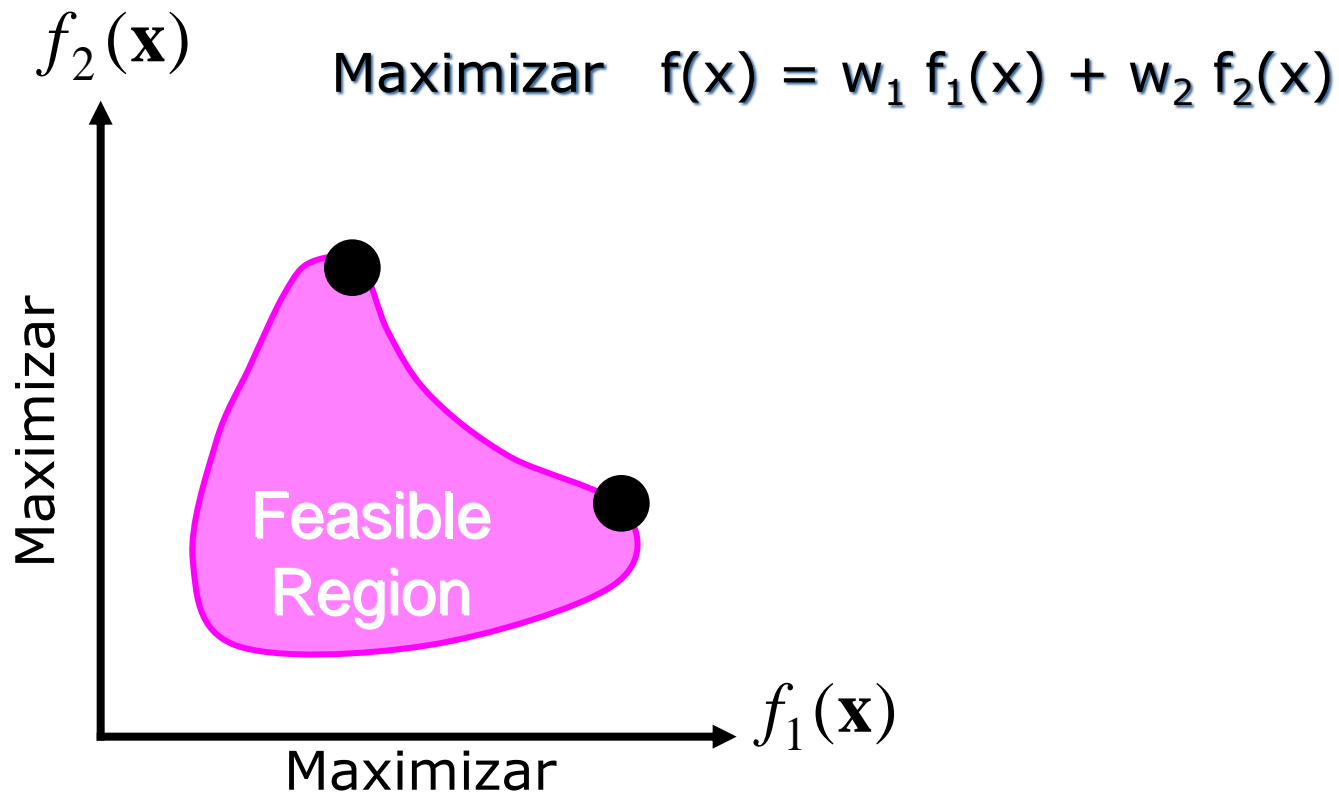
Modelos Evolutivos utilizando Pesos (4)

- El enfoque es muy sensible a la especificación de los pesos
- No puede encontrar soluciones en regiones cóncavas del frente de Pareto



Modelos Evolutivos utilizando Pesos (5)

- El enfoque es muy sensible a la especificación de los pesos
- No puede encontrar soluciones en regiones cóncavas del frente de Pareto



Modelos Evolutivos utilizando Pesos (6)

- VOW-GA: Variable Objective Weighting GA
(Hajela & Lin 1992)

- RW-GA: Random Weights GA
(Ishibuchi & Murata, 1998)

Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas

Primera generación de Algoritmos Evolutivos Multiobjetivo basados en soluciones no-dominadas.

- **MOGA: Multi-objective Optimization GA**

C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. S. Forrest (Ed.), Proc. 5th Int. Conf. on Genetic Algorithms, Morgan Kaufmann, 1993, 416-423.

- **NPGA: Niche Pareto GA**

J. Horn, N. Nafpliotis. Multiobjective Optimization Using the Niche Pareto Genetic Algorithms. IlliGAL Report 93005, University of Illinois, Urbana, Champaign, July 1993.

- **NSGA: Non-dominated Sorting GA**

N. Srinivas, K. Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation 2 (1995) 221-248.

Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas (2)

- **MOGA: Multi-objective Optimization GA**
(*Fonseca & Fleming 1993*)

C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. S. Forrest (Ed.), Proc. 5th Int. Conf. on Genetic Algorithms, Morgan Kaufmann, 1993, 416-423.

A cada individuo de la población se le asignará un rango de acuerdo al cual será ordenado para la selección.

El rango se asigna según un criterio de no dominancia.

Si x_i ***es no dominado entonces*** $\text{rango}(x_i) = 1$

En otro caso $\text{rango}(x_i) = 1 + (\text{no. de individuos que lo dominan})$.

Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas (3)

- **MOGA: Multi-objective Optimization GA**
(*Fonseca & Fleming 1993*)

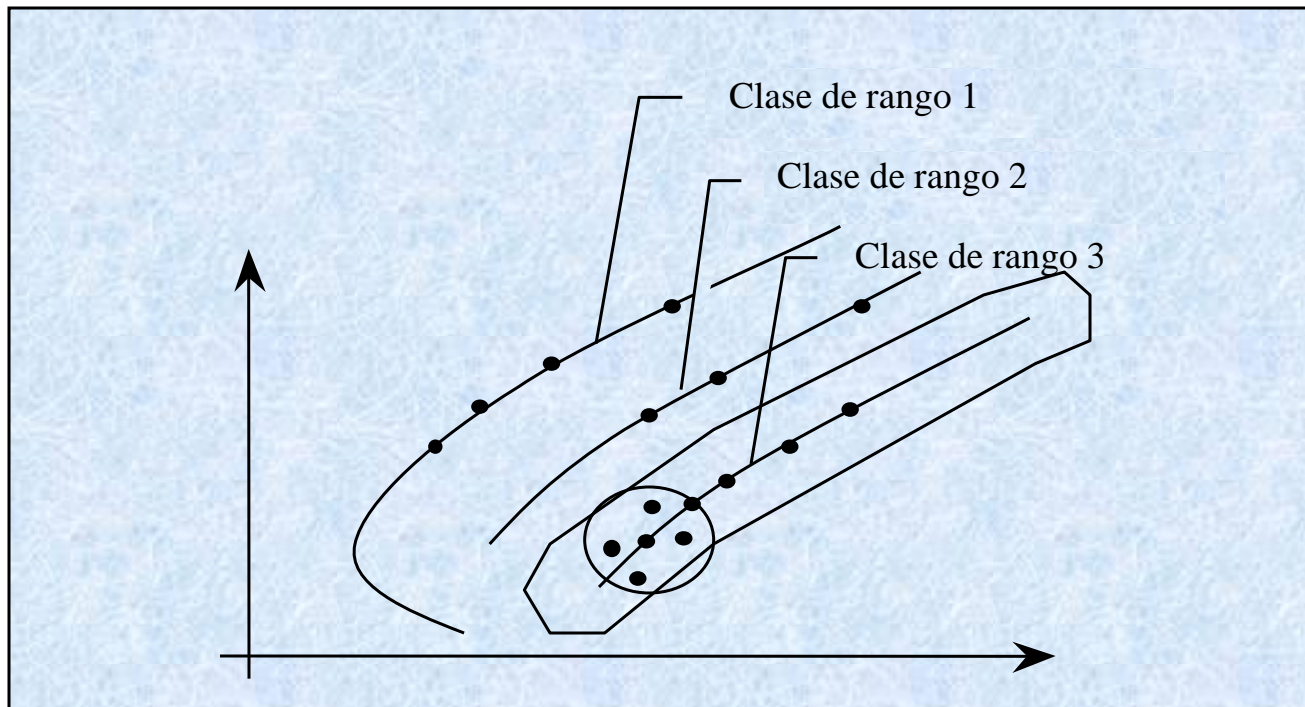
Una vez calculado el rango de los individuos de la población se siguen los siguientes pasos:

1. La población se ordena de menor a mayor de acuerdo al rango que se le ha asignado a cada individuo.
2. Se asigna el valor de adaptación para cada individuo por interpolación desde el mejor (rango 1) hasta el peor.
3. Se promedia la adaptación de los individuos con el mismo rango, para que tengan el mismo valor de adaptación.

Nuevas versiones utilizan técnicas de proporción de nichos (sharing) sobre los objetivos.

Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas (4)

- MOGA: Multi-objective Optimization GA
(Fonseca & Fleming 1993)



Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas (5)

■ NPGA: Niche Pareto GA (*Horn & Nafpliotis, 1993*)

J. Horn, N. Nafpliotis. Multiobjective Optimization Using the Niche Pareto Genetic Algorithms. IlliGAL Report 93005, University of Illinois, Urbana, Champaign, July 1993.

Este algoritmo se basa en:

la combinación de la selección de torneo y el concepto de Pareto Dominancia.

Dados dos individuos que compiten, se selecciona un subconjunto aleatorio de la población de tamaño t_{dom} .

Si un individuo es dominado por cualquier miembro del conjunto y el otro no, entonces este último se considera el ganador del torneo. Si ambos individuos son dominados, el resultado del torneo se decide por el método de proporción. El individuo con menos cromosomas en su nicho es el seleccionado. La técnica de sharing se aplica a nivel de objetivos.

Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas (6)

- NSGA: Non-dominated Sorting GA (*Srinivas & Deb, 1995*)

N. Srinivas, K. Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation 2 (1995) 221-248.

Este algoritmo se basa en:

- Una ordenación de la población según el criterio de no-dominancia (distinto de MOGA), mediante fronteras de no-dominancia.
- El uso de una técnica de proporción de nichos (sharing) aplicada sobre los valores de las variables de decisión de cada individuo. (método de proporción continua y selección por torneo en implementaciones recientes).

Modelos Evolutivos que Generan Poblaciones de Soluciones No-dominadas (7)

■ NSGA: Non-dominated Sorting GA (*Srinivas & Deb, 1995*)

El algoritmo actúa extrayendo frentes de individuos progresivamente, a los que se asigna un valor de adaptación menor que el del frente anterior.

- 1. En el primer frente se toman los individuos no dominados, a los que se asigna un valor hipotético alto como valor de adaptación.**
- 2. Estos individuos se penalizan según un criterio de proporción en el fenotipo (sharing en las variables) con $\alpha=2$.**
- 3. Los individuos anteriores son ignorados temporalmente para procesar el resto de la población de igual forma, identificando el segundo conjunto de individuos no dominados (entre los dominados).**

A este segundo conjunto de individuos se le asigna un valor hipotético más pequeño que el mínimo valor alcanzado por el conjunto anterior tras la aplicación del método de proporción.

- 4. El proceso continua hasta que toda la población se clasifica en frentes.**

3. Elitismo en la Búsqueda Evolutiva Multiobjetivo

Elitismo en la Búsqueda Evolutiva Multiobjetivo. Conjunto Elite

SPEA y SPEA2

Elitismo Dentro de la Población: NSGA II

Elitismo en la Búsqueda Evolutiva Multiobjetivo. Conjunto Elite

Recientemente se ha diseñado un nuevo modelo evolutivo que usa una población externa, donde se almacenan soluciones no-dominadas encontradas a lo largo de la búsqueda.

Esto permite al algoritmo cubrir de un modo más adecuado el Frente del Pareto.

Este conjunto de soluciones no dominadas se suele llamar "conjunto elite", P^e , con tamaño N^e . Su uso lleva asociadas dos cuestiones:

Población ➡ **Conjunto elite.**

¿Qué soluciones de P se mantienen en P^e ?.

Conjunto elite ➡ **Población.**

¿Cómo y cuando los elementos de P^e se reinsertan en P ?.

Ejemplo: **Modelo elitista SPEA.**

Elitismo en la Búsqueda Evolutiva Multiobjetivo. Conjunto Elite (2)

Modelo Genérico de Algoritmo Evolutivo Multiobjetivo Elitista

t := 0

(A⁰, B⁰, p⁰) := inicializar

MIENTRAS ^eterminar(A^t, B^t, t) = falso **HACER**

t := t + 1

A^t := truncar(actualizar(A^{t-1}, B^{t-1}))

p^t := adaptar(A^t, B^{t-1}, p^{t-1})

B^t := operadores(selección^e(evaluación^e(A^t, B^{t-1}, p^t)))

Fin MIENTRAS

A^t : población élite

B^t : población

p^t := intensidad del elitismo en la generación t.

SPEA: Modelo Evolutivo Elitista que Genera Poblaciones de Soluciones No-dominadas

■ STRENGTH PARETO EVOLUTIONARY ALGORITHMS (SPEA) (Zitzler, Thiele, 1998)

Zitzler, E., Thiele, L. (1998a) An evolutionary algorithm for multiobjective optimization: The strength Pareto Approach. Technical Report 43, Zürich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH).

E. Zitzler, L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3:4 (1999) 257-217.

Zitzler, E., Deb, K., Thiele, L. (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation Journal 8(2), 125-148.

Este algoritmo introduce el concepto de CONJUNTO ELITISTA EXTERNO DE SOLUCIONES NO DOMINADAS, actualmente muy importante en el desarrollo de AGs Multiobjetivo.

SPEA (2)

Algoritmo SPEA

- Paso 1.** Generar la población inicial P y el conjunto P^e vacío.
- Paso 2.** Copiar las soluciones no dominadas de P en P^e .
- Paso 3.** Quitar en P^e aquellas soluciones dominadas por otras.
- Paso 4.** Si $|P^e| > N^e$, entonces reducir el conjunto a tamaño N^e mediante técnicas de clustering.
- Paso 5.** Calcular el fitness de los individuos de $P' = P + P^e$.
- Paso 6.** Seleccionar N individuos a partir de P' (mediante torneo binario).
- Paso 7.** Aplicar cruce y mutación.
- Paso 8.** Volver al Paso 2 si no se ha alcanzado el máximo número de iteraciones.

SPEA (3)

- Se introduce elitismo manteniendo una población externa P^e con tamaño N^e que guarda las soluciones no dominadas encontradas.
- En cada generación, se va actualizando la población externa P^e copiando todos los individuos no-dominados de P a P^e y eliminando los duplicados o dominados en P^e .
- Si $|P^e| > N^e$, se reduce el conjunto a tamaño N^e mediante técnicas de clustering, quedándose con el cromosoma del cluster que tenga mínima distancia al resto (el más cercano al centro).
- En la iteración $t+1$ se utilizan las soluciones externas para el proceso de reproducción. El fitness utilizado es el siguiente (Fitness(i) = Strength(i)):

Strength de la solución i en P^e : $S_i = n_i / (N + 1)$

n_i es el número de soluciones dominadas por i en P

El fitness de i es $f_i = s_i$

SPEA (4)

- Fitness del cromosoma j en P : $f_j = 1 + \sum_{i \in P_e \text{ e } i \text{ domina } j} S_i$
El Fitness se mide en términos de minimización.
- Crear $P' = P + P^e$ con tamaño $N' = N + N^e$.
- Seleccionar N individuos a partir de los N' en P' , utilizando la selección por torneo binario, y aplicar el cruce y mutación para crear la siguiente población P'' .
- CONTINUAR CON EL PROCESO EVOLUTIVO ITERATIVO DE IGUAL FORMA HASTA QUE SE CUMPLA LA CONDICIÓN DE PARADA.

SPEA (5)

Utiliza técnicas de clustering para reducir el número de soluciones no dominadas almacenadas para no destruir las características de equilibrio de la frontera (clustering sobre el fitness).

Todas las soluciones externas participan en la selección.

SPEA (6)

ALGORITMO DE CLUSTERING UTILIZADO EN P^e

Paso 1. Inicialmente, cada elemento pertenece a un cluster. Si $|P^e| \leq N^e$ entonces ir al Paso 5.

Paso 2. Para cada par de clusters, calcular la distancia entre clusters como la distancia media entre todos sus elementos

$$D_{ij} = 1/|C_i| \cdot |C_j| \sum_{i \in C_i, j \in C_j} d(i,j).$$

Paso 3. Combinar los dos clusters cuya distancia entre sí sea mínima.

Paso 4. Si (número de clusters) $\leq N^e$ entonces ir a 5, si no ir a 2.

Paso 5. Elegir una solución de cada cluster, la de mínima distancia media al resto de elementos del cluster.

SPEA (7)

ALGUNOS COMENTARIOS

- El algoritmo de clustering se puede utilizar en el espacio de decisiones (cromosoma) o en el espacio de objetivos (los autores de SPEA sugieren este último).
- Ratio entre poblaciones 1:4 (ejemplo: $|P^e| = 20$, $|P| = 80$)
- SPEA presenta una desventaja frente a otros AGs-MO en términos de eficiencia por el tiempo necesario para la aplicación del algoritmo de clustering en el conjunto elite.
- La diferencia básica entre los distintos algoritmos multiobjetivo (MOGA, NSGA, SPEA, ...) es la modificación en la asignación de fitness, más el elitismo.

SPEA (8)

- **Experimentos de comparación entre NSGA y SPEA dan mejores resultados para SPEA.**
- **Los resultados de NSGA+Elitismo son similares a SPEA.**

Referencia: E. Zitzler, K. Deb, L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8:2 (2000) 173-195.

SPEA2

Una versión revisada de SPEA (llamada SPEA2) fue propuesta por E. Zitzler y sus colegas en 2001.

SPEA2 tiene 3 diferencias fundamentales con respecto al SPEA original:

- 1. Incorpora una estrategia de grano fino para asignar aptitud, la cual toma en cuenta la cantidad de individuos que cada solución domina y la cantidad de individuos que la dominan.**
- 2. Incorpora una técnica de estimación de densidad de vecinos que guía la búsqueda de manera más eficiente.**
- 3. Tiene un esquema de truncamiento de archivo que garantiza la preservación de soluciones de frontera.**

SPEA2

Eckart Zitzler, Marco Laumanns, Lothar Thiele: **SPEA2: Improving the Strength Pareto Evolutionary Algorithm.**

Zürich, TIK Report Nr. 103, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, May, 2001.



Eckart Zitzler

<http://www.tik.ee.ethz.ch/~zitzler/>

Source code

<http://www.tik.ee.ethz.ch/%7ezitzler/testdata.html#source>

PISA

A Platform and Programming Language Independent Interface for Search Algorithms

<http://www.tik.ee.ethz.ch/pisa/>

Clasificación de MOEAS Elitistas

Características que permiten clasificar a los MOEA Elitistas:

Utilizan una población secundaria “elite”: A^t .

Estrategia de elitismo: Cómo se actualiza la población elitista, A^t .

Estrategia de evaluación: Cómo afectan los individuos del conjunto elite a la asignación de fitness en la población y viceversa.

Estrategia de reinserción: Cómo toman parte los individuos elite en el proceso de reproducción de descendientes.

Los diferentes MOEA elitistas difieren, entre otras cosas, en la aplicación de estas tres estrategias

4. NSGA-II: Elitismo dentro de la Población

Considerado el Mejor Modelo

NSGA II: *K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6:2 (2002) 182-197.*

Nondominated Sorting Genetic Algorithm II (NSGA II) fue propuesto por K. Deb y sus estudiantes en 2000.

Es una versión mejorada del NSGA que utiliza un operador de *crowding* que no requiere parámetros en vez de usar nichos.

NSGA II utiliza un esquema de selección más en el cual la población de padres se compara con la población de hijos.

NSGA-II además de contar con el uso de elitismo es mucho más eficiente (computacionalmente) que NSGA y es un algoritmo altamente competitivo en convergencia al pareto.

NSGA-II (2)

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $ P_{t+1} + \mathcal{F}_i \leq N$	until the parent population is filled
$\text{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding-distance in \mathcal{F}_i
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include i th nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
Sort(\mathcal{F}_i, \prec_n)	sort in descending order using \prec_n
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - P_{t+1})]$	choose the first $(N - P_{t+1})$ elements of \mathcal{F}_i
$Q_{t+1} = \text{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

- En cada generación, se crea un conjunto mediante la unión de la actual población y la creada mediante selección, cruce y mutación
- De este conjunto se extraen los diferentes frentes (agrupados según el número de soluciones que los dominan). El frente F_1 coincide con el actual frente Pareto
- La nueva población se crea incluyendo los frentes (de mejor a peor) hasta alcanzar el tamaño máximo. Si es necesario, el último frente se trunca atendiendo al orden basado en el *crowding* (medida de concentración)

NSGA-II (3)

fast-non-dominated-sort(P)

for each $p \in P$

$S_p = \emptyset$

$n_p = 0$

for each $q \in P$

if ($p \prec q$) then

$S_p = S_p \cup \{q\}$

else if ($q \prec p$) then

$n_p = n_p + 1$

if $n_p = 0$ then

$p_{\text{rank}} = 1$

$\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$

$i = 1$

while $\mathcal{F}_i \neq \emptyset$

$Q = \emptyset$

for each $p \in \mathcal{F}_i$

for each $q \in S_p$

$n_q = n_q - 1$

if $n_q = 0$ then

$q_{\text{rank}} = i + 1$

$Q = Q \cup \{q\}$

$i = i + 1$

$\mathcal{F}_i = Q$

If p dominates q

Add q to the set of solutions dominated by p

Increment the domination counter of p
 p belongs to the first front

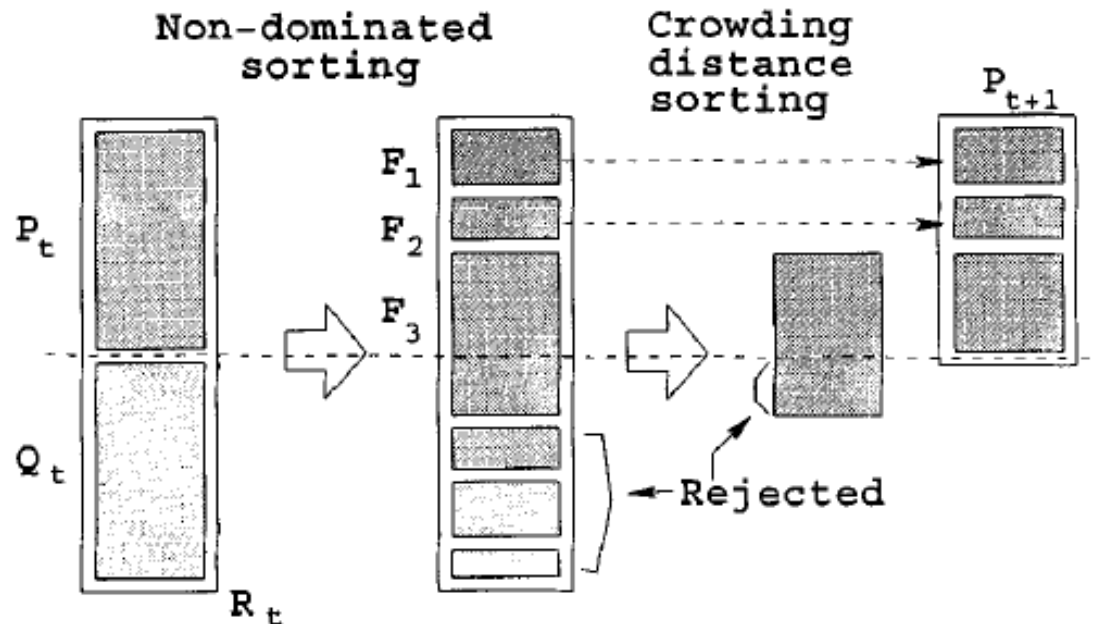
Initialize the front counter

Used to store the members of the next front

q belongs to the next front

NSGA-II (4)

- La medida de *crowding* se utiliza para seleccionar las soluciones más dispersas entre los individuos del último frente utilizado en la nueva población (F_3 en este ejemplo)
- Cuanto mayor sea la distancia de *crowding* de una solución al resto de su frente mejor, ya que hay menos concentración en esa zona



NSGA-II (5)

crowding-distance-assignment(\mathcal{I})

$l = |\mathcal{I}|$

for each i , set $\mathcal{I}[i]_{\text{distance}} = 0$

for each objective m

$\mathcal{I} = \text{sort}(\mathcal{I}, m)$

$\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$

for $i = 2$ to $(l - 1)$

$\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i + 1].m - \mathcal{I}[i - 1].m) / (f_m^{\max} - f_m^{\min})$

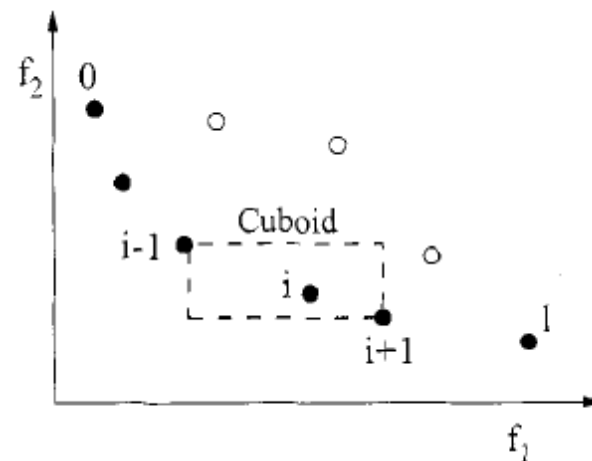
number of solutions in \mathcal{I}

initialize distance

sort using each objective value

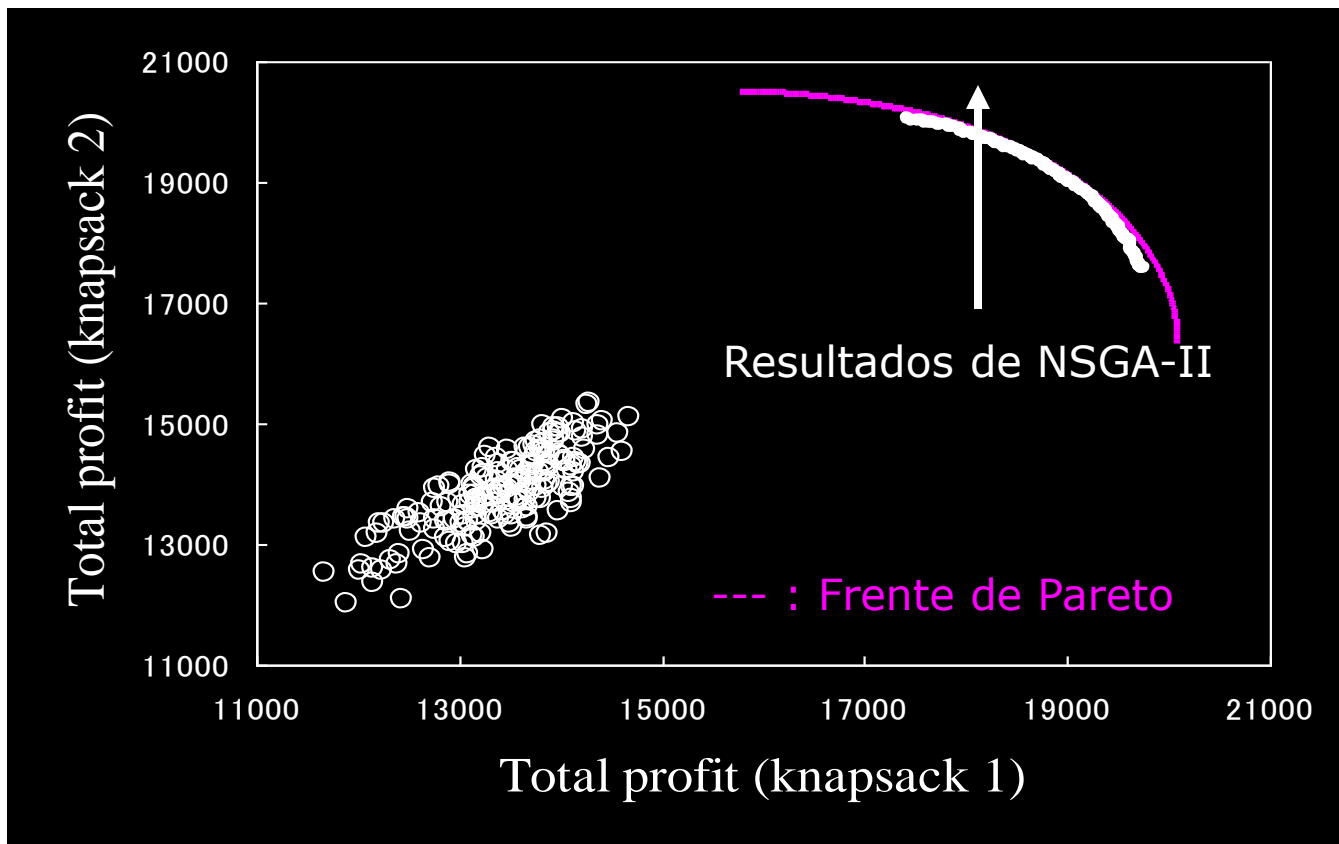
so that boundary points are always selected

for all other points



NSGA-II (6)

Problema de la Mochila bi-objetivo con 500 objetos (Maximización)



200 soluciones aleatorias y el frente de Pareto óptimo

NSGA-II (7)

Problemas:

Parece tener un comportamiento más pobre cuando se utiliza con representación binaria.

Tiende a tener problemas exploratorios conforme se incrementa el número de funciones objetivo (lo tienen todos los algoritmos actuales)

NSGA-II (6)



Kalyanmoy Deb

<http://www.iitk.ac.in/kangal/>

The IEEE TEC paper describing NSGA-II for multi-objective optimization is judged as the **FAST-BREAKING PAPER IN ENGINEERING** by Web of Science (ESI) in February 2004



Prof. Deb receives Shanti Swarup Bhatnagar Prize from Honorable Prime Minister of India on 28 September 2005 in New Delhi.

Softwares Developed at KanGAL

<http://www.iitk.ac.in/kangal/codes.shtml>

- Multi-objective NSGA-II code in C
 - Original Implementation (for Windows and Linux): [NSGA-II in C \(Real + Binary + Constraint Handling\)](#)
 - New (10 April 2005) (for Linux only): [NSGA-II in C \(Real + Binary + Constraint Handling\)](#)
 - Revision 1.1 (10 May 2005) (for Linux only): [NSGA-II in C \(Real + Binary + Constraint Handling\)](#)
 - Revision 1.1 (10 June 2005) (for Linux only): [NSGA-II in C with gnuplot \(Real + Binary + Constraint Handling\)](#)

NSGA-II (7)



Prof. Deb receives Shanti Swarup Bhatnagar Prize from Honorable Prime Minister of India on 28 September 2005 in New Delhi.

MOEA/D

New model:

CEC 2009 competition on multiobjective optimization: First rank was taken by a DE-based algorithm MOEA/D for unconstrained problems.

MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition.
IEEE transactions on Evolutionary Computation, 2007, 11:6, 712 – 731.

A simple version of MOEA/D is introduced in this paper. It won the IEEE TEC Outstanding Paper Award.

H. Li and Q. Zhang, Multiobjective Optimization Problems with
Complicated Pareto Sets, MOEA/D and NSGA-II,
IEEE Trans on Evolutionary Computation, 2009, 13:2, 284-302

Two different neighbourhoods are used and a new solution is allowed to replace a very small number of old solutions in this version

Qingfu Zhang Homepage

<http://dces.essex.ac.uk/staff/qzhang/>



MOEA/D Homepage

<http://dces.essex.ac.uk/staff/qzhang/webofmoead.htm>

5. Métricas de Comparación

Dados dos conjuntos de soluciones no dominadas (Pareto) X' y X'' , la función C nos permite ordenarlos en $[0,1]$:

$$C(X', X'') := \frac{|\{a'' \in X''; \exists a' \in X' : a' \preceq a''\}|}{|X''|}$$

$C(X', X'')$ mide el grado de dominancia de X' sobre X'' .

Véase que $C(X', X'') \neq C(X'', X')$.

Métricas de Comparación (2)

■ M_1

Distancia
al Pareto
Optimal

$$M_1(X') = \frac{1}{|X'|} \sum_{a' \in X'} \min \{ \|a' - \bar{a}\|_H; \bar{a} \in \bar{X} \}$$
$$M_1^*(X') = \frac{1}{|X'|} \sum_{p' \in X'} \min \{ \|p' - \bar{p}\|; \bar{p} \in \bar{Y} \}$$

■ M_2

Distribución
de soluciones
No-dominadas.

$$M_2(X') = \frac{1}{|X'-1|} \sum_{a' \in X'} | \{b' \in X'; \|a' - b'\| > \sigma\} |$$

Igual sobre los objetivos

■ M_3

Extensión
de la
Frontera

$$M_3(X') = \sqrt{\sum_{i=1}^m \max \{ \|a'_i - b'_i\|_H; a', b' \in X' \}}$$
$$M_3^*(Y') = \sqrt{\sum_{i=1}^n \max \{ \|p'_i - q'_i\|; p', q' \in Y' \}}$$

CONCLUSIONES

Los MOEAs se han convertido en una de las áreas de investigación más activas en el ámbito de la Computación Evolutiva.

Su aplicabilidad en los problemas multiobjetivo es clara e inmediata, y se han convertido en una herramienta muy importante para abordar dichos problemas.

Es un área consolidada y a la vez muy abierta desde la doble perspectiva de la investigación en el desarrollo de nuevos MOEAs (incorporación de preferencias, funciones dinámicas, espacios con restricciones, escalabilidad en cuanto al número de objetivos, trade-off eficiencia-eficacia en problemas complejos, paralelismo, ...), como de la aplicación.

La comparación de las soluciones obtenidas por un MOEA (de las aproximaciones del frente de Pareto) es un problema complejo

Conclusiones (2)

¿Qué aproximación del frente de Pareto es mejor?

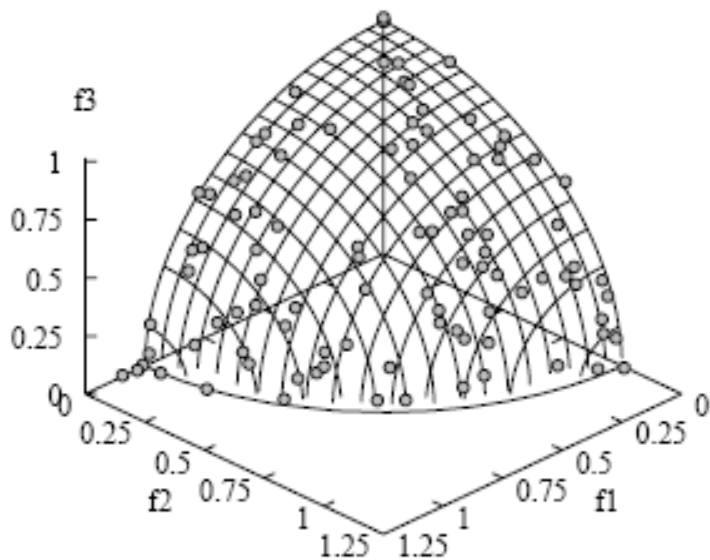


Figure 9: Non-dominated points obtained using NSGA-II.

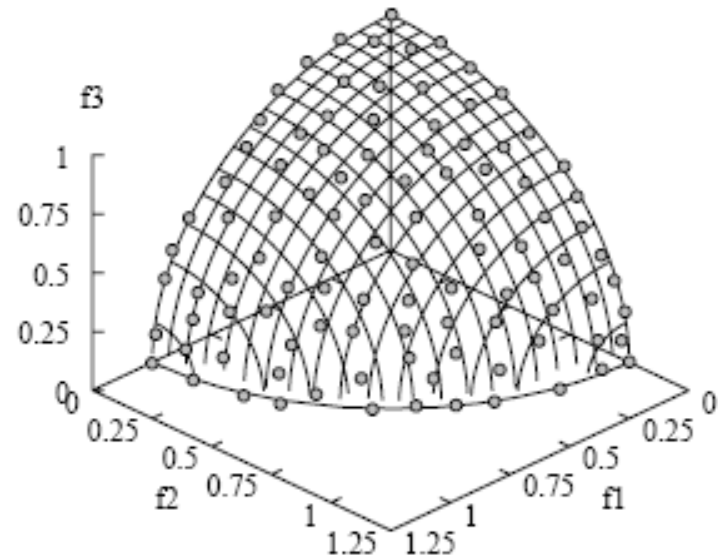


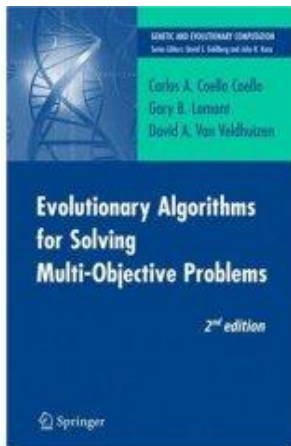
Figure 10: Non-dominated points obtained using SPEA2.

MÁS SOBRE MOEAS

PARA SABER MAS SOBRE OPTIMIZACIÓN EVOLUTIVA MULTI OBJETIVO

Visitar el repositorio de EMOO localizado en:

<http://delta.cs.cinvestav.mx/~ccoello/EMOO>



C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Pub., 2007 (second-edition) .



Evolutionary Multi-Criterion Optimization

Third Int. Conf, EMO 2005, Guanajuato, Mexico, March 9-11, 2005, Proceedings
Series: [Lecture Notes in Computer Science](#), Vol. 3410

Coello Carlos A.; Hernández, Arturo; Zitzler, Eckart (Eds.) 2005, XVI, 912 p.,



C.A. Coello

LECTURAS

Lecturas Básicas: <http://sci2s.ugr.es/seminars/seminar.php?id=4>

C.A. Coello. Evolutionary Multiobjective Optimization: Current and Future Challenges. In J. Benitez, O. Cordon, F. Hoffmann, and R. Roy (Eds.), Advances in Soft Computing--- Engineering, Design and Manufacturing. Springer-Verlag, September, 2003, pp. 243 - 256.

E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7:2, April, 2003, pp. 117 - 132. ([PDF, 1172 Kb](#))

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6:2, April, 2002, pp. 182 - 197.

M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation 10:3, Fall, 2002, pp. 263 - 282.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg (Eds.), Evolutionary Multiobjective Optimization. Theoretical Advances and Applications. Springer, USA, 2005, pp. 105 - 145.

Enlaces Software

SPEA

<http://www.tik.ee.ethz.ch/%7ezitzler/testdata.html#source>

NSGAI I

<http://www.iitk.ac.in/kangal/codes.shtml>

MOMHLib++

Open source Multiple-Objective MetaHeuristics Library in C++

<http://www-idss.cs.put.poznan.pl/~jaszkiewicz/MOMHLib/>

At present the library includes the following methods:

- Pareto simulated annealing (PSA) [PSA's home page](#),
- Serafini's multiple objective simulated annealing (SMOSA) [4][5],
- Ulungu's et al. multiple objective simulated annealing (MOSA) [7],
- Pareto memetic algorithm [8],
- multiple objective genetic local search (MOGLS) [MOGLS's home page](#),
- Ishibuchi's and Murata's multiple objective genetic local search (IMMOGLS) [3],
- multiple objective multiple start local search (MOMSLS),
- non-dominated sorting genetic algorithm (NSGA) [6] and controlled NSGA II [1],
- Strength Pareto Evolutionary Algorithm [9].

EMOO-Software link:

<http://www.lania.mx/~ccoello/EMOO/EMOOsoftware.html>

BIOINFORMÁTICA

2013 - 2014

PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- **Tema 13. Programación Genética**
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados