Universidad de Oviedo

Intelligent Scheduling and Optimization

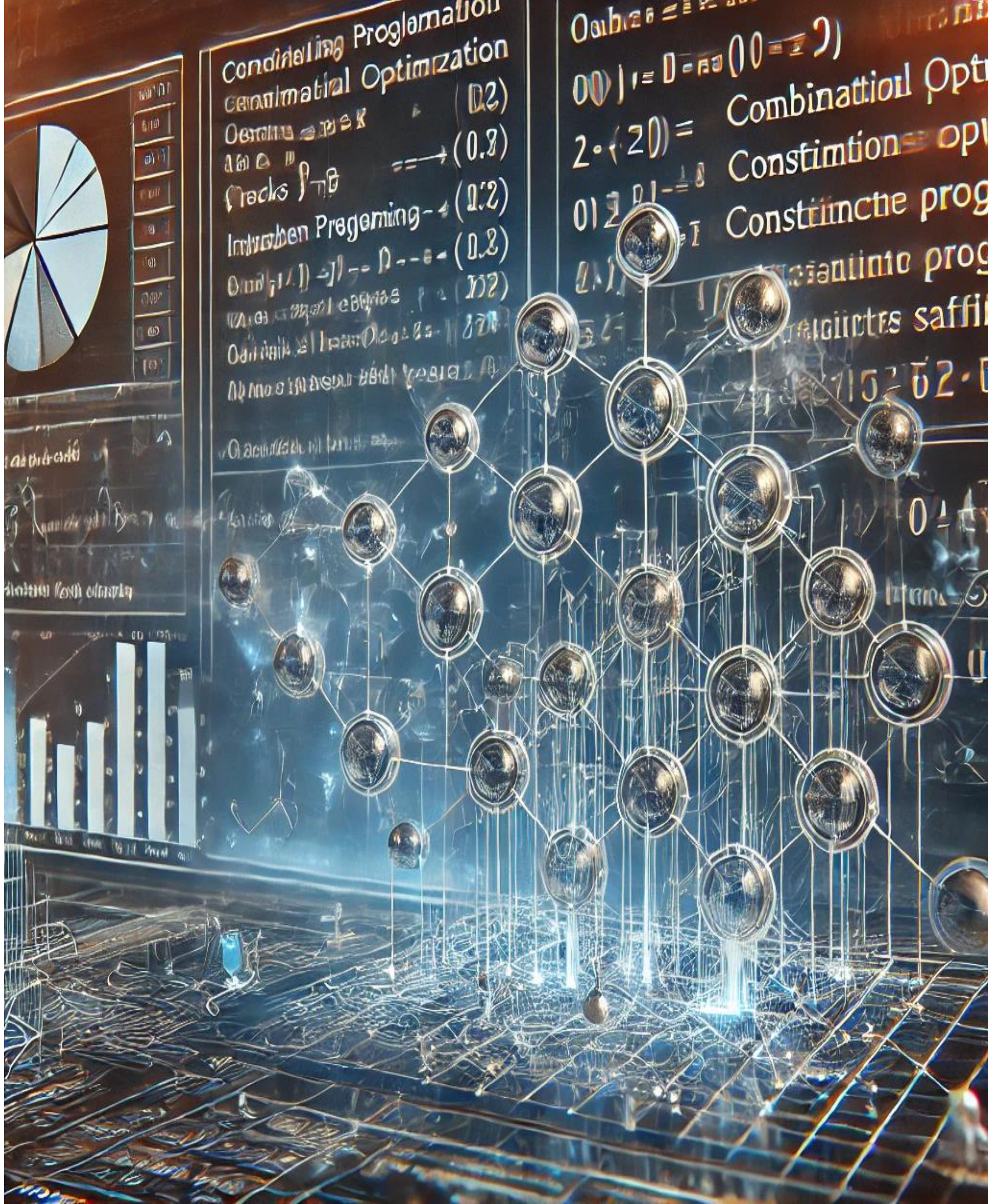
Técnicas de Inteligencia Artificial para la Optimización y Programación de Recursos

Optimización Multiobjetivo

Jorge Puente Peinador
puente@uniovi.es

Ciencia de la Computación e Inteligencia Artificial
Departamento de Informática

ceisIA Centro de Estudios sobre el Impacto Social de la Inteligencia Artificial | Universidad de Oviedo



Contenidos

Optimización Multiobjetivo

Introducción

- Muchos problemas requieren optimizar varios objetivos simultáneamente
- Por ejemplo, la mayor parte de los problemas en ingeniería de diseño requieren minimizar costes y maximizar el rendimiento y fiabilidad
- El coste, el rendimiento y la fiabilidad son a menudo objetivos contradictorios: para reducir los costes tenemos que reducir la fiabilidad o el rendimiento
- La optimización mono-objetivo es un caso particular de optimización multiobjetivo

Optimización Multiobjetivo

Formulación del problema

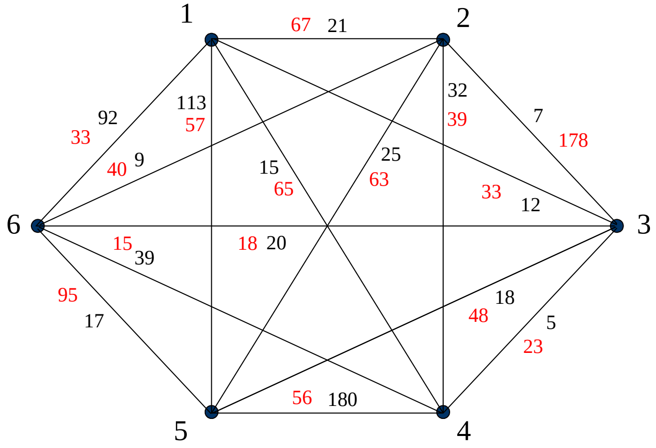
- Dado un vector n-dimensional de variables de decisión $x=\{x_1,\ldots,x_n\} \in \mathbf{X}$, encontrar un vector x^* que minimice (o maximice) un conjunto de K funciones objetivo $z(x) = \{z_1(x),\ldots,z_K(x)\} \in \mathbf{Y}$
  - **X** es el espacio de decisión
  - **Y** es el espacio objetivo. Generalmente $\mathbf{Y} \subseteq R^K$
  - $\{z_1,\ldots,z_K\}$ es el conjunto de funciones objetivo
  - Puede haber restricciones adicionales:
    - Desigualdades
    - Igualdades
    - Otras

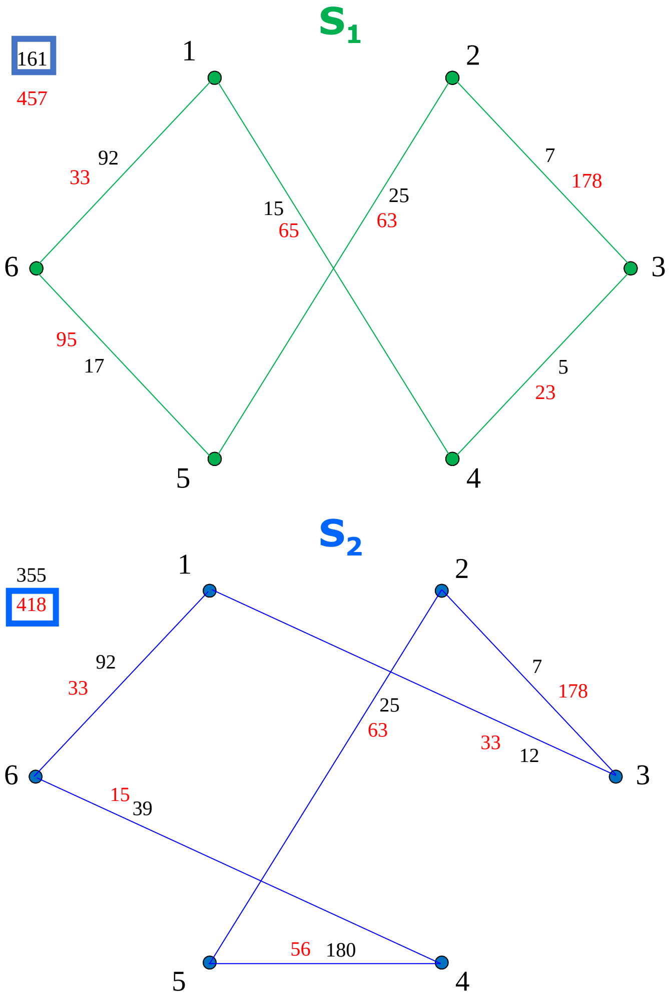
Optimización Multiobjetivo
Un ejemplo: TSP con distancia y coste

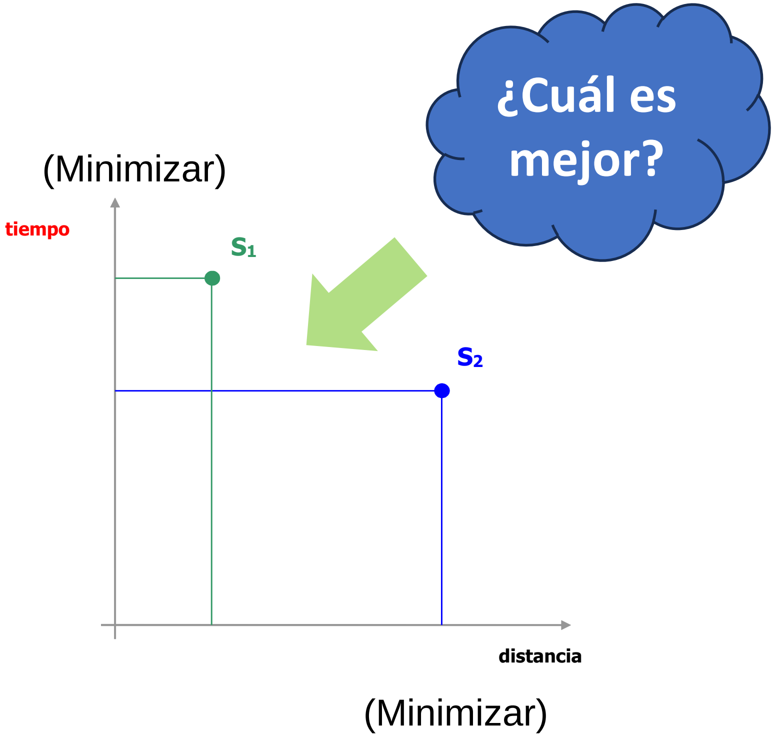
- Ejemplo: TSP con distancias y costes en cada arco
  - Podemos imaginar que utilizamos autopistas de peaje, y entonces puede ocurrir que caminos más cortos sean más caros, por lo que distancia y coste podrían ser objetivos contradictorios
  - **X = C**^{n},
    - **C** es el conjunto de ciudades
    - **n** es el número de ciudades
  - **Y** $\subseteq R^2$
  - Funciones objetivo
    - $z_1(x)$ = distancia de la ruta x
    - $z_2(x)$ = coste necesario para recorrer la ruta x
  - Restricciones

Optimización Multiobjetivo

Un ejemplo: TSP con distancia y tiempo







Intelligent Scheduling and Optimization

Optimización Multiobjetivo

Tres enfoques

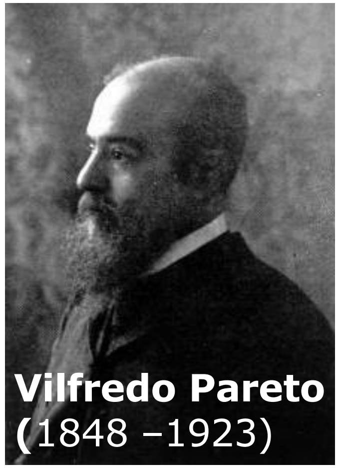
- Agregación de funciones
  - Todos los objetivos se agregan en uno único, por ejemplo ponderando cada uno de ellos mediante pesos. Y luego se aplica una metaheurística mono-objetivo
    - $f(x) = F(z_1(x),\ldots,z_K(x))$ (por ejemplo $z_1w_1+z_2w_2+\ldots+z_kw_k$)
    - Debemos tener cuidado con la magnitud de los objetivos, podría ser necesario escalar el fitness
- Orden lexicográfico
  - Los objetivos se consideran en un orden jerárquico
    - Primero min $z_{[1]}(x)$, después min $z_{[2]}(x)$, después min $z_{[3]}(x)$, …
    - Es decir, le damos más importancia al primer objetivo y utilizamos los demás para desempatar entre soluciones cuyo valor del primer objetivo sea idéntico
- Optimización Pareto
  - El objetivo es encontrar el frente Pareto, o al menos una buena aproximación

Optimización Pareto

Conceptos de dominancia

(Minimizando: inferior es mejor)

- Dadas dos soluciones x e y, x domina y (se denota $x \succ y$), si y solo si $z_i(x) \leq z_i(y)$ para todo $1 \leq i \leq K$, y además $z_j(x) < z_j(y)$ para al menos un valor de j (en términos generales denotamos por $z(x) \succ z(y)$ que z(x) domina a z(y))
  - Es decir, que una solución domina a otra si es mejor o igual en todos los objetivos, y estrictamente mejor en al menos uno de ellos
- Una solución es Pareto óptima si no está dominada por ninguna otra en el espacio solución
- El conjunto de todas las soluciones no dominadas $X^* \subset \mathbf{X}$ es el Conjunto Pareto óptimo y representa la solución óptima del problema multiobjetivo
- Los valores de la función objetivo del conjunto Pareto óptimo $z(X^*) \subset \mathbf{Y}$ forman el Frente Pareto

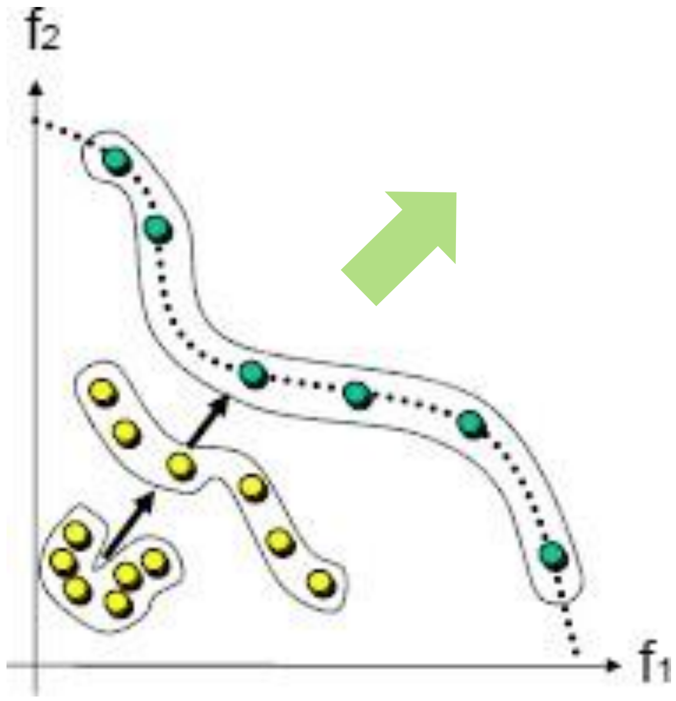


Optimización Pareto

Objetivo

- En la optimización Pareto el objetivo es encontrar una aproximación al frente Pareto que se aproxime al verdadero frente Pareto tanto como se pueda
  - Debe estar lo más cerca posible del verdadero frente Pareto
  - Las soluciones deben distribuirse uniformemente y ser distintas en el frente Pareto
  - Debe capturar todo el espectro del frente Pareto, incluyendo los extremos finales

(Maximizando)



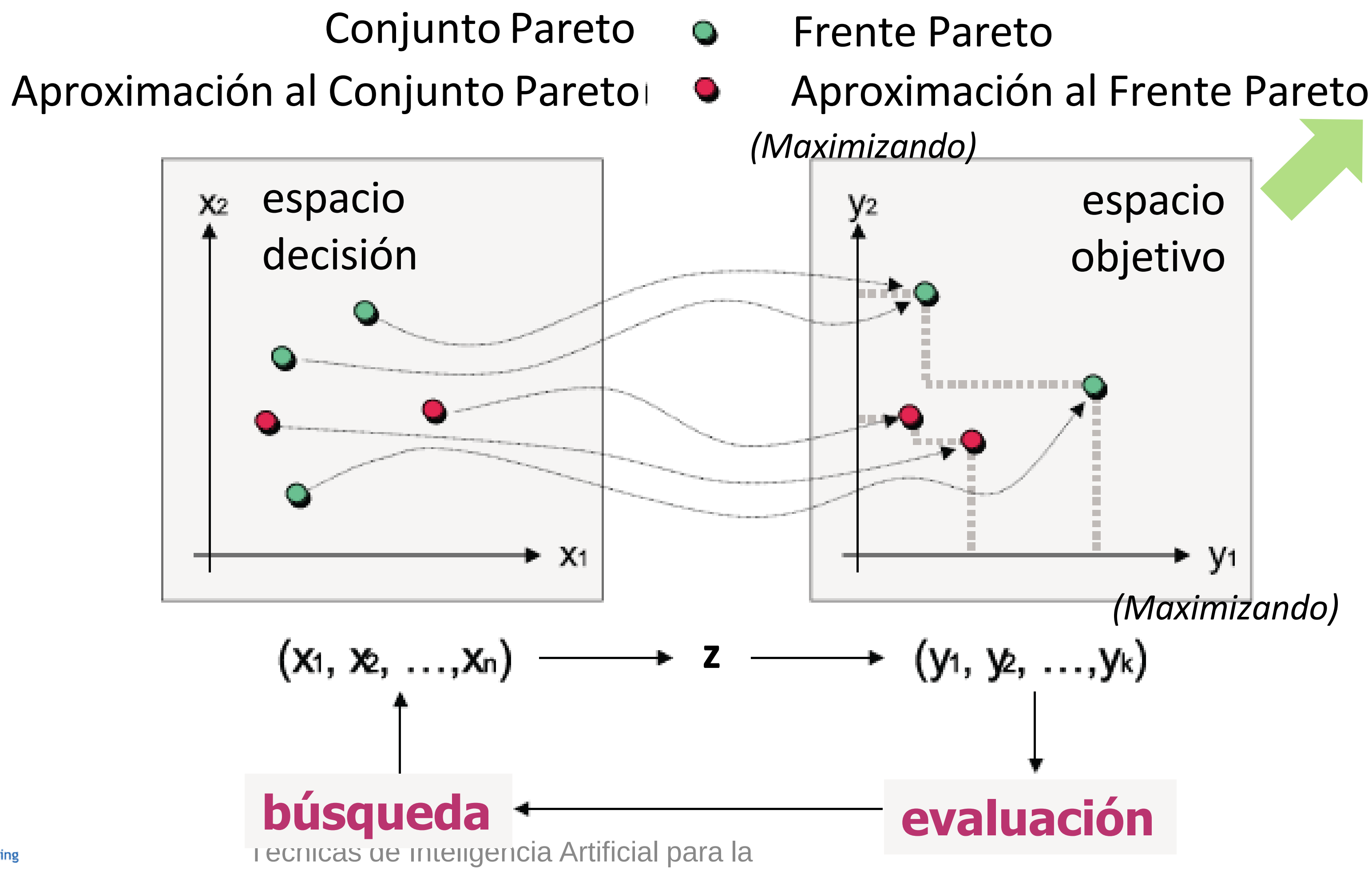
(Maximizando)

Solución del Frente Pareto

Solución de una aproximación del frente Pareto

Optimización Pareto

Un ejemplo con $X, Y \subseteq \mathbb{R}^2$ *[Zitzler, 2004]*



Contenidos

1. Introducción a la Optimización Multiobjetivo
2. **Algoritmos Genéticos Multiobjetivo**
3. Métricas para Algoritmos Multiobjetivo

Algoritmos Genéticos Multiobjetivo

Introducción

- Los algoritmos genéticos son muy prometedores como algoritmos de búsqueda multiobjetivo, por varias razones
  - Están basados en poblaciones, así que son adecuados para obtener múltiples soluciones
  - Se pueden adaptar para buscar en diferentes regiones del espacio de búsqueda simultáneamente
  - No son sensibles a la forma que tenga el frente Pareto
- La mayoría de las propuestas multiobjetivo en la literatura son metaheurísticas, en particular algoritmos evolutivos (70%)
- La mayoría de ellos (90%) tratan de aproximar el conjunto Pareto

Algoritmos Genéticos Multiobjetivo

Algunos enfoques clásicos [Abdullah, 2006]

- VEGA: Vector Evaluated GA *[Schaffer, 1985]*
- MOGA: MultiObjective GA *[Fonseca, 1993]*
- WBGA: Weight Based GA *[Hajela, 1992]*
- RWGA: Random Weighted GA *[Murata, 1995]*
- NSGA: Non-dominated Sorting GA *[Srinivas, 1994]*
- $\boxed{\textbf{NSGA-II: Non-dominated Sorting GA - II } \textit{[Deb, 2002]}}$
- SPEA: Strength Pareto EA *[Zitzler, 1999]*
- SPEA2: Strength Pareto EA 2 *[Zitzler, 2001]*
- PESA: Pareto Envelope-based Selection Algorithm *[Corne, 2000]*
- PAES: Pareto Archived Evolution Strategy *[Knowles, 1999]*
- RDGA: Rank-Density based GA *[Lu, 2003]*
- DMOEA: Dynamic MO EA *[Yen, 2003]*
- NPGA: Niched Pareto GA *[Horn, 1994]*
- …

Algoritmos Genéticos Multiobjetivo

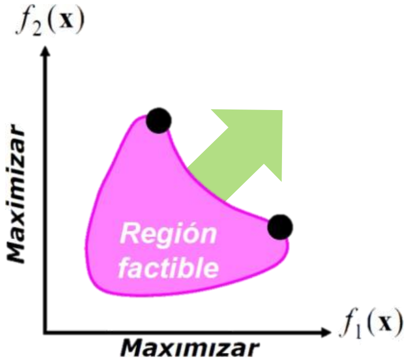
Aspectos relevantes [Abdullah, 2006]

- **Funciones de Fitness**
  - **Ponderaciones**
  - **Alternando las funciones de fitness**
  - **Aproximaciones Pareto por ranking (1ª generación MOEAs)**
    - **Rank, depth y count**
- Ajuste del Fitness en pro de la diversidad (2ª generación MOEAs)
  - Medidas de densidad: fitness sharing y niching, crowding, . . .
- Elitismo (3ª generación MOEAs)
  - Estrategias para el Elitismo
  - Poblaciones externas

Ponderaciones de la función objetivo

Descripción

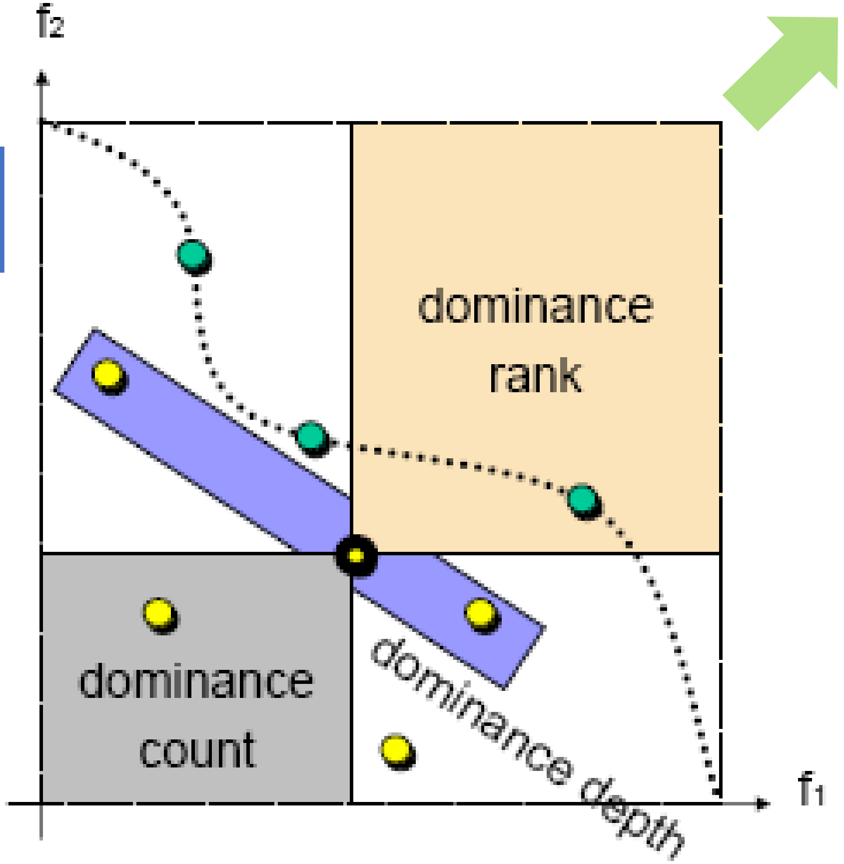
- Se define una única función objetivo:
  - $z(x) = w_1 z_1'(x) + \ldots + w_K z_K'(x)$
    - $z_1' \ldots z_K'$ son las funciones objetivo **normalizadas**
    - **w** $=\{w_1,\ldots,w_K\}$ son los pesos, $\sum_{i=1,K} w_i = 1$
  - ¿Cómo determinar el vector de pesos **w**? Varias opciones:
    - **w** constante
    - **w** variable en cada ejecución: cada **w** devuelve una solución distinta
    - **w** evoluciona: un cromosoma representa (**x**,**w**) (propuesta utilizada en WBGA *[Hajela,1992]*)
    - **w** aleatorio para cada **x** (propuesta utilizada en RWGA *[Murata, 1996]*)
- Ventajas:
  - Es fácil de implementar y no es costoso en tiempo de ejecución
- Inconvenientes:
  - Puede tener dificultades para encontrar el Frente Pareto (por ejemplo, en **regiones cóncavas**)



Aproximaciones del Pareto por ranking

Descripción y variantes

- La población se ordena por **ranking**, y a cada cromosoma se le asigna un fitness basado en su rango y no en el valor de la función objetivo
- En *[Zitzler, 2004]* se distinguen tres métodos de ranking:
  - Depth: La población se divide en frentes a diferentes profundidades — *NSGA, NSGA-II*
  - Rank: El número de individuos que dominan a un individuo — *MOGA, NPGA*
  - Count: El número de individuos que un individuo domina — *SPEA, SPEA2*



Aproximaciones del Pareto por ranking

Dominance Depth

- La primera aproximación se debe a Goldberg *[Goldberg, 1989]*

Step 1: Set $i = 1$ and $TP = P$.
Step 2: Identify non-dominated solutions in TP and assigned them set to F_i.
Step 3: Set $TP = TP \backslash F_i$. If $TP = \varnothing$ go to Step 4, else set $i = i + 1$ and go to Step 2.
Step 4: For every solution $\mathbf{x} \in P$ at generation t, assign rank $\boxed{r_1(\mathbf{x}, t) = i \text{ if } \mathbf{x} \in F_i.}$

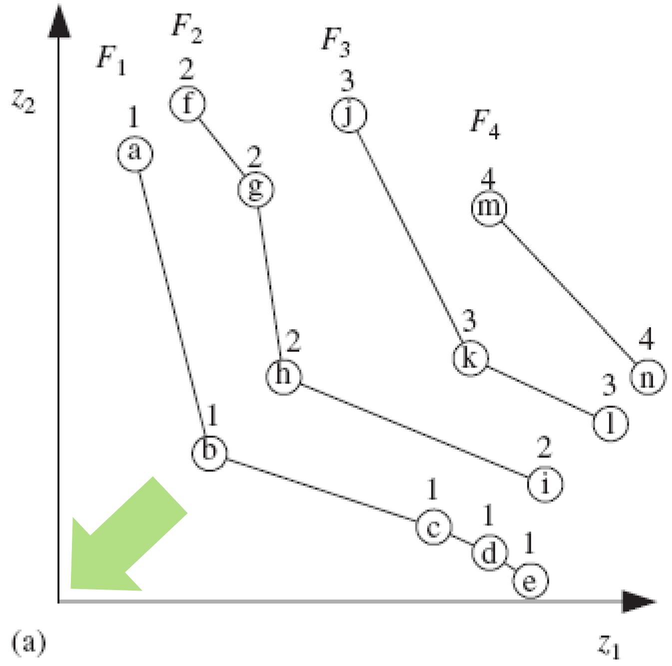
- $F_1, \ldots, F_n$ son los frentes no-dominados y F_1 es el frente Pareto de P(t)

$\forall x \in P(t)$

$$\boxed{r_1(x,t) = i, \text{ si } x \in F_i}$$

- Método utilizado en NSGA-II *[Deb, 2002]*

(Minimizando)



(Minimizando)

Algoritmos Genéticos Multiobjetivo

Aspectos relevantes [Abdullah, 2006]

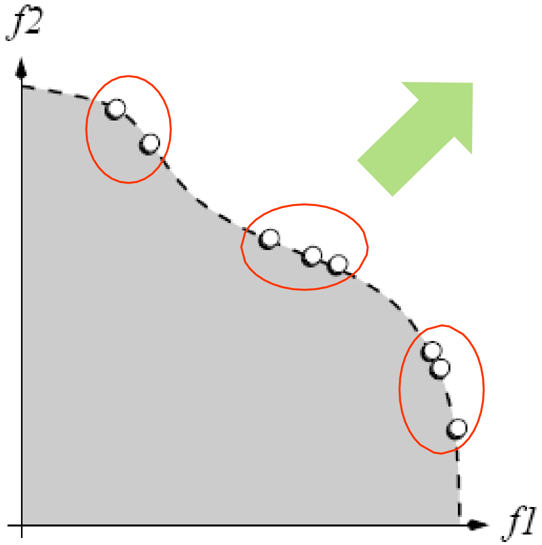
- Funciones de Fitness
  - Ponderaciones
  - Alternando las funciones de fitness
  - Aproximaciones Pareto por ranking (1ª generación MOEAs)
    - Rank, depth y count
- **Ajuste del Fitness en pro de la diversidad (2ª generación MOEAs)**
  - **Medidas de densidad: fitness sharing y niching, crowding, . . .**
- Elitismo (3ª generación MOEAs)
  - Estrategias para el Elitismo
  - Poblaciones externas

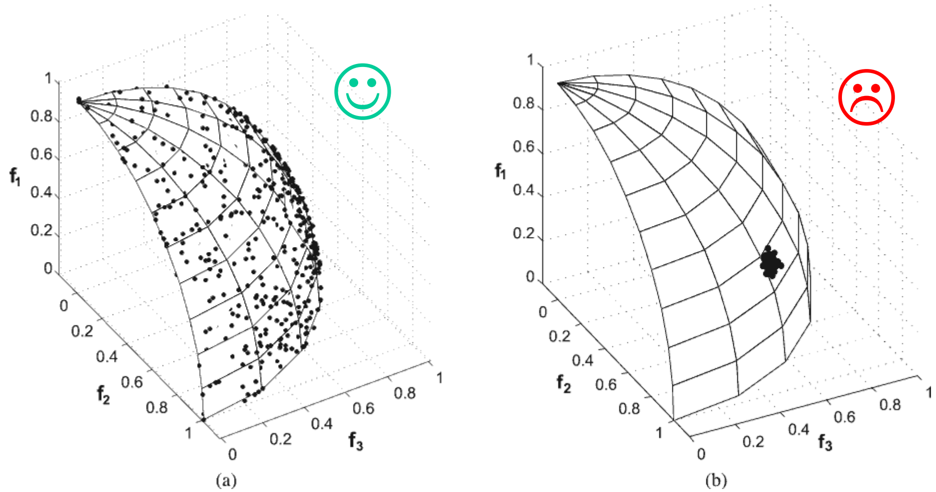
Ajuste del Fitness en pro de la diversidad

Deriva Genética

MOGA SPEA2

- Los rankings r_2 y r_3 favorecen la diversidad, ya que penalizan las soluciones dominadas por muchas, o que dominan a muchas otras, en el espacio objetivo
- Aun así, en los Algoritmos Genéticos multiobjetivo la población tiende a formar relativamente pocos grupos en el espacio objetivo: este efecto se denomina Deriva Genética (Genetic Drift)





Ajuste del Fitness en pro de la diversidad

Técnicas

- Las técnicas habituales para gestionar la diversidad utilizan información de densidad en el proceso de selección *[Zitzler, 2004]*
  - La densidad necesita de medidas de distancia, ya sea en el espacio de decisión o en el espacio de objetivos
  - La posibilidad de selección se reduce en función de la densidad de individuos en su vecindad
- Principales propuestas para prevenir la deriva genética y mejorar la diversidad *[Abdullah, 2006]*
  - Funciones Kernel
    - Fitness sharing and niching: MOGA, NPGA
  - Vecino más próximo
    - K-closest neighbor: SPEA2
    - Crowding distance: NSGA II
  - Densidad basada en celdas: PESA, PAES

Ajuste del Fitness en pro de la diversidad

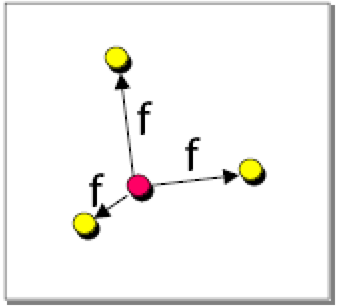
Técnicas

Density estimation techniques: [Silverman 86]

Kernel

MOGA, NPGA

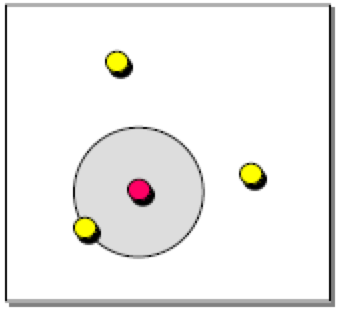
density estimate
=
sum of f values where f is a function of the distance



Nearest neighbor

NSGA-II, SPEA2

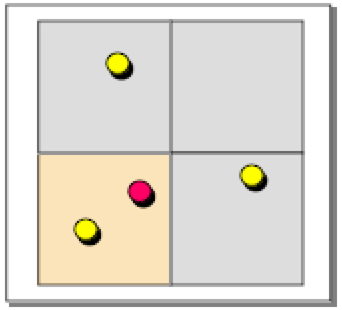
density estimate
=
volume of the sphere defined by the nearest neighbor



Histogram

PAES, PESA

density estimate
=
number of solutions in the same box



Ajuste del Fitness en pro de la diversidad

Crowding distance: NSGA-II [Deb, 2002]

- La crowding distance es una medida de la dispersión local de las soluciones en un frente Pareto F_i
- El objetivo es obtener una distribución uniforme de las soluciones a lo largo del frente Pareto sin utilizar parámetros como σ_{share} o k
- Es el método utilizado en NSGA-II *[Deb, 2002]*
- La crowding distance se utiliza para desempatar en el operador de selección de torneo por crowding
  - Seleccionamos al azar dos soluciones *x* e *y*. La ganadora será la que tenga el dominance depth más bajo. En el caso de que estén en el mismo frente, la que tenga mayor distancia de crowding es la ganadora

Ajuste del Fitness en pro de la diversidad

Crowding distance: NSGA-II [Deb, 2002]

$\boldsymbol{cd}$: Crowding Distance

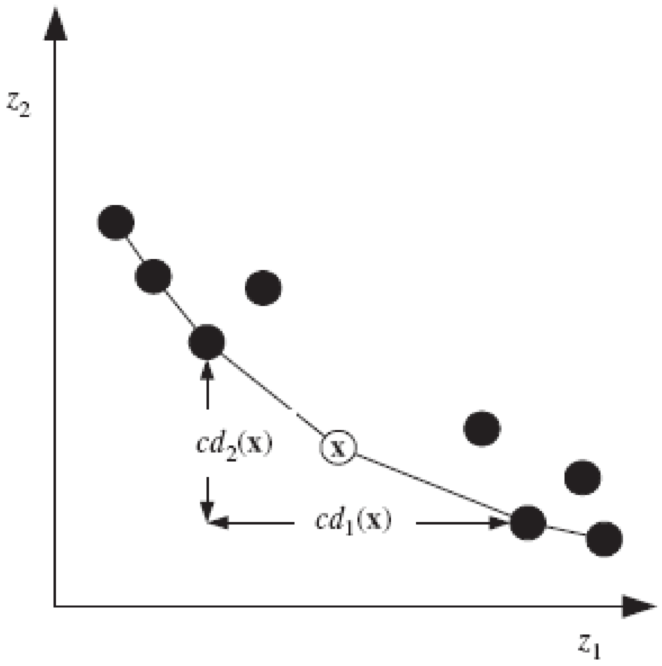
Calcula el Rankig de frentes no-dominados $F_1, F_2, \ldots, F_R$ y para cada frente F_i repetir:

Para cada función objetivo k, ordenamos las $l = |F_i|$ soluciones en orden ascendente, y calculamos $\boldsymbol{cd_k}$

$$\boldsymbol{cd_k}(x_{[j,k]}) = \begin{cases} \infty, si\ k \in \{1, l\} \\ \dfrac{z_k(x_{[j+1,k]}) - z_k(x_{[j-1,k]})}{z_k^{max} - z_k^{min}}, en\ otro\ caso \end{cases}$$

Calculamos $\boldsymbol{cd}$ para el conjunto de objetivos

$$\boldsymbol{cd}(x) = \sum_k \boldsymbol{cd_k}(x)$$



Algoritmos Genéticos Multiobjetivo

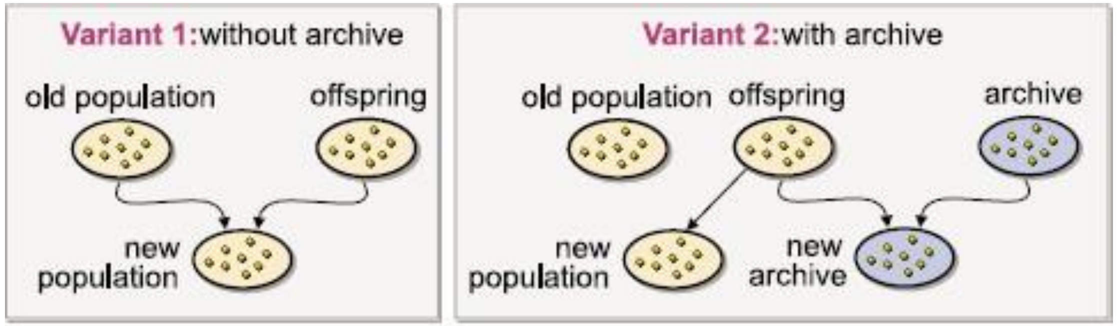
Aspectos relevantes [Abdullah, 2006]

- Funciones de Fitness
  - Ponderaciones
  - Alternando las funciones de fitness
  - Aproximaciones Pareto por ranking (1ª generación MOEAs)
    - Rank, depth y count
- Ajuste del Fitness en pro de la diversidad (2ª generación MOEAs)
  - Medidas de densidad: fitness sharing y niching, crowding, . . .
- **Elitismo (3ª generación MOEAs)**
  - **Estrategias para el Elitismo**
  - **Poblaciones externas**

Elitismo

Motivación y posibles enfoques

- En un algoritmo evolutivo pueden perderse muchas soluciones buenas de una generación a la siguiente, debido a los efectos aleatorios
- Una solución a este problema es el elitismo. En algoritmos evolutivos multi-objetivo, hay dos enfoques habituales *[Zitzler, 2004]*
  - Mantener las soluciones no dominadas en la población
    - Elitismo puro: mantenemos solo individuos no dominados en la población (población de tamaño variable
    - NSGA-II: elitismo con tamaño de población fija
  - Uso de un archivo externo con las soluciones no dominadas



Elitismo

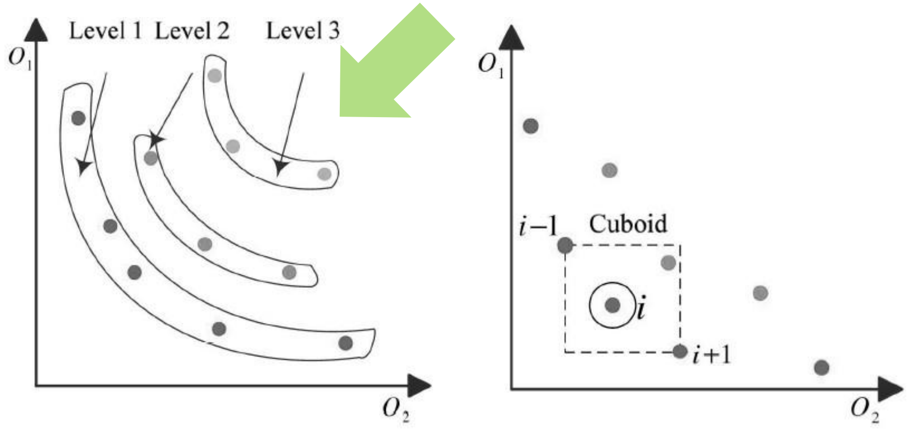
Elitismo sin archivo externo: NSGA-II [Deb, 2002]

Selección:

1.- Torneo (N:1)

Criterio: *Ranking* → (empate) → *cd*

Mantiene la diversidad, ya que se basa en crowding distance



(a) Non-domination levels.

(b) Crowding distance calculation between solutions in the same non-domination level.

Figure 7: The NSGA-II replacement strategy.

Non-dominated Sorting Genetic Algorithm II

NSGA-II

```
Parámetros de Entrada (Pc, Pm, NumGen, PopSize, … );
t ← 0;
Init(P(t));                  // Población Inicial
Evaluate(P(t));              // Función Fitness: (Decodificación y Evaluación)
while ( t < NroGen)   {
    t ← t+1;
    P'(t) = Select(P(t-1));            // Selección
    Q(t) = Alter(P'(t));               // Cruce y Mutación
    Evaluate(Q(t));                    // Función fitness
    P(t) = Replace(P(t-1), Q(t));      // Remplazamiento
}
end.
```

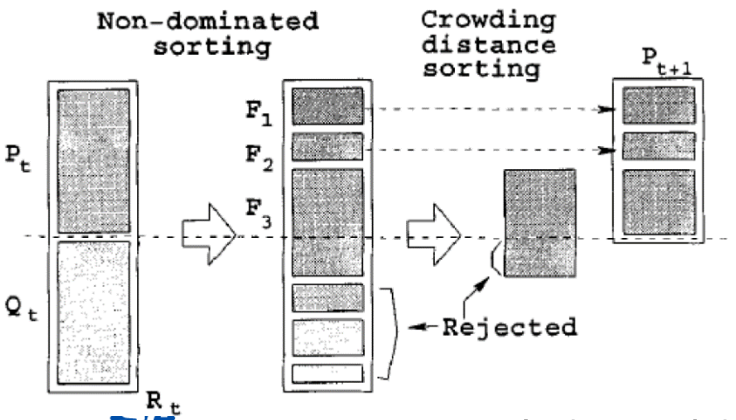
Elitismo

Elitismo sin archivo externo: NSGA-II [Deb, 2002]

Selección:
1.- Torneo (N:1)
Criterio: *Ranking* → (empate) → *cd*

Mantiene la diversidad, ya que se basa en crowding distance

Remplazamiento:
1.- Incluye, enteros, los mejores frentes Pareto que quepan en la población
2.- Y las soluciones menos "crowded" del siguiente frente



Non-dominated Sorting Genetic Algorithm II

NSGA-II

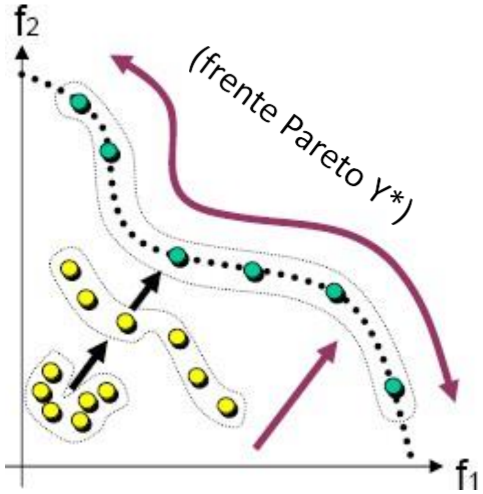
```
Parámetros de Entrada (Pc, Pm, NumGen, PopSize, … );
t ← 0;
Init(P(t));                  // Población Inicial
Evaluate(P(t));              // Función Fitness: (Decodificación y Evaluación)
while ( t < NroGen)  {
    t ← t+1;
    P'(t) = Select(P(t-1));          // Selección
    Q(t) = Alter(P'(t));             // Cruce y Mutación
    Evaluate(Q(t));                  // Función fitness
    P(t) = Replace(P(t-1), Q(t));    // Remplazamiento
}
end.
```

Contenidos

Métricas para algoritmos Multiobjetivo

Introducción

- ¿Cómo podemos evaluar el rendimiento de los algoritmos multiobjetivo?
  - En cuanto a los recursos computacionales
    - Tiempo de ejecución general
    - Número de evaluaciones del fitness (en EAs)
  - En cuanto a la calidad de la aproximación del conjunto Pareto
    - Cercanía a Y*
    - Distribución uniforme
    - Cobertura del espacio objetivo
    - Otros

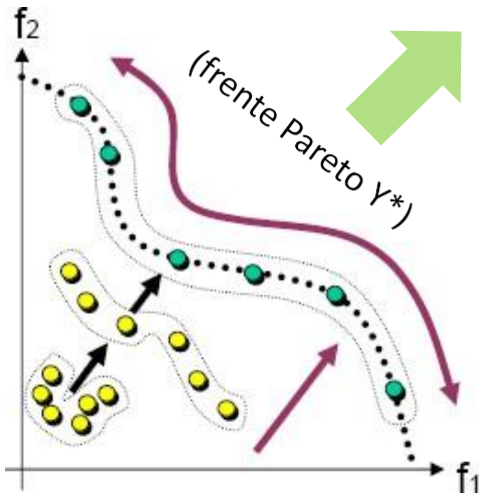


Métricas para algoritmos Multiobjetivo

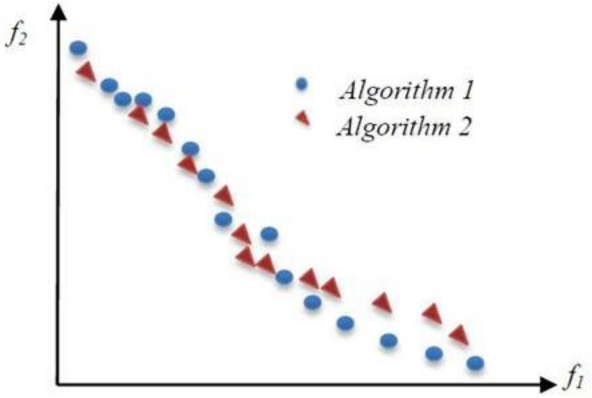
Introducción

- ¿Cómo se pueden comparar dos algoritmos MO A1 y A2 si consumen recursos similares?
  - A_2 es mejor que A_1 si la aproximación al frente Pareto Y_2 calculada para A_2 "domina" al frente Y_1 procedente de A_1, es decir:

$$\boxed{\forall a_1 \in Y_1 \ \exists \ a_2 \in Y_2 \text{ tal que } a_2 \succ a_1}$$



  - Pero si ni Y_1 domina a Y_2, ni Y_2 domina a Y_1, en principio ambos algoritmos no son comparables y no se podría decir que uno es mejor que otro

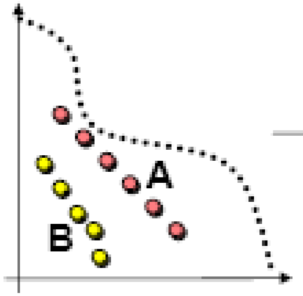


Métricas para algoritmos Multiobjetivo

Indicadores de calidad

- Indicadores unarios: asignan a cada aproximación un número real I(A)
- Indicadores binarios: asignan a cada par de aproximaciones un número real I(A,B)

Example: unary indicators combined



	A	B
hypervolume	*432.34*	*420.13*
distance	*0.3308*	*0.4532*
diversity	*0.3637*	*0.3463*
spread	*0.3622*	*0.3601*
cardinality	*6*	*5*

→ "A better"

© Eckart Zitzler ETH Zurich Two Decades of EMO 26

- *[Zitzler et al., 2003]*: se necesitaría un número infinito de indicadores unarios para detectar, en general, si A es mejor que B

Métricas para algoritmos Multiobjetivo

Ejemplo: distribución uniforme [Deb, 2002]

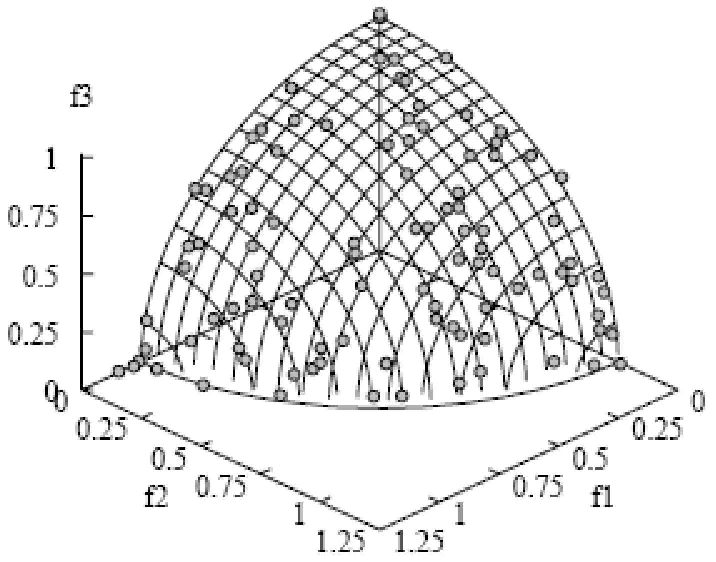


Figure 9: Non-dominated points obtained using NSGA-II.

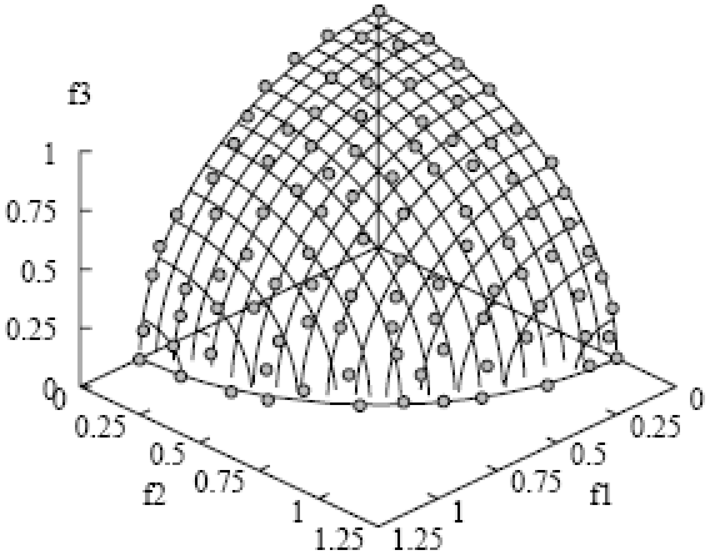
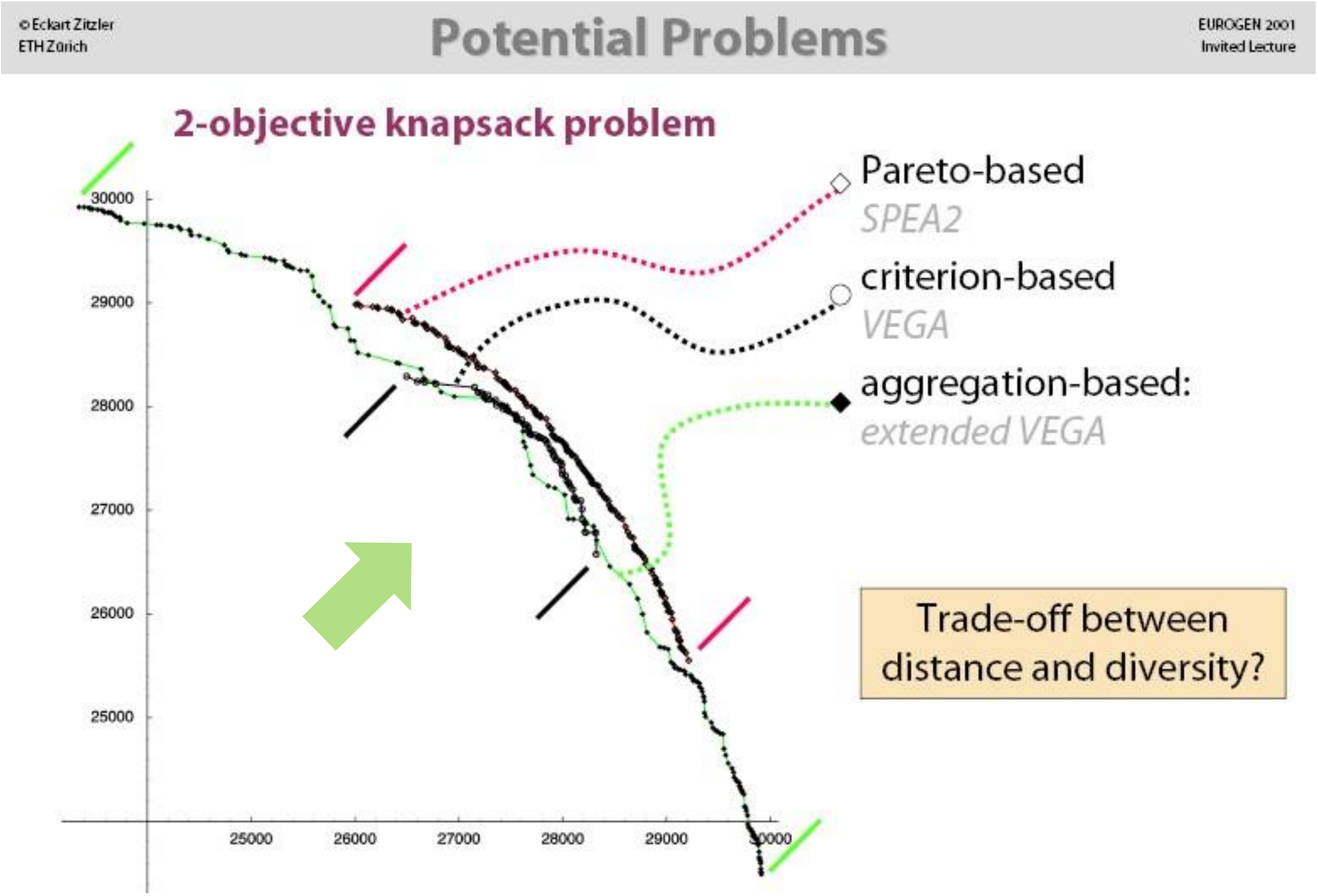


Figure 10: Non-dominated points obtained using SPEA2.

Métricas para algoritmos Multiobjetivo

Ejemplo: distancia y cobertura



Métricas para algoritmos Multiobjetivo

Un ejemplo engañoso [Zitzler, 2006]

- Respecto a distancia media, diversidad y cardinalidad

S mejor que T

¿i T mejor que S !?

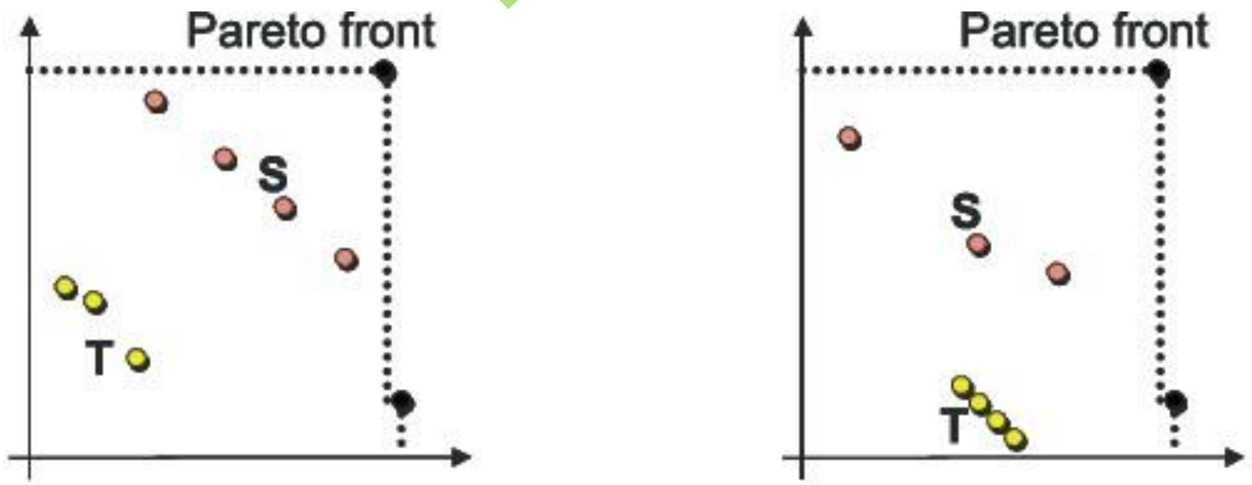
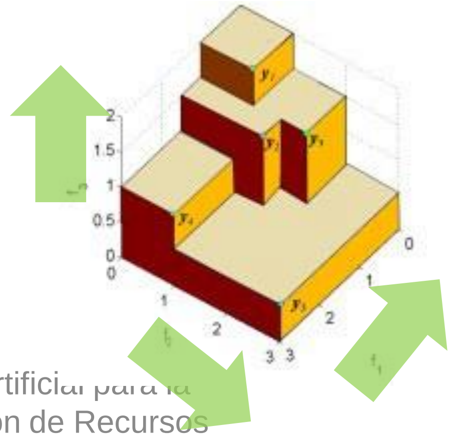
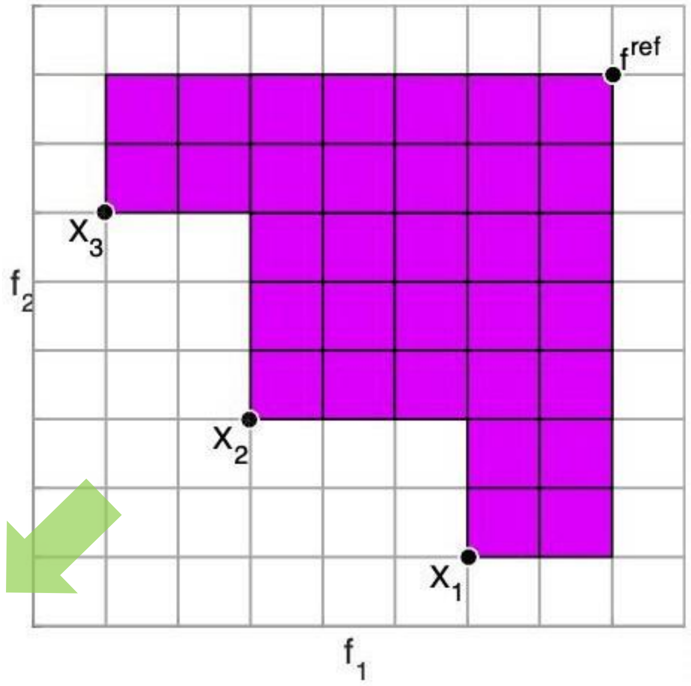


Fig. 14. Two scenarios where a Pareto front approximation entirely dominates another Pareto front approximation

Métricas para algoritmos Multiobjetivo
Indicadores unarios: hipervolumen

- Hipervolumen (Hypervolume) *[Zitzler, 1999]*
  - Es posiblemente el indicador más utilizado en la literatura
  - Volumen del espacio de objetivos dominado por la aproximación al frente Pareto *Y*, y delimitado por un punto de referencia (dicho punto de referencia debe ser peor en todos los objetivos)
  - Como punto de referencia, una práctica habitual es utilizar el peor valor de cada objetivo que hayamos encontrado durante nuestras ejecuciones, multiplicado por 1,05
  - Es aconsejable normalizar los objetivos
  - Valores altos de hipervolumen son mejores que bajos





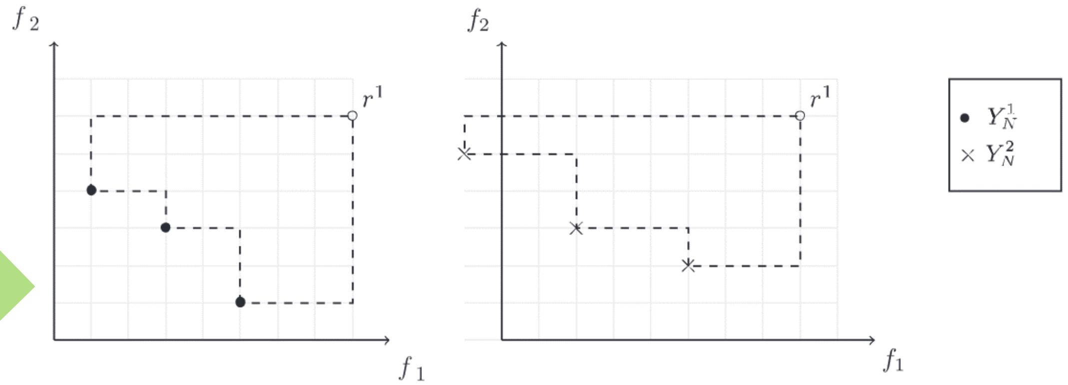
Métricas para algoritmos Multiobjetivo

Indicadores unarios: hipervolumen

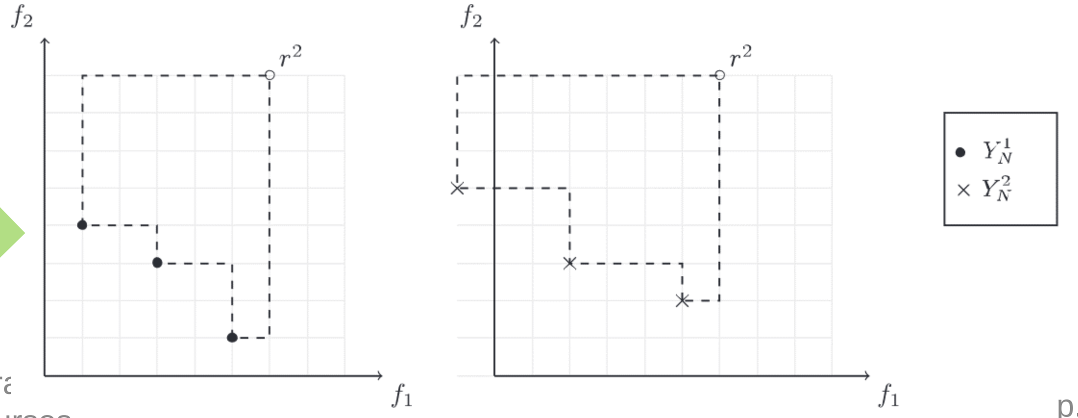
- Uno de los problemas que puede presentar el hipervolumen es que el resultado depende del punto de referencia

---

- Utilizando el punto de referencia r^1=(8,6), el hipervolumen de la aproximación Y^1 es 25, mientras que el de Y^2 es 24
  - ¿Y^1 es mejor que Y^2?



- Utilizando el punto de referencia r^1=(6,8), el hipervolumen de la aproximación Y^1 es 25, mientras que el de Y^2 es 30
  - ¿Y^2 es mejor que Y^1?



Métricas para algoritmos Multiobjetivo

Algunos otros ejemplos de indicadores unarios

- Distancia generacional *[Zitzler et al., 2000]*: Distancia promedio de las soluciones en la aproximación Pareto Y al óptimo más cercano en Y* (para calcularlo requiere el verdadero frente pareto Y*). Un menor valor se considera mejor

$$M_1^*(Y) = \frac{1}{|Y|} \sum_{a \in Y} \min\{dist(a,b), b \in Y^*\}$$

- Distribución uniforme de soluciones *[Zitzler et al., 2000]*: número de subconjuntos de Y de un cierto radio σ. Un mayor valor se considera mejor

$$M_2^*(Y,\sigma) = \frac{1}{|Y|-1} \sum_{a \in Y} |\{b \in Y; dist(a,b) > \sigma\}|$$

- Extensión de la aproximación al frente pareto *[Zitzler et al, 2000]*: solo tiene en cuenta los puntos extremos de la aproximación (en cada objetivo *i*). Un mayor valor se considera mejor

$$M_3^*(Y) = \sqrt{\sum_{i=1}^{n} \max\{dist(a_i,b_i); a,b \in Y\}}$$

- Cardinalidad: número de soluciones en Y. Un mayor valor se considera mejor

Métricas para algoritmos Multiobjetivo

Indicadores binarios: coverage indicator

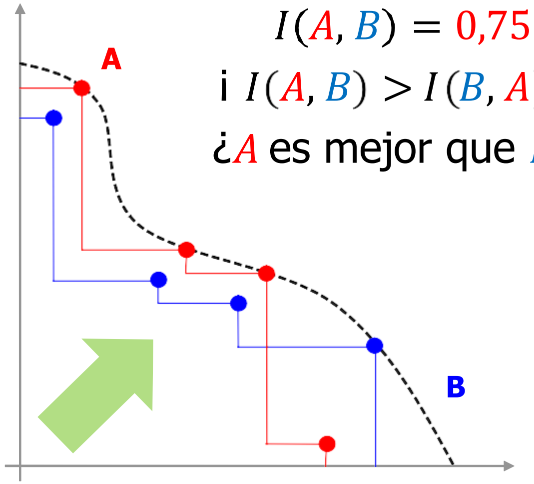
- Mientras que un indicador unario es una métrica absoluta, un indicador binario es una métrica relativa, ya que es el resultado de comparar dos frentes
- Un ejemplo de indicador binario es el Coverage Indicator:
  - $I(A,B) = \dfrac{n^{o}\ de\ soluciones\ de\ B\ que\ son\ dominadas\ por\ alguna\ solución\ de\ A}{n^{o}\ de\ soluciones\ de\ B}$
  - Si $I(A,B) = 1$, entonces todos los elementos de B son dominados por algún elemento de A (o sea que A es mejor que B)
  - Si $I(A,B) = 0$, entonces ningún elemento de B es dominado por elementos de A (ojo: esto no implica que B sea mejor que A)
  - Si ocurre que $I(A,B) \neq 1$ y además $I(B,A) \neq 1$, entonces no se puede decir que una aproximación sea mejor que la otra
    - Hay que calcular los dos valores, ya que no siempre ocurre $I(A,B) = 1 - I(B,A)$, aunque en el ejemplo mostrado sí ocurra
  - Tiene varios inconvenientes: por ejemplo, si el número de soluciones de A y B es diferente, los resultados son menos fiables

$I(B,A) = 0{,}25$

$I(A,B) = 0{,}75$

¡ $I(A,B) > I(B,A)$!

¿A es mejor que B?



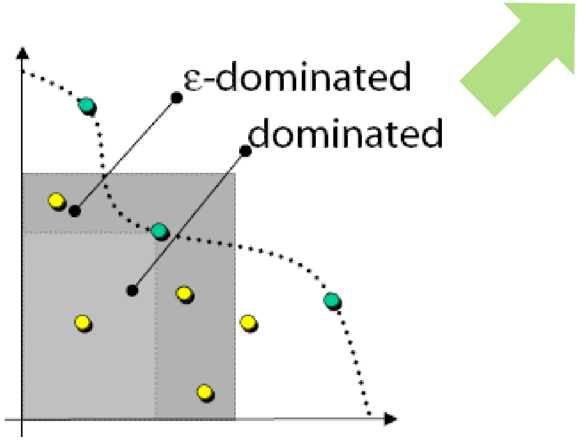
Métricas para algoritmos Multiobjetivo

Concepto de ε-dominancia [Zitzler, 2004]

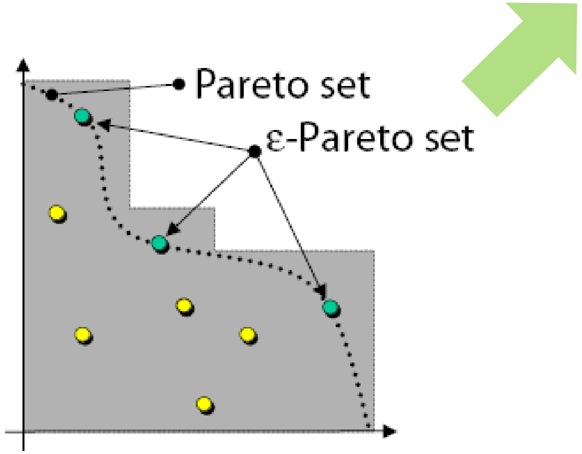
- Uno de los indicadores binarios más utilizados se basa en el concepto de ε-dominancia:

    (Maximizando: superior es mejor)

    - ε-dominancia: A ε-domina a B sí y solo sí $\varepsilon \cdot f(A) \geq f(B)$, siendo $\varepsilon > 0$, y se denota por $A \succ_{\epsilon} B$



- ε-aproximación del frente Pareto: un conjunto de soluciones Y que ε-dominan a todas las soluciones Pareto-óptimas (para algún $\varepsilon \geq 1$)
    - Es decir, que toda solución Pareto-óptima es ε-dominada por alguna solución del conjunto Y
- Frente ε-Pareto: es un subconjunto de soluciones Pareto-óptimas tales que son una ε-aproximación del frente Pareto



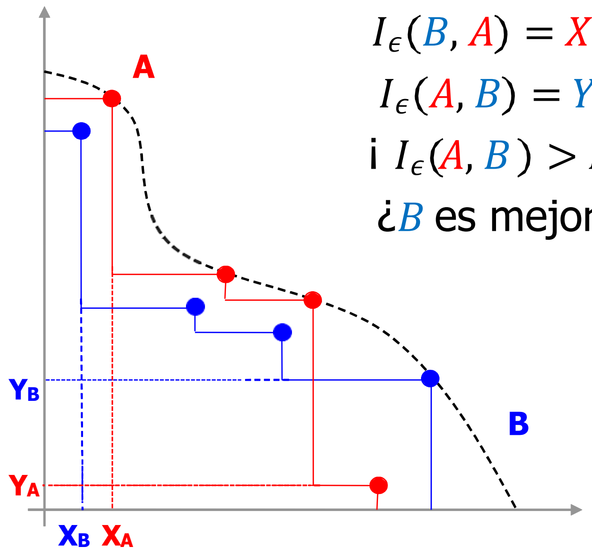
Métricas para algoritmos Multiobjetivo

Indicadores binarios: binary ε-indicator

- Uno de los indicadores binarios más utilizados es el binary ε–indicator:

$$I_\epsilon(S,T) = \min\{\epsilon \in \mathbb{R} \mid \forall b \in T\ \exists a \in S : a \succ_\epsilon b\}$$

- Es decir, el mínimo ϵ necesario para que todas las soluciones de T estén ϵ–dominadas por al menos una solución de S. Se podría decir que $I_\epsilon(S,T)$ es el factor en el que S es peor que T
- Propiedades:
  - Si $I_\epsilon(S,T) < 1$, todas las soluciones de T están dominadas por alguna solución de S
  - Si $I_\epsilon(S,T) = I_\epsilon(T,S) = 1$, S y T representan la misma aproximación al Frente Pareto
  - Si $I_\epsilon(S,T) = 1$ y $I_\epsilon(T,S) > 1$, S "domina a" T
  - Si $I_\epsilon(S,T) > 1$ y $I_\epsilon(T,S) > 1$, S y T son incomparables
    - Pero…¿si $I_\epsilon(S,T) < I_\epsilon(T,S)$, S es mejor que T? ¡No se puede afirmar eso!
- Inconveniente: Observemos que la evaluación solo tiene en cuenta un punto de cada conjunto!



$I_\epsilon(B,A) = X_A/X_B = 2$

$I_\epsilon(A,B) = Y_B/Y_A = 5$

¡ $I_\epsilon(A,B) > I_\epsilon(B,A)$!

¿B es mejor que A?

Optimización Multiobjetivo

Conclusiones

- Hemos formulado el problema general de la optimización multiobjetivo
- Hemos visto cómo los algoritmos evolutivos se pueden aplicar para resolver este problema
- Hemos discutido los principales factores en el diseño de MOEAs (asignación de fitness, diversidad, elitismo)
- Hemos revisado alguna de las metaheurísticas más conocidas
  - Existen muchísimos más
- Por último, hemos visto cómo evaluar el rendimiento de los MOEA y algunos ejemplos de indicadores unarios y binarios
  - Existen muchísimos más, por ejemplo, en *[Audet et al., 2021]* se describen 62 indicadores diferentes

Referencias

- C. A. Coello, D. A. Van Veldhuizen, G. B. Lamont. Evolutionary Algorithms for Multi-Objective Problems. Kluwer, NewYork, 2002
- E. Zitzler, M. Laumanns, S. Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In X. Gandibleux and others, editors, *Metaheuristics for Multiobjective Optimisation*, Lecture Notes in Economics and Mathematical Systems, Springer, 2004
- Abdullah Konak, David W. Coit, Alice E. Smith. Multi-objective optimization using genetic algorithms: A Tutorial. Reliability Engineering and System Safety, 91(9), pp. 992-1007, 2006
- Francisco Herrera. Genetic Algorithms: basic notions and some advanced topics. Talk in the course Future Directions in Soft Computing, European Centre for Soft Computing. http://sci2s.ugr.es
- E. Zitzler. Two decades of EMO: A Glance and a Look Ahead. IEEE Symposium on Computational Intelligence in Multi-Criterion Decision Making, 5 April 2007. http://www.tik.ee.ethz.ch/sop/publicationListFiles/zitz2007a.pdf
- Dimo Brockhoff. Tutorial on evolutionary multiobjective optimization. GECCO '19:Proceedings of the Genetic and Evolutionary Computation Conference Companion July 2019 Pages 461–484 https://doi.org/10.1145/3319619.3323396

Referencias

- C. A. Coello. Evolutionary Multiobjective Optimization: Current and Future Challenges. In J. Benítez, O. Cordón, 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
- 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
- C. Auden, J. Bigeon, D. Cartier, S. Le Digabel, L. Salomon. Performance indicators in multiobjective optimization. European Journal of Operational Research 292, pp. 397-422 (2021)