

Optimización multiobjetivo y algoritmos meméticos

Dra. Adriana Lara
adriana@esfm.ipn.mx

Escuela Superior de Física y Matemáticas del IPN

Escuela de Cómputo Evolutivo ECE18,
CIMAT 18/octubre/2018

Algoritmos Meméticos

El físico no lo es todo...

Genes & memes

La sesión de hoy:

El problema multiobjetivo

Herramientas de Programación Matemática

MOEAs

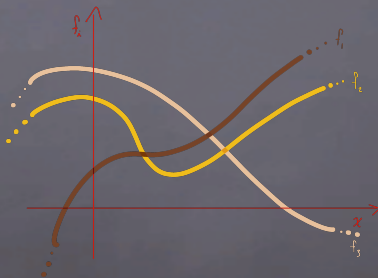
Optimización se refiere a...

Un conjunto de técnicas para buscar soluciones a problemas que involucran la utilización de recurso limitados:

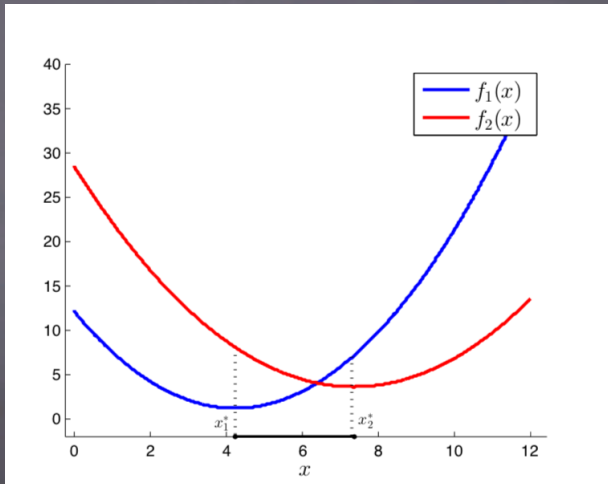
- Tiempo
- Memoria
- Personal
- Materiales
- Equipo
- Dinero...

Es común en problemas prácticos que se presente más de un criterio (función) a optimizar a la vez. Por ejemplo:

- Ganancias & riesgo (finanzas)
- Precio & calidad
- Tiempo & costos
- Diferentes propiedades físicas en diseño de partes en ingeniería
- Etc...



Cuando los valores de estos criterios se encuentran en conflicto, se dice que se tiene un problema de optimización multiobjetivo.



El *problema de optimización multiobjetivo* (MOP) se define formalmente como:

$$\text{Minimizar } F(x) := [f_1(x), f_2(x), \dots, f_k(x)]^T \quad (1)$$

sujeto a:

$$g_i(x) \leq 0 \quad i = 1, 2, \dots, m$$

$$h_i(x) = 0 \quad i = 1, 2, \dots, p$$

donde $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ es el vector de *variables de decision* (o parámetros).

Las funciones componente $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, k$ son las *funciones objetivo* (o simplemente *objetivos*) del problema; y las funciones $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, $j = 1, \dots, p$ se conocen como las *restricciones* del problema.

Si dichas funciones g_i y h_i no están presentes, se habla de un MOP *sin restricciones*.

Notemos que no se pierde generalidad al definir todas las funciones como funciones de minimización.

Recordemos que minimizar f_i es equivalente a maximizar $-f_i$.

Aquí se dió la definición de un MOP continuo, pero el *dominio* de las funciones objetivo puede ser discreto también; la definición dada y la mayoría de las cosas dichas en este curso siguen siendo válidas en estos casos.

Optimalidad en multiobjetivo:

Definición

Dados dos vectores $x, y \in \mathbb{R}^n$, decimos que x **domina** a y (denotado como $x \preceq y$) si

$$f_i(x) \leq f_i(y)$$

para $i = 1, \dots, k$, y $F(x) \neq F(y)$.

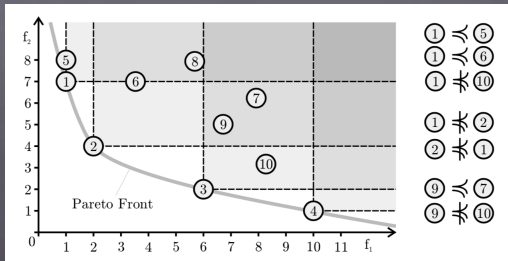
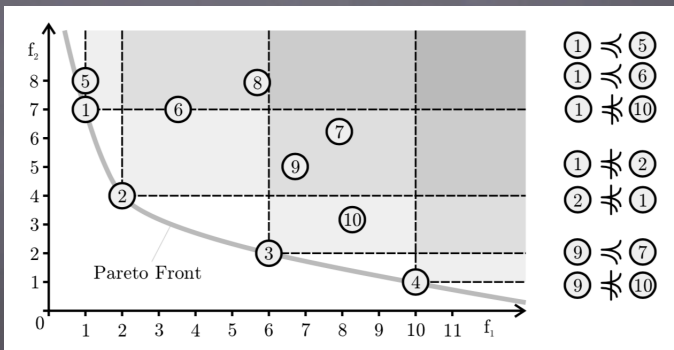


Imagen tomada de [Weise et al., 2009]

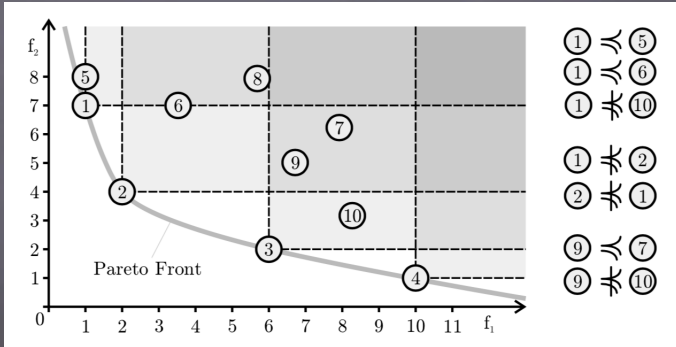
Definición

Decimos que un vector de variables de decisión $x \in \mathcal{X} \subseteq \mathbb{R}^n$ es **no-dominado** con respecto a $\mathcal{X}$ (donde $\mathcal{X}$ es la región factible), si no existe $x' \in \mathcal{X}$ tal que $x' \preceq x$.



Definición

Decimos que el vector de variables de decisión $x^* \in \mathcal{X} \subset \mathbf{R}^n$ es un **óptimo de Pareto** si es no-dominado con respecto a $\mathcal{X}$.



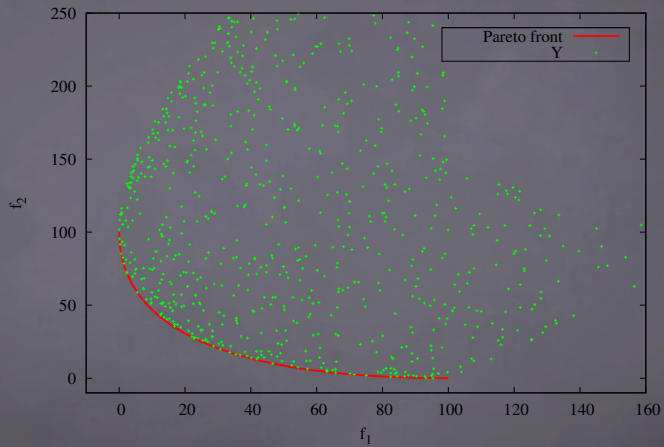
Definición

a) El conjunto de óptimos de Pareto, o **Conjunto de Pareto** $\mathcal{P}^*$ se define como:

$$\mathcal{P}^* = \{x \in \mathcal{X} \mid x \text{ es un óptimo de Pareto}\}.$$

b) Su conjunto imagen, conocido como el **Frente de Pareto** y denotado por $\mathcal{PF}^*$ se define como:

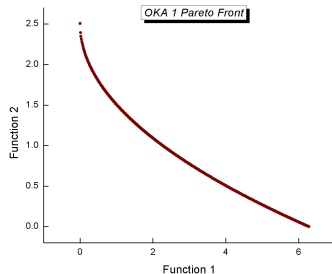
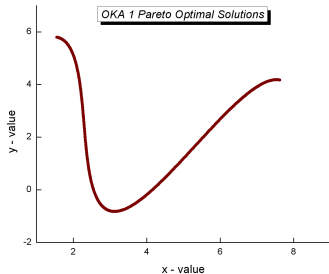
$$\mathcal{PF}^* = \{F(x) \in \mathbb{R}^k \mid x \in \mathcal{P}^*\}.$$

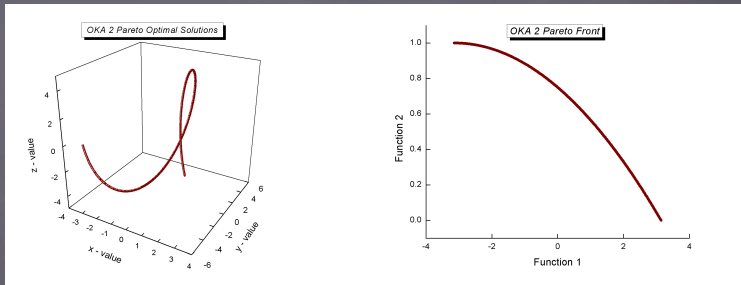


La solución de un MOP como el aquí descrito (bajo ciertas condiciones) resulta ser una superficie de dimensión $k - 1$.
Para ver los detalles puede consultarse
[Hillermeier, 2001, Johannes, 1986]

Los siguientes son ejemplos de conjuntos óptimos y sus imágenes. Para poder visualizarlos, se presentan problemas de dos y tres variables.

Se prefiere visualizar el frente de Pareto pues es generalmente de menor dimensión que el espacio de las variables. Además, para el usuario tiene principal relevancia el espacio de los objetivos.





Imágenes tomadas de [Coello Coello et al., 2007]

Ejercicio

- *Escribir un algoritmo que dados n vectores en $\mathbb{R}^k$, simulando ser los valores de F , encuentre los vectores asociados a puntos no dominados.*
- *¿Cómo se comporta su algoritmo conforme crece n ?*
- *¿Será posible construir un algoritmo para esto que tenga orden menor a kn^2 ?*

Herramientas de Programación Matemática

Regularmente estas técnicas requieren de suposiciones fuertes acerca de las funciones y dominios (espacios de búsqueda) para su aplicación. Por ejemplo, continuidad o diferenciabilidad.

Son costosas al requerir información de primer y segundo orden.

Cuentan con sustento matemático para predecir su comportamiento.

Son técnicas puntuales por lo que al final de su ejecución se obtendrá la aproximación a un punto del conjunto de Pareto; sin embargo pueden ejecutarse diferentes configuraciones y obtener, en ciertos casos, diferentes puntos.

Métodos de escalarización:

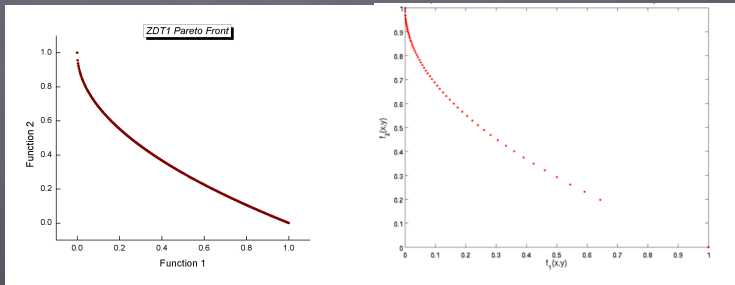
Estas técnicas se han utilizado ampliamente dentro de la Investigación de Operaciones, nos referiremos a ellos como métodos *clásicos* o *tradicionales*. Los métodos que así se conocen consisten en transformar el problema multiobjetivo en un problema de un solo objetivo. Esto puede hacerse combinando las funciones; por ejemplo, mediante el uso de funciones agregativas

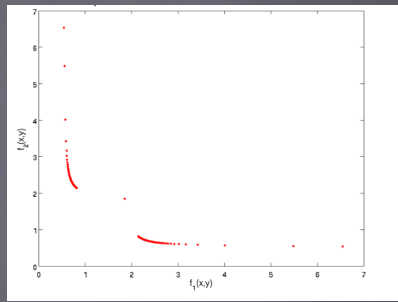
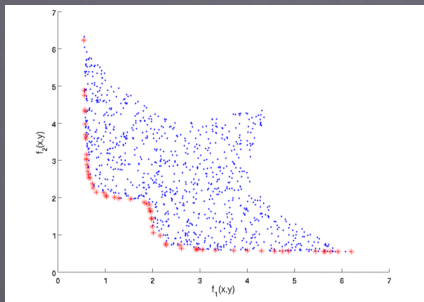
$$\text{Minimizar } g_{\omega} = \sum_{i=0}^k \omega_i f_i(x)$$

$$\text{sujeto a: } 0 < \omega_i \text{ para todo } i \in \{1, \dots, k\};$$

$$\text{y } \sum_{i=0}^k \omega_i = 1; x \in S.$$

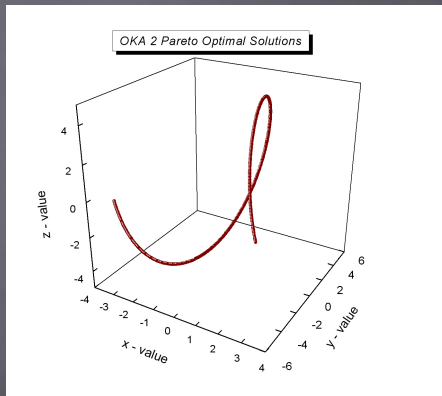
También puede hacerse replanteando el problema mediante herramientas geométricas, como lo hace el método NBI [Das and Dennis, 1998]. Existen diversas propuestas para este tipo de replanteamientos (ver [Eichfelder, 2008]), una de las más útiles y que generaliza a varias de ellas es la propuesta de Pascoletti-Serafini [Pascoletti and Serafini, 1984].





Métodos de continuación numérica

De estos métodos se hablará en la plática de mañana (Dr. Oliver Schütze). Consisten en ir construyendo porciones de curvas solución mediante predicciones (linearizaciones) y métodos correctores. Pueden consultarse [Allgower and Georg, 1990, Schütze et al., 2009a] para una introducción concisa. También dejamos fuera de este curso el popular método de la restricción ϵ .



Métodos con direcciones de descenso

Desde la experiencia de la Programación Matemática para funciones de un solo objetivo, existe manera de explotar la información contenida en los gradientes de **todas**, y cada una de, las funciones objetivo para construir lo que llamaremos **direcciones de descenso multiobjetivo**.

Una dirección de descenso multiobjetivo es aquella hacia cual se obtiene un **decremento en todas** las funciones objetivo f_i .

Para formalizar este concepto, recordemos la definición del gradiente de una función:

$$\nabla f_i = \left[\frac{\partial f_i}{\partial x_1}, \dots, \frac{\partial f_i}{\partial x_n} \right],$$

y que la derivada direccional de f_i sobre la dirección ν se puede calcular mediante

$$\langle \nabla f_i, \nu \rangle.$$

En optimización no lineal tradicional, se ha estudiado fuertemente el concepto de direcciones de descenso, para establecer direcciones de búsqueda, e ir **de manera iterativa disminuyendo los valores de la función.**

Ejemplos de este tipo de métodos de descenso son el método de la pendiente máxima y los métodos de direcciones conjugadas [Luenberger et al., 1984].

Es bien sabido que la dirección del vector gradiente $\nabla f(x^i)$ de una función f coincide con la dirección de **máximo incremento** de ésta.

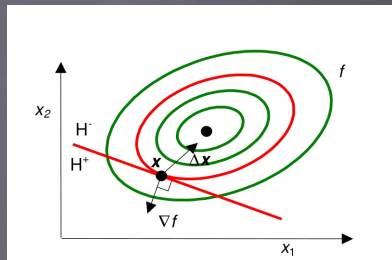
Se cumple también que la dirección $-\nabla f(x^i)$ coincide con la dirección de búsqueda de **máximo decremento** de f .

Ejercicio

Probar que la dirección $-\nabla f(x^i)$ coincide con la dirección de búsqueda de máximo decremento de f , a partir de un cierto vector x^i .

Sugerencia: recuerde la definición de las derivadas direccionales y establezca una cota.

A partir de la observación anterior, es útil pensar en todas las posibles direcciones de ascenso y descenso en términos de su relación con $\nabla f(x^i)$.

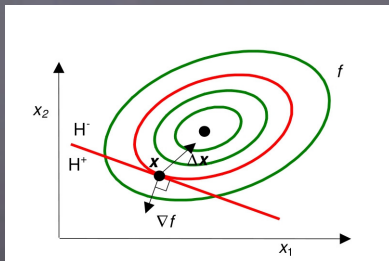


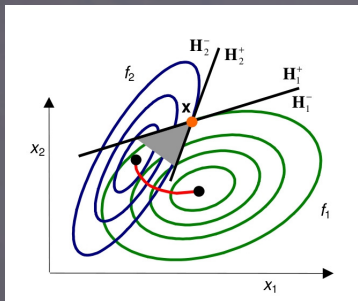
Entonces, ν es una **dirección de descenso multiobjetivo** sí y solo si ocurre que

$$\langle \nabla f_i, \nu \rangle \leq 0 \quad \text{para todo } i \in \{1, \dots, m\}$$

y

$$\langle \nabla f_j, \nu \rangle < 0 \quad \text{para algún } j \in \{1, \dots, m\}$$





Uno de los conceptos útiles para combinar la información de las derivadas direccionales de varias funciones es el de cono de descenso (introducido en [Brown and Smith, 2003]):

De manera alternativa, una dirección es de descenso multiobjetivo, sí y solo sí se encuentra en el cono de descenso definido por los hiperplanos ortogonales a los gradientes de las funciones f_i .

Respecto a estos métodos se tienen varios temas de trabajo en cuanto a costos y eficacia.

Tres aspectos son básicos para usar estas ideas como buscadores locales, y son los siguientes:

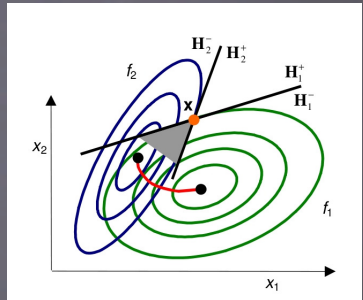
1) El cálculo de la dirección de descenso:

Existen diferentes propuestas que calculan direcciones de descenso multiobjetivo

- [Fliege and Fux Svaiter, 2000]
- [Schäffler et al., 2002, Harada et al., 2006].
- [Harada et al., 2006].
- [Lara et al., 2010]
- [Schütze et al., 2009b].

En general este cálculo implica resolver problemas lineales o cuadráticos.

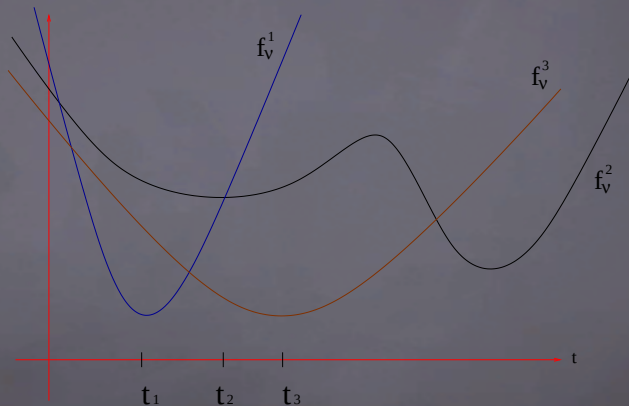
El cálculo de la dirección de descenso es de nuevo un problema multiobjetivo.



2) El costo de los gradientes:

- Diferencias finitas
- Diferenciación Automática [Griewank et al., 2002]
- Propuestas gradient-free [Schütze et al., 2016, Lara et al., 2013]

3) El cálculo adecuado del tamaño de paso:
Esto es de nuevo un problema multiobjetivo.



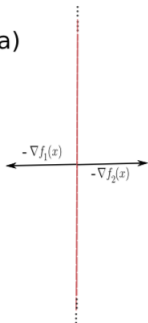
Otro aspecto interesante de estos conceptos es que se tiene un criterio de paro a la mano para detener el algoritmo de búsqueda.

Pues se sabe que una condición necesaria para que x^* sea un punto óptimo de Pareto es que

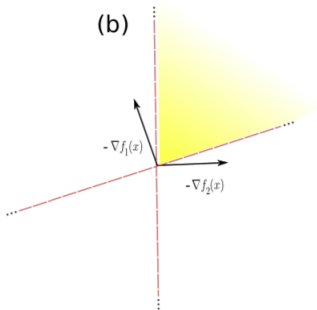
$$\sum_{i=1}^k \alpha_i \nabla f(x_i) = 0$$

$$\alpha \neq 0$$

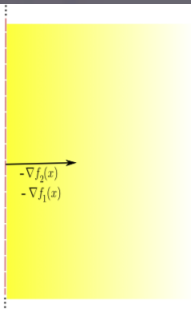
(a)



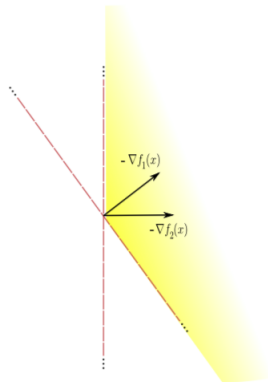
(b)



(e)



(d)



Con estas técnicas solo podemos **garantizar** la convergencia local.

Un enfoque global viene a continuación.

Algoritmos Evolutivos

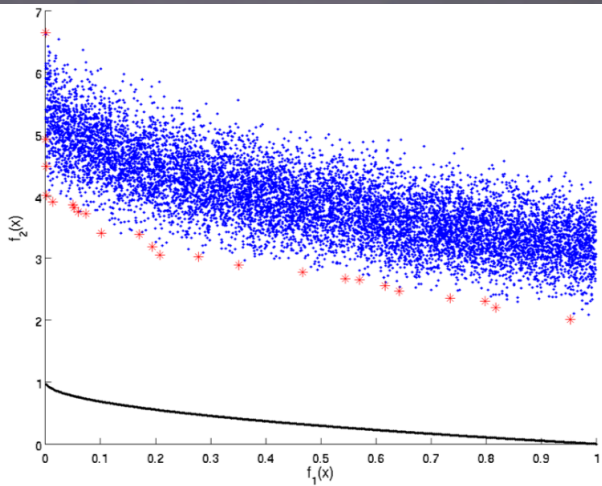
Multiobjetivo

Los algoritmos evolutivos se han usado de manera exitosa para atacar problemas de optimización multiobjetivo, dando origen al concepto de *algoritmo evolutivo multiobjetivo* (MOEA)

Algunas de las razones de su popularidad han sido:

- MOEAs son de naturaleza poblacional
- MOEAs hacen pocas suposiciones
- MOEAs son robustos

Dentro de la teoría de optimización multiobjetivo, los algoritmos evolutivos se clasifican como técnicas *a posteriori* para diferenciarse de técnicas estilo “goal programming” [Lee et al., 1972] (*a priori*) y las interactivas como el "Light beam search" [Jaszkiewicz and Słowiński, 1999].



¿Cuál es el estado de mi aplicación?

“Si no está descompuesto no lo arregles”

“cualquier ayuda es bienvenida”

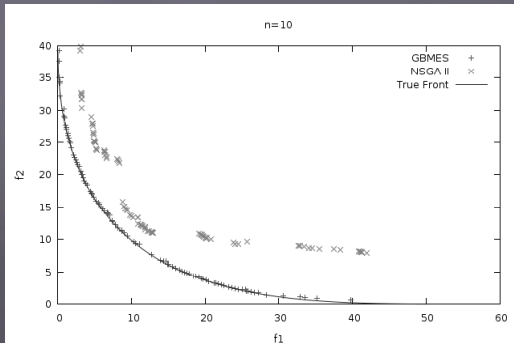
¿Cuál es el estado de mi aplicación?

¿Qué es lo que está fallando?

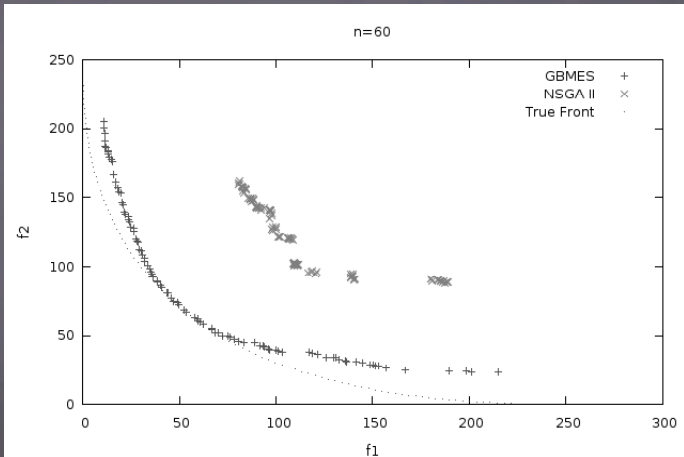
¿Qué tipo de ayuda requiere?

¿Existen métodos matemáticos/heurísticos que ya atacan estas aplicaciones/problemas?

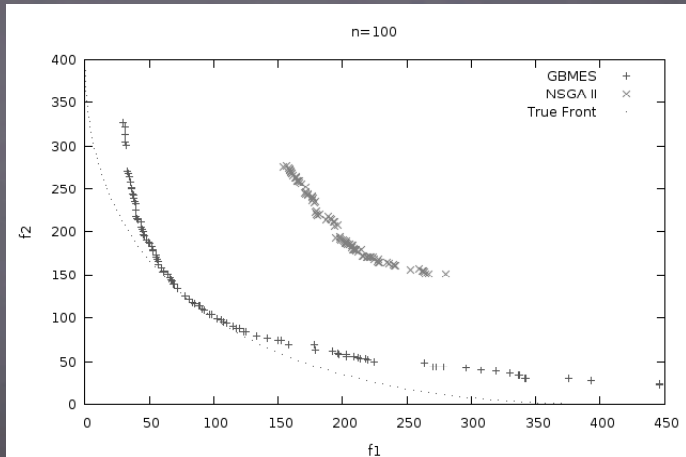
Ejemplo: acelerando la “convergencia”



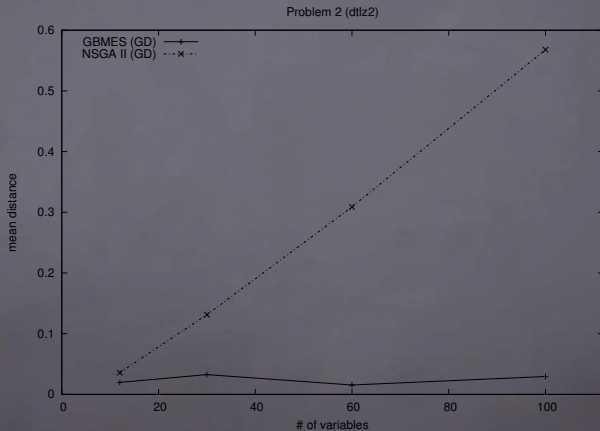
Con 60 variables

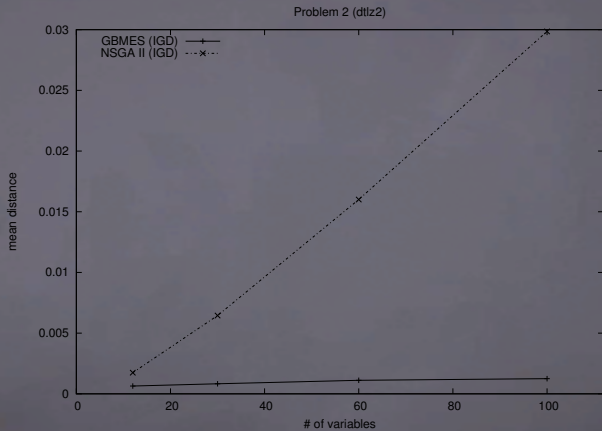


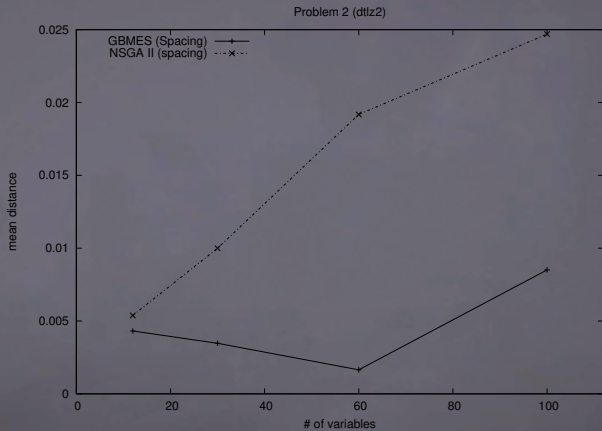
Con 100 variables



Escalabilidad







Conclusiones (del día)

Los problemas multiobjetivo son modelos actuales y útiles, hay mucho campo de aplicación, inclusive en problemas ya resueltos puede darse un enfoque multiobjetivo y así tratar de mejorar las soluciones.

La optimización multiobjetivo es un área ampliamente estudiada en programación matemática, se deben conocer y tratar de aplicar los métodos matemáticos además (antes) de utilizar heurísticas.

Si se está abierto a diferentes metodologías es posible aprovechar lo mejor de cada una. Se trata de subsanar las deficiencias de una técnica con las ventajas de la otra.

Mañana...

- Aspectos fundamentales de diseño en meméticos
- Ejemplos:
 - ▶ Refinamiento de soluciones
 - ▶ Mutación memética
 - ▶ Programación mixta

References I



Allgower, E. L. and Georg, K. (1990).
Numerical Continuation Methods.
Springer.



Brown, M. and Smith, R. (2003).
Effective use of directional information in multi-objective
evolutionary computation.
In Genetic and Evolutionary Computation – GECCO 2003,
volume Volume 2723/2003 of Lecture Notes in Computer Science,
pages 778–789. Springer Berlin / Heidelberg.



Coello Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A.
(2007).
Evolutionary Algorithms for Solving Multi-Objective Problems.
Springer, New York, second edition.
ISBN 978-0-387-33254-3.

References II



Das, I. and Dennis, J. (1998).

Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization*, 8:631–657.



Eichfelder, G. (2008).

Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Berlin Heidelberg.
ISBN 978-3-540-79157-7.



Fliege, J. and Fux Svaiter, B. (2000).

Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494.

References III



Griewank, A., Corliss, G., Faure, C., Hascoët, L., and Naumann, U., editors (2002).

Automatic Differentiation of Algorithms: From Simulation to Optimization.

Springer-Verlag New York, Inc., New York, NY, USA.



Harada, K., Sakuma, J., and Kobayashi, S. (2006).

Local Search for Multiobjective Function Optimization: Pareto Descent Method.

In GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 659–666, New York, NY, USA. ACM Press.

References IV



Hillmermeier, C. (2001).

Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach, volume 135 of International Series of Numerical Mathematics.

Birkhäuser.



Jaszkiewicz, A. and Słowiński, R. (1999).

The 'light beam search' approach—an overview of methodology applications.

European Journal of Operational Research, 113(2):300–314.



Johannes, J. (1986).

Mathematical vector optimization in partially ordered linear spaces.

Frankfurt am Main ; New York : Lang.

References V



Lara, A., Alvarado, S., Salomon, S., Avigad, G., Coello, C. A. C., and Schütze, O. (2013).

The gradient free directed search method as local search within multi-objective evolutionary algorithms.

In EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II, pages 153–168. Springer.



Lara, A., Coello Coello, C. A., and Schütze, O. (2010).

A painless gradient-assisted multi-objective memetic mechanism for solving continuous bi-objective optimization problems.

In IEEE Congress on Evolutionary Computation (CEC 2010), pages 1–8. IEEE Press.



Lee, S. M. et al. (1972).

Goal programming for decision analysis.
Auerbach Publishers Philadelphia.

References VI



Luenberger, D. G., Ye, Y., et al. (1984).
Linear and nonlinear programming, volume 2.
Springer.



Pascoletti, A. and Serafini, P. (1984).
Scalarizing vector optimization problems.
Journal of Optimization Theory and Applications, 42(4):499–524.



Schäffler, S., Schultz, R., and Weinzierl, K. (2002).
Stochastic method for the solution of unconstrained vector
optimization problems.
Journal of Optimization Theory and Applications,
114(1):209–222.



Schütze, O., Lara, A., and Coello, C. A. C. (2009a).
Evolutionary continuation methods for optimization problems.
Proceedings of Genetic and Evolutionary Computation
Conference, (GECCO 2009).

References VII



Schütze, O., Lara, A., and Coello Coello, C. A. (2009b).
The directed search method for multi-objective optimization problems.
Technical report, http://delta.cs.cinvestav.mx/~schuetze/technical_reports/index.html.



Schütze, O., Martín, A., Lara, A., Alvarado, S., Salinas, E., and Coello, C. A. C. (2016).
The directed search method for multi-objective memetic algorithms.
Computational Optimization and Applications, 63(2):305–332.



Weise, T., Zapf, M., Chiong, R., and Nebro, A. J. (2009).
Why is optimization difficult?
In Nature-Inspired Algorithms for Optimisation, pages 1–50.
Springer.