Joel Chacón Castillo - Proyecto de verano

Análisis de algoritmos multiobjetivos

1 Optimización Multi-Objetivo

Un problema de optimización multi-objetivo (MOOP) consiste en encontrar un vector que corresponde a las variables de decisión las cuales satisfacen restricciones y optimizan un vector de funciones las cuales representan las funciones objetivo.

1.1 Definición de un problema de optimización Multi-Objetivo

Un problema de optimización tiene un número de funciones objetivo las cuales deben ser minimizadas o maximizadas. Estos problemas usualmente tienen un número de restricciones en donde cualquier solucón factible debe ser cumplida. La forma general de un problema de optimización multiobjetivo (MOOP "Multiobjetive Optimization Problem") se define como:

$$\begin{array}{ccc} Minimize/Maximize & f_m(x) & m = 1, 2, ..., M; \\ subjectto & g_j(x) \geq 0 & j = 1, 2, ..., J; \\ & h_k(x) = 0 & k = 1, 2, ..., K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, 2, ..., n. \end{array} \tag{1}$$

Una solucion $\mathbf{x}$ es un vector de n variables de decisión: $\mathbf{x} = (x_1, x_2, ..., x_n)^T$. El último conjunto de restricciones son llamados como variables límite, restringiendo a cada variale de desición x_i a tomar un valor con el límite inferior como $x_i^{(L)}$y un límite superior $x_i^{(U)}$.
M es el número de objetivos f_i.
J son las restricciones de desigualdad y K son las restricciones de igualdades.
Una solución que no cumple todas las restricciones ($J+K$) y los límites, se define como *Solución no factible*, de forma contraria es una *Solución factible*.

1.2 Espacio de decisión de variables y espacio objetivo

En optimización multi-objetivo la función objetivo constituye un espacio multidimensional, donde para cada solución $\mathbf{x}$ en el espacio de decisión de variables x existe un punto en el espacio objetivo Z denotado por:

$$f(\mathbf{x}) = \mathbf{z} = (z_1, z_2, ..., z_M)^T \tag{2}$$

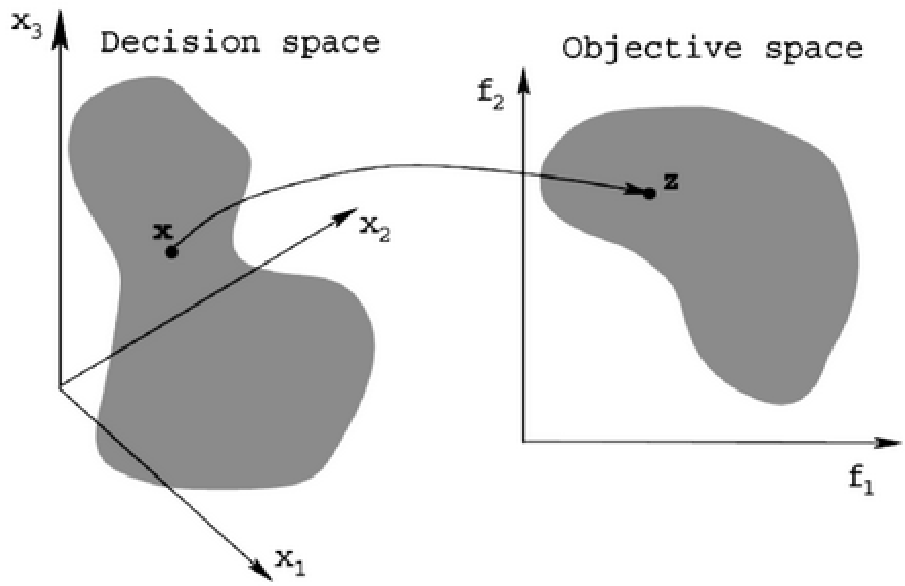

Figure 1: Representación de la variable del espacio de variable de desición y el correspondiente espacio objetivo.

1.3 Problemas de optimización Multi-Objetivo convexos y no convexos

Definicion Una función $f : \Re^n \longrightarrow \Re$ es una función convexa si para cualquier par de soluciones $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \Re^n$ la siguiente condición se cumple

$$f(\lambda x^{(1)} + (1-\lambda)x^{(2)}) \leq \lambda f(x^{(1)}) + (1-\lambda) f(x^{(2)}) \quad \forall \quad 0 \leq \lambda \leq 1 \qquad (3)$$

La definición 3 proporciona las siguientes propiedades de una función convexa.

- La aproximación lineal de $f(\mathbf{x})$ siempre es subestimada en cualquier punto del intérvalo $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$.
- La matriz Hessiana de $f(\mathbf{x})$ es definida positiva $\forall \mathbf{x}$.
- Para una función convexa, el mínimo local es siempre un mínimo global.

1.4 Objetivos en optimización multi-objetivo

Los dos objetivos en optimización multi-objetivo son:

- Encontrar un conjunto de soluciones que se encuentren muy cercanas al frente de pareto óptimo.
- Encontrar un conjunto de soluciones los más diversos que se pueda.

1.5 Optimalidad de pareto y dominancia

1.5.1 Vector Objetivo ideal

Definición El m-ésimo componente de un vector objetivo ideal $\mathbf{x}^*$ es la solución mínima restringida del siguiente problema:

$$\begin{array}{ll} Minimize & f_m(\mathbf{x}) \\ subject\ to & \mathbf{x} \in S. \end{array} \tag{4}$$

Entonces si la solución mínima para la m-ésima función objetivo es el vector de decisión $\mathbf{x}^{*(m)}$ con valor de función f_m^*, el vector ideal es:

$$\mathbf{z}^* = \mathbf{f}^* = (f_1^*, f_2^*, ..., f_M^*)^T.$$

En general el vector objetivo ideal corresponde a una solución no existente, esto es porque la solución del mínimo para cada función objetivo no necesariamente tiene la misma solución. La única forma de que esl vector objetivo ideal corresponda a una solución factible es cuando el mínimo de todas las funciones objetivo son idénticas.

1.5.2 Vector Objetivo utópico

El vector objetivo utópico denoma un array de límites inferirores para todas las funciones objetivo.
Definición. Un vector objetivo utópico $\mathbf{z}^{**}$ tiene cada uno de sus componentes marginalmente más pequeños que el vector objetivo ideal , o $\mathbf{z}_i^{**} = \mathbf{z}_i^* - \epsilon_i$ con $\epsilon_i > 0\ \forall\ i = 1, 2, ..., M$.

1.5.3 Vector Objetivo Nadir

A diferencia que el vector objetivo ideal el cual representa el límite inferior de cada objetivo, en la región factible del espacio de búsqueda, el vector objetivo nadir $\mathbf{z}^{nad}$ representa el límite superior de cada objetivo en el conjunto óptimo de pareto y no en todo el espacio de búsqueda.
El vector objetivo nadir podría representar una solución existente o no existente (dependienddo de la convexidad y continuidad del conjunto de pareto óptimo). Para normalizar cada función objetivo conociendo el vector ideal y nadir se puede implementar $f_i^{norm} = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*}$.

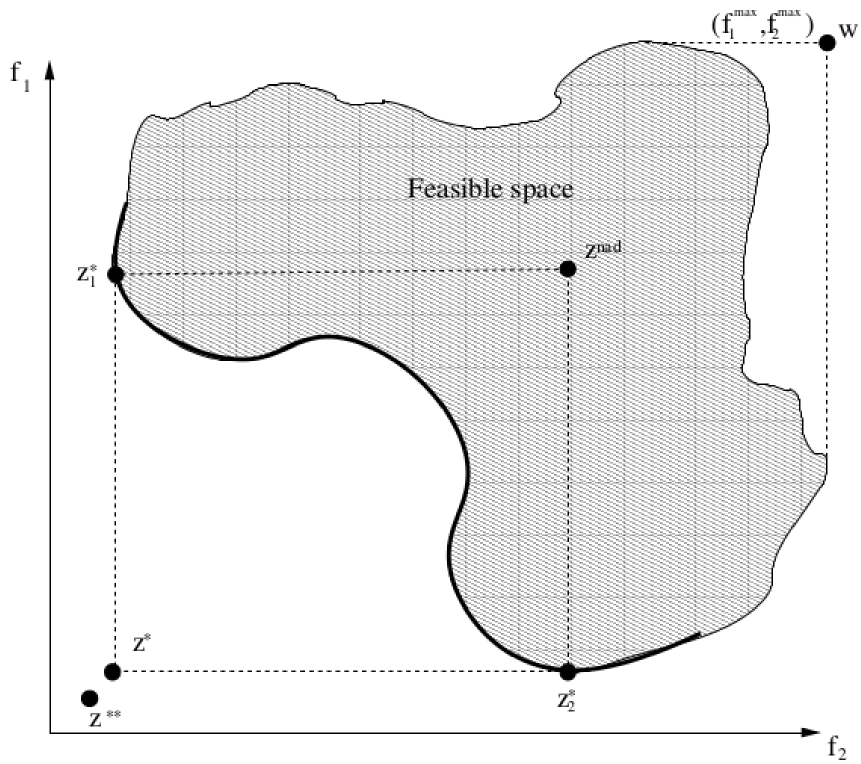

Figure 2: Vectores Ideal, Utópico y Nadir en el espacio objetivo.

1.6 Concepto de Dominación

Definición Se dice que una solución $\mathbf{x}^{(1)}$ domina a otra solución $\mathbf{x}^{(2)}$ si las siguientes condiciones son ciertas:

- La solución $\mathbf{x}^{(1)}$ no es peor que $\mathbf{x}^{(2)}$ en todos los objetivos.
- La solución $\mathbf{x}^{(1)}$ es estrictamente mejor que $\mathbf{x}^{(2)}$ en al menos un objetivo.

$$\mathbf{x}_1 \preceq \mathbf{x}_2 \;\; iff \;\; \begin{array}{cc} f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) & \forall i \in 1, ..., M \\ \exists j \in 1, ..., M & f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{array} \tag{5}$$

Si cualquiera de las condiciones es violada, entonces $\mathbf{x}_1$ no domina a la solución $\mathbf{x}_2$.

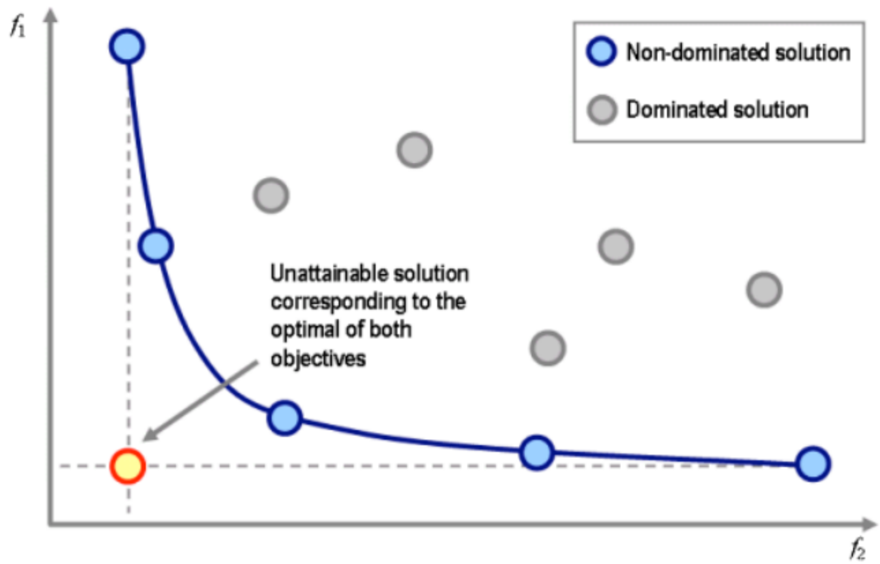

Figure 3: Ejemplo de la relación de dominancia.

1.6.1 Propiedades de la relación de dominancia

- **Reflexivo**. La relación de dominancia es no reflexiva desde que cualquier solución $\mathbf{x}^{(1)}$ no se domina a sí misma (por definición de dominancia).
- **Simétrico**. La relación de dominancia es no simétrico porque $\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)}$ no implica que $\mathbf{x}^{(2)} \preceq \mathbf{x}^{(1)}$, de hecho lo opuesto es verdadero, si $\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)}$ entonces $\mathbf{x}^{(2)} \npreceq \mathbf{x}^{(1)}$.
- **Antisimétrico**. Desde que la relación de dominancia es no simétrica entonces no puede ser antisimétrica.
- **Transitiva**. La relación de dominancia es transitiva, si $\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)}$ y $\mathbf{x}^{(2)} \preceq \mathbf{x}^{(3)}$ entonces $\mathbf{x}^{(2)} \preceq \mathbf{x}^{(3)}$.

Otra propiedad interesante es que si una solución $\mathbf{x}^{(1)}$ no domina una solución $\mathbf{x}^{(2)}$, esto no implica que $\mathbf{x}^{(2)}$ domina $\mathbf{x}^{(1)}$ (Por ejemplo ambas pueden ser no dominadas).

1.6.2 Optimalidad de pareto

Definición (Conjunto no dominado). Entre un conjunto de soluciones **P**, el conjunto no dominado de soluciones **P**' son las que no son dominadas por ningún miembro del conjunto **P**.

Definición (Conjunto global del pareto óptimo). El conjunto no dominado de todo el espacio de búsqueda factible S el el conjunto global de pareto óptimo.

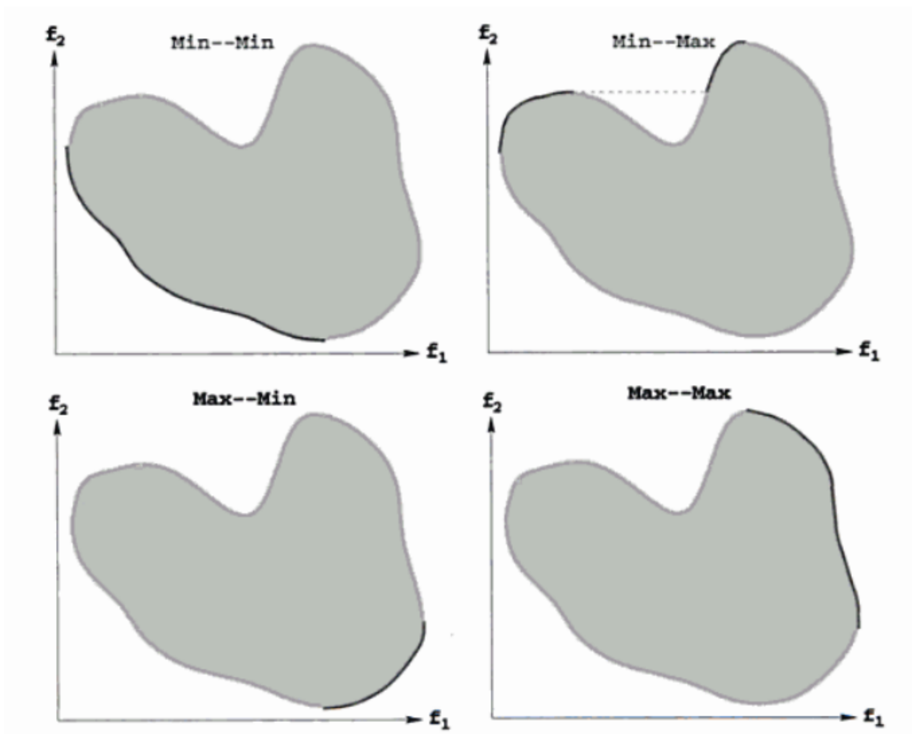

Figure 4: Las soluciones de pareto optimas son marcadas con cruvas continuas para cuatro combinaciones de dos tipos de función objetivo.

1.6.3 Dominancia fuerte y débil del pareto óptimo

La definición de dominancia es usualmente conocida como relación de dominancia débil.

Definición (Dominancia Fuerte). Una solución $\mathbf{x}^{(1)}$ domina fuertemente a una solución $\mathbf{x}^{(2)}$, $(\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)})$, si la solución $\mathbf{x}^{(1)}$ es estrictamente mejor que la solución $\mathbf{x}^{(2)}$ en todos los M objetivos.

Definición (Conjunto débil no dominado). Entre un conjunto de soluciones **P**, el conjunto débilmente no dominado de soluciones **P**' son las que no son fuertemente dominadas por algún otro miembro del conjunto **P**.

2 Algoritmo NSGA II

Algorithm 1 Main Loop

Initialize Pool(P_1)
f = fast-non-dominated-sort(P_1)
Q_1 = make-new-pop(P_1)
for each generation t **do**
    $R_t = P_t \cup Q_t$
    f = fast-non-dominated-sort(R_t)
    $P_{t+1} = \emptyset$ and $i = 1$
    while $|P_{t+1}| + |f_i| \leq N$ **do**
        crowding-distance-assignment(f_i).
        $P_{t+1} \cup f_i$
        $i = i + 1$
    sort($f_i, \prec_n$)
    $P_{t+1} = P_{t+1} \cup f_i[1 : (N - |P_{t+1}|)]$
    Q_{t+1} = make-new-pop(P_{t+1})
    $t = t + 1$

Algorithm 2 fast-non-dominated-sort

Inicialización de las partículas
for each $p \in P$ **do**
    $S_p = \emptyset$
    $n_p = 0$
    for each $q\ inP$ **do**
        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**
        $p_{rank} = 1$
        $f_1 = f_1 \cup \{p\}$
$i = 1$
while $f_i \neq \emptyset$ **do**
    $Q = \emptyset$
    for each $p \in f_i$ **do**
        for each $q \in S_p$ **do**
            $n_q = n_q - 1$
            if $n_q = 0$ **then**
                $q_{rank} = i + 1$
                $Q = Q \cup \{q\}$
    $i = i + 1$
    $f_i = Q$

Algorithm 3 crowding-distance-assignment

```
l = |I|
for each i do
    set I[i]_distance = 0
for each objective m do
    I = sort(I, m)
    I[1]_distance = I[l]_distance = inf
    for i=2 to (l − 1) do
        I[i]_distance = I[i]_distance + (I[i + 1].m − I[i − 1].m)/(f_m^max − f_m^min)
```