



imus
Instituto de Matemáticas de la
Universidad de Sevilla

Programa de Doctorado “Matemáticas”

PHD DISSERTATION

**MÉTODOS ALGEBRAICOS BASADOS
EN TEST-SETS PARA OPTIMIZACIÓN
LINEAL ENTERA MULTIOBJETIVO**

Author

Haydee Jiménez Tafur

Supervisors

Prof. Dr. José María Ucha Enríquez

Prof. Dr. María Isabel Hartillo Hermoso

May 31, 2017

Métodos algebraicos basados en test-sets para optimización lineal entera multiobjetivo

Memoria presentada por Haydee Jiménez Tafur
para optar al grado de Doctora en Matemáticas
por la Universidad de Sevilla

Bajo la supervisión de:

José María Ucha Enríquez
Profesor contratado doctor
Dpto. Matemática Aplicada I
Universidad de Sevilla

María Isabel Hartillo Hermoso
Profesora contratada doctora
Dpto. Matemática Aplicada I
Universidad de Sevilla

Programa de Doctorado Matemáticas

Sevilla, Mayo 31 de 2017

A mis padres, Marina y Santos,
y a mi esposo, Carlos,
mis grandes pilares en esta travesía.

Agradecimientos

Todos estos años entre Colombia y España, han sido una experiencia académica muy enriquecedora, entrar en el mundo de la investigación y vivir sus diferentes etapas, pero sobre todo han sido una gran experiencia de vida, pues me dieron la oportunidad de conocer diferentes lugares, costumbres y personas maravillosas. Por eso quiero agradecer inicialmente a los principales responsables de todo esto: a mis padres, por su amor, por su entrega, su apoyo constante en todos los proyectos que he emprendido, en particular este; y a mi esposo, mi amor, mi amigo, quien se cruzó en mi camino en el momento indicado (así a él a veces le parece un poco tarde) y se animó a compartir su vida conmigo, le agradezco todo su amor, su apoyo, sus detalles, paciencia y fortaleza en esta etapa llena de vicisitudes.

Agradezco a mis tutores José María y Maribel por darme la oportunidad de trabajar con ustedes, por animarme a caminar entre dos mundos: el álgebra y la optimización. He quedado motivada con los vínculos que hemos encontrado, gracias por plantar esa semilla. Gracias también por el tiempo que me han dado, por los muchos trámites que me ayudaron a gestionar y por dejarme conocer y compartir con ustedes un poco más que la academia.

No puedo dejar de mencionar a mis hermanos, familia y amigos que han estado pendientes de este proceso, deseándome lo mejor. Quiero resaltar en particular a Neelima y Marek, que siempre tienen el consejo indicado en el momento indicado, con los que hemos compartido tantas experiencias y de los que he aprendido mucho. Me siento afortunada de tenerlos como amigos.

En mi paso por Sevilla, tuve la oportunidad de estar inicialmente en el departamento de álgebra y luego en matemática aplicada I. En estos lugares conocí personas que me recibieron con la mejor disposición e hicieron agradable mi estancia.

En el departamento de álgebra quiero agradecer a Lola por toda la ayuda administrativa, al equipo directivo y a Francisco Castro por su apoyo académico y económico. A los profesores y compañeros por las conversaciones alrededor de una taza de café y uno que otro pastelito. Gracias a Helena, Jorge, Alberto y Marithania por los tiempos compartidos y por supuesto a mis dos compañeras de despacho: Guadalupe gracias por ser un gran apoyo en mi primera estancia en Sevilla; y Laura gracias por ofrecerme tu amistad, por tu generosidad y todo lo compartido dentro y fuera de la Universidad. Gracias por permitirme contar contigo.

En el departamento de matemática aplicada I quiero agradecer a los profesores y administrativos por su acogida y en especial a mi compañero de despacho Juan Carlos, por permitirme compartir su espacio y tratarme como si nos conociéramos de antes, por romper la monotonía con sus historias y datos curiosos, gracias por contagiarme de esa buena energía. Gracias a Luisa y Manuel Jesús por todas sus atenciones y amabilidad.

También quiero mencionar a Teresa, quien estuvo en una estancia corta en el Imus, y agradecerle por todo el tiempo que compartimos en Sevilla, por confiar en mí y ofrecerme su amistad, por los planes de cada sábado recorriendo un rinconcito de Sevilla y por ser la mejor guía turística de Colonia.

Gracias a todos los que me han apoyado en este recorrido.

Índice general

Introducción	XI
1. Optimización multiobjetivo	1
1.1. Introducción	1
1.2. Optimalidad Pareto	6
1.3. Cotas para la frontera de Pareto	10
1.4. Métodos para resolver problemas de optimización multiobjetivo	13
1.4.1. Métodos a priori	14
1.4.2. Métodos a posteriori	14
1.4.3. Métodos interactivos	19
2. Programación entera desde un punto de vista algebraico	21
2.1. Introducción	21
2.2. Test set	22
2.3. Ideales tóricos	26
2.4. Cálculo de \mathcal{T}_{\succ_c} con bases de Gröbner	29
2.5. Un ejemplo motivador	33
3. Test-sets para resolver problemas de optimización lineal entera biobjetivo	41
3.1. Motivación: el problema BBV	41
3.2. Caraterización de soluciones eficientes	43
3.3. Algoritmo para el caso biobjetivo	46
3.4. Cálculo teórico de los test-sets para el problema BBV	50
3.5. Resultados computacionales	57

3.5.1. Test-sets vs. AUGMECON2	58
3.5.2. Problema de la mochila caso binario	61
3.5.3. Límites de mochila no acotada tipo A	61
3.5.4. Mochila no acotada vs. Figueira	61
4. Test-sets para resolver problemas de optimización lineal entera multiobjetivo	65
4.1. Motivación: un problema RAP para sistemas serie-paralelo . .	66
4.2. Caraterización de soluciones eficientes	68
4.3. Algoritmo para el caso multiobjetivo	73
4.4. Algunos ejemplos de problemas con tres objetivos	77
4.5. Aplicación del método en RAP triobjetivo	86
4.6. Resultados Computacionales	88
Bibliografía	95

Introducción

La toma de decisiones en los problemas de la vida real rara vez depende de un único criterio sino de varios que suelen estar en conflicto. Podemos intentar reunir estos criterios en una sola función objetivo que los pondere, pero es bien sabido que esta simplificación muchas veces no es capaz de reflejar adecuadamente la complejidad de los problemas. En esta memoria vamos a presentar nuevos algoritmos exactos para tratar problemas multiobjetivo lineales enteros. Es decir, todas las funciones, objetivo y restricciones son lineales y todas las variables toman valores naturales (esto ocurre siempre que dichas variables reflejan aspectos combinatorios del problema).

El problema de la optimización multiobjetivo ha sido tratado de manera general en muchos manuales y trabajos como, por ejemplo, [15], [23], [42], [47] o [28]. El caso entero aparece tratado de manera particular en [61], [25] o [24].

El interés de estos problemas reside desde luego tanto en su parte teórica como en sus aplicaciones, entre las que podemos destacar el diseño de horarios con flujo [32], diseño de redes de transporte [21], diseño de rutas de transporte [26, 34], análisis de presupuestos [53, 39, 36], decisiones de inversión [56, 2, 5] o remedio de condiciones de polución [33].

Suele situarse el comienzo de la optimización multiobjetivo de alguna manera en el Manual de Economía Política [50] del economista italiano Vilfredo Pareto en el que se introduce explícitamente una manera de tratar varios objetivos en conflicto. Dada una asignación inicial de bienes entre un conjunto de individuos, un cambio hacia una nueva asignación que al menos mejora la situación de un individuo sin hacer que empeore la situación de los demás se denomina mejora de ofelimidad (del griego $\omega\varphi\epsilon\lambda\iota\mu\nu$, beneficioso). Una asignación se define como eficiente u óptima cuando no pueden lograrse nuevas mejoras de ofelimidad (que en realidad es casi un sinónimo de la palabra utilidad, más ampliamente usada en economía). Pareto desarrolló para in-

troducir este concepto los diagramas del economista británico Francis Ysidro Edgeworth. Con todo, aunque su interés crece con su aparición en el contexto de la economía, el problema de encontrar los óptimos en un conjunto parcialmente es un problema clásico tratado por Cantor [12] o Cayley [14].

El problema multiobjetivo lineal entero es un NP-duro y #P-duros (veáse por ejemplo [25]), pero la existencia de métodos exactos es de gran interés, ya que sirven para comprobar la precisión y el alcance de los métodos aproximados. Dentro de los métodos exactos generales para cualquier número de objetivos podemos citar [49], [40], [55], [41], [46], [60] o [35]. Existen muchos otros trabajos para sólo dos objetivos o para problemas específicos como [11], [51] y [52].

Dentro de los métodos exactos, nuestro trabajo se enmarca concretamente en la familia de métodos algebraicos. Entre ellos, cabe mencionar [8] que propone las bases de Gröbner parciales, un objeto combinatorio similar a las bases de Graver. En [9] se utilizan ideas de [7] para proponer un método general para problemas polinómicos discretos. De gran interés es también el trabajo de [4] que utiliza las funciones generatrices para contar puntos enteros dentro de un poliedro. En [19], por último, se propone un algoritmo para problemas multiobjetivo lineales enteros polinomial en tiempo, fijados la dimensión y el número de objetivos.

En el capítulo 1 mostramos una breve introducción a los problemas multiobjetivo, a los conceptos principales que aparecen en su estudio (las *soluciones eficientes* que forman el conjunto eficiente en el espacio de variables y los *puntos no dominados* que forman la frontera de Pareto en el espacio de las funciones objetivo a optimizar) y a algunos métodos clásicos para resolverlos. Apuntamos en los ejemplos algunas dificultades que se pueden presentar al tratar problemas multiobjetivo, como son la aparición de puntos débilmente eficientes o la necesidad del punto *nadir* para más de dos objetivos. Son problemas que el método que presentamos resuelven de una manera novedosa.

En el capítulo 2 recordamos los resultados algebraicos que fundamentan el ingrediente algebraico básico en el que se basa nuestro método: los *test-sets* asociados a un problema lineal entero $\min\{cx, \text{s.a. } Ax = b, x \in \mathbb{N}^n\}$, que son válidos para todo b . Esta característica hace natural el uso de los test-sets para aplicar el método clásico de las ϵ -restricciones. Los test-sets serán calculados con bases de Gröbner respecto de órdenes que, elegidos convenientemente, producen ventajas añadidas. Presentamos al final de este capítulo un ejemplo en el que adelantamos cuáles son las ventajas de nuestro método para el caso biobjetivo.

En el capítulo 3 presentamos los resultados teóricos que sustentan el algoritmo para el caso biobjetivo lineal entero. Aunque en sentido estricto el algoritmo es un caso particular del que aparece en el capítulo 4, el caso biob-

jetivo presenta características absolutamente singulares. Demostramos que nuestro algoritmo calcula todos los puntos no dominados (y una solución eficiente para cada uno) sin resolver ningún problema de un sólo objetivo innecesario y sin producir soluciones débilmente eficientes. Además, mostramos una ventaja adicional del uso de los test-sets: en algunas familias de problemas pueden ser calculados teóricamente a priori. Hacemos precisamente esto con un problema de la literatura, el problema BBV, que presenta un número de soluciones eficientes lineal en el número de variables pero que necesita para el método clásico de las ϵ -restricciones un número exponencial de problemas de un solo objetivo. Terminamos el capítulo con varias tablas de resultados computacionales en las que mostramos que nuestro método se adapta especialmente bien al problema de la mochila no acotada.

En el capítulo 4 tratamos el caso de cualquier número de objetivos que, como es bien sabido, presenta siempre un salto de complejidad al pasar de dos a tres objetivos. En nuestro caso conseguimos un algoritmo que produce todos los puntos no dominados y sólo puntos no dominados, pero sin poder asegurar que no se resolverá algún problema de un sólo objetivo innecesario al aplicar el método de las ϵ -restricciones. Como aplicación del algoritmo resolvemos un problema de la literatura de tres objetivos en sistemas serie-paralelo. Las tablas que mostramos al final de este capítulo muestran los resultados al tratar el problema de la mochila no acotada para 3, 4 y 5 objetivos con un número de variables que no se ha alcanzado para el caso binario en la literatura.

CAPÍTULO 1

Optimización multiobjetivo

1.1. Introducción

En la vida cotidiana nos enfrentamos a situaciones en las que debemos tomar una decisión de entre varias posibilidades, esto es, seleccionar lo “mejor” de un conjunto de “alternativas” donde asumimos la existencia de ciertos criterios que nos permiten establecer la calidad de las opciones.

Los problemas de optimización no son más que un modelo matemático de problemas de decisión. Veamos algunos ejemplos de problemas de decisión:

Ejemplo 1.1. [23] Queremos comprar un carro y hemos considerado cuatro modelos: un VW Golf, un Opel Astra, un Ford Focus y un Toyota Corolla. Debemos decidir de acuerdo al precio, consumo de gasolina y potencia. Las características de cada carro están consignadas en la tabla (1.1).

		Alternativas			
		VW	Opel	Ford	Toyota
	Precio (1000 euros)	16.2	14.9	14.0	15.2
Criterios	Consumo ($l/100$ Km)	7.2	7.0	7.5	8.2
	Potencia (kW)	66.0	62.0	55.0	71.0

Tabla 1.1: Criterios y alternativas del ejemplo 1.1.

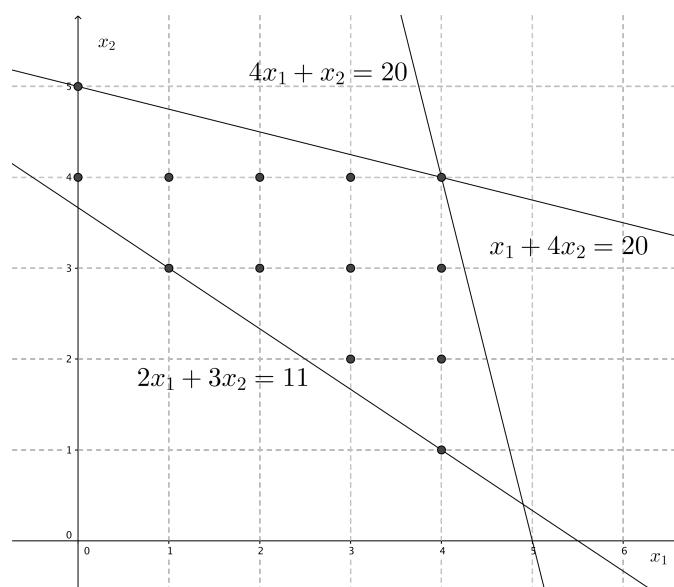


Figura 1.1: Los trece puntos representan el conjunto de alternativas del ejemplo 1.2.

Como preferimos un carro barato, potente y con bajo consumo de gasolina, nos enfrentamos a un problema de decisión con cuatro alternativas y tres criterios, pero cómo podemos decidir cuál es la “mejor” alternativa si el Toyota, que es el carro más potente (71.0 kW) es el que consume más gasolina (8.2 l/100 Km) y el Ford, que es el caro más barato (14 mil euros) es el menos potente (55.0 kW). Luego no podemos comprar un carro que sea tanto barato, como potente y con un consumo de gasolina eficiente. Sin embargo, si consideramos cada uno de los criterios por separado, la elección sería fácil.

Ejemplo 1.2. Consideremos un problema matemático con dos criterios: minimizar simultáneamente las funciones $f_1(x) = x_1$ y $f_2(x) = x_2$ sobre la región determinada por $2x_1 + 3x_2 \geq 11$, $x_1 + 4x_2 \leq 20$ y $4x_1 + x_2 \leq 20$ con $x \geq 0$ en \mathbb{Z}^2 . (Figura 1.1). Aunque en este problema el número de alternativas que tenemos es finito, ¿cuál sería ese mínimo? Sin embargo, si consideramos las funciones individualmente, el problema de decisión es fácil: los puntos (0, 4) y (0, 5) minimizan f_1 y el único punto que minimiza f_2 es (4, 1).

Ejemplo 1.3. En este ejemplo asumimos los mismos dos criterios del ejemplo 1.2 y la región determinada por las mismas restricciones pero con $x \geq 0$ en \mathbb{R}^2 (figura 1.2), lo que nos proporciona un conjunto infinito de alternativas.

Nuevamente tenemos inconveniente con decidir cuál punto corresponde a una solución en este ejemplo, pues los puntos que están en el segmento que

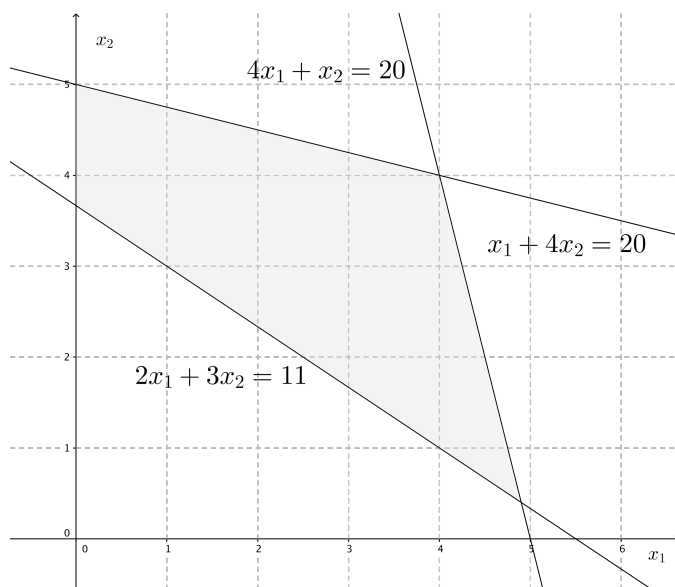


Figura 1.2: La región sombreada representa el conjunto de alternativas del ejemplo 1.3.

une el punto $(0, \frac{11}{3})$ con el punto $(0, 5)$ minimizan f_1 , mientras que el punto $(\frac{49}{10}, \frac{2}{5})$ minimiza f_2 .

Los ejemplos anteriores nos muestran que dentro del contexto de los problemas de decisión multicriterio, no está definido tan claramente qué es lo “mejor”, cuando los criterios o *funciones objetivo* están en conflicto.

Cuando establecemos las alternativas o *soluciones factibles* existentes en un problema de decisión, dicho conjunto puede tener un número contable de alternativas como en los ejemplos 1.1 y 1.2, y en este caso los llamamos *discretos*; los otros, como en el ejemplo 1.3, los llamamos *continuos*. También podemos clasificar los problemas de decisión por la forma en que se describe el conjunto de alternativas: en el ejemplo 1.1 el conjunto finito de alternativas se da de forma explícita, mientras que en los ejemplos 1.2 y 1.3 se describen de manera implícita mediante restricciones.

En la búsqueda de un concepto de solución para los problemas de decisión multicriterio, aparece en el escenario el concepto de *solución Pareto óptima*, basado en el criterio enunciado por el economista italiano Vilfredo Pareto en su libro *Manual de economía política* [50] (capítulo VI §33) a quien se le atribuye la primera referencia a dicha situación de objetivos en conflicto: se menciona que los miembros de una comunidad disfrutan, en determinada posición, de *máximo de ofelinidad* (del griego $\omega\varphi\epsilon\lambda\iota\mu\eta$, ofélimi, que significa beneficioso), cuando es imposible encontrar una manera de alejarse un poco

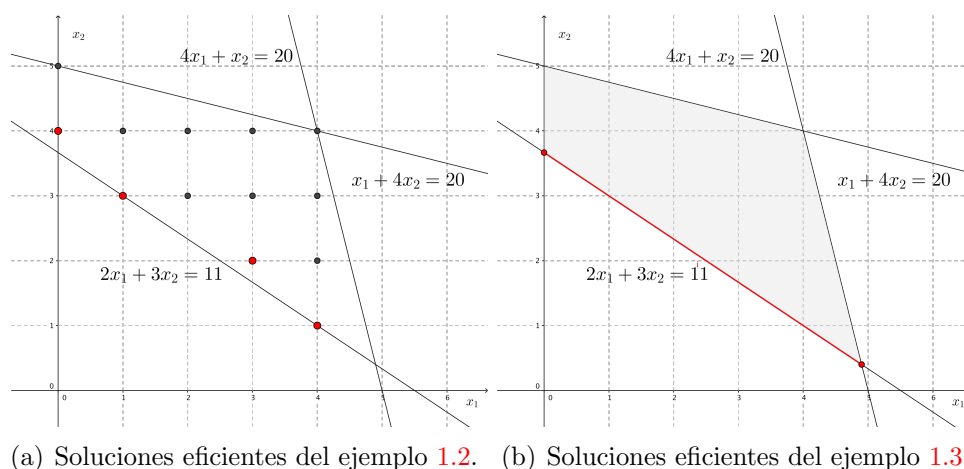


Figura 1.3

de esa posición de tal manera que la ofelinidad que cada uno de los miembros de la comunidad disfruta, aumente o disminuya.

Aplicando este concepto a los ejemplos dados, vemos que en el ejemplo 1.1 todas las alternativas son soluciones Pareto óptimas o *soluciones eficientes*, pues no hay un carro que mejore a otro en todos los criterios considerados; si comparamos el Toyota con el VW, vemos que el Toyota cuesta menos, tiene mayor potencia pero consume más gasolina, o si consideramos el Opel y el Toyota, tenemos que el Opel cuesta menos y consume menos gasolina que el Toyota, pero tiene menor potencia.

En el ejemplo 1.2 corresponden a los puntos $(0, 4)$, $(1, 3)$, $(3, 2)$ y $(4, 1)$ (figura 1.3 (a)); mientras que en el ejemplo 1.3 son los puntos que están en el segmento que une el punto $(0, \frac{11}{3})$ con el punto $(\frac{49}{10}, \frac{2}{5})$ (figura 1.3 (b)).

Al conjunto de alternativas de un problema de decisión lo llamamos *conjunto factible*. En el ejemplo 1.1 corresponde a $\mathcal{X} = \{\text{VW}, \text{Opel}, \text{Ford}, \text{Toyota}\}$ y en el ejemplo 1.2 al determinado por las restricciones dadas, $\mathcal{X} = \{x \in \mathbb{Z}^2 : 2x_1 + 3x_2 \geq 11, x_1 + 4x_2 \leq 20, 4x_1 + x_2 \leq 20, x \geq 0\}$. El espacio genérico que contiene al conjunto factible lo llamamos *espacio decisión*. En el ejemplo 1.1 puede ser *carros en el mercado*, en el ejemplo 1.2 corresponde a \mathbb{Z}^2 y en el ejemplo 1.3 a \mathbb{R}^2 .

Ahora, si en el ejemplo 1.1 consideramos solo dos criterios, digamos, el precio y el consumo de gasolina, podemos representar en \mathbb{R}^2 los valores de las alternativas según los criterios dados (figura 1.4). Estos criterios podemos expresarlos como funciones f_1 y f_2 respectivamente con $f_i : \mathcal{X} \rightarrow \mathbb{R}$ y escribir

simbólicamente el problema como:

$$\text{“mín”}_{x \in \mathcal{X}} (f_1(x), f_2(x)) \quad (1.1)$$

En la figura 1.4 podemos observar que no tiene sentido comprar un Toyota o un VW pues estos son más costosos y consumen más gasolina que el Opel; y de las alternativas que quedan, vemos que tanto el Ford como el Opel son soluciones eficientes.

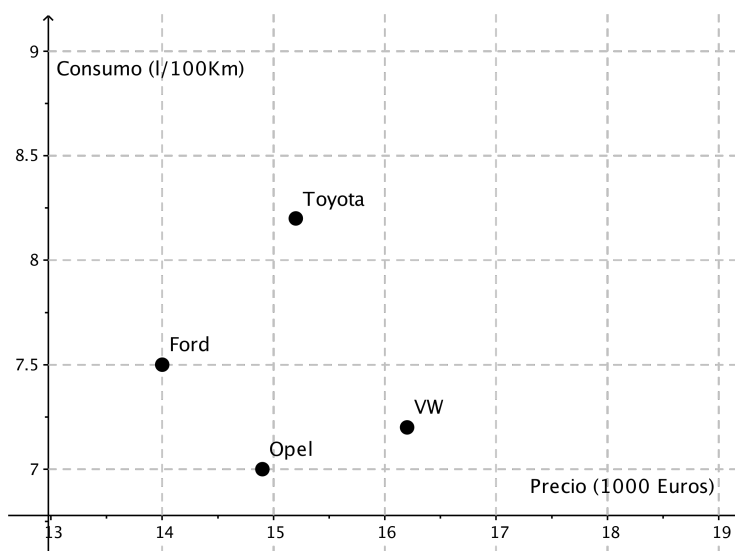


Figura 1.4: Espacio objetivo del ejemplo 1.1.

Al conjunto $\mathcal{Z} = f(\mathcal{X}) = \{y \in \mathbb{R}^2 : y = f(x) \text{ para algún } x \in \mathcal{X}\}$ de las imágenes de las alternativas bajo las funciones objetivo $f = (f_1, f_2)$, lo llamamos *conjunto objetivo* que es un subconjunto del *espacio objetivo o criterio*. En los ejemplos 1.2 y 1.3 tenemos que $\mathcal{Z} = f(\mathcal{X}) = \mathcal{X}$ y el espacio objetivo corresponde a \mathbb{Z}^2 y \mathbb{R}^2 respectivamente.

En (1.1) la palabra “mín” significa que queremos minimizar todas las funciones objetivo, sin embargo como dichas funciones, en general, están en conflicto, no podemos hablar de una única solución óptima como en el caso de problemas de optimización de un objetivo, por tanto debemos buscar una solución que represente un compromiso entre los diferentes objetivos, de manera que las componentes del vector objetivo no puedan ser simultáneamente mejoradas, que formalmente es lo que llamamos *optimalidad Pareto*.

1.2. Optimalidad Pareto

Un problema de optimización multiobjetivo es presentado como

$$\begin{aligned} \text{mín } f(x) &= (f_1(x), \dots, f_p(x)) \\ \text{s. a. } x &\in \mathcal{X} \end{aligned} \tag{1.2}$$

donde $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ son funciones objetivo, con $i = 1, \dots, p$ y $p \geq 2$. El vector decisión o solución factible $x = (x_1, x_2, \dots, x_n)$ pertenece al conjunto factible \mathcal{X} que es un subconjunto de \mathbb{R}^n definido como $\mathcal{X} = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, m\}$. La imagen de \mathcal{X} bajo f , $\mathcal{Z} = f(\mathcal{X})$, es el conjunto objetivo y $z = (z_1, \dots, z_p)$ con $z_i = f_i(x)$ para algún $x \in \mathcal{X}$ es un vector objetivo.

Todo problema de maximización es equivalente a un problema de minimización, pues se cumple que $\text{máx } f(x) = \text{mín } -f(x)$, por tanto los problemas abordados en este trabajo serán vistos como un problema de minimización.

En esta formulación general, las funciones objetivo y restricciones a utilizar pueden ser lineales o no lineales, con variables continuas donde el número de soluciones factibles puede ser infinito. En [42] se puede consultar sobre los problemas de optimización lineal multiobjetivo y en [47] sobre los problemas de optimización no lineal multiobjetivo.

Cuando en (1.2) consideramos coeficientes enteros y variables enteras no negativas, estamos hablando de la optimización discreta multiobjetivo, dentro de la que se destacan los problemas de optimización lineal entera multiobjetivo que son el objeto de estudio de este trabajo.

Definición 1.4. Una solución factible $x' \in \mathcal{X}$ es *eficiente* o *Pareto óptima* si no existe otra solución x tal que $f_i(x) \leq f_i(x')$ para $i = 1, \dots, p$ con al menos una desigualdad estricta. Si x' es eficiente, a $f(x')$ lo llamamos un *punto no dominado*. Al conjunto de todas las soluciones eficientes $x' \in \mathcal{X}$ lo notamos \mathcal{X}_E y llamamos *conjunto eficiente*. El conjunto de todos los puntos no dominados $z = f(x') \in \mathcal{Z}$, donde $x' \in \mathcal{X}_E$, lo notamos \mathcal{F} y llamamos *conjunto no dominado* o *frontera de Pareto*.

La definición de eficiencia nos conduce a la definición de una relación de dominancia de Pareto,

Definición 1.5. Si $x, x' \in \mathcal{X}$ y $f_i(x) \leq f_i(x')$ para $i = 1, \dots, p$ con al menos una desigualdad estricta, decimos que x *domina* a x' y $f(x)$ *domina* a $f(x')$.

Notemos que la definición 1.4 nos indica que para establecer cómo los vectores objetivo $(f_1(x), \dots, f_p(x))$ tienen que ser comparados para diferentes soluciones factibles $x \in \mathcal{X}$ y así obtener un significado para la expresión “min” en los problemas de optimización multiobjetivo, hemos elegido el orden \leq sobre \mathbb{R}^p definido como: dados $y = (y_1, \dots, y_p)$ y $y' = (y'_1, \dots, y'_p)$ en \mathbb{R}^p ,

$y \leq y'$ si y solo si $y_k \leq y'_k$ para todo $k = 1, \dots, p$.

Es decir, una solución factible $x' \in \mathcal{X}$ es eficiente si no existe otra solución $x \neq x'$ tal que $f(x) \leq f(x')$. En el caso del ejemplo 1.1, cuando consideramos solo los criterios precio (f_1) y consumo de gasolina (f_2), el carro Toyota no es una solución eficiente, pues para el carro Opel se cumple que $f(\text{Opel}) = (14, 2, 7, 0) \leq f(\text{Toyota}) = (15, 2, 8, 2)$.

El orden \leq en \mathbb{R}^p no es total, luego no siempre podemos comparar dos vectores objetivo, siendo esta una diferencia fundamental entre la optimización multiobjetivo y la mono-objetivo.

Si somos más restrictivos y obligamos a que las desigualdades sean estrictas en todas las componentes, obtendremos un nuevo concepto, el de débilmente eficiente.

Definición 1.6. Una solución factible $x' \in \mathcal{X}$ es *débilmente eficiente* o *débilmente Pareto óptima* si no existe otra solución x tal que $f(x) < f(x')$, es decir, $f_i(x) < f_i(x')$ para todo $i = 1, \dots, p$. A $f(x')$ lo llamamos un *punto débilmente no dominado*. Los conjuntos de las soluciones débilmente eficientes y los puntos débilmente no dominados los notamos \mathcal{X}_{wE} y \mathcal{F}_w respectivamente.

En las figuras 1.5 y 1.6 observamos que las soluciones débilmente eficientes son dominadas por las soluciones eficientes, por ejemplo en la figura 1.5 (a) vemos que tanto $(0, 4)$ como $(1, 3)$ dominan a $(1, 4)$.

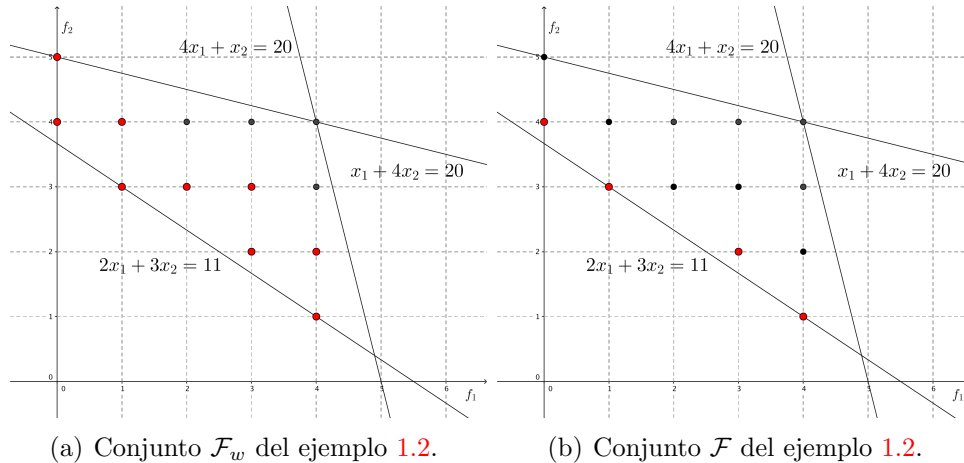


Figura 1.5

A continuación presentamos una interpretación geométrica de las definiciones 1.4 y 1.6 para caracterizar las soluciones eficientes y débilmente eficientes, representada en las figuras 1.7 y 1.8.

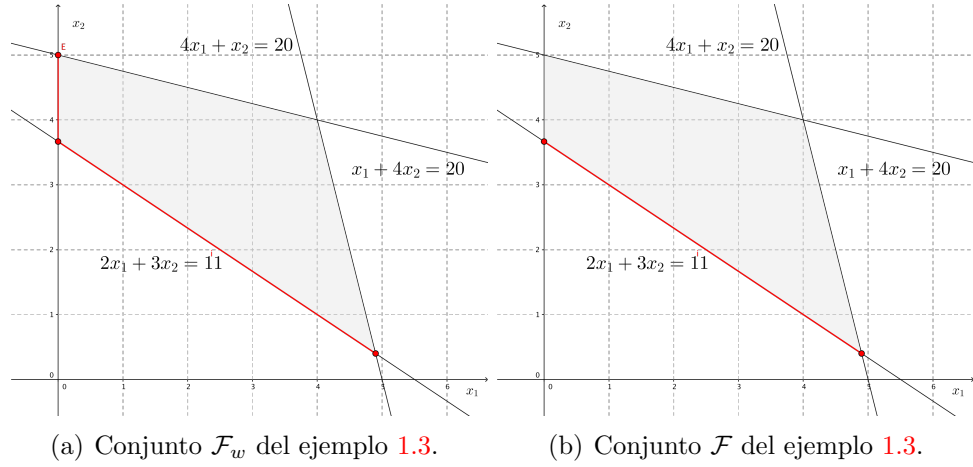


Figura 1.6

Proposición 1.7. 1. $x' \in \mathcal{X}$ es eficiente si y solo si

$$f(\mathcal{X}) \cap (f(x') - \mathbb{R}_{\leq}^p) = \{f(x')\}$$

donde $f(x') - \mathbb{R}_{\leq}^p = \{f(x') - y : y \in \mathbb{R}_{\leq}^p\}$ con $\mathbb{R}_{\leq}^p = \{y \in \mathbb{R}^p : y \geq 0\}$ (El octante no-negativo de \mathbb{R}^p).

2. $x' \in \mathcal{X}$ es débilmente eficiente si y solo si

$$f(\mathcal{X}) \cap (f(x') - \mathbb{R}_{>}^p) = \emptyset$$

donde $f(x') - \mathbb{R}_{>}^p = \{f(x') - y : y \in \mathbb{R}_{>}^p\}$ con $\mathbb{R}_{>}^p = \{y \in \mathbb{R}^p : y > 0\}$ (El octante positivo de \mathbb{R}^p).

En la figura 1.7 vemos que la solución factible $x' = (3, 3)$ del ejemplo 1.2 es débilmente eficiente pues $f(\mathcal{X}) \cap (f(x') - \mathbb{R}_{>}^p) = \emptyset$; mientras que $x = (2, 4)$ no es débilmente eficiente ya que $f(\mathcal{X}) \cap (f(x) - \mathbb{R}_{>}^p) \neq \emptyset$, el punto $(1, 3)$ está en la intersección.

La figura 1.8 nos señala que la solución factible $x' = (1, 3)$ del ejemplo 1.2 es eficiente pues $f(\mathcal{X}) \cap (f(x') - \mathbb{R}_{\leq}^p) = \{f(x')\}$; mientras que $x = (4, 2)$ no lo es ya que la intersección $f(\mathcal{X}) \cap (f(x) - \mathbb{R}_{\leq}^p)$ tiene dos puntos diferentes de $f(x)$.

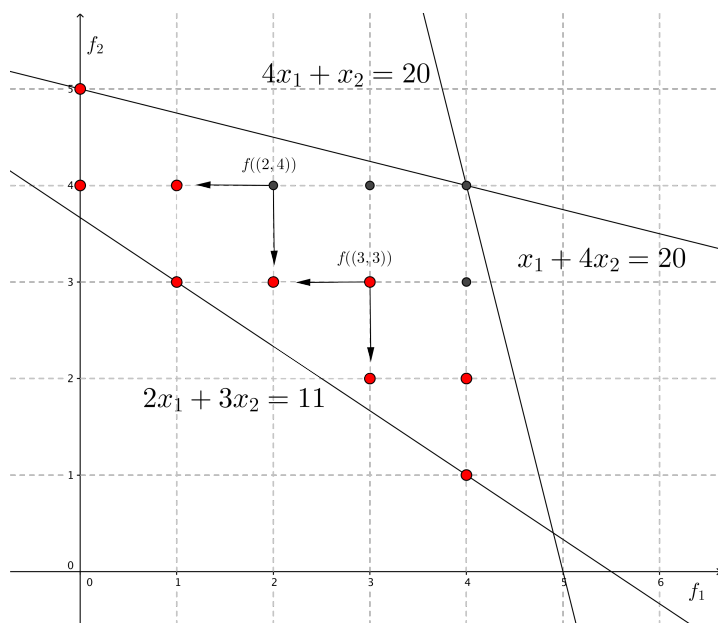


Figura 1.7: Interpretación geométrica de solución débilmente eficiente en el ejemplo 1.2.

Cuando abordamos un problema en matemáticas, nuestro principal objetivo es garantizar que existe una solución. En el caso de los problemas de optimización multiobjetivo, [23] presenta un teorema donde se mencionan las condiciones que garantizan la existencia de soluciones eficientes, que resulta ser una extensión del teorema de Weierstrass [43, p. 40] para problemas de optimización mono-objetivo:

Teorema 1.8. ([23], lema 2.17, teorema 2.19) Sea $\mathcal{X} \subseteq \mathbb{R}^n$ un conjunto no vacío y compacto y $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ una función cuyas funciones componentes $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ son semicontinuas inferiormente para todo $i = 1, \dots, p$, entonces $\mathcal{X}_E \neq \emptyset$.

En el caso particular de los problemas de optimización lineal multiobjetivo, se cumple que el conjunto factible $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ es cerrado y convexo y las funciones lineales son continuas, luego para garantizar que se cumplen las condiciones que exige el teorema 1.8 solo falta asumir que $\mathcal{X} \neq \emptyset$ es acotado.

En el caso de los problemas de optimización lineal entera multiobjetivo, donde la región factible del problema corresponde a $\mathcal{S} = \mathcal{X} \cap \mathbb{Z}^n$, asumimos que \mathcal{X} es acotado, lo que implica que tanto \mathcal{S} como $\mathcal{Z} = f(\mathcal{S})$ son finitos.

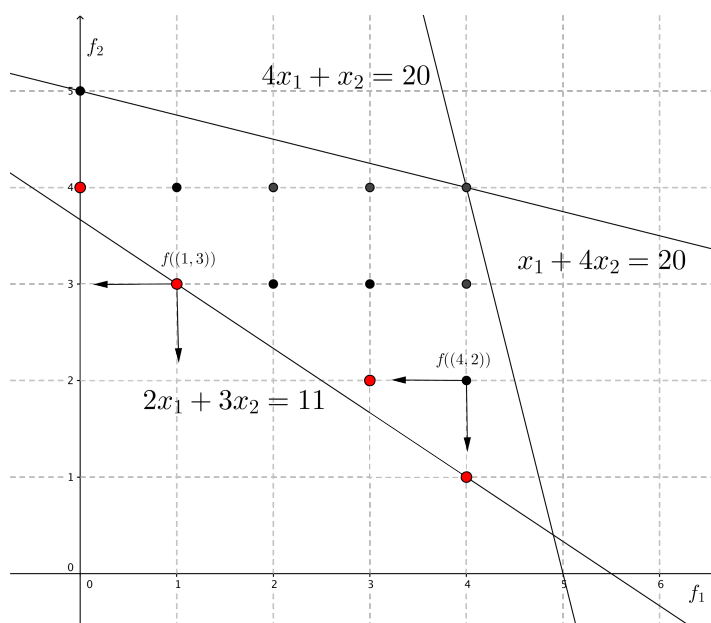


Figura 1.8: Interpretación geométrica de solución eficiente en el ejemplo 1.2.

1.3. Cotas para la frontera de Pareto

Cuando los conjuntos \mathcal{X}_E y \mathcal{F} son no vacíos, podemos hablar de cotas para la frontera de Pareto, que representamos por dos puntos, el punto ideal que corresponde a la cota inferior del conjunto \mathcal{F} y el punto nadir que corresponde a la cota superior. Entonces, el propósito es encontrar dos vectores, \underline{z} y \bar{z} , en \mathbb{R}^p tales que $\underline{z}_i < z_i < \bar{z}_i$ para todo $i = 1, \dots, p$ y $z \in \mathcal{F}$.

Para el punto \underline{z} una buena opción es seleccionar para la coordenada i -ésima el mejor valor para cada objetivo i . Formalmente,

Definición 1.9. El punto $z^I = (z_1^I, \dots, z_p^I)$ con

$$\underline{z}_i = z_i^I = \min_{x \in \mathcal{X}} f_i(x) = \min_{z \in \mathcal{Z}} z_i$$

es llamado el *punto ideal* del problema de optimización multiobjetivo (1.2).

En problemas de optimización multiobjetivo, el punto ideal normalmente no forma parte del conjunto objetivo \mathcal{Z} , pues las funciones objetivo generalmente están en conflicto. Cuando dicho punto está en \mathcal{Z} , no se requiere seguir buscando ya que este es el único punto no dominado disponible.

Para el punto \bar{z} una posible cota es

$$\bar{z}_i = \max_{x \in \mathcal{X}} f_i(x) = \max_{z \in \mathcal{Z}} z_i$$

Sin embargo esta cota suele estar muy lejos de los puntos no dominados, por tanto es preferible usar como cota:

Definición 1.10. El punto $z^N = (z_1^N, \dots, z_p^N)$ con

$$z_i^N = \max_{x \in \mathcal{X}_E} f_i(x) = \max_{z \in \mathcal{F}} z_i$$

es llamado el *punto nadir* del problema de optimización multiobjetivo (1.2).

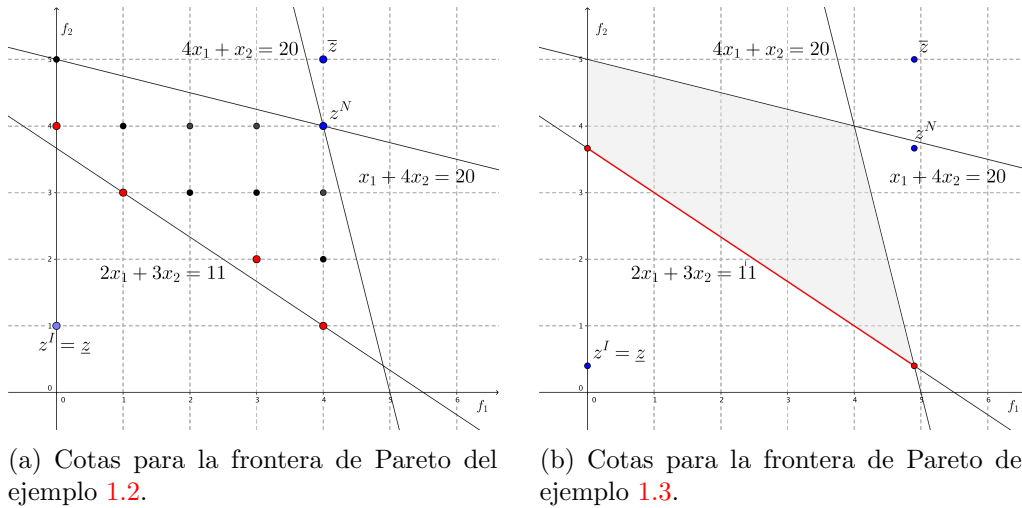


Figura 1.9

El punto ideal lo podemos calcular resolviendo p problemas de optimización mono-objetivo, sin embargo no existe un método eficiente para calcular el punto nadir de manera general, pues en este caso necesitamos optimizar sobre el conjunto \mathcal{F} .

Debido a la dificultad para calcular el punto nadir, se hace uso de la tabla de pagos (*pay-off table*), ([23, p.p. 34-35]), para obtener una aproximación de dicho punto. La matriz de pagos tiene p filas y p columnas. Si resolvemos el problema $\min_{x \in \mathcal{X}} f_i(x)$ y notamos x^i su solución, el elemento j, i de la matriz de pagos se calcula evaluando $f_j(x^i)$.

Usando esta información calculamos la tabla de pagos como se muestra en la tabla 1.2.

En la tabla de pagos las componentes del vector ideal estan localizadas en la diagonal de la tabla y el valor aproximado de la componente z_i^N del vector nadir lo obtenemos seleccionando el mayor elemento de la fila i .

	x^1	x^2	\dots	x^p
f_1	z_1^I	$f_1(x^2)$	\dots	$f_1(x^p)$
f_2	$f_2(x^1)$	z_2^I	\dots	$f_2(x^p)$
\vdots	\vdots	\vdots	\ddots	\vdots
f_p	$f_p(x^1)$	$f_p(x^2)$	\dots	z_p^I

Tabla 1.2

La tabla pagos presenta inconvenientes cuando hay más de dos objetivos. Por ejemplo, cuando al minimizar cada función objetivo de manera individual, existe más de una solución óptima, la tabla de pagos nos puede llevar a subestimar o sobreestimar el vector nadir, como sucede en el siguiente ejemplo presentado en [37] y retomado en [23].

Ejemplo 1.11.

$$\begin{aligned} \min(f_1(x) = & -11x_2 & -11x_3 & -12x_4 & -9x_5 & -9x_6 & +9x_7, \\ f_2(x) = & -11x_1 & & -11x_3 & -9x_4 & -12x_5 & -9x_6 & +9x_7, \\ f_3(x) = & -11x_1 & -11x_2 & & -9x_4 & -9x_5 & -12x_6 & -12x_7) \end{aligned}$$

s. a. $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 1, \quad x \geq 0$

Para construir la tabla de pagos, primero resolvemos los tres problemas mono-objetivo:

- El problema $\min_{x \in \mathcal{X}} f_1(x)$ solo tiene una solución, $x_4 = 1$ y $x_i = 0$ con $i \neq 4$, es decir, $x^1 = e_4$.
- El problema $\min_{x \in \mathcal{X}} f_2(x)$ también tiene solo una solución, $x_5 = 1$ y $x_i = 0$ con $i \neq 5$, es decir, $x^2 = e_5$.
- El problema $\min_{x \in \mathcal{X}} f_3(x)$ tiene infinitas soluciones, $x_6 = \alpha, x_7 = 1 - \alpha$ y $x_i = 0$ con $i \neq 6, 7$ y $\alpha \in [0, 1]$ es decir, $x^3 = \alpha e_6 + (1 - \alpha)e_7$.

La tabla de pagos corresponde a la tabla 1.3 donde hemos seleccionado dos soluciones óptimas diferentes, e_6 y e_7 , para el tercer problema.

Por otro lado tenemos que cada i -ésimo vector unidad e_i con $i = 1, \dots, 6$ es eficiente, mientras que el vector e_7 es débilmente eficiente. Como $f(e_1) = (0, -11, -11), f(e_2) = (-11, 0, -11)$ y $f(e_3) = (-11, -11, 0)$ entonces $z_j^N \geq 0$ con $j = 1, 2, 3$, pero debido a que en este ejemplo ningún punto no dominado puede tener coordenadas positivas, el punto nadir es $z^N = (0, 0, 0)$. Sin

	e_4	e_5	e_6	e_7
f_1	-12	-9	-9	9
f_2	-9	-12	-9	9
f_3	-9	-9	-12	-12

Tabla 1.3

embargo, ninguno de estos valores aparece en la tabla de pagos: con $x^3 = e_7$ sobreestimamos z_1^N y z_2^N y con $x^3 = e_6$ subestimamos z_1^N y z_2^N .

La sobreestimación se presenta porque e_7 es una solución débilmente eficiente, e_6 lo domina. Sin embargo, esta se puede evitar, si logramos garantizar que las soluciones x^k son eficientes.

En el caso de dos objetivos, podemos determinar el vector nadir, siguiendo el algoritmo 1 presentado en [23, p. 38], donde al resolver los últimos dos problemas con las restricciones adicionales (líneas 3 y 4), lo que estamos haciendo es eliminando las soluciones débilmente eficientes en la tabla de pagos.

Algoritmo 1 Algoritmo para obtener el punto nadir cuando $p = 2$

- 1: **Entrada:** Conjunto factible \mathcal{X} y funciones objetivo $f = (f_1, f_2)$
 - 2: Resolver los problemas mono-objetivo $\min_{x \in \mathcal{X}} f_1(x)$ y $\min_{x \in \mathcal{X}} f_2(x)$. Notar los valores óptimos como z_1^I y z_2^I .
 - 3: Resolver $\min_{x \in \mathcal{X}} f_2(x)$ con la restricción adicional $f_1(x) \leq z_1^I$.
 - 4: Resolver $\min_{x \in \mathcal{X}} f_1(x)$ con la restricción adicional $f_2(x) \leq z_2^I$.
 - 5: Notar los valores óptimos como z_2^N y z_1^N respectivamente.
 - 6: **Salida:** El punto nadir $z^N = (z_1^N, z_2^N)$ y el punto ideal $z^I = (z_1^I, z_2^I)$.
-

1.4. Métodos para resolver problemas de optimización multiobjetivo

Los métodos y técnicas que se utilizan para resolver los problemas de optimización multiobjetivo admiten diferentes clasificaciones. Una de ellas se establece de acuerdo a la intervención que tiene, durante el proceso de solución, el tomador de decisiones, quien es aquella persona que selecciona una alternativa de todas las que tiene disponibles en el conjunto de soluciones.

Siguiendo esta clasificación podemos distinguir los métodos a priori, a posteriori y los interactivos.

1.4.1. Métodos a priori

En los *métodos a priori* el tomador de decisiones especifica sus preferencias para cada objetivo antes de que el proceso de búsqueda de solución inicie y no interfiere durante el proceso. Después de finalizado el método solo se obtiene una solución. La estrategia consiste en resolver uno o varios problemas de un solo objetivo, sin embargo, una dificultad que se presenta es que el tomador de decisiones no necesariamente conoce con anterioridad el comportamiento del problema, lo que supone un obstáculo al determinar sus preferencias.

Los métodos que se destacan en esta categoría [47] son el *método de la función de valor*, donde el tomador de decisiones debe especificar la expresión matemática de una función de valor $U : \mathbb{R}^p \rightarrow \mathbb{R}$ que refleje sus preferencias, convirtiendo el problema (1.2) en uno mono-objetivo.

Otro método es la *ordenación lexicográfica*, donde el decisor debe ordenar las funciones objetivo de acuerdo a su importancia. Se minimiza la función objetivo situada en primer lugar sujeta a las restricciones originales, a continuación la segunda función objetivo priorizada es minimizada y a las restricciones originales se adiciona una más, aquella que garantiza que la función objetivo situada en primer lugar conserva su valor óptimo. El proceso continúa de manera similar hasta agotar las funciones objetivo.

Finalmente tenemos el método de *programación por metas*, donde el tomador de decisiones especifica un nivel de aspiración \tilde{z}_j para cada función objetivo. Luego formula las metas, entendida como la conexión que hay entre un objetivo y su nivel de aspiración, que en problemas de minimización son de la forma $f_j(x) \leq \tilde{z}_j$. Y por último minimiza las desviaciones $\delta_j = \max[0, f_j(x) - \tilde{z}_j]$ de los valores de la función objetivo.

1.4.2. Métodos a posteriori

Este tipo de métodos no requiere que el tomador de decisiones exprese sus preferencias con anterioridad, su propósito es encontrar toda la frontera de Pareto (o un subconjunto de ésta), modificando algunos parámetros, para que después de ejecutar el algoritmo cierto número de veces y encontrar un conjunto de puntos no dominados, el tomador de decisiones seleccione la solución que considere adecuada, por esta razón estos métodos también son llamados *métodos generadores*.

Dentro de estos métodos hay dos [15, 47] considerados básicos: el *método de la suma ponderada* (ponderaciones o coeficientes de peso) que consiste en asociar a cada función objetivo un coeficiente de ponderación y luego se minimiza la suma ponderada de los objetivos conservando las restricciones originales del problema, esto es, la idea es convertir el problema original en uno mono-objetivo, lo que se conoce como escalarizar el problema.

El otro método es el de ϵ -restricciones propuesto por Haimes [29], también recurre a la escalarización del problema original pero estableciendo cotas para cada función objetivo.

En esta categoría también se inscriben el método híbrido que combina los dos anteriores, el método de restricción elástica, el método de Benson y otros cuyos detalles pueden ser consultados en [23, 47].

A continuación presentamos algunos detalles de los dos métodos básicos.

Método de la suma ponderada

En este método debemos resolver problemas mono-objetivo de la forma

$$\min_{x \in \mathcal{X}} \sum_{i=1}^p \lambda_i f_i(x) \quad (1.3)$$

con $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}_{\geq}^p \setminus \{0\}$.

Este método es muy sencillo de llevar a cabo, pero no nos asegura el cálculo de la frontera de Pareto. Dada una solución de dicho método, tenemos los siguientes resultados:

Teorema 1.12. Sea x^* una solución óptima de (1.3) entonces se cumple que:

1. Si $\lambda \in \mathbb{R}_{\geq}^p \setminus \{0\}$ entonces x^* es una solución débilmente eficiente de (1.2).
2. Si $\lambda \in \mathbb{R}_{>}^p$ entonces x^* es una solución eficiente de (1.2).
3. Si $\lambda \in \mathbb{R}_{\geq}^p \setminus \{0\}$ y x^* es la única solución óptima de (1.3), entonces x^* es una solución eficiente de (1.2).

Este teorema implica que algunos puntos de la frontera de Pareto pueden ser encontrados resolviendo (1.3) con una selección apropiada de los λ_i sin fijar alguna condición particular sobre las funciones objetivo o la región factible.

De hecho, cuando asumimos que tanto las funciones objetivo como el conjunto factible \mathcal{X} son convexos podemos encontrar toda la frontera de Pareto.

Teorema 1.13. Sean \mathcal{X} un conjunto convexo y f_i una función convexa para cada $i = 1, \dots, p$. Si x^* es una solución débilmente eficiente de (1.2) entonces existe $\lambda \in \mathbb{R}_{\geq}^p \setminus \{0\}$ tal que x^* es una solución óptima de (1.3).

Notemos que el teorema anterior sigue siendo válido si x^* es una solución eficiente, pues \mathcal{X}_E es un subconjunto del conjunto de las soluciones débilmente eficientes. En [23] se presenta una línea teórica completa que sustenta los dos teoremas anteriores.

En esta tesis nos centramos en la optimización discreta y el uso del método de las ponderaciones no nos asegura el cálculo completo de la frontera de Pareto, pues la región factible $\mathcal{X} = \{x \in \mathbb{Z}^n : Ax = b, x \geq 0\}$, de dichos problemas, no es convexa. A manera de ilustración, si retomamos el ejemplo 1.2 tenemos que:

- Si $\lambda = (1, 1)$ los puntos $(0, 4)$ y $(1, 3)$ son soluciones óptimas del problema $\min_{x \in \mathcal{X}} x_1 + x_2$ con $\mathcal{X} = \{x \in \mathbb{Z}^2 : 2x_1 + 3x_2 \geq 11, x_1 + 4x_2 \leq 20, 4x_1 + x_2 \leq 20, x \geq 0\}$.
- Si $\lambda = (1, 2)$ el punto $(4, 1)$ es solución óptima del problema $\min_{x \in \mathcal{X}} x_1 + 2x_2$.

Usando el teorema 1.12, los puntos $(0, 4)$, $(1, 3)$ y $(4, 1)$ son soluciones eficientes del problema 1.2. Sin embargo, para este problema $\mathcal{X}_E = \{(0, 4), (1, 3), (3, 2), (4, 1)\}$. No existe un $\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}_{>}^2$ tal que $(3, 2)$ sea solución óptima de un problema de la forma (1.3), pues si $\lambda_1 < \lambda_2$ el coste de $(3, 2)$ va a ser mayor que el valor óptimo de $(4, 1)$; y si $\lambda_1 > \lambda_2$ el coste de $(3, 2)$ va a ser mayor que el valor óptimo de $(1, 3)$.

Lo anterior sugiere una distinción de las soluciones eficientes y en consecuencia de los puntos no dominados.

Definición 1.14. Sea $\min_{x \in \mathcal{X}} f(x) = (f_1(x), \dots, f_p(x))$ un problema de optimización lineal entera multiobjetivo con \mathcal{X}_E el conjunto eficiente y \mathcal{F} el conjunto no dominado.

Sea $x \in \mathcal{X}_E$. Decimos que x es una *solución eficiente soportada* si existe $\lambda \in \mathbb{R}_{>}^p$ tal que x es una solución óptima de $\min_{x \in \mathcal{X}} \sum_{i=1}^p \lambda_i f_i(x)$. A $z = f(x)$ lo llamamos un *punto no dominado soportado*. Al conjunto de las soluciones eficientes soportadas lo notamos \mathcal{X}_{sE} y al conjunto de los puntos no dominados soportados lo notamos \mathcal{F}_s . En caso contrario, decimos que x es una *solución eficiente no soportada* y $z = f(x)$ un *punto no dominado no soportado*. A los conjuntos correspondientes a dichas soluciones los notamos \mathcal{X}_{nE} y \mathcal{F}_n respectivamente.

Método de las ϵ -restricciones

Este método, introducido en [29] y formulado para resolver problemas de optimización multiobjetivo en general,

$$\min_{x \in \mathcal{X}} f(x) = (f_1(x), \dots, f_p(x)) \quad (1.4)$$

nos permite transformar dicho problema en una familia de problemas de optimización mono-objetivo con restricciones adicionales, específicamente debemos seleccionar una función objetivo para optimizar y las $p - 1$ funciones objetivo restantes, añadirlas como restricciones, esto es, debemos resolver varios problemas de la forma

$$P_k(\epsilon) \begin{cases} \min f_k(x) \\ \text{s. a. } x \in \mathcal{X} \\ f_j(x) \leq \epsilon_j, \quad 1 \leq j \leq p, \quad j \neq k \end{cases} \quad (1.5)$$

con $\epsilon = (\epsilon_1, \dots, \epsilon_{k-1}, \epsilon_{k+1}, \dots, \epsilon_p)$, donde cada problema $P_k(\epsilon)$ se obtiene asignando valores diferentes a la cota ϵ_j de cada restricción.

Con una adecuada elección del vector ϵ podemos obtener todo el conjunto Pareto, como se deduce del teorema:

Teorema 1.15. Una solución factible x^* de (1.4) es eficiente si y solo si x^* es una solución óptima de $P_k(\epsilon^*)$ donde $\epsilon_j^* = f_j(x^*)$, $j = 1, \dots, p$, $j \neq k$, para cada $k = 1, \dots, p$.

Demostración. Sea x^* una solución eficiente de (1.4). Supongamos que x^* no es solución óptima de $P_k(\epsilon^*)$ para algún k donde $\epsilon_j^* = f_j(x^*)$, $j = 1, \dots, p$, $j \neq k$. Entonces existe $x \in \mathcal{X}$ tal que $f_k(x) < f_k(x^*)$ y $f_j(x) \leq f_j(x^*)$ con $j = 1, \dots, p$, $j \neq k$, pero esto contradice que $x^* \in \mathcal{X}_E$.

Sea x^* una solución óptima de $P_k(\epsilon^*)$ para cada $k = 1, \dots, p$, entonces no existe $x \in \mathcal{X}$ tal que $f_j(x) \leq f_j(x^*)$ para $j = 1, \dots, p$ con al menos una desigualdad estricta. Por tanto concluimos que $x^* \in \mathcal{X}_E$. \square

A diferencia del método de las ponderaciones el resultado anterior nos asegura que el método de las ϵ -restricciones nos proporciona la frontera de Pareto completa. No obstante, este método implica un gran costo computacional, ya que la cantidad de problemas que debemos resolver puede ser enorme, tan sólo delimitado por las cotas que tengamos de las funciones de coste para los puntos eficientes. Además necesitamos obtener la misma solución para $p - 1$ problemas para poder asegurar su condición de eficiente.

Otra condición para obtener soluciones eficientes es dada por la unicidad de la solución de $P_k(\epsilon^*)$.

Teorema 1.16. Si x^* es una solución óptima de $P_k(\epsilon^*)$ para algún k , donde $\epsilon_j^* = f_j(x^*)$, $j = 1, \dots, p$, $j \neq k$ y esta es única, entonces x^* es una solución eficiente de (1.4).

Demostración. Sea x^* la única solución óptima de $P_k(\epsilon^*)$ para algún k donde $\epsilon_j^* = f_j(x^*)$, $j = 1, \dots, p$, $j \neq k$. Supongamos que $x^* \notin \mathcal{X}_E$, luego existe $x' \in \mathcal{X}$ tal que $f_j(x') \leq f_j(x^*)$ para $j = 1, \dots, p$ con al menos una desigualdad estricta. Sin embargo, la unicidad de x^* nos indica para todo $x \in \mathcal{X}$ tal que $f_j(x) \leq f_j(x^*)$ con $j = 1, \dots, p$, $j \neq k$ se cumple que $f_k(x^*) < f_k(x)$ lo que nos conduce a una contradicción. Por tanto, $x^* \in \mathcal{X}_E$. \square

La unicidad de la solución es una condición muy restrictiva, sin embargo, podemos asegurar una condición más débil sobre el óptimo del problema $P_k(\epsilon^*)$, esto es,

Teorema 1.17. Si x^* es una solución óptima de $P_k(\epsilon^*)$ para algún k , donde $\epsilon_j^* = f_j(x^*)$, $j = 1, \dots, p$, $j \neq k$, entonces x^* es una solución débilmente eficiente de (1.4).

Demostración. Sea x^* una solución óptima de $P_k(\epsilon^*)$ para algún k y supongamos que x^* no es una solución débilmente eficiente de (1.4). Entonces existe $x \in \mathcal{X}$ tal que $f_j(x) < f_j(x^*)$ para todo $j = 1, \dots, p$. Como x es factible para $P_k(\epsilon^*)$ y en particular se tiene que $f_k(x) < f_k(x^*)$, esto contradice que x^* sea una solución óptima de $P_k(\epsilon^*)$. \square

Otros resultados sobre este método pueden ser consultados en [15].

Aunque teóricamente todo punto de la frontera de Pareto de un problema de optimización multiobjetivo puede ser obtenido mediante el método de ϵ -restricciones, puede ser difícil especificar las cotas para cada función objetivo, sin embargo, podemos utilizar las componentes del punto ideal como punto de partida, esto es, definir $\epsilon_j = z_j^I + e_j$ con $j = 1, \dots, p$, $j \neq k$ y e_j un número real positivo que vamos cambiando.

En los problemas de optimización lineal entera multiobjetivo, cuya región factible es no convexa, podemos obtener toda la frontera de Pareto con este método, por ejemplo, si partimos de los valores que nos indican las componentes del punto ideal e incrementamos cada ϵ_j de uno en uno hasta el valor que nos indique cada componente del punto nadir. Siguiendo con el ejemplo 1.2 tenemos que $z^I = (0, 1)$ y $z^N = (4, 4)$. Si seleccionamos la función $f_1(x) = x_1$ como restricción, debemos resolver los problemas

$$P_2(\epsilon_1) \begin{cases} \text{mín } f_2(x) = x_2 \\ \text{s. a. } x \in \mathcal{X} \\ f_1(x) \leq \epsilon_1 \end{cases} \quad (1.6)$$

con $0 \leq \epsilon_1 \leq 4$, de donde tenemos que:

- La solución óptima de $P_2(0)$ es $(0, 4)$.
- La solución óptima de $P_2(1)$ es $(1, 3)$.
- Las soluciones óptimas de $P_2(2)$ son $(1, 3)$ y $(2, 3)$.
- La solución óptima de $P_2(3)$ es $(3, 2)$.
- La solución óptima de $P_2(4)$ es $(4, 1)$.

Luego debemos seleccionar solo las soluciones eficientes y así obtenemos que $\mathcal{X}_E = \{(0, 4), (1, 3), (3, 2), (4, 1)\}$.

1.4.3. Métodos interactivos

Los *métodos interactivos* son aquellos donde el tomador de decisiones define sus preferencias antes de iniciar el proceso de búsqueda de solución, pero puede modificar dichas preferencias durante la ejecución del algoritmo correspondiente, en la medida en que conoce las soluciones parciales del problema y obtiene más información sobre él. Este proceso de búsqueda termina cuando el tomador de decisiones considera buena la solución obtenida. En [47] se encuentra una descripción detallada de algunos métodos que se destacan en esta categoría.

Como vimos hay diferentes métodos para abordar los problemas de optimización multiobjetivo, sin embargo en este trabajo estamos interesados en aquellos donde las preferencias del tomador de decisiones no son tenidas en cuenta en su implementación, por tanto, el objetivo es determinar todo el conjunto \mathcal{Z} y \mathcal{X}_E (un representante para cada punto no dominado). Después de esto, se debe escoger una solución del conjunto de soluciones eficientes proporcionado, usando algún método para ello (véase [31] o [30]), pero esta parte está fuera del alcance de este trabajo.

CAPÍTULO 2

Programación entera desde un punto de vista algebraico

La programación entera utiliza diferentes técnicas para resolver problemas de optimización discreta. El uso efectivo de herramientas algebraicas comienza con el trabajo de Conti y Traverso [16], donde el algoritmo de Buchberger permite resolver problemas de optimización lineal entera. Posteriormente Sturmfels [54] y Thomas [57] enriquecen este punto de vista, conectando dicha teoría con el concepto de test-set, introducido desde un punto de vista teórico por Graver [27] en 1975. En este capítulo describiremos el algoritmo de mejora basado en test-sets.

2.1. Introducción

Consideramos problemas de programación lineal entera de la forma

$$\begin{aligned} & \text{mín } c(x) \\ & \text{s.a. } Ax = b, \quad x \in \mathbb{N}^n \end{aligned} \tag{2.1}$$

donde $A \in \mathbb{Z}^{m \times n}$ con $m \leq n$ es una matriz de rango máximo, $b \in \mathbb{Z}^m$ y $c \in \mathbb{Z}^n$. A este problema lo notamos $IP_{A,c}(b)$. La familia de problemas lineales enteros obtenida variando b mientras A y c permanecen fijos, será $IP_{A,c}$.

El conjunto de soluciones factibles de $IP_{A,c}(b)$ es $\mathcal{X} = P_b \cap \mathbb{N}^n$, donde $P_b = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$, que asumimos acotado para cualquier vector

b , es el conjunto de soluciones factibles del programa lineal $LP_{A,c}(b)$:

$$\begin{aligned} & \text{mín } c(x) \\ & \text{s.a. } Ax = b, \quad x \geq 0 \end{aligned} \tag{2.2}$$

Sea $\pi : \mathbb{N}^n \rightarrow \mathbb{Z}^m$ el homomorfismo de semigrupos dado por $\pi(x) = Ax$, luego el conjunto \mathcal{X} de soluciones factibles de $IP_{A,c}(b)$ corresponde a $\pi^{-1}(b)$, la pre-imagen de b bajo π que llamaremos la b -fibra de $IP_{A,c}$.

Todo vector $c \in \mathbb{Z}^n$ ordena los puntos x en \mathbb{N}^n vía el producto escalar $c \cdot x$, esto es, dados α y β en \mathbb{N}^n ,

$$\alpha >_c \beta \text{ si y solo si } \alpha = \beta \text{ o } c \cdot \alpha > c \cdot \beta$$

Sin embargo $>_c$ es un orden parcial, en particular α y β no son comparables cuando $\alpha \neq \beta$ y $c \cdot \alpha = c \cdot \beta$.

2.2. Test set

Rekha Thomas en [57] presenta un método para resolver un programa lineal entero que recurre a la idea natural de buscar en la vecindad de un punto factible, otra solución con un mejor valor de coste, de manera que si se encuentra dicha mejora, se reemplaza la actual y se repite la búsqueda.

La vecindad proviene de un conjunto de vectores, del núcleo entero de la matriz A , por lo que al sumarlos o restarlos, se verifican las restricciones impuestas. Al mismo tiempo queremos asegurar que el óptimo se encuentra mediante estos vectores, estas condiciones se resumen en el concepto de test-set acuñado por Graver [27].

Definición 2.1. Un conjunto $\mathcal{T} \subseteq \{x \in \mathbb{Z}^n : Ax = 0\}$ es un *test set* para la familia de programas enteros $IP_{A,c}$ si

1. para cada solución factible no óptima α de cada programa de $IP_{A,c}$, existe $g \in \mathcal{T}$ tal que $\alpha - g$ es una solución factible del mismo programa con $\alpha >_c \alpha - g$, y
2. para la solución óptima β del programa en $IP_{A,c}$, $\beta - g$ es no factible para todo $g \in \mathcal{T}$.

La existencia de un test set \mathcal{T} para $IP_{A,c}$ implica un algoritmo sencillo (algoritmo 2) para encontrar una solución óptima de cada programa entero en $IP_{A,c}$ con $b \in \mathbb{N}\mathcal{A}$.

El algoritmo 2 efectivamente termina, pues al ejecutar el ciclo “Mientras” (líneas 3-5) vamos obteniendo una cadena decreciente de elementos de

Algoritmo 2 Algoritmo para obtener una solución óptima de un programa entero en $IP_{A,c}$ utilizando un test-set

- 1: **Entrada:** Un test-set \mathcal{T} para el programa entero $IP_{A,c}(b)$ y una solución factible α .
 - 2: $\beta := \alpha$
 - 3: **Mientras** exista $g \in \mathcal{T}$ tal que $g \leq \beta$ **hacer:**
 - 4: Seleccione un $g \in \mathcal{T}$ tal que $g \leq \beta$
 - 5: $\beta := \beta - g$
 - 6: **Fin Mientras**
 - 7: **Salida:** El vector β que es un óptimo de $IP_{A,c}(b)$.
-

$P_b \cap \mathbb{N}^n$ y esta se detiene, ya que asumimos que $IP_{A,c}(b)$ tiene por lo menos una solución óptima.

Cuando $b \in \pi(\mathbb{N}^n)$ el problema $IP_{A,c}(b)$ puede tener varias soluciones óptimas, sin embargo solo estamos interesados en encontrar una. Para esto, vamos a definir un orden total que notaremos \succ_c a partir del orden inducido por c , esto es, un orden producto de $>_c$ por un orden monomial \succ sobre \mathbb{N}^n , de la siguiente manera:

$$\alpha \succ_c \beta \text{ si } \begin{cases} c \cdot \alpha > c \cdot \beta & \text{o} \\ c \cdot \alpha = c \cdot \beta & \text{y } \alpha \succ \beta \end{cases}$$

Este refinamiento del orden $>_c$ nos ofrece una gran ventaja, hace que el problema $IP_{A,\succ_c}(b)$ tenga una única solución óptima, sin embargo el valor óptimo de $IP_{A,\succ_c}(b)$ y de $IP_{A,c}(b)$ es el mismo.

Recordemos que un orden monomial sobre \mathbb{N}^n se define como:

Definición 2.2. Un orden monomial sobre \mathbb{N}^n es un un orden total \succeq que además cumple:

1. \succeq es compatible con la adición en \mathbb{N}^n , esto es, si $\alpha \succeq \beta$ entonces $\alpha + \gamma \succeq \beta + \gamma$ para todo α, β y γ en \mathbb{N}^n .
2. $\alpha \succeq 0$ para todo α en $\mathbb{N}^n \setminus \{0\}$.

Algunos ejemplos de órdenes monomiales definidos sobre el monoide $(\mathbb{N}^n, +)$ son: sean $\alpha = (a_1, \dots, a_n)$ y $\beta = (b_1, \dots, b_n)$ en \mathbb{N}^n ,

- *Orden lexicográfico:* $\alpha <_{lex} \beta$ si y solo si $a_1 = b_1, a_2 = b_2, \dots, a_{i-1} = b_{i-1}$ y $a_i < b_i$ para algún $i = 1, \dots, n$, o si $\alpha = \beta$.
- *Orden lexicográfico graduado:* $\alpha <_{deglex} \beta$ si y solo si $\sum_{i=1}^n a_i < \sum_{i=1}^n b_i$ o si $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i$ y $\alpha <_{lex} \beta$.

- *Orden producto* $\prec_{u,\sigma}$ [17, p. 72]: sea $u = (u_1, \dots, u_n) \in \mathbb{N}^n$ y un orden monomial \prec_σ (por ejemplo, \prec_{lex} , \prec_{deglex} o $\prec_{degrevlex}$), sobre \mathbb{N}^n , entonces decimos que $\alpha \prec_{u,\sigma} \beta$ si y solo si $u \cdot \alpha < u \cdot \beta$, o $u \cdot \alpha = u \cdot \beta$ y $\alpha \prec_\sigma \beta$.

Cabe resaltar que aunque \succ_c es un orden total, no siempre es un orden monomial, pues como $c \in \mathbb{Z}^n$ puede no satisfacer la condición $c \cdot \alpha > c \cdot 0$ para todo $\alpha \in \mathbb{N}^n - \{0\}$; por ejemplo cuando el vector c tiene componentes negativas, $c = (-2, 1)$, tenemos que $c \cdot (0, 0) = 0$ y $c \cdot (1, 0) = -2$, luego $(0, 0) \succ_c (1, 0)$.

Podemos asumir que $c \geq 0$ usando el razonamiento de [20], el hecho de que P_b sea acotado para todo b es equivalente a la inexistencia de direcciones extremas, es decir, el sistema $Ax = 0$ con $x \geq 0$ no tiene alguna solución con alguna componente estrictamente positiva. Por el teorema de Tucker [38], una variante del lema de Farkas, esto implica que existe una solución a $y^T A > 0$, es decir, existe un vector estrictamente positivo en el espacio generado por las filas de A , esto es, $y^T A = d^T$ con $d > 0$. Por tanto, $y^T b = y^T Ax = d^T x$ para todo $x \in P_b$.

Como $d^T x$ es el mismo para todo $x \in P_b \cap \mathbb{N}^n$, el programa entero $IP_{A,c}(b)$ es equivalente a

$$\begin{aligned} & \text{mín } (c + \lambda d)(x) \\ \text{s.a. } & Ax = b, \quad x \in \mathbb{N}^n \end{aligned} \tag{2.3}$$

con $\lambda \in \mathbb{R}$. Luego haciendo λ lo suficientemente grande, podemos encontrar el vector de coste con la condición $c \geq 0$.

Para garantizar la existencia de un test-set \mathcal{T} que solo dependa de la matriz A y del vector de coste c , primero vamos a caracterizar el conjunto \mathcal{S}_{\succ_c} , de todos los puntos en \mathbb{N}^n que son soluciones no óptimas con respecto a \succ_c de las diferentes fibras de IP_{A,\succ_c} , hecho que se deriva de la siguiente versión equivalente del Lema de Dickson presentada en [17, pp.69-71].

Lema 2.3. [Lema de Dickson] Dado $A \subseteq \mathbb{N}^n$ existe un único conjunto finito minimal $\{\alpha_1, \dots, \alpha_s\} \subseteq A$ tal que para todo $\alpha \in A$ existe algún i y $\gamma \in \mathbb{N}^n$ de manera que $\alpha = \alpha_i + \gamma$, es decir,

$$A \subseteq \bigcup_{i=1}^s (\alpha_i + \mathbb{N}^n)$$

donde $\alpha + \mathbb{N}^n = \{\alpha + \gamma : \gamma \in \mathbb{N}^n\}$.

Teorema 2.4. [57, Lema 2.1.4] Consideremos \mathcal{S}_{\succ_c} el conjunto de puntos factibles de IP_{A,\succ_c} que no son óptimos en su fibra. Existe un único conjunto finito y minimal de vectores $\alpha_1, \alpha_2, \dots, \alpha_t$ en \mathbb{N}^n tal que

$$\mathcal{S}_{\succ_c} = \bigcup_{i=1}^t (\alpha_i + \mathbb{N}^n)$$

Demostración. Como $\mathcal{S}_{\succ_c} \subseteq \mathbb{N}^n$, por el lema de Dickson, existe un conjunto finito y minimal de vectores $\alpha_1, \alpha_2, \dots, \alpha_t$ en \mathcal{S}_{\succ_c} tal que $\mathcal{S}_{\succ_c} \subseteq \bigcup_{i=1}^t (\alpha_i + \mathbb{N}^n)$.

Veamos la inclusión contraria: si $\alpha \in \mathcal{S}_{\succ_c}$ entonces α es una solución no óptima de la $A\alpha$ -fibra de $IP_{A,c}$ y sea β el único óptimo de dicha fibra. Dado $\gamma \in \mathbb{N}^n$, tenemos que $\alpha + \gamma \in \mathcal{S}_{\succ_c}$, es decir, $\alpha + \gamma$ es una solución no óptima en la $A(\alpha + \gamma)$ -fibra pues se cumple que

- $A(\alpha + \gamma) = A(\beta + \gamma)$ ya que $A\alpha = A\beta$.
- $(\alpha + \gamma), (\beta + \gamma) \in \mathbb{N}^n$ pues $\alpha, \beta, \gamma \in \mathbb{N}^n$.
- Como $\alpha \succ_c \beta$ entonces $\alpha + \gamma \succ_c \beta + \gamma$

Por tanto $\mathcal{S}_{\succ_c} = \bigcup_{i=1}^t (\alpha_i + \mathbb{N}^n)$. □

La construcción anterior nos permite definir un conjunto de vectores a partir de los elementos minimales no óptimos y los óptimos de su fibra.

Definición 2.5. Sea $\mathcal{T}_{\succ_c} = \{g_i = \alpha_i - \beta_i : i = 1, \dots, t\}$ donde $\alpha_1, \alpha_2, \dots, \alpha_t$ son los únicos elementos minimales de \mathcal{S}_{\succ_c} y β_i es el único óptimo del programa $IP_{A,\succ_c}(A\alpha_i)$.

Al pertenecer dichas parejas a la misma fibra, $\mathcal{T}_{\succ_c} \subseteq \{x \in \mathbb{Z}^n : Ax = 0\}$ y los vectores α_i y β_i tienen soporte disjunto. Supongamos que α_i y β_i no tienen soporte disjunto, luego podemos definir el vector entero w diferente de cero y no negativo de manera que

$$w_j = \min\{\alpha_{i_j}, \beta_{i_j}\} \text{ para } j = 1, \dots, n$$

De la definición de w tenemos que $\alpha_i - w \geq 0$ y $\beta_i - w \geq 0$, luego son soluciones factibles en la $A(\alpha_i - w)$ -fibra y $\alpha_i - w \succ_c \beta_i - w$ pues $\alpha_i \succ_c \beta_i$. Entonces $\alpha_i - w$ es una solución no óptima en su fibra y por tanto pertenece a \mathcal{S}_{\succ_c} . Pero $\alpha_i - w < \alpha_i$ lo que contradice que α_i sea un generador minimal de \mathcal{S}_{\succ_c} .

La definición de test-set permite que diferentes conjuntos admitan el uso del Algoritmo 2, también denominado de aumentación, como sucede con las bases de Graver. Sin embargo el conjunto que acabamos de describir nos proporciona un test-set minimal.

Teorema 2.6. \mathcal{T}_{\succ_c} es el único test-set minimal para la familia IP_{A,\succ_c} , que solo depende de la matriz A y del orden \succ_c .

Demostración. Probaremos en primer lugar que el conjunto \mathcal{T}_{\succ_c} es un test set. Sea α una solución factible no óptima para su fibra con β el óptimo. Por el teorema 2.4, existe $\alpha_i \in \mathcal{S}_{\succ_c}$ tal que $\alpha = \alpha_i + \gamma$ para algún $\gamma \in \mathbb{N}^n$, entonces $\alpha - g_i = \alpha - (\alpha_i - \beta_i) = \beta_i + \gamma \in \mathbb{N}^n$ está en la misma fibra de α y $\alpha \succ_c (\alpha - g_i)$ pues $\alpha_i \succ_c \beta_i$.

Sea β solución óptima y supongamos que existe $g_i = \alpha_i - \beta_i \in \mathcal{T}_{\succ_c}$ tal que $\beta - g_i$ es factible, así, $A\beta = A(\beta - g_i)$ y $\beta - g_i = \beta - \alpha_i + \beta_i \in \mathbb{N}^n$. Distingamos dos casos: si $\beta - \alpha_i \in \mathbb{N}^n$ entonces $\beta = \alpha_i + \gamma$ con $\gamma \in \mathbb{N}^n$, es decir, $\beta \in \mathcal{S}_{\succ_c}$, lo que contradice la minimalidad de β . Como segundo caso, supongamos que $\beta - \alpha_i \notin \mathbb{N}^n$ entonces existe k tal que la k -ésima componente es negativa, esto es, $\beta_k - \alpha_{ik} < 0$, luego $\alpha_{ik} > 0$. Como $\beta_k - \alpha_{ik} + \beta_{ik} \geq 0$, entonces $\beta_{ik} > 0$ pero esto contradice que α_i y β_i tienen soporte disjunto. Por tanto, para todo $g \in \mathcal{T}$, se cumple que $\beta - g$ es no factible.

Por último, probemos la minimalidad. Debemos mostrar que ningún elemento de \mathcal{T}_{\succ_c} puede ser eliminado de este test-set. Para cualquier $g_i = \alpha_i - \beta_i \in \mathcal{T}_{\succ_c}$ el conjunto $\mathcal{T}_{\succ_c} \setminus \{g_i\}$ no es un test-set, pues la solución no óptima α_i no puede ser mejorada por algún elemento de $\mathcal{T}_{\succ_c} \setminus \{g_i\}$. La unicidad de \mathcal{T}_{\succ_c} está garantizada por la minimalidad del conjunto $\{\alpha_1, \dots, \alpha_t\}$ de \mathcal{S}_{\succ_c} . \square

2.3. Ideales tóricos

Los ideales son estructuras algebraicas dentro de un anillo que nos permiten representar variedades algebraicas, aquéllas definidas por un conjunto de polinomios. El teorema de la base de Hilbert [3] inspira el cálculo efectivo en estas estructuras y posteriormente el algoritmo de Buchberger [10] formaliza el cálculo efectivo en ideales. En los años 90 se aplica dicho algoritmo a la optimización, el trabajo seminal de Conti y Traverso [16] proporciona un algoritmo de búsqueda del óptimo mediante el uso de un ideal y órdenes de eliminación [18]. Otros puntos de vista son aportados posteriormente por Sturmfels [54].

Notamos $k[x_1, \dots, x_n]$ el anillo de polinomios en n variables sobre un cuerpo k y los monomios en este anillo como $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ donde $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$. Como cada monomio x^α queda determinado de manera única por la n -upla del exponente $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$, podemos establecer una correspondencia biyectiva entre el conjunto de monomios de $k[x_1, \dots, x_n]$ y \mathbb{N}^n . Así, cualquier orden monomial \succeq en \mathbb{N}^n define un orden en el conjunto de los monomios, $x^\alpha \succeq x^\beta$ si y solo si $\alpha \geq \beta$. Dado

un polinomio f , el término líder respecto del orden monomial lo notamos $in_{\succeq}(f)$, lo que nos permite definir el ideal inicial.

Definición 2.7. Sea I un ideal de $k[x_1, \dots, x_n]$ distinto de cero y \succeq un orden monomial sobre $k[x_1, \dots, x_n]$, el *ideal inicial* de un ideal I es el ideal generado por los términos iniciales de I , esto es,

$$in_{\succeq}(I) := \langle in_{\succeq}(f) : f \in I \rangle$$

Como consecuencia directa del lema de Dickson (lema 2.3) o por el teorema de la base de Hilbert, el ideal monomial $in_{\succeq}(I)$ tiene un conjunto finito de generadores.

Definición 2.8. Sea \succeq un orden monomial sobre $k[x_1, \dots, x_n]$. Un subconjunto finito de polinomios $\mathcal{G}_{\succeq} = \{g_1, g_2, \dots, g_s\}$ de un ideal I es una *base de Gröbner* para I con respecto a \succeq si

$$in_{\succeq}(I) = \langle in_{\succeq}(g_1), in_{\succeq}(g_2), \dots, in_{\succeq}(g_s) \rangle.$$

Con esta definición no podemos asegurar la unicidad de la base de Gröbner, para ello introducimos dos conceptos, la base minimal y la base reducida. Ésta última nos asegura la unicidad, para un ideal y orden monomial.

Definición 2.9. \mathcal{G}_{\succeq} es una *base de Gröbner minimal* para I con respecto a \succeq si cada uno de sus elementos es un polinomio mónico y para todo $g \in \mathcal{G}_{\succeq}$, $in_{\succeq}(g) \notin in_{\succeq}(\mathcal{G}_{\succeq} - \{g\})$.

Definición 2.10. \mathcal{G}_{\succeq} es una *base de Gröbner reducida* para I con respecto a \succeq si g_i es reducido con respecto a $\mathcal{G}_{\succeq} - \{g_i\}$ para todo $i = 1, \dots, s$, esto es, ningún término de g_i es divisible por algún $in_{\succeq}(g_j)$ con $i \neq j$.

De forma similar a como identificamos \mathbb{N}^n con los monomios en $k[x_1, \dots, x_n]$, podemos identificar $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{Z}^m$ con el monomio $t^\alpha = t_1^{\alpha_1} \dots t_m^{\alpha_m}$ en el anillo de polinomios de Laurent $k[t^{\pm 1}] = k[t_1, t_1^{-1}, \dots, t_m, t_m^{-1}]$.

Definición 2.11. Sea k un campo y $\mathcal{A} = \{a_1, \dots, a_n\} \subseteq \mathbb{Z}^m \setminus \{0\}$. Al núcleo del k -homomorfismo de álgebras

$$\varphi : k[x_1, \dots, x_n] \rightarrow k[t_1, t_1^{-1}, \dots, t_m, t_m^{-1}]$$

inducido por $\varphi(x_j) = t^{a_j} = t_1^{a_{1j}} t_2^{a_{2j}} \dots t_m^{a_{mj}}$ donde $a_j = (a_{1j}, a_{2j}, \dots, a_{mj})$, lo llamamos el *ideal tórico* asociado a \mathcal{A} y lo notamos $I_{\mathcal{A}}$.

El conjunto $\mathcal{A} = \{a_1, \dots, a_n\}$ se identifica con las columnas de la matriz $A \in \mathbb{Z}^{m \times n}$. Luego la imagen de la función π que notamos como $\mathbb{N}\mathcal{A}$ es el *monoide generado por \mathcal{A}* . Además, notemos que para $u = (u_1, \dots, u_n)$ se cumple que $\varphi(x^u) = t^{u_1 a_1 + \dots + u_n a_n} = t^{Au} = t^{\pi(u)}$.

De la definición de $I_{\mathcal{A}}$ podemos establecer que un binomio $x^u - x^v \in I_{\mathcal{A}}$ con $u, v \in \mathbb{N}^n$ si y solo si $\pi(u) = \pi(v)$. Un conjunto de generadores de $I_{\mathcal{A}}$ son precisamente este tipo de binomios.

Proposición 2.12. [54, lema 4.1, corolario 4.3] El ideal tórico $I_{\mathcal{A}}$ es generado como un k -espacio vectorial por el conjunto de binomios $\{x^u - x^v : \pi(u) = \pi(v), u, v \in \mathbb{N}^n\}$, por tanto, $I_{\mathcal{A}} = \langle x^u - x^v : \pi(u) = \pi(v) \rangle$.

Todo vector $u \in \mathbb{Z}^n$ lo podemos escribir como una diferencia de dos n -uplas de números naturales con soporte disjunto, $u = u^+ - u^-$, donde u^+ es el vector con $u_i^+ = u_i$ si $u_i > 0$ y $u_i^+ = 0$ en otro caso. Luego si $u \in \mathbb{Z}^n$ tal que $Au = 0$, entonces $x^{u^+} - x^{u^-} \in I_{\mathcal{A}}$.

Iniciando con un conjunto finito de generadores a partir de la proposición anterior, y por tanto binomios, el algoritmo de Buchberger nos seguirá proporcionando binomios durante todo el proceso.

Proposición 2.13. [54, corolario 4.4] Toda base de Gröbner reducida de $I_{\mathcal{A}}$ con respecto a un orden monomial \succeq consiste de un conjunto finito de binomios de la forma $x^u - x^v \in I_{\mathcal{A}}$.

El siguiente teorema establece una relación entre ideales tóricos y el problema de programación entera (2.1) que nos sirve de base para formular un algoritmo que nos permita encontrar la solución óptima de cada programa entero en IP_{A, \succ_c} cuando $b \in \mathbb{N}\mathcal{A}$:

Teorema 2.14. [58, corolario 3.3] Al dividir (reducir) un monomio $x^v \in k[x_1, \dots, x_n]$ por una base de Gröbner reducida de $I_{\mathcal{A}}$ con respecto a un orden monomial \succeq , si notamos $Av = b'$, todo resto parcial durante el algoritmo (y en particular la forma normal de x^v , $nf_{\mathcal{G}_{\succeq}}(x^v)$) es un monomio cuyo exponente está en $P_{Av} \cap \mathbb{N}^n$. Además $nf_{\mathcal{G}_{\succeq}}(x^v) \preceq x^v$.

Demostración. Recordemos que la forma normal de un polinomio $nf_{\mathcal{G}_{\succeq}}(f)$ es el único residuo de la división de f por una base de Gröbner \mathcal{G}_{\succeq} y es reducido con respecto a \mathcal{G}_{\succeq} , esto es, $nf_{\mathcal{G}_{\succeq}}(f) = 0$ o ninguno de sus monomios es divisible por algún monomio principal de un elemento de la base.

Sea $g = x^{u^+} - x^{u^-} \in \mathcal{G}_{\succeq}$, sin pérdida de generalidad asumimos que el monomio principal de g es $in_{\succeq}(g) = x^{u^+}$ tal que x^{u^+} divide a un monomio $x^v \in k[x_1, \dots, x_n]$, esto es, $x^v = x^{u^+} x^\gamma$ para algún $\gamma \in \mathbb{N}^n$. En el primer paso de la división obtenemos como residuo parcial $r = x^v - (x^v/x^{u^+})(x^{u^+} - x^{u^-}) =$

$(x^v/x^{u^+})x^{u^-} = x^{v-u}$ el cual es un monomio en $k[x_1, \dots, x_n]$, pues $v \geq u^+$ y por tanto $v - u \in \mathbb{N}^n$. Como $Au = 0$, se sigue que $Av = A(v - u)$, es decir, $v - u \in P_{Av} \cap \mathbb{N}^n$. Además $x^{v-u} \preceq x^v$. \square

El resultado anterior nos permite describir el algoritmo 3 enunciado a continuación.

Algoritmo 3 Algoritmo para obtener la solución óptima de un programa entero en IP_{A, \succ_c} [54, algoritmo 5.6].

- 1: **Entrada:** Una matriz $A \in \mathbb{Z}^{m \times n}$ y una función de coste $c \in \mathbb{Z}^n$.
 - 2: Calcule la base de Gröbner reducida \mathcal{G}_{\succ_c} de I_A con respecto al orden \succ_c .
 - 3: **Para** algún vector $b \in \mathbb{N}^m$ **hacer:**
 - 4: Encuentre una solución factible v del programa $IP_{A, \succ_c}(b)$.
 - 5: Calcule la forma normal de x^v con respecto a \mathcal{G}_{\succ_c} , esto es $x^{u'} = n_{f_{\succ_c}}(x^v)$.
 - 6: **Fin**
 - 7: **Salida:** El vector u' que es el óptimo de $IP_{A, \succ_c}(b)$.
-

Prueba del algoritmo 3: El cálculo de la base de Gröbner reducida \mathcal{G}_{\succ_c} de I_A con respecto al orden \succ_c es equivalente al cambiar el orden por uno con un vector con todas las componentes positivas, lo que garantiza la finitud de dicha base. Comenzamos en la solución factible v del programa $IP_{A, \succ_c}(b)$, por el teorema 2.14, u' es solución factible de $IP_{A, \succ_c}(b)$. Supongamos que existe w en $P_b \cap \mathbb{N}^n$ tal que $u' \succ_c w$, entonces $x^{u'}$ es el monomio principal de $x^{u'} - x^w \in I_A$ con respecto a \succ_c . Luego existe un binomio en \mathcal{G}_{\succ_c} cuyo monomio principal divide a $x^{u'}$, pero esto contradice que $x^{u'}$ es la forma normal de x^v con respecto a \mathcal{G}_{\succ_c} . Entonces, u' es el óptimo de $IP_{A, \succ_c}(b)$. \square

2.4. Cálculo de \mathcal{T}_{\succ_c} con bases de Gröbner

El algoritmo de Buchberger nos proporciona un método efectivo de resolución de programas enteros lineales. El test-set minimal que hemos descrito es puramente teórico, veamos que dicho test-set es la base de Gröbner del ideal tórico I_A .

Teorema 2.15. El conjunto de vectores $\mathcal{T}_{\succ_c} = \{u_i^+ - u_i^- : i = 1, \dots, p\}$ asociados a la base de Gröbner reducida $\mathcal{G}_{\succ_c} = \{x^{u_i^+} - x^{u_i^-} : A(u_i^+) = A(u_i^-), i = 1, \dots, p\}$ de I_A es el único test set minimal para IP_{A, \succ_c} .

Demostración. Recordemos que la base de Gröbner reducida de $I_{\mathcal{A}}$ es el conjunto $\mathcal{G}_{\succ_c} = \{x^{u_i^+} - x^{u_i^-} : A(u_i^+) = A(u_i^-), i = 1, \dots, p\}$ donde $x^{u_i^+}$ es el monomio principal de cada binomio. Luego el ideal inicial de $I_{\mathcal{A}}$ con respecto a \succ_c es el ideal monomial $in_{\succ_c}(I_{\mathcal{A}}) = \langle x^{u_i^+} : i = 1, \dots, p \rangle$, cuyos monomios están en biyección con los elementos del subconjunto $\bigcup_{i=1}^p (u_i^+ + \mathbb{N}^n)$ de \mathbb{N}^n .

El conjunto de soluciones no óptimas de los programas de IP_{A, \succ_c} es el conjunto $\bigcup_{i=1}^p (u_i^+ + \mathbb{N}^n)$ ya que para cada $b \in \mathbb{N}(\mathcal{A})$ una solución factible u de $IP_{A, \succ_c}(b)$ es óptima si y solo si $x^u \notin in_{\succ_c}(I_{\mathcal{A}})$. Pues del algoritmo 3 tenemos que u es una solución no óptima de $IP_{A, \succ_c}(b)$ si y solo si $nf_{\succ_c}(x^u) \neq x^u$. Y de la definición de base de Gröbner podemos concluir que $nf_{\succ_c}(x^u) \neq x^u$ si y solo si $x^u \in in_{\succ_c}(I_{\mathcal{A}})$. Así, $\mathcal{S}_{\succ_c} = \bigcup_{i=1}^p (u_i^+ + \mathbb{N}^n)$.

Veamos ahora que el conjunto de exponentes líderes $\{u_1^+, \dots, u_p^+\}$ es minimal. Sea $x^{u_i^+} - x^{u_i^-} \in \mathcal{G}_{\succ_c}$ luego $A(u_i^+) = A(u_i^-)$, es decir, u_i^+ y u_i^- están en la misma fibra. Si asumimos que para algún $j \neq i$ se cumple que $u_i^+ = u_j^+ + k$ con $k \in \mathbb{N}^n$, tendríamos que $x^{u_j^+}$ divide a $x^{u_i^+}$, pero esto es una contradicción pues \mathcal{G}_{\succ_c} es una base de Gröbner reducida.

Tan solo nos queda ver que u_i^- es la solución óptima de su fibra. Supongamos que no lo es, por tanto, si $u_i^- \in \mathcal{S}_{\succ_c}$ entonces existe $u_j^+ \in \mathcal{S}_{\succ_c}$ tal que $u_i^- = u_j^+ + v$, esto implica que $x^{u_j^+}$ divide a $x^{u_i^-}$ pero esto es una contradicción pues \mathcal{G}_{\succ_c} es una base de Gröbner reducida. \square

Anteriormente vimos en el teorema 2.14 que la reducción de un monomio x^v por un binomio $x^{u^+} - x^{u^-}$ corresponde a restarle al exponente v el vector $u = u^+ - u^-$. Además, como para todo vector $g_i = u_i^+ - u_i^-$ de \mathcal{T}_{\succ_c} se cumple que $u_i^+ \succ_c u_i^-$, podemos considerar u como el vector orientado de u_i^+ a u_i^- que notamos $\overrightarrow{[u_i^+, u_i^-]}$.

El teorema 2.15 implica que existe un camino dirigido (posiblemente más de uno) desde toda solución no óptima de $IP_{A, \succ_c}(b)$ al único óptimo, que consta de los vectores $\overrightarrow{[u_i^+, u_i^-]}$ con $g_i \in \mathcal{T}_{\succ_c}$. De hecho, los elementos de \mathcal{T}_{\succ_c} construyen un grafo dirigido conexo en cada fibra de IP_{A, \succ_c} , donde los vértices del grafo son todos los puntos factibles y las aristas son los elementos de \mathcal{T}_{\succ_c} . El grafo en $IP_{A, \succ_c}(b)$ es conexo, pues toda solución no óptima de $IP_{A, \succ_c}(b)$ tiene un grado de salida de al menos uno y el grafo tiene un único pozo o sumidero en el único óptimo de la fibra, pues su grado de salida es cero.

En la practica, podemos usar paquetes de algebra computacional como 4ti2 [1] para calcular estos test-set, es decir, para calcular una base de Göbner reducida para $I_{\mathcal{A}}$.

Ejemplo 2.16. Dada la matriz $A = \begin{pmatrix} 1 & 2 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 2 & 0 & 1 \end{pmatrix}$ y el vector de coste

$c = (1, 2, 1, 3, 1)$, siguiendo el algoritmo 3, tenemos que la base de Gröbner reducida del ideal I_A corresponde a $\mathcal{G}_{>c} = \{\underline{x_2x_3} - x_1^2, \underline{x_3x_4} - x_1x_5, \underline{x_1x_4} - x_2x_5\}$ donde el monomio subrayado es el monomio principal. Para $b = (33, 30, 18)$ el punto $(6, 9, 6, 9, 0)$ es un punto factible de $IP_{A, >c}(b)$ y la forma normal del monomio $x_1^6x_2^9x_3^6x_4^9$ con respecto a $\mathcal{G}_{>c}$ es $x_1^9x_2^{12}x_5^9$, por tanto la solución óptima de $IP_{A, >c}(b)$ es $(9, 12, 0, 0, 9)$.

Si notamos $g_1 = \underline{x_2x_3} - x_1^2$, $g_2 = \underline{x_3x_4} - x_1x_5$ y $g_3 = \underline{x_1x_4} - x_2x_5$, algunas reducciones de $x_1^6x_2^9x_3^6x_4^9$ son:

- Al reducir $x_1^6x_2^9x_3^6x_4^9$ por g_3 (tres veces) obtenemos $x_1^3x_2^{12}x_3^6x_4^3$, y al reducir este por g_2 (seis veces) obtenemos $x_1^9x_2^{12}x_5^9$.
- Al reducir $x_1^6x_2^9x_3^6x_4^9$ por g_1 obtenemos $x_1^8x_2^8x_3^5x_4^9$, luego al reducir este por g_2 (cinco veces) obtenemos $x_1^{13}x_2^8x_4^5x_5^9$ y finalmente al reducir este último por g_3 (cuatro veces) obtenemos $x_1^9x_2^{12}x_5^9$.

En términos de vectores, el test-set asociado corresponde a $\mathcal{T}_{>c} = \{g_1 = (-2, 1, 1, 0, 0), g_2 = (-1, 0, 1, 1, -1), g_3 = (1, -1, 0, 1, -1)$ y las dos reducciones anteriores interpretadas como restas, que equivale a seguir el algoritmo 2, son:

- $(6, 9, 6, 9, 0) - g_3 - g_3 - g_3 - g_2 - g_2 - g_2 - g_2 - g_2 - g_2 = (9, 12, 0, 0, 9)$.
- $(6, 9, 6, 9, 0) - g_1 - g_2 - g_2 - g_2 - g_2 - g_2 - g_3 - g_3 - g_3 - g_3 = (9, 12, 0, 0, 9)$.

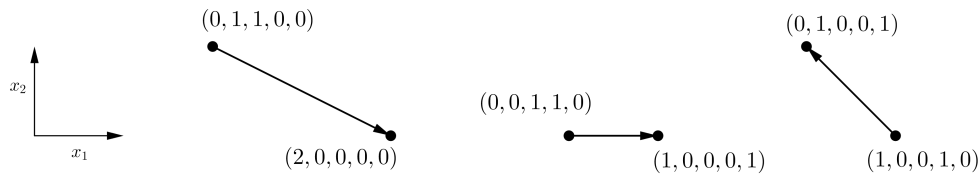


Figura 2.1: Elementos de $\mathcal{T}_{>c}$

En la figura 2.1 mostramos las proyecciones sobre el plano x_1, x_2 de los elementos de $\mathcal{T}_{>c}$. En la figura 2.2 presentamos los dos caminos dirigidos (en azul y verde) relativos a las dos reducciones anteriores de $(6, 9, 6, 9, 0)$, donde los puntos negros son las proyecciones sobre las coordenadas x_1, x_2 de los elementos de la $(33, 30, 18)$ -fibra de $IP_{A, >c}$ y en la figura 2.3 presentamos el grafo completo de la $(33, 30, 18)$ -fibra de $IP_{A, >c}$.

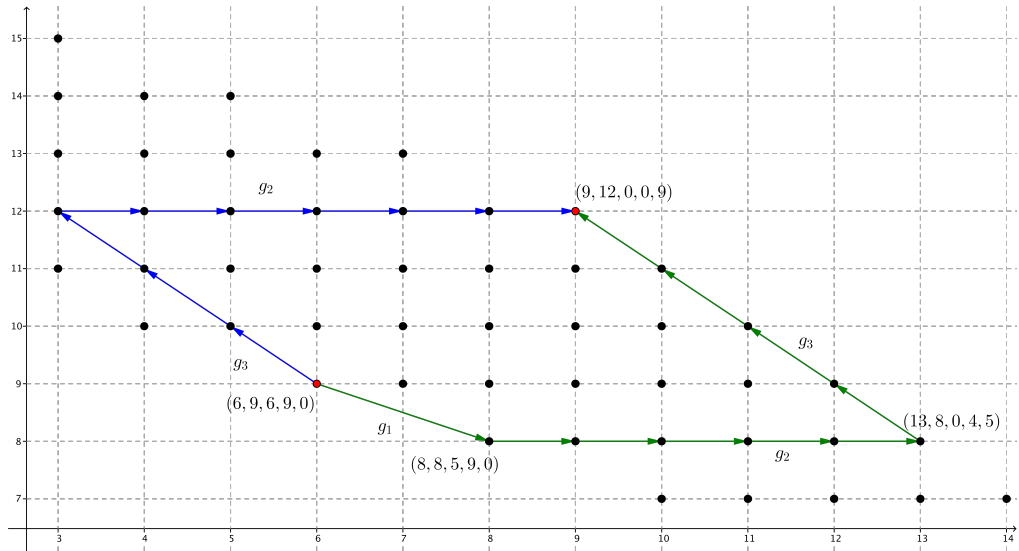


Figura 2.2: Caminos de reducción del punto $(6, 9, 6, 9, 0)$

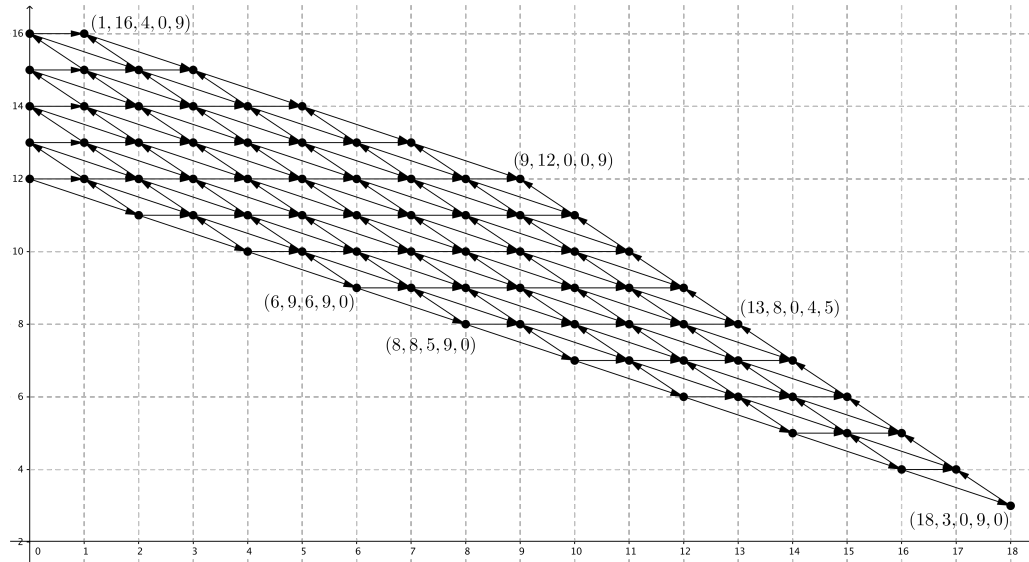


Figura 2.3: Grafo de la $(33, 30, 18)$ -fibra de IP_{A, \succ_c} proyectado en el plano x_1, x_2

2.5. Un ejemplo motivador

En las secciones anteriores, vimos que un problema IP_{A, \succ_c} tiene asociado un test-set, $\mathcal{T}_{\succ_c} \subset \mathbb{Z}^n$, que solo depende de la matriz A y de la función objetivo c , esto es, es válido para todo b . Además, se pudo establecer que \mathcal{T}_{\succ_c} es equivalente a la base de Gröbner reducida con respecto a \succ_c del ideal tórico I_A .

Por otro lado, el método de las ϵ -restricciones nos asegura, que en caso de dos objetivos, para resolver un problema $IP_{A,c}$, es suficiente resolver problemas de la forma $P_2(\epsilon_1)$, para valores adecuados de ϵ_1 , definidos como

$$\begin{aligned} \text{mín} \quad & c_2x \\ \text{s.a.} \quad & Ax = b \\ & c_1x \leq \epsilon_1, \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \tag{2.4}$$

Es decir, debemos resolver varios problemas que pertenecen a una misma familia de problemas enteros, donde la matriz de restricciones y el vector objetivo son los mismos. Este hecho nos lleva a establecer una conexión directa con el uso de test-set para su solución. Veamos un ejemplo para ilustrar el uso de los test-set en este caso.

Ejemplo 2.17. Consideremos el problema

$$\begin{aligned} \text{mín} \quad & c_1x = x_1, c_2x = x_2 \\ \text{s.a.} \quad & x_1 + 3x_2 \leq 23 \\ & 4x_1 + x_2 \leq 33 \\ & -x_1 - 4x_2 \leq -12 \quad x \in \mathbb{Z}_{\geq 0}^2 \end{aligned} \tag{2.5}$$

cuyo conjunto factible y conjunto objetivo coinciden y que se encuentra representado en la figura 2.4. Para obtener la frontera de Pareto con el método de las ϵ -restricciones, resolvemos en particular $P_2(\epsilon_1)$ que reescribimos en su forma estándar como:

$$\begin{aligned} \text{mín} \quad & c_2x = x_2 \\ \text{s.a.} \quad & x_1 + 3x_2 + t_1 = 23 \\ & 4x_1 + x_2 + t_2 = 33 \\ & -x_1 - 4x_2 + t_3 = -12 \\ & c_1x + h = \epsilon_1, \quad x \in \mathbb{Z}_{\geq 0}^6 \end{aligned} \tag{2.6}$$

y así obtener, con la ayuda de 4ti2, que el test-set de esta familia de problemas

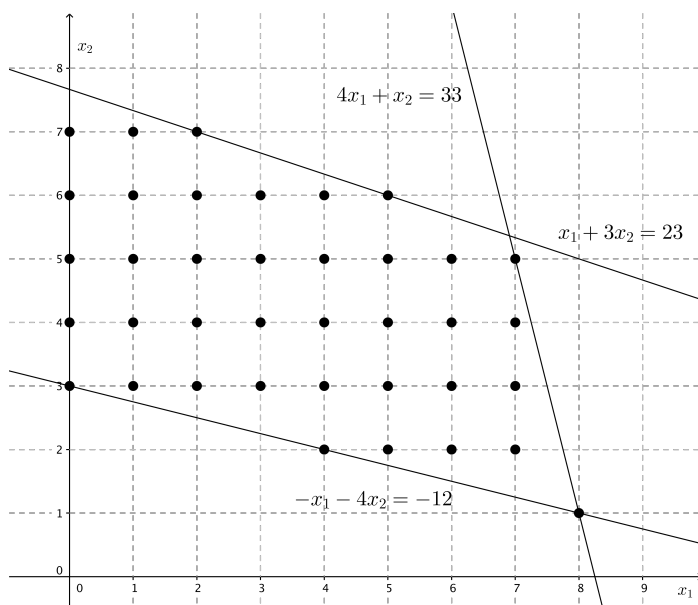


Figura 2.4: Conjunto objetivo del ejemplo 2.17.

asociado al coste c_2 y a la matriz $\begin{pmatrix} A_{3 \times 2} & I_3 & 0 \\ c_1 & 0 & 1 \end{pmatrix}_{4 \times 6}$ es

$$\begin{aligned} \mathcal{T}_{\preceq_{c_2}} = \{ & g_1 = (-3, 1, 0, 11, 1, 3), \\ & g_2 = (-2, 1, -1, 7, 2, 2), \\ & g_3 = (-1, 0, 1, 4, -1, 1), \\ & g_4 = (-1, 1, -2, 3, 3, 1), \\ & g_5 = (0, 1, -3, -1, 4, 0) \} \end{aligned}$$

En la sección 1.4.2 cuando describimos el método de las ϵ -restricciones, señalamos que como valor de inicio para ϵ_1 podíamos utilizar la primera componente del vector ideal. Así que al resolver $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$, con `4ti2` con respecto al orden \preceq_{c_1} obtenemos como punto óptimo $(0, 7, 2, 26, 16)$, es decir, el valor inicial para $\epsilon_1 = 0$.

Para resolver el problema $P_2(0)$ utilizando el test-set, debemos encontrar una solución factible del problema, y una opción es utilizar el óptimo del problema anterior cuyo valor en c_1 es ϵ_1 , por tanto, la última variable de holgura debe ser igual 0, esto es, el punto $(0, 7, 2, 26, 16, 0)$ es factible para $P_2(0)$ y de su reducción obtenemos como óptimo del problema al punto $(0, 3, 14, 30, 0, 0)$.

Ahora a partir de $\epsilon_1 = 0$ vamos realizando incrementos de uno en uno. Luego debemos resolver el problema $P_2(1)$. Como debemos partir de una so-

lución factible, para luego utilizar el test-set y encontrar el óptimo, podemos utilizar el punto óptimo anterior e incrementar en uno la última variable de holgura, esto es, $(0, 3, 14, 30, 0, 1)$ es un punto factible para $P_2(1)$ y al realizar la reducción obtenemos a $(1, 3, 13, 26, 1, 0)$ como óptimo.

Nuevamente incrementamos el valor de ϵ_1 en uno, lo que nos conduce al problema $P_2(2)$ con $(1, 3, 13, 26, 1, 1)$ como punto factible y cuya reducción es el punto óptimo $(2, 3, 12, 22, 2, 0)$.

De manera similar seguimos el proceso para los siguientes valores de ϵ_1 :

- Para $P_2(3)$, con solución inicial $(2, 3, 12, 22, 2, 1)$ obtenemos como óptimo $(3, 3, 11, 18, 3, 0)$.
- Para $P_2(4)$, con solución inicial $(3, 3, 11, 18, 3, 1)$ obtenemos como óptimo $(4, 2, 13, 15, 0, 0)$.
- Para $P_2(5)$, con solución inicial $(4, 2, 13, 15, 0, 1)$ obtenemos como óptimo $(5, 2, 12, 11, 1, 0)$.
- Para $P_2(6)$, con solución inicial $(5, 2, 12, 11, 1, 1)$ obtenemos como óptimo $(6, 2, 11, 7, 2, 0)$.
- Para $P_2(7)$, con solución inicial $(6, 2, 11, 7, 2, 1)$ obtenemos como óptimo $(7, 2, 10, 3, 3, 0)$.
- Para $P_2(8)$, con solución inicial $(7, 2, 10, 3, 3, 1)$ obtenemos como óptimo $(8, 1, 12, 0, 0, 0)$.

Ahora, si seguimos incrementando el valor de ϵ_1 de uno en uno, no obtenemos un nuevo punto como óptimo, es decir, si consideramos el problema $P_2(9)$ con solución inicial $(8, 1, 12, 0, 0, 1)$, la reducción es el mismo punto o si resolvemos el problema $P_2(10)$ con solución inicial $(8, 1, 12, 0, 0, 2)$, la reducción es el mismo punto. De hecho, podemos decir que el proceso termina al resolver $P_2(8)$, pues al modificar la última componente de $(8, 1, 12, 0, 0, 0)$, es decir, al incrementar el valor de ϵ_1 en 9 o más, ningún vector del test-set lo va a reducir. La figura 2.5 ilustra este proceso, donde los puntos azules corresponden a los óptimos obtenidos en cada problema.

Visto de otra manera, con esta estrategia no necesitamos calcular el vector nadir para establecer el máximo valor de incremento de ϵ_1 , pues el test-set nos indica cuándo detenernos. Entonces, como resultado hemos obtenido nueve soluciones factibles del problema (2.5), de las cuales solo tres son soluciones eficientes, es decir, $\mathcal{X}_E = \{(0, 3), (4, 2), (8, 1)\}$.

Una manera de intentar evitar obtener soluciones débilmente eficientes, es modificar el orden \preceq_{c_2} y cambiarlo por un orden, que notamos \preceq_{c_2, c_1} ,

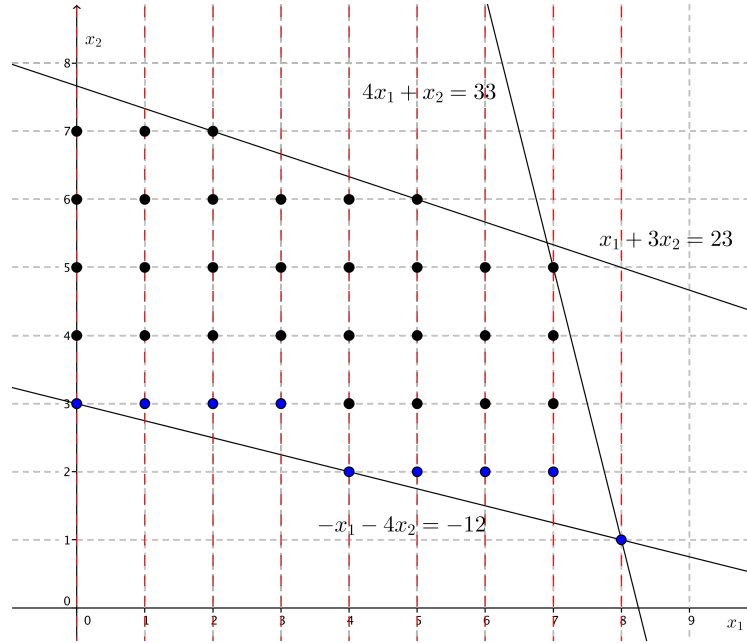


Figura 2.5: Ilustración de aplicación del método de las ϵ -restricciones al ejemplo 2.17 utilizando el test-set $\mathcal{T}_{\preceq_{c_2}}$ con incrementos de ϵ_1 de uno en uno.

que primero compara con respecto a c_2 ; en caso de igualdad compara según c_1 y si persiste la igualdad compara con un orden monomial (4ti2 utiliza concretamente el lexicográfico reverso graduado). De manera general, sean $\alpha, \beta \in \mathbb{N}^n$

$$\alpha \preceq_{c_2, c_1} \beta \text{ si } \begin{cases} c_2 \cdot \alpha < c_2 \cdot \beta & \text{o} \\ c_2 \cdot \alpha = c_2 \cdot \beta \text{ y } c_1 \cdot \alpha < c_1 \cdot \beta & \text{o} \\ c_2 \cdot \alpha = c_2 \cdot \beta, c_1 \cdot \alpha = c_1 \cdot \beta & \text{y } \alpha \preceq \beta \end{cases} \quad (2.7)$$

Notemos que \preceq_{c_2, c_1} es un orden total de tipo producto como vimos en la sección 2.2.

Con esta modificación del orden, el test-set para la familia de problemas $P_2(\epsilon_1)$ es:

$$\begin{aligned} \mathcal{T}_{\preceq_{c_2, c_1}} = \{ & g_1 = (-3, 1, 0, 11, 1, 3), \\ & g_2 = (-2, 1, -1, 7, 2, 2), \\ & -g_3 = (1, 0, -1, -4, 1, -1), \\ & g_4 = (-1, 1, -2, 3, 3, 1), \\ & g_5 = (0, 1, -3, -1, 4, 0), \\ & g_6 = (-4, 1, 1, 15, 0, 4) \} \end{aligned}$$

Partiendo de $(0, 7, 2, 26, 16, 0)$ y resolviendo $P_2(0)$ obtenemos como óptimo a $(0, 3, 14, 30, 0, 0)$, luego siguiendo con el incremento de uno en uno de ϵ_1 , tenemos que (figura 2.6):

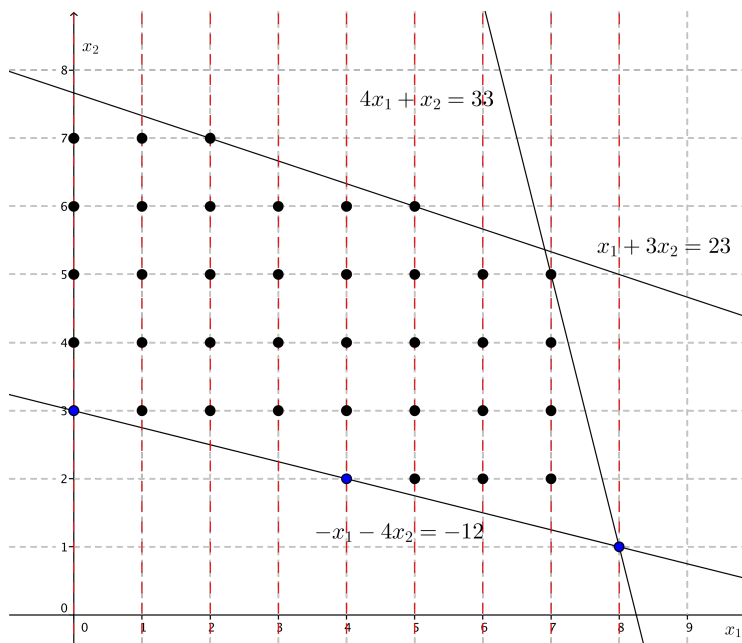


Figura 2.6: Ilustración de aplicación del método de las ϵ -restricciones al ejemplo 2.17 utilizando el test-set $\mathcal{T}_{\leq c_2, c_1}$ con incrementos de ϵ_1 de uno en uno.

- Para $P_2(1)$, con solución inicial $(0, 3, 14, 30, 0, 1)$ obtenemos como óptimo $(0, 3, 14, 30, 0, 1)$.
- Para $P_2(2)$, con solución inicial $(0, 3, 14, 30, 0, 2)$ obtenemos como óptimo $(0, 3, 14, 30, 0, 2)$.
- Para $P_2(3)$, con solución inicial $(0, 3, 14, 30, 0, 3)$ obtenemos como óptimo $(0, 3, 14, 30, 0, 3)$.
- Para $P_2(4)$, con solución inicial $(0, 3, 14, 30, 0, 4)$ obtenemos como óptimo $(4, 2, 13, 15, 0, 0)$.
- Para $P_2(5)$, con solución inicial $(4, 2, 13, 15, 0, 1)$ obtenemos como óptimo $(4, 2, 13, 15, 0, 1)$.
- Para $P_2(6)$, con solución inicial $(4, 2, 13, 15, 0, 2)$ obtenemos como óptimo $(4, 2, 13, 15, 0, 2)$.

- Para $P_2(7)$, con solución inicial $(4, 2, 13, 15, 0, 3)$ obtenemos como óptimo $(4, 2, 13, 15, 0, 3)$.
- Para $P_2(8)$, con solución inicial $(4, 2, 13, 15, 0, 4)$ obtenemos como óptimo $(8, 1, 12, 0, 0, 0)$.

En los problemas $P_2(1)$, $P_2(2)$ y $P_2(3)$ el óptimo corresponde a la solución inicial dada, pues no hay reducción hasta llegar a $P_2(4)$ donde el punto $(0, 3, 14, 30, 0, 4)$ es reducible por g_6 . Cuando llegamos a un punto que no es reducible para ningún valor de la variable de holgura, como ocurre con el punto $(8, 1, 12, 0, 0, 0)$, termina el proceso.

Es decir, el test-set nos ha indicado qué cálculos son innecesarios, pues finalmente obtenemos las tres soluciones eficientes esperadas, $\mathcal{X}_E = \{(0, 3), (4, 2), (8, 1)\}$. En este sentido, podemos incrementar ϵ_1 de acuerdo con la información que nos señala el test-set, como mostramos a continuación y cuya ilustración se encuentra en la figura 2.7.

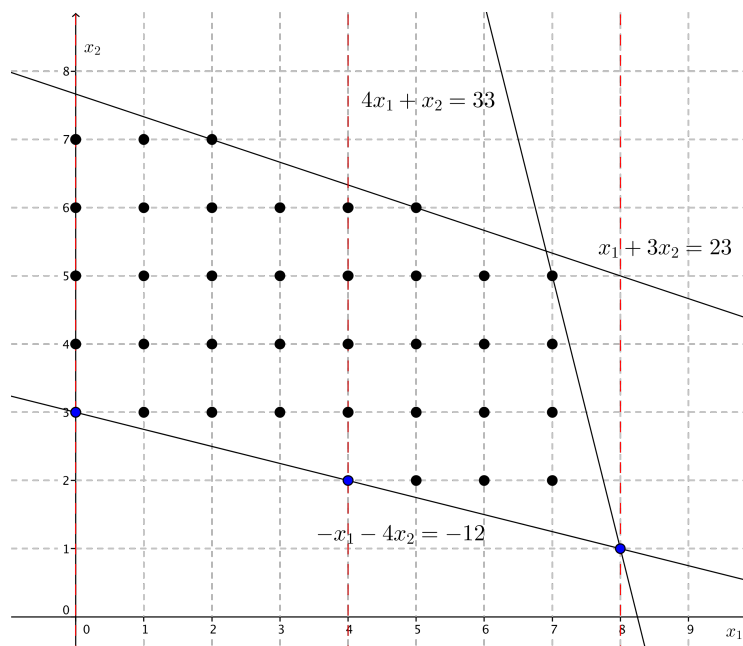


Figura 2.7: Ilustración de aplicación del método de las ϵ -restricciones al ejemplo 2.17 utilizando el test-set $\mathcal{T}_{\leq \epsilon_2, \epsilon_1}$ con incrementos de ϵ_1 dados por dicho test-set.

- Partiendo de $(0, 7, 2, 26, 16, 0)$ y resolviendo $P_2(0)$ obtenemos como óptimo a $(0, 3, 14, 30, 0, 0)$.

- Luego, según nos indica el test-set, hasta que no incrementemos la última variable de holgura en 4 no hay reducción posible. Por tanto, debemos resolver el problema $P_2(4)$, con solución inicial $(0, 3, 14, 30, 0, 4)$ y cuyo óptimo es $(4, 2, 13, 15, 0, 0)$.
- Siguiendo con la estrategia para modificar el valor de ϵ_1 , el siguiente incremento en el que se presenta una reducción es 4. Entonces, debemos resolver el problema $P_2(8)$ donde la reducción de la solución inicial $(4, 2, 13, 15, 0, 4)$ es el punto óptimo $(8, 1, 12, 0, 0, 0)$, con el que, como hemos señalado antes, terminamos.

Hemos pasado de resolver 9 problemas de la familia $P_2(\epsilon_1)$ a 3. También podemos evitar el problema $P_2(0)$ si modificamos el orden utilizado para encontrar una solución de $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$, es decir, si utilizamos el orden \preceq_{c_1, c_2} , pues de todos los puntos con igual coste c_2 obtenemos el que tiene menor coste c_1 , que en este caso es $(0, 3, 14, 30, 0)$. Observemos que finalmente hemos necesitado resolver tantos problemas como puntos no dominados hay en la frontera de Pareto.

En la siguiente capítulo veremos que los resultados que arroja este procedimiento se pueden aplicar de manera general para cualquier problema de optimización lineal entera con dos objetivos para el que se pueda calcular el test-set correspondiente.

CAPÍTULO 3

Test-sets para resolver problemas de optimización lineal entera biobjetivo

En el ejemplo 2.17 señalamos que, de manera natural, el uso de los test-sets permitía dos mejoras importantes: 1) usando el orden producto obtuvimos solo soluciones eficientes y, 2) detectamos los problemas innecesarios. En este capítulo, veremos los resultados teóricos que proporcionan un algoritmo que posee estas dos ventajas. Esto es especialmente interesante, debido a que el método clásico de las ϵ -restricciones no proporciona una estrategia eficiente para elegir los problemas a resolver. De hecho, varios trabajos en optimización lineal entera multiobjetivo que utilizan este método como [11], [40], [49], [45], [46], [35] y [60], lo han adaptado, aumentando la función de coste, cambiando el valor de ϵ en cada problema, e integrando heurísticas.

3.1. Motivación: el problema BBV

En [40] se presenta un ejemplo de optimización lineal entera biobjetivo llamado *problema BBV*. Para dicho problema con el método de las ϵ -restricciones en su formulación original (que en el caso discreto consiste en incrementos de 1, cf. [11]), el número de problemas a resolver es exponencial en el número de variables de decisión n , mientras que el tamaño de la frontera de Pareto es lineal.

El problema BBV consiste en maximizar la función *pseudobooleana* BBV :

$\{0, 1\}^n \rightarrow \mathbb{N}^2$ definida como

$$\text{BBV}(x_1, \dots, x_n) = \left(\text{BBV}_1(x) = \sum_{i=1}^n 2^{n-i} x_i, \text{BBV}_2(x) = \sum_{i=1}^n 2^{i-1} (1 - x_i) \right)$$

que es una generalización de la función BV propuesta en [22], pues esta última corresponde a la función BBV_1 . El siguiente resultado nos indica que la frontera de Pareto del problema BBV de tamaño n , tiene exactamente $n + 1$ elementos y además describe cómo es cada uno de ellos.

Teorema 3.1. [40, Proposición 1]. Dado el problema BBV de tamaño n ,

$$\begin{aligned} \text{máx} \quad & \left(\sum_{i=1}^n 2^{n-i} x_i, \sum_{i=1}^n 2^{i-1} (1 - x_i) \right) \\ \text{s.a.} \quad & x \in \{0, 1\}^n \end{aligned} \quad (3.1)$$

una solución factible $x^* \in \mathcal{X}$ es eficiente para el problema 3.1 si y solo si tiene la forma $(\underbrace{1, \dots, 1}_k, \underbrace{0, \dots, 0}_{n-k})$ con $k = 0, 1, \dots, n$.

Demostración. Sea $x^* = (x_1^*, \dots, x_n^*)$ una solución eficiente para el problema 3.1 y supongamos que no tiene la forma mencionada. Luego, existe un k tal que $x_k^* = 0$ y $x_{k+1}^* = 1$, esto es,

$$x^* = (\underbrace{1, \dots, 1}_{k-1}, 0, 1, \underbrace{0, \dots, 0}_{n-k-1})$$

con $\text{BBV}_1 = 2^{n-1} + \dots + 2^{n-(k-1)} + 2^{n-(k+1)}$ y $\text{BBV}_2 = 2^{k-1} + 2^{k+1} + \dots + 2^{n-1}$. Pero si intercambiamos esas dos coordenadas, la solución factible $x' = (\underbrace{1, \dots, 1}_k, \underbrace{0, \dots, 0}_{n-k})$ tiene valores en BBV_1 y BBV_2 mayores que x^* . Por tanto x^* no puede ser una solución eficiente.

Ahora supongamos que x^* es una solución factible que tiene la forma $(\underbrace{1, \dots, 1}_k, \underbrace{0, \dots, 0}_{n-k})$ y es dominada por otra solución factible x' , entonces el valor de BBV_1 o el valor de BBV_2 deben ser más grandes que los respectivos valores objetivo de x^* . Si $\text{BBV}_1(x') > \text{BBV}_1(x^*)$ entonces x' debe tener las primeras k componentes iguales a 1 y un número arbitrario de componentes iguales a 1 en las $n-k$ restantes; pero esto implica que $\text{BBV}_2(x') < \text{BBV}_2(x^*)$. Si $\text{BBV}_2(x') > \text{BBV}_2(x^*)$ entonces x' debe tener las últimas $n-k$ componentes iguales a 0 y un número arbitrario de componentes iguales a 0 en las k primeras; pero esto implica que $\text{BBV}_1(x') < \text{BBV}_1(x^*)$.

En ambos casos, x' no domina a x^* , lo cual contradice lo que asumimos. \square

La tabla 3.1, presentada en [40], compara el número de problemas que resuelve su método y el de las ϵ -restricciones clásico para encontrar la frontera de Pareto (\mathcal{F}). Cabe resaltar que en [40] cada problema es en realidad un problema de maximización lexicográfica, luego para dos objetivos implica resolver dos problemas estándar, en el sentido del planteamiento original [15, p. 285].

n	$ F $	Método [40]	ϵ -restricción clásico
2	3	4	5
4	5	6	17
10	11	12	1025
20	21	22	1048577
40	41	42	
80	81	82	

Tabla 3.1: Problema BBV: número de subproblemas utilizados por el método de ϵ -restricciones original y uno adaptado [40].

Veremos que este problema es especialmente adecuado para aplicar el algoritmo que proponemos en este capítulo.

3.2. Caracterización de soluciones eficientes

En esta sección presentamos los resultados teóricos que fundamentan el algoritmo que, como mencionamos antes, utiliza el método de las ϵ -restricciones resolviendo los problemas correspondientes con los test-sets. Podemos decir que el algoritmo es exacto en dos sentidos: encuentra toda la frontera de Pareto y lo hace resolviendo tantos problemas como elementos hay en dicha frontera.

Asumimos el problema de optimización lineal entera biobjetivo en su forma estándar, esto es,

$$\begin{aligned} \text{mín} \quad & c_1x, c_2x \\ \text{s.a.} \quad & Ax = b \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (3.2)$$

Para resolver el problema (3.2) hay que resolver problemas de la forma

$$\begin{aligned} \text{mín} \quad & c_2x \\ \text{s.a.} \quad & Ax = b \\ & c_1x \leq \epsilon_1, \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (3.3)$$

que en forma estándar corresponde a

$$\begin{aligned} \text{mín} \quad & c_2x \\ \text{s.a.} \quad & Ax = b \\ & c_1x + h = \epsilon_1, \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (3.4)$$

y notamos $P_2(\epsilon_1)$. Llamamos $\mathcal{T}_{\preceq_{c_2, c_1}} \subset \mathbb{Z}^{n+1}$ al test-set asociado a la familia de problemas $P_2(\epsilon_1)$ con respecto al orden total \preceq_{c_2, c_1} que se definió en (2.7).

Un problema $P_2(\epsilon_1)$ puede tener varias soluciones con el mismo coste c_2 , pero cuando hablemos del problema $P_2(\epsilon_1)_{\preceq_{c_2, c_1}}$ entenderemos que estamos seleccionando la única solución con respecto al orden total \preceq_{c_2, c_1} .

El siguiente teorema nos muestra que utilizando el orden total \preceq_{c_2, c_1} solo obtenemos puntos eficientes.

Teorema 3.2. $(x^*, 0)$ es solución del problema $P_2(\epsilon_1^*)_{\preceq_{c_2, c_1}}$ con $\epsilon_1^* = c_1(x^*)$ si y solo si x^* es solución eficiente del problema biobjetivo (3.2) y de las soluciones eficientes con costes $(c_1(x^*), c_2(x^*))$ es la menor según el orden \preceq_{c_2, c_1} .

Demostración. \Leftarrow) Sea x^* solución eficiente del problema biobjetivo (3.2) tal que, de las soluciones eficientes con igual coste c_1, c_2 , es la menor según el orden \preceq_{c_2, c_1} . Por el teorema 1.15 tenemos que x^* es solución óptima de $P_2(c_1(x^*))$. Si dicho problema tiene otras soluciones, estas deben tener igual coste c_2 , y por tanto igual coste c_1 que x^* , pues este punto es eficiente. Como x^* es la menor de ellas según \preceq_{c_2, c_1} , concluimos que $(x^*, 0)$ es solución del problema $P_2(c_1(x^*))_{\preceq_{c_2, c_1}}$.

\Rightarrow) Ahora, sea $(x^*, 0)$ solución de $P_2(c_1(x^*))_{\preceq_{c_2, c_1}}$, luego $(x^*, 0)$ también es solución de $P_2(c_1(x^*))$. Para mostrar que x^* es una solución eficiente de (3.2), por el teorema 1.15, es suficiente probar que $(x^*, 0)$ es solución del problema $P_1(c_2(x^*))$

$$\begin{aligned} \text{mín} \quad & c_1x \\ \text{s.a.} \quad & Ax = b \\ & c_2x + h = c_2(x^*), \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (3.5)$$

Tenemos que $(x^*, 0)$ es factible para (3.5). Por reducción al absurdo, supongamos que $(x^*, 0)$ no es una solución óptima de dicho problema. Entonces sea (x', t) con $t \geq 0$, una solución de (3.5), por tanto $c_1(x') < c_1(x^*)$ y $c_2(x') + t = c_2(x^*)$, es decir, $c_2(x') \leq c_2(x^*)$. Estas desigualdades nos permiten afirmar que existe $s > 0$ tal que (x', s) es factible para $P_2(c_1(x^*))_{\preceq_{c_2, c_1}}$ y que $(x', s) \preceq_{c_2, c_1} (x^*, 0)$ lo que contradice que $(x^*, 0)$ es la única solución de $P_2(c_1(x^*))_{\preceq_{c_2, c_1}}$. Por tanto $(x^*, 0)$ es solución del problema (3.5). Además, las soluciones eficientes con iguales costes que x^* también son factibles para $P_2(\epsilon_1^*)_{\preceq_{c_2, c_1}}$, pero como $(x^*, 0)$ es la única solución de dicho problema, esto implica que es la menor de ellas según el orden \preceq_{c_2, c_1} . \square

Una aplicación del teorema anterior nos permitirá calcular una solución eficiente con el menor coste c_1 , que utilizaremos como punto de inicio en el algoritmo.

Proposición 3.3. Sea \tilde{x}_1 la solución del problema $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$ con respecto al orden \preceq_{c_1, c_2} . Entonces \tilde{x}_1 es una solución eficiente del problema biobjetivo (3.2).

Demostración. Sea \tilde{x}_1 la solución del problema $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$ con respecto al orden \preceq_{c_1, c_2} , esto es, el punto $(\tilde{x}_1, 0)$ es la única solución del problema $P_1(c_2(e_1))_{\preceq_{c_1, c_2}}$. Por el teorema 3.2 aplicado a $P_1(\epsilon_2)_{\preceq_{c_1, c_2}}$ concluimos que \tilde{x}_1 es una solución eficiente del problema biobjetivo (3.2). \square

Como el conjunto factible $\mathcal{X} \neq \emptyset$ para el problema biobjetivo (3.2) lo hemos considerado finito, la frontera de Pareto \mathcal{F} es un conjunto finito. Definimos $\tilde{\mathcal{X}}$ como el conjunto

$$\tilde{\mathcal{X}} = \{\tilde{x} \in \mathcal{X} : (\exists z = (z_1, z_2) \in \mathcal{F})((c_1(\tilde{x}), c_2(\tilde{x})) = (z_1, z_2) \text{ y } (\tilde{x}, 0) \text{ es solución de } P_2(c_1(\tilde{x}))_{\preceq_{c_2, c_1}})\} \quad (3.6)$$

Esto es, para cada punto de \mathcal{F} , en el conjunto $\tilde{\mathcal{X}}$ tenemos una solución eficiente que es la menor según el orden \preceq_{c_2, c_1} de todas las que producen el mismo punto no dominado. Si $\tilde{\mathcal{X}} = \{\tilde{x}_1, \dots, \tilde{x}_r\}$ está ordenado de forma creciente por los costes en c_1 , en el teorema siguiente vamos a mostrar que en el test-set $\mathcal{T}_{\preceq_{c_2, c_1}}$ existe un elemento que permite obtener una solución a partir de la anterior.

Cada elemento $\mathbf{g} = (g_1, \dots, g_n, g_{n+1}) \in \mathcal{T}_{\preceq_{c_2, c_1}}$ lo notamos $[g, a]$ con $g = (g_1, \dots, g_n)$ y $a = g_{n+1}$.

Teorema 3.4. Sea $\tilde{\mathcal{X}} = \{\tilde{x}_1, \dots, \tilde{x}_r\}$ definido como en (3.6) y $\mathcal{T}_{\preceq_{c_2, c_1}} \subset \mathbb{Z}^{n+1}$ el test-set asociado a la familia de problemas $P_2(\epsilon_1)_{\preceq_{c_2, c_1}}$. Entonces para todo $i = 1, \dots, r-1$ existe $[g_i, a_i] \in \mathcal{T}_{\preceq_{c_2, c_1}}$ tal que $g_i \leq x_i$ y $a_i = c_1(\tilde{x}_{i+1}) - c_1(\tilde{x}_i)$.

Demostración. Dado $\tilde{x}_i \in \tilde{\mathcal{X}}$ tenemos que $(\tilde{x}_i, 0)$ es la única solución del problema $P_2(c_1(\tilde{x}_i))_{\preceq_{c_2, c_1}}$, luego no existe un elemento en el test-set que lo reduzca, en particular, no existe $[g, a] \in \mathcal{T}_{\preceq_{c_2, c_1}}$ con $a \leq 0$ tal que $g \leq \tilde{x}_i$.

Por otro lado, también tenemos que $(\tilde{x}_{i+1}, 0)$ es la única solución del problema $P_2(c_1(\tilde{x}_{i+1}))_{\preceq_{c_2, c_1}}$ con (\tilde{x}_i, a_i) factible para dicho problema, donde $a_i = c_1(\tilde{x}_{i+1}) - c_1(\tilde{x}_i)$. Como el punto (\tilde{x}_i, a_i) no es el óptimo para $P_2(c_1(\tilde{x}_{i+1}))_{\preceq_{c_2, c_1}}$, existe un elemento del test-set $\mathbf{g} = [g_i, a']$ que lo reduce, es decir, $\mathbf{g} \leq (\tilde{x}_i, a_i)$, con $a' > 0$, pues de lo contrario $[g_i, a']$ reduciría a $(\tilde{x}_i, 0)$ y esto es una contradicción.

Veamos que $a_i = a'$. Para esto, supongamos lo contrario, que $a_i \neq a'$. Como tenemos que $\mathbf{g} \leq (\tilde{x}_i, a_i)$, entonces debe cumplirse que $0 < a' < a_i$. Si consideramos el problema $P_2(c_1(\tilde{x}_i) + a')_{\preceq_{c_2, c_1}}$, tenemos que (\tilde{x}_i, a') es factible para este problema, pero no es solución óptima de él, pues $[g_i, a']$ lo reduce. Por tanto, el punto $(\tilde{x}_i - g_i, 0)$ es factible para $P_2(c_1(\tilde{x}_i) + a')_{\preceq_{c_2, c_1}}$ con $(\tilde{x}_i - g_i, 0) \preceq_{c_2, c_1} (\tilde{x}_i, a')$ lo que implica que $c_2(\tilde{x}_i - g_i) \leq c_2(\tilde{x}_i)$.

Además, se cumple que $c_1(\tilde{x}_i) < c_1(\tilde{x}_i - g_i) < c_1(\tilde{x}_{i+1})$ pues $c_1(\tilde{x}_i - g_i) = c_1(\tilde{x}_i) + a'$ y $a' > 0$, y estamos suponiendo que $a' < a_i$. Esta desigualdad, junto con la definición del conjunto $\tilde{\mathcal{X}}$ nos permite afirmar que no hay un punto en la frontera de Pareto cuyo valor de c_1 sea igual al de $\tilde{x}_i - g_i$. Por tanto, $\tilde{x}_i - g_i$ es dominado por alguna solución eficiente \tilde{x}_k de $\tilde{\mathcal{X}}$ con $k \leq i$, pues $c_1(\tilde{x}_i) < c_1(\tilde{x}_i - g_i)$. Entonces tenemos que $c_1(\tilde{x}_k) < c_1(\tilde{x}_i - g_i)$ y $c_2(\tilde{x}_k) \leq c_2(\tilde{x}_i - g_i)$, luego se deduce la igualdad $c_2(\tilde{x}_i - g_i) = c_2(\tilde{x}_i)$. Debido a que $(\tilde{x}_i - g_i, 0) \preceq_{c_2, c_1} (\tilde{x}_i, a')$, concluimos que $c_1(\tilde{x}_i - g_i) \leq c_1(\tilde{x}_i)$, lo que nos conduce a una contradicción. Entonces, $a_i = a'$. \square

3.3. Algoritmo para el caso biobjetivo

En esta sección presentamos finalmente el algoritmo para problemas de optimización lineal entera biobjetivo, basado en el método de las ϵ -restricciones, y que gracias al uso de los test-sets solo obtenemos soluciones eficientes sin resolver ningún problema innecesario. La siguiente proposición nos permite identificar qué valor de ϵ determina un problema cuyo óptimo es una nueva solución eficiente. Pero antes, para cada $x \in \mathbb{N}^n$ definimos el conjunto $G_x = \{[g, a] \in \mathcal{T}_{\preceq_{c_2, c_1}} : g \leq x\}$. Recordemos que si $(x, 0)$ es el óptimo de $P_2(\epsilon_1)_{\preceq_{c_2, c_1}}$ para algún ϵ_1 con $x \in \mathbb{N}^n$ y $G_x \neq \emptyset$, entonces para todo $[g, a] \in G_x$ se tiene que $a > 0$ (si no, se podría reducir y no sería óptimo).

Proposición 3.5. Sea el punto $(x^*, 0)$ solución del problema $P_2(c_1(x^*))_{\preceq_{c_2, c_1}}$ y $G_{x^*} \neq \emptyset$ con a_0 el mínimo de los a tales que $[g, a] \in G_{x^*}$. Entonces se cumple que:

1. Para todo $0 < a' < a_0$ la solución del problema $P_2(c_1(x^*) + a')_{\preceq_{c_2, c_1}}$ es (x^*, a') .
2. Para a_0 la solución del problema $P_2(c_1(x^*) + a_0)_{\preceq_{c_2, c_1}}$ es un punto de la forma $(x', 0)$ (o sea una nueva solución eficiente).

Demostración. 1. El punto (x^*, a') es factible para $P_2(c_1(x^*) + a')_{\preceq_{c_2, c_1}}$ y supongamos que no es el óptimo de dicho problema, entonces existe $[g, a] \in \mathcal{T}_{\preceq_{c_2, c_1}}$ tal que $g \leq x^*$ y $a \leq a' < a_0$, pero esto contradice la minimalidad de a_0 .

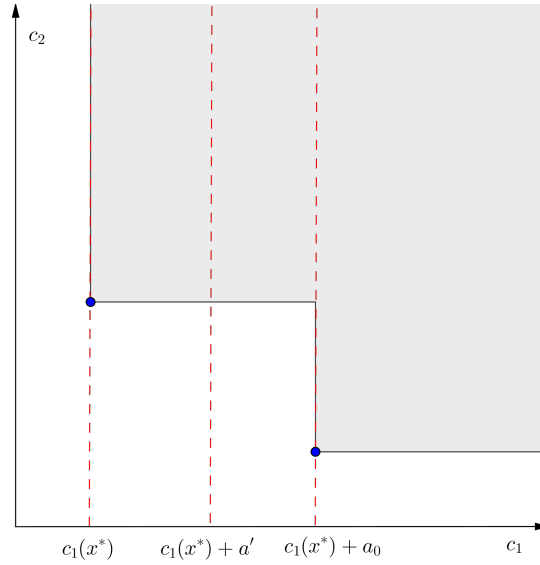


Figura 3.1: Ilustración de la proposición 3.5

2. El punto (x^*, a_0) es factible para $P_2(c_1(x^*) + a_0)_{\preceq_{c_2, c_1}}$ pero no es la solución óptima del problema, pues existe un elemento del test-set $\mathbf{g} = [g, a_0]$ que lo reduce, es decir, $\mathbf{g} \leq (x^*, a_0)$, por tanto, el punto $(x^* - g, 0)$ es factible para $P_2(c_1(x^*) + a_0)_{\preceq_{c_2, c_1}}$ con $(x^* - g, 0) \preceq_{c_2, c_1} (x^*, a_0)$ lo que implica que $c_2(x^* - g) \leq c_2(x^*)$. Como $c_1(x^* - g) = c_1(x^*) + a_0$ y $a_0 > 0$ entonces $c_1(x^*) < c_1(x^* - g)$. Esta última desigualdad, junto con la definición del orden \preceq_{c_2, c_1} , nos permiten concluir que $c_2(x^* - g) < c_2(x^*)$.

Ahora, supongamos que la solución de $P_2(c_1(x^*) + a_0)_{\preceq_{c_2, c_1}}$ es un punto de la forma (x', t) con $t > 0$, entonces $c_1(x') + t = c_1(x^*) + a_0 = c_1(x^* - g)$ y $c_2(x') \leq c_2(x^* - g) < c_2(x^*)$.

Al comparar $t > 0$ y $a_0 > 0$ tenemos dos opciones. Si $0 < t < a_0$, por el apartado 1 de esta proposición, tenemos que la solución óptima del problema $P_2(c_1(x^*) + a_0 - t)_{\preceq_{c_2, c_1}}$ es $(x^*, a_0 - t)$. Además, como $(x', 0)$ es factible, se cumple que $(x^*, a_0 - t) \preceq_{c_2, c_1} (x', 0)$, lo que implica que $c_2(x^*) \leq c_2(x')$, pero esto nos conduce a una contradicción.

Si $0 < a_0 < t$ entonces podemos afirmar que $(x', t - a_0)$ es factible para $P_2(c_1(x^*))_{\preceq_{c_2, c_1}}$ pues $c_1(x') + t - a_0 = c_1(x^*)$. Sin embargo como $(x^*, 0)$ es la solución de este problema, tenemos que $(x^*, 0) \preceq_{c_2, c_1} (x', t - a_0)$, luego $c_2(x^*) \leq c_2(x')$, pero esto nuevamente nos conduce a una contradicción.

Por tanto, concluimos que $a_0 = t = 0$ y la solución de $P_2(c_1(x^*) +$

$a_0)_{\preceq_{c_2, c_1}}$ es un punto de la forma $(x', 0)$. □

Los teoremas 3.2 y 3.4 y las proposiciones 3.3 y 3.5 son el fundamento del algoritmo 4 que damos en pseudo-código. El algoritmo calcula un conjunto de soluciones eficientes \mathcal{X}' con las características que cumple el conjunto definido en (3.6).

Primero se calcula la solución eficiente \tilde{x}_1 usando la proposición 3.3. A partir de este punto, dada una solución eficiente \tilde{x}_i seleccionamos de $G_{\tilde{x}_i}$ el $[g, a]$ con a mínimo que, como hemos visto en la proposición 3.5, verifica que la solución de $P_2(c_1(\tilde{x}_i) + a)$ es \tilde{x}_{i+1} .

El algoritmo 4 calcula una solución eficiente por cada punto en la frontera de Pareto, como nos lo asegura el teorema 3.4 y la proposición 3.5, y termina cuando obtenemos un punto \tilde{x}_r tal que $G_{\tilde{x}_r} = \emptyset$. Esto ocurre necesariamente porque el conjunto factible es finito y los valores de c_1 están acotados.

Algoritmo 4 Algoritmo para obtener la frontera de Pareto exacta de un problema de optimización lineal entera biobjetivo

- 1: **Entrada:** el vector de costes $c = (c_1, c_2)$, la matriz de restricciones A y el vector b del problema (3.2).
 - 2: Calcular $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$ utilizando el test-set asociado con respecto al orden \preceq_{c_1, c_2} . Notamos \tilde{x}_1 el óptimo y $c_1(\tilde{x}_1) = c_1^*$.
 - 3: $\mathcal{X}' := \{\tilde{x}_1\}$.
 - 4: $\epsilon_1 := c_1^*$.
 - 5: $e := \tilde{x}_1$.
 - 6: Calcular $\mathcal{T}_{\preceq_{c_2, c_1}}$ el test-set para la familia de problemas $P_2(\epsilon_1)_{\preceq_{c_2, c_1}}$.
 - 7: **Mientras** $G_e \neq \emptyset$ **hacer:**
 - 8: Seleccione $[g, a] \in G_e$ con $a > 0$ y $g \leq e$ tal que a sea mínimo.
 - 9: $\epsilon_1 := \epsilon_1 + a$.
 - 10: Calcular el óptimo de $P_2(\epsilon_1)_{\preceq_{c_2, c_1}}$ con solución inicial (e, a) , que notaremos $(p, 0)$.
 - 11: $e := p$.
 - 12: $\mathcal{X}' := \mathcal{X}' \cup \{e\}$.
 - 13: **Fin Mientras**
 - 14: **Salida:** El conjunto \mathcal{X}' tal que $c(\mathcal{X}') = \mathcal{F}$ que cumple las características del conjunto definido en (3.6).
-

A continuación, ilustramos con un ejemplo el algoritmo 4.

Ejemplo 3.6. Para encontrar la frontera de Pareto del ejemplo BBV en el caso $n = 4$ primero resolvemos

$$\begin{aligned} \text{mín} \quad & -8x_1 - 4x_2 - 2x_3 - x_4 \\ \text{s.a.} \quad & x_i + t_i = 1 \end{aligned} \tag{3.7}$$

utilizando el test set con respecto al orden \preceq_{c_1, c_2} donde $c_1 = (-8, -4, -2, -1, 0, 0, 0, 0)$ y $c_2 = (1, 2, 4, 8, 0, 0, 0, 0)$. Así, obtenemos el punto $\tilde{x}_1 = (1, 1, 1, 1, 0, 0, 0, 0)$ que es la primera solución eficiente que encontramos (los últimos cuatro ceros se deben a las variables de holgura), con costes $(-15, 15)$.

A continuación consideramos el problema

$$\begin{aligned} \text{mín} \quad & x_1 + 2x_2 + 4x_3 + 8x_4 \\ \text{s.a.} \quad & x_i + t_i = 1 \\ & -8x_1 - 4x_2 - 2x_3 - x_4 + c = \epsilon_1 \end{aligned} \quad (3.8)$$

El test-set con respecto al orden \preceq_{c_2, c_1} donde $c_2 = (1, 2, 4, 8, 0, 0, 0, 0)$ y $c_1 = (-8, -4, -2, -1, 0, 0, 0, 0)$, asociado a esta familia de problemas corresponde a

$$\begin{aligned} \mathcal{T}_{\preceq_{c_2, c_1}} = \{ & (-1, 0, 0, 1, 1, 0, 0, -1, -7), (-1, 0, 1, 0, 1, 0, -1, 0, -6), \\ & (-1, 1, 0, 0, 1, -1, 0, 0, -4), (0, -1, 0, 1, 0, 1, 0, -1, -3), \\ & (0, -1, 1, 0, 0, 1, -1, 0, -2), (0, 0, -1, 1, 0, 0, 1, -1, -1), \\ & (0, 0, 0, 1, 0, 0, 0, -1, 1), (0, 0, 1, 0, 0, 0, -1, 0, 2), \\ & (0, 1, 0, 0, 0, -1, 0, 0, 4), (1, 0, 0, 0, -1, 0, 0, 0, 8) \} \end{aligned}$$

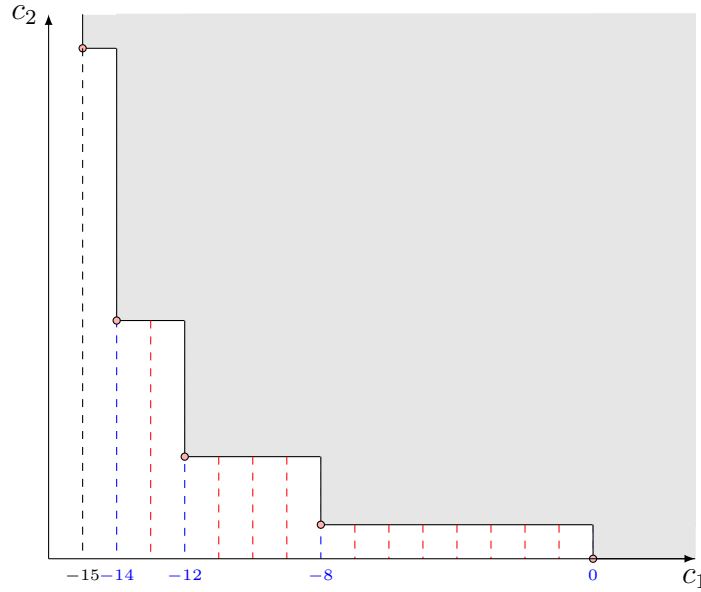


Figura 3.2: Ilustración de la aplicación del algoritmo 4 al problema BBV con $n = 4$.

La solución del problema (3.7) nos indica que el valor de ϵ_1 debe iniciar en -15 , pero para formular el primer problema $P_2(\epsilon_1)_{\preceq_{c_2, c_1}}$ primero verificamos

que $G_{e_1} \neq \emptyset$ y luego seleccionamos el mínimo de los a tales que $[g, a] \in G_{e_1}$, que corresponde a 1. Por tanto, debemos encontrar la solución del problema $P_2(-14)_{\leq c_2, c_1}$ reduciendo el punto $(1, 1, 1, 1, 0, 0, 0, 0, 1)$. De esta forma, siguiendo el algoritmo 4, solo necesitamos considerar incrementos de 1, 2, 4 y 8 (figura 3.2) para obtener el conjunto de soluciones eficientes que nos proporciona la frontera de Pareto.

- $(1, 1, 1, 1, 0, 0, 0, 0, 1)$ reduce a $(1, 1, 1, 0, 0, 0, 0, 0, 1, 0)$
- $(1, 1, 1, 0, 0, 0, 0, 0, 1, 2)$ reduce a $(1, 1, 0, 0, 0, 0, 0, 1, 1, 0)$
- $(1, 1, 0, 0, 0, 0, 0, 0, 0, 4)$ reduce a $(1, 0, 0, 0, 0, 0, 1, 1, 1, 0)$
- $(1, 1, 1, 0, 0, 0, 0, 0, 1, 8)$ reduce a $(0, 0, 0, 0, 1, 1, 1, 1, 1, 0)$

Terminamos el proceso con el punto $(0, 0, 0, 0, 1, 1, 1, 1, 0)$ pues aunque incrementemos la última variable de holgura, no vamos a obtener una reducción que nos genere una nueva solución eficiente.

Luego el problema BBV tiene 5 soluciones eficientes:

$$(1, 1, 1, 1), (1, 1, 1, 0), (1, 1, 0, 0), (1, 0, 0, 0), (0, 0, 0, 0)$$

cuyos valores (c_1, c_2) respectivos son

$$(-15, 15), (-14, 7), (-12, 3), (-8, 1), (0, 0)$$

que corresponden a toda la frontera de Pareto.

3.4. Cálculo teórico de los test-sets para el problema BBV

En algunos casos no tenemos la necesidad de calcular un test-set para un problema (o una familia de problemas) pues es posible darlo de forma teórica y entonces el cálculo de la frontera de Pareto siguiendo el algoritmo 4 es lineal. En el caso particular del problema BBV de tamaño n , el teorema que se menciona en esta sección, expone de forma explícita el test-set asociado a la familia de problemas $P_2(\epsilon_1)_{\leq c_2, c_1}$.

Formulamos el problema BBV como un problema de minimización

$$\begin{aligned} \text{mín} \quad & \sum_{i=1}^n -2^{n-i}x_i, \sum_{i=1}^n -2^{i-1}(1-x_i) \\ \text{s.a.} \quad & x_i \leq 1 \end{aligned} \tag{3.9}$$

Por simplicidad llamaremos c_1 a la primera función y c_2 a la segunda. Notaremos t_i a las variables de holgura para las restricciones anteriores, por

tanto, la matriz de las restricciones con las variables de holgura corresponde a

$$\begin{pmatrix} I_n & I_n \end{pmatrix}$$

La siguiente proposición caracteriza el test set asociado al problema $\min_{x \in \mathcal{X}} c_1 x$ con respecto al orden \preceq_{c_1, c_2} .

Proposición 3.7. Consideremos el problema

$$\begin{aligned} \min \quad & \sum_{i=1}^n -2^{n-i} x_i \\ \text{s.a.} \quad & x_i + t_i = 1 \end{aligned} \tag{3.10}$$

con respecto al orden \preceq_{c_1, c_2} , el orden inducido por los vectores de costes $c_1 = (-2^{n-1}, -2^{n-2}, \dots, -1, 0, \dots, 0)$, $c_2 = (1, 2, 2^2, \dots, 2^{n-1}, 0, \dots, 0)$ y después un orden lexicográfico, donde t_i son las variables de holgura para las n restricciones. El test-set asociado viene dado por

$$\mathcal{T}_{\preceq_{c_1, c_2}} = \{t_i - x_i\}$$

con $i = 1, \dots, n$ y

$$x_i = (\underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, \underbrace{0, \dots, 0, 0}_n) \text{ con } 1 \text{ en la posición } i.$$

$$t_i = (\underbrace{0, \dots, 0, 0}_n, \underbrace{0, \dots, 0, 1, 0, \dots, 0}_n) \text{ con } 1 \text{ en la posición } n + i.$$

Demostración. Notemos que cada uno de los elementos de $\mathcal{T}_{\preceq_{c_1, c_2}}$, pertenece al núcleo de la matriz de restricciones $A = \begin{pmatrix} I_n & I_n \end{pmatrix}$ y que $x_i \preceq_{c_1, c_2} t_i$.

Para demostrar que $\mathcal{T}_{\preceq_{c_1, c_2}}$ es un test-set asociado a la base de Gröbner reducida del ideal tórico definido por la matriz A , es suficiente probar que dado $z = z^+ - z^- \in \mathbb{Z}^{2n}$ tal que $z \in \text{Ker}(A)$ donde z^+ representa el término líder, esto es, $z^- \preceq_{c_1, c_2} z^+$, existe un elemento del test-set $t_i - x_i$ tal que $t_i \leq z^+$.

Para un vector $u \in \mathbb{Z}^{2n}$ notamos sus primeras n componentes como u_{x_i} con $1 \leq i \leq n$ y las últimas n componentes como u_{t_i} con $1 \leq i \leq n$. Si $u \in \text{Ker}(A)$ entonces $u_{x_i} = -u_{t_i}$.

Sea $z = z^+ - z^-$ con $z \in \text{Ker}(A)$. Si las últimas n componentes de z son todas iguales a 0, esto es, $z_{t_i} = 0$ para todo $1 \leq i \leq n$, entonces $z \in \text{Ker}(A_x)$ donde $A_x = \begin{pmatrix} I_n & 0 \end{pmatrix}$, por tanto $z = 0$.

Luego, supongamos que alguna de las últimas n componentes de z es positiva. Sea p el primer índice tal que $z_{t_p} > 0$, entonces $z_{t_p}^+ > 0$ y $t_p \leq z^+$.

Ahora supongamos que todas las últimas n componentes no nulas de z son negativas y sea p el primer índice tal que $z_{t_p} < 0$. Entonces z^+ tiene las últimas n componentes iguales a 0, esto es, $z_{t_i}^+ = 0$ para todo $1 \leq i \leq n$ y

z^- tiene las n primeras componentes iguales a 0, es decir, $z_{x_i}^- = 0$ para todo $1 \leq i \leq n$; por tanto, $c_1(z^+) < 0$ y $c_1(z^-) = 0$ pero esto contradice que z^+ es el término líder. \square

Teorema 3.8. Consideremos la familia de problemas

$$\begin{aligned} \text{mín} \quad & \sum_{i=1}^n -2^{i-1}(1 - x_i) \\ \text{s.a.} \quad & x_i + t_i = 1 \\ & \sum_{i=1}^n -2^{n-i}x_i + c = \epsilon_1 \end{aligned} \quad (3.11)$$

con respecto al orden \preceq_{c_2, c_1} , el orden inducido por los vectores de costes $c_2 = (1, 2, 2^2, \dots, 2^{n-1}, 0, \dots, 0)$, $c_1 = (-2^{n-1}, -2^{n-2}, \dots, -1, 0, \dots, 0)$ y después un orden lexicográfico, donde t_i son las variables de holgura para las n restricciones y c es la variable de holgura para la última restricción. El test-set asociado viene dado por

$$\mathcal{T}_{\preceq_{c_2, c_1}} = \{\underline{x_i t_j} - \underline{x_j t_i c^{2^{n-j} - 2^{n-i}}}, \underline{x_i c^{2^{n-i}}} - t_i\}$$

con $1 \leq i, j \leq n$ y $j < i$, donde

$$x_i = (\underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, \underbrace{0, \dots, 0, 0, 0}_n) \text{ con 1 en la posición } i.$$

$$t_i = (\underbrace{0, \dots, 0, 0, 0}_n, \underbrace{0, \dots, 0, 1, 0, \dots, 0}_n) \text{ con 1 en la posición } n + i.$$

$$x_i t_j = (\underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, \underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, 0, 0) \text{ con 1 en la posición } i$$

y 1 en la posición $n + j$.

$$x_j t_i c^{2^{n-j} - 2^{n-i}} = (\underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, \underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, 2^{n-j} - 2^{n-i}) \text{ con}$$

1 en la posición j y 1 en la posición $n + i$.

$$x_i c^{2^{n-i}} = (\underbrace{0, \dots, 0, 1, 0, \dots, 0}_n, \underbrace{0, \dots, 0, 0}_n, 2^{n-i}) \text{ con 1 en la posición } i.$$

Demostración. Cada uno de los elementos del test-set $\mathcal{T}_{\preceq_{c_2, c_1}}$, cuyo líder está subrayado, pertenece al núcleo de la matriz de restricciones:

$$A = \begin{pmatrix} I_n & I_n & 0 \\ c_1 & 0 & 1 \end{pmatrix}$$

Para un vector $u \in \mathbb{Z}^{2n+1}$ notamos sus primeras n componentes como u_{x_i} con $1 \leq i \leq n$, las n siguientes como u_{t_i} con $1 \leq i \leq n$ y la última componente como u_c . Si $u \in \text{Ker}(A)$ entonces $u_{x_i} = -u_{t_i}$ y $u_c = 2^{n-1}u_{x_1} + 2^{n-2}u_{x_2} + \dots + u_{x_n}$.

Sea $z = z^+ - z^- \in \mathbb{Z}^{2n+1}$ tal que $z \in \text{Ker}(A)$ y z^+ es el término líder según el orden \preceq_{c_2, c_1} , esto es, $z^- \preceq_{c_2, c_1} z^+$. A continuación veremos que existe un elemento del test-set que lo reduce, esto es, que el término líder del elemento del test-set es menor o igual que z^+ .

Si suponemos que $z_{x_i}^+ = 0$, para todo $1 \leq i \leq n$, esto es, el vector z^+ tiene las n primeras componentes iguales a 0, entonces $c_2(z^+) = 0$ pero como se cumple que $c_2(z^+) \geq c_2(z^-)$, entonces $z_{x_i}^- = 0$ para todo $1 \leq i \leq n$. Esto implica que $z_{x_i} = 0$ y como $z_{x_i} = -z_{t_i}$ y z_c es una combinación lineal de los z_{x_i} , concluimos que $z = 0$.

Luego, supongamos que alguna de las n primeras componentes de z^+ es positiva. Sea i el primer índice tal que $z_{x_i}^+ > 0$, entonces $z_{t_i}^- = z_{x_i}^+ > 0$, pues como $z = z^+ - z^- \in \text{Ker}(A)$ y z^+ , z^- tienen soporte disyunto, se cumple que $z_{x_i}^+ = z_{t_i}^-$ y $-2^{n-1}z_{x_1}^+ - 2^{n-2}z_{x_2}^+ - \dots - z_{x_n}^+ + z_c^+ = -2^{n-1}z_{x_1}^- - 2^{n-2}z_{x_2}^- - \dots - z_{x_n}^- + z_c^-$. Como esta última igualdad debe cumplirse, tenemos dos opciones: existe otro índice j tal que $z_{x_j}^- > 0$ o $z_c > 0$.

1. Supongamos que existe otro índice j tal que $z_{x_j}^- > 0$. Si $j < i$, entonces $z_{t_j}^+ = z_{x_j}^- > 0$ y por tanto $x_i t_j \leq z^+$.

Si $i < j$, supongamos que las componentes j_1, \dots, j_p en las variables x , que tiene z^- cumplen que $i < j_k$ para todo $1 \leq k \leq p$, pues si existe $z_{x_{i'}}^+ > 0$ tal que $j < i'$ entonces como $z_{t_j}^- > 0$, tenemos que $x_{i'} t_j \leq z^+$. Por tanto, sean i_1, \dots, i_r tales que $z_{x_{i_n}}^+ > 0$ y j_1, \dots, j_p tales que $z_{x_{j_k}}^- > 0$ con $i_1 < i_2 < \dots < i_r < j_1 < j_2 < \dots < j_p$. Como $z = z^+ - z^- \in \text{Ker}(A)$, la última fila de A nos indica que debe cumplirse que:

$$\begin{aligned} -2^{n-i_1} z_{x_{i_1}}^+ - 2^{n-i_2} z_{x_{i_2}}^+ - \dots - 2^{n-i_r} z_{x_{i_r}}^+ + 2^{n-j_1} z_{x_{j_1}}^- \\ + 2^{n-j_2} z_{x_{j_2}}^- + \dots + 2^{n-j_p} z_{x_{j_p}}^- + (z_c^+ - z_c^-) = 0 \end{aligned}$$

$$\text{con } z_c^+ - z_c^- = z_c = 2^{n-i_1} z_{x_{i_1}}^+ + \dots + 2^{n-i_r} z_{x_{i_r}}^+ - 2^{n-j_1} z_{x_{j_1}}^- - \dots - 2^{n-j_p} z_{x_{j_p}}^-.$$

Si demostramos que $z_c \geq 2^{n-i_1}$ tendremos que $x_i c^{2^{n-i_1}} \leq z^+$, pero esto es equivalente a demostrar que

$$2^{j_p-i_1} z_{x_{i_1}}^+ + \dots + 2^{j_p-i_r} z_{x_{i_r}}^+ - 2^{j_p-j_1} z_{x_{j_1}}^- - \dots - z_{x_{j_p}}^- \geq 2^{j_p-i_1} \quad (3.12)$$

Como $c_2(z^+) \geq c_2(z^-)$ tenemos que

$$2^{i_1-1} z_{x_{i_1}}^+ + 2^{i_2-1} z_{x_{i_2}}^+ + \dots + 2^{i_r-1} z_{x_{i_r}}^+ \geq 2^{j_1-1} z_{x_{j_1}}^- + 2^{j_2-1} z_{x_{j_2}}^- + \dots + 2^{j_p-1} z_{x_{j_p}}^-$$

o equivalentemente

$$z_{x_{i_1}}^+ + 2^{i_2-i_1} z_{x_{i_2}}^+ + \dots + 2^{i_r-i_1} z_{x_{i_r}}^+ - 2^{j_1-i_1} z_{x_{j_1}}^- - 2^{j_2-i_1} z_{x_{j_2}}^- - \dots - 2^{j_p-i_1} z_{x_{j_p}}^- \geq 0$$

Ahora, si a la expresión del lado izquierdo de la desigualdad (3.12) le restamos la cantidad positiva anterior, obtenemos que

$$\begin{aligned} & 2^{j_p-i_1} z_{x_{i_1}}^+ + \dots + 2^{j_p-i_r} z_{x_{i_r}}^+ - 2^{j_p-j_1} z_{x_{j_1}}^- - 2^{j_p-j_2} z_{x_{j_2}}^- - \dots - z_{x_{j_p}}^- \geq \\ & (2^{j_p-i_1} - 1) z_{x_{i_1}}^+ + (2^{j_p-i_2} - 2^{i_2-i_1}) z_{x_{i_2}}^+ + \dots + (2^{j_p-i_r} - 2^{i_r-i_1}) z_{x_{i_r}}^+ \\ & + (2^{j_1-i_1} - 2^{j_p-j_1}) z_{x_{j_1}}^- + (2^{j_2-i_1} - 2^{j_p-j_2}) z_{x_{j_2}}^- + \dots + (2^{j_p-i_1} - 1) z_{x_{j_p}}^- \end{aligned}$$

Por otro lado, tenemos dos sucesiones:

$$0 < i_2 - i_1 < \dots < i_r - i_1 < j_1 - i_1 < \dots < j_{p-1} - i_1 < j_p - i_1$$

$$j_p - i_1 > \dots > j_p - i_r > j_p - j_1 > j_p - j_2 > \dots > j_p - j_{p-1} > 0$$

Luego, existe un índice k_0 tal que:

- Para todo $i_n < k_0$ se cumple que $i_n - i_1 < j_p - i_n$ y para todo $i_n \geq k_0$ se tiene que $i_n - i_1 \geq j_p - i_n$ con $1 \leq n \leq r$.
- Para todo $j_k < k_0$ se cumple que $j_k - i_1 < j_p - j_k$ y para todo $j_k \geq k_0$ se tiene que $j_k - i_1 \geq j_p - j_k$ con $1 \leq k \leq p$.

Vamos a distinguir tres casos dependiendo de dónde se encuentre este índice k_0 .

- Si $i_1 < \dots < i_r \leq k_0 < j_1 < \dots < j_p$ entonces

$$\begin{aligned} & \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} z_{x_{i_1}}^+ + \underbrace{(2^{j_p-i_2} - 2^{i_2-i_1})}_{\geq 0} z_{x_{i_2}}^+ + \dots + \underbrace{(2^{j_p-i_r} - 2^{i_r-i_1})}_{\geq 0} z_{x_{i_r}}^+ \\ & + \underbrace{(2^{j_1-i_1} - 2^{j_p-j_1})}_{\geq 0} z_{x_{j_1}}^- + \underbrace{(2^{j_2-i_1} - 2^{j_p-j_2})}_{\geq 0} z_{x_{j_2}}^- + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} z_{x_{j_p}}^- \end{aligned}$$

Así esta suma de términos positivos, que llamamos S , es mayor o igual que la suma de su primer y último término, además como $z_{x_{i_1}}^+ + z_{x_{j_p}}^- \geq 2$, $j_p - i_1 \geq 1$ y para todo $m \geq 2$ se cumple que $2(m-1) \geq m$, tenemos que

$$\begin{aligned} S & \geq (2^{j_p-i_1} - 1) z_{x_{i_1}}^+ + (2^{j_p-i_1} - 1) z_{x_{j_p}}^- \\ & = (2^{j_p-i_1} - 1) (z_{x_{i_1}}^+ + z_{x_{j_p}}^-) \\ & \geq 2(2^{j_p-i_1} - 1) \geq 2^{j_p-i_1} \end{aligned}$$

Por tanto se cumple (3.12).

- Si $i_1 < \dots < i_{k'} \leq k_0 < i_{k'+1} < i_r < j_1 < \dots < j_p$, entonces

$$\begin{aligned}
 S &= \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} z_{x_{i_1}}^+ + \dots + \underbrace{(2^{j_p-i_{k'}} - 2^{i_{k'}-i_1})}_{\geq 0} z_{x_{i_{k'}}}^+ \\
 &+ \underbrace{(2^{j_p-i_{k'+1}} - 2^{i_{k'+1}-i_1})}_{\leq 0} z_{x_{i_{k'+1}}}^+ + \dots + \underbrace{(2^{j_p-i_r} - 2^{i_r-i_1})}_{\leq 0} z_{x_{i_r}}^+ \\
 &+ \underbrace{(2^{j_1-i_1} - 2^{j_p-j_1})}_{\geq 0} z_{x_{j_1}}^- + \underbrace{(2^{j_2-i_1} - 2^{j_p-j_2})}_{\geq 0} z_{x_{j_2}}^- + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} z_{x_{j_p}}^-
 \end{aligned}$$

Sea t^* el menor elemento del conjunto $\{z_{x_{i_1}}^+, \dots, z_{x_{i_r}}^+, z_{x_{j_1}}^-, \dots, z_{x_{j_p}}^-\}$, entonces tenemos que

$$\begin{aligned}
 S &\geq \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_{k'}} - 2^{i_{k'}-i_1})}_{\geq 0} t^* \\
 &+ \underbrace{\sum_{k=k'+1}^r (2^{j_p-i_k} - 2^{i_k-i_1})}_{\leq 0} t^* + \underbrace{(2^{j_1-i_1} - 2^{j_p-j_1})}_{\geq 0} t^* \\
 &+ \underbrace{(2^{j_2-i_1} - 2^{j_p-j_2})}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^*
 \end{aligned}$$

Ahora, usando que $\sum_{i=1}^t 2^i < 2^{t+1}$, que para todo $2 \leq k \leq r$, $i_k - i_1 < j_1 - i_1$ y que $j_p - i_r > j_p - j_1$, obtenemos:

$$\begin{aligned}
 S &\geq \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_{k'}} - 2^{i_{k'}-i_1})}_{\geq 0} t^* \\
 &+ \underbrace{\sum_{k=k'+1}^{r-1} 2^{j_p-i_k}}_{\geq 0} t^* + \underbrace{(2^{j_p-i_r} - 2^{j_p-j_1})}_{\geq 0} t^* + (2^{j_1-i_1} - 2^{j_1-i_1}) t^* \\
 &+ \underbrace{(2^{j_2-i_1} - 2^{j_p-j_2})}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^* \\
 &\geq 2t^*(2^{j_p-i_1} - 1) \geq 2(2^{j_p-i_1} - 1) \geq 2^{j_p-i_1}
 \end{aligned}$$

- Si $i_1 < \dots < i_r < j_1 < \dots < j_{k'} \leq k_0 < j_{k'+1} < \dots < j_p$, entonces

$$\begin{aligned}
 S &= \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} z_{x_{i_1}}^+ + \dots + \underbrace{(2^{j_p-i_r} - 2^{i_r-i_1})}_{\geq 0} z_{x_{i_r}}^+ \\
 &+ \underbrace{(2^{j_1-i_1} - 2^{j_p-j_1})}_{\leq 0} z_{x_{j_1}}^- + \dots + \underbrace{(2^{j_{k'}-i_1} - 2^{j_p-j_{k'}})}_{\leq 0} z_{x_{j_{k'}}}^- \\
 &+ \underbrace{(2^{j_{k'+1}-i_1} - 2^{j_p-j_{k'+1}})}_{\geq 0} z_{x_{j_{k'+1}}}^- + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} z_{x_{j_p}}^-
 \end{aligned}$$

Sea t^* el menor elemento del conjunto $\{z_{x_{i_1}}^+, \dots, z_{x_{i_r}}^+, z_{x_{j_1}}^-, \dots, z_{x_{j_p}}^-\}$, entonces tenemos que

$$\begin{aligned}
 S &\geq \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_{r-1}} - 2^{i_{r-1}-i_1})}_{\geq 0} t^* \\
 &+ \underbrace{(2^{j_p-i_r} - 2^{i_r-i_1})}_{\geq 0} t^* + \underbrace{\sum_{k=1}^{k'} (2^{j_k-i_1} - 2^{j_p-j_k})}_{\leq 0} t^* \\
 &+ \underbrace{(2^{j_{k'+1}-i_1} - 2^{j_p-j_{k'+1}})}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^*
 \end{aligned}$$

Si tenemos en cuenta que $\sum_{i=1}^t 2^i < 2^{t+1}$, que para todo $1 \leq k \leq p-1$, $j_p - j_k < j_p - i_r$ y que $j_1 - i_1 > i_r - i_1$, obtenemos:

$$\begin{aligned}
 S &\geq \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_{r-1}} - 2^{i_{r-1}-i_1})}_{\geq 0} t^* \\
 &+ \underbrace{(2^{j_1-i_1} - 2^{i_r-i_1})}_{\geq 0} t^* + \underbrace{\sum_{k=2}^{k'} 2^{j_k-i_1}}_{\geq 0} t^* + (2^{j_p-i_r} - 2^{j_p-i_r}) t^* \\
 &+ \underbrace{(2^{j_{k'+1}-i_1} - 2^{j_p-j_{k'+1}})}_{\geq 0} t^* + \dots + \underbrace{(2^{j_p-i_1} - 1)}_{\geq 0} t^* \\
 &\geq 2t^*(2^{j_p-i_1} - 1) \geq 2(2^{j_p-i_1} - 1) \geq 2^{j_p-i_1}
 \end{aligned}$$

- Ahora veamos el caso donde no existe otro índice j tal que $z_{x_j}^- > 0$. Como tenemos que i es el primer índice tal que $z_{x_i}^+ > 0$, entonces dados otros índices $i = i_1 < i_2 < \dots < i_r$ tales que $z_{x_{i_k}}^+ > 0$ con $1 \leq k \leq r$, usando la última fila de la matriz A tendríamos que $z_c = 2^{n-i_1} z_{x_{i_1}}^+ + \dots + 2^{n-i_r} z_{x_{i_r}}^+ \geq 2^{n-i} z_{x_i}^+$, por tanto, $x_i c^{2^{n-i}} \leq z^+$. \square

3.5. Resultados computacionales

Podemos encontrar en la literatura diferentes métodos exactos para resolver el caso biobjetivo lineal entero. En esta sección nos centraremos en el muy citado trabajo de Mavrotas y Florios [46] en el que se presenta el algoritmo AUGMECON2, que es una mejora de AUGMECON [45]. Este es un método general de resolución de programas multiobjetivos lineales enteros, basado en el método de la ϵ -restricciones mediante una elección adecuada de parámetros. El código en GAMS de AUGMECON2 está disponible, lo que nos ha permitido la comparación con nuestro método en el mismo equipo. Otro trabajo que hemos utilizado es el de Rong y Figueira [52], en él se presenta un método de programación dinámica, tratando especialmente el problema de la mochila con variables no acotadas.

El algoritmo 4 está implementado en C++ y utiliza 4ti2 para calcular los test-set solicitados. Los experimentos se llevaron a cabo en un 3.7 GHz Quad-Core Intel Xeon con 12 GB de RAM bajo un sistema operativo OSX versión 10.9.5. El algoritmo AUGMECON2 escrito en GAMS, se ha ejecutado en GAMS 28.4.5 que usa Cplex 12.7.1.0.

El tipo de problema que hemos elegido para realizar los experimentos es el problema de la mochila multiobjetivo centrándonos en el caso no acotado (multiobjective unbounded knapsack problem). Tal y como se dice en [52] el caso con variables binarias es más simple y fácil de tratar a priori. En este caso, b es la capacidad límite de peso de la mochila y hay n productos disponibles, donde cada producto j ($j = 1, \dots, n$) tiene un peso a_j y costes, c_{kj} , uno por cada objetivo k ($k = 1, 2, \dots, p$). Cada producto j tiene un número de copias disponible, acotado por b . Además, asumimos que c_{kj} , a_j y b son enteros positivos. El problema consiste en determinar el número de copias para cada producto tal que su peso total no exceda la capacidad de la mochila y sus costes totales sean maximizados. Cuando $p = 2$ el problema tiene la siguiente formulación:

$$\begin{aligned} \text{máx} \quad & \sum_{j=1}^n c_{kj} x_j \quad k = 1, 2 \\ \text{s.a.} \quad & \sum_{j=1}^n a_j x_j \leq b, \quad x_j \geq 0 \end{aligned} \tag{3.13}$$

Consideramos cuatro grupos de problemas, presentados en [6], generados aleatoriamente teniendo en cuenta la correlación existente entre los diferentes pesos y costes de los productos del problema.

- (A) *Instancias completamente no correlacionadas*: los objetivos y los coeficientes de peso no están relacionados entre sí. Los costes c_{1j} , c_{2j} y a_j se seleccionan aleatoriamente del intervalo $[1, 1000]$ con $j = 1, \dots, n$.

- (B) *Instancias no correlacionadas sin conflicto*: dos objetivos están correlacionados positivamente pero los coeficientes de peso no están relacionados con estos. Los costes $c_{1j} \in [1 + 100, 1000]$, $c_{2j} \in [c_{1j} - 100, c_{1j} + 100]$ y $a_j \in [1, 1000]$ con $j = 1, \dots, n$.
- (C) *Instancias no correlacionadas en conflicto*: dos objetivos están correlacionados negativamente pero los coeficientes de peso no están relacionados con estos. Los costes $c_{1j} \in [1, 1000]$, $c_{2j} \in [\text{máx}(1, 900 - c_{1j}), \text{mín}(1000, 1100 - c_{1j})]$ y $a_j \in [1, 1000]$ con $j = 1, \dots, n$.
- (D) *Instancias correlacionadas*: dos objetivos están correlacionados negativamente y los coeficientes de peso están correlacionados positivamente con estos. Los costes $c_{1j} \in [1, 1000]$, $c_{2j} \in [\text{máx}(1, 900 - c_{1j}), \text{mín}(1000, 1100 - c_{1j})]$ y $a_j \in [\text{máx}(1, c_{1j} + c_{2j} - 200), c_{1j} + c_{2j} + 200]$ con $j = 1, \dots, n$.

En todos los caso $b = \lfloor \frac{1}{2} \sum_{j=1}^n a_j \rfloor$.

3.5.1. Test-sets vs. AUGMECON2

En la tabla 3.2 se resumen los resultados obtenidos para 10 ejemplos de problemas tipo A, B, C y D generados aleatoriamente.

En las tablas de esta sección aparecen:

- El tamaño de la frontera de Pareto ($|\mathcal{F}|$).
- El tamaño del último test-set calculado ($|\mathcal{T}|$).
- El tiempo CPU que tarda el algoritmo 4 en obtener la frontera de Pareto, discriminado como el tiempo que emplea 4ti2 en calcular todos los test-sets requeridos (4ti2) y el tiempo restante (t_r).
- Finalmente el tiempo que tarda AUGMECON2.

El tiempo de cálculo del algoritmo 4 está relacionado con el tamaño de la frontera de Pareto y el tamaño del test-set, pues al incrementar el tamaño de la base y la complejidad del problema, se requiere mayor tiempo computacional. Además al llevar a cabo las reducciones debemos realizar una búsqueda en el test-set para seleccionar elementos. Al llegar a los problemas tipo D la combinatoria de los problemas producen test-sets de mucho mayor tamaño que convierten a nuestro método en menos competitivo.

El método AUGMECON2 resuelve, prácticamente, el número exacto de subproblemas necesarios. En su código se indica que con el objetivo de calcular la frontera de Pareto se requiere de un mallado completo, en casos biobjetivo es el rango de variación del segundo coste. En los ejemplos tratados,

dicho rango crecía enormemente, necesitando en muchos casos de más de 24 horas de ejecución. Hemos seleccionado valores menores que nos proporcionan la frontera de Pareto con unos tiempos más razonables para AUGMECON2, especialmente en las tablas 3.3 y 3.4, donde los rangos eran mayores.

Podemos observar que la reducción por el test-set es mucho más rápida que la resolución de un subproblema por CPLEX en los problemas tipo A, B y C que tarda del orden de 10 veces más que nosotros.

Problema de la mochila no acotada biobjetivo con $n = 50$						
Tipo		$ \mathcal{T} $	$ F $	Algoritmo 4		AUGMECON2
				4ti2	t_r	
A	media	104,45	233,335	0,0865 sg	1,0345 sg	10,65 sg
	máx.	145	2390	0,09 sg	6,84 sg	101,2 sg
B	media	103	46,9	0,072 sg	0,214 sg	4,61 sg
	máx.	104	306	0,1 sg	0,89 sg	34,91 sg
C	media	127,2	559,7	0,102 sg	4,972 sg	131,3 sg
	máx.	209	3684	0,12 sg	40,46 sg	1493 sg
D	media	1620	200,7	3,42 sg	1219,76 sg	80,1 sg
	máx.	7981	1061	7,05 sg	13094,5 sg	140,09 sg

Tabla 3.2

Problemas de la tabla 3.2 con lado derecho igual a $10b$						
Tipo		$ \mathcal{T} $	$ F $	Algoritmo 4		AUGMECON2
				4ti2	t_r	
A	media	104,45	2319,34	0,0865 sg	11,772 sg	102,77 sg
	máx.	145	23756	0,09 sg	77,839 sg	976,58 sg
B	media	103	466,18	0,072 sg	2,4353 sg	44,48 sg
	máx.	104	3041	0,1 sg	10,128 sg	336,88 sg
C	media	127,2	5563,41	0,102 sg	56,581 sg	1267,045 sg
	máx.	209	36618	0,12 sg	460,43 sg	14407,45 sg

Tabla 3.3

Problemas de la tabla 3.2 con lado derecho igual a 100b						
Tipo		$ \mathcal{T} $	$ F $	Algoritmo 4		AUGMECON2
				4ti2	t_r	
A	media	104,45	23184,16	0,0865 sg	147,77 sg	1064,14 sg
	máx.	145	237470	0,09 sg	977,1 sg	10111,9 sg
B	media	103	4659,98	0,072 sg	30,56 sg	460,63 sg
	máx.	104	30404	0,1 sg	127,13 sg	3488,2 sg
C	media	127,2	55611,79	0,102 sg	710,25 sg	13119,49 sg
	máx.	209	366042	0,12 sg	5779,71 sg	149180,56 sg

Tabla 3.4

Aumentando el número de restricciones en los problemas considerados el problema tiene asociado test-sets mayores y pueden complicar su resolución, como se puede observar en la tabla 3.5.

Problema de la mochila multidimensional no acotada (tipo A)						
n	No. filas		$ \mathcal{T} $	$ F $	Algoritmo 4	
					4ti2	t_r
10	2	media	99	9,2	0,055 sg	0,053 sg
		máx.	160	18	0,06 sg	0,12 sg
	3	media	753	5,4	0,098 sg	0,3152 sg
		máx.	1930	11	0,2 sg	1,16 sg
20	2	media	453,8	20,8	0,08 sg	0,9314 sg
		máx.	793	48	0,1 sg	2,89 sg
	3	media	2102,75	13	0,425 sg	4,5775 sg
		máx.	4211	20	0,87 sg	8,11 sg
40	2	media	999	93,5	0,205 sg	20,11 sg
		máx.	1408	130	0,27 sg	21,85 sg
	3	media	9777	70	15,13 sg	1161,06 sg
		máx.	18638	120	30,11 sg	2320,63 sg

Tabla 3.5

Este tipo de problema se conocen como el *problema de la mochila no acotada multidimensional multiobjetivo*, que es una generalización del problema de la mochila no acotada multiobjetivo, al considerar varias mochilas cada una con un peso límite. Específicamente,

$$\begin{aligned} \text{máx} \quad & \sum_{j=1}^n c_{kj}x_j \quad k = 1, 2 \\ \text{s.a.} \quad & \sum_{j=1}^n a_{ij}x_j \leq b_i \quad i = 1, \dots, m \quad x_j \geq 0 \end{aligned} \tag{3.14}$$

donde n es el número de productos; c_{kj} representa el coste del producto j sobre el criterio k ($k = 1, 2$); m es el número de mochilas con a_{ij} el peso del j -ésimo producto respecto de la mochila i ($i = 1, \dots, m$); b_i es la capacidad límite de peso de la mochila i y la variable de decisión x_j indica el número de copias del producto j colocadas en cada mochila. Además asumimos que c_{kj} , a_{ij} y b_i son enteros positivos.

3.5.2. Problema de la mochila caso binario

El caso binario ha sido mucho más tratado en la literatura, posiblemente por ser potencialmente más sencillo como se destacó anteriormente. En nuestro caso, para tratar estos problemas, añadimos tantas variables de holgura como variables de decisión tenga el problema original y el mismo número de restricciones de tipo $x_i + t_i = 1$. El cálculo general de bases de Gröbner es especialmente sensible al número de variables [20]. Y ya hemos citado la sensibilidad al número de restricciones. Hemos mejorado los tiempos de cómputo al usar las bases de Gröbner truncadas [59], sin embargo, los tiempos pasan a ser poco competitivos con pocas variables (tabla 3.6).

3.5.3. Límites de mochila no acotada tipo A

En la tabla 3.7 mostramos los cálculos obtenidos para 100, 200, \dots , 500 variables, que proponemos como benchmarks límites de nuestro método.

3.5.4. Mochila no acotada vs. Figueira

Terminamos el caso biobjetivo comparándonos con [52]. Salvo en el caso D, obtenemos tiempos muy competitivos (tabla 3.8). Es llamativo que los casos tratados siguen el mismo formato, pero los tamaños medios de la frontera de Pareto en [52] son mucho mayores.

Problemas de la mochila 0-1 biobjetivo (tipo A)						
n		$ \mathcal{T} $	$ F $	Algoritmo 4		AUGMECON2
				4ti2	t_r	
10	media	190,3	4,3	0,088 sg	0,036 sg	0,365 sg
	máx	302	8	0,11 sg	0,08 sg	0,61 sg
15	media	691,1	6,4	0,366 sg	0,539 sg	0,519 sg
	máx	1018	12	1,64 sg	1,81 sg	0,9 sg
20	media	3267,9	10,8	1,534 sg	10,645 sg	0,84 sg
	máx	4945	18	2,78 sg	24,93 sg	1,38 sg

Tabla 3.6

Límites problemas de la mochila no acotada (tipo A)						
n		$ \mathcal{T} $	$ F $	Algoritmo 4		
				4ti2	t_r	
100	media	205,2	1194,8	0,101 sg	14,29 sg	
	máx	210	4441	0,11 sg	67,68 sg	
200	media	403,9	3884,9	0,384 sg	121,56 sg	
	máx	416	11936	0,32 sg	431,3 sg	
300	media	603,1	16879,7	0,953 sg	1835,64 sg	
	máx	610	48563	2,3 sg	5862,96 sg	
400	media	805,8	43909,5	1,7 sg	1735,31 sg	
	máx	835	48167	6,42 sg	9230,49 sg	
500	media	1002,9	50100,2	3,29 sg	23826,61 sg	
	máx	1010	129923	10,76 sg	99173,3 sg	

Tabla 3.7

Problema de la mochila no acotada biobjetivo vs. Figueira							
Tipo	n		$ \mathcal{T} $	$ F $	Algoritmo 4		Figueira
					4ti2	t_r	
A	120	media	242,33	308,33	0,18 sg	17,46 sg	78,1 sg
		máx.	254	1475	0,25 sg	63,67 sg	1139,7 sg
	140	media	282,83	523,75	0,21 sg	20,84 sg	76,6 sg
		máx.	291	5339	0,23 sg	134,102 sg	1061,4 sg
	160	media	321,22	1092,78	0,26 sg	47,30 sg	99,8 sg
		máx.	324	7397	0,33 sg	639,46 sg	1099,2 sg
	180	media	361,2	366	0,306 sg	66,098 sg	122,0 sg
		máx.	362	1826	0,35 sg	111,462 sg	1324,6 sg
	200	media	403	3884,9	0,384 sg	121,56 sg	116,5 sg
		máx.	416	11936	0,32 sg	431,3 sg	1040,2 sg
C	120	media	249,6	1605,54	0,18 sg	36,18 sg	78,4 sg
		máx.	277	7633	0,23 sg	235,97 sg	349,3 sg
	140	media	284,4	1434,2	0,196 sg	31,87 sg	131,5 sg
		máx.	300	5890	0,27 sg	127,67 sg	862,2 sg
	160	media	321,8	2162	0,287 sg	194,99 sg	168,5 sg
		máx.	328	5772	0,44 sg	1467,91sg	895,4 sg
	180	media	363,3	4748,2	0,33 sg	625,67 sg	190,9 sg
		máx.	381	28853	0,48 sg	5249,61 sg	977,0 sg
	200	media	402,3	5674,2	0,358 sg	310,10 sg	215,9 sg
		máx.	414	17020	0,44 sg	851,2 sg	967,5 sg

Tabla 3.8

CAPÍTULO 4

Test-sets para resolver problemas de optimización lineal entera multiobjetivo

En este capítulo presentamos un método exacto para encontrar la frontera de Pareto de un problema lineal entero con p objetivos, que nuevamente hace uso de la estrecha relación entre el método clásico de las ϵ -restricciones y el concepto de test-set, calculado vía bases de Gröbner, para problemas lineales enteros con un objetivo.

En el caso biobjetivo, partiendo de la solución óptima de la primera función objetivo y adaptando apropiadamente la última variable de holgura, obtuvimos una secuencia de reducciones que nos proporcionaron la frontera de Pareto. Aunque la dificultad se incrementa cuando tenemos p objetivos, logramos generalizar la estrategia presentada en el capítulo anterior para realizar los incrementos del vector ϵ , pero esta vez no conseguimos evitar todos los problemas innecesarios.

La formulación de métodos exactos para resolver problemas lineales enteros multiobjetivo generales o específicos, es de gran interés pues estos permiten comprobar la precisión y el alcance de los métodos aproximados, tal y como se evidencia en trabajos como [49], [40], [55], [41], [46], [60], [35], [11], [51] o [52].

4.1. Motivación: un problema RAP para sistemas serie-paralelo

Los problemas de asignación con redundancia (*Redundancy Allocation Problem* - RAP) son un tipo de problemas de optimización de fiabilidad, en los que se permite la selección de componentes redundantes, con el objetivo de maximizar la fiabilidad del sistema y otras características, dadas ciertas restricciones sobre el sistema. Aunque la fiabilidad de un sistema se puede incrementar asignando componentes redundantes a sus subsistemas, esto también puede incrementar el coste del diseño y afectar de forma negativa otras propiedades como el peso del sistema o el volumen.

Este tipo de problemas tiene aplicaciones en muchas situaciones prácticas de diseño (cf. [48], [44]) que requieren una fiabilidad alta, como las telecomunicaciones, la fabricación, los sistemas de transporte, sistemas de seguridad y los sistemas de energía eléctrica entre otros; por eso cada vez es más importante desarrollar soluciones eficientes para el RAP.

El problema que presentamos a continuación es un problema RAP para sistemas serie-paralelo presentado en [13] como un problema de tres objetivos: maximizar la fiabilidad, minimizar el coste y minimizar el peso del sistema; formulado como:

$$\begin{aligned} & \text{máx} && f_R(\mathbf{x}), \\ & \text{mín} && f_C(\mathbf{x}), f_W(\mathbf{x}) \\ & \text{s.a.} && 1 \leq \sum_{j=1}^{m_i} x_{ij} \leq n_{\text{máx},i}, \forall i \in S \end{aligned} \quad (4.1)$$

donde $\mathbf{x} = \{x_{ij} : \forall i \in S, j = 1, \dots, m_i\}$ y $x_{ij} \in \{0, 1, 2, \dots, n_{\text{máx},i}\}$,

$$f_R(\mathbf{x}) = \left[\prod_{i=1}^s \left(1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right] \quad (4.2)$$

$$f_C(\mathbf{x}) = \left[\sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} \right] \quad (4.3)$$

$$f_W(\mathbf{x}) = \left[\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \right] \quad (4.4)$$

El sistema serie-paralelo tiene s subsistemas independientes organizados en serie y el conjunto de éstos es notado como S . Cada subsistema i puede tener hasta $n_{\text{máx},i}$ componentes funcionalmente equivalentes dispuestas en paralelo y opera correctamente si al menos una de sus componentes funciona;

además dispone de m_i tipos de componentes de las que se pueden seleccionar varias copias de cada tipo.

La variable de decisión x_{ij} indica la cantidad del j -ésimo tipo de componente usada en el subsistema i . Los parámetros r_{ij} , c_{ij} y w_{ij} indican la fiabilidad, coste y peso (estos dos últimos se asumen enteros) respectivamente, del j -ésimo tipo de componente disponible para el subsistema i .

Notemos que el problema RAP mencionado es un problema de programación no lineal, pues la función objetivo que indica la fiabilidad del sistema no es lineal. Sin embargo en [13] se propone una descomposición del problema RAP en subproblemas lineales multiobjetivo para su solución. Primero se linealiza la función objetivo de fiabilidad a través de una transformación logarítmica y luego se descompone el problema RAP multiobjetivo en subproblemas multiobjetivo más pequeños correspondientes a cada subsistema.

Se demuestra que la maximización del producto de la fiabilidad de todos los subsistemas, es equivalente a la maximización de la suma de los logaritmos de la fiabilidad de todos los subsistemas, sujeta a las restricciones dadas en (4.1), esto es, la función objetivo (4.2), se puede cambiar por:

$$\text{máx } f_{R_1}(\mathbf{x}) = \left[\sum_{i=1}^s \log \left(1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right] \quad (4.5)$$

De esta manera, la función de fiabilidad se puede descomponer por subsistemas y resolver cada problema de maximización correspondiente de manera independiente:

$$\text{máx } f_{R_2}(\mathbf{x}_i) = \left[\log \left(1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right], \quad \forall i \in S \quad (4.6)$$

donde $\mathbf{x}_i = \{x_{ij} : j = 1, \dots, m_i\}$.

Sin embargo, como la función logarítmica es creciente, la maximización de la función f_{R_2} , corresponde a la minimización de la función

$$\text{mín } f_{R_3}(\mathbf{x}_i) = \left[\log \left(\prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right] = \left[\sum_{j=1}^{m_i} \log(1 - r_{ij}) x_{ij} \right], \quad \forall i \in S \quad (4.7)$$

Las funciones (4.3), (4.4) y las restricciones del problema (4.1) se descomponen en subsistemas como

$$\text{mín } f_{C_1}(\mathbf{x}_i) = \left[\sum_{j=1}^{m_i} c_{ij} x_{ij} \right] \quad (4.8)$$

$$\text{mín } f_{W_1}(\mathbf{x}_i) = \left[\sum_{j=1}^{m_i} w_{ij} x_{ij} \right] \quad (4.9)$$

$$1 \leq \sum_{j=1}^{m_i} x_{ij} \leq n_{\text{máx},i} \quad (4.10)$$

En [13] se demuestra que toda la frontera de Pareto (conjunto de todos los puntos no dominados) del problema original RAP se obtiene de una combinación cartesiana de toda la frontera de Pareto de cada subsistema y un posterior filtrado de las soluciones dominadas. Este hecho nos muestra que es de gran interés e importancia contar con un método exacto para resolver los problemas lineales multiobjetivo correspondientes a cada subsistema. En este sentido, aunque la función objetivo de fiabilidad tiene valores objetivos continuos, podemos aproximar sus coeficientes y cambiarla por otra que tome valores enteros de manera que no se afecte la frontera de Pareto del subsistema original y así aplicar el algoritmo que proponemos en la siguiente sección.

4.2. Caracterización de soluciones eficientes

En esta sección presentamos los resultados que nos permitirán enunciar un algoritmo con el que daremos la frontera de Pareto de manera recursiva, esto es, primero resolveremos el problema con un objetivo, luego con 2, 3, \dots , $p-1$, p , de manera que el conjunto de puntos obtenido en una etapa i -ésima (minimizar los primeros i objetivos), nos sirva como conjunto de partida para la siguiente etapa, en el sentido de calcularlas para que sean soluciones eficientes para el problema de la etapa $i+1$ -ésima y por recurrencia del problema con p objetivos.

Para lograr esto, es importante modificar adecuadamente el orden inducido por el coste del problema que el método de las ϵ -restricciones propone resolver en cada etapa. Más precisamente, en una etapa i -ésima debemos modificar el orden inducido por el coste c_i para el problema $P_i(\epsilon)$:

$$\begin{aligned} \text{mín } & c_i x \\ \text{s.a. } & Ax = b \\ & c_1 x \leq \epsilon_1 \\ & \vdots \\ & c_{i-1} x \leq \epsilon_{i-1} \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.11)$$

con $\epsilon = (\epsilon_1, \dots, \epsilon_{i-1})$.

Dicha modificación consiste en definir un orden total tipo producto similar al definido en (2.7), que primero compara dos vectores con respecto a c_i ; en caso de igualdad compara según c_1 luego con $c_2, \dots, c_{i-1}, c_{i+1}, \dots, c_p$ y finalmente desempata con un orden monomial. Este orden lo notamos $\preceq_{\hat{c}_i}$. Esto se desarrollará formalmente en la sección 4.3.

En una etapa i -ésima con $i \geq 3$, la selección de los problemas que el método de las ϵ -restricciones propone resolver, al igual que el caso biobjetivo, también va a estar determinada por cierta información que nos aporta el test-set correspondiente. Sin embargo, esto es más complicado cuando debemos modificar varias variables de holgura al mismo tiempo: en lugar de obtener una cadena que parte de una solución eficiente y que nos conduce a toda la frontera de Pareto, a partir de cada punto del conjunto solución de la etapa anterior —que también será eficiente para el problema de la etapa i -ésima— tendremos que abrir un árbol, en el cual no podremos evitar subproblemas inútiles, es decir, aquellos que nos conduce a soluciones duplicadas.

Asumimos el problema de optimización lineal entera multiobjetivo en su forma estándar, esto es,

$$\begin{aligned} \text{mín} \quad & c_1x, c_2x, \dots, c_px \\ \text{s.a.} \quad & Ax = b \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.12)$$

Para resolver el problema (4.12) utilizando el método de las ϵ -restricciones debemos resolver problemas de la forma

$$\begin{aligned} \text{mín} \quad & c_px \\ \text{s.a.} \quad & Ax = b \\ & c_ix \leq \epsilon_i, \quad 1 \leq i \leq p-1, \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.13)$$

que en forma estándar corresponde a

$$\begin{aligned} \text{mín} \quad & c_px \\ \text{s.a.} \quad & Ax = b \\ & c_ix + h_i = \epsilon_i, \quad 1 \leq i \leq p-1, \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.14)$$

y notamos $P_p(\epsilon)$ con $\epsilon = (\epsilon_1, \dots, \epsilon_{p-1})$. Cuando seleccionemos la única solución con respecto al orden total $\preceq_{\hat{c}_p}$, haremos referencia al problema como $P_p(\epsilon)_{\preceq_{\hat{c}_p}}$. Llamamos $\mathcal{T}_{\preceq_{\hat{c}_p}} \subset \mathbb{Z}^{n+(p-1)}$ al test-set asociado a la familia de problemas $P_p(\epsilon)$ con respecto a $\preceq_{\hat{c}_p}$.

El siguiente teorema nos muestra que el orden $\preceq_{\hat{c}_p}$ nos permite descartar las soluciones débilmente eficientes al utilizar el método de las ϵ -restricciones.

Teorema 4.1. Un punto $(x^*, 0) \in \mathbb{Z}^{n+(p-1)}$ es solución del problema $P_p(\epsilon)_{\preceq_{\hat{c}_p}}$ con $\epsilon = (\epsilon_1, \dots, \epsilon_{p-1})$ donde $\epsilon_i = c_i(x^*)$ si y solo si $x^* \in \mathbb{Z}^n$ es solución eficiente del problema multiobjetivo (4.12) y de las soluciones eficientes con igual coste $c_1(x^*), c_2(x^*), \dots, c_p(x^*)$ es la menor según el orden $\preceq_{\hat{c}_p}$.

Demostración. \Leftarrow) Sea x^* solución eficiente del problema multiobjetivo (4.12) tal que de las soluciones eficientes con iguales costes $c_1(x^*), c_2(x^*), \dots, c_p(x^*)$, es la menor según el orden $\preceq_{\hat{c}_p}$. Por el teorema 1.15 tenemos que x^* es solución óptima de $P_p(\epsilon)$. Si dicho problema tiene otras soluciones, estas deben tener igual coste $c_p(x^*)$ y por tanto igual coste c_1, c_2, \dots, c_{p-1} que x^* , pues este punto es eficiente. Como x^* es la menor de ellas según $\preceq_{\hat{c}_p}$, concluimos que $(x^*, 0)$ es solución del problema $P_p(\epsilon)_{\preceq_{\hat{c}_p}}$.

\Rightarrow) Ahora, sea $(x^*, 0)$ solución de $P_p(\epsilon)_{\preceq_{\hat{c}_p}}$, luego $(x^*, 0)$ también es solución de $P_p(\epsilon)$. Para mostrar que x^* es una solución eficiente de (4.12), por el teorema 1.15, es suficiente probar que para todo $k = 1, \dots, p-1$, el punto $(x^*, 0)$ es solución de los problemas $P_k(\epsilon^*)$ con $\epsilon^* = (\epsilon_1, \dots, \epsilon_{k-1}, \epsilon_{k+1}, \dots, \epsilon_p)$ con $\epsilon_i = c_i(x^*)$

$$\begin{aligned} \text{mín} \quad & c_k x \\ \text{s.a.} \quad & Ax = b \\ & c_i x + h_i = c_i(x^*), \quad 1 \leq i \leq p, i \neq k, x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.15)$$

Tenemos que $(x^*, 0)$ es factible para (4.15) pero por reducción al absurdo, supongamos que no es una solución óptima de dicho problema. Entonces sea (x', t) con $t \geq 0$, una solución de (4.15), por tanto $c_k(x') < c_k(x^*)$ y $c_i(x') + t_i = c_i(x^*)$, es decir, $c_i(x') \leq c_i(x^*)$ con $1 \leq i \leq p$ y $i \neq k$.

Las anteriores desigualdades nos permiten concluir que existe $s \geq 0$ tal que (x', s) es factible para $P_p(\epsilon)_{\preceq_{\hat{c}_p}}$ con $c_p(x^*) = c_p(x')$, pues $(x^*, 0)$ es la solución de dicho problema. Pero por la definición del orden $\preceq_{\hat{c}_p}$ se cumple que $c_1(x^*) \leq c_1(x')$. Si $k \neq 1$, tenemos que $c_1(x') \leq c_1(x^*)$, por tanto, $c_1(x^*) = c_1(x')$. Con este mismo razonamiento logramos establecer igualdades para los costes c_i de x^* y x' con $i < k$, pero cuando el índice es k entonces se cumple que $c_k(x^*) \leq c_k(x')$ pero también que $c_k(x') < c_k(x^*)$ lo que nos conduce a una contradicción. Por tanto, $(x^*, 0)$ es solución del problema (4.15). Además, las soluciones eficientes con iguales costes que x^* también son factibles para $P_p(\epsilon)_{\preceq_{\hat{c}_p}}$, pero como $(x^*, 0)$ es la única solución de dicho problema, esto implica que es la menor de ellas según el orden $\preceq_{\hat{c}_p}$. \square

Definición 4.2. A la solución eficiente x^* del problema multiobjetivo (4.12), tal que $(x^*, 0)$ es solución del problema $P_i(c'(x^*))_{\preceq_{\hat{c}_i}}$ con $c'(x^*) = (c_1(x^*), \dots, c_{p-1}(x^*))$, la llamamos *solución $\preceq_{\hat{c}_p}$ -eficiente*.

Cada elemento $\mathbf{g} = (g_1, \dots, g_n, g_{n+1}, \dots, g_{n+(p-1)}) \in \mathcal{T}_{\preceq_{\hat{c}_p}}$ lo notamos $[g, a]$ con $g = (g_1, \dots, g_n)$ y $a = (g_{n+1}, \dots, g_{n+(p-1)})$.

Corolario 4.3. Si $(x^*, t) \in \mathbb{Z}^{n+(p-1)}$ con $t \geq 0$ es solución del problema $P_p(\epsilon')_{\preceq_{\hat{c}_p}}$ con $\epsilon' = (\epsilon'_1, \dots, \epsilon'_{p-1})$ entonces $x^* \in \mathbb{Z}^n$ es solución eficiente del problema multiobjetivo (4.12).

Demostración. Sea $(x^*, t) \in \mathbb{Z}^{n+(p-1)}$ con $t \geq 0$ solución de $P_p(\epsilon')_{\preceq \hat{c}_p}$. Por el teorema 4.1 es suficiente probar que $(x^*, 0)$ es solución del problema $P_p(\epsilon)_{\preceq \hat{c}_p}$ con $\epsilon = (\epsilon_1, \dots, \epsilon_{p-1})$ donde $\epsilon_i = c_i(x^*)$.

Si asumimos que $(x^*, 0)$ no es solución de dicho problema, debe existir $g \in \mathcal{T}_{\preceq \hat{c}_p} \subset \mathbb{Z}^{n+(p-1)}$ que lo reduce, esto es, $g \leq (x^*, 0)$; pero $g \leq (x^*, t)$ lo que contradice que (x^*, t) es la única solución óptima del problema $P_p(\epsilon')_{\preceq \hat{c}_p}$. \square

En el caso de p objetivos con $p > 2$ no podemos ordenar totalmente los puntos eficientes según sus costes c_1, \dots, c_{p-1} , a diferencia del caso biobjetivo donde podemos ordenar las soluciones eficientes por el coste c_1 , lo que nos asegura que ningún punto (x^*, t) es solución del problema $P_2(\epsilon_1)_{\preceq c_2, c_1}$. En el caso multiobjetivo con $p \geq 3$ se presentará esta situación como lo veremos en el ejemplo 4.8 que desarrollamos posteriormente.

Teorema 4.4. Sea x^* una solución $\preceq_{\hat{c}_p}$ -eficiente. Entonces se cumple solo uno de los siguientes enunciados:

1. $c'(x^*) = (c_1(x^*), c_2(x^*), \dots, c_{p-1}(x^*))$ pertenece a la frontera de Pareto del problema

$$\begin{aligned} \text{mín} \quad & c_1x, c_2x, \dots, c_{p-1}x \\ \text{s.a.} \quad & Ax = b \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \tag{4.16}$$

2. Existe x' , una solución $\preceq_{\hat{c}_p}$ -eficiente tal que $c_i(x') \leq c_i(x^*)$ con $1 \leq i \leq p-1$ con al menos una desigualdad estricta, y existe $[g, a] \in \mathcal{T}_{\preceq \hat{c}_p}$ tal que $g \leq x^*$, $a \geq 0$, $a \neq 0$ y $a = c'(x^*) - c'(x')$.

Demostración. Sea x^* una solución $\preceq_{\hat{c}_p}$ -eficiente y supongamos que $c'(x^*)$ no es un punto de la frontera de Pareto del problema (4.16). Por tanto, existe otra solución eficiente x para el problema (4.12) tal que $c'(x) \leq c'(x^*)$ con $c'(x) \neq c'(x^*)$. Entonces definimos el conjunto

$$D_{x^*} = \{x \in \mathcal{X}_E : c'(x) \leq c'(x^*) \text{ y } c'(x) \neq c'(x^*)\}$$

Sea $x' \in D_{x^*}$ tal que x' es el mínimo según el orden $\preceq_{\hat{c}_p}$. Luego, $(x', c'(x^*) - c'(x'))$ es factible para $P_p(c'(x^*))_{\preceq \hat{c}_p}$ y como $(x^*, 0)$ es su solución podemos reducir, entonces existe $[g, a] \in \mathcal{T}_{\preceq \hat{c}_p}$ tal que $g \leq x'$ y $a \leq c'(x^*) - c'(x')$. Al reducir, $(x' - g, c'(x^*) - c'(x') - a)$ también es factible para $P_p(c'(x^*))_{\preceq \hat{c}_p}$ con $(x' - g, c'(x^*) - c'(x') - a) \preceq_{\hat{c}_p} (x', c'(x^*) - c'(x'))$, lo que implica que $c_p(x' - g) \leq c_p(x')$.

A continuación veremos que $a \geq 0$, $a \neq 0$ y $a = c'(x^*) - c'(x')$:

- Iniciamos probando que $a \geq 0$ con $a \neq 0$. Como x' es una solución eficiente y es el mínimo según el orden $\preceq_{\hat{c}_p}$ del conjunto D_{x^*} , por el

teorema 4.1 tenemos que x' es una solución $\preceq_{\hat{c}_p}$ -eficiente. Luego no es posible que $a \leq 0$.

Ahora supongamos que $a \not\leq 0$, pero que tiene al menos una componente negativa. Sea k el menor índice tal que $a_k < 0$ con $k = 1, \dots, p-1$. Como $c_k(x') + a_k < c_k(x') \leq c_k(x^*)$, tenemos que para el problema $P_p(c'(x') + a)_{\preceq_{\hat{c}_p}}$, ni x' ni x^* son puntos factibles. Como $c'(x' - g) = c'(x') + a$, el punto $(x' - g, 0)$ sí lo es. Sea (x'', u) con $u \geq 0$ solución de $P_p(c'(x') + a)_{\preceq_{\hat{c}_p}}$, luego $(x'', u) \preceq_{\hat{c}_p} (x' - g, 0)$, lo que implica que $c_p(x'') \leq c_p(x' - g)$. Por el corolario 4.3, concluimos que x'' es una solución eficiente del problema (4.12). Además, se cumple que $c'(x'') \leq c'(x') + a \leq c'(x^*)$ con $c'(x'') \neq c'(x^*)$ pues $c_k(x') + a_k < c_k(x^*)$, por tanto, $x'' \in D_{x^*}$.

Como $c_p(x'') \leq c_p(x' - g)$ y $c_p(x' - g) \leq c_p(x')$, si $c_p(x'') < c_p(x')$ tenemos que $x'' \preceq_{\hat{c}_p} x'$ y esto contradice la minimalidad de x' .

Veamos qué sucede si $c_p(x'') = c_p(x')$. En este caso, obtenemos que $c_p(x' - g) = c_p(x')$. Como $(x' - g, c'(x^*) - c'(x') - a) \preceq_{\hat{c}_p} (x', c'(x^*) - c'(x'))$, debe cumplirse que $c_1(x' - g) \leq c_1(x')$, pero además tenemos que $c'(x' - g) = c'(x') + a$, entonces si $k \neq 1$, se cumple que $a_1 = 0$ y $c_1(x' - g) = c_1(x')$. Con este mismo razonamiento logramos establecer la igualdad de los costes c_i para $x' - g$ y x' con $i < k$. Cuando el índice es k se cumple que $c_k(x' - g) < c_k(x')$. Entonces como $c_p(x'') = c_p(x')$, $c_1(x'') \leq c_1(x'), \dots, c_{k-1}(x'') \leq c_{k-1}(x')$ y $c_k(x'') < c_k(x')$, podemos concluir que $x'' \preceq_{\hat{c}_p} x'$, pero esto contradice la minimalidad de x' . Por tanto, $a \geq 0$ con $a \neq 0$.

- Veamos que $a = c'(x^*) - c'(x')$. Sea $a \geq 0$ con $a \neq 0$ tal que $a \leq c'(x^*) - c'(x')$ pero $a \neq c'(x^*) - c'(x')$, entonces existe k tal que $a_k < c_k(x^*) - c_k(x')$. El punto (x', a) es factible para el problema $P_p(c'(x') + a)_{\preceq_{\hat{c}_p}}$, pero no es su solución óptima, pues $[g, a]$ lo reduce, esto es, el punto $(x' - g, 0)$ es factible para $P_p(c'(x') + a)_{\preceq_{\hat{c}_p}}$ con $c'(x' - g) = c'(x') + a$.

Ahora, si (x'', u) con $u \geq 0$ es solución de $P_p(c'(x') + a)_{\preceq_{\hat{c}_p}}$, por el corolario 4.3, tenemos que x'' es una solución eficiente del problema (4.12) con $x'' \neq x'$. Además, se cumple que $c'(x'') \leq c'(x') + a \leq c'(x^*)$ con $c'(x'') \neq c'(x^*)$ pues $c_k(x') + a_k < c_k(x^*)$, por tanto, $x'' \in D_{x^*}$.

Como $(x'', u) \preceq_{\hat{c}_p} (x', a)$, implica que $c_p(x'') \leq c_p(x')$, si $c_p(x'') < c_p(x')$ tenemos que $x'' \preceq_{\hat{c}_p} x'$ y esto contradice la minimalidad de x' .

Veamos qué sucede si $c_p(x'') = c_p(x')$. Tenemos que $c_p(x'') \leq c_p(x' - g)$ pues $(x'', u) \preceq_{\hat{c}_p} (x' - g, 0)$ y además $c_p(x' - g) \leq c_p(x')$, podemos concluir que $c_p(x' - g) = c_p(x')$. Como $(x' - g, c'(x^*) - c'(x') - a) \preceq_{\hat{c}_p}$

$(x', c'(x^*) - c'(x'))$, debe cumplirse que $c_1(x' - g) \leq c_1(x')$, pero además tenemos que $c'(x' - g) = c'(x') + a$ y $a \geq 0$, entonces $a_1 = 0$ y $c_1(x' - g) = c_1(x')$. Con este mismo razonamiento logramos establecer la igualdad de los costes c_i para $x' - g$ y x' con $i = 1, \dots, p - 1$, pero esto contradice que $a \neq 0$. Por tanto, $a = c'(x^*) - c'(x')$. \square

Observación 4.5. Dadas soluciones eficientes x' y x^* para el problema con p objetivos (4.12) como en la prueba del teorema anterior, no puede existir otra solución eficiente x'' tal que $c'(x') \leq c'(x'') \leq c'(x^*)$, pues de lo contrario, debería cumplirse que $c_p(x'') < c_p(x')$ y esto contradice la minimalidad de x' . Esto permite afirmar que el algoritmo que proponemos a continuación encuentra toda la frontera de Pareto.

4.3. Algoritmo para el caso multiobjetivo

El teorema 4.4 pone de manifiesto que procede formular un algoritmo para encontrar la frontera de Pareto de un problema lineal entero con p objetivos de manera recursiva. Iniciar con un objetivo, de manera que su solución sea una proyección de algún punto de la frontera con dos objetivos y luego calcular la frontera de Pareto con dos objetivos a partir de este punto. A continuación, calcular la frontera de Pareto con tres objetivos a partir de este conjunto, que también será una proyección de algunos puntos en la frontera de Pareto del problema con tres objetivos, y así sucesivamente. El siguiente resultado nos muestra que esto es posible con la ayuda del orden $\preceq_{\hat{e}_i}$ (primero comparo con c_1 , luego con $c_1, c_2, \dots, c_{i-1}, c_{i+1}, \dots, c_p$ y finalmente con un orden monomial).

Teorema 4.6. Sea x^* una solución eficiente del problema

$$\begin{aligned} \text{mín} \quad & c_1x, c_2x, \dots, c_ix \\ \text{s.a.} \quad & Ax = b \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.17)$$

tal que (x^*, t) con $t \geq 0$ es la única solución del problema $P_i(\epsilon_1, \dots, \epsilon_{i-1})_{\preceq_{\hat{e}_i}}$ con respecto al orden $\preceq_{\hat{e}_i}$, donde $c_k(x^*) + t = \epsilon_k$, $k = 1, \dots, i - 1$ y $i = 1, \dots, p - 1$, entonces x^* es eficiente para el problema

$$\begin{aligned} \text{mín} \quad & c_1x, c_2x, \dots, c_ix, c_{i+1}x \\ \text{s.a.} \quad & Ax = b \quad x \in \mathbb{Z}_{\geq 0}^n \end{aligned} \quad (4.18)$$

Demostración. Veamos que el teorema se cumple para $i = 1$: sea e_1 solución del problema $\text{mín}\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$ con respecto al orden $\preceq_{\hat{e}_1}$. Por el teorema 1.15 es suficiente probar que e_1 es solución de los problemas $P_1(c_2(e_1))$ y $P_2(c_1(e_1))$.

El punto e_1 es factible para $P_1(c_2(e_1))$ y supongamos que no es óptimo, entonces sea x' su solución, por tanto $c_1(x') < c_1(e_1)$. Como x' es factible para el problema $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$, tenemos que $c_1(e_1) \leq c_1(x')$, lo que nos conduce a una contradicción. Por tanto e_1 es solución de $P_1(c_2(e_1))$.

Por otro lado, e_1 es factible para $P_2(c_1(e_1))$ y supongamos que no es óptimo. Sea x'' su solución, por tanto $c_2(x'') < c_2(e_1)$ y $c_1(x'') \leq c_1(e_1)$. Como x'' es factible para el problema $\min\{c_1(x) : Ax = b, x \in \mathbb{N}^n\}$, tenemos que $c_1(e_1) \leq c_1(x'')$, por tanto $c_1(e_1) = c_1(x'')$. Luego por el orden $\preceq_{\hat{c}_1}$, debe cumplirse que $c_2(e_1) \leq c_2(x'')$, lo que nos conduce a una contradicción. Por tanto e_1 es solución de $P_2(c_1(e_1))$.

Ahora supongamos que x^* solución eficiente del problema (4.17) tal que (x^*, t) con $t \geq 0$ es la única solución del problema $P_i(\epsilon_1, \dots, \epsilon_{i-1})_{\preceq_{\hat{c}_i}}$ donde $c_k(x^*) + t = \epsilon_k$.

Primero notemos que si $(x^*, 0)$ es solución del problema $P_j(\epsilon'_1, \dots, \epsilon'_{j-1})_{\preceq_{\hat{c}_j}}$ con respecto al orden $\preceq_{\hat{c}_j}$ donde $\epsilon'_l = c_l(x^*)$, $l = 1, \dots, j-1$, entonces es solución eficiente del problema $\min\{c_1(x), \dots, c_j(x) : Ax = b, x \in \mathbb{N}^n\}$ con $j = 2, \dots, p-1$. La prueba es similar a la del teorema 4.1 pues, aunque para j objetivos el orden que se usa en el teorema es aquel que primero compara con c_j , luego con c_1, c_2, \dots, c_{j-1} y finaliza con un orden monomial, el orden $\preceq_{\hat{c}_j}$ es una extensión, en el sentido de incluir los vectores $c_{j+1}, c_{j+2}, \dots, c_p$ antes de comparar con el orden monomial. De acuerdo con esto, para probar que x^* es eficiente para el problema (4.18), es suficiente mostrar que $(x^*, 0)$ es solución del problema $P_{i+1}(\epsilon_1, \dots, \epsilon_i)_{\preceq_{\hat{c}_{i+1}}}$ con respecto al orden $\preceq_{\hat{c}_i}$ donde $c_k(x^*) = \epsilon_k$, $k = 1, \dots, i$.

Tenemos que $(x^*, 0)$ es factible para el problema $P_{i+1}(\epsilon_1, \dots, \epsilon_i)_{\preceq_{\hat{c}_{i+1}}}$ y supongamos que (x', m) con $m \geq 0$, es su solución. Por tanto $(x', m) \preceq_{\hat{c}_{i+1}} (x^*, 0)$, lo que implica que $c_{i+1}(x') \leq c_{i+1}(x^*)$; además $c_k(x') \leq c_k(x^*)$, con $k = 1, \dots, i$. Estas últimas igualdades nos permiten afirmar que (x', s) con $s \geq 0$, es factible para $P_i(\epsilon_1, \dots, \epsilon_{i-1})_{\preceq_{\hat{c}_i}}$, luego $c_i(x^*) \leq c_i(x')$ y en consecuencia, $c_i(x^*) = c_i(x')$. Por el orden $\preceq_{\hat{c}_i}$, tenemos que $c_1(x^*) \leq c_1(x')$, pero nuevamente obtenemos que $c_1(x^*) = c_1(x')$. Con este mismo razonamiento concluimos que $c_k(x') = c_k(x^*)$, con $k = 1, \dots, i-1, i+1$.

Para $k = i+2, \dots, p$, por el orden $\preceq_{\hat{c}_i}$ tenemos que $c_k(x^*) \leq c_k(x')$ y por el orden $\preceq_{\hat{c}_{i+1}}$, se cumple que $c_k(x') \leq c_k(x^*)$, por tanto, $c_k(x') = c_k(x^*)$. Entonces en ambos casos debemos comparar con el mismo orden monomial \preceq , llevándonos a establecer que $x' = x^*$. \square

Finalmente presentamos un algoritmo para encontrar toda la frontera de Pareto para el problema multiobjetivo (4.12), dado en pseudo-código como Algoritmo 5, que requiere de la siguiente notación.

Sea $\mathcal{T}_{\preceq_{\hat{c}_i}} \subseteq \mathbb{Z}^{n+(i-1)}$, el test-set para la familia de problemas $P_i(\epsilon_1, \dots, \epsilon_{i-1})_{\preceq_{\hat{c}_i}}$. Para cada $x \in \mathbb{N}^n$ definimos el conjunto $G_x = \{[g, a] \in \mathcal{T}_{\preceq_{\hat{c}_i}} : g \leq x, a \geq 0, a \neq 0\}$ y el conjunto \tilde{G}_x como aquel cuyos elementos de G_x tienen las últimas $i - 1$ componentes no comparables según el orden \leq (componente a componente), esto es, dados $[g, a]$ y $[g, a']$ en \tilde{G}_x se cumple que a y a' son no comparables. Notemos que cuando consideramos dos objetivos ($i = 2$), el conjunto \tilde{G}_x solo tiene un vector.

Explicaremos brevemente los pasos del algoritmo:

- En el algoritmo 5 iniciamos con e_1 , por el teorema 4.6, $(e_1, 0)$ es la única solución del problema $P_2(c_1(e_1))_{\preceq_{\hat{c}_2}}$ y en consecuencia eficiente para el problema biobjetivo correspondiente.
- Para $i = 2$ obtenemos las soluciones eficientes del problema de minimizar los dos primeros objetivos que generan la frontera de Pareto de dicho problema y que además de ser una proyección de algunas soluciones eficientes del problema con tres objetivos, son soluciones $\preceq_{\hat{c}_p}$ -eficientes.
- Para $i \geq 3$, al inicio de este paso, el conjunto \mathcal{X}' tiene las soluciones eficientes que generan la frontera de Pareto para el problema con $i - 1$ objetivos que a su vez son soluciones $\preceq_{\hat{c}_p}$ -eficientes. El teorema 4.4 nos permite calcular todas las soluciones eficientes que generan la frontera de Pareto para el problema con i objetivos que además son $\preceq_{\hat{c}_p}$ -eficientes.

De manera similar al caso biobjetivo, el algoritmo 5 termina, pues el conjunto factible $\mathcal{X} \neq \emptyset$ es finito. Los teoremas presentados nos permiten asegurar que los elementos del conjunto \mathcal{X}' , que obtenemos con el algoritmo 5, son soluciones eficientes del problema con p objetivos y además que su imagen bajo $c = (c_1, c_2, \dots, c_p)$ es toda la frontera de Pareto \mathcal{F} .

Algoritmo 5 Algoritmo para obtener la frontera de Pareto exacta de un problema de optimización lineal entera con p objetivos ($p \geq 2$)

- 1: **Entrada:** el vector de costes $c = (c_1, c_2, \dots, c_p)$, la matriz de restricciones A y el vector b del problema (4.12).
 - 2: Calcular $\min c_1 x$ s.a $Ax = b$ utilizando el test-set asociado con respecto al orden $\preceq_{\hat{c}_1}$. Notamos e_1 al óptimo.
 - 3: $\mathcal{X}' := \{e_1\}$.
 - 4: $P := \{e_1\}$.
 - 5: **Para** $i = 2$ **hasta** p **hacer:**
 - 6: Calcular $\mathcal{T}_{\preceq_{\hat{c}_i}}$, el test-set para la familia de problemas
 - 7: $P_i(\epsilon_1, \dots, \epsilon_{i-1})_{\preceq_{\hat{c}_i}}$.
 - 8: **Para todo** $x \in P$ **hacer:**
 - 9: $P := P \setminus \{x\}$.
 - 10: Calcular \tilde{G}_x
 - 11: **Si** $\tilde{G}_x \neq \emptyset$ **entonces:**
 - 12: $G_{\text{saltos}} := \{(x, a) : [g, a] \in \tilde{G}_x\}$
 - 13: $j := |G_{\text{saltos}}|$
 - 14: **Para** $k = 1$ **hasta** j **hacer:**
 - 15: Calcular el óptimo de $P_i(\epsilon_1, \dots, \epsilon_{i-1})_{\preceq_{\hat{c}_i}}$ con solución inicial (x_k, a_k) , tal que $(\epsilon_1, \dots, \epsilon_{i-1}) = (c_1(x_k) + a_{k,1}, \dots, c_{i-1}(x_k) + a_{k,i-1})$; que notamos (y, t) .
 - 16: **Si** $y \notin \mathcal{X}'$: **entonces**
 - 17: $\mathcal{X}' := \mathcal{X}' \cup \{y\}$
 - 18: $P := P \cup \{y\}$
 - 19: **Fin Si**
 - 20: **Fin Para**
 - 21: **Fin Si**
 - 22: **Fin Para**
 - 23: **Fin Si**
 - 24: **Fin Para**
 - 25: $P := \mathcal{X}'$
 - 26: **Fin Para**
 - 27: **Salida:** El conjunto \mathcal{X}' tal que $c(\mathcal{X}') = \mathcal{F}$.
-

4.4. Algunos ejemplos de problemas con tres objetivos

Para calcular la frontera de Pareto de un problema lineal entero multi-objetivo, algunos algoritmos, como el presentado en [46], requieren el cálculo del punto ideal y nadir, siendo este último difícil de calcular de forma exacta, como se mencionó en el capítulo 1. Un problema en el que se evidencia la dificultad de obtener dicho punto mediante la tabla de pagos fue el ejemplo 1.11. A continuación retomamos este ejemplo para encontrar su frontera de Pareto usando el algoritmo 5, donde resaltamos que no tenemos que calcular el punto nadir para su implementación.

Ejemplo 4.7. Para encontrar la frontera de Pareto del problema

$$\begin{aligned} \text{mín}(c_1(x) &= & -11x_2 & -11x_3 & -12x_4 & -9x_5 & -9x_6 & +9x_7, \\ c_2(x) &= & -11x_1 & & -11x_3 & -9x_4 & -12x_5 & -9x_6 & +9x_7, \\ c_3(x) &= & -11x_1 & -11x_2 & & -9x_4 & -9x_5 & -12x_6 & -12x_7) \end{aligned}$$

$$\text{s. a. } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 1, \quad x \in \mathbb{Z}_{\geq 0}^7 \quad (4.19)$$

usando el algoritmo 5, primero resolvemos el problema

$$\text{mín } c_1(x) = -11x_2 - 11x_3 - 12x_4 - 9x_5 - 9x_6 + 9x_7 \quad (4.20)$$

$$\text{s. a. } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 1, \quad x \in \mathbb{Z}_{\geq 0}^7$$

utilizando el test-set asociado, con respecto al orden $\preceq_{\hat{c}_1}$. Así, obtenemos el punto $e_1 = (0, 0, 0, 1, 0, 0, 0)$. Luego $\mathcal{X}' = \{e_1\}$ y $P = \{e_1\}$.

Ahora, damos inicio al bucle “Para” (línea 5) con $i = 2$. A continuación calculamos el test-set $\mathcal{T}_{\preceq_{\hat{c}_2}}$, para la familia de problemas $P_2(\epsilon_1)_{\preceq_{\hat{c}_2}}$, que corresponde a:

$$\begin{aligned} \mathcal{T}_{\preceq_{\hat{c}_2}} = & \{(0, 0, -2, 1, 1, 0, 0, -1), (0, 0, 0, 0, -1, 0, 1, -18), \\ & (0, 0, 0, 0, -1, 1, 0, 0), (0, 1, -1, 0, 0, 0, 0, 0), (1, 0, 0, 0, -1, 0, 0, -9), \\ & (0, 0, -1, 1, 0, 0, 0, 1), (0, 0, 1, 0, -1, 0, 0, 2)\} \end{aligned}$$

Seleccionamos e_1 para iniciar, por tanto

- $P = \emptyset$ y $\tilde{G}_{e_1} = \{(0, 0, -1, 1, 0, 0, 0, 1)\}$.
- $G_{\text{saltos}} = \{(e_1, 1)\}$ y $|G_{\text{saltos}}| = 1$.
- Calculamos el óptimo de $P_2(c_1(e_1) + 1)_{\preceq_{\hat{c}_2}}$, es decir, realizamos una reducción:

$$(e_1, 1) = (0, 0, 0, 1, 0, 0, 0, 1) \text{ se reduce a } (0, 0, 1, 0, 0, 0, 0, 0) = (y_1, 0)$$

- Como $y_1 \notin \mathcal{X}'$, entonces $\mathcal{X}' = \{e_1, y_1\}$ y $P = \{y_1\}$.

Después seleccionamos y_1 , luego

- $P = \emptyset$ y $\tilde{G}_{y_1} = \{(0, 0, 1, 0, -1, 0, 0, 2)\}$.
- $G_{\text{saltos}} = \{(y_1, 2)\}$ y $|G_{\text{saltos}}| = 1$.
- Calculamos el óptimo de $P_2(c_1(y_1) + 2)_{\preceq \hat{e}_2}$, es decir, realizamos una reducción:

$$(y_1, 2) = (0, 0, 1, 0, 0, 0, 0, 2) \text{ se reduce a } (0, 0, 0, 0, 1, 0, 0, 0) = (y_2, 0)$$

- Como $y_2 \notin \mathcal{X}'$, entonces $\mathcal{X}' = \{e_1, y_1, y_2\}$ y $P = \{y_2\}$.

Cuando seguimos con y_2 , tenemos que $P = \emptyset$ pero $\tilde{G}_{y_2} = \emptyset$, por tanto el bucle “Para” (línea 8) termina y actualizamos P , esto es, $P = \mathcal{X}' = \{e_1, y_1, y_2\}$.

Continuamos con $i = 3$ en el bucle “Para” (línea 5) y calculamos el test-set $\mathcal{T}_{\preceq \hat{e}_3}$, para la familia de problemas $P_3(\epsilon_1, \epsilon_2)_{\preceq \hat{e}_3}$, que corresponde a:

$$\begin{aligned} \mathcal{T}_{\preceq \hat{e}_2} = & \{(-2, 0, 0, 0, 1, 1, 0, 18, -1), (0, -2, 0, 1, 0, 1, 0, -1, 18), \\ & (0, 0, 0, 0, 0, -1, 1, -18, -18), (0, 0, 1, -1, -1, 1, 0, -1, -1), \\ & (0, 0, 1, -1, 0, 0, 0, -1, 2), (0, 0, 1, 0, -1, 0, 0, 2, -1), \\ & (0, 1, 0, -1, 1, -1, 0, -1, -6), (0, 1, 0, 0, 0, -1, 0, 2, -9), \\ & (0, 1, 1, -2, 0, 0, 0, -2, -7), (1, 0, 0, 0, 0, -1, 0, -9, 2), \\ & (1, 0, 0, 1, -1, -1, 0, -6, -1), (1, 0, 1, 0, -2, 0, 0, -7, -2), \\ & (1, 1, 0, 0, 0, -2, 0, -7, -7), (-1, 0, 0, 0, 1, 0, 0, 9, 1), \\ & (0, -1, 0, 1, 0, 0, 0, 1, 9), (0, 0, 0, 0, 1, -1, 0, 0, 3), \\ & (0, 0, 0, 1, 0, -1, 0, 3, 0), (0, 0, 2, -1, -1, 0, 0, 1, 1)\} \end{aligned}$$

Iniciamos con e_1 , luego

- $P = \{y_1, y_2\}$ y $\tilde{G}_{e_1} = \{(0, -1, 0, 1, 0, 0, 0, 1, 9), (0, 0, 0, 1, 0, -1, 0, 3, 0)\}$.
- $G_{\text{saltos}} = \{(e_1, 1, 9), (e_1, 3, 0)\}$ y $|G_{\text{saltos}}| = 2$.
- Calculamos el óptimo de $P_3(c_1(e_1)+1, c_2(e_1)+9)_{\preceq \hat{e}_3}$, es decir, realizamos una reducción:

$$(e_1, 1, 9) \text{ se reduce a } (0, 1, 0, 0, 0, 0, 0, 0, 0) = (z_1, 0, 0)$$

- Como $z_1 \notin \mathcal{X}'$, entonces $\mathcal{X}' = \{e_1, y_1, y_2, z_1\}$ y $P = \{y_1, y_2, z_1\}$.

- Luego calculamos el óptimo de $P_3(c_1(e_1) + 3, c_2(e_1))_{\preceq_{\hat{e}_3}}$, es decir, realizamos una reducción:

$$(e_1, 3, 0) \text{ se reduce a } (0, 0, 0, 0, 0, 1, 0, 0, 0) = (z_2, 0, 0)$$

- Como $z_2 \notin \mathcal{X}'$, entonces $\mathcal{X}' = \{e_1, y_1, y_2, z_1, z_2\}$ y $P = \{y_1, y_2, z_1, z_2\}$.

Continuamos con y_1 , luego $P = \{y_2, z_1, z_2\}$, pero $\tilde{G}_{y_1} = \emptyset$. Por tanto, revisamos otro punto de P .

Al seguir con y_2 tenemos que

- $P = \{z_1, z_2\}$ y $\tilde{G}_{y_2} = \{(-1, 0, 0, 0, 1, 0, 0, 9, 1), (0, 0, 0, 0, 1, -1, 0, 0, 3)\}$.

- $G_{\text{saltos}} = \{(y_2, 9, 1), (y_2, 0, 3)\}$ y $|G_{\text{saltos}}| = 2$.

- Calculamos el óptimo de $P_3(c_1(y_2)+9, c_2(y_2)+1)_{\preceq_{\hat{e}_3}}$, es decir, realizamos una reducción:

$$(y_2, 9, 1) \text{ se reduce a } (1, 0, 0, 0, 0, 0, 0, 0, 0) = (z_3, 0, 0)$$

- Como $z_3 \notin \mathcal{X}'$, entonces $\mathcal{X}' = \{e_1, y_1, y_2, z_1, z_2, z_3\}$ y $P = \{z_1, z_2, z_3\}$.

- Luego calculamos el óptimo de $P_3(c_1(y_2), c_2(y_2) + 3)_{\preceq_{\hat{e}_3}}$, es decir, realizamos una reducción:

$$(y_2, 0, 3) \text{ se reduce a } (0, 0, 0, 0, 0, 1, 0, 0, 0) = (z_2, 0, 0)$$

- Como $z_2 \in \mathcal{X}'$, entonces \mathcal{X}' y P no se modifican.

Cuando revisamos z_1 , tenemos que $P = \{z_2, z_3\}$ y $\tilde{G}_{z_1} = \emptyset$. Continuando con z_2 , tenemos que $P = \{z_3\}$ y $\tilde{G}_{z_2} = \emptyset$; finalmente con z_3 , tenemos que $P = \emptyset$ y $\tilde{G}_{z_3} = \emptyset$, por tanto, el algoritmo termina y el conjunto factible cuya imagen por $c = (c_1, c_2, c_3)$ es la frontera de Pareto para el problema (4.19) es $\mathcal{X}' = \{e_1, y_1, y_2, z_1, z_2, z_3\}$.

Al aplicar el método AUGMECON2 [46] para solucionar el problema (4.19), primero se realiza una estimación del vector nadir mediante la matriz de pagos dada por la tabla 4.1.

Luego, al ejecutar el algoritmo solo se obtienen 4 puntos no dominados correspondientes a las soluciones eficientes e_1, y_2, z_2 y z_3 , empleando un tiempo CPU de 1,47 sg. Nuestro algoritmo además de generar toda la frontera de Pareto, emplea menos tiempo CPU en su ejecución: 0,08 sg en el cálculo de los test-set utilizados y 0,004 en los otros procesos.

	x^1	x^2	x^3
f_1	-12	-9	-9
f_2	-9	-12	-9
f_3	-9	-9	-12

Tabla 4.1

Ejemplo 4.8. Consideremos el problema

$$\begin{aligned}
 \text{mín} \quad & c_1(x) = -27x_1 - 95x_2 - 67x_3 - 75x_4 - 67x_5 - 57x_6 \\
 & c_2(x) = -46x_1 - 77x_2 - 73x_3 - 81x_4 - 77x_5 - 60x_6 \\
 & c_3(x) = -55x_1 - 39x_2 - 28x_3 - 80x_4 - 61x_5 - 70x_6 \\
 \text{s.a.} \quad & 85x_1 + 12x_2 + 15x_3 + 17x_4 + 37x_5 + 86x_6 + z = 126 \\
 & x \in \mathbb{Z}_{\geq 0}^7
 \end{aligned} \tag{4.21}$$

Mediante el algoritmo 5, vamos a encontrar la frontera de Pareto. Para esto, primero solucionamos el problema

$$\begin{aligned}
 \text{mín} \quad & c_1(x) = -27x_1 - 95x_1 - 67x_3 - 75x_4 - 67x_5 - 57x_6 \\
 \text{s.a.} \quad & 85x_1 + 12x_2 + 15x_3 + 17x_4 + 37x_5 + 86x_6 + z = 126 \\
 & x \in \mathbb{Z}_{\geq 0}^7
 \end{aligned} \tag{4.22}$$

recurriendo al test-set asociado, con respecto al orden $\preceq_{\hat{c}_1}$, cuyo óptimo corresponde al punto $e_1 = (0, 10, 0, 0, 0, 0, 6)$. Luego $\mathcal{X}' = \{e_1\}$ y $P = \{e_1\}$.

A continuación, calculamos el test-set $\mathcal{T}_{\preceq_{\hat{c}_2}}$, para la familia de problemas $P_2(\epsilon_1)_{\preceq_{\hat{c}_2}}$:

$$\begin{aligned}
 \mathcal{T}_{\preceq_{\hat{c}_2}} = \quad & \{(0, -7, 0, 0, 0, 1, -2, -608), (0, -4, 0, 3, 0, 0, -3, -155), \\
 & (0, -3, 0, 0, 1, 0, -1, -218), (0, -3, 0, 2, 0, 0, 2, -135), \\
 & (0, -2, 0, 1, 0, 0, 7, -115), (0, -1, 0, 0, 0, 0, 12, -95), \\
 & (0, -1, 1, 0, 0, 0, -3, -28), (0, 1, 0, -1, 0, 0, 5, 20), \\
 & (1, -7, 0, 0, 0, 0, -1, -638)\}
 \end{aligned}$$

Partiendo de e_1 , modificamos la última variable de holgura según nos indique el test-set, y seguimos aplicando el mismo procedimiento a los puntos nuevos que vamos obteniendo. Esquemáticamente, esto corresponde a:

$$\begin{array}{c} e_1 \\ | \\ (20) \\ | \\ \underbrace{(0, 9, 0, 1, 0, 0, 1, 0)} \\ (y_1, 0) \end{array}$$

Aquí terminamos pues $\tilde{G}_{y_1} = \emptyset$ y no quedan más puntos por revisar, es decir, $P = \emptyset$. Entonces actualizamos P , esto es, $P = \mathcal{X}' = \{e_1, y_1\}$ y calculamos el test-set $\mathcal{T}_{\leq \epsilon_3}$, para la familia de problemas $P_3(\epsilon_1, \epsilon_2)_{\leq \epsilon_3}$:

$$\begin{aligned} \mathcal{T}_{\leq \epsilon_2} = \{ & (0, -1, 0, 0, 0, 0, 12, -95, -77), (0, -1, 1, 0, 0, 0, -3, -28, -4), \\ & (0, 0, 0, -5, 0, 1, -1, -318, -345), (0, 0, 0, -2, 1, 0, -3, -83, -85), \\ & (1, 0, 0, -5, 0, 0, 0, -348 - 359), (0, 1, 0, -1, 0, 0, 5, 20, -4), \\ & (0, 2, 0, -1, 0, 0, -7, \mathbf{115, 73}), (0, 3, 0, -2, 0, 0, -2, \mathbf{135, 69}), \\ & (0, 4, 0, -3, 0, 0, 3, \mathbf{155, 65}) \} \end{aligned}$$

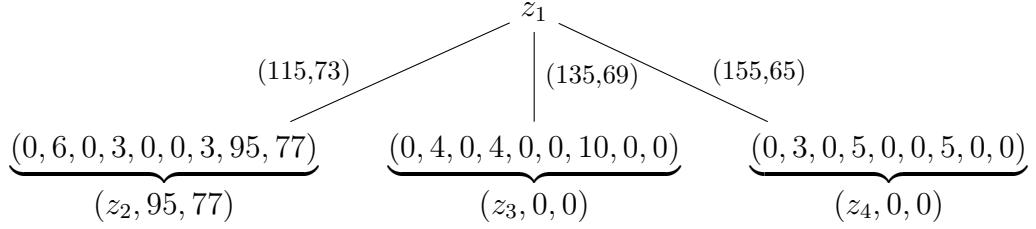
Luego, para cada punto de P modificamos las dos últimas variables de holgura según indique el test-set, al igual que para cada punto nuevo que vamos obteniendo. Ilustramos este proceso con los siguientes esquemas:

$$\begin{array}{c} e_1 \\ \swarrow \quad \downarrow \quad \searrow \\ \begin{array}{ccc} (115,73) & & (135,69) \quad (155,65) \\ \underbrace{(0, 9, 0, 1, 0, 0, 1, 95, 77)} & \underbrace{(0, 7, 0, 2, 0, 0, 8, 0, 0)} & \underbrace{(0, 6, 0, 3, 0, 0, 3, 0, 0)} \\ (y_1, 95, 77) & (z_1, 0, 0) & (z_2, 0, 0) \end{array} \end{array}$$

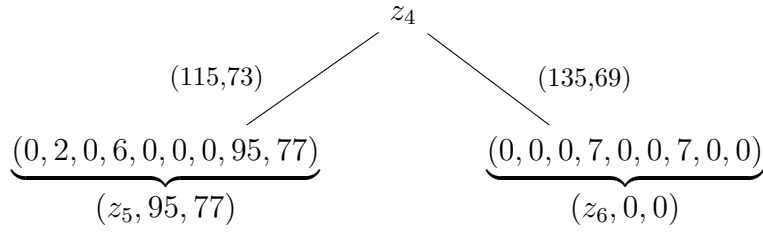
$P = \{y_1, z_1, z_2\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2\}$. En la figura 4.1 ilustramos el proceso en el espacio objetivo proyectado sobre el plano c_1, c_2 .

$$\begin{array}{c} y_1 \\ \swarrow \quad \searrow \\ \begin{array}{cc} (115,73) & (135,69) \\ \underbrace{(0, 7, 0, 2, 0, 0, 8, 0, 0)} & \underbrace{(0, 6, 0, 3, 0, 0, 3, 0, 0)} \\ (z_1, 0, 0) & (z_2, 0, 0) \end{array} \end{array}$$

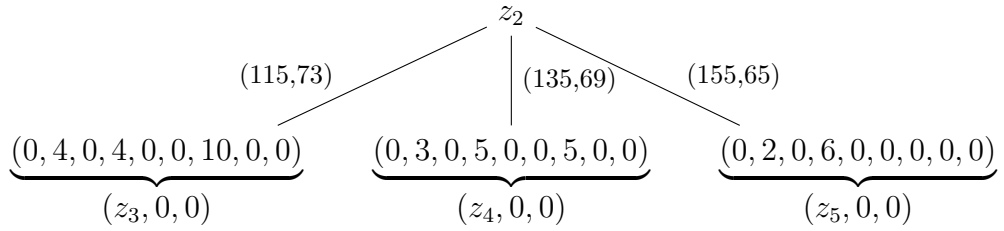
$P = \{z_1, z_2\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2\}$ (figura 4.2).



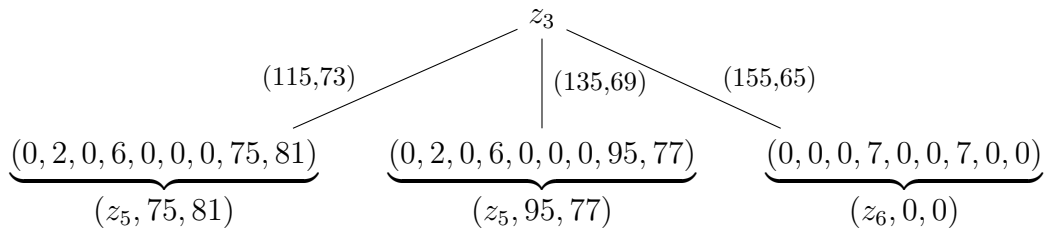
$P = \{z_2, z_3, z_4\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2, z_3, z_4\}$ (figura 4.3).



$P = \{z_2, z_3, z_5, z_6\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2, z_3, z_4, z_5, z_6\}$ (figura 4.4).



$P = \{z_3, z_5, z_6\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2, z_3, z_4, z_5, z_6\}$ (figura 4.5).



$P = \{z_5, z_6\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2, z_3, z_4, z_5, z_6\}$ (figura 4.6).

$$\begin{array}{c} z_5 \\ | \\ (115,73) \\ | \\ \underbrace{(0, 0, 0, 7, 0, 0, 7, 0, 0)}_{(z_6, 0, 0)} \end{array}$$

$P = \{z_6\}$ y $\mathcal{X}' = \{e_1, y_1, z_1, z_2, z_3, z_4, z_5, z_6\}$ (figura 4.7).

Terminamos el algoritmo, pues $\tilde{G}_{z_6} = \emptyset$ y no quedan más puntos por revisar, es decir, $P = \emptyset$. Entonces $\mathcal{X}' = \{e_1, y_1, z_1, z_2, z_3, z_4, z_5, z_6\}$ es el conjunto de soluciones eficientes tal que $c(\mathcal{X}') = \mathcal{F}$.

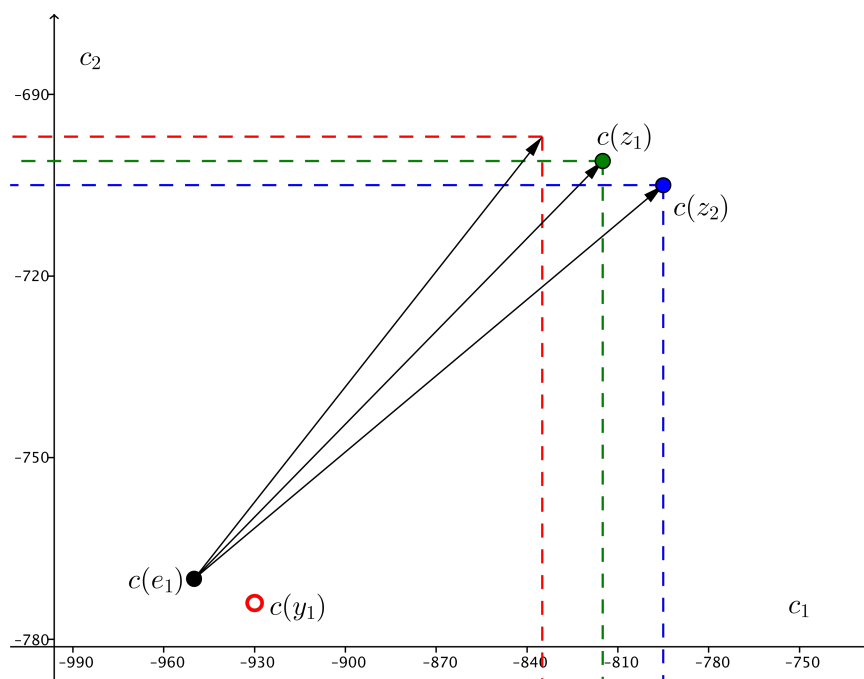


Figura 4.1

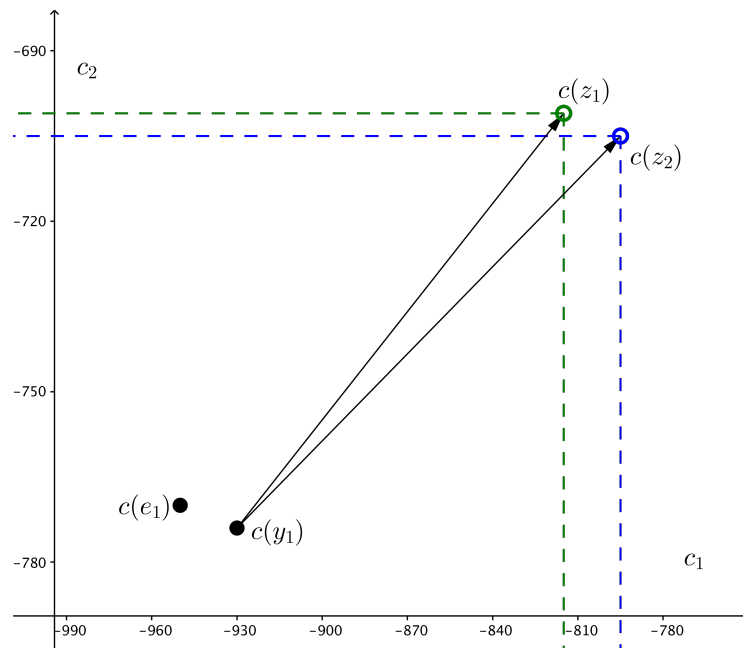


Figura 4.2

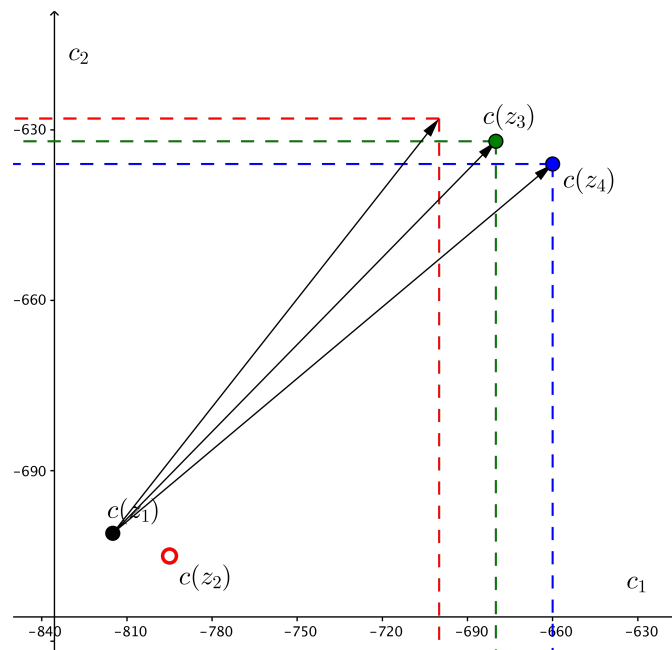


Figura 4.3

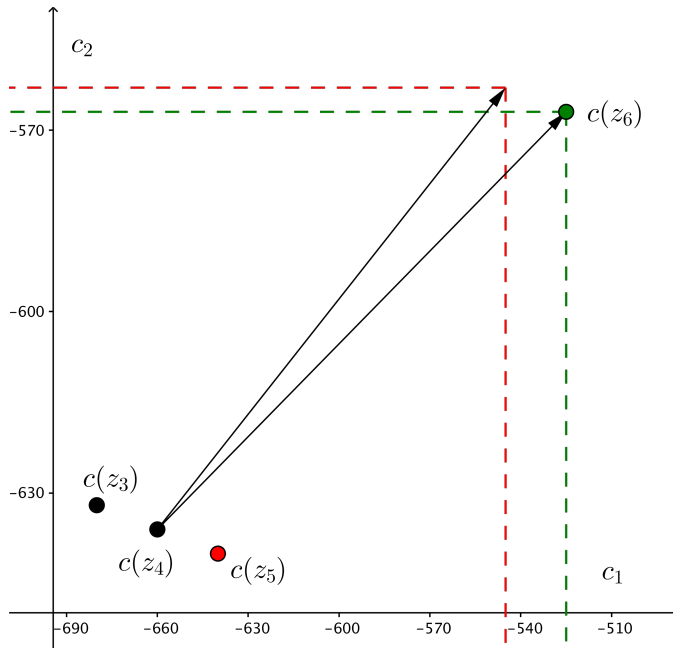


Figura 4.4

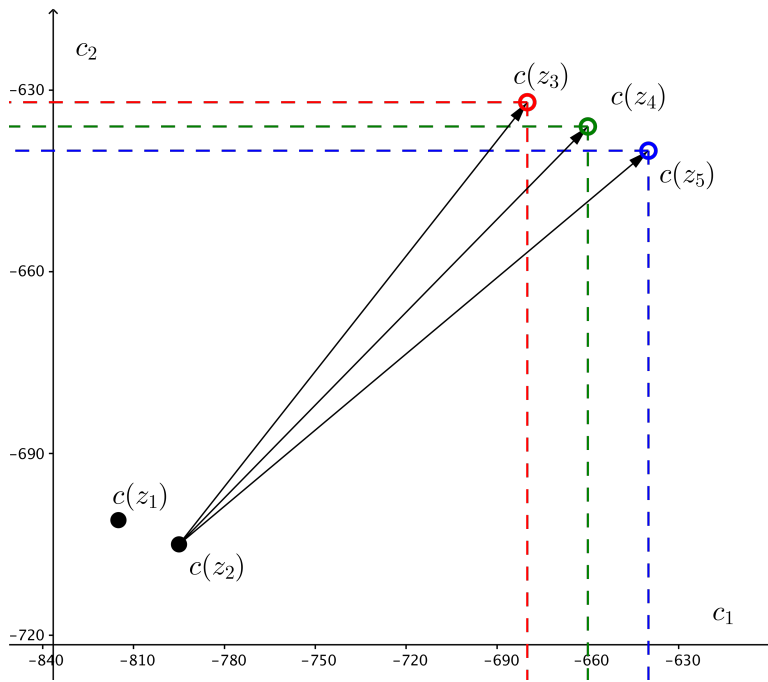


Figura 4.5

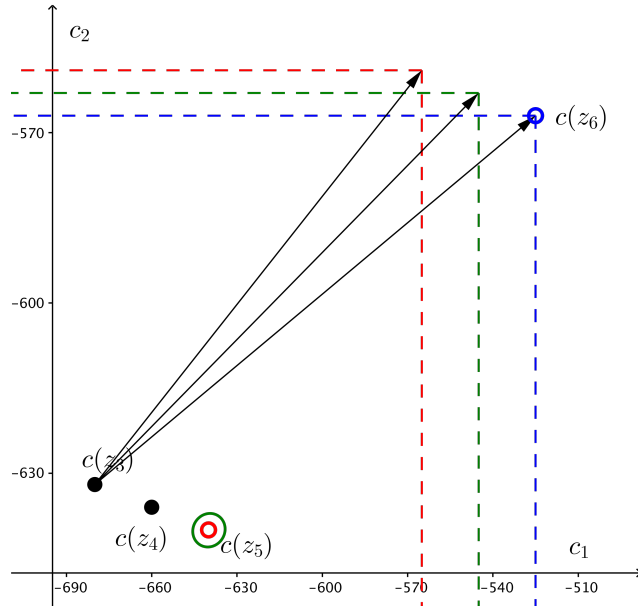


Figura 4.6

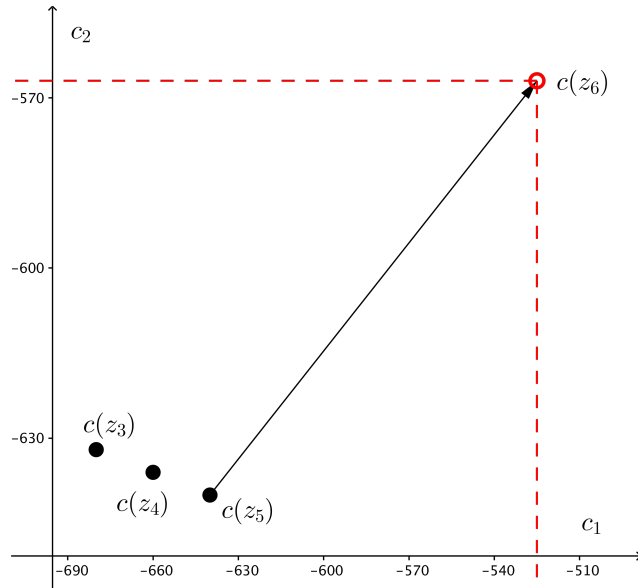


Figura 4.7

4.5. Aplicación del método en RAP triobjetivo

Aplicamos nuestro algoritmo a un caso particular de un problema RAP para sistemas serie-paralelo, presentado en [13], el cual consta de tres sub-

sistemas, $s = 3$, con $m_1 = 5$, $m_2 = 4$ y $m_3 = 5$. Además, cada subsistema tiene como máximo siete componentes, $n_{\text{máx},1} = n_{\text{máx},2} = n_{\text{máx},3} = 7$. Luego, debemos resolver tres problemas de optimización lineal multiobjetivo, esto es, minimizar las funciones (4.7), (4.8) y (4.9) sujetas a la restricción (4.10), para cada subsistema, teniendo en cuenta los parámetros de fiabilidad (r_{ij}), coste (c_{ij}) y peso (w_{ij}) presentados en la tabla 4.2.

Tipo de componente	Subsistema i								
	1			2			3		
j	r_{1j}	c_{1j}	w_{1j}	r_{2j}	c_{2j}	w_{2j}	r_{3j}	c_{3j}	w_{3j}
1	0.94	9	9	0.97	12	5	0.96	10	6
2	0.91	6	6	0.86	3	7	0.89	6	8
3	0.89	6	4	0.70	2	3	0.72	4	2
4	0.75	3	7	0.66	2	4	0.71	3	4
5	0.72	2	8				0.67	2	4

Tabla 4.2: Parámetros de cada subsistema del problema RAP serie-paralelo con $s = 3$.

En el caso de la función de fiabilidad no tenemos coeficientes enteros. Los coeficientes relativos a la fiabilidad $\log(1 - r_{ij})$, se multiplican por 10, 100, \dots , y dichos valores se redondean. Podemos observar como los test-sets y la frontera de Pareto estabiliza a partir de pocas cifras decimales: en el caso de los subsistemas 1 y 2 basta multiplicar por 10, y en el subsistema 3, por 100.

En las tablas 4.3 a 4.14, se señalan los coeficientes utilizados para las funciones a minimizar junto con el tamaño de la frontera de Pareto ($|\mathcal{F}|$), el tamaño del último test-set calculado ($|\mathcal{T}|$) y el tiempo CPU que tarda el algoritmo 5 en obtener la frontera, discriminado como el tiempo que emplea 4ti2 en calcular todos los test-sets requeridos (4ti2) y el tiempo restante (t_r).

En todos los subsistemas multiplicamos por 100 y redondeamos. Luego realizamos el producto cartesiano y posterior filtrado (comparación dos a dos) de las soluciones no dominadas de los subsistemas 1 y 3; a continuación se hace el producto cartesiano con el subsistema 2 restante. Finalmente obtenemos un conjunto con 6112 puntos en 351 sg, mientras que el método propuesto en [13] identifica los 6112 puntos no dominados en 1728 sg. Nuestro método invierte la mayor parte del tiempo en el filtrado, el cálculo de la frontera para cada subsistema es casi despreciable. En [13] no se especifica el reparto de tiempo.

Subsistema 1				Subsistema 1			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}	Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-28	9	9	1	-281	9	9
2	-24	6	6	2	-241	6	6
3	-22	6	4	3	-221	6	4
4	-14	3	7	4	-139	3	7
5	-13	2	8	5	-127	2	8
$ \mathcal{F} $		311		$ \mathcal{F} $		311	
$ \mathcal{T} $		48		$ \mathcal{T} $		47	
Algoritmo 5	4ti2	0.08 sg		Algoritmo 5	4ti2	0.08 sg	
	t_r	0.49 sg			t_r	0.36 sg	

Tabla 4.3

Tabla 4.4

Subsistema 1				Subsistema 1			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}	Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-2813	9	9	1	-28134	9	9
2	-2408	6	6	2	-24079	6	6
3	-2207	6	4	3	-22073	6	4
4	-1386	3	7	4	-13863	3	7
5	-1273	2	8	5	-12730	2	8
$ \mathcal{F} $		311		$ \mathcal{F} $		311	
$ \mathcal{T} $		48		$ \mathcal{T} $		48	
Algoritmo 5	4ti2	0.07 sg		Algoritmo 5	4ti2	0.08 sg	
	t_r	0.44 sg			t_r	0.36 sg	

Tabla 4.5

Tabla 4.6

4.6. Resultados Computacionales

Como destacamos anteriormente el problema de la mochila binario es el más utilizado para estudiar el alcance y eficiencia de los diferentes métodos para cualquier número de objetivos, véanse [40], [46], [41], [60], [35] y [55].

Aunque nuestro método no permite tratar de forma eficiente un gran número de variables en el caso binario, al comparar con AUGMECON2 el caso de 3 objetivos y 10 variables sin correlación lo batimos ampliamente (tabla 4.15). En todos estos casos hemos tenido que proporcionar a AUG-

Subsistema 2			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-35	12	5
2	-20	3	7
3	-12	2	3
4	-11	2	4
$ \mathcal{F} $		119	
$ \mathcal{T} $		18	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.03 sg	

Tabla 4.7

Subsistema 2			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-351	12	5
2	-197	3	7
3	-120	2	3
4	-108	2	4
$ \mathcal{F} $		119	
$ \mathcal{T} $		19	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.03 sg	

Tabla 4.8

Subsistema 2			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-3507	12	5
2	-1966	3	7
3	-1204	2	3
4	-1079	2	4
$ \mathcal{F} $		119	
$ \mathcal{T} $		19	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.02 sg	

Tabla 4.9

Subsistema 2			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-35066	12	5
2	-19661	3	7
3	-12040	2	3
4	-10788	2	4
$ \mathcal{F} $		119	
$ \mathcal{T} $		19	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.02 sg	

Tabla 4.10

MECON2 un parámetro que no modela toda la malla necesaria, pero que al menos proporciona aquellos puntos eficientes con costes acotados por los valores de la tabla de pagos que él mismo calcula. En [45] se reconoce la necesidad del punto nadir para el cálculo de la frontera exacta. En esta familia de ejemplos en un 57% de las instancias AUGMECON2 no proporciona la frontera completa. Por esta razón no se ha comparado nuestro algoritmo con AUGMECON2. Nuestro algoritmo no es competitivo con el tamaño de problemas mochila binaria que aparecen en la literatura.

De nuevo, el problema de la mochila multiobjetivo no acotado es al que mejor se adecua nuestro método. Tal y como se explica en [6], para más de 2 objetivos los casos tratados serán la generalización natural de A y C. En

Subsistema 3			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-32	10	6
2	-22	6	8
3	-13	4	2
4	-12	3	4
5	-11	2	4
$ \mathcal{F} $		297	
$ \mathcal{T} $		32	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.09 sg	

Tabla 4.11

Subsistema 3			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-322	10	6
2	-221	6	8
3	-127	4	2
4	-124	3	4
5	-111	2	4
$ \mathcal{F} $		388	
$ \mathcal{T} $		48	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.2 sg	

Tabla 4.12

Subsistema 3			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-3219	10	6
2	-2207	6	8
3	-1273	4	2
4	-1238	3	4
5	-1109	2	4
$ \mathcal{F} $		388	
$ \mathcal{T} $		40	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.37 sg	

Tabla 4.13

Subsistema 3			
Tipo de comp. j	Aprox. $\log(1 - r_{1j})$	c_{1j}	w_{1j}
1	-32189	10	6
2	-22073	6	8
3	-12730	4	2
4	-12379	3	4
5	-11087	2	4
$ \mathcal{F} $		388	
$ \mathcal{T} $		40	
Algoritmo 5	4ti2	0.08 sg	
	t_r	0.42 sg	

Tabla 4.14

dichas tablas se han ejecutado 10 instancias por cada caso.

Queremos observar como este problema no se ha tratado anteriormente en la literatura para más de dos objetivos. Por otra parte, Kirlik y Sayin [35] tratan el problema de la mochila binario para 3, 4 y 5 objetivos. Ellos reportan como límite para su método el caso de 5 objetivos y 20 variables, no pudiendo tratar el caso para 30 variables. Nosotros proporcionamos ejemplos de 5 objetivos y 75 variables, en el caso C.

Problema de la mochila 0-1 multiobjetivo ($p = 3$)							
p	n		$ \mathcal{T} $	$ F $	Algoritmo 5		AUGMECON2
					4ti2	t_r	
3	10	media	226	8	0,144 sg	0,074 sg	12208,086 sg
		máx.	336	26	0,19 sg	0,34 sg	24059,8 sg
	15	media	1866,5	24,3	1,144 sg	14,414 sg	
		máx.	3368	51	1,91 sg	65,86 sg	

Tabla 4.15

Problema de la mochila no acotado multiobjetivo ($p = 3$)							
Tipo	p	n		$ \mathcal{T} $	$ F $	Algoritmo 5	
						4ti2	t_r
A	3	25	media	83,9	290,1	0,11 sg	2,61 sg
			máx.	235	1342	0,13 sg	13,51 sg
		50	media	120,3	24003,5	0,105 sg	3493,509 sg
			máx.	193	216611	0,13 sg	32691,2 sg
		75	media	158,4	19912,9	0,202 sg	1444,512 sg
			máx.	191	148083	0,23 sg	13144,1 sg
		100	media	213,5	46382,4	0,285 sg	5040,221 sg
			máx.	272	432968	0,34 sg	47211,8 sg
C	3	25	media	59,36	124,45	0,16 sg	0,64 sg
			máx.	95	529	0,18 sg	5,64 sg
		50	media	105,1	1357,6	0,205 sg	15,495 sg
			máx.	123	10362	0,22 sg	136,91 sg
		75	media	182,7	2115,9	0,256 sg	110,45 sg
			máx.	394	8622	0,32 sg	713,04 sg
		100	media	205,3	1709	0,3 sg	30,894 sg
			máx.	219	9268	0,34 sg	109,4 sg

Tabla 4.16

Problema de la mochila no acotado multiobjetivo ($p = 4$)							
Tipo	p	n		$ \mathcal{T} $	$ F $	Algoritmo 5	
						4ti2	t_r
A	4	25	media	67,7	474,6	0,214 sg	3,94 sg
			máx.	122	2404	0,22 sg	21,7 sg
		50	media	119	4825,9	0,246 sg	316,9 sg
			máx.	171	36921	0,27 sg	2851,35 sg
		75*	media	153,5	1371	0,34 sg	16,91 sg
			máx.	163	5323	0,4 sg	93,5 sg
C	4	25	media	57,4	133,3	0,217 sg	0,257 sg
			máx.	77	379	0,22 sg	0,9 sg
		50	media	112,6	819,8	0,26 sg	19,36 sg
			máx.	138	3816	0,28 sg	175,31 sg
		75	media	158,3	835,4	0,321 sg	19,19 sg
			máx.	178	4069	0,34 sg	128,4 sg

* Obtenemos el 57,14 % de los ejemplos.

Tabla 4.17

Problema de la mochila no acotado multiobjetivo ($p = 5$)							
Tipo	p	n		$ \mathcal{T} $	$ F $	Algoritmo 5	
						4ti2	t_r
A	5	25	media	61	8429,3	0,151 sg	182,74 sg
			máx.	97	81032	0,18 sg	1820,92 sg
		50	media	116,6	12991,3	0,19 sg	1007,94 sg
			máx.	163	74240	0,236 sg	8668,66 sg
		75*	media	153	1938,83	0,75 sg	21,55 sg
			máx.	162	5352	0,8 sg	86,8 sg
C	5	25	media	65,7	585,9	0,313 sg	45,82 sg
			máx.	177	4724	0,41 sg	455,51 sg
		50	media	110	457,7	0,4 sg	3,05 sg
			máx.	180	2621	0,47 sg	10,7 sg
		75	media	170	8149,9	0,427 sg	243,24 sg
			máx.	217	49670	0,48 sg	1541,8 sg

* Obtenemos el 60 % de los ejemplos.

Tabla 4.18

Bibliografía

- [1] 4ti2 team, *4ti2 – a software package for algebraic, geometric and combinatorial problems on linear spaces*, <http://www.4ti2.de>.
- [2] K. Alanne, *Selection of renovation actions using multi-criteria “knapsack” model*, *Automation in Construction* **13** (2004), 377–391.
- [3] M.F Atiyah and I.G Macdonald, *Introduction to commutative algebra*, Addison-Wesley, 1969.
- [4] A. Barvinok, *A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed*, *Mathematics of Operations Research* **19** (1994), 769–779.
- [5] E. Bas, *An investment plan for preventing child injuries using risk priority number of failure mode and effects analysis methodology and a multi-objective, multi-dimensional mixed 0-1 knapsack model*, *Reliability Engineering & System Safety* **96** (2011), 748–756.
- [6] C. Bazgan, H. Hugot, and D. Vanderpooten, *Solving efficiently the 0-1 multi-objective knapsack problem*, *Computers & Operations Research* **36** (2009), 260–279.
- [7] D. Bertsimas, G. Perakis, and S. Tayur, *A new algebraic geometry algorithm for integer programming*, *Management Science* **46** (2000), 999–1008.
- [8] V. Blanco and J. Puerto, *Partial gröbner bases for multiobjective integer linear optimization*, *SIAM Journal on Discrete Mathematics* **23** (2009), 571–595.

- [9] ———, *Some algebraic methods for solving multiobjective polynomial integer programs*, *Journal of Symbolic Computation* **46** (2011), 511–533.
- [10] B. Buchberger, *An algorithm for finding the bases elements of the residue class ring modulo a zero dimensional polynomial ideal (German)*, Ph.D. thesis, University of Innsbruck, Austria, 1965.
- [11] J.F. Bérubé, M. Gendreau, and J.Y. Potvin, *An exact ϵ -constraint method for bi-objective combinatorial optimization problems: application to the traveling salesman problem with profits*, *European Journal of Operational Research* **194** (2009), 39–50.
- [12] G. Cantor, *Beiträge zur begründung der transfiniten mengenlehre*, *Mathematische Annalen* **49** (1897), 207–246.
- [13] D. Cao, A. Murat, and R.B. Chinnam, *Efficient exact optimization of multi-objective redundancy allocation problems in series-parallel systems*, *Reliability Engineering and System Safety* **111** (2013), 154–163.
- [14] A. Cayley, *A theorem on trees*, *Quarterly Journal of Mathematics* **23** (1889), 376–378.
- [15] V. Chankong and Y. Haimes, *Multiobjective decision making: theory and methodology*, North Holland, New York, 1983.
- [16] P. Conti and C. Traverso, *Buchberger algorithm and integer programming*, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: 9th International Symposium, AAECC-9 New Orleans, LA. Lecture Notes in Computer Science* (Harold F. Mattson, Teo Mora, and T. R. N. Rao, eds.), vol. 539, Springer, Berlin, 1991, pp. 130–139.
- [17] D. Cox, J. Little, and D. O’Shea, *Ideals, varieties and algorithms*, second ed., Springer, New York, 1997.
- [18] ———, *Using algebraic geometry*, second ed., Springer, New York, 2005.
- [19] J.A. De Loera, R. Hemmecke, and M. Köppe, *Pareto optima of multi-criteria integer linear programs*, *INFORMS Journal on Computing* **21** (2009), 39–48.
- [20] ———, *Algebraic and geometric ideas in the theory of discrete optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 2013.

- [21] X. Delorme, X. Gandibleux, and F. Degoutin, *Résolution approchée du problème de set packing bi-objectifs*, Proceedings de l'école d'Automne de Recherche Opérationnelle de Tours (EARO), 2003, pp. 74–80.
- [22] S. Droste, T. Jansen, and I. Wegener, *On the analysis of the (1+1) evolutionary algorithm*, Theoretical Computer Science **276** (2002), 51–81.
- [23] M. Ehrgott, *Multicriteria optimization*, second ed., Springer, Berlin, 2005.
- [24] M. Ehrgott, J. Figueira, and X. Gandibleux (eds.), *Multiobjective discrete and combinatorial optimization*, Annals of operations research, Springer, 2006.
- [25] M. Ehrgott and X. Gandibleux, *A survey and annotated bibliography of multiobjective combinatorial optimization*, OR Spektrum **22** (2000), 425–460.
- [26] N. El-Sherbeny, *Resolution of a vehicle routing problem with multi-objective simulated annealing method*, Ph.D. thesis, Faculté Polytechnique de Mons, Mons, Belgique, 2001.
- [27] J.E. Graver, *On the foundations of linear and integer linear programming I*, Mathematical Programming **9** (1975), 207–226.
- [28] S. Greco, M. Ehrgott, and J. Figueira, *Multiple criteria decision analysis: State of the art surveys*, International Series in Operations Research & Management Science, vol. 78, Springer, New York, 2005.
- [29] Y. Haimes, L.S. Lasdon, and D.A. Wismer, *On a bicriterion formulation of the problems of integrated system identification and system optimization*, IEEE Transactions on Systems, Man, and Cybernetics **1** (1971), no. 3, 296–297.
- [30] C.L. Hwang and M.J. Lin, *Group decision making under multiple criteria*, Lecture Notes in Economics and Mathematical Systems, vol. 281, Springer, 1987.
- [31] C.L. Hwang and A.S. Md. Masud, *Multiple objective decision making. methods and applications*, Lecture Notes in Economics and Mathematical Systems, vol. 164, Springer, 1979.

- [32] H. Ishibuchi and T. Murata, *A multi-objective genetic local search algorithm and its application to flowshop scheduling*, Trans. Sys. Man Cyber Part C **28** (1998), 392–403.
- [33] L. Jenkins, *A bicriteria knapsack program for planning remediation of contaminated lightstation sites*, European Journal of Operational Research **140** (2002), 427–433.
- [34] N. Jozefowicz, F. Semet, and E-G. Talbi, *A multi-objective evolutionary algorithm for the covering tour problem*, ch. 11, pp. 247–267, World Scientific, 2004, C. A. Coello and G. B. Lamont (ed.).
- [35] G. Kirlik and S. Sayın, *A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems*, European Journal of Operational Research **232** (2014), 479–488.
- [36] K. Klamroth and M. Wiecek, *Time dependent capital budgeting with multiple criteria*, vol. 487, pp. 421–432, 2000.
- [37] P. Korhonen, S. Salo, and R. Steuer, *A heuristic for estimating nadir criterion values in multiple objective linear programming*, Operations Research **45** (1997), no. 5, 751–757.
- [38] H.W. Kuhn and A.W. Tucker (eds.), *Linear inequalities and related systems*, Annals of Mathematics Studies, no. 38, Princeton University Press, Princeton, 1956.
- [39] W. Kwak, Y. Shi, H. Lee, and C.F. Lee, *Capital budgeting with multiple criteria and multiple decision makers*, Review of Quantitative Finance and Accounting **7** (1996), 97–112.
- [40] M. Laumanns, L. Thiele, and E. Zitzler, *An efficient, adaptative parameter variation scheme for metaheuristics based on the epsilon-constraint method*, European Journal of Operational Research **169** (2006), 932–942.
- [41] B. Lokman and M. Köksalan, *Finding all nondominated points of multi-objective integer programs*, Journal of Global Optimization **57** (2013), 347–365.
- [42] D.T. Luc, *Multiobjective linear programming: An introduction*, Springer, New York, 2016.
- [43] D.G. Luenberger, *Optimization by vector space methods*, John Wiley & Sons, New York, 1969.

- [44] M.R. Lyu, S. Rangarajan, and A.P.A. van Moorsel, *Optimal allocation of test resources for software reliability growth modeling in software development*, IEEE Transactions on Reliability **51** (2002), 183–192.
- [45] G. Mavrotas, *Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems*, Applied Mathematics and Computation **213** (2009), 455–465.
- [46] G. Mavrotas and K. Florios, *An improved version of the augmented ϵ -constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems*, Applied Mathematics and Computation **219** (2013), 9652–9669.
- [47] K. Miettinen, *Nonlinear multiobjective optimization*, Kluwer Academic Publishers, Boston, 1999.
- [48] R. Ouiddir, M. Rahli, R. Meziane, and A. Zeblah, *Ant colony optimization for new redesign problem of multi-state electrical power systems*, Journal of Electrical Engineering **55** (2004), 57–63.
- [49] M. Özlen and M. Azizoğlu, *Multi-objective integer programming: A general approach for generating all nondominated solutions*, European Journal of Operational Research **199** (2009), 25–35.
- [50] V. Pareto, *Manual of political economy. a critical and variorum edition*, Oxford University Press, 2014, ed. A. Montesano, A. Zanni, L. Bruni, J.S. Chipman and M. McLure.
- [51] A. Rong and J.R. Figueira, *A reduction dynamic programming algorithm for the bi-objective integer knapsack problem*, European Journal of Operational Research **231** (2013), 299–313.
- [52] ———, *Dynamic programming algorithms for the bi-objective integer knapsack problem*, European Journal of Operational Research **236** (2014), 85–99.
- [53] M. Rosenblatt and Z. Sinuany-Stern, *Generating the discrete efficient frontier to the capital budgeting problem*, Operations Research **37** (1989), no. 3, 384–394.
- [54] B. Sturmfels, *Gröbner bases and convex polytopes*, University Lecture Series, vol. 8, American Mathematical Society, Providence, RI, 1996.

- [55] J. Sylva and A. Crema, *A method for finding the set of non-dominated vectors for multiple objective integer linear programs*, European Journal of Operational Research **158** (2004), 46–55.
- [56] J. Teng and G. Tzeng, *A multiobjective programming approach for selecting non-independent transportation investment alternatives*, Transportation Research Part B **30** (1996), 291–307.
- [57] R.R. Thomas, *A geometric buchberger algorithm for integer programming*, Mathematics of Operations Research **20** (1995), no. 4, 864–884.
- [58] ———, *Gröbner bases in integer programming*, Handbook of Combinatorial Optimization (Ding-Zhu Du and Panos M. Pardalos, eds.), vol. 1, Kluwer Academic Publishers, 1998, pp. 533–572.
- [59] R.R. Thomas and R. Weismantel, *Truncated gröbner bases for integer programming*, Applicable Algebra in Engineering, Communication and Computing **8** (1997), 241–256.
- [60] W. Zhang and M. Reimann, *A simple augmented ϵ -constraint method for multi-objective mathematical integer programming problems*, European Journal of Operational Research **234** (2014), 15–24.
- [61] S. Zionts and J. Wallenius, *Identifying efficient vectors: Some theory and computational results*, Operations Research **28** (1980), 785–793.