

Noé Rodríguez Castro

*Frontera eficiente en Problemas de
Transporte Biobjetivo: un estudio
algorítmico y computacional*

Efficient frontier in Bi-objective Transport
Problems: an algorithmic and computational
study

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Mes de 2021

DIRIGIDO POR

Carlos González Martín

Antonio Alberto Sedeño Noda

Carlos González Martín
Estadística e Investigación
Operativa
Universidad de La Laguna
38200 La Laguna, Tenerife

Antonio Alberto Sedeño Noda
Estadística e Investigación
Operativa
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

A mis tutores, Carlos y Antonio, por su enorme comprensión y dedicación, así como por sus consejos y conocimientos para progresar y finalizar este trabajo.

A todas las personas que representan un pilar en mi vida, mis padres, mis hermanos y Noelia.

A Isidro, Olga, Nelson, Mari, Alba, y el resto de la familia que siempre han estado disponibles para brindarme su ayuda.

A todos mis amigos, que en todo momento han sido un apoyo, y a todos mis profesores, por ser referentes para mí.

Noé Rodríguez Castro
La Laguna, 10 de septiembre de 2021

Resumen · Abstract

Resumen

En este trabajo se realiza un estudio algorítmico y computacional del Problema de Transporte Biobjetivo, un caso particular de los problemas de flujo de coste mínimo sobre redes, englobados, a su vez, dentro de la Programación Lineal.

Antes de abordar el Problema de Transporte Biobjetivo, se introduce el Problema de Transporte Uniobjetivo o Clásico, utilizando una variante del Método del Simplex para su resolución. Luego se pasa al estudio del Problema de Transporte Biobjetivo, implementando una variante del algoritmo Simplex Paramétrico clásico para su resolución.

Finalmente, se realiza un estudio computacional donde, mediante los resultados obtenidos en diversas experiencias computacionales, observamos las ventajas que puede aportar el algoritmo propuesto frente al algoritmo clásico.

Palabras clave: *Problema de Transporte – Programación Lineal – Problema de Transporte Biobjetivo – Estudio Computacional – Problemas de Flujo de Coste Mínimo*

Abstract

In this work, an algorithmic and computational study of the Bi-objective Transport Problem, a particular case of the minimum cost flow problems over networks, included, in turn, within Linear Programming is carried out.

Before dealing with the Bi-objective Transport Problem, the Uniobjective or Classical Transport Problem is introduced, using a variant of the Simplex Method for its solution. Then, we move on to the study of the Bi-objective Transport Problem, implementing a variant of the classical Parametric Simplex algorithm for its resolution.

Finally, a computational study is carried out where, by means of the results obtained in different computational experiences, we observe the advantages that the proposed algorithm can provide against the classical algorithm.

Keywords: *Transport Problem – Linear Programming – Biobjective Transport Problem – Computational Study – Minimum Cost Flow Problems*

Contenido

Agradecimientos	III
Resumen/Abstract	V
Introducción	IX
1. Problema de Transporte Uniobjetivo	1
1.1. Introducción	1
1.1.1. Antecedentes históricos	1
1.1.2. Modelo del Problema de Transporte	3
1.1.3. Ejemplo	4
1.2. Resolución del problema de transporte	7
1.2.1. Métodos para la determinación de una solución básica factible inicial	7
1.2.2. Test de Optimalidad	9
2. Problema de Transporte Biobjetivo	15
2.1. Introducción	15
2.2. Problema de Transporte Biobjetivo	15
2.3. Problema paramétrico	16
2.4. Resolución del Problema de Transporte Biobjetivo	17
2.4.1. Determinación de una solución extrema eficiente soportada inicial	17
2.4.2. Búsqueda de soluciones extremas eficientes soportadas	21
2.4.3. Pseudocódigo del algoritmo	22
2.5. Ejemplo	24
3. Experiencias computacionales	33
3.1. Introducción	33
3.2. Resultados computacionales	33
3.2.1. Problemas con N=1000	34

3.2.2. Problemas con $M=1000$	37
3.2.3. Problemas con $N=M$	40
A. Apéndice	47
A.1. Problema equilibrado	47
Bibliografía	49
Poster	51

Introducción

El Problema de Transporte Clásico es un problema de Programación Lineal, planteado como uno de los primeros problemas dentro de la Investigación Operativa. Como principales autores que realizaron importantes avances en el estudio de este problema, cabe destacar a: A.N. Tolstoï, Leonid V. Kantoróvich, Tjalling C. Koopmans, Frank L. Hitchcock y G. B. Dantzig. Este problema trata de determinar una solución óptima de un grafo dirigido bipartito, con arcos que salen de los almacenes y llegan a los mercados, cumpliendo las restricciones de oferta y demanda respectivos. En este problema se considera un único objetivo, es decir, tratamos de minimizar (o maximizar) una función objetivo, por ejemplo, costo global.

El Problema de Transporte adquiere mayor interés si planteamos dos objetivos, es decir, queremos minimizar (o maximizar) dos funciones objetivo, por ejemplo, costo y tiempo global. Este problema se denomina Problema de Transporte Biobjetivo, y su estudio algorítmico y computacional es el propósito de este trabajo.

El trabajo se compone de tres capítulos, secuenciados de la siguiente forma:

El primer capítulo se inicia con una exposición de los antecedentes históricos del Problema de Transporte Clásico, para posteriormente introducir el modelo del problema. Una vez conocemos el modelo, imponemos la condición de consistencia (la cantidad total ofertada debe ser igual a la cantidad total demandada), y representamos el modelo en forma matricial. Con el problema equilibrado, pasamos a su resolución, para lo que hacemos uso del algoritmo del Simplex Primal. Comenzamos utilizando un método para determinar una solución básica factible inicial y, una vez obtenida esta solución básica inicial, comprobamos que sea óptima. Esto lo hacemos calculando unos potenciales del problema dual, y obteniendo los costos relativos del mismo. Cuando estos costos relativos sean no negativos, entonces obtendremos una solución óptima del problema. En caso contrario, se determina el único ciclo que induce la variable no básica con costo relativo más negativo, y se elimina este ciclo.

En el segundo capítulo, se introduce el modelo del Problema de Transporte Biobjetivo. Definimos *solución eficiente*, *punto débilmente eficiente*, *solución soportada*, *solución eficiente soportada*, y *solución eficiente extrema soportada*. Continuamos, con el problema paramétrico del Problema Transporte Biobjetivo, para posteriormente, conseguir una forma de determinar una solución extrema eficiente inicial, mostrada de forma práctica mediante un ejemplo. Una vez obtenida una solución extrema eficiente inicial, realizamos una búsqueda del resto de soluciones extremas eficientes soportadas, mediante la implementación de una variante un algoritmo clásico. Concluimos este capítulo con la aplicación del algoritmo sobre un ejemplo.

En el tercer capítulo, se analizan los resultados obtenidos en diversas experiencias computacionales, con el objetivo de comprobar la ventaja práctica que presenta el algoritmo propuesto frente al algoritmo clásico.

Además de los capítulos presentados anteriormente, el trabajo finaliza con la presentación de conclusiones, un apéndice, la bibliografía utilizada y un póster.

Problema de Transporte Uniobjetivo

1.1. Introducción

El problema de transporte se engloba dentro del campo de la Investigación Operativa, siendo uno de los problemas más antiguos estudiados en este campo, y siendo un problema de referencia en la Programación Lineal. Su formulación clásica es un caso particular de los problemas de flujo de coste mínimo sobre redes normalizadas dirigidas, donde se tienen vértices de oferta y vértices de demanda y únicamente existen arcos entre los primeros vértices y los segundos. Para su resolución, se aplica sobre el problema (verificando la condición de consistencia) un método para determinar una solución básica factible inicial. Y, concluimos comprobando la optimalidad de la solución básica factible, calculando, en su caso, una nueva solución básica factible.

1.1.1. Antecedentes históricos

El estudio del problema de transporte representa un hito en el desarrollo de la Programación Lineal (PL) en la década de 1940 [1]. Sin embargo, ya en 1930, A.N. Tolstoï [5], publica, en un libro sobre la planificación del transporte de la Unión Soviética, un artículo titulado *Methods of finding the minimal total kilometrage in cargo-transportation planning* donde estudió el problema de transporte y describió una serie de enfoques de la solución, incluida la idea de que una solución óptima no tiene ningún ciclo de costo negativo. Tolstoï pudo haber sido el primero en observar que esta condición es necesaria para la optimalidad. Este enfoque fue planteado en aplicaciones para el transporte de sal, cemento y otras cargas entre fuentes y destinos a lo largo de la red ferroviaria de la Unión Soviética, resolviendo de forma óptima el problema de transporte, para la red ferroviaria con 10 fuentes, 68 destinos y 155 enlaces entre fuentes y destinos. [10]

En 1938, Leonid V. Kantoróvich realizó un estudio sobre la distribución de materias primas para maximizar la productividad bajo ciertas condiciones. Matemáticamente, se trataba de un problema de maximizar una función lineal. Sin

embargo, el número de vértices de los polítopos era enorme, incluso en problemas muy simples. Este problema se encontraba en muchos problemas económicos diferentes con la misma forma matemática: distribución del trabajo en un equipo, mejor aprovechamiento del área de siembra, corte racional del material, uso de recursos complejos, distribución de flujos de transporte, etc. Un método eficaz para resolver este problema, lo encontró bajo el marco del análisis funcional, denominado "método de resolución de multiplicadores". En 1939, la prensa de la Universidad de Leningrado, publicó: *El método matemático de planificación y organización de la producción*. [9] En este artículo, Kantoróvich aporta las ideas principales de las teorías y algoritmos de programación lineal. En 1975, recibe el Premio Nobel de Economía, compartido con Tjalling C. Koopmans, por su contribución al desarrollo de métodos para el análisis de problemas económicos enfocados en la asignación óptima de recursos escasos. [8]

En 1941, Frank L. Hitchcock formula el problema de transporte lineal. Y lo describe como sigue:

Se tienen m fábricas o centros productores que se dedican a la fabricación de un mismo producto y n industrias transformadoras o centros transformadores de dicho producto. Como consecuencia de la diferente localización de los centros productores y consumidores, del coste de los portes y de otras causas, el coste de una unidad de producto varía de una ciudad a otra. Se trata de determinar cómo debe hacerse la distribución del producto con objeto de que se tenga el menor coste de transporte posible.

Además de formalizar el problema, aporta una solución constructiva, lo que supone un gran avance en la Investigación Operativa. [7]

En 1942, Tjalling C. Koopmans, en un memorando llamado *Relaciones de intercambio entre cargas en varias rutas*, presenta un plan para decidir la asignación óptima de buques de carga entre rutas. También mostró cómo los "potenciales" o los precios sombra podrían ayudar a tomar decisiones sobre la asignación de buques. [11]

En 1947, G. B. Dantzig marca el comienzo efectivo de la Programación Lineal, ya que introdujo e implementó el Método del Simplex. Esto supone uno de los hitos científicos del siglo XX. [3]

En 1949, Tjalling C. Koopmans organiza en Chicago un congreso sobre la programación lineal, cuyas actas fueron publicadas en 1951 bajo el título *Activity Analysis of Production and Allocation* [6] En este documento George B. Dantzig resuelve el problema de transporte mediante la aplicación del Método del Simplex. [4]

Desde esta época hasta la actualidad, los avances experimentados en el estudio y resolución de diferentes problemas de Investigación Operativa son de gran relevancia.

1.1.2. Modelo del Problema de Transporte

El Problema de Transporte consiste en buscar una solución óptima para el transporte de unas determinadas cantidades de producto desde m orígenes a n destinos teniendo en cuenta las ofertas de cada origen, las demandas de cada destino, y los costes de enviar cada unidad desde cada origen a cada destino. Por tanto, tratamos de establecer el número de unidades que se deben transportar desde cada uno de los orígenes a cada uno de los destinos, minimizando los costos globales y satisfaciendo las restricciones de oferta y demanda. El correspondiente modelo es:

$$\begin{aligned} \text{mín } c^t x &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} & (1.1) \\ \text{sujeto a: } & \sum_{i=1}^n x_{ij} = a_i, \forall i \in U \\ & \sum_{j=1}^m x_{ij} = b_j, \forall j \in V \\ & x_{ij} \geq 0, \forall (i, j) \in A \end{aligned}$$

En este caso, $U = \{1, \dots, m\}$ es el conjunto de vértices de oferta (orígenes), $V = \{1, \dots, n\}$ es el conjunto de vértices de demanda (destinos). Por tanto, el conjunto de vértices es $W = U \cup V$ y $A = U \times V$ es el conjunto de arcos. Representamos por a_i a la oferta de flujo (producto) en el vértice i , b_j es la demanda de flujo en el vértice j , y c_{ij} es el costo por unidad transportada desde el origen i al destino j .

De las restricciones del problema se tiene un poliedro que define el conjunto factible. Este problema genera un grafo bipartito, $G = (W, A) = (U \cup V, U \times V)$. Los valores de a_i y b_j , $\forall (i, j) \in A$ son valores enteros, además la matriz de restricciones es totalmente unimodular, por tanto, los puntos extremos del problema son coordenadas enteras.

Problema de Transporte equilibrado

Podemos observar que el modelo anterior (1.1), debe verificar que la cantidad total de oferta debe ser la misma que el total de la demanda, esto es:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

Esta condición se denomina de consistencia y garantiza que el problema tendrá siempre solución óptima. Por tanto, no se darán casos de no acotación ni de no factibilidad.

Si la condición de consistencia no se verifica, añadimos (según el caso) un origen o un destino ficticio, donde los costos de los arcos que, salen o llegan a este, respectivamente, son iguales a cero. (Véase Apéndice A.1)

Representación del modelo

Debido a las características del problema de transporte, podemos representar el modelo de forma matricial. En este formato, como se ve en la tabla 1.1, aparecen las ofertas, las demandas y los costes de transporte.

	D_1	D_2	\dots	D_n	Oferta
O_1	c_{11}	c_{12}	\dots	c_{1n}	a_1
O_2	c_{21}	c_{22}	\dots	c_{2n}	a_2
O_3	c_{31}	c_{32}	\dots	c_{3n}	a_3
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
O_m	c_{m1}	c_{m2}	\dots	c_{mn}	a_m
Demanda	b_1	b_2	\dots	b_n	

Tabla 1.1. Forma matricial del problema de transporte

Las filas corresponden a los vértices de oferta y las columnas a los vértices de demanda. En consecuencia, por cada fila y cada columna se lee una de las restricciones del modelo. La tabla contiene los costos de la función objetivo. En la última columna se tienen las ofertas de los almacenes, y en la última fila de la tabla se tienen las demandas de los mercados. Para completar la representación del modelo de transporte sobre la tabla anterior, sólo queda especificar que el problema es de mínimo. Análogamente, en los casos que se tenga que maximizar la función objetivo, bastaría con una precisión de este hecho.

1.1.3. Ejemplo

Una empresa dedicada a la producción de barras de pan dispone de tres almacenes, A_1 , A_2 y A_3 desde los cuales debe suministrar a cuatro panaderías P_1 , P_2 , P_3 y P_4 . En el siguiente grafo podemos visualizar las ofertas, las demandas y los costes de envío.

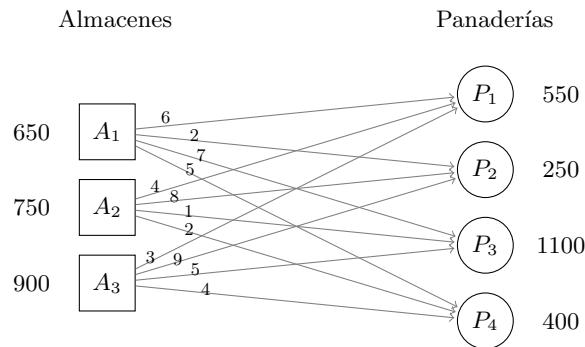


Figura 1.1. Ejemplo 1

Mediante esta información planteamos un modelo lineal que represente el problema, siendo x_{ij} la cantidad de pan enviada del almacén i a la panadería j , y entendiendo que se trata de un problema de mínimo. El modelo lineal es el siguiente:

$$\text{mín } c^t x = 6x_{11} + 2x_{12} + 7x_{13} + 5x_{14} + 4x_{21} + 8x_{22} + x_{23} + 2x_{24} + 3x_{31} + 9x_{32} + 5x_{33} + 4x_{34}$$

sujeto a:

$$\begin{array}{rccccr} x_{11} + x_{12} + x_{13} + x_{14} & & & & & = 650 \\ & x_{21} & + x_{22} & + x_{23} & + x_{24} & = 750 \\ & & & & x_{31} + x_{32} + x_{33} + x_{34} & = 900 \\ x_{11} & & + x_{21} & & x_{31} & = 550 \\ & x_{12} & & + x_{22} & & + x_{32} & = 250 \\ & & x_{13} & & + x_{23} & & + x_{33} & = 1100 \\ & & & x_{14} & & + x_{24} & & + x_{34} & = 400 \end{array}$$

$$x_{ij} \geq 0, i \in \{1, 2, 3\}, j \in \{1, 2, 3, 4\}$$

Observamos que en este caso las restricciones se ponen como igualdades porque se debe transportar, exactamente, todo que está en cada almacén y tiene que llegar a las panaderías, exactamente, lo que se demanda en cada una. Hay que tener en cuenta que esto se puede imponer sin que se dé la condición de consistencia.

Para observar la estructura de la matriz A escribimos el modelo como sigue:

$$\text{mín } z = (6, 2, 7, 5, 4, 8, 1, 2, 3, 9, 5, 4) \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{34} \end{pmatrix}$$

sujeto a:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{34} \end{pmatrix} = \begin{pmatrix} 650 \\ 750 \\ 900 \\ 550 \\ 250 \\ 1100 \\ 400 \end{pmatrix}$$

$$x_{ij} \geq 0, i = 1, 2, 3, j = 1, 2, 3, 4$$

En este ejemplo tenemos 3 orígenes, $m = 3$, y 4 destinos, $n = 4$. La matriz A tiene $3 + 4$ filas y 3×4 columnas. Esta matriz es de rango 6.

La correspondiente tabla de costes sería:

	P_1	P_2	P_3	P_4	Oferta
A_1	6	2	7	5	650
A_2	4	8	1	2	750
A_3	3	9	5	4	900
Demanda	550	250	1100	400	

Tabla 1.2. tabla de costes del ejemplo 1

1.2. Resolución del problema de transporte

Definiciones previas

Definición 1 (Soluciones básicas). *Las **soluciones básicas** se corresponden con árboles generadores, denominados árboles básicos.*

Definición 2 (Soluciones básicas factibles). *Una **solución básica factible** es un árbol básico que verifica las restricciones del problema.*

Definición 3 (árbol básico fuertemente factible). *Un árbol básico (con r fijado como nodo raíz) es **fuertemente factible** si en las únicas cadenas del árbol básico desde $r \in U$ hasta cada nodo, los arcos con $x_{ij} = 0$ son los arcos que están orientados hacia el nodo raíz.*

El óptimo se alcanza en un árbol básico factible.

Como el problema de transporte es un caso particular del Problema de Flujo de Coste Mínimo, la matriz A es totalmente unimodular, lo cual garantiza la obtención de soluciones enteras si al aplicar el Método del Simplex, las cantidades que se ofertan en los orígenes y las demandas de los destinos son números enteros. Además, como indicamos antes, las soluciones factibles básicas son árboles generadores de la correspondiente red, es decir, son estructuras que conectan todos los vértices de la red sin crear ciclos. Como se tienen $m + n$ vértices, hay $m + n - 1$ arcos básicos. Por tanto, el rango de la matriz de restricciones es igual a $m + n - 1$. En adelante, los árboles básicos serán árboles básicos fuertemente factibles.

Un esquema de búsqueda de una solución óptima para el Problema de Transporte podría ser el siguiente:

Algoritmo del Simplex Primal para el problema de transporte

Se inicia encontrando una solución básica factible inicial.

Mientras no se verifique la condición de optimalidad, esto es, existe una variable no básica con costo relativo negativo, se detecta el único ciclo que induce la nueva variable, y sobre este ciclo se determina la variable que deja de ser básica. De esta forma se tiene una nueva solución básica factible.

1.2.1. Métodos para la determinación de una solución básica factible inicial

Para seleccionar las $m + n - 1$ variables básicas que constituyen el árbol generador de la red, y asignarle valores que verifiquen las restricciones del problema, podemos hacer uso de varios métodos:

Método de la esquina noroeste. Se selecciona la esquina superior izquierda de la tabla actual, y en cada iteración, se le asigna el mayor valor posible, saturando así su correspondiente fila y/o columna. Por tanto, al finalizar este método, se obtiene $m + n - 1$ celdas, correspondientes a la solución básica factible inicial.

Método de costos mínimos. Este método es análogo al anterior, con la diferencia que en lugar de seleccionar la celda superior izquierda en cada iteración, seleccionamos en la tabla actual la celda que tenga el menor costo.

Método de Vogel. Para la selección de la variable a asignar se calculan en cada fila y cada columna, las diferencias (en valor absoluto) de los costos de las dos celdas de menor costo. Se selecciona la fila o columna con una penalización mayor y, dentro de ella, se toma la celda de menor costo. Como en los casos anteriores, se concluye con $m + n - 1$ celdas básicas.

Ejemplo del método de la esquina noroeste.

Vamos a encontrar una solución básica factible inicial para el ejemplo del problema de transporte mínimo 1.1

	P_1	P_2	P_3	P_4	Oferta
A_1	6	2	7	5	650
A_2	4	8	1	2	750
A_2	3	9	5	4	900
Demanda	550	250	1100	400	

Aplicando el método de la esquina noroeste, obtenemos la siguiente tabla:

	P_1	P_2	P_3	P_4	Oferta
A_1	6 550	2 100	7	5	650
A_2	4	8 150	1 600	2	750
A_3	3	9	5 500	4 400	900
Demanda	550	250	1100	400	

Esta solución inicial es factible básica. Además tiene $m+n-1 = 3+4-1 = 6$ variables básicas positivas. Por tanto, la solución del ejemplo mediante este método es:

$$\begin{aligned} x_{11} &= 550, x_{12} = 100, x_{13} = 0, x_{14} = 0, \\ x_{21} &= 0, x_{22} = 150, x_{23} = 600, x_{24} = 0, \\ x_{31} &= 0, x_{32} = 0, x_{33} = 500, x_{34} = 400 \end{aligned}$$

Ejemplo del método de costos mínimos.

La tabla final, al aplicar el método de costos mínimos sobre la tabla del ejemplo 1.1 es:

	P_1	P_2	P_3	P_4	Oferta
A_1	6	2 250	7 350	5 50	650
A_2	4	8	1 750	2	750
A_3	3 550	9	5	4 350	900
Demanda	550	250	1100	400	

Esta solución inicial es factible básica, y como el caso anterior tiene $m + n - 1 = 3 + 4 - 1 = 6$ variables básicas positivas. Por tanto, la solución mediante este método es:

$$\begin{aligned} x_{11} &= 0, x_{12} = 250, x_{13} = 350, x_{14} = 50, \\ x_{21} &= 0, x_{22} = 0, x_{23} = 750, x_{24} = 0, \\ x_{31} &= 550, x_{32} = 0, x_{33} = 0, x_{34} = 350 \end{aligned}$$

Ejemplo del método de Vogel.

La tabla final, al aplicar el método de Vogel sobre la tabla del ejemplo 1.1 es:

	P_1	P_2	P_3	P_4	Oferta
A_1	6	2 250	7	5 400	650
A_2	4	8	1 750	2	750
A_3	3 550	9	5 350	4	900
Demanda	550	250	1100	400	

Como en los ejemplos anteriores, tenemos $m + n - 1 = 3 + 4 - 1 = 6$ variables básicas. Concluimos que la solución es:

$$\begin{aligned} x_{11} &= 0, x_{12} = 250, x_{13} = 0, x_{14} = 400, \\ x_{21} &= 0, x_{22} = 0, x_{23} = 750, x_{24} = 0, \\ x_{31} &= 550, x_{32} = 0, x_{33} = 350, x_{34} = 0 \end{aligned}$$

Es posible que mediante los métodos anteriores, se obtengan una solución básica parcial con menos de $m + n - 1$ celdas con valores asignados mayores que cero. En estos casos, se dice que la solución básica es degenerada. Para obtener la solución básica, seleccionamos tantas celdas sean necesarias para alcanzar las $m + n - 1$ celdas, completando la estructura de árbol generador, y les asignamos a estas celdas el valor cero.

1.2.2. Test de Optimalidad

Una vez obtenida una solución básica factible inicial, podemos obtener los costos relativos a través de la resolución del correspondiente problema dual. Si denotamos por $u_1 \dots u_m$ y $v_1 \dots v_n$ las variables duales, el problema dual del problema de transporte es:

$$\begin{aligned} & \text{máx} \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \\ \text{sujeto a:} \quad & u_i + v_j \leq c_{ij}, \forall i \in \{1, \dots, m\} \forall j \in \{1, \dots, n\} \end{aligned}$$

Para calcular los costos relativos hacemos:

$$u_i + v_j = c_{ij}, \text{ para todas las celdas básicas.}$$

De esta forma determinamos un sistema de ecuaciones formado por $m+n-1$ ecuaciones con $m+n$ incógnitas. La resolución de este sistema nos permite encontrar los costos relativos de las variables no básicas. Una vez resuelto el sistema, los costos relativos se calculan mediante la siguiente expresión:

$$\bar{c}_{ij} = c_{ij} - (u_i + v_j)$$

Ejemplo

Continuando con el ejemplo 1.1. Habíamos calculado la siguiente solución básica factible por el método de la esquina noroeste:

	P_1	P_2	P_3	P_4	Oferta
A_1	6 550	2 100	7	5	650
A_2	4	8 150	1 600	2	750
A_3	3	9	5 500	4 400	900
Demanda	550	250	1100	400	

La correspondiente solución dual saldrá del siguiente sistema:

$$u_1 + v_1 = 6$$

$$u_1 + v_2 = 2$$

$$u_2 + v_2 = 8$$

$$u_2 + v_3 = 1$$

$$u_3 + v_3 = 5$$

$$u_3 + v_4 = 4$$

Una solución de este sistema viene dada por:

$$u_1 = 0, v_1 = 6, v_2 = 2, u_2 = 6, v_3 = -5, u_3 = 10, v_4 = -6$$

Esta solución se puede representar en la siguiente tabla:

	$v_1 = 6$	$v_2 = 2$	$v_3 = -5$	$v_4 = -6$	Oferta
$u_1 = 0$	6 550	2 100	7	5	650
$u_2 = 6$	4	8 150	1 600	2	750
$u_3 = 10$	3	9	5 500	4 400	900
Demanda	550	250	1100	400	

Calculamos sus costos relativos:

$$\begin{aligned}\bar{c}_{11} &= 6 - (0 + 6) = 0 \\ \bar{c}_{12} &= 2 - (0 + 2) = 0 \\ \bar{c}_{13} &= 7 - (0 - 5) = 12 \\ \bar{c}_{14} &= 5 - (0 - 6) = 11 \\ \bar{c}_{21} &= 4 - (6 + 6) = -8 \\ \bar{c}_{22} &= 8 - (6 + 2) = 0 \\ \bar{c}_{23} &= 1 - (6 - 5) = 0 \\ \bar{c}_{24} &= 2 - (6 - 6) = 2 \\ \bar{c}_{31} &= 3 - (10 + 6) = -13 \\ \bar{c}_{32} &= 9 - (10 + 2) = -3 \\ \bar{c}_{33} &= 5 - (10 - 5) = 0 \\ \bar{c}_{34} &= 4 - (10 - 6) = 0\end{aligned}$$

Obtenemos costos relativos negativos, que son los correspondientes a x_{21} , x_{31} y x_{32} . Por tanto, esta solución no verifica la condición de optimalidad. Si incrementamos el valor actual del arco no básico con costo relativo más negativo (x_{31}), al valor $\lambda \geq 0$, y compensamos los valores actuales de los arcos básicos tenemos:

	P_1	P_2	P_3	P_4	Oferta
A_1	6 $550 - \lambda$	2 $100 + \lambda$	7	5	650
A_2	4	8 $150 - \lambda$	1 $600 + \lambda$	2	750
A_3	3	λ 9	5 $500 - \lambda$	4 400	900
Demanda	550	250	1100	400	

Tomando el mayor valor posible de λ , obtenemos $\lambda = 150$. La nueva solución se escribe en la siguiente tabla:

	P_1	P_2	P_3	P_4	Oferta
A_1	6 400	2 250	7	5	650
A_2	4	8	1 750	2	750
A_3	3 150	9	5 350	4 400	900
Demanda	550	250	1100	400	

La correspondiente solución dual saldrá del siguiente sistema:

$$u_1 + v_1 = 6$$

$$u_1 + v_2 = 2$$

$$u_2 + v_3 = 1$$

$$u_3 + v_1 = 3$$

$$u_3 + v_3 = 5$$

$$u_3 + v_4 = 4$$

Una solución de este sistema viene dada por:

$$u_1 = 0, v_1 = 6, v_2 = 2, u_2 = -7, u_3 = -3, v_3 = 8, v_4 = 7$$

Esta solución se puede representar en la siguiente tabla:

	$v_1 = 6$	$v_2 = 2$	$v_3 = 8$	$v_4 = 7$	Oferta
$u_1 = 0$	6 400	2 250	7	5	650
$u_2 = -7$	4	8	1 750	2	750
$u_3 = -3$	3 150	9	5 350	4 400	900
Demanda	550	250	1100	400	

Calculamos sus costos relativos:

$$\bar{c}_{11} = 6 - (0 + 6) = 0$$

$$\bar{c}_{12} = 2 - (0 + 2) = 0$$

$$\bar{c}_{13} = 7 - (0 + 8) = -1$$

$$\bar{c}_{14} = 5 - (0 + 7) = -2$$

$$\bar{c}_{21} = 4 - (-7 + 6) = 5$$

$$\bar{c}_{22} = 8 - (-7 + 2) = 13$$

$$\bar{c}_{23} = 1 - (-7 + 8) = 0$$

$$\bar{c}_{24} = 2 - (-7 + 7) = 2$$

$$\bar{c}_{31} = 3 - (-3 + 6) = 0$$

$$\bar{c}_{32} = 9 - (-3 + 2) = 10$$

$$\bar{c}_{33} = 5 - (-3 + 8) = 0$$

$$\bar{c}_{34} = 4 - (-3 + 7) = 0$$

Obtenemos costos relativos negativos, que son los correspondientes a x_{13} y (x_{14}) . Por tanto, esta solución no verifica la condición de optimalidad. Si

incrementamos el valor actual del arco no básico con costo relativo más negativo x_{14} , al valor $\lambda \geq 0$, y compensamos los valores actuales de los arcos básicos tenemos:

	P_1	P_2	P_3	P_4	Oferta
A_1	6 $400 - \lambda$	2 250	7	5	λ 650
A_2	4	8	1 750	2	750
A_3	3 $150 + \lambda$	9	5 350	4 $400 - \lambda$	900
Demanda	550	250	1100	400	

Tomando el mayor valor posible de λ , obtenemos $\lambda = 400$. En este caso, hay dos árcos básicos candidatos a salir de la base, pero como se trata de un árbol básico fuertemente factible, el arco que se hace cero y permanece en la base es x_{34} . En consecuencia, la nueva solución se puede escribir como:

	P_1	P_2	P_3	P_4	Oferta
A_1	6	2 250	7	5 400	650
A_2	4	8	1 750	2	750
A_3	3 550	9	5 350	4 0	900
Demanda	550	250	1100	400	

La correspondiente solución dual saldrá del siguiente sistema:

$$\begin{aligned}
 u_1 + v_2 &= 2 \\
 u_1 + v_4 &= 5 \\
 u_2 + v_3 &= 1 \\
 u_3 + v_1 &= 3 \\
 u_3 + v_3 &= 5 \\
 u_3 + v_4 &= 4
 \end{aligned}$$

Una solución de este sistema viene dada por:

$$u_1 = 0, v_2 = 2, v_4 = 5, u_3 = -1, v_3 = 6, v_1 = 4, u_2 = -5$$

Esta solución se puede representar en la siguiente tabla:

	$v_1 = 4$	$v_2 = 2$	$v_3 = 6$	$v_4 = 5$	Oferta
$u_1 = 0$	6	2 250	7	5 400	650
$u_2 = -5$	4	8	1 750	2	750
$u_3 = -1$	3 550	9	5 350	4 0	900
Demanda	550	250	1100	400	

Calculamos sus costos relativos:

$$\begin{aligned}\bar{c}_{11} &= 6 - (0 + 4) = 2 \\ \bar{c}_{12} &= 2 - (0 + 2) = 0 \\ \bar{c}_{13} &= 7 - (0 + 6) = 1 \\ \bar{c}_{14} &= 5 - (0 + 5) = 0 \\ \bar{c}_{21} &= 4 - (-5 + 4) = 5 \\ \bar{c}_{22} &= 8 - (-5 + 2) = 11 \\ \bar{c}_{23} &= 1 - (-5 + 6) = 0 \\ \bar{c}_{24} &= 2 - (-5 + 5) = 2 \\ \bar{c}_{31} &= 3 - (-1 + 4) = 0 \\ \bar{c}_{32} &= 9 - (-1 + 2) = 8 \\ \bar{c}_{33} &= 5 - (-1 + 6) = 0 \\ \bar{c}_{34} &= 4 - (-1 + 5) = 0\end{aligned}$$

Como $\bar{c}_{ij} \geq 0, \forall (i, j) \in A$, concluimos que la solución básica factible anterior es óptima, esto es:

$$\begin{aligned}x_{11} &= 0, x_{12} = 250, x_{13} = 0, x_{14} = 400, \\ x_{21} &= 0, x_{22} = 0, x_{23} = 750, x_{24} = 0, \\ x_{31} &= 550, x_{32} = 0, x_{33} = 350, x_{34} = 0\end{aligned}$$

Problema de Transporte Biobjetivo

2.1. Introducción

En este capítulo se estudia el Problema de Transporte Biobjetivo, es decir, cuando queremos minimizar (o maximizar) dos funciones objetivo. Para ello, plantearémos el problema paramétrico y obtendremos una solución eficiente extrema soportada. Y, partiendo de esta solución, usando una variante del Método Simplex paramétrico clásico, buscaremos el resto de soluciones eficientes extremas soportadas.

2.2. Problema de Transporte Biobjetivo

Si $U = \{1, 2, \dots, m\}$ y $V = \{1, 2, \dots, n\}$, $G = (W, A) = (U \cup V, U \times V)$ define la red dirigida del problema de transporte. Un modelo del Problema de Transporte Biobjetivo (de mínimo) es:

$$\text{mín } c(x) = (c^1(x), c^2(x)) = \left(\sum_{(i,j) \in A} c_{ij}^1 x_{ij}, \sum_{(i,j) \in A} c_{ij}^2 x_{ij} \right) \quad (2.1)$$

$$\text{sujeto a:} \quad - \sum_{\{j | (i,j) \in A\}} x_{ij} = -a_i, \quad \forall i \in U \quad (2.2)$$

$$\sum_{\{i | (i,j) \in A\}} x_{ij} = b_j, \quad \forall j \in V \quad (2.3)$$

$$x_{ij} \geq 0, \quad \forall (i, j) \in A \quad (2.4)$$

siendo c_{ij}^β el coste por unidad en el arco (i,j) en el β -ésimo objetivo, $\beta \in \{1, 2\}$ y $x_{ij} \geq 0$ denota la variable de decisión asociada al arco (i,j) .

Por las propiedades de la matriz de coeficientes de las restricciones, sabemos que, si a_i y b_j son enteros, los puntos extremos de este problema tienen coordenadas enteras.

De las restricciones (2.2) , (2.3) y (2.4), se tiene el poliedro que define el *Espacio de Decisiones*. La imagen de X por c , $c(X)$, es el *Espacio Objetivo*.

A diferencia del caso uniobjetivo, en el caso biobjetivo, en general, no existe una solución en la que ambos objetivos se minimicen simultáneamente. Los niveles mínimos alcanzados por los objetivos definen el denominado *punto ideal*. Dicho punto, no pertenece, en general, a $c(X)$.

2.3. Problema paramétrico

Tomando $\mu \in (0, 1]$, tenemos:

$$\begin{aligned} \text{mín } & \mu \sum_{(i,j) \in A} c_{ij}^1 x_{ij} + (1 - \mu) \sum_{(i,j) \in A} c_{ij}^2 x_{ij} & (2.5) \\ \text{sujeto a: } & - \sum_{\{j | (i,j) \in A\}} x_{ij} = -a_i, \forall i \in U \\ & \sum_{\{i | (i,j) \in A\}} x_{ij} = b_j, \forall j \in V \\ & x_{ij} \geq 0, \forall (i, j) \in A \end{aligned}$$

Definiciones previas

Definición 4 (Solución eficiente). *Un punto $x \in X$, se define como un **solución eficiente** si $\nexists x' \in X$ tal que $c^1(x') \leq c^1(x)$ y $c^2(x') \leq c^2(x)$, siendo al menos una desigualdad menor estricta. Estas soluciones también son llamadas soluciones no dominadas u óptimas de Pareto.*

Definición 5 (Punto débilmente eficiente). *Un punto $x \in X$, se define como un **punto débilmente eficiente**, si $\nexists x' \in X$ tal que $c^1(x') < c^1(x)$ y $c^2(x') < c^2(x)$.*

Definición 6 (Solución soportada). *Sea $x \in X$ una solución, se define como una **solución soportada**, si x es solución de $\lambda_1 c^1(x) + \lambda_2 c^2(x)$, para algún $\lambda_1, \lambda_2 \geq 0$, en el correspondiente problema paramétrico.*

Definición 7 (Solución eficiente soportada). *Sea $x \in X$ una solución, se define como una **solución eficiente soportada**, si x es una solución óptima de $\lambda_1 c^1(x) + \lambda_2 c^2(x)$, para algunos $\lambda_1, \lambda_2 > 0$, en el correspondiente problema paramétrico.*

Definición 8 (Solución extrema eficiente soportada). Sea $x \in X$ una solución, se define como una ***solución extrema eficiente soportada***, si x es un punto extremo eficiente, que es solución de $\lambda_1 c^1(x) + \lambda_2 c^2(x)$, para algún $\lambda_1, \lambda_2 \geq 0$, en el correspondiente problema paramétrico.

Si $x \in X$ es una solución eficiente, $(c^1(x), c^2(x))$ es un *punto eficiente o punto no dominado* en $c(X)$. Los *puntos extremos no dominados soportados* están en frontera inferior del poliedro dado por $c(X)$.

En el problema paramétrico 2.5, para cada $\mu \in (0, 1]$, existe un punto extremo eficiente soportado $x \in X$ que es una solución óptima del problema paramétrico. Por tanto, se puede realizar una variación sistemática, decreciente o creciente, de μ en $(0, 1]$ para generar el conjunto de puntos extremos eficientes soportados.

Tomando $\theta = \frac{1-\mu}{\mu}$, al sacar como factor común μ en la función objetivo, el problema paramétrico lo podemos reescribir como sigue:

$$\begin{aligned} \text{mín} \quad & \sum_{(i,j) \in A} c_{ij}^1 x_{ij} + \theta \sum_{(i,j) \in A} c_{ij}^2 x_{ij} & (2.6) \\ \text{sujeto a:} \quad & - \sum_{\{j | (i,j) \in A\}} x_{ij} = -a_i, \forall i \in U \\ & \sum_{\{i | (i,j) \in A\}} x_{ij} = b_j, \forall j \in V \\ & x_{ij} \geq 0, \forall (i,j) \in A \end{aligned}$$

Nótese que cuando $\mu = 0$, se tiene $\theta = \infty$, y cuando $\mu = 1$, se tiene $\theta = 0$. Por tanto, $\theta \in [0, \infty[$

2.4. Resolución del Problema de Transporte Biobjetivo

2.4.1. Determinación de una solución extrema eficiente soportada inicial

Cuando trabajamos con el problema de Transporte Biobjetivo, nos interesa obtener una solución básica factible inicial, es decir, árbol básico factible. Primero resolveremos el problema asociado a la primera función objetivo, y para ello, basta tomar $\theta = 0$ quedando el siguiente problema:

$$\begin{aligned}
& \text{mín} \sum_{(i,j) \in A} c_{ij}^1 x_{ij} \\
\text{sujeto a:} \quad & - \sum_{\{j | (i,j) \in A\}} x_{ij} = -a_i, \forall i \in U \\
& \sum_{\{i | (i,j) \in A\}} x_{ij} = b_j, \forall j \in V \\
& x_{ij} \geq 0, \forall (i,j) \in A
\end{aligned}$$

cuyo problema dual asociado es:

$$\begin{aligned}
& \text{máx} - \sum_{i \in U} a_i u_i + \sum_{j \in V} b_j v_j \\
\text{sujeto a:} \quad & -u_i + v_j \leq c_{ij} \quad \forall (i,j) \in A
\end{aligned}$$

Una vez considerado el problema asociado a la primera función objetivo, procedemos de forma similar a lo expuesto en el primer capítulo, con las siguientes apreciaciones:

Una vez obtenido el árbol básico inicial, del cuál calculamos los correspondientes potenciales $(u_i, v_j), \forall i \in \{1, \dots, m\}, \forall j \in 1, \dots, n$, para cada arco básico (i, j) tendremos que: $c_{ij} = v_j - u_i$. Obteniendo el coste relativo de cada arco (i, j) mediante $\bar{c}_{ij} = c_{ij} + u_i - v_j$. Si tenemos una solución dual factible, con $\bar{c}_{ij} \geq 0, \forall (i, j) \in G$, tenemos que el árbol básico factible es una solución óptima del problema asociado a la primera función objetivo.

Para la obtención de una solución dual, exploramos el árbol básico desde el nodo raíz, $u_r = 0, r \in \{1, \dots, m\}$ y seguimos la cadena única desde dicho nodo hasta cada nodo del árbol básico factible, con arcos orientados desde el nodo raíz y arcos orientados hacia el nodo raíz (al tratarse de un grafo bipartito dirigido). De esta forma obtenemos una solución óptima para la primera función objetivo y los potenciales del problema dual. Nótese que esta solución debe ser eficiente, es decir, entre los óptimos (soluciones débilmente eficientes) tomamos la que sea mejor para la segunda función objetivo.

Realizamos el mismo procedimiento con la segunda función objetivo. Obteniendo de esta forma los costos relativos de cada función objetivo con sus respectivos potenciales iniciales.

Definimos como T_1^* al árbol básico fuertemente factible obtenido. Evidentemente, si tuviéramos $\bar{c}_{ij}^2 \geq 0, \forall (i, j) \notin T_1^*$, entonces T_1^* también es una solución óptima del problema asociado a la segunda función objetivo, $c^2(x)$. Y, por tanto, T_1^* sería única solución eficiente del problema de Transporte Biobjetivo (punto ideal).

Ejemplo

Consideremos las siguiente tabla de costes del problema asociado a la primera función objetivo:

7	4	6	5	8	9	7	43
5	8	3	9	6	7	5	26
8	6	7	5	8	10	8	31
9	2	3	4	2	3	1	60
23	17	37	6	43	15	19	

Aplicando el método de la esquina noroeste, obtenemos la siguiente solución factible inicial.

23	17	3	0	0	0	0	43
0	0	26	0	0	0	0	26
0	0	8	6	17	0	0	31
0	0	0	0	26	15	19	60
23	17	37	6	43	15	19	

Calculamos los potenciales del siguiente problema dual:

$$\begin{aligned} \max & - \sum_{i \in U} a_i u_i + \sum_{j \in V} b_j v_j \\ \text{sujeto a: } & -u_i + v_j \leq c_{ij} \quad \forall (i, j) \in A \end{aligned}$$

Obteniendo los siguientes potenciales y costos relativos, \bar{c}_{ij}^1 :

0	0	0	1	1	1	1	$u_1 = 0$
0	7	0	8	2	2	2	$u_2 = 3$
0	1	0	0	0	1	1	$u_3 = -1$
7	3	2	5	0	0	0	$u_4 = 5$
$v_1 = 7$	$v_2 = 4$	$v_3 = 6$	$v_4 = 4$	$v_5 = 7$	$v_6 = 8$	$v_7 = 6$	

Nótese que $\bar{c}_{ij}^1 \geq 0, \forall (i, j) \in G$, por tanto, la solución es óptima para el problema asociado a la primera función objetivo.

El árbol básico fuertemente factible es:

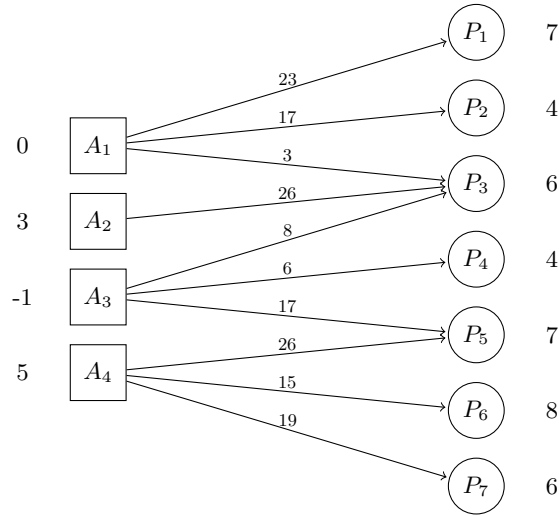


Figura 2.1. Problema Uniobjetivo: 1

A partir de esta solución para el primer objetivo, calculamos los siguientes potenciales y costos relativos para el segundo objetivo, \bar{c}_{ij}^2 :

0	0	0	-8	0	10	2	$u_1 = 0$
5	-7	0	-1	-3	2	0	$u_2 = -1$
5	-4	0	0	0	4	0	$u_3 = 0$
-6	-10	-4	-12	0	0	0	$u_4 = -5$
$v_1 = 2$	$v_2 = 9$	$v_3 = 6$	$v_4 = 9$	$v_5 = 4$	$v_6 = -2$	$v_7 = 3$	

El árbol básico fuertemente factible es:

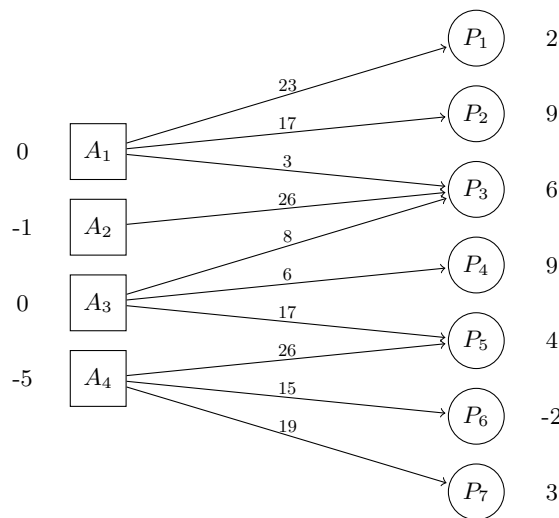


Figura 2.2. Problema Uniobjetivo: 2

En consecuencia, un árbol básico fuertemente factible viene dado por:

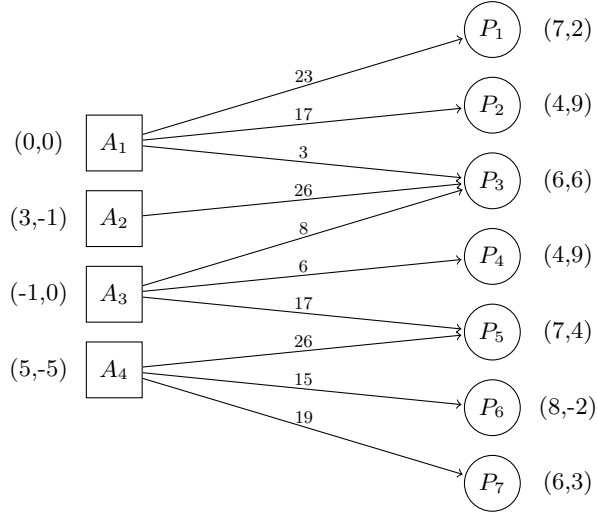


Figura 2.3. Grafo Biobjetivo

Lo que ocurre es que esta solución es óptima para la primera función objetivo, pero no lo es para la segunda función objetivo, ya que existen arcos no básicos tal que $\bar{c}_{ij}^2 < 0$. Por tanto, lo que tenemos es una solución extrema eficiente soportada inicial.

En la siguiente sección propondremos un algoritmo para encontrar, partiendo de esta solución, las demás soluciones extremas eficientes soportadas.

2.4.2. Búsqueda de soluciones extremas eficientes soportadas

Para la resolución del problema biobjetivo, debemos considerar el ratio θ_{ij} que viene definido en función de los costos relativos de cada objetivo, esto es:

$$\theta_{ij} = \begin{cases} -\frac{\bar{c}_{ij}^1}{\bar{c}_{ij}^2} & , \bar{c}_{ij}^2 < 0 \\ \infty & , \bar{c}_{ij}^2 \geq 0 \end{cases}$$

Valiéndonos de este ratio, identificaremos el arco no básico sobre el que se hace la operación de pivoteo. Establecemos el arco que entra a la base como

$$(x, y) = \underset{(u,v) \in A}{\operatorname{arglexmin}} \{ (\theta_{uv}, \bar{c}_{uv}^2) : \theta_{uv} < \infty \}$$

Además, considerando el conjunto de arcos no básicos entrantes en cada nodo i , $A_i = \{(j, i) : i \in V, (j, i) \notin T_k\}$, podemos vincular un ratio a cada nodo, de la siguiente forma:

$$(\theta_i, \bar{c}_i^2) = \underset{(u,v) \in A_i}{lexmin} \{(\theta_{uv}, \bar{c}_{uv}^2)\}$$

siendo θ_i el ratio mínimo de cualquier arco de A_i . Cuando existe más de un arco $(u, v) \in A_i$ con el mismo valor de θ_i , seleccionaremos el que tenga el mínimo valor \bar{c}_{uv}^2 .

En consecuencia, si $\bar{\theta}_k = \min_{i \in V} \theta_i = \theta_{xy}$ la siguiente solución eficiente soportada es obtenida mediante el pivoteo sobre (x, y) .

Una vez realizada la operación de pivoteo, es necesario actualizar los potenciales y los ratios de una forma eficiente. Para actualizar los potenciales, antes de introducir el arco entrante en la operación de pivoteo, eliminamos el arco que debe salir del árbol básico (p, q) , quedando dos subárboles T_k^1 y T_k^2 , con el nodo raíz r contenido en el conjunto de los nodos de T_k^1 .

Distinguimos los siguientes casos:

- 1) Si $x \in W(T_k^1)$, sumamos \bar{c}_{xy}^k , $k = 1, 2$ a los potenciales de $W(T_k^2)$
- 2) Si $x \in W(T_k^2)$, sumamos \bar{c}_{xy}^k , $k = 1, 2$ a los potenciales de $W(T_k^1)$

Una vez realizada la actualización de los potenciales, la relación que existe entre los ratios cada vez que se realiza una operación de pivoteo es la siguiente:

- 1) Si $x \in W(T_k^1)$
 - a) $i \in W(T_k^2), j \in W(T_k^1)$, entonces, $\theta_{ij}^{k+1} \leq \theta_{ij}^k$
 - b) $i \in W(T_k^1), j \in W(T_k^2)$, entonces, $\theta_{ij}^{k+1} \geq \theta_{ij}^k$
- 2) Si $x \in W(T_k^2)$
 - a) $i \in W(T_k^2), j \in W(T_k^1)$, entonces, $\theta_{ij}^{k+1} \geq \theta_{ij}^k$
 - b) $i \in W(T_k^1), j \in W(T_k^2)$, entonces, $\theta_{ij}^{k+1} \leq \theta_{ij}^k$

2.4.3. Pseudocódigo del algoritmo

El siguiente pseudocódigo, es una adaptación del algoritmo dado en [14] para la obtención de las soluciones extremas soportadas del problema de Flujo de Mínimo Coste Biobjetivo.

Algorithm 1 *Ratio – Labeling BTP*

```

1: Sea  $x, T, \pi = (\pi^1, \pi^2)$  la información de la solución óptima de  $\underset{x \in X}{lexmin}(c^1(x), c^2(x))$ ;
2: CreateHeap(H); lastratio=0;  $c^1 = c^1(x)$ ;  $c^2 = c^2(x)$ ;
3: Print el actual  $x$  con objetivos  $(c^1, c^2)$  como solución extrema soportada;
4: Establezcamos  $\Theta_i = \infty, j_i = 0, \forall i \in V$ ; ▷ Solo para los nodos destino
5: For all  $i \in V$  do ComputeLabels( $i, \Theta_i, \bar{c}_i^1, \bar{c}_i^2, j_i, H$ ); ▷ Solo para los nodos destino
6: While  $(H \neq \emptyset)$  do
7:    $\{\Theta_i, i, \bar{c}_i^2\} = Findmin(H)$ ;  $Deletemin(H)$ ;  $(x, y) = (i, j_i)$ ; ▷ Arco entrante
8:   Sea  $(p, q)$  el arco que sale y  $\delta$  la cantidad máxima de unidades que puede ser enviada
     a lo largo del ciclo  $A(T) \cup (x, y)$ ;
9:    $c^1 = c^1 + \bar{c}_i^1 \delta$ ;  $c^2 = c^2 + \bar{c}_i^2 \delta$ ;
10:   $A(T) = A(T) - \{(p, q)\} \cup \{(x, y)\}$ ; Actualizar  $T$  y los flujos ; ▷ Hace la operación de pivoteaje
     primal
11:  If  $(\Theta_i > lastratio \ \& \ \delta > 0)$  then
12:    Print el intervalo óptimo de la solución actual es  $[lastratio, \Theta_i]$ ;  $lastratio = \Theta_i$ ;
13:    Print la solución actual  $x$  con objetivos  $(c^1, c^2)$  como una solución extrema eficiente
     soportada;
14:     $S = \emptyset$ ;
15:    For all  $k \in U \cup V$  do  $inS_k = 0$ ;
16:    If  $(pred(y) == x)$  then  $IdentifyS(y, x, \bar{c}_i)$ ; ▷ Caso hacia adelante
17:    Else  $IdentifyS(root, x, \bar{c}_i)$ ; ▷ Caso hacia atrás
18:    For all  $k \in S$  do:
19:      If  $k \in V$  then  $ComputeLabels(k, \Theta_k, \bar{c}_k^1, \bar{c}_k^2, J_k, H)$ ;
20:      else
21:        For all  $j \in \Gamma_k^+ \cap (V - S)$  do
22:          If  $((\bar{c}_{kj}^2 < 0) \text{ and } (\bar{c}_{kj}^1 > 0))$  then
23:            If  $((-\frac{\bar{c}_{kj}^1}{\bar{c}_{kj}^2} < \Theta_j) \text{ or } ((-\frac{\bar{c}_{kj}^1}{\bar{c}_{kj}^2} == \Theta_j) \text{ and } (\bar{c}_{kj}^2 < c_{kj}^2)))$  then
24:              If  $(J_j == 0)$  then  $Insert(\{-\frac{\bar{c}_{kj}^1}{\bar{c}_{kj}^2}, \bar{c}_{kj}^2, j\}, H)$ ;
25:              else  $Decreasekey(\{-\frac{\bar{c}_{kj}^1}{\bar{c}_{kj}^2}, \bar{c}_{kj}^2, j\}, H)$ ;
26:               $\Theta_j = -\frac{\bar{c}_{kj}^1}{\bar{c}_{kj}^2}$ ;  $\bar{c}_j = \bar{c}_{kj}$ ;  $J_j = k$ ;
27:            Print el intervalo óptimo de la solución actual es  $[lastratio, \infty]$ ;

```

Algorithm 2 *ComputeLabels($i, \Theta_i, \bar{c}_i^1, \bar{c}_i^2, j_i, H$)*

```

1: If  $j_i \neq 0$  then ▷ Como el ratio del nodo  $i$  debe ser recalculado, debe eliminarse de  $H$ ;
2:    $Decreasekey\{-1, 0, i\}, H$ ;  $deletemin(H)$ ;
3:    $\Theta_i = +\infty$ ;  $j_i = 0$ 
4:   For all  $j \in \Gamma_i^-$  do
5:     If  $((\bar{c}_{ji}^2 < 0) \text{ and } (\bar{c}_{ji}^1 > 0))$  then
6:       If  $(\bar{c}_{ji}^1/\bar{c}_{ji}^2 < \Theta_i)$  or  $((\bar{c}_{ji}^1/\bar{c}_{ji}^2 == \Theta_i) \text{ and } (\bar{c}_{ji}^2 < \hat{c}_{ji}^2))$  then
7:          $\Theta_i = \bar{c}_{ji}^1/\bar{c}_{ji}^2$ ;  $\hat{c}_i^2 = \bar{c}_{ji}^2$ ;  $j_i = j$ ;
8:       If  $(j_i \neq 0)$  then  $Insert\{\theta_i, \hat{c}_i^2, i\}, H$ ;

```

Algorithm 3 *IdentifyS(y, x, \bar{c}_i)*

```

1:  $InS_k = 1$ ;  $\pi_k = \pi_k + change$ 
2: For all  $(k, j) \in A(T)$  do If  $(InS_j == 0)$  and  $(j \neq end)$  then  $IdentifyS(j, end, change)$ 
3: For all  $(j, k) \in A(T)$  do If  $(InS_j == 0)$  and  $(j \neq end)$  then  $IdentifyS(j, end, change)$ 

```

2.5. Ejemplo

Continuando con el anterior ejemplo: Calculamos la siguiente tabla de ratios:

-	-	-	1/8	∞	∞	∞
∞	1	-	8	2/3	∞	∞
∞	1/4	-	-	-	∞	∞
7/6	3/10	1/2	5/12	-	-	-
7/6	1/4	1/2	1/8	2/3	∞	∞

Tenemos que, $\theta_i = \theta_{xy} = 1/8$, por tanto, el arco que entra es el (1,4). El arco que sale es (1,3). Donde $\lambda = 3$. Diferenciaremos en azul el subárbol T_k^1 , y en rojo el subárbol T_k^2 , resultantes en cada iteración. Además el arco que entra aparece en negro y el arco que sale aparece con líneas discontinuas y flujo 0. En la primera etapa tenemos el siguiente árbol:

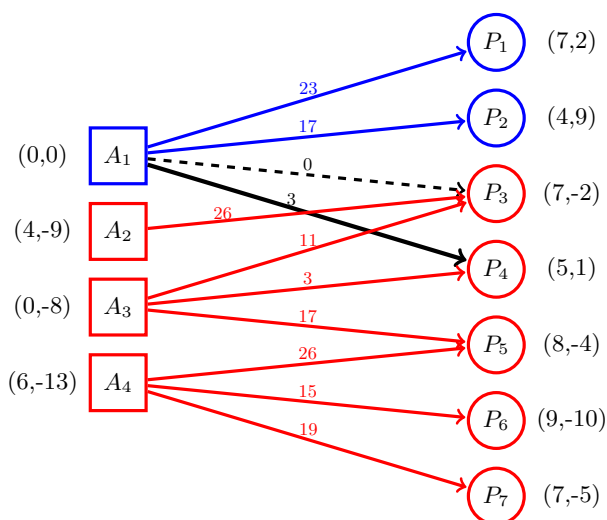


Figura 2.4. Iteración 1

Primera iteración. Como $x = 1 \in W(T_1^1)$ sumamos (1,-4) a los potenciales de $W(T_1^2)$, obteniendo sus correspondientes ratios:

-	-	∞	-	∞	∞	∞
2/3	8/15	-	8	2/3	∞	∞
1/3	1/6	-	-	-	∞	∞
4/7	2/9	1/2	5/12	-	-	-
1/3	1/6	1/2	5/12	2/3	∞	∞

Como $\theta_i = \theta_{xy} = 1/6$, por tanto, el arco que entra es el (3,2) y el arco que sale es (3,4), con $\lambda = 3$.

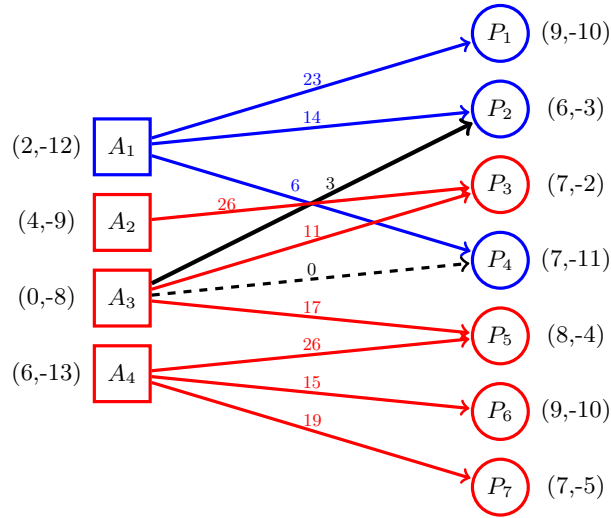


Figura 2.5. Iteración 2

Segunda iteración. Como $x = 3 \in W(T_2^2)$ sumamos (2,-12) a los potenciales de $W(T_2^1)$, obtenemos sus correspondientes ratios, representados en la siguiente tabla:

-	-	1/4	-	1/2	∞	1
∞	2	-	∞	2/3	∞	∞
∞	-	-	∞	-	∞	∞
3	1/3	1/2	∞	-	-	-
3	1/3	1/4	∞	1/2	∞	1

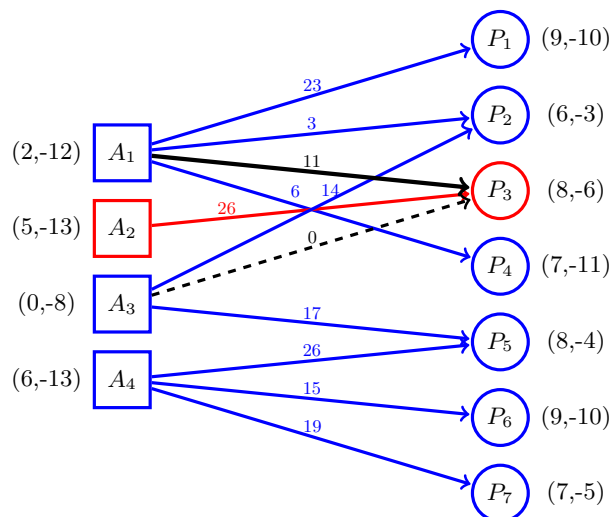


Figura 2.6. Iteración 3

Como $\theta_i = \theta_{xy} = 1/4$, por tanto, el arco que entra es el (1,3) y el arco que sale es (3,3), con $\lambda = 11$.

Tercera iteración. Como $x = 1 \in W(T_3^1)$ sumamos (1,-4) a los potenciales de $W(T_3^2)$, quedando siguiente tabla de ratios actualizada:

-	-	-	-	1/2	∞	1
∞	1	-	∞	3/7	3/2	3/4
∞	-	∞	∞	-	∞	∞
3	1/3	∞	∞	-	-	-
3	1/3	∞	∞	3/7	3/2	3/4

Tenemos que, $\theta_i = \theta_{xy} = 1/3$, por tanto, el arco que entra es (4,2) y el arco que sale es (3,2), con $\lambda = 14$.

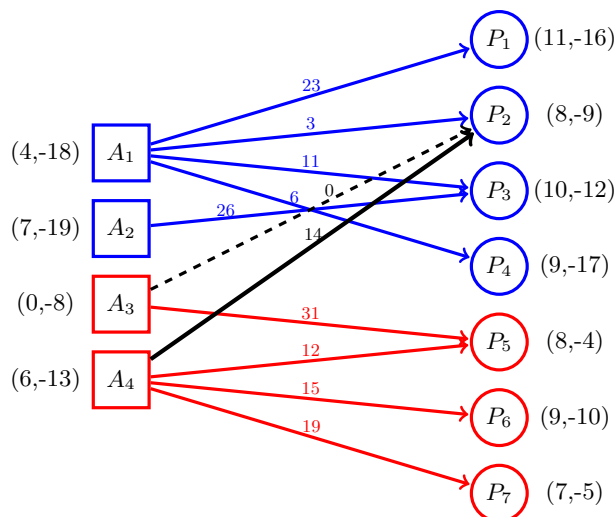


Figura 2.7. Iteración 4

Cuarta iteración. Como $x = 4 \in W(T_4^2)$ sumamos (2,-6) a los potenciales de $W(T_4^1)$, actualizamos sus correspondientes ratios, representados en la siguiente tabla:

-	-	-	-	2/5	∞	1/2
∞	1	-	∞	5/13	5/8	1/2
∞	∞	∞	∞	-	∞	∞
∞	-	∞	∞	-	-	-
∞	1	∞	∞	5/13	5/8	1/2

Como $\theta_i = \theta_{xy} = 5/13$, entra (2,5) y el arco que sale es (1,2), con $\lambda = 3$.

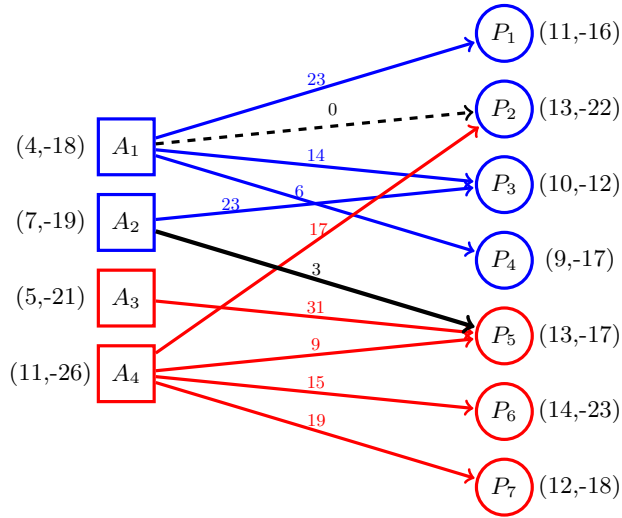


Figura 2.8. Iteración 5

Quinta iteración. Como $x = 2 \in W(T_5^1)$ sumamos (5,-13) a los potenciales de $W(T_5^2)$, actualizamos sus correspondientes ratios, representados en la siguiente tabla:

-	∞	-	-	∞	∞	∞
∞	∞	-	∞	-	∞	∞
∞	∞	2/3	∞	-	∞	∞
1	-	4/7	6/7	-	-	-
1	∞	4/7	6/7	∞	∞	∞

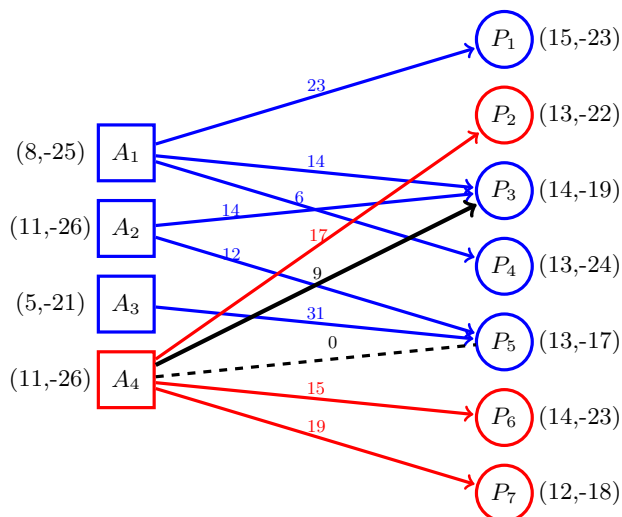


Figura 2.9. Iteración 6

Como $\theta_i = \theta_{xy} = 4/7$, por tanto, entra (4,3) y el arco que sale es (4,5), con $\lambda = 9$.

Sexta iteración. Como $x = 4 \in W(T_6^2)$ sumamos (4,-7) a los potenciales de $W(T_6^1)$, quedando la siguiente tabla de ratios actualizada:

-	∞	-	-	∞	∞	$3/2$
∞	6	-	∞	-	2	1
∞	2	$2/3$	∞	-	$7/6$	$5/7$
$5/2$	-	-	∞	∞	-	-
$5/2$	2	$2/3$	∞	∞	$7/6$	$5/7$

Tenemos que, $\theta_i = \theta_{xy} = 2/3$, por tanto, entra (3,3) y el arco que sale es (2,3), con $\lambda = 14$.

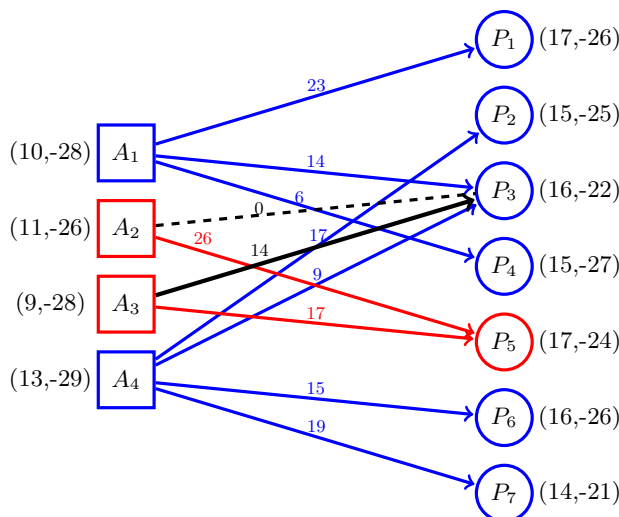


Figura 2.10. Iteración 7

Séptima iteración. Como $x = 3 \in W(T_7^2)$ sumamos (2,-3) a los potenciales de $W(T_7^1)$, actualizamos sus correspondientes ratios, representados en la siguiente tabla:

-	∞	-	-	∞	∞	$3/2$
∞	∞	∞	∞	-	∞	2
∞	∞	-	∞	-	∞	$3/4$
$5/2$	-	-	∞	∞	-	-
$5/2$	∞	∞	∞	∞	∞	$3/4$

Tenemos que, $\theta_i = \theta_{xy} = 3/4$, entra (3,7) y el arco que sale es (3,3), con $\lambda = 14$.

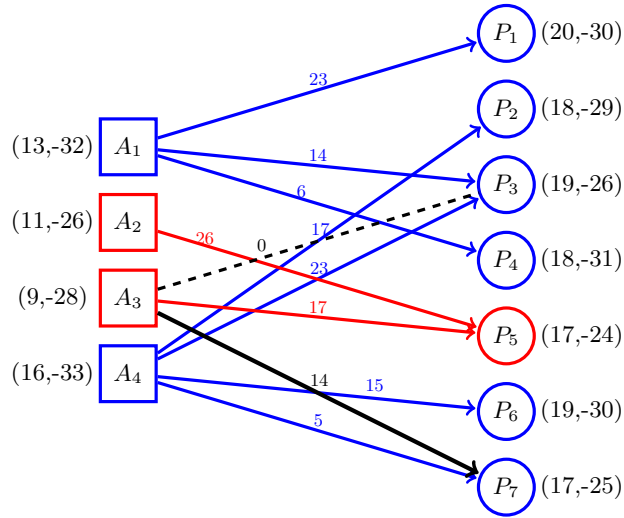


Figura 2.11. Iteración 8

Octava iteración. Como $x = 3 \in W(T_8^2)$ sumamos $(3,-4)$ a los potenciales de $W(T_8^1)$, actualizamos sus correspondientes ratios:

-	∞	-	-	1	∞	$3/2$
∞	∞	∞	∞	-	∞	∞
∞	∞	∞	∞	-	∞	-
$5/2$	-	-	∞	∞	-	-
$5/2$	∞	∞	∞	1	∞	$3/2$

Tenemos que, $\theta_i = \theta_{xy} = 1$, por tanto, entra $(1,5)$ y el arco que sale es $(4,7)$, con $\lambda = 5$.

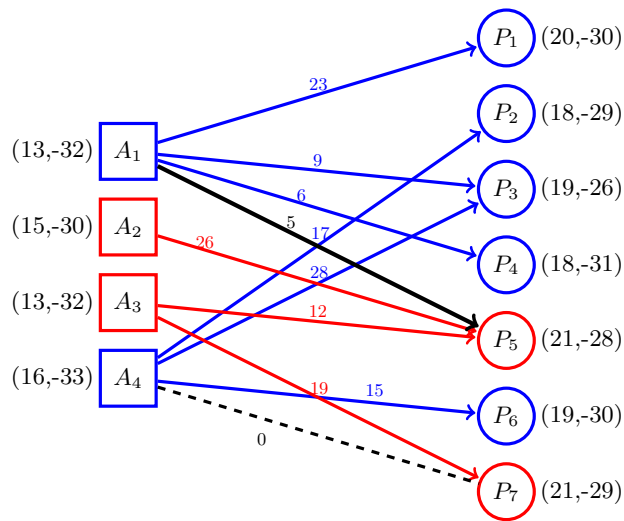


Figura 2.12. Iteración 9

Novena iteración. Como $x = 1 \in W(T_8^1)$ sumamos (4,-4) a los potenciales de $W(T_8^2)$, quedando la siguiente tabla de ratios actualizada:

-	∞	-	-	-	∞	∞
∞	∞	∞	∞	-	∞	∞
∞	∞	∞	∞	-	∞	-
$5/2$	-	-	∞	∞	-	∞
$5/2$	∞	∞	∞	∞	∞	∞

Tenemos que, $\theta_i = \theta_{xy} = 5/2$, por tanto, entra (4,1) y el arco que sale es (1,1), con $\lambda = 23$.

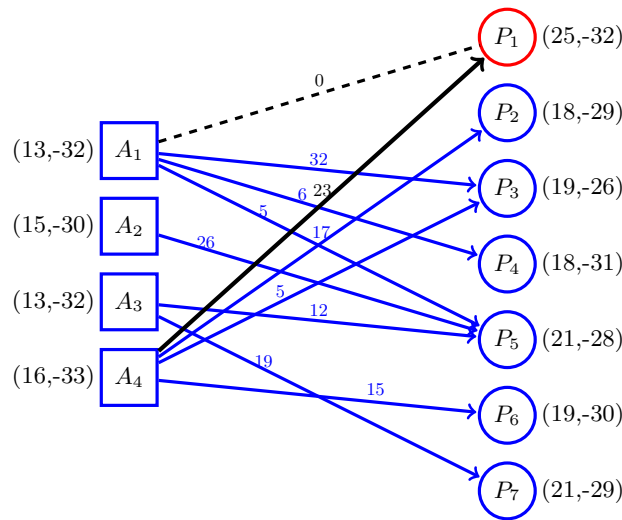


Figura 2.13. Iteración 10

Décima iteración. Como $x = 4 \in W(T_8^2)$ sumamos (5,-2) a los potenciales de $W(T_8^1)$, actualizamos sus correspondientes ratios:

∞	∞	-	-	-	∞	∞
∞	∞	∞	∞	-	∞	∞
∞	∞	∞	∞	-	∞	-
-	-	-	∞	∞	-	∞
∞	∞	∞	∞	∞	∞	∞

En este punto, $\bar{c}_{ij}^2 \geq 0$ para todo arco (i,j) no básico, por tanto, finalizamos con el siguiente árbol:

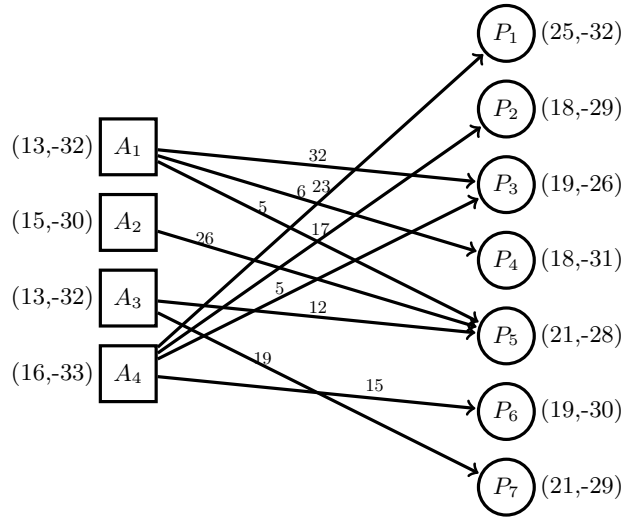


Figura 2.14. Grafo final

De las iteraciones realizadas extraemos los siguientes resultados:

Iteración	lastratio	Θ_i	$(c^1(x), c^2(x))$	Arco que entra	Arco que sale
0	0	1/8	(663,1000)	(1,4)	(1,3)
1	1/8	1/6	(666,976)	(3,2)	(3,4)
2	1/6	1/4	(672,940)	(1,3)	(3,3)
3	1/4	1/3	(683,896)	(4,2)	(3,2)
4	1/3	5/13	(711,812)	(2,5)	(1,2)
5	5/13	4/7	(726,773)	(4,3)	(4,5)
6	4/7	2/3	(762,710)	(3,3)	(2,3)
7	2/3	3/4	(790,668)	(3,7)	(3,3)
8	3/4	1	(832,612)	(1,5)	(4,7)
9	1	5/2	(852,592)	(4,1)	(1,1)
10	5/2	∞	(967,546)		

Por tanto, la gráfica que nos da la frontera inferior izquierda (al ser un problema de mínimo) del poliedro definido por $c(x)$ es de la siguiente forma:

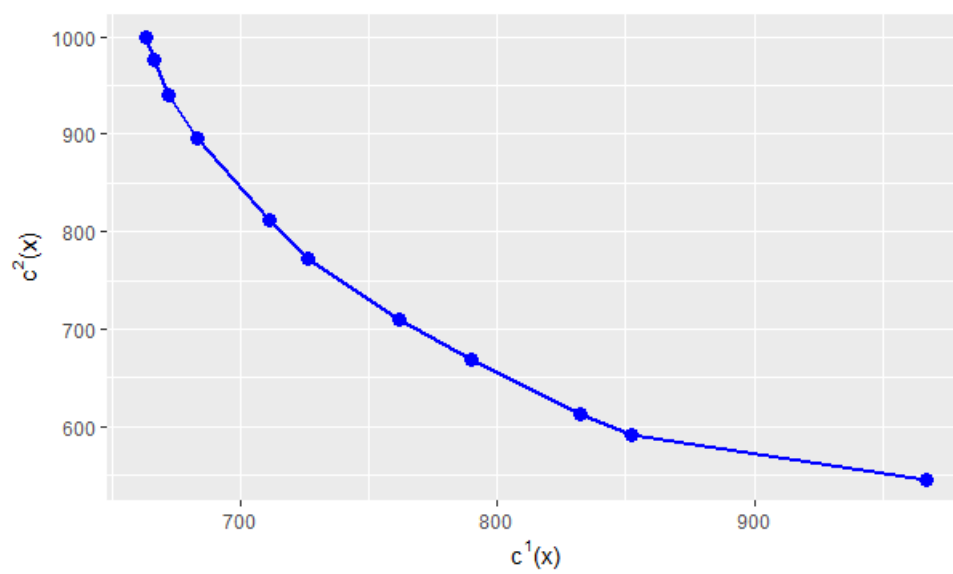


Figura 2.15. Fronteira eficiente

Experiencias computacionales

3.1. Introducción

En este capítulo se pretende contrastar la eficiencia práctica del algoritmo propuesto frente al algoritmo paramétrico clásico. Los ejemplos de entrada han sido generados usando el generador NETGEN [13], donde los costes de ambos objetivos son seleccionados aleatoriamente con distribución uniforme en el intervalo $[1, 10000]$. Se ha implementado el algoritmo paramétrico clásico (TRANSC) y el propuesto (TRANS) en lenguaje C y compilado con el compilador gcc (versión 4.8.4) y la opción de compilación `-O3`. Para la realización de las pruebas computacionales se utilizó un ordenador con Ubuntu 14.04 con Intel(R) Core(TM) i7-4610 CPU @ 3.00 GHz, y 16GB RAM.

El hecho de que en el problema de Transporte Biobjetivo Clásico subyace un grafo bipartito completo, permitirá observar que el algoritmo propuesto en el capítulo anterior, presenta mejores resultados en las pruebas realizadas.

3.2. Resultados computacionales

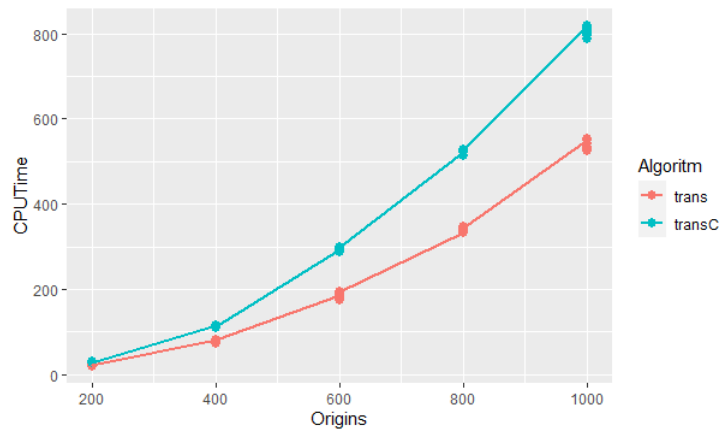
Se estudian tres tipos de problemas, en el primer tipo el número de orígenes es fijado, esto es, $M = 1000$ y el número de destinos varía, $N \in \{200, 400, \dots, 1000\}$. En el segundo tipo fijamos el número de destinos, esto es, $N = 1000$ y el número de orígenes varía, $M \in \{200, 400, \dots, 1000\}$. Y finalmente, en el último tipo el número de orígenes y destinos es igual, y ambos varían, esto es, $M = N = 200, M = N = 400, \dots, M = N = 1000$. Además, para cada combinación de parámetros se han creado 10 réplicas donde varía la semilla (costos de los objetivos). De esta forma, en las pruebas se han generado 150 casos particulares para cada algoritmo, resultando un total de 300 observaciones.

3.2.1. Problemas con N=1000

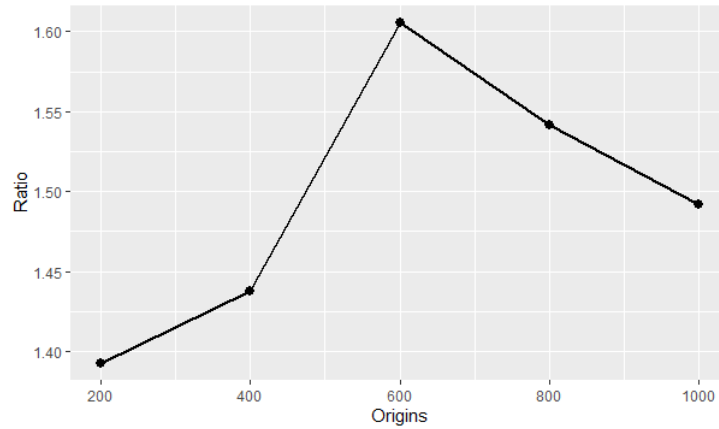
En esta sección se representan los resultados obtenidos en las pruebas realizadas fijando el número de destinos en 1000. En la siguiente tabla se muestran los principales promedios:

Algoritmo	M	N	Arcos	CPUTime medio	Media de Sol. Ext. Sop.	Media de Sol.Sop.*Arcos
trans	200	1000	200000	20,2378736	14847,9	4,62353E+14
transC	200	1000	200000	28,1814325	14847,9	5,99903E+14
trans	400	1000	400000	77,9024709	20738,6	1,02064E+15
transC	400	1000	400000	111,9644765	20738,6	8,70992E+14
trans	600	1000	600000	182,0450584	26811,3	1,05346E+15
transC	600	1000	600000	292,3064177	26811,3	2,09107E+15
trans	800	1000	800000	338,6426136	32457,4	1,71432E+15
transC	800	1000	800000	521,9985392	32457,4	6,17641E+14
trans	1000	1000	1000000	538,95165	37970	2,16878E+15
transC	1000	1000	1000000	804,2010306	37970	1,21854E+15

En el siguiente gráfico podemos visualizar el tiempo que tarda cada algoritmo en función del número de orígenes, donde se observa la ventaja en tiempo que presenta el algoritmo propuesto:



Para visualizar cuántas veces es más rápido un algoritmo que otro, se realiza el ratio de los tiempos del algoritmo clásico entre los tiempos del algoritmo propuesto:



Se observa cómo el algoritmo propuesto es del rango de 1.39 a 1.60 veces más rápido que el clásico.

Para estudiar la complejidad práctica que presenta el algoritmo clásico frente al algoritmo propuesto, se propone estudiar el comportamiento de la variable $CPU\ Time$, tomando las siguientes variables independientes: X = número de soluciones eficientes extremas soportadas (Ext.Sol.Sop), Y = número de arcos (Arcs). El modelo que se propone es:

$$CPU\ Time = C_1 + C_2 * X * Y^\alpha \quad (3.1)$$

En C_1 queremos estimar el tiempo que tardó en obtener la primera solución, y en $C_2 * X * Y^\alpha$ queremos estimar el tiempo empleado por el algoritmo propuesto. Desde que el algoritmo realiza tantas iteraciones como el número de soluciones soportadas, la variable X debe aparecer elevada a 1 en el modelo. Mientras que en cada iteración del algoritmo se examina una fracción del número de arcos (pudiendo ser todos en el peor de los casos). Para esto en el modelo aparece Y^α , donde α deberá ser menor o igual que 1 (y mayor que cero).

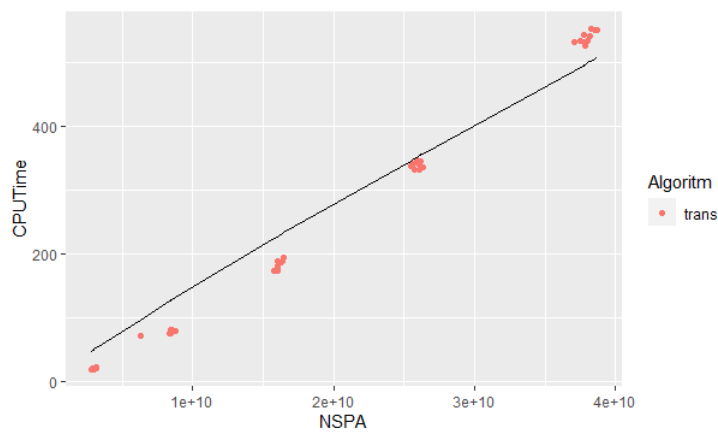
Se implementa este modelo no lineal en R , usando $R\ Studio$, mediante la función $nls()$. Tomando como rangos iniciales $(0,100]$ para C_1 , $0,0000001$ para C_2 y $(0,1]$ para α . Los coeficientes estimados y coeficiente de determinación obtenidos aplicando el modelos sobre los datos del algoritmo propuesto son:

Coefficient	Estimate
c_1	$1.000e - 02$
c_2	$1.286e - 07$
α	$8.364e - 01$
R^2	0.9869792

Se puede observar como el coeficiente de determinación es muy próximo a 1, por tanto, el modelo explica bastante bien la variabilidad en la variable

CPUTime, en el algoritmo propuesto.

En la siguiente gráfica se observa el comportamiento del modelo propuesto sobre el algoritmo propuesto, donde NSPA es el número de soluciones extremas soportadas * número de arcos (orígenes*destinos) y CPUTime el tiempo computacional:

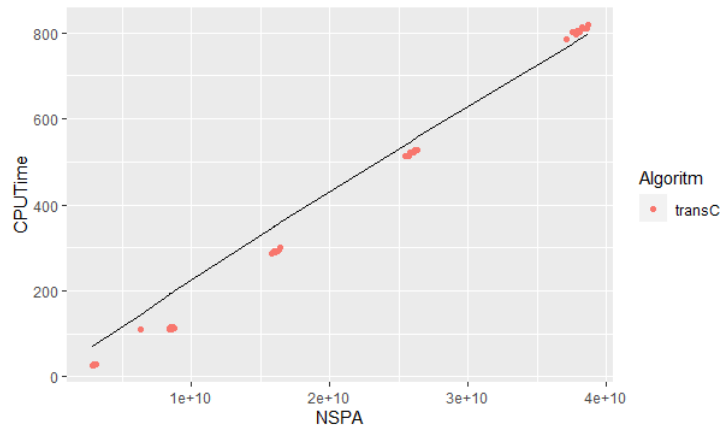


Los coeficientes estimados y coeficiente de determinación obtenidos aplicando el modelo sobre los datos del algoritmo clásico son:

Coefficient	Estimate
c_1	$1.000e - 04$
c_2	$1.106e - 07$
α	$8.783e - 01$
R^2	0.9780218

Se puede observar como el coeficiente de determinación es muy próximo a 1, por tanto, el modelo explica bastante bien la variabilidad en CPUTime, en el algoritmo clásico.

En la siguiente gráfica se observa el comportamiento del modelo propuesto sobre el algoritmo clásico, donde NSPA es el número de soluciones extremas soportadas * número de arcos (orígenes*destinos) y CPUTime el tiempo computacional:



Concluimos que la complejidad del algoritmo se puede estimar con el modelo propuesto. Además se observa una mejora del algoritmo propuesto respecto al algoritmo clásico.

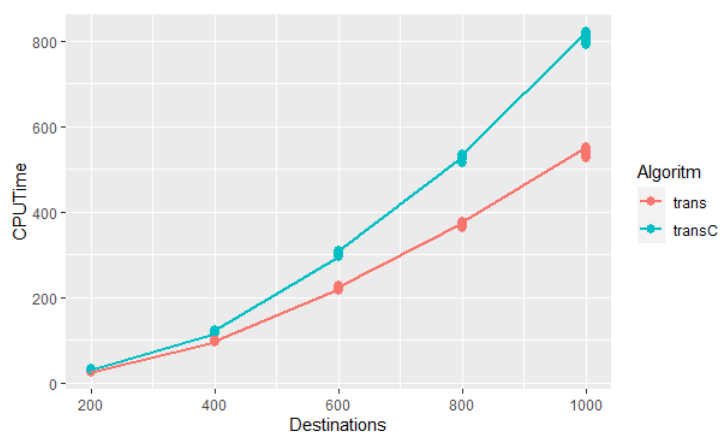
3.2.2. Problemas con $M=1000$

Ahora trataremos los resultados obtenidos en las pruebas realizadas fijando el número de orígenes ($M=1000$).

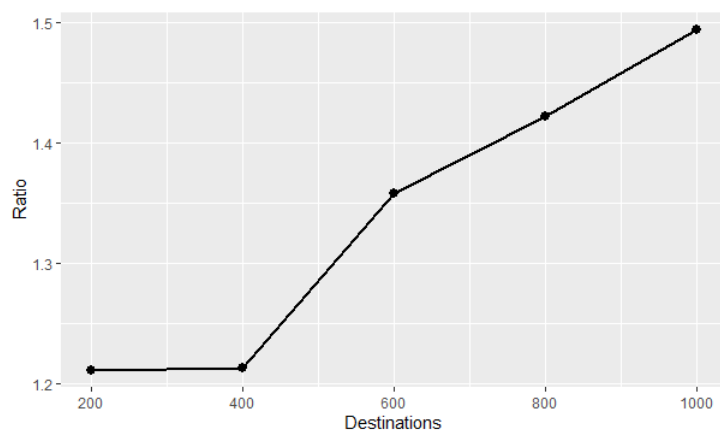
En la siguiente tabla se introducen los principales promedios:

Algoritmo	M	N	Arcos	CPUTime medio	Media de Sol. Ext. Sop.	Media de Sol.Sop.*Arcos
trans	1000	200	200000	26,2119361	15543,4	6,40207E+14
transC	1000	200	200000	31,7521229	15543,4	6,27851E+14
trans	1000	400	400000	98,439106	21989,9	1,78186E+14
transC	1000	400	400000	119,460601	21989,9	1,32467E+14
trans	1000	600	600000	221,5516992	27768,2	1,64374E+15
transC	1000	600	600000	301,0227352	27768,2	1,42605E+15
trans	1000	800	800000	371,3004475	32966,9	4,48425E+14
transC	1000	800	800000	527,9743422	32966,9	9,57529E+14
trans	1000	1000	1000000	539,1522359	37970	2,16878E+15
transC	1000	1000	1000000	805,758686	37970	1,21854E+15

En el siguiente gráfico podemos visualizar el tiempo que tarda cada algoritmo en función del número de orígenes, donde se observa la ventaja en tiempo que presenta el algoritmo propuesto:



Para visualizar cuántas veces es más rápido un algoritmo que otro, se realiza el ratio de los tiempos del algoritmo clásico entre los tiempos del algoritmo propuesto:



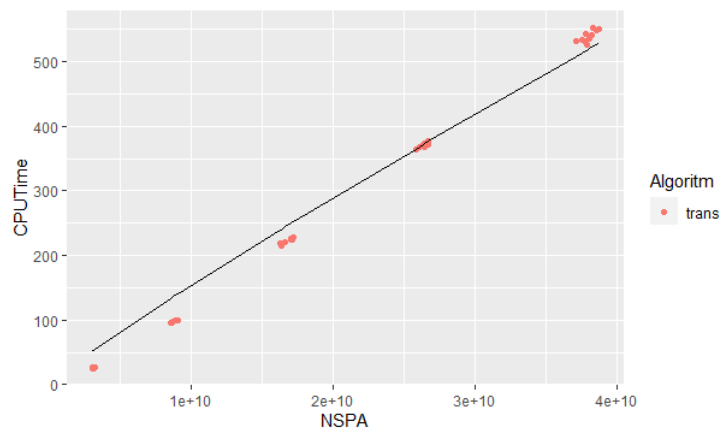
Se observa cómo el algoritmo propuesto es del rango de 1,21 a 1,49 veces más rápido que el clásico.

Los coeficientes estimados y coeficiente de determinación obtenidos aplicando el modelo sobre los datos del algoritmo propuesto son:

Coefficient	Estimate
c_1	$1.000e - 02$
c_2	$1.286e - 07$
α	$8.523e - 01$
R^2	0.9944208

Se puede observar como el coeficiente de determinación es muy próximo a 1, por tanto, el modelo explica bastante bien la variabilidad en la variable CPUTime, en el algoritmo propuesto.

En la siguiente gráfica se observa el comportamiento del modelo propuesto sobre el algoritmo propuesto, donde NSPA es el número de soluciones extremas soportadas * número de arcos (orígenes*destinos) y CPUTime el tiempo computacional:

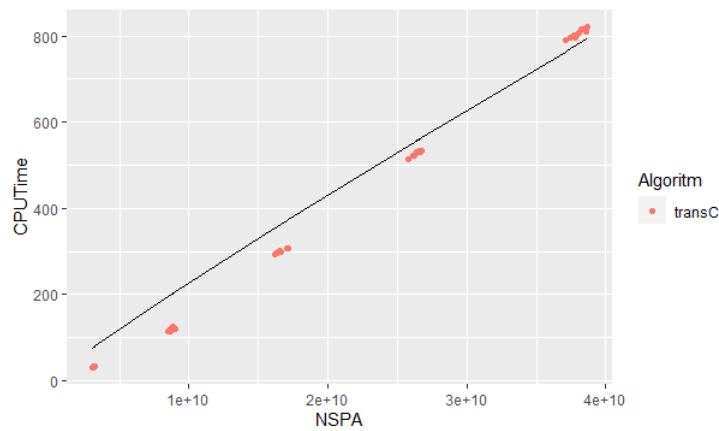


Los coeficientes estimados y coeficiente de determinación obtenidos aplicando el modelo sobre los datos del algoritmo clásico son:

Coefficient	Estimate
c_1	$1.000e - 04$
c_2	$1.020e - 07$
α	$8.823e - 01$
R^2	0.9783016

Se puede observar como el coeficiente de determinación es muy próximo a 1, por tanto, el modelo explica bastante bien la variabilidad en la variable CPUTime, en el algoritmo clásico.

En la siguiente gráfica se observa el comportamiento del modelo propuesto sobre el algoritmo clásico, donde NSPA es el número de soluciones extremas soportadas * número de arcos (orígenes*destinos) y CPUTime el tiempo computacional:



Concluimos que la complejidad del algoritmo se puede estimar con el modelo propuesto. Además se observa una mejora del algoritmo propuesto respecto al algoritmo clásico.

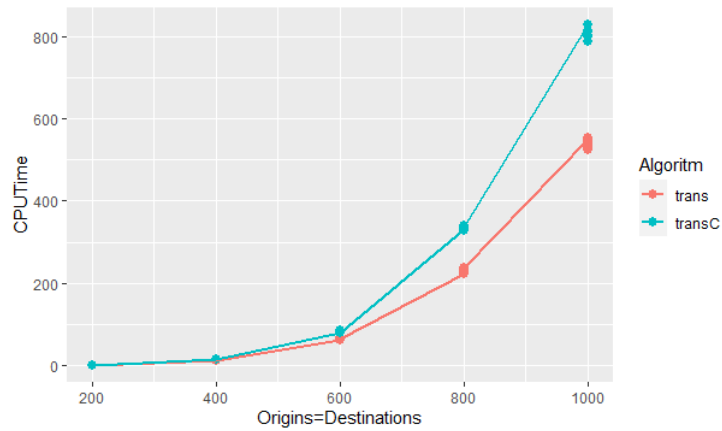
3.2.3. Problemas con $N=M$

Por último trataremos los resultados obtenidos en las pruebas realizadas igualando el número de orígenes y destinos, variando en $\{200, 400, \dots, 1000\}$.

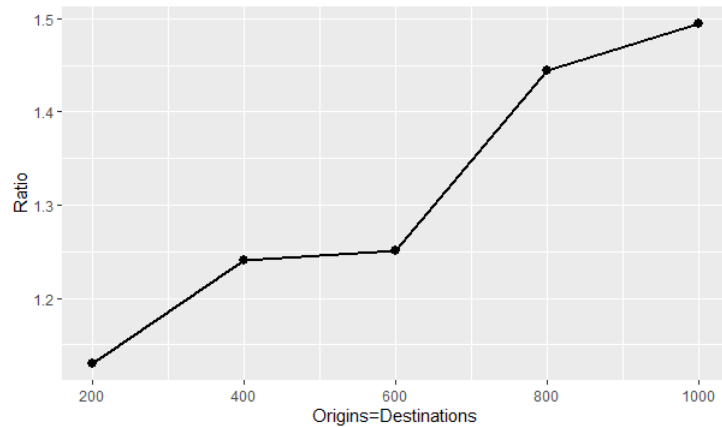
Los principales promedios de los resultados obtenidos en las pruebas realizadas, se muestran en la siguiente tabla:

Algoritmo	$N=M$	Arcos	CPUTime medio	Media de Sol. Ext. Sop.	Media de Sol.Sop.*Arcos
trans	200	40000	1,1474515	3784,2	5,85159E+12
transC	200	40000	1,2967974	3784,2	6,07094E+12
trans	400	160000	12,6243418	10302,7	2,52666E+14
transC	400	160000	15,665899	10302,7	2,67431E+14
trans	600	360000	64,7014775	18526,3	7,51918E+14
transC	600	360000	80,9636599	18526,3	8,83838E+14
trans	800	640000	231,5776174	28044,3	8,37391E+14
transC	800	640000	334,4205045	28044,3	7,00784E+14
trans	1000	1000000	539,46469	37970	2,16878E+15
transC	1000	1000000	806,3772191	37970	1,21854E+15

En el siguiente gráfico podemos visualizar el tiempo que tarda cada algoritmo en función del número de orígenes, donde se observa la ventaja en tiempo que presenta el algoritmo propuesto:



Para visualizar cuántas veces es más rápido un algoritmo que otro, se realiza el ratio de los tiempos del algoritmo clásico entre los tiempos del algoritmo propuesto:



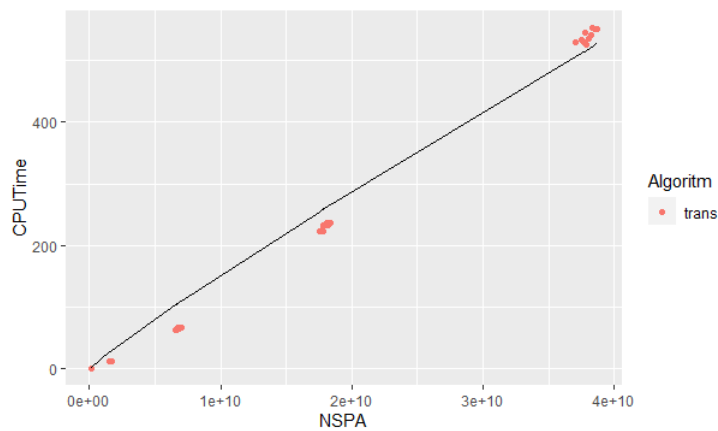
Se observa cómo el algoritmo propuesto es del rango de 1,13 a 1,49 veces más rápido que el clásico.

Los coeficientes estimados y coeficiente de determinación obtenidos aplicando el modelo sobre los datos del algoritmo propuesto son:

Coefficient	Estimate
c_1	$1.000e - 02$
c_2	$1.004e - 07$
α	$8.553e - 01$
R^2	0.9951207

Se puede observar como el coeficiente de determinación es muy próximo a 1, por tanto, el modelo explica bastante bien la variabilidad en la variable CPUTime, en el algoritmo propuesto.

En la siguiente gráfica se observa el comportamiento del modelo propuesto sobre el algoritmo propuesto, donde NSPA es el número de soluciones extremas soportadas * número de arcos (orígenes*destinos) y CPUTime el tiempo computacional:

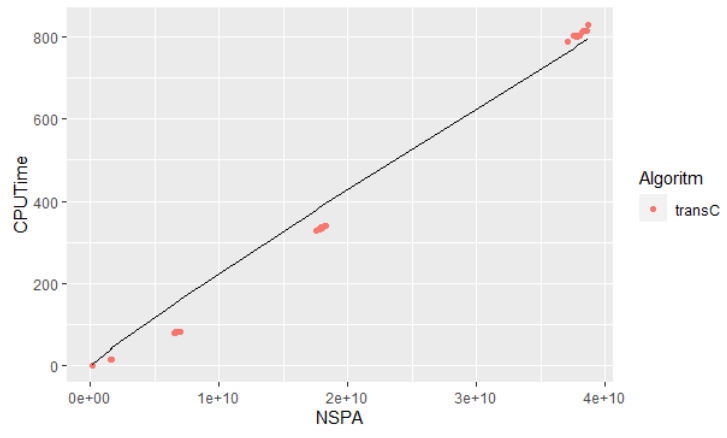


Los coeficientes estimados y coeficiente de determinación obtenidos aplicando el modelo sobre los datos del algoritmo clásico son:

Coefficient	Estimate
c_1	$1.000e - 04$
c_2	$1.020e - 07$
α	$8.850e - 01$
R^2	0.9860345

Se puede observar como el coeficiente de determinación es muy próximo a 1, por tanto, el modelo explica bastante bien la variabilidad en la variable CPUTime, en el algoritmo clásico.

En la siguiente gráfica se observa el comportamiento del modelo propuesto sobre el algoritmo clásico, donde NSPA es el número de soluciones extremas soportadas * número de arcos (orígenes*destinos) y CPUTime el tiempo computacional:



Conclusiones

En este trabajo se realiza un estudio algorítmico y computacional del Problema de Transporte Biobjetivo. Proponiendo una variante del algoritmo clásico para la resolución de dicho problema.

Se busca que el algoritmo propuesto presente mejores resultados frente al algoritmo clásico, por ello, se analizan los datos obtenidos en diversas pruebas computacionales.

El hecho de que en el problema de Transporte Biobjetivo Clásico subyace un grafo bipartito completo, permite observar que el algoritmo propuesto en el capítulo anterior, presenta mejores resultados prácticos.

A

Apéndice

A.1. Problema equilibrado

- Cuando tenemos un exceso en la oferta, esto es,

$$\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$$

creamos un destino ficticio (n+1) para el que su demanda sería

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

y el costo de trasportar ese exceso de demanda desde cualquier origen al destino (n+1) es cero, es decir,

$$c_{in+1} = 0, \forall i \in \{1, \dots, m\}$$

- Análogamente, cuando tenemos un exceso en la demanda, esto es,

$$\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$$

creamos un origen ficticio (m+1) para el que su oferta sería

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$$

y el costo de trasportar ese exceso de demanda desde el origen (m+1) a cualquier destino es cero, es decir,

$$c_{m+1j} = 0, \forall j \in \{1, \dots, n\}$$

Bibliografía

- [1] SCHRIJVER, Alexander. *Theory of linear and integer programming*. Reading: John Wiley and Sons, 1998.
- [2] TATON, René. *L'oeuvre scientifique de Monge* Reading: Presses Universitaires de France, París. 1951.
- [3] GONZÁLEZ MARTÍN, Carlos. *Programación lineal : Introducción a la programación entera y a la programación combinatoria*. Reading: La Laguna : Departamento de Estadística, Investigación Operativa y Computación, Universidad de La Laguna, 2003.
- [4] DANTZIG, George B. KOOPMANS, Tjalling C. JOHN WILEY AND SONS, INC. *Activity analysis of production and allocation*. New York: Conference held at the University of Chicago, 1949, pp. 187–200.
- [5] TOLSTOÏ A.N. *Transportation Planning, Volume I Methods of finding the minimal total kilometrage in cargotransportation planning*. Moscow: National Commissariat of Transportation, 1930, pp. 23–25.
- [6] MILLÁN G., Ana. *La aplicación de las matemáticas a los problemas de administración y organización: antecedentes históricos*. *ILUIL* Reading: Università di Roma: La Sapienza, 2003, vol 26, pp. 929-961.
- [7] HITCHCOCK, Frank L. *The distribution of a product from several sources to numerous localities*. *Journal of Mathematics and Physics*, 1941, vol.20, pp 222–378
- [8] *Biografías y Vidas*. La enciclopedia biográfica en línea. *Biografía de Leonid Kantoróvich* [en línea]. [Fecha de consulta: 16-4-2021]. Disponible en: <https://www.biografiasyvidas.com/biografia/k/kantorovich.htm>.
- [9] The Nobel Prize *Leonid Vitaliyevich Kantorovich Biographical* [en línea]. [Fecha de consulta: 25-4-2021]. Disponible en: <https://www.nobelprize.org/prizes/economic-sciences/1975/kantorovich/biographical/>.
- [10] A.Schrijver. *On the history of the transportation and maximum flow problems* [en línea]. 2001. University of Amsterdam, The Netherlands. [Fecha de consulta: 23-04-2021]. Disponible en: <https://homepages.cwi.nl/~lex/files/histtrpclean.pdf>.

- [11] The Nobel Prize in Economics. *Tjalling C. Koopmans* [en línea]. [Fecha de consulta: 26-4-2021]. Disponible en: <http://prizeineconomics.blogspot.com/2008/03/tjalling-c-koopmans-191085.html>.
- [12] Matias Andina. *Introducción a estadística con R* [en línea]. [Fecha de consulta: 15-8-2021]. Disponible en: <https://bookdown.org/matiasandina/R-intro/>.
- [13] D. Klingman, A. Napier and J. Stutz *NETGEN: A Program for Generating Large Scale Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems* [en línea]. [Fecha de consulta: 15-8-2021]. Disponible en: https://www.jstor.org/stable/2630092?seq=1#metadata_info_tab_contents.
- [14] Andrea Raith and Antonio Sedeño Noda *Finding extreme supported solutions of biobjective network flow problems: An enhanced parametric programming approach* [en línea]. [Fecha de consulta: 15-8-2021]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0305054817300035>.
- [15] R Core Team. *R: A language and environment for statistical computing*. 2019. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Efficient frontier in Bi-objective Transport Problems: an algorithmic and computational study



Noé Rodríguez Castro
 Facultad de Ciencias · Sección de Matemáticas
 Universidad de La Laguna
 alu0100781571@ull.edu.es

Abstract

In this work, an algorithmic and computational study of the Biobjective Transport Problem, a particular case of the minimum cost flow problems over networks, included, in turn, within Linear Programming is carried out. Before dealing with the Biobjective Transport Problem, the Uniojective or Classical Transport Problem is introduced, using a variant of the Simplex Method for its solution. Then we move on to the study of the Biobjective Transport Problem, implementing a variant of the classical Parametric Simplex algorithm for its resolution. Finally, a computational study is carried out where, by means of the results obtained in different computational experiences, we observe the advantages that the proposed algorithm can provide against the classical algorithm.

1. Introduction

The Classical Transportation Problem is a Linear Programming problem, posed as one of the first problems within Operations Research. This problem tries to determine an optimal solution of a bipartite directed graph, with arcs leaving the warehouses and arriving at the markets, satisfying the respective supply and demand constraints. When we have two objectives, i.e., we want to minimize (or maximize) two objective functions. This problem is called the Biobjective Transportation Problem, and its algorithmic and computational study is the purpose of this paper.

2. The Classical Transportation Problem

In the classical transportation problem, we try to establish the number of units to be transported from each of the origins to each of the destinations, minimizing the overall costs and satisfying the supply and demand constraints. The corresponding model is:

$$\begin{aligned} \min c^t x &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{sujeto a: } & \sum_{i=1}^m x_{ij} = a_i, \forall i \in U \\ & \sum_{j=1}^n x_{ij} = b_j, \forall j \in V \\ & x_{ij} \geq 0, \forall (i, j) \in A \end{aligned}$$

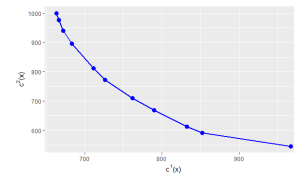
3. Bi-objective Transport Problem

In the bi-objective transport problem, we want to minimize (or maximize) two objective functions. To do this, we will pose the parametric problem and obtain a supported extreme efficient solution. And, starting from this solution, we will look for the rest

of the supported extreme efficient solutions. The corresponding model is:

$$\begin{aligned} \min c(x) = (c^1(x), c^2(x)) &= \left(\sum_{(i,j) \in A} c_{ij}^1 x_{ij}, \sum_{(i,j) \in A} c_{ij}^2 x_{ij} \right) \\ \text{sujeto a: } & - \sum_{\{j|(i,j) \in A\}} x_{ij} = -a_i, \forall i \in U \\ & \sum_{\{i|(i,j) \in A\}} x_{ij} = b_j, \forall j \in V \\ & x_{ij} \geq 0, \forall (i, j) \in A \end{aligned}$$

An algorithm for its resolution is proposed and executed in an example, obtaining the following efficient frontier of the problem:



4. Computational experiences

Several computational tests are performed, where 150 particular cases are generated for each algorithm, resulting in a total of 300 observations. We try to study the behavior of the CPUTime variable by using the following non-linear model:

$$CPUTime = C_1 + C_2 * X * Y^\alpha$$

5. Conclusions

The fact that a complete bipartite graph underlies the Classical Bi-objective Transport problem, allows us to observe that the proposed algorithm presents better practical results.

References

- [1] GONZÁLEZ MARTÍN, Carlos. *Programación lineal : Introducción a la programación entera y a la programación combinatoria*. Reading: La Laguna : Departamento de Estadística, Investigación Operativa y Computación, Universidad de La Laguna, 2003.
- [2] Matias Andina. *Introducción a estadística con R* [en línea]. [Fecha de consulta: 15-8-2021]. Disponible en: <https://bookdown.org/matiasandina/R-intro/>.
- [3] A.Schrijver. *On the history of the transportation and maximum flow problems* [en línea]. 2001. University of Amsterdam, The Netherlands. [Fecha de consulta: 23-04-2021]. Disponible en: <https://homepages.cwi.nl/~lex/files/histtrpclean.pdf>.