

**SOLUCIÓN DEL PROBLEMA CVRP MEDIANTE EL ALGORITMO DE  
DESCOMPOSICIÓN DE DANTZIG WOLFE**

**DANIEL ORLANDO MARTÍNEZ QUEZADA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2014**

**SOLUCIÓN DEL PROBLEMA CVRP MEDIANTE EL ALGORITMO DE  
DESCOMPOSICIÓN DE DANTZIG WOLFE**

**DANIEL ORLANDO MARTÍNEZ QUEZADA**

**Trabajo de grado para optar al título de:  
Ingeniero Industrial**

**Director:  
Henry Lamos Díaz  
Ph.D. Física - Matemática**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2014**

## **AGRADECIMIENTOS**

Al Profesor Henry Lamos Díaz por su paciencia y esmero en la realización de este proyecto.

A Silvia Galván Núñez y Miguel Ángel Beltrán Marín quienes me compartieron sus conocimientos y deseos por investigar.

A familiares y amigos que me apoyaron de una u otra manera a lo largo de este proceso, especialmente a Silvia Juliana Serrano Flórez por su tiempo y dedicación.

## CONTENIDO

	<b>pág.</b>
INTRODUCCIÓN .....	13
1.PLANTEAMIENTO DEL PROBLEMA .....	16
2.JUSTIFICACIÓN .....	17
3.OBJETIVOS .....	19
3.2.Objetivos Específicos .....	19
4.MARCO TEÓRICO.....	20
4.1.PROGRAMACIÓN LINEAL .....	20
4.2.PROGRAMACIÓN ENTERA.....	34
4.3.TEORÍA DE GRAFOS .....	43
4.4.PROBLEMA DE RUTEO DE VEHÍCULOS VRP .....	45
4.5.PROBLEMA DE RUTEO DE VEHÍCULOS CON RESTRICCIONES DE CAPACIDAD (CVRP) .....	48
5.ESTADO DEL ARTE .....	50
5.1.PROBLEMAS CON ESTRUCTURA DE ESCALERA (STAIRCASE LINEAR PROGRAMS, SLPS) .....	50
5.2.PROBLEMA DE RUTEO DE VEHICULOS (CVRP) Y MÉTODOS EXACTOS	50
5.3.GAMS (GENERAL ALGEBRAIC MODELING SYMTEM).....	56
6.DESARROLLO DE LA HERRAMIENTA.....	59
6.1.PROGRAMACIÓN LINEAL .....	59
6.2.PROBLEMA DUAL .....	59
6.3.DESCOMPOSICIÓN DE DANTZIG WOLFE .....	60
6.4.ANÁLISIS DE INSTANCIAS PARA EL CVRP .....	65

6.5.ARCHIVOS GAMS DATA EXCHANGE.....	68
6.6.ALGORITMO DE DESCOMPOSICIÓN DE DANTZIG WOLFE APLICADO AL CVRP .....	69
7.RESULTADOS OBTENIDOS .....	71
8.CONCLUSIONES.....	78
9.RECOMENDACIONES .....	79
BIBLIOGRAFÍA .....	81
ANEXOS .....	92

## LISTA DE FIGURAS

	<b>pág.</b>
Figura 1. Estructura de Bloque.....	31
Figura 2. Representación mediante un grafo del Problema de Transporte .....	45
Figura 3. Variantes de los Problemas de Transporte .....	47
Figura 4. Red de Shortest Path Problem.....	61
Figura 5. Seudocódigo de Dantzig-Wolfe a un SSP .....	63
Figura 6. Estructura de un archivo GDX.....	68
Figura 7. Seudocódigo Dantzig-Wolfe aplicado al CVRP .....	69
Figura 8. Tiempo en segundos vs máxima cardinalidad de S P_n16_k8 .....	72
Figura 9. Tiempo en segundos vs máxima cardinalidad de S P_n19_k2 .....	72
Figura 10. Tiempo en segundos vs máxima cardinalidad de S DW P_n16_k8 .....	74
Figura 11. Tiempo en segundos vs máxima cardinalidad de S DW P_n19_k2 .....	74
Figura 12. Comparación DW y la versión no descompuesta P_n16_k8.....	75
Figura 13. Comparación DW y la versión no descompuesta P_n19_k2.....	75
Figura 14. Red de Shorest Path Problem.....	92

## LISTA DE TABLAS

	<b>pág.</b>
Tabla 1. Resumen de operaciones entre simplex convencional y revisado .....	26
Tabla 2. Iteración 0 de la solución del SPP .....	64
Tabla 3. Ramificaciones del SPP .....	65
Tabla 4. Tiempos en segundos Exacto sin descomponer .....	71
Tabla 5. Tiempo en segundos método exacto descompuesto .....	73
Tabla 6. Columna 1 .....	95
Tabla 7. Columna 2 .....	96
Tabla 8. Columna 3 .....	98
Tabla 9. Columna 4 .....	99
Tabla 10. Columna 1 rama 1 .....	102
Tabla 11. Columna 2 rama 1 .....	103
Tabla 12. Columna 3 rama 1 .....	104
Tabla 13. Columna 1 Rama 2 .....	106
Tabla 14. Columna 2 rama 2 .....	108
Tabla 15. Columna 3 rama 2 .....	109
Tabla 16. Columna 4 rama 2 .....	110
Tabla 17. Columna 1 rama 3 .....	112
Tabla 18. Columna 1 rama 4 .....	114
Tabla 19. Columna 2 rama 4 .....	115

## LISTA DE ANEXOS

	<b>pág.</b>
Anexo A. Solución aplicación Branch Bound and Pricing manual para un SPP....	92
Anexo B. Guía de Ejecución de códigos .....	117
Anexo C. Script del método exacto completo sin descomponer.....	118
Anexo D. GMS del método exacto sin descomponer .....	120
Anexo E. Script del método de descomposición D-W .....	124
Anexo F. GMS del problema Maestro de la descomposición D-W.....	131
Anexo G. GMS del sub-problema de la descomposición D-W .....	133
Anexo H. Script para generar gráficas de las soluciones .....	137
Anexo I. Instancia P_n16_k8.....	138
Anexo J. Instancia P_n19_k2.....	139
Anexo K. Resultados P_n16_k8 sin descomponer.....	140
Anexo L. Resultados P_n19_k2 sin descomponer .....	154
Anexo M. Resultados P_n16_k8 descompuesto .....	164
Anexo N. Resultados P_n19_k2 descompuesto .....	178

## RESUMEN

**TÍTULO:** SOLUCIÓN DEL PROBLEMA CVRP MEDIANTE EL ALGORITMO DE DESCOMPOSICIÓN DE DANTZIG WOLFE\*

**AUTOR:** DANIEL ORLANDO MARTÍNEZ QUEZADA\*\*

**PALABRAS CLAVE:** Descomposición de Dantzig-Wolfe, problema de ruteo de vehículos con restricciones de capacidad, métodos exactos, generación de columnas.

### DESCRIPCIÓN:

En el siguiente trabajo se presenta la aplicación de un algoritmo de descomposición de Dantzig-Wolfe aplicado a dos instancias del benchmark para el CVRP, limitando la enumeración de las restricciones de sub-tours o ciclo hamiltonianos y a la vez comparándolo con una formulación sin descomponer usando como entorno de programación Matlab y GAMS de forma simultánea mediante el uso de archivos y librerías.gdx (GAMS data exchange) mostrando los resultados obtenidos, analizando la factibilidad de las respuestas generadas por cada límite en la cardinalidad del conjunto de restricciones de eliminación de sub-tours enumerados, encontrándose que los límites de memoria de los ordenadores comerciales son insuficientes para formular por completo un CVRP superior a 16 nodos en sistemas operativos de 32bits, pero también que en algunas instancias no es necesario formular el CVRP en su totalidad solo para llegar a una solución factible u óptima. Además se encuentra que en una formulación de descomposición converge más rápido que una formulación completa para algunos casos, finalmente se sugiere una formulación descompuesta para implementar con algoritmos mixtos entre estocásticos y determinísticos limitando el conjunto de sub-tours y evitar formularlo en su totalidad, viéndolo como un posible nuevo campo de investigación en problemas de optimización combinatoria junto con la aplicación de la herramienta de parallel pool de las nuevas versiones de Matlab para aplicaciones de computación en paralelo.

---

\* Proyecto de Grado

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales.  
Director: PhD. Henry Lamos Díaz.

## ABSTRACT

**TITLE:** SOLVING CVRP PROBLEM USING DANTZIG-WOLFE DECOMPOSITION ALGORITHM\*

**AUTHOR:** DANIEL ORLANDO MARTÍNEZ QUEZADA\*\*

**KEYWORDS:** Dantzig-Wolfe decomposition, capacitated vehicle routing problem , exact methods, column generation

### DESCRIPTION:

in this work, the application of a Dantzig-Wolfe decomposition algorithm to two instances for the capacitated vehicle routing problem (CVRP) benchmark is presented limiting the sub-tours restrictions or hamiltonian cycles enumeration, and it is compared to a non-decomposed formulation using simultaneously Matlab and GAMS as the programming environments by using GDX (gams data exchange) files and libraries.

the results are shown by analyzing the feasibility of the answers generated by each cardinality limit of the numerated sub-tours constrains set. It was found that the commercial computers memory limits are not big enough to formulate a complete 16 nodes CVRP in a 32 bits operating system, but in some instances it is not necessary to formulate the complete CVRP in order to find a feasible or optimal solution.

besides, it was found that a decomposition formulation converge faster than a complete formulation in some cases, finally it is suggested to use a decomposed formulation to execute with mixed stochastic and deterministic algorithms limiting the sub-tours set, avoiding to formulate it entirely and looking at it like a possible new research field in combinatorial optimization problems including the use of the parallel pool tool of the new Matlab version for parallel computing applications.

---

\* Degree Project

\*\* Faculty of Physicomechanical Engineering. Industrial and Business School. Director: PhD. Henry Lamos Díaz

## INTRODUCCIÓN

Un problema importante en el sector de transporte de carga colombiano son los cambios económicos, reflejados en las variaciones del precio de la gasolina, del ACPM, los fletes, etc<sup>1</sup>. En busca de soluciones a esta situación, el Estado ha creado mesas de trabajo que han permitido identificar algunos problemas de las empresas del sector transporte y crear indicadores como el ICTC (Índice de Costos del Transporte de Carga), para evaluar las variaciones económicas exclusivas del sector<sup>2</sup>. Uno de estos problemas identificados es la informalidad del sector, desactualización del parque automotor de las empresas de transporte colombianas, con una antigüedad promedio de 24,4 años<sup>3</sup>. Por esto, el Estado ha propuesto iniciativas para la formalización y actualización del sector de transporte de carga. Una iniciativa reciente es facilitar créditos a los empresarios para el cambio del parque automotor por nuevos modelos<sup>4</sup>.

Una dificultad no menor en la problemática es la infraestructura vial. Este factor ha sido identificado por el estado colombiano no sólo para el problema del sector de transporte de carga sino también en el transporte público<sup>5</sup>, es así que se incluye como una de las locomotoras que impulsa el desarrollo<sup>6</sup>. Pese a los conflictos

---

<sup>1</sup> PRENSA-LATINA.CU. Camioneros colombianos y Gobierno en nueva mesa de diálogo. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.prensa-latina.cu/index.php?option=com\\_content&task=view&idioma=1&id=2315641&Itemid=1](http://www.prensa-latina.cu/index.php?option=com_content&task=view&idioma=1&id=2315641&Itemid=1)

<sup>2</sup> HILLER, Frederick S. and LIBERMAN, Gerald J., 2010, *Introducción a la Investigación de Operaciones*. 9. Mexico. ISBN 970-10-5621-3.

<sup>3</sup> CONPES 3489. Política Nacional de Transporte Público Automotor de Carga [online]. 2007. Available from: <https://www.mintransporte.gov.co/descargar.php?id=3128>

<sup>4</sup> COLOMBIANO.COM. Camioneros tendrán líneas de crédito por 120.000 millones de pesos - El Colombiano. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.elcolombiano.com/BancoConocimiento/C/camioneros\\_tendran\\_lineas\\_de\\_credito\\_por\\_120000\\_millones\\_de\\_pesos/camioneros\\_tendran\\_lineas\\_de\\_credito\\_por\\_120000\\_millones\\_de\\_pesos.asp](http://www.elcolombiano.com/BancoConocimiento/C/camioneros_tendran_lineas_de_credito_por_120000_millones_de_pesos/camioneros_tendran_lineas_de_credito_por_120000_millones_de_pesos.asp)

<sup>5</sup> EL-TIEMPO.CO. Avanzan obras para recuperación vial en Sogamoso - Colombia Noticias: Actualidad Nacional - ELTIEMPO.COM. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.eltiempo.com/colombia/boyaca/ARTICULO-WEB-NEW\\_NOTA\\_INTERIOR-13378439.html](http://www.eltiempo.com/colombia/boyaca/ARTICULO-WEB-NEW_NOTA_INTERIOR-13378439.html)

<sup>6</sup> HILLER, Frederick S. and LIBERMAN, Gerald J., 2010, *Introducción a la Investigación de Operaciones*. 9. Mexico. ISBN 970-10-5621-3.

administrativos internos, el país ha evolucionado en cuanto a la estructura vial<sup>7</sup> y se proyecta para el 2016 con la construcción de 36 kilómetros de vías, pavimentar 718 kilómetros, construcción de 64 puentes y 4 túneles en el país<sup>8</sup>. Estas iniciativas reflejan un compromiso por parte del estado colombiano a mejorar el sector de transporte, generando un mercado más competitivo en el que se le dará importancia a la administración de recursos en las operaciones de transporte, enfocándose en una reducción de costos.

Reducir costos representa un escenario de competencia en un mercado desarrollado. La forma más utilizada para alcanzar una reducción de costos es mediante la construcción de aplicaciones informáticas que ayuden a diseñar y programar la flota de vehículos para la distribución. En el mercado mundial se encuentran aplicaciones para la programación y seguimiento de rutas, enfocadas a reducir el costo de transporte en las organizaciones. Estos aplicativos se basan en algoritmos de búsqueda inteligentes (heurísticas, metaheurísticas) que generan soluciones sub-óptimas con un mínimo consumo de tiempo computacional.

El creciente desarrollo económico colombiano y el incremento poblacional<sup>9</sup> ha generado la necesidad que las empresas de transporte compitan con herramientas computacionales robustas, para asegurar una solución óptima para cualquier instancia y con tiempo computacional bajo. La comunidad científica elabora modelos matemáticos con el propósito de entender los fenómenos del transporte,

---

<sup>7</sup> LA-REPUBLICA.COM. Concesión para malla vial de Bogotá se suscribió con empresa de papel: Cardozo | La República. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.larepublica.co/economia/concesión-para-malla-vial-de-bogotá-se-suscribió-con-empresa-de-papel-cardozo\\_99266](http://www.larepublica.co/economia/concesión-para-malla-vial-de-bogotá-se-suscribió-con-empresa-de-papel-cardozo_99266).

<sup>8</sup> CONPES 3706. Importancia Estratégica del Programa “Corredores Prioritarios para la Prosperidad” [online]. 2011. [Cited 21 February 2014]. Available from: <https://www.dnp.gov.co/LinkClick.aspx?fileticket=johYvpSM6Mc=&tabid=1260>

<sup>9</sup> ELTIEMPO.COM. Balance sobre el Sistema Integrado. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.eltiempo.com/colombia/bogota/balance-sobre-el-sistema-integrado\\_13411237-4](http://www.eltiempo.com/colombia/bogota/balance-sobre-el-sistema-integrado_13411237-4)

los cuales brindan buenas aproximaciones a modelos reales con los problemas de ruteo de vehículos incluso desde sus inicios<sup>10</sup>. El presente trabajo busca estudiar técnicas y modelos de ruteo de vehículos caracterizados por su complejidad NP-hard<sup>11</sup>, con el propósito de desarrollar una herramienta computacional aplicada a instancias del Problema de Ruteo de Vehículos con Restricciones de Capacidad CVRP, el desarrollo de este trabajo incluye la solución de dos instancias del benchmark para CVRP aplicando la descomposición de Dantzig Wolfe comparando los resultados con un método exacto aplicado a una formulación no descompuesta.

.

---

<sup>10</sup> DANTZIG, G. B. and RAMSER, J. H. The Truck Dispatching Problem. *Management Science* [online]. 1 October 1959. Vol. 6, no. 1, [Cited 16 January 2014]. p. 80–91. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.6.1.80>, DOI 10.1287/mnsc.6.1.80.

<sup>11</sup> TOTH, Paolo and VIGO, Daniele (eds.). *The Vehicle Routing Problem*. 1. Bologna, Italy : SIAM, 2002. ISBN 0898715792.

.

## 1. PLANTEAMIENTO DEL PROBLEMA

El problema de ruteo de vehículos VRP (Vehicle Routing Problem) determina el conjunto óptimo de rutas para un parque automotor destinado a servir un conjunto de clientes. Éste es considerado como uno de los problemas más estudiados en el campo de la optimización combinatoria.

La primera variante del VRP es el CVRP (Capacitated Vehicle Routing Problem), propuesto por primera vez por Dantzig y Rasmer en 1959 para el despacho de gasolina a estaciones de servicio. Como su nombre lo indica, el objetivo de este problema es encontrar el conjunto de rutas óptimas para un grupo de camiones que tienen una capacidad máxima de carga dada en unidades de volumen, peso o unidades de productos. Todas las variantes posteriores se basan en el CVRP, lo que ha contribuido a que ésta sea la variante más estudiada en cuanto a métodos de solución.

Los primeros intentos de solución para este tipo de problemas con complejidad NP-Hard fueron realizados a través de métodos exactos con tiempos computacionales altos para alcanzar una solución óptima. Lo anterior obligó a la comunidad científica a desarrollar métodos o estrategias que disminuyan el tiempo, para alcanzar la mejor solución en el caso de métodos exactos o de soluciones factibles buenas con tiempos computacionales aceptables (comparados con aquellos empleados por los métodos exactos) en el caso de métodos heurísticos, metaheurísticos, híbridos entre otros.

## 2. JUSTIFICACIÓN

En un entorno económico competitivo es necesario generar soluciones óptimas para problemas de la vida real. Debido a la alta complejidad y el sinnúmero de aplicaciones del Problema de Ruteo de Vehículos con Restricciones de Capacidad, es de gran importancia desarrollar técnicas y explorar nuevos métodos para obtener soluciones óptimas o suficientemente competitivas.

Los métodos de solución exacta para los problemas de ruteo han sido muy eficientes para la solución de problemas con instancias de tamaño pequeño. Por consiguiente, es de interés para la comunidad seguir explorando estrategias que permitan usarlos en instancias de mayor tamaño. En estudios recientes como Fukasawa, Longo, Lysgaard, Aragão, Reis, Uchoa, Werneck (2005)<sup>12</sup>; Santos, Cunha, Mateus (2012)<sup>13</sup> y Subramanian, Uchoa, Pessoa, Ochi (2012)<sup>14</sup>, han vuelto a considerar el método de descomposición para la solución de problemas combinatorios.

En este trabajo se usará el enfoque de Branch Cut and Pricing como estrategia de solución para el CVRP. Los experimentos numéricos que se realizarán para medir la eficiencia del algoritmo se desarrollarán en GAMS, usando algunas de las

---

<sup>12</sup> FUKASAWA, Ricardo, LONGO, Humberto, LYSGAARD, Jens, ARAGÃO, Marcus Poggi de, REIS, Marcelo, UCHOA, Eduardo and WERNECK, Renato F. Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. *Mathematical Programming* [online]. 12 October 2005. Vol. 106, no. 3 [Cited 20 December 2013]. , p. 491–511. Available from: <http://link.springer.com/10.1007/s10107-005-0644-x>, DOI 10.1007/s10107-005-0644-x.

<sup>13</sup> SANTOS, Fernando Afonso, CUNHA, Alexandre Salles and MATEUS, Geraldo Robson. Branch-and-price algorithms for the Two-Echelon Capacitated Vehicle Routing Problem. *Optimization Letters* [online]. 17 October 2012. Vol. 7, no. 7, [Cited 20 December 2013]. p. 1537–1547. Available from: <http://link.springer.com/10.1007/s11590-012-0568-3>. DOI 10.1007/s11590-012-0568-3.

<sup>14</sup> SUBRAMANIAN, Anand, UCHOA, Eduardo, PESSOA, Artur Alves and OCHI, Luiz Satoru. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters* [online]. 21 October 2012. Vol. 7, no. 7, [Cited 20 December 2013]. p. 1569–1581 Available from: <http://link.springer.com/10.1007/s11590-012-0570-9>. . DOI 10.1007/s11590-012-0570-9.

librerías de Matlab; los buenos resultados que ha obtenido anteriormente, lo hacen un buen campo de investigación para desarrollar herramientas computacionales para un entorno económico competitivo.

### **3. OBJETIVOS**

**3.1. Objetivo General.** Resolver el Problema de Ruteo de Vehículos con Restricciones de Capacidad (CVRP), usando métodos de descomposición.

#### **3.2. Objetivos Específicos**

- Organizar y presentar la literatura relacionada con las técnicas de descomposición para el CVRP.
- Implementar el algoritmo de descomposición de Dantzig Wolfe para la solución del CVRP, usando las herramientas de Matlab y GAMS mediante archivos GDX (GAMS Data Exchange).
- Presentar experimentos numéricos sobre las medidas de eficiencia del algoritmo desarrollado mediante instancias del benchmarking.

## 4. MARCO TEÓRICO

### 4.1. PROGRAMACIÓN LINEAL

Un problema de programación lineal está definido como

$$\text{Max o Min } z = cx \quad (1)$$

$$Ax \leq b \quad (2)$$

$$x \geq 0 \quad (3)$$

Dados

$$c = (c_1, \dots, c_n)$$

$$b = (b_1, \dots, b_m)$$

$$A = (a_1, \dots, a_n)$$

$$a_j = (a_{j1}, \dots, a_{mn})$$

Siendo  $m$  el número de restricciones (2 y 3) que limitan la función objetivo (1) y  $n$  la cantidad de variables de decisión.

En la literatura existen diversas realidades que se estudian mediante el modelo (1)-(3)<sup>15</sup>.

**4.1.1 Métodos exactos para problemas de programación lineal.** Dentro de los métodos exactos para la solución de problemas de programación lineal se encuentra el algoritmo Simplex, Simplex revisado, Factorización LU, Descomposición de Dantzig Wolfe, generación de columnas, entre otros; los cuales se revisan a continuación.

---

<sup>15</sup>GARFINKEL, ROBERT S. NEMHAUSER, George L. Integer Programming. I. United States of America : JOHN WILEY & SONS, 1972. ISBN 0-471-29196-1. p.24

**4.1.1.1 Método Simplex.** El algoritmo Simplex es un método iterativo reconocido como el mejor en cuanto a la aplicación de problemas de programación lineal. Para estudiar el simplex se deben identificar los siguientes elementos:

El conjunto definido como

$$X = \{x | Ax \leq b, x \geq 0\} \quad (4)$$

Se denomina dominio factible a los valores de  $x$  que satisfacen el conjunto de desigualdades  $Ax \leq b$ .

El símbolo  $\leq$  es utilizado de forma didáctica, pero en su lugar se utiliza cualquier combinación de  $<$ ,  $>$  ó  $=$ . Tratar sistemas de desigualdades representa un conjunto infinito de puntos del dominio factible, que para aplicaciones computacionales son imposibles de enumerar. Por lo tanto, es común tratar las desigualdades como igualdades, para lo cual se procede a establecer las ecuaciones de frontera de las restricciones sustituyendo los signos  $<$ ,  $>$  ó  $=$  sólo por un signo  $=$ <sup>16</sup>. El anterior tratamiento reduce las dificultades de obtener soluciones de un sistema de inecuaciones al convertirlo en un sistema de ecuaciones, generando una programación matemática aumentada. Un problema de maximización estándar con sólo restricciones de forma  $\leq$  se representa como sigue<sup>17</sup>

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1 \quad (1)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2 \quad (2)$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} = b_m \quad (m)$$

<sup>16</sup> HILLER, Frederick S. and LIBERMAN, Gerald J. Introducción a la Investigación de Operaciones. 9. Mexico, 2010. ISBN 970-10-5621-3. p.148

<sup>17</sup> Ibíd. p.150

Dada la estructura del problema, existen aún infinitas soluciones. Para controlar esto, se dividen las variables en dos conjuntos. El primer conjunto corresponde a variables básicas  $BV$ , las cuales se definen en función de otras variables no básicas  $NV$ , éstas últimas toman el valor de 0 en el algoritmo Simplex. Lo anterior restringe la cantidad de soluciones al número de posibles combinaciones de variables básicas ( $m$ ) del total de variables ( $n + m$ ), garantizando así una cantidad finita de soluciones<sup>18</sup>.

*“Una solución factible en un vértice es una solución que no se encuentra en cualquier segmento rectilíneo que conecta a otras dos soluciones factibles”<sup>19</sup>.*

El enunciado anterior, de manera individual, no presenta un impacto relevante. Estas soluciones factibles en los vértices tienen una propiedad característica en un problema de programación lineal con región factible acotada; si se garantiza sólo una solución óptima, ésta debe ser la solución en un vértice; si existen múltiples soluciones óptimas, entonces al menos dos deben ser soluciones factibles en los vértices adyacentes<sup>20</sup>.

El Simplex es un método iterativo que inicia tomando el origen como valor para las variables básicas. Luego, se verifica la optimalidad con los índices de mejora de las soluciones adyacentes a la actual. Una solución adyacente es aquella que con respecto a otra tiene en común  $m - 1$  variables básicas<sup>21</sup>. Si las soluciones adyacentes no mejoran la solución actual, ésta última es considerada como la óptima, dando por terminado el algoritmo; si la solución es mejorada, se modifican los conjuntos de variables básicas y no básicas para generar la solución que mejore la actual.

---

<sup>18</sup> Ibíd. p.153

<sup>19</sup> Ibíd. p.149

<sup>20</sup> Ibíd. p.152

<sup>21</sup> Ibíd. p.152

Como se puede encontrar en los diferentes textos de investigación de operaciones, hay muchas maneras de abordar este tipo de problemas. El más común es el Simplex por tableros, el cual consiste en realizar operaciones de pivoteo en un arreglo de los coeficientes de la función objetivo (1) y los coeficientes de las restricciones (2); esta versión del Simplex es recomendada para aplicaciones pequeñas y manuales.

**4.1.1.2 Método Simplex Revisado.** Para aplicaciones computacionales, se ha preferido el uso matricial del simplex (el resto de la exposición se realizará mediante matrices). El método Simplex Revisado se utiliza para la solución de problemas de gran tamaño, es decir es cuando el problema tiene cientos de variables de decisión y, también, miles de restricciones de igualdad o desigualdad, como se describe a continuación.

Sea el problema de programación lineal escrito en forma matricial<sup>22</sup>

$$\text{Max o Min } z = C_{BV}x_{BV} + C_{NBV}x_{NBV} \quad (4)$$

Sujeto a

$$Bx_{BV} + Nx_{NBV} = b \quad (5)$$

$$x_{BV}, x_{NBV} \geq 0 \quad (6)$$

donde

$z$  es el valor de la función objetivo.

$C_{BV}$  son los coeficientes asociados a las variables básicas.

$C_{NBV}$  son los coeficientes asociados a las variables no básicas.

$x_{BV}$  las variables de la programación lineal que son básicas.

$x_{NBV}$  las variables de la programación lineal que no son básicas.

---

<sup>22</sup> WINSTON, Wayne L. Investigación de Operaciones Aplicaciones y Algoritmos. 4. p.269-271

$B$  es la matriz que resulta de las columnas de  $A$ , seleccionando las columnas  $a_j$  que corresponden a las variables básicas.

$N$  la matriz que se obtiene de las columnas de  $A$ , con  $a_j$  que corresponde a las variables no básicas.

$b$  es el vector de los lados derechos respectivos de las restricciones de  $A$  aumentada.

De esta forma, se puede obtener cualquier parte de una iteración del algoritmo simplex, sólo es cuestión de despejar el término deseado, por ejemplo<sup>23</sup>:

Al multiplicar la ecuación (5) por  $B^{-1}$  se tiene

$$x_{BV} + B^{-1}Nx_{NBV} = B^{-1}b \quad (7)$$

Un análisis de (7) sugiere que el término de  $B^{-1}Nx_{NBV}$  es igual a 0, dado que uno de los supuestos del simplex es que  $x_{NBV}$  se eligen igual 0, por lo tanto, la ecuación (7) se reduce a  $x_{BV} = B^{-1}b$ .

De forma análoga para encontrar alguna columna específica de  $A$  se procede a realizar el cálculo:  $\bar{a}_j = B^{-1}a_j$ .

Multiplicando (7) por  $C_{BV}$  se genera

$$C_{BV}x_{BV} + C_{BV}B^{-1}Nx_{NBV} = C_{BV}B^{-1}b \quad (8)$$

Despejando  $z$  de (4)

$$z - C_{BV}x_{BV} - C_{NBV}x_{NBV} = 0 \quad (9)$$

Al Sumar (8) y (9)

$$z + (C_{BV}B^{-1}N - C_{NBV})x_{NBV} = C_{BV}B^{-1}b \quad (10)$$

---

<sup>23</sup> Ibíd. p.269-271

De la ecuación (10), se puede identificar que los valores de  $z$  dependen de las variables no básicas de la solución, y el criterio que define si se puede o no mejorar la solución actual es el término  $(C_{BV}B^{-1}N - C_{NBV})$ . Si algún elemento de este arreglo es positivo en un problema de maximización, la solución actual se puede mejorar; para un problema de minimización es el caso contrario. El término  $(C_{BV}B^{-1}N - C_{NBV})$  se conoce como arreglo de costos o coeficientes reducidos, y se asocia a las variables no básicas de acuerdo a

$$\bar{c}_j = C_{BV}B^{-1}a_j - c_j \quad (11)$$

El dominio de  $j$  está determinado por el conjunto de variables no básicas, y el término  $C_{BV}B^{-1}$  se aplica repetidamente en el método, representado usualmente como  $\lambda$  y se calcula una vez por iteración. Compilando lo anterior, cada iteración del Simplex Revisado consta de tres sistemas de ecuaciones

$$Bx_{BV} = b \quad (12)$$

$$B\bar{a}_j = a_j \quad (13)$$

$$\lambda B = C_{BV} \quad (14)$$

Para efectuar cada iteración del sistema de ecuaciones (14), se obtiene  $\lambda$  y con él se calculan los costos reducidos de (11), identificando que la variable no básica debe convertirse en una variable básica; de igual manera, si no se logra identificar alguna posible mejora, la solución actual es óptima. Paso seguido, con los sistemas de ecuaciones (12) y (13) se busca cuál de las variables básicas se hace primero 0 al variar la variable no básica identificada; en la siguiente iteración, se hace un intercambio entre las variables seleccionadas, pasando la no básica a básica y la básica a no básica, al igual que en el simplex convencional.

### Simplex y Simplex Revisado

Para el método de Simplex Revisado es necesario un arreglo de  $(m + 1) \times (m + 1)$ , por el contrario, el Simplex convencional requiere de una matriz de  $(m + 1) \times (n + 1)$ . A simple vista, se puede ver que el Simplex Revisado tendría una ventaja en cuanto al ahorro de memoria, siempre y cuando el problema a solucionar tenga un número de variables mucho mayor que el número de restricciones. Por otro lado, hay una pequeña ventaja en la cantidad de operaciones del Simplex convencional sobre la del Simplex Revisado. En la Tabla 1 se muestra un resumen de las operaciones que se efectúan en cada uno.

Tabla 1. Resumen de operaciones entre simplex convencional y revisado

Método		Pivoteo	$z_j - c_j$	Total Operaciones
Simplex	Multiplicaciones	$(m+1)(n-m+1)$		$m(n-m)+n+1$
	Sumas	$m(n-m+1)$		$m(n-m+1)$
Simplex revisado	Multiplicaciones	$(m + 1)^2$	$m(n-m)$	$m(n-m)+(m + 1)^2$
	Sumas	$m(m+1)$	$m(n-m)$	$m(n+1)$

Fuente: Bazaraa, Jarvis, Sherali (1998)<sup>24</sup>

**4.1.1.3 Factorización LU.** La descomposición o factorización LU es un método numérico ampliamente estudiado para la solución de sistemas de ecuaciones lineales. Este método se basa en operaciones de eliminación Gaussianas que transforman una matriz de coeficientes en dos matrices fundamentales, para aplicar algoritmos de sustitución sucesiva y así encontrar la solución del sistema.

Sea

<sup>24</sup> HILLER, Frederick S. and LIBERMAN, Gerald J. Introducción a la Investigación de Operaciones. 9. Mexico, 2010. ISBN 970-10-5621-3

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad \text{Matriz de coeficientes}$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad \text{Lado derecho del sistema de ecuaciones.}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \text{Variables del sistema de ecuaciones.}$$

El sistema de ecuaciones de la forma

$$Ax = b \quad (15)$$

Una solución de manera directa sería

$$x = A^{-1}b \quad (16)$$

La inversa de una matriz requiere espacio de memoria en su cálculo, lo cual se traduce en un mayor tiempo de cómputo; el método de Factorización LU evita esta operación y desarrolla el problema descomponiendo la matriz  $A$  de la siguiente manera

$$A = LU$$

Siendo

$$U = \begin{pmatrix} u_{11} & \cdots & \cdots & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \vdots & \vdots & \vdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & u_{ii} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & u_{nn} \end{pmatrix} \quad \text{Matriz triangular superior.}$$

$$L = \begin{pmatrix} 1 & \cdots & \cdots & \cdots & \cdots & 0 \\ l_{21} & 1 & \vdots & \vdots & \vdots & \vdots \\ \vdots & l_{33} & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ l_{n1} & \cdots & \cdots & \cdots & l_{n(n-1)} & 1 \end{pmatrix} \quad \text{Matriz triangular inferior.}$$

Para obtener los arreglos anteriormente presentados, se utilizan operaciones de eliminación gaussiana, conocidas también como operaciones fundamentales entre renglones para las matrices de coeficientes. De igual manera que el Simplex, las operaciones de eliminación gaussiana tienen una versión manual que involucra un esfuerzo mayor a su contraparte matricial.

En la mayoría de textos de Algebra Lineal o de métodos numéricos<sup>25</sup> se puede encontrar la versión matricial de la Factorización LU. Este método básicamente propone que existe una matriz  $E_i$  que, al pre-multiplicarse por la matriz de coeficientes  $A$ , puede efectuar la  $i$ -ésima operación fundamental entre renglones. La estructura de la matriz  $E_i$  puede estar dada de la siguiente forma

$$E_i = \begin{pmatrix} 1 & 0 & -a_{1j}/a_{ij} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ 0 & 0 & 1/a_{ij} & \vdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \vdots & -a_{mj}/a_{ij} & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Esta estructura se cumple si y sólo si la matriz de coeficientes no tiene ceros en la diagonal principal. Si el sistema tiene algún 0 en el arreglo de coeficientes, la estructura cambia por la permutación de una idéntica, para cambiar de posición dos filas y así continuar con el procedimiento.

El método de descomposición o Factorización LU es muy importante en la aplicación a problemas de programación lineal, ya que al combinarse con el Simplex Revisado presenta grandes ventajas si se utilizan estrategias de pivotes

---

<sup>25</sup>STRANG, Gilbert. Linear Algebra. Second. New York, NY, 1980. p.22

parciales en las iteraciones del Simplex Revisado, este variante del simplex se explicará a continuación.

### **Simplex Revisado y Factorización LU utilizando pivotes parciales**

Como se discutió anteriormente, el Simplex Revisado permite que desde un enfoque matricial el método simplex se reduzca a resolver los siguientes tres sistemas de ecuaciones

$$Bx_{BV} = b \quad (17)$$

$$B\bar{a}_j = a_j \quad (18)$$

$$\lambda B = C_{BV} \quad (19)$$

Como se observa, los tres sistemas tienen en común el término de la base  $B$  que corresponde a las columnas de la matriz  $A$ , las cuales están asociadas a las variables básicas de la iteración en curso. La Factorización LU involucra este término que otorga una ventaja en la reducción de cálculos; al igual como se planteó para sistemas de ecuaciones lineales cuadrados, la base debe ser linealmente independiente entre sus columnas.

Sea

$$E_t \dots E_2 E_1 B = U \quad (20)$$

Si (20) se cumple, entonces

$$L^{-1} = E_t \dots E_2 E_1 \quad (21)$$

De esta forma, se obtienen los elementos para que en una iteración se puedan resolver los sistemas de ecuaciones (17), (18) y (19). Ésta es sólo una forma de combinar el Simplex con la Factorización LU, que puede mejorarse cambiando un poco la estructura de la base y de las variables a solucionar en los sistemas de ecuaciones planteados.

Suponiendo

$$X = \{x_1, x_2 \dots x_n\} \quad (22)$$

$$B = \{B_1, B_2, \dots, B_m\} \quad (23)$$

$$b = \{b_1, b_2, \dots, b_m\} \quad (24)$$

De forma ilustrativa, el término (22) representa las variables a solucionar; el (24), el lado derecho del sistema; Y el término (23) representa las columnas asociadas a la base. Cabe recordar que estos términos se asumen linealmente independientes entre sí para que el problema pueda tener solución.

Ahora, suponiendo que al finalizar una iteración en un Simplex Revisado se obtenga que  $x_k$  es la variable básica que pasa a ser no básica y que  $x_q$  es la variable no básica candidata a ser básica en la siguiente iteración, la base se reescribiría de la siguiente forma

$$\hat{B} = \{B_1, B_2, \dots, B_{k-1}, B_{k+1}, \dots, B_m, B_q\} \quad (25)$$

De esta forma, los elementos de la descomposición para el caso de  $x$  y  $b$  son

$$L^{-1}\hat{B} = \{L^{-1}B_1, L^{-1}B_2, \dots, L^{-1}B_{k-1}, L^{-1}B_{k+1}, \dots, L^{-1}B_m, L^{-1}B_q, \} \quad (26)$$

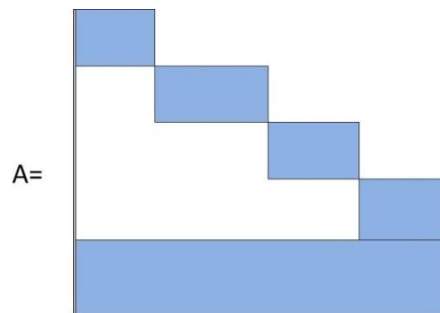
$$L^{-1}\hat{B} = \{U_1, U_2, \dots, U_{k-1}, U_{k+1}, \dots, U_m, U_p, \} = M \quad (27)$$

El término  $M$  en la ecuación (27) corresponde a la matriz  $U$  con dislocación en las columnas anterior y posterior de la variable  $k$  que deja de ser básica. Hasta el momento no se ha hecho ninguna operación de eliminación gaussiana, sólo se ha reescrito la base y las variables con su respectivo lado derecho. Sin embargo, para terminar la Factorización LU, de aquí en adelante, se comienzan a efectuar las operaciones fundamentales entre filas, sólo desde la columna predecesora de

la variable básica que salió, lo cual disminuye la cantidad de operaciones al no necesitarse la realización completa de la Factorización LU sino sólo parcialmente. Finalmente, se procede a resolver los sistemas de ecuaciones y continuar con las siguientes iteraciones del Simplex Revisado si las hay.

**4.1.1.4 Algoritmo de Descomposición de Dantzig Wolfe.** Se dice que un problema tiene estructura de bloque si la matriz de restricciones tiene una forma parecida a la que enseña la Figura 1; en donde las áreas de color azul representan coeficientes diferentes de 0, y las áreas no sombreadas representan coeficientes iguales a 0.

Figura 1. Estructura de Bloque



En estudios acerca de la descomposición de Dantzig Wolfe como Cornuejols, Harche (1993)<sup>26</sup>, los dominios factibles de los problemas a resolver se representan mediante politopos o poliedros, que tienen como característica principal el ser acotados; como en  $\mathbb{R}^3$  se puede observar que un poliedro realmente es acotado, se supone así para espacios numéricos superiores a tres dimensiones.

---

<sup>26</sup> CORNUEJOLS, Gerard and HARCHE, Farid. Polyhedral study of the capacitated vehicle routing problem. Mathematical Programming [online]. June 1993. Vol. 60, no. 1-3, [Cited 14 November 2013]. p. 21–52. DOI 10.1007/BF01580599.

Para identificar la convexidad de un conjunto de restricciones, estas deben cumplir con el teorema de la representación de Minkowski enunciado a continuación<sup>27</sup>:

Sea

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \quad \text{una matriz de restricciones}$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \quad \text{vector del lado derecho}$$

$x^i$  es un punto extremo del poliedro.

$r^i$  rayo extremo del poliedro.

$\lambda_i, \mu_i$  las ponderaciones asociadas a la combinación lineal del poliedro.

Dado un poliedro  $P = \{x \in \mathbb{R}^n ; Ax \preceq b\}$ . Si el poliedro  $P$  es no vacío y  $A$  es de grado  $n$ , entonces  $P = P'$

$$P' = \left\{ x \in \mathbb{R}^n ; \sum_{i \in \Pi} \lambda_i x^i + \sum_{j \in R} \mu_j r^j ; \sum_{i \in \Pi} \lambda_i = 1 ; \lambda, \mu \geq 0 \right\}$$

El teorema de la representación de Minkowski expresa que en un poliedro convexo, todo punto perteneciente al poliedro puede ser expresado como una combinación lineal de los puntos o rayos extremos del poliedro. Para poder estudiar el método de descomposición de Dantzig Wolfe se supone un problema de programación lineal con la siguiente estructura

$$\text{Minimizar } Z = cx \tag{28}$$

Sujeto a:

---

<sup>27</sup> TEBBOTH, James Richard. A Computational Study of Dantzig-Wolfe Decomposition. 2001.

$$Ax = b \quad (29)$$

$$x \in X \quad (30)$$

$X$  es un conjunto poliédrico de estructura especial, por lo que  $A$  también debe tener una estructura especial o de bloque. Es de notar que debe ser fácil reconocer subconjuntos de restricciones de estructura especial que sólo dependan de un grupo específico de variables de decisión por cada subgrupo y otro gran conjunto de las cuales pueden depender todas las variables. Suponiendo que el conjunto poliédrico es convexo y acotado, es decir que cualquier punto  $x$  se puede expresar como una combinación lineal de los puntos extremos y sólo de estos; además, al ser acotado no intervienen direcciones extremas, así  $x$  se expresa según el teorema de Minkowski.

$$x = \sum_{j=1}^t \lambda_j x^j \quad (31)$$

Aquí,

$$\sum_{j=1}^t \lambda_j = 1 \quad (32)$$

$$\lambda_j \geq 0 \quad j = 1, 2, \dots, t \quad (33)$$

Se reemplaza (31) en el problema (28)-(30), y se tiene la descomposición de Dantzig Wolfe, conocida como el problema maestro, con la siguiente estructura.

$$\text{Minimizar } Z = \sum_{j=1}^t (c x^j) \lambda_j \quad (34)$$

Sujeto a:

$$\sum_{j=1}^t (Ax^j) \lambda_j \quad (35)$$

$$\sum_{j=1}^t \lambda_j \quad (36)$$

$$\lambda_j \geq 0 \quad j = 1, 2, \dots, t \quad (37)$$

El conjunto de puntos extremos  $x^j$  del conjunto poliédrico es, en muchos de los casos, un número considerablemente grande que consume una gran cantidad de tiempo al enumerarlos. Lo anterior lleva a que se usen métodos eficientes como la generación de columnas, que ayuden a hallar los puntos extremos necesarios para determinar la solución óptima del problema sin tener que enumerarlos.

### Generación de Columnas

La Generación de Columnas consiste en hallar la mejor variable no básica candidata a ingresar a la base de la siguiente iteración de un Simplex sin necesidad de conocer todas las variables no básicas, para esto se propone el costo reducido como un problema de programación lineal de la siguiente forma

$$Z_k - \hat{c}_j = \text{Maximo}_{1 \leq j \leq t} Z_j - \hat{c}_j = \text{Maximo}_{1 \leq j \leq t} (w, \alpha) \begin{pmatrix} Ax^j \\ 1 \end{pmatrix} - cx^j \quad (38)$$

$$= \text{Maximo}_{1 \leq j \leq t} wAx^j + \alpha - cx^j \quad (39)$$

De esta forma, la descomposición de Dantzig Wolfe se puede combinar con el Simplex Revisado, resolviendo eficientemente los sub-problemas que alimentan al master con los puntos extremos que se obtienen en cada iteración asociada a la solución de los sub-problemas y, a la vez, evaluando la optimalidad del master. El criterio de parada de este método es cuando no haya opción de mejora, es decir todos los sub-problemas dan como solución óptima 0.

## 4.2. PROGRAMACIÓN ENTERA

Un problema de programación matemática de manera general se puede definir así<sup>28</sup>

$$\text{Max o Min } f(x) \quad (40)$$

---

<sup>28</sup> GARFINKEL, Robert S. NEMHAUSER, G. L. Óp cit. p.2

$$x \in X \subseteq R^n \tag{41}$$

La función objetivo (40) se expresa de manera general como una función de  $x$  sin importar el tipo u orden. El conjunto  $X$  (41) se conoce también como dominio factible y está contenido en un espacio n-dimensional de los reales.

Los problemas de programación entera tienen la característica<sup>29</sup>

$$X \subseteq Z^n \subset R^n \tag{42}$$

La proposición (42) señala que el dominio factible sólo se encontrará definido en un campo n-dimensional de los enteros<sup>30</sup>.

El dominio factible se define de la siguiente manera<sup>31</sup>

$$X = \{x | Ax \leq b, x \geq 0 \text{ enteras}\} \tag{43}$$

La función objetivo se define como<sup>32</sup>

$$\text{Max o Min } z = cx \tag{44}$$

Muchos de los problemas de decisión, como el problema de localización de centros de distribución, el problema de secuenciación de tareas, el problema de asignación generalizada o el problema de ruteo de vehículos, se formulan como un problema de programación entera. En el presente trabajo se discute el VRP o Problema de ruteo de vehículos (por sus siglas en inglés).

---

<sup>29</sup> Ibíd. p.3

<sup>30</sup> Ibíd. p.3

<sup>31</sup> Ibíd. p.5

<sup>32</sup> Ibíd. p.5

**4.2.1 Métodos exactos para problemas de programación lineal entera.** Dentro de los métodos exactos de programación lineal es usual hacer uso de “relajaciones”. Una relajación es una modificación de la formulación del problema estudiado con el fin de reducir su complejidad disminuyendo ligeramente la calidad de la solución.

La solución óptima de un problema de programación lineal entera se halla mediante métodos exactos y, como es natural, se basa en los de algoritmos de problemas de programación lineal existentes para su solución. Entre los métodos exactos se encuentran el Branch and Bound y el Branch and Cut, de éstos se derivan entre otros el Branch Cut and Pricing y el Robust Branch Cut and Pricing, los cuales se revisan a continuación.

**4.2.1.1 Branch and Bound (BB).** Branch and Bound (BB) es un método exacto para la solución de problemas de programación matemática lineal entera que se fundamenta en la enumeración de posibilidades. Normalmente descrita por un árbol, la enumeración va ligada al cálculo de límites inferiores y superiores. De forma general, para un problema dado se calcula de la siguiente forma.

**Limite superior**<sup>33</sup>

Para el problema

$$\max z(x), \quad x \in X_j \tag{45}$$

$$\text{Sea } z_j^* = \begin{cases} z(x^*(j)) & \text{si } x_j^* \text{ es solución de (45)} \\ -\infty & \text{si } X_j = \emptyset \\ \infty & \text{si (45) es irrestricto} \end{cases}$$

---

<sup>33</sup>GARFINKEL, Robert S. NEMHAUSER, G. L. (1972). Integer Programming (I., p. 427). United States of America: JOHN WILEY & SONS. p.111

El límite superior se puede calcular usando una relajación de (45)

$$\max z(x), \quad x \in T_j \supseteq X_j \quad (46)$$

$$\text{Sea } \bar{z}_j \begin{cases} z_j^o(x^o(j)) & \text{si } x_j^o \text{ es solución de (46)} \\ -\infty & \text{si } T_j = \emptyset \\ \infty & \text{si (46) es irrestricto} \end{cases}$$

Elegir el conjunto  $T_j$  es muy importante para que sea relativamente fácil encontrar una solución, aunque éste no debe ser muy pequeño para que sea equivalente a  $X_j$ . Cuando se habla de BB, se refiere a enumeraciones seguidas por un árbol, siendo cada iteración un vértice de este árbol. Para un vértice  $v_k$  que es predecesor de  $v_j$ , se cumple la condición de  $T_j \supseteq X_j \supseteq X_k$ , en particular para problemas de programación lineal entera  $T_j$  y  $X_j$  se definen

$$X_j = \{x | A^j x = b^j, x \geq 0 \text{ enteras}\} \quad (47)$$

$$T_j = \{x | A^j x = b^j, x \geq 0\} \quad (48)$$

Los supraindices indican la iteración o el vértice del árbol a que corresponden los conjuntos.

### Límite Inferior<sup>34</sup>

Un límite inferior  $\underline{z}_j$  satisface  $\underline{z}_j \leq z_j^*$ . Una forma de calcular un límite inferior es encontrando algún  $x' \in X_j$  que sea  $\underline{z}_j = z(x')$ . Si  $v_k$  es predecesor de  $v_j$ ,  $\underline{z}_j \leq z_k^*$ , se proporciona un importante resultado como  $\underline{z}_j \leq z_0^*$ , ya que entre más se descienda del árbol, se asegura que no se pueden obtener soluciones mayores a las anteriores.

---

<sup>34</sup> GARFINKEL, Robert S. NEMHAUSER, G. L. Óp cit. p.112

### Evaluación por límites<sup>35</sup>

Un vértice es evaluado si

$$\text{a) } \bar{z}_j = \underline{z}_j$$

$$\text{b) } \bar{z}_j \leq \underline{z}_0$$

Para el caso (a), no hay una mejor solución para (45); cuando el caso (b) ocurre, ningún sucesor de  $v_j$  puede producir una solución que mejore la óptima actual de (46). El caso (b) se presenta si  $T_j = \emptyset$  ya que  $\bar{z}_j = -\infty$ .

Con las anteriores definiciones, un algoritmo de BB se inicia en el origen con  $v_0$ . Seguidamente, se establecen los límites y se evalúa el vértice por límites, evaluando optimalidad y la condición entera de la solución; si ésta se cumple, termina el algoritmo y, si no, se ramifica e itera nuevamente<sup>36</sup>.

**4.2.1.2 Branch and Cut (BC).** Branch and Cut (BC) es un enfoque que plantea la idea de resolver un problema de programación lineal entera como una sucesión de diferentes problemas de programación lineal, mediante la aplicación de cortes al dominio factible hasta encontrar soluciones factibles enteras, lo anterior se representa de la siguiente manera<sup>37</sup>

Se considera un ILP de la forma

$$\max cx, \quad x \in X = \{x | Ax = b, x \geq 0 \text{ enteras}\} \quad (49)$$

---

<sup>35</sup> Ibíd. p.112

<sup>36</sup> Ibíd. p.113

<sup>37</sup> GARFINKEL, Robert S. NEMHAUSER, Óp cit. p.155

Existen  $\bar{A}$  y  $\bar{b}$  tal que  $T = \{x | \bar{A}x = \bar{b}, x \geq 0\}$  y  $X \subseteq T$ , de esta manera una relajación de (49) es

$$\max cx, \quad x \in T \tag{50}$$

Entonces la solución entera de (50) es la solución del sistema (49).

### **Idea fundamental de corte**<sup>38</sup>

Un corte es una restricción que garantiza que su frontera esté formada por al menos una solución factible en el vértice, es decir tiene una coordenada entera. Una definición formal es la que se presenta a continuación.

$$\max cx, \quad x \in X = \{x | Ax = b, x \geq 0\} \tag{51}$$

Suponiendo que una representación de las variables correspondientes a una base de simplex de (51) esta dada por

$$x_{BVi} = y_{i0} - \sum_{j \in R} y_{ij} x_j \quad \forall i \tag{52}$$

Al multiplicar por un número  $h \neq 0$  se tiene que

$$hx_{BVi} + \sum_{j \in R} hy_{ij} x_j = hy_{i0} \quad \forall i \tag{53}$$

Para generar una solución entera se requiere que  $x \geq 0$ , implicando

$$[h]x_{BVi} + \sum_{j \in R} [hy_{ij}]x_j \leq hy_{i0} \quad \forall i \tag{54}$$

---

<sup>38</sup> Ibíd. p.158

Para una solución entera se debe cumplir que el lado derecho de (54) sea entero; si se multiplica (52) por  $h$  y se resta de (54), se tiene la definición matemática de corte.

$$\sum_{j \in R} ([h]y_{ij} - [hy_{ij}])x_j \geq [h]y_{i0} - [hy_{i0}] \quad (55)$$

**4.2.1.3 Branch Cut and Pricing (BCP).** El término Branch Cut and Pricing se refiere a la combinación de dos técnicas: Descomposición de Dantzig Wolfe y relajación de Branch and Cut. Actualmente, se ha desarrollado una versión robusta de este enfoque para la solución de problemas de programación entera mixta con estructura escalonada.

Suponiendo que se tiene un problema de programación lineal entera como

$$Z_{IP} = \min cx \quad (56)$$

Sujeto a

$$Ax = b \quad (57)$$

$$Dx \leq d \quad (58)$$

$$x \in Z_+^n \quad (59)$$

La ecuación (56) representa la función objetivo; el término  $Z_{IP}$  se refiere a que el problema está restringido a soluciones con variables de decisión enteras. Las ecuaciones (57) y (58) representan el conjunto de restricciones, pero es importante aclarar que el conjunto de restricciones de (58) están asociadas a una estructura de bloque. La ecuación (59) especifica que el conjunto de variables de decisión está contenido en un campo numérico de los enteros positivos.

Para el desarrollo de la técnica, se define una matriz  $Q = \{x \in Z_+^n | Dx \leq d\}$ , teniendo una cantidad finita de columnas  $x^1, \dots, x^p$ , cada una de ellas de dimensión  $n$  que

es la cantidad de variables de decisión, siendo así  $Q$  una matriz  $[n \times p]$  que corresponde a un conjunto de soluciones factibles. Ahora, el conjunto de variables  $x$  se puede representar de la siguiente manera

$$x = Q\lambda \quad (60)$$

Sujeto a:

$$1\lambda = 1 \quad (61)$$

$$\lambda \in \{0,1\}^p \quad (62)$$

Con los arreglos (56)-(62), se expresa el problema maestro para la descomposición de Dantzig Wolfe de la siguiente forma

$$Z_{DWM} = \min(c Q) \lambda \quad (63)$$

Sujeto a:

$$(A Q)\lambda = b \quad (64)$$

$$1\lambda = 1 \quad (65)$$

$$\lambda \geq 0 \quad (66)$$

Con lo anterior, se asegura una formulación finita del problema; el conjunto  $Q$  se define de tal forma que sea finito., sin embargo, debido al gran tamaño de este arreglo, es necesario el uso de la generación de columnas y mediante una serie de subproblemas que se resuelven de manera eficiente, se va identificando una posible mejora en soluciones básicas adyacentes sin necesidad de determinar  $Q$  completamente o totalmente.

$$u(\mu, v) = \min(c - \mu A)x - v \quad (67)$$

Sujeto a:

$$Dx \leq d \tag{68}$$

$$x \in Z_+^n \tag{69}$$

$\mu$  y  $\nu$  son variables duales asociadas a las filas de las restricciones del problema maestro de la descomposición de Dantzig Wolfe (63)–(66). Por otro lado, este esquema es aplicable en la práctica para una estructura la (68) adecuada para una descomposición.

Suponiendo que  $\bar{\lambda}$  es una solución parcial del problema maestro y si se define  $\bar{x} = Q\bar{\lambda}$  como una solución parcial del problema de estudio, se tiene un corte de la forma  $a^i x \leq b_i$  tal que  $a^i \bar{x} > b_i$ , entonces una restricción de la forma  $(a^i Q)\lambda \leq b_i$  puede ser agregada al problema maestro. Esto proporciona que para el subproblema o pricing sólo cambie  $A$  incluyendo una fila extra a las previas del corte, lo cual no garantiza conservar la complejidad del pricing; a esto se le conoce en la literatura como Branch Cut and Pricing (BCP).

El desarrollo de un enfoque BCP ofrece beneficios sobre otros en cuanto a la formulación del máster, ya que es un problema de menor tamaño comparado con el problema original y así hay menos posibles cortes para agregar, lo cual representa una ventaja para la solución de problemas de gran tamaño. Como es un enfoque de métodos exactos y los tiempos siguen siendo altos, se busca relajar la formulación aún más, con el fin de disminuir el tiempo de ejecución sacrificando la calidad de la solución. En primer lugar, se reformula el problema maestro de la siguiente forma

$$Z_{EM} = \min cx \tag{70}$$

Sujeto a

$$Q\lambda - x = 0 \tag{71}$$

$$1\lambda = 1 \tag{72}$$

$$Ax = b \tag{73}$$

$$\lambda, x \geq 0 \tag{74}$$

Esta reformulación del máster se conoce en la literatura como “Explicit Master”. Es fácil notar que el tamaño de la formulación es mayor que el maestro del BCP, pero presenta la ventaja de ser un problema que encuentra los valores  $\lambda$  y  $x$  en un sólo programa, a diferencia del máster de Dantzig Wolfe.

Los sub-problemas se relajan al igual que el master, con el objetivo de no cambiarlos de tamaño a lo largo de cada iteración o al agregar más cortes. Es necesario definir las variables  $\pi, \nu$  y  $\mu$  correspondientes a las restricciones (71), (72) y (73), respectivamente.

$$u(\pi, \nu) = \min -\pi x - \nu \tag{75}$$

Sujeto a

$$Dx \leq d \tag{76}$$

$$x \in Z_+^n \tag{77}$$

En este caso, esta versión de sub-problemas no dependen de  $A$ , lo que asegura que su complejidad a lo largo de cada iteración o cada corte sea la misma. Lo anterior está sujeto a una relajación que sugiere que los coeficientes reducidos sean directamente proporcionales a la combinación de las variables duales de (75).

### 4.3. TEORÍA DE GRAFOS

Dentro del campo de investigación de operaciones, en los últimos años se ha profundizado el estudio de problemas mediante el uso de redes o grafos que permitan una representación del problema usando un conjunto de nodos

relacionados a través de arcos, líneas o aristas. Obligando a estos problemas a cumplir con las ecuaciones de Kirchhoff (conservación de flujo y equilibrio nodal) reduciendo su complejidad al momento de modelar este tipo de problemas mediante el uso de sistemas de ecuaciones o un sistema de programación lineal.

Para la notación de una red, considérese un grafo  $G \{V,A\}$  con un conjunto de nodos o vértices  $V \{1,2,\dots,m\}$  y un grupo de aristas  $A \{1,2,\dots,n\}$ . Un arco dirigido define la dupla  $(i,j)$  de nodos, limitando el sentido en el cual este arco puede ser cruzado. De manera análoga, un arco que carece de orientación se representa por las duplas  $(i,j)$  y  $(j,i)$ , lo que indica que no hay restricciones de sentido. En la mayoría de problemas modelados por redes, se puede encontrar que en un grafo no dirigido, el esfuerzo o costo asociado para recorrer un arco en un sentido es igual al del sentido contrario.

En el estudio de redes, es importante destacar conceptos como trayectoria (de un nodo  $i_o$  a  $i_p$ ), es decir una sucesión de arcos de la forma  $P = \{(i_o, i_1), (i_1, i_2), \dots, (i_{p-1}, i_p)\}$ , en la cual se cumple que el vértice inicial de cada arco es el final del arco predecesor en la sucesión. Una cadena tiene una estructura parecida a la de una trayectoria, pero se diferencia en que la cadena une los nodos sin ser necesario conservar el sentido del nodo  $i_o$  a  $i_p$ . Un circuito es una trayectoria con un arco adicional de  $(i_p, i_o)$ , lo que crea una trayectoria cerrada. Un ciclo es una cadena cerrada; cumpliendo que todo circuito es un ciclo, pero un ciclo no es necesariamente un circuito<sup>39</sup>.

En el estudio de redes se caracterizan dos tipos de redes: conexa y no conexa. Una red conexa es aquella en la cual se puede establecer como mínimo una cadena entre cualquier par de nodos que pertenezcan a la red; en la no conexa, no se puede asegurar una cadena o trayectoria en cualquier par de vértices. Las

---

<sup>39</sup>BAZARAA, Mokhtar S., JARVIS, John J. and SHERALI, Hanif D. Programación Lineal y Flujo de Redes. Segunda Ed. Mexico, 1998. ISBN 968-18-4867-5.. p.490

definiciones anteriores ayudan en la definición del concepto de árbol como un grafo conexo sin ciclos; de vital importancia para los problemas de transporte, ya que tienen características que facilitan el desarrollo de soluciones en base a estos tal como la propiedad de no tener ciclos ya que estos ciclos en transporte involucran un mayor valor a los costos de transporte.

#### **4.4. PROBLEMA DE RUTEO DE VEHÍCULOS VRP**

El problema de ruteo de vehículos o VRP (por sus siglas en inglés) es un grupo de modelos matemáticos que tienen como objetivo encontrar el conjunto de rutas que minimicen el costo de transporte (tiempo de viaje, distancia recorrida y demás) para un número de vehículos que transportan bienes desde uno o varios depósitos hacia un grupo determinado de clientes a servir<sup>40</sup>.

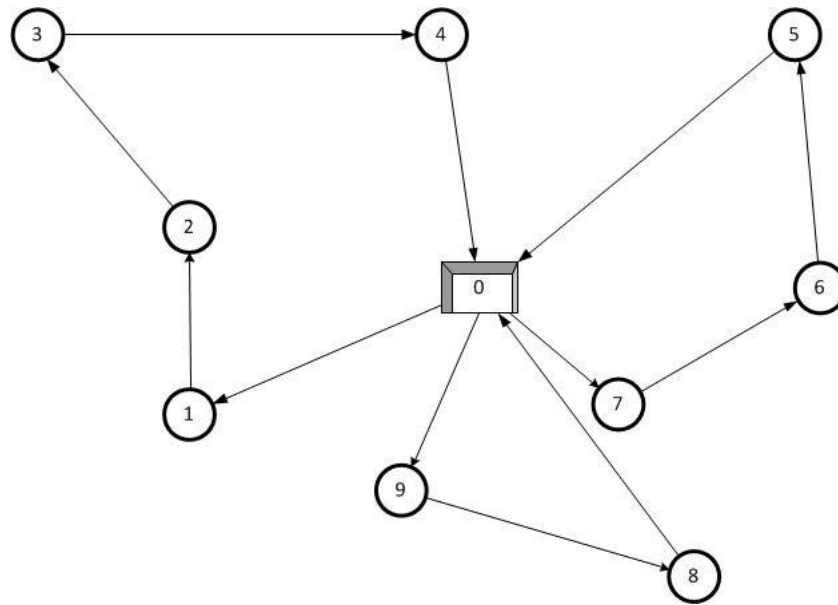
La representación del VRP se realiza mediante un grafo (Figura 2). Los clientes se presentan como vértices, los arcos—dirigidos o no dirigidos representan los posibles caminos entre dos clientes que deben ser visitados. Los arcos dirigidos representan caminos que sólo pueden transitarse de acuerdo a un único sentido; los no dirigidos, no restringen el sentido, éstos pueden transitarse en cualquier dirección<sup>41</sup>.

Figura 2. Representación mediante un grafo del Problema de Transporte

---

<sup>40</sup> TOTH, Paolo and VIGO, Daniele (eds.), Op cit., p.2

<sup>41</sup> Ibíd.



Características típicas de los clientes<sup>42</sup>(Toth, Vigo 2002, p. 4):

- Localización del vértice en la red de caminos del grafo.
- Cantidad demandada de bienes.
- Periodos de tiempo en el día en los cuales puede ser atendido, se conocen como Ventanas de tiempo o Time Windows.
- Tiempos de despacho o recolección una vez se haya alcanzado la localización del mismo.
- Subconjunto de vehículos que pueden servir a determinado cliente.

Normalmente es importante considerar algunas características especiales a los vehículos involucrados en este tipo de problemas, aumentando su realismo y generando mayor versatilidad en su aplicación. Las siguientes propiedades se han logrado identificar a lo largo de su estudio<sup>43</sup>:

- **Depósito de destino:** usualmente puede requerirse que el depósito de origen sea diferente al de destino.

---

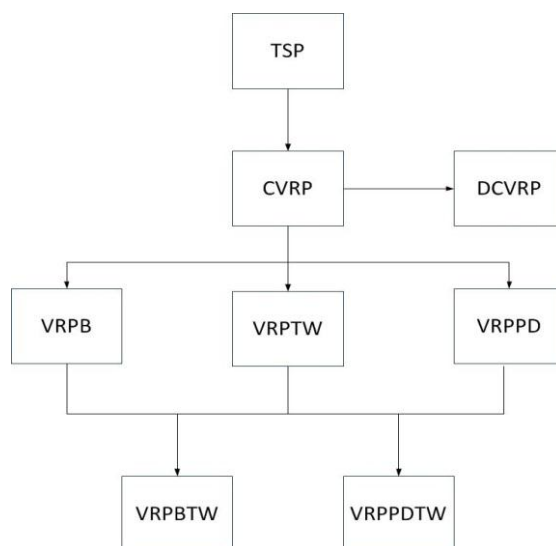
<sup>42</sup> Ibíd., p.4

<sup>43</sup> Ibíd., p.5

- **Capacidad del vehículo:** el parque automotor puede tener una capacidad igual para todos los vehículos, a esto se le llama flete homogéneo. Cuando hay variaciones de capacidad entre los camiones, recibe el nombre de flete heterogéneo.
- **Posibles compartimientos o divisiones:** en ellas sólo se puede transportar un tipo exclusivo de bien.
- **Posibilidades de carga y descarga:** las cuales pueden ser independientes o darse de forma simultánea.
- **Determinado subconjunto de caminos:** por los cuales los camiones pueden transitar.
- **Costo asociado por la utilización del vehículo:** el cual puede estar dado por unidad de distancia, por unidad de tiempo o por ruta seleccionada, entre otras.

Las combinaciones de las características mencionadas y algunas distinciones o particularidades de los elementos matemáticos del problema de ruteo de vehículos han generado variantes reconocidas por la comunidad científica, en la Figura 3 se presentan algunas de estas variantes ya reconocidas en la literatura.

Figura 3. Variantes de los Problemas de Transporte



## 4.5. PROBLEMA DE RUTEO DE VEHÍCULOS CON RESTRICCIONES DE CAPACIDAD (CVRP)

Esta variante conocida como problema de ruteo de vehículos con restricciones de capacidad CVRP consiste en obtener el conjunto de rutas de un grupo de camiones que tienen un límite de capacidad (galones, unidades, peso, etc.) para transportar desde un depósito hacia un número determinado de clientes.

### Formulación matemática<sup>44</sup>

En la literatura se han presentado varias formulaciones del CVRP, pero la más extensa y completa es la presentada a continuación; otras formulaciones usualmente se basan en ésta, con algunas suposiciones para disminuir su dificultad.

### VARIABLES DE DECISIÓN

$x_{ijk}$  es una variable binaria que acumula el número de veces que el arco (i,j) es atravesado por el camión k.

$y_{ik}$  variable binaria que toma el valor de 1 si el cliente i ha sido visitado por el camión k.

### Parámetros

$c_{ij}$  costo de atravesar el arco (i,j).

$C$  capacidad de los vehículos utilizados.

$d_i$  demanda del cliente i.

$n$  número de clientes a servir.

### Función Objetivo

$$\text{Min } Z = \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \quad (78)$$

---

<sup>44</sup> TOTH Y VIGO. Óp. Cit., p.5.

## Restricciones

$$\sum_{k=1}^K y_{ik} = 1 \forall i \in V \setminus \{0\} \quad (79)$$

$$\sum_{k=1}^K y_{0k} = K \quad (80)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \forall i \in V, k = 1, \dots, K \quad (81)$$

$$\sum_{i \in V} d_i y_{ik} \leq C \quad (82)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk} \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, \dots, K \quad (83)$$

$$y_{ik} \in \{0,1\} \forall i \in V, k = 1, \dots, K \quad (84)$$

$$x_{ijk} \in \{0,1\} \forall i, j \in V, k = 1, \dots, K \quad (85)$$

La función objetivo del problema tiene como propósito disminuir los costos totales de transporte en los que se incurre al servir un número  $n$  de clientes (78). Como se mencionó antes, las restricciones delimitan el espacio de solución: la restricción (79) asegura que cada cliente sea visitado por un solo camión, la (80) garantiza que del depósito cada camión salga una sola vez, la (81) actualiza el valor de la variable  $y_{ik}$  cuando el arco anterior o posterior del cliente  $i$  es atravesado, la (82) limita la capacidad de los camiones para que no excedan la suma de las demandas de los clientes que visitará cada camión, la (83) también conocida como restricción de subtours o de ciclos hamiltonianos evita que se generen ciclos en la solución, las restricciones (84) y (85) sirven para especificar que las variables  $x_{ijk}$  y  $y_{ik}$  son de naturaleza binaria ya que sólo pueden tomar los valores de 0 ó 1.

## 5. ESTADO DEL ARTE

### 5.1. PROBLEMAS CON ESTRUCTURA DE ESCALERA (STAIRCASE LINEAR PROGRAMS, SLPS)

Ho y Loute (1980)<sup>45</sup> destacan el estudio de problemas de programación matemática con estructuras de escalera o SLP's y comparan dos enfoques: la factorización de la base de un Simplex y la descomposición de Dantzig Wolfe, resaltando las bondades sobre otros métodos exactos y destacando la oportunidad para avanzar en este campo.

El problema de descomposición como tal fue definido formalmente para SLPs por Ho (1977)<sup>46</sup>, después tuvo aplicaciones en problemas del sector financiero como enseña Fourer (1982)<sup>47</sup>, en problemas de asignación de tráfico por Merchant y Nemhauser (1978)<sup>48</sup> y en la programación de producción por Glassey (1971)<sup>49</sup>, entre otros.

### 5.2. PROBLEMA DE RUTEO DE VEHICULOS (CVRP) Y MÉTODOS EXACTOS

---

<sup>45</sup>HO, James K. and LOUTE, Etienne. A Comparative Study of Two Methods for Staircase Linear Programs. *ACM Transactions on Mathematical Software* [online]. 1 March 1980. Vol. 6, no. 1, [Accessed 2 April 2014]. p. 17–30. Available from: <http://dl.acm.org/citation.cfm?id=355873.355875>. DOI 10.1145/355873.355875.

<sup>46</sup>HO, James K. Note—Nested Decomposition of a Dynamic Energy Model. *Management Science* [online]. 1 May 1977. Vol. 23, no. 9, [Accessed 3 April 2014]. p. 1022–1026. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.23.9.1022>. DOI 10.1287/mnsc.23.9.1022.

<sup>47</sup>FOURER, Robert. Solving staircase linear programs by the simplex method, 1: Inversion. *Mathematical Programming* [online]. December 1982. Vol. 23, no. 1, [Accessed 2 April 2014]. p. 274–313. Available from: <http://link.springer.com/10.1007/BF01583795>. DOI 10.1007/BF01583795.

<sup>48</sup>MERCHANT, Deepak K. and NEMHAUSER, George L., A Model and an Algorithm for the Dynamic Traffic Assignment Problems. *Transportation Science* [online]. 1 August 1978. Vol. 12, no. 3, p. 183–199.

<sup>49</sup>GLASSEY, C. Roger, Dynamic Linear Programs for Production Scheduling. *Operations Research* [online]. 1 February 1971. Vol. 19, no. 1, p. 45–56.

Dentro de los problemas de optimización combinatoria, se encuentran los problemas de ruteo de vehículos VRP, caracterizados por su complejidad NP-hard y sus aplicaciones prácticas en la industria logística. Una variante de este grupo de problemas es el Problema de Ruteo de Vehículos con restricciones de Capacidad o CVRP (1959)<sup>50</sup>. El CVRP se formula mediante programación lineal entera mixta y la solución se determina en instancias relativamente no grandes por métodos exactos.

El siguiente evento significativo para encontrar una solución al CVRP lo propusieron Clarke y Wright (1964)<sup>51</sup>, quienes formulan un algoritmo iterativo para encontrar una solución óptima o subóptima en el caso de instancias grandes. En los años posteriores, se usaron cuatro enfoques para el desarrollo de métodos o técnicas de solución del CVRP, los cuales se presentan a continuación.

**3.2.1 Branch and Bound (BB).** El término de Branch and Bound fue acuñado por Little, Murty, Sweeney, Karel (1963)<sup>52</sup>, éste es considerado como un método de enumeración en el cual se definen límites para encontrar la solución de cada ramificación o branch. Sus aplicaciones computacionales futuras sólo fueron mejoradas gracias a principios identificados en los trabajos de Bertier, Roy (1964)<sup>53</sup>; Lawler, Wood (1966)<sup>54</sup>; Agin (1966)<sup>55</sup>. La primera aplicación de un

---

<sup>50</sup> DANTZIG, G. B. and RAMSER, J. H. The Truck Dispatching Problem. Management Science [online]. 1 October 1959. Vol. 6, no. 1, [Cited 16 January 2014]. p. 80–91. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.6.1.80>, DOI 10.1287/mnsc.6.1.80.

<sup>51</sup> CLARKE, G. and WRIGHT, J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research [online]. 1 August 1964. Vol. 12, no. 4, [Cited 26 January 2014]. p. 568–581. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.12.4.568>, DOI 10.1287/opre.12.4.568.

<sup>52</sup> LITTLE, John D. C., MURTY, Katta G., SWEENEY, Dura W. and KAREL, Caroline. An Algorithm for the Traveling Salesman Problem. Operations Research [online]. 1 December 1963. Vol. 11, no. 6, [Cited 23 March 2014]. p. 972–989. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.11.6.972A>. DOI 10.1287/opre.11.6.972.

<sup>53</sup> BERTIER, P. and ROY, B. Procédure de Résolution pour une Classe de Problèmes Pouvant Avoir un Caractère Combinatoire. In : Cahiers Cent. d'Etudes Recherche Operationelle 6. 1964. p. 202–208.

método exacto en un problema de transporte bajo el enfoque de Branch and Bound fue Gavish, Graves (1978)<sup>56</sup> para el TSP, y el CVRP fue realizado por Laporte, Mercure y Nobert (1986)<sup>57</sup>. De manera general, el método consistía en resolver K-TSP's, cada uno con restricciones de capacidad para ajustarlo al CVRP. El siguiente en abordar este problema mediante BB fue Fischetti, Toth y Vigo (1994)<sup>58</sup>, con un caso particular de CVRP donde se considera la matriz de costos asimétrica (ACVRP). En ese mismo año, Fisher (1994)<sup>59</sup> resolvió el problema con instancias de 100 clientes, usando K problemas de árboles de expansión mínima. Posteriormente, se aplicó de forma similar por Miller (1995)<sup>60</sup>, y Hadjiconstantinou, Christofides y Mingozzi (1995)<sup>61</sup>, el enfoque de BB combinado con el enfoque de Branch and Price o descomposición de Dantzig Wolfe como lo realizó Kohl, Desrosiers, Madsen, Solomon, Soumis (1999)<sup>62</sup>.

**3.2.2 Branch and Cut (BC).** Un trabajo seminal de Branch and Cut es el de Dantzig y Fulkerson (1959)<sup>63</sup>, se aplicó en un TSP de bajo tamaño para ilustrar su procedimiento, pero no se evidenciaba si era o no finito, esto sólo se pudo

---

<sup>54</sup> LAWLER, E. L. and WOOD, D. E., Branch and Bound Methods: A Survey. Operational Research [online]. 1966. Vol. 14, no. 4, p. 699–719.

<sup>55</sup> AGIN, Norman, Optimum Seeking with Branch and Bound. Management Science [online]. 1966. Vol. 13, no. 4, p. B176–B185.

<sup>56</sup> GAVISH, Bezalel and GRAVES, Stephen C. The Travelling Salesman Problem and Related Problems [online]. 1 July 1978. Massachusetts Institute of Technology, Operations Research Center.

<sup>57</sup> LAPORTE, Gilbert, MERCURE, Hélène and NOBERT, Yves, An exact algorithm for the asymmetrical capacitated vehicle routing problem. Networks [online]. 1986. Vol. 16, no. 1, p. 33–46.

<sup>58</sup> FISCHETTI, Matteo, TOTH, Paolo and VIGO, Daniele, A Branch-and-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs. [online]. 1 October 1994.

<sup>59</sup> FISHER, Marshall L. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. [online]. 1 August 1994.

<sup>60</sup> MILLER, D L, A MATCHING BASED EXACT ALGORITHM FOR CAPACITATED VEHICLE ROUTING PROBLEMS. ORSA journal on computing. Vol. 7, no. 1 [online]. 1995.

<sup>61</sup> HADJICONSTANTINO, Eleni, CHRISTOFIDES, Nicos and MINGOZZI, Aristide, A new exact algorithm for the vehicle routing problem based on  $\alpha$ -paths and  $k$ -shortest paths relaxations. Annals of Operations Research [online]. December 1995. Vol. 61, no. 1, p. 21–43.

<sup>62</sup> KOHL, Niklas, DESROSIERS, Jacques, MADSEN, Ole B. G., SOLOMON, Marius M. and SOUMIS, François, 2-Path Cuts for the Vehicle Routing Problem with Time Windows. [online]. 1 February 1999.

<sup>63</sup> KALVELANGEN, Erwin, Branch-and-Bound Methods for an MINLP model with semi-continuous variables [online]. 2003.

encontrar hasta el trabajo de uno de los mayores exponentes de este enfoque Gomory y Baumol (1960)<sup>64</sup> proponiendo un nuevo método que aseguraba un número finito de iteraciones. Se han realizado estudios bajo otras perspectivas, como el estudio de poliedros nuevos de Jeroslow (1971)<sup>65</sup>; Rubin (1970)<sup>66</sup> o el de análisis de convergencia de Gomory, Hoffman, Watson (1963)<sup>67</sup>.

Algunas relajaciones del enfoque buscan la eliminación de restricciones importantes en los problemas de transporte, tal como lo hizo Miller, Tucker y Zemlin (1960)<sup>68</sup> para el TSP con las restricciones de subtours; esta técnica fue aplicada luego al CVRP por Kulkarni y Bhave (1985)<sup>69</sup>. Por los resultados obtenidos a lo largo de las aplicaciones encontradas en la literatura, se considera esta técnica mejor que el enfoque de BB<sup>70</sup>.

En los últimos 15 años, se ha aplicado este enfoque al CVRP y a algunas subvariantes, como el simétrico por Ulrich Blasum (2002)<sup>71</sup> y CVRP's de multicomienzo o de multicomienzo parcial Letchford, Eglese, Lysgaard (2002)<sup>72</sup>. Otras variantes del método son los planos de Gomory entero-mixto Lysgaard,

---

<sup>64</sup> GOMORY, Ralph E. and BAUMOL, William J, Integer Programming and Pricing. *Econometrica* [online]. 1960. Vol. 28, no. 3, p. 521–550.

<sup>65</sup> JEROSLOW, R. G., Comments on Integer Hulls of Two Linear Constraints. *Operations Research* [online]. 1971. Vol. 19, no. 4, p. 1061–1069.

<sup>66</sup> RUBIN, David S., On the Unlimited Number of Faces in Integer Hulls of Linear Programs with a Single Constraint. *Operations Research* [online]. 1970. Vol. 18, no. 5, p. 940–946.

<sup>67</sup> GOMORY, R. E., HOFFMAN, A. J. and WATSON, Thomas J., On the convergence of an integer-programming process. *Naval Research Logistics Quarterly* [online]. March 1963. Vol. 10, no. 1, p. 121–123.

<sup>68</sup> MILLER, C. E., TUCKER, A. W. and ZEMLIN, R. A., Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*. 1 October 1960. Vol. 7, no. 4, p. 326–329.

<sup>69</sup> KULKARNI, R.V. and BHAVE, P.R., Integer programming formulations of vehicle routing problems. *European Journal of Operational Research* [online]. 1985. Vol. 20, no. 1, p. 58–67.

<sup>70</sup> MARTÍ, Rafael and REINELT, Gerhard, *The Linear Ordering Problem*. Berlin, Heidelberg : Springer Berlin Heidelberg [online]. 2011

<sup>71</sup> ULRICH BLASUM, Winfried Hochstättler, 2002, Application of the branch and cut method to the vehicle routing problem. [online]. 2002.

<sup>72</sup> LETCHFORD, Adam N., EGGLESE, Richard W. and LYSGAARD, Jens, Multistars, partial multistars and the capacitated vehicle routing problem. *Mathematical Programming* [online]. 1 December 2002. Vol. 94, no. 1, p. 21–40.

Letchford y Eglese (2004)<sup>73</sup>, las Relajaciones de límites inferiores por Jepsen, Spoorendonk y Ropke (2013)<sup>74</sup> y la combinación con métodos heurísticos para mejorar el tiempo computacional sacrificando la optimalidad de la solución por Letchford, Lysgaard y Eglese (2006)<sup>75</sup>. Finalmente, se han utilizado herramientas físicas para mejorar los tiempos, como la computación en paralelo para simular mayor tiempo computacional en un menor tiempo real de Ralphs (2003)<sup>76</sup>.

**3.2.3 Branch and Price (BP).** El enfoque de Branch and Price es el conjunto de aplicaciones en la literatura del método de descomposición de Dantzig Wolfe presentado inicialmente en 1960<sup>77</sup>, el cual consiste en transformar el problema de estudio en un problema maestro (Máster Problem) y varios sub-problemas (Pricing Problems), si y sólo si la matriz de restricciones A presenta una estructura de bloque y el espacio de solución es un poliedro convexo.

Las aplicaciones de la descomposición de Dantzig Wolfe y sus variantes han tenido lugar en problemas de localización por Tang, Wang, Liu (2007)<sup>78</sup>, en áreas de transporte como el de Bin Packing Problem o el de Puchinger, Raidl (2007)<sup>79</sup> y

---

<sup>73</sup> LYSGAARD, Jens, LETCHFORD, Adam N. and EGGLESE, Richard W., A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* [online]. 1 June 2004. Vol. 100, no. 2, p. 423–445.

<sup>74</sup> JEPSEN, Mads, SPOORENDONK, Simon and ROPKE, Stefan, A Branch-and-Cut Algorithm for the Symmetric Two-Echelon Capacitated Vehicle Routing Problem. *Transportation Science* [online]. February 2013. Vol. 47, no. 1, p. 23–37.

<sup>75</sup> LETCHFORD, A N, LYSGAARD, J and EGGLESE, R W, A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society* [online]. 13 December 2006. Vol. 58, no. 12, p. 1642–1651.

<sup>76</sup> RALPHS, T.K., Parallel branch and cut for capacitated vehicle routing. *Parallel Computing* [online]. 2003. Vol. 29, no. 5, p. 607–629.

<sup>77</sup> DANTZIG, George B. and WOLFE, Philip, Decomposition Principle for Linear Programs. *Operations Research* [online]. 1 February 1960. Vol. 8, no. 1, p. 101–111.

<sup>78</sup> TANG, Lixin, WANG, Gongshu and LIU, Jiying, A branch-and-price algorithm to solve the molten iron allocation problem in iron and steel industry. *Computers & Operations Research* [online]. October 2007. Vol. 34, no. 10, p. 3001–3015.

<sup>79</sup> PUCHINGER, Jakob and RAIDL, Günther R., Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research* [online]. December 2007. Vol. 183, no. 3, p. 1304–1327.

en el sector de hidrocarburos propuesto por Gunnerud, Foss (2010)<sup>80</sup>; Gunnerud, Foss, Torgnes (2010)<sup>81</sup>. Los trabajos anteriormente mencionados demuestran un firme interés en el desarrollo de este método debido a los resultados obtenidos.

En los problemas de transporte son pocas las aplicaciones de este enfoque, dado que su formulación requiere restricciones enteras en las variables. Estas aplicaciones normalmente tienen relajaciones o estrategias sencillas para obtener soluciones, como las de los trabajos de Christiansen y Lysgaard (2007)<sup>82</sup> para un problema de ruteo de vehículos capacitados con demanda estocástica, o las presentadas por Dell'Amico, Righini y Salani (2006)<sup>83</sup> en un modelo de entrega y recolección simultánea.

**3.2.4 Branch Cut and Pricing (BCP).** Los enfoques de BC y BP en los años 1980 y 1990 generaron buenos resultados, pero la idea que se aplicaran simultáneamente se consideraba impráctica por Barnhart, Johnson, Nemhauser, Savelsbergh y Vance (1998)<sup>84</sup>. Autores como Akker, Hurkens y Savelsbergh (1996)<sup>85</sup>, Kohl, Desrosiers, Madsen, Solomon y Soumis (1999)<sup>86</sup>, y Kim, Barnhart, Ware y Reinhardt (1999)<sup>87</sup> intentaron aplicar la combinación de estos dos

---

<sup>80</sup> GUNNERUD, Vidar and FOSS, Bjarne, Oil production optimization—A piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering* [online]. November 2010. Vol. 34, no. 11, p. 1803–1812.

<sup>81</sup> GUNNERUD, Vidar, FOSS, Bjarne and TORGNES, Erlend, Parallel Dantzig–Wolfe decomposition for real-time optimization—Applied to a complex oil field. *Journal of Process Control* [online]. October 2010. Vol. 20, no. 9, p. 1019–1026.

<sup>82</sup> CHRISTIANSEN, Christian H. and LYSGAARD, Jens, A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands [online]. 2007.

<sup>83</sup> DELL'AMICO, Mauro, RIGHINI, Giovanni and SALANI, Matteo, A Branch-and-Price Approach to the Vehicle Routing Problem with Simultaneous Distribution and Collection. [online]. 1 May 2006.

<sup>84</sup> BARNHART, Cynthia, JOHNSON, Ellis L., NEMHAUSER, George L., SAVELSBERGH, Martin W. P. and VANCE, Pamela H., Branch-and-Price: Column Generation for Solving Huge Integer Programs. [online]. 1 June 1998.

<sup>85</sup> AKKER, JM van den, HURKENS, CAJ and SAVELSBERGH, MWP, Time-Indexed Formulations for Machine Scheduling Problems: Column Generation. [online]. 1996.

<sup>86</sup> KOHL, Niklas, DESROSIERS, Jacques, MADSEN, Oli B. G., SOLOMON, Marius M. and SOUMIS, François, 2-Path Cuts for the Vehicle Routing Problem with Time Windows. [online]. 1 February 1999.

<sup>87</sup> KIM, Daeki, BARNHART, Cynthia, WARE, Keith and REINHARDT, Gregory, Multimodal Express Package Delivery: A Service Network Design Application. [online]. 1 November 1999.

enfoques, logrando identificar propiedades en la estructura de la descomposición que fue de utilidad para posteriores aplicaciones.

El enfoque de BCP evolucionó a un Robust Branch Cut and Pricing (RBCP) presentado por Marcus Poggi de Aragao (2003)<sup>88</sup>, el cual resolvía un problema de complejidad NP-hard en un tiempo computacional pseudo polinomial como por ejemplo el CARP por Bartolini, Cordeau, Laporte (2011)<sup>89</sup>; en el mismo año, fue aplicado al CVRP por Ricardo Fukasawa, Marcus Poggi De Aragão y Marcelo Reis (2003)<sup>90</sup>. Posteriormente, Fukasawa, Longo, Lysgaard, Aragão, Reis, Uchoa y Werneck (2005) prueban instancias hasta de 135 clientes para el CVRP y 100 clientes para el VRPTW<sup>91</sup>, obteniendo soluciones óptimas.

Actualmente, RBCP no sólo se ha aplicado al CVRP sino a otras variantes del VRP, implementando también recursos físicos para reducir el tiempo de procesamiento con el uso de la computación en paralelo<sup>92</sup>.

### **5.3. GAMS (GENERAL ALGEBRAIC MODELING SYMTEM)**

GAMS es una herramienta diseñada para modelar problemas grandes y complejos que permite al usuario concentrarse en el trabajo de modelado, puesto que se

---

<sup>88</sup> MARCUS POGGI DE ARAGAO, Eduardo Uchoa, Integer Program Reformulation for Robust Branch-and-Cut-and-Price Algorithms. In Proceedings of the Conference Mathematical Program in Rio: A Conference in Honour of Nelson Maculan [online]. 2003. P. 56–61 p.

<sup>89</sup> BARTOLINI, Enrico, CORDEAU, Jean-François and LAPORTE, Gilbert, Improved lower bounds and exact algorithm for the capacitated arc routing problem. Mathematical Programming [online]. 3 November 2011. Vol. 137, no. 1-2, p. 409–452.

<sup>90</sup> FUKASAWA, Ricardo, POGGI DE ARAGÃO, Marcus, REIS Marcelo, UCHOA, Eduardo, Robust branch-and-cut-and-price for the capacitated vehicle routing problem. IN PROCEEDINGS OF THE INTERNATIONAL NETWORK OPTIMIZATION CONFERENCE [online]. 2003. p. 231–236.

<sup>91</sup> FUKASAWA, Ricardo, LONGO, Humberto, LYSGAARD, Jens, ARAGÃO, Marcus Poggi de, REIS, Marcelo, UCHOA, Eduardo and WERNECK, Renato F., Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. Mathematical Programming [online]. 12 October 2005. Vol. 106, no. 3, p. 491–511.

<sup>92</sup> SUBRAMANIAN, Anand, UCHOA, Eduardo, PESSOA, ArturAlves and OCHI, Luiz Satoru, Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. Optimization Letters [online]. 21 October 2012. Vol. 7, no. 7, p. 1569–1581.

pueden hacer cambios en el método de solución o en el tipo de problema cambiando sólo una sentencia.

Un algoritmo desarrollado en GAMS es considerado superior a otros en lenguajes como C o Fortran, ya que el esfuerzo en su formulación e implementación en GAMS es más didáctica. Además, es más sencillo reescribir un algoritmo desarrollado en GAMS que en otro lenguaje para poder comercializarlo.

La utilización de GAMS para la aplicación y desarrollo de métodos exactos es sencilla pero a la vez robusta. Dentro de las aplicaciones de métodos exactos encontramos: Kalvelagen (2011)<sup>93</sup>, en el caso del método de generación de columnas para un problema de cutting stocks, aplicado junto con un enfoque de Branch and Cut por ser un problema de programación entera; Kalvelangen (2005)<sup>94</sup>, para el caso de la descomposición de Bender resolviendo un problema de asignación de transporte con carga fija de naturaleza determinística, su contraparte estocástica fue desarrollada también por Kalvelangen (2003)<sup>95</sup> en un problema de transporte con demandas estocásticas; aplicación de la Descomposición de Dantzig Wolfe por Çelebi, Fuller (2013)<sup>96</sup> a un problema de redes de energía, o a un problema de flujo multiproducto por Kalvelangen (2003)<sup>97</sup>.

Los algoritmos exactos desarrollados en GAMS no sólo se han restringido a enfoques de descomposición, también se encuentran enfoques de relajaciones

---

<sup>93</sup> KALVELAGEN, Erwin, Column Generation with GAMS [online]. 2011, Available from: <http://www.amsterdamoptimization.com/pdf/colgen.pdf>

<sup>94</sup> KALVELANGEN, Erwin, Bender Decomposition with GAMS [online]. 2005, Available from: <http://www.amsterdamoptimization.com/pdf/benders.pdf>

<sup>95</sup> KALVELANGEN, Erwin, Benders decomposition for stochastic programming with GAMS [online]. 2003, Available from: <http://www.amsterdamoptimization.com/pdf/stochbenders.pdf>

<sup>96</sup> ÇELEBI, Emre and FULLER, J. David, Máster problem approximations in Dantzig–Wolfe decomposition of variational inequality problems with applications to two energy market models. Computers & Operations Research [online]. November 2013. Vol. 40, no. 11, p. 2724–2739.

<sup>97</sup> KALVELANGEN, Erwin, Dantzig-Wolfe decomposition with GAMS [online]. 2003, Available from: <http://amsterdamoptimization.com/pdf/dw.pdf>

lagrangeanas por Kalvelangen (2009)<sup>98</sup>, de método de dos fases por Kalvelangen (2003)<sup>99</sup>, o para problemas de programaciones enteras como el Branch and Bound por Kalvelangen (2003)<sup>100</sup>.

---

<sup>98</sup> KALVELANGEN, Erwin, Lagrangia relaxation with GAMS [online]. 2009, Available from: <http://www.amsterdamoptimization.com/pdf/lagrange.pdf>

<sup>99</sup> KALVELANGEN, Erwin, Two stage stochastic linear programming with GAMS [online]. 2003, Available from: <http://www.amsterdamoptimization.com/pdf/twostage.pdf>

<sup>100</sup> KALVELANGEN, Erwin, Branch-and-Bound Methods for an MINLP model with semi-continuous variables [online]. 2003, Available from: <http://www.amsterdamoptimization.com/pdf/bb.pdf>

## 6. DESARROLLO DE LA HERRAMIENTA.

### 6.1. PROGRAMACIÓN LINEAL

En las aplicaciones computacionales, el uso del algebra lineal hace que sea más cómodo la formulación de los algoritmos de solución de un problema de programación lineal. Las ecuaciones (86)-(88) muestra la formulación matemática de un PL mediante vectores y matrices.

$$\text{Max o Min } z = cx \quad (86)$$

Sujeto a

$$Ax = b \quad (87)$$

$$x \geq 0 \quad (88)$$

Siendo  $c$  un vector de coeficientes de la función objetivo (86) y  $x$  las variables de decisión del problema modelado, se formulan las restricciones de forma de igualdad dada por la ecuación (87), donde se han incluido tanto las variables de decisión como las variables de holgura, excedentes y/o artificiales según sea el caso.

### 6.2. PROBLEMA DUAL

Para cualquier problema de programación lineal que se resuelve, existe otro que se resuelve de forma simultánea, este último se conoce como el problema dual. La pareja de problemas primal y dual posee características importantes en el contexto de los métodos exactos. Considere el siguiente problema.

$$\text{Min } z = cx \quad (89)$$

Sujeto a

$$Ax \leq b \quad (90)$$

$$x \geq 0 \quad (91)$$

El problema formulado en las ecuaciones (89)-(90) tiene la siguiente contraparte dual

$$\text{Max } z = wb \quad (92)$$

Sujeto a

$$wA \leq c \quad (93)$$

$$w \geq 0 \quad (94)$$

Las dos formulaciones tienen los mismos términos, exceptuando el término  $w$  que es una variable asignada a cada restricción de la ecuación (90) de la primera formulación, ese arreglo de variables  $w$  se conoce como arreglo de variables duales y en términos de la formulación (89)-(91) es

$$w = C_{BV}B^{-1} \quad (95)$$

Los términos de la ecuación (95) corresponden a una iteración de un Simplex Revisado, de aquí el hecho de que sean dependientes uno del otro, puesto que los coeficientes de la función objetivo del problema primal (89)-(91) se encuentran asociados a la base.

### 6.3. DESCOMPOSICIÓN DE DANTZIG WOLFE

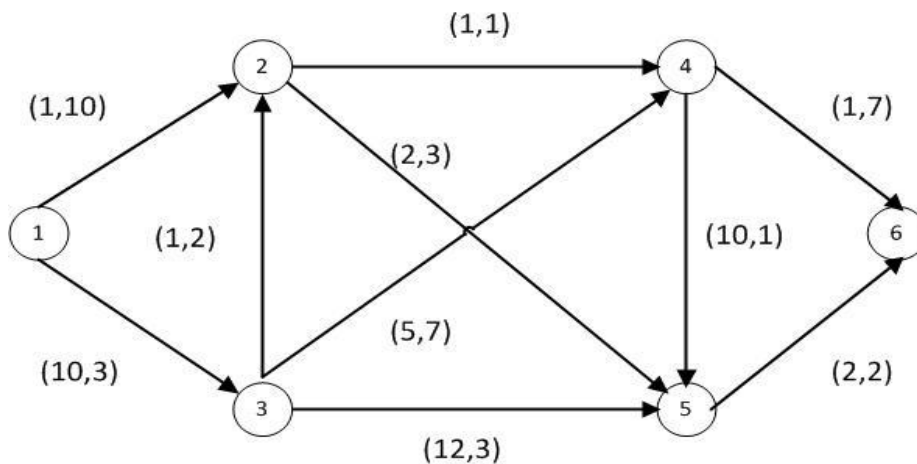
Como se estudió en el numeral 2.1.1.4, la descomposición de Dantzig Wolfe se desarrolló originalmente para problemas de programación lineal mas no de programación entera como el caso del CVRP. Sin embargo, es una herramienta

excepcional cuando se usa en las relajaciones del problema original, como el BC y el BB.

En el numeral 2.1.1.4 se sugiere que los problemas tengan una estructura de bloque para obtener ventajas en solucionar sub-problemas más pequeños, sin embargo no es obligatorio tener esta estructura Para el diseño del algoritmo como en el caso del CVRP. Un ejemplo pequeño permitirá evaluar la bondad de la descomposición DW. El ejemplo se explica paso a paso para que el lector tenga una mayor comprensión sobre el algoritmo.

**4.3.1 Descomposición de Dantzig Wolfe en un problema de optimización combinatoria.** Para estudiar el método de descomposición de Dantzig Wolfe, se toma como problema de estudio el problema de la ruta más corta, SPP por sus siglas en inglés, sujeto al grafo mostrado en la Figura 4. La solución detallada se encuentra en el Anexo A. Se recomienda enumerar sólo los arcos conectados en el grafo de la Figura 4.

Figura 4. Red de Shortest Path Problem



Fuente: Desaulniers, Desrosiers, Solomon (2005)<sup>101</sup>

<sup>101</sup> DESAULNIERS, G, DESROSIERS, Jacques and SOLOMON, Marius, Column Generation [online]. 2005,

Las etiquetas de cada arco corresponden a la forma  $(c_{ij}, t_{ij})$ . Donde,  $c_{ij}$  indica el costo de tomar determinado arco; y  $t_{ij}$ , el tiempo de recorrido o viaje. El objetivo de este problema es encontrar la ruta con el costo mínimo y tiempo no mayor a 14 unidades.

La formulación matemática del problema se muestra a continuación.

$$\text{Min } z = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (95)$$

Sujeto a

$$\sum_{j:(1,j) \in A} x_{1j} = 1 \quad (96)$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = 0 \quad (97)$$

$$\sum_{i:(i,6) \in A} x_{i6} = 1 \quad (98)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq 14 \quad (99)$$

$$x_{ij} = \begin{cases} 0 \\ 1 \text{ si } (i,j) \in A \end{cases} \quad (100)$$

La función objetivo (95) es el costo total de transporte; la restricción (96) garantiza que el origen de la ruta sea el nodo 1 y, la restricción (98), que el final de la ruta sea el nodo 6; el conjunto de restricciones (97) se conocen como restricciones de balance de flujo, es decir que si en la solución hay un arco incidente en un nodo  $j$ , también se tiene un arco que abandona el nodo; la restricción (99) restringe el espacio de soluciones a rutas que sólo consuman un tiempo menor a 14 unidades y la ecuación (100) corresponde a la condición de enteras para las variables  $x_{ij}$ .

En la estrategia de descomposición se asume que en la formulación original las restricciones se pueden desagregar, la restricción (99) se incluye al master, y las demás restricciones en el sub-problema. De esta manera, el máster y el sub-problema se definen como sigue

### Máster

$$\text{Min } z = \sum_{p \in P} (\sum_{(i,j) \in A} c_{ij} x_{pij}) \lambda_p \quad (101)$$

Sujeto a

$$\sum_{p \in P} (\sum_{(i,j) \in A} t_{ij} x_{pij}) \lambda_p \leq 14 \quad (102)$$

$$\sum_{p \in P} \lambda_p = 1 \quad (103)$$

### Sub-problema

$$\text{Min } \bar{c} = \sum_{(i,j) \in A} (c_{ij} - \pi_1 t_{ij}) x_{ij} - \pi_0 \quad (104)$$

Sujeto a

$$\sum_{j; (1,j) \in A} x_{1j} = 1 \quad (105)$$

$$\sum_{j; (i,j) \in A} x_{ij} - \sum_{j; (j,i) \in A} x_{ji} = 0 \quad (106)$$

$$\sum_{i; (i,6) \in A} x_{i6} = 1 \quad (107)$$

$$x_{ij} = 0 \text{ } \forall (i,j) \in A \quad (108)$$

En la figura 5 se muestra el pseudocódigo del algoritmo desarrollado por el autor para la solución del problema SPP

Figura 5. Pseudocódigo de Dantzig-Wolfe a un SSP

#### **Inicio**

**Paso 1:** Establecer la base del máster para encontrar sus variables duales.

**Paso 2:** Con las variables duales  $\pi_0$  y  $\pi_1$  encontradas, construir y resolver el sub-problema.

**Mientras**  $\bar{c} \neq 0$

**Paso 3:** Con la ruta de solución del sub-problema, encontrar el costo y tiempo asociados a la ruta.

**Paso 4:** De los parámetros del paso 3, generar la nueva columna.

**Paso 5:** Encontrar la variable básica que debe abandonar la base actual.

**Paso 6:** Agregar la nueva columna a la base del máster.

**Paso 7:** Con las variables duales  $\pi_0$  y  $\pi_1$  encontradas, proponer y solucionar el sub-problema.

**Fin Mientras**

**Fin**

Es de anotar que el sub-problema debe generar soluciones enteras sin embargo, a pesar que el master recibe soluciones enteras no existe garantía alguna que éste genere soluciones enteras. Por esto se debe utilizar alguna relajación como BB o BC y repetir el procedimiento hasta encontrar una solución entera.

Las soluciones del procedimiento anterior se presentan en la Tabla 2, las cuales han sido probadas por el autor usando como herramienta GAMS (Anexo A). Observe que en la primera iteración no se encuentran soluciones enteras para el máster, la solución obtenida es  $\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$ , y la solución del problema tampoco es entera en términos de  $x_{ij}$  puesto que se obtiene  $x_{12} = 0.8 x_{13} = 0.2 x_{32} = 0.2 x_{25} = 1 x_{56} = 1$ .

Tabla 2. Iteración 0 de la solución del SPP

iteración	Solución del Master	$z$	$\pi_0$	$\pi_1$	$\bar{c}$	ruta	$c$	$t$
BB.01	$A_1 = 1$	100	100	0	-97	1246	3	18
BB.02	$A_1 = 0.22, \lambda_{1246} = 0.78$	24.6	100	-5.39	-32.9	1356	24	8
BB.03	$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.8	-2.1	-4.8	13256	15	10
BB.04	$\lambda_{1246} = \lambda_{13256} = 0.5$	9	30	-1.5	-2.5	1256	5	15
BB.05	$\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$	7	35	-1	0			

Fuente: Desaulniers, Desrosiers, Solomon (2005)<sup>102</sup>

Puede que la solución obtenida en la iteración 0 no sea entera en términos de  $x_{ij}$ , pero dos de las cinco variables que integran la solución lo son. Con esta solución se hace uso de una relajación como las presentadas en el numeral 2.2, que para

<sup>102</sup> Ibid p.4

este caso se utiliza BB y, debido a que es una aplicación manual, se plantea una estrategia de ramificación agrupando variables dependientes entre sí. Se proponen dos árboles: el primer árbol está asociado a la combinación de variables  $x_{13} + x_{32}$ , ya que cualquier posible ruta obtenida de la solución de la iteración 0 las hace dependientes una de la otra; el segundo árbol corresponde a la variable  $x_{12}$  que con la combinación del primer árbol es completamente excluyente ya que si  $x_{12}$  entra en una solución entera, tanto  $x_{13}$  como  $x_{32}$  no pueden hacer parte de esa solución. Los resultados de las ramificaciones mencionadas se muestran en la Tabla 3.

Tabla 3. Ramificaciones del SPP

iteración	Solución del Master	$z$	$\pi_0$	$\pi_1$	$\pi_2$	$\bar{c}$	ruta	$c$	$t$
BB.11	$A_1 = 0.067, \lambda_{1256} = 0.933$	11.3	100	-6.33	-	0			
se incrementa el valor de $M=1000$									
BB.12	$A_1 = 0.067, \lambda_{1256} = 0.933$	71.3	1000	-66.33	-	-57.3	12456	14	14
BB.13	$\lambda_{12456} = 1$	14	1000	-70.43	-	0			
BB.21	$\lambda_{1246} = \lambda_{13256} = 0.5$	9	15	-0.67	3.33	0			
BB.31	$\lambda_{13256} = 1$	15	15	0	0	-2	13246	13	13
BB.32	$\lambda_{13246} = 1$	13	13	0	0	0			
BB.41	$A_1 = 0.067, \lambda_{1256} = 0.933$	111.3	100	-6.33	100	0			
Infactible BB.4									

Fuente: Desaulniers, Desrosiers, Solomon (2005)<sup>103</sup>

En resumen, la solución óptima entera del problema SPP corresponde a la ramificación BB3 que corresponde a  $x_{12} = 0$  y a la ruta 13246.

#### 6.4. ANÁLISIS DE INSTANCIAS PARA EL CVRP

En la literatura revisada los trabajos como el de Baldacci, Christofides, Mingozzi (2007)<sup>104</sup>, Fukasawa, Longo, Lysgaard, Aragão, Reis, Uchoa, Werneck (2005)<sup>105</sup> y

<sup>103</sup> Ibíd p.6

<sup>104</sup> BALDACCII, Roberto, CHRISTOFIDES, Nicos and MINGOZZI, Aristide, An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming [online]. 1 August 2007. Vol. 115, no. 2, p. 351–385.

<sup>105</sup> FUKASAWA, Ricardo, LONGO, Humberto, LYSGAARD, Jens, ARAGÃO, Marcus Poggi de, REIS, Marcelo, UCHOA, Eduardo and WERNECK, Renato F., Robust Branch-and-Cut-and-Price

Ralphs (2003)<sup>106</sup>, entre otros, han utilizado instancias del Benchmark que se pueden encontrar en el sitio web [www.branchandcut.org](http://www.branchandcut.org).

Las instancias tienen parámetros fijos y conocidos, cada instancia se describe de la forma A-n63-k10. La primera letra representa al autor que la propuso, en este caso Augerat et al; n63, indica que la red consta de 63 nodos incluyendo el depósito, y k10 indica que el problema requiere la asignación de rutas a 10 vehículos.

Las instancias elegidas en el trabajo son de tipo P\_nXX\_kXX debido a que son más variadas, en lo que respecta a los nodos que van de 16 nodos hasta 101, y a los vehículos de 2 a 15, el grafo es no dirigido y los valores de los pesos son iguales en ambas direcciones, la distancia entre nodos es euclidia en 2 dimensiones. El modelo matemático para las instancias discutidas se presenta a continuación:.

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (109)$$

$$\sum_{i \in V} x_{ij} = 1 \quad (110)$$

$$\sum_{j \in V} x_{ij} = 1 \quad (111)$$

$$\sum_{i \in V} x_{i0} = K \quad (112)$$

$$\sum_{j \in V} x_{0j} = K \quad (113)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S) \quad \forall S \subseteq V \setminus 0, S \neq \emptyset \quad (114)$$

$$x_{ij} \in \{0,1\} \quad (115)$$

Se eligió esta formulación ya que en aplicaciones de Baldacci, Christofides, Mingozzi (2007)<sup>107</sup> y Fukasawa, Longo, Lysgaard, Aragão, Reis, Uchoa, Werneck

---

for the Capacitated Vehicle Routing Problem. Mathematical Programming [online]. 12 October 2005. Vol. 106, no. 3, p. 491–511.

<sup>106</sup> RALPHS, T.K., Parallel branch and cut for capacitated vehicle routing. Parallel Computing [online]. 2003. Vol. 29, no. 5, p. 607–629.

<sup>107</sup> BALDACCI, Roberto, CHRISTOFIDES, Nicos and MINGOZZI, Aristide, Op cit, p. 351–385.

(2005)<sup>108</sup> se tomaron formulaciones similares para estas instancias. Inicialmente, se estudia la formulación completa sin realizar ninguna descomposición con el propósito de evidenciar la complejidad del problema. Seguidamente, se plantea una descomposición de DW, donde se incluye en el problema maestro las restricciones (112) y (113); las restricciones restantes en el sub-problema. Lo anterior se hace dado que la formulación no tiene una estructura de bloque y es necesario garantizar que en el sub-problema se generen soluciones, lo cual no se logra sólo con las restricciones (114) ya que el politopo generado por éstas incluye como solución factible el origen, degenerando así el problema porque siempre se generaría la misma columna al máster. Por lo tanto, se propone la siguiente formulación para el problema maestro y el sub-problema.

### Problema Maestro

$$\min \sum_{p \in P} \sum_{i \in V} \sum_{j \in V} c_{ij} \lambda_p x_{pij} \quad (116)$$

$$\sum_{p \in P} \lambda_p \sum_{i \in V} x_{pi0} = K \quad (117)$$

$$\sum_{p \in P} \lambda_p \sum_{j \in V} x_{p0j} = K \quad (118)$$

$$\sum_{p \in P} \lambda_p = 1 \quad (119)$$

### Sub-problema

$$\min \bar{c} = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} - \pi_2 \sum_{i \in V} x_{i0} - \pi_1 \sum_{j \in V} x_{0j} - \pi_0 \quad (120)$$

$$\sum_{i \in V} x_{ij} = 1 \quad (121)$$

$$\sum_{j \in V} x_{ij} = 1 \quad (122)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S) \quad \forall S \subseteq V \setminus 0, S \neq \emptyset \quad (123)$$

$$x_{ij} \in \{0,1\} \quad (124)$$

El nuevo conjunto  $P$  se usa en cada iteración para las rutas generadas. Se observa que el problema es el mismo, sólo se descompone en dos problemas con

---

<sup>108</sup> FUKASAWA, Ricardo, LONGO, Humberto, LYSGAARD, Jens, ARAGÃO, Marcus Poggi de, REIS, Marcelo, UCHOA, Eduardo and WERNECK, Renato F., Op cit , p. 491–511.



dimensiones agregando los elementos relacionados entre dimensiones para casos de conjuntos con más de una dimensión.

Para un caso práctico, un GDX para una matriz idéntica de [2X2] tiene dos dimensiones, las filas y columnas, las cuales se deben representar en GAMS como conjuntos {1,2}. Para ingresar la matriz a GAMS, ésta se define como un “parameter” y el GDX solo tendría dos registros Dimensión 1 =1, Dimensión2= 1 Dato=1 y Dimensión 1 =2, Dimensión2= 2 Dato=1 o resumiendo un arreglo

```
1 1 1
2 2 1
```

## 6.6. ALGORITMO DE DESCOMPOSICIÓN DE DANTZIG WOLFE APLICADO AL CVRP

Para el desarrollo de la formulación planteada en el numeral 4.4 se establecen tres sub-rutinas, la primera en Matlab y las dos restantes en GAMS, que corresponden a la formulación del problema maestro y del sub-problema. Para efectos del intercambio, se necesitó seis archivos GDX, dos para alimentar a cada modelo .gms, cuatro en total, y dos adicionales para recoger los resultados de estas sub rutinas. El programa general descrito en el archivo de Matlab se encuentra resumido en la Figura 7.

Figura 7. Seudocódigo Dantzig-Wolfe aplicado al CVRP

### **Inicio**

**Paso 1:** Establecer el conjunto de variables en el maestro y el lado derecho e incluir en el GDX MásterA.

**Paso 2:** Establecer la base y el costo del máster e incluir en el GDX MásterB.

**Paso 3:** Solucionar el problema maestro (Máster.gms).

**Paso 4:** Extraer las variables duales del GDX CVRP\_MÁSTER.

**Paso 5:** Organizar en el GDX PricingA los conjuntos de nodos, sub-tours a enumerar y parámetros constantes y en GDX PricingB las variables duales.

**Paso 6:** Solucionar el Sub-problema (Pricing.gms) y extraer los parámetros para generar la nueva columna en el GDX CVRP\_PRICING.

**Mientras  $\bar{c} < 0$**

**Paso 7:** Establecer la nueva columna que ingresa a la base y cual debe salir de esta.

**Paso 8:** Modificar la base y los costos del maestro e incluirlos en el GDX MásterB.

**Paso 9:** Solucionar el problema maestro y extraer las variables duales en ingresarlas en el GDX PricingB.

**Paso 10:** Solucionar el sub-problema (Pricing.gms) extrayendo los parámetros de la nueva columna generada.

**Fin Mientras**

**Paso 11:** Consolidar la respuesta.

**Fin**

## 7. RESULTADOS OBTENIDOS

En esta sección se discuten los resultados de la aplicación del método de descomposición de Dantzig Wolfe comparado con la solución obtenida por el solver CEPLEX, La eficiencia del algoritmo se prueba una serie de instancias de la literatura. Se seleccionan dos instancias la P\_n16\_k8 y la P\_n19\_k2. En la tabla 4 se presentan los resultados arrojados por el algoritmo:

Tabla 4. Tiempos en segundos Exacto sin descomponer

<b>S</b>	<b>P_n16_k8</b>	<b>P_n19_k2</b>
2	0,807612	0,17877
3	0,243615	0,248763
4	0,443328	1,271368
5	1,947178	10,840778
6	7,518216	72,561255
7	21,082206	591,183741
8	42,849806	2385,72468
9	71,43045	5860,1733
10	93,435029	10420,8079
11	104,25444	-
12	109,464665	-
13	112,199789	-
14	113,280404	-
15	111,94763	-

La primera columna de la Tabla corresponde a la selección de los conjuntos S con su respectivas cardinalidades máximas de las restricción de eliminación de sub-tours o restricción de cortes de capacidad (CCC's por sus siglas en ingles). Se seleccionó este conjunto para analizar qué tanto variaba la complejidad de acuerdo al tamaño del conjunto, las columnas 2 y 3 corresponde al tiempo en segundos de las instancias seleccionadas.

Las corridas de las instancias se hicieron en un equipo con un procesador Intel® Core™ i7-2600 CPU @ 3.40 GHz 3.40 GHz, memoria RAM 8.00 GB (3.17 GB utilizable) y sistema operativo Windows 7 profesional de 32 bits.

En la instancia P\_n19\_k2 no se obtienen soluciones factibles, en el problema no se alcanza a enumerar una parte representativa del conjunto S. Por el contrario, en la instancia P\_n16\_k8 se obtienen soluciones factibles desde que  $|S| \geq 5$  y soluciones óptimas a partir de  $|S| \geq 13$ . En la figura 8 y 9 se presentan las gráficas del tiempo contra la cardinalidad de los sub-tour.

Figura 8. Tiempo en segundos vs máxima cardinalidad de S P\_n16\_k8

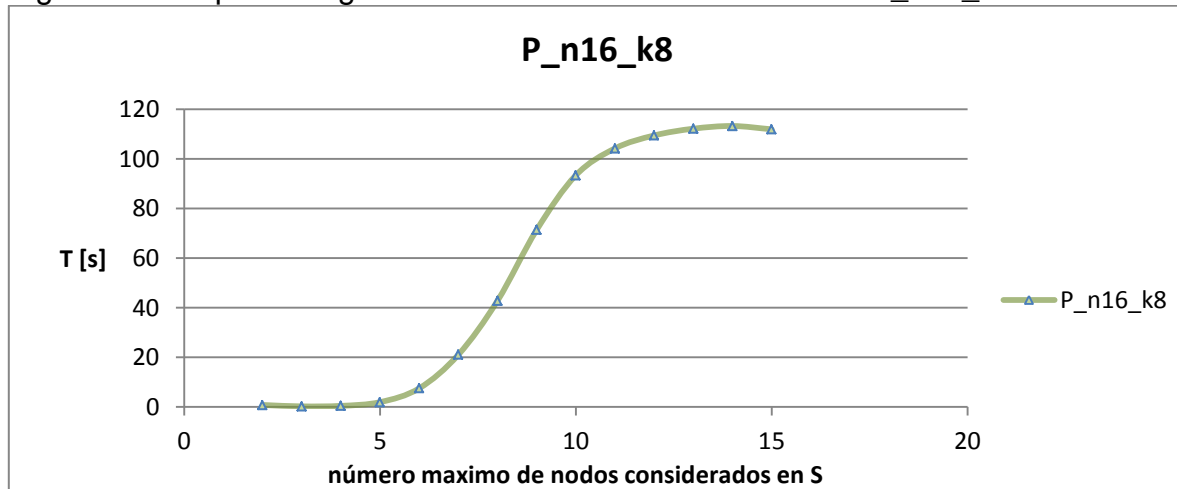
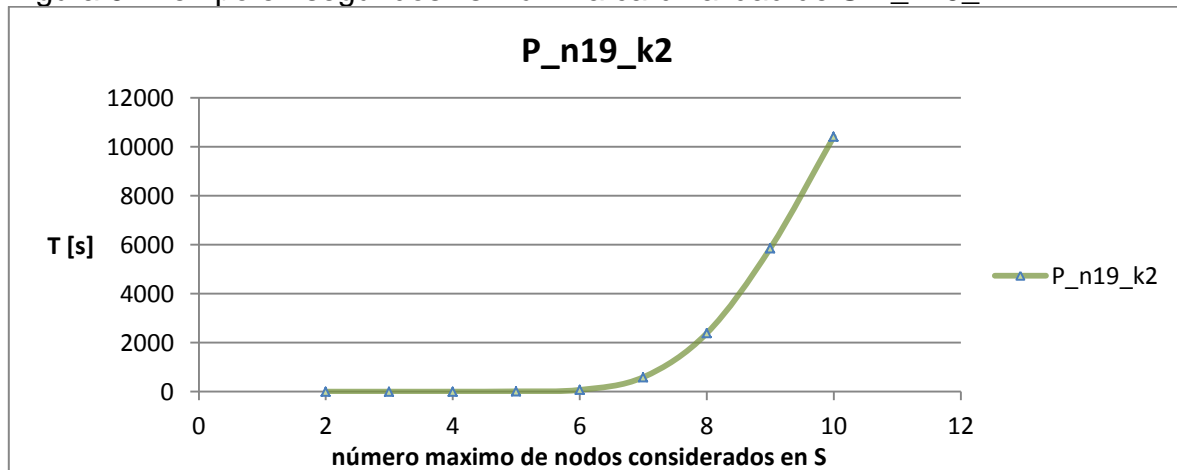


Figura 9. Tiempo en segundos vs máxima cardinalidad de S P\_n19\_k2



Se puede observar que para pequeños cambios de cardinalidades del sub-tour, las variaciones de tiempo son altas, corroborando así que la principal fuente de explosión combinatoria en este tipo de problemas es la restricción de eliminación de sub-tours, y que en otros no es necesario enumerar todos los posibles sub-tours para conseguir soluciones factibles aunque dependa del tipo de instancia resolver. Para el algoritmo de descomposición, los resultados para estas instancias se presentan en la Tabla 5:

Tabla 5. Tiempo en segundos método exacto descompuesto

S	P_n16_k8	P_n19_k2
2	1,241541	1,258734
3	1,317767	1,770973
4	18,12027	8,469487
5	70,87927	74,26878
6	98,94007	516,7709
7	184,0691	4051,57
8	309,4432	16539,03
9	372,4164	46850,5
10	441,0136	31131,04
11	449,1886	-
12	454,4909	-
13	338,8535	-
14	480,1204	-
15	441,558	-

Se puede observar que los tiempos de computo con el algoritmo son mayores que para la versión no descompuesta, sin embargo la factibilidad en la instancia P\_n16\_k8 converge a la solución óptima desde  $|S| \geq 4$ , mucho antes que el no descompuesto (Figura 12).

Figura 10. Tiempo en segundos vs máxima cardinalidad de S DW P\_n16\_k8

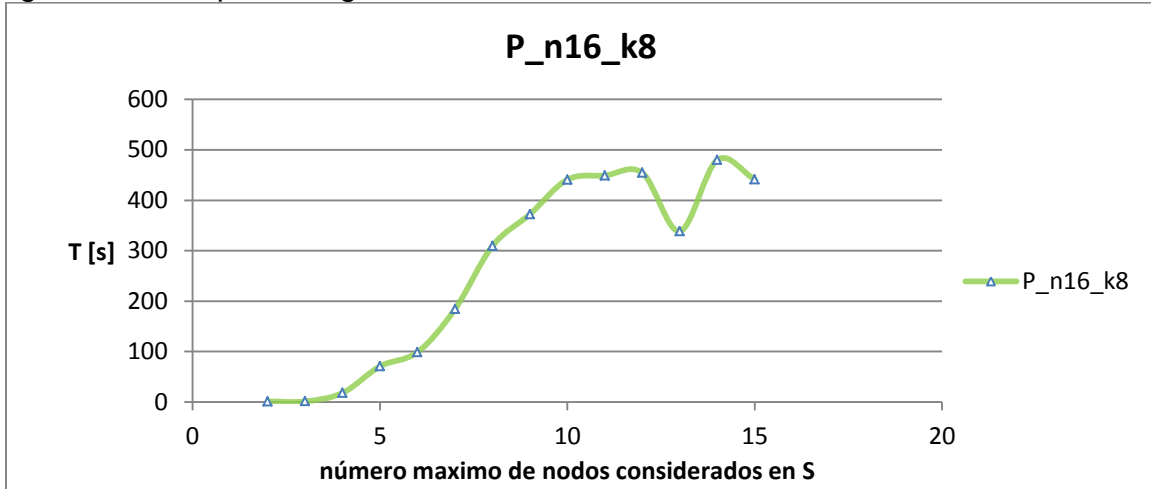
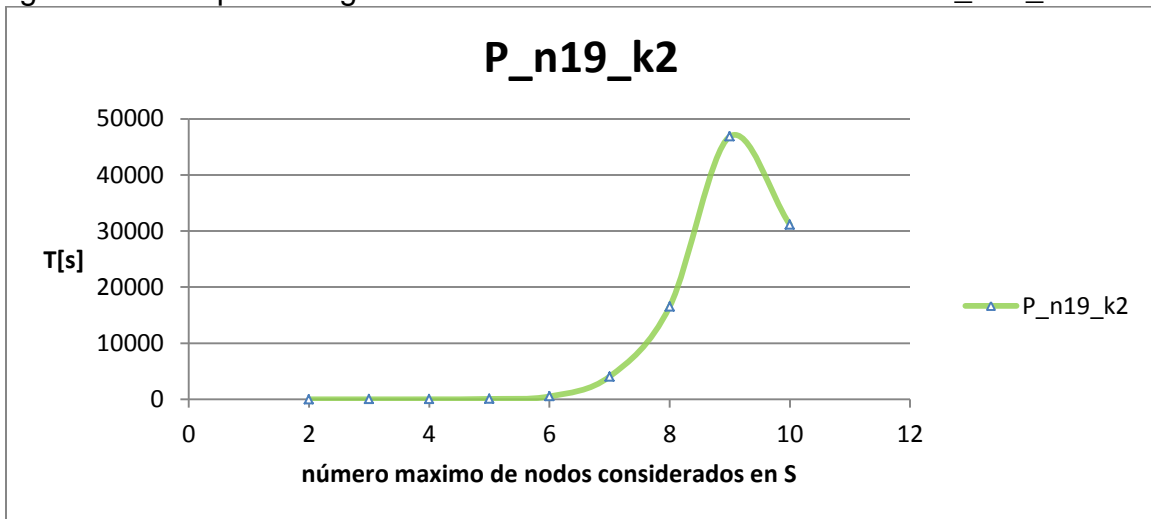


Figura 11. Tiempo en segundos vs máxima cardinalidad de S DW P\_n19\_k2



Los tiempos de computo dependen de la máxima cardinalidad enumerada, esto es, pequeños cambios de cardinalidad del conjunto de sub-tours, se traduce en grandes incrementos de tiempo de ejecución. En el caso de la instancia P\_n19\_k2, ésta no alcanza una solución factible antes de encontrar los límites de memoria aunque para el caso de cardinalidades de 11 en la versión descompuesta se alcanza agregar una columna al máster, es decir una solución que no cumple con la restricciones de requerimiento de camiones pero es óptima para el sub-problema generando, una pequeña ventaja para el uso de memoria.

La Figura 13 compara los resultados en esta instancia, sin incluir la onceava cardinalidad ya que no se completó la corrida. Lo anterior, indica que se puede encontrar una ventaja en un algoritmo de descomposición, pero esto también depende de la instancia evaluada, no debe generalizarse.

Figura 12. Comparación DW y la versión no descompuesta P\_n16\_k8

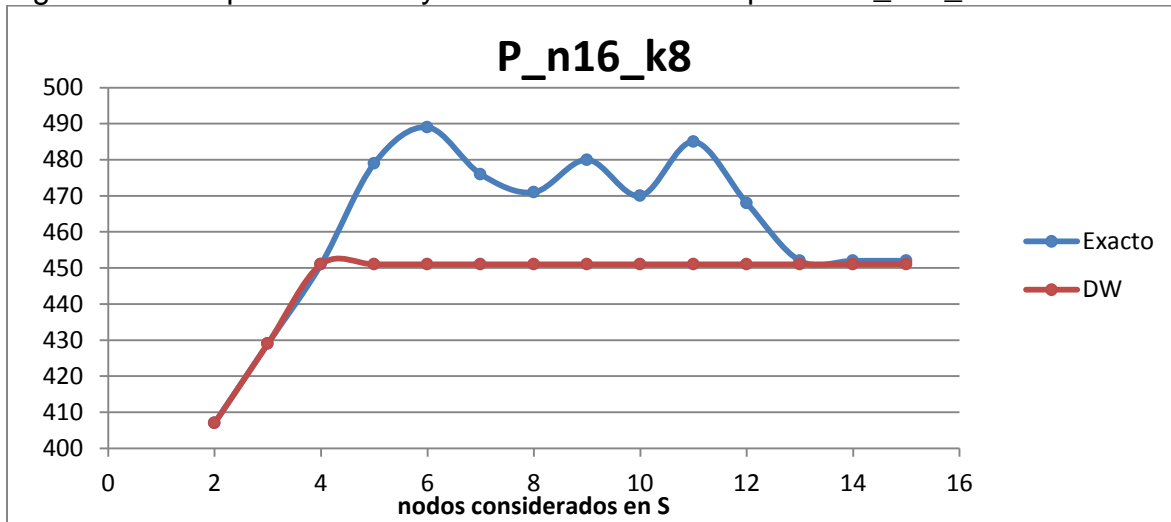
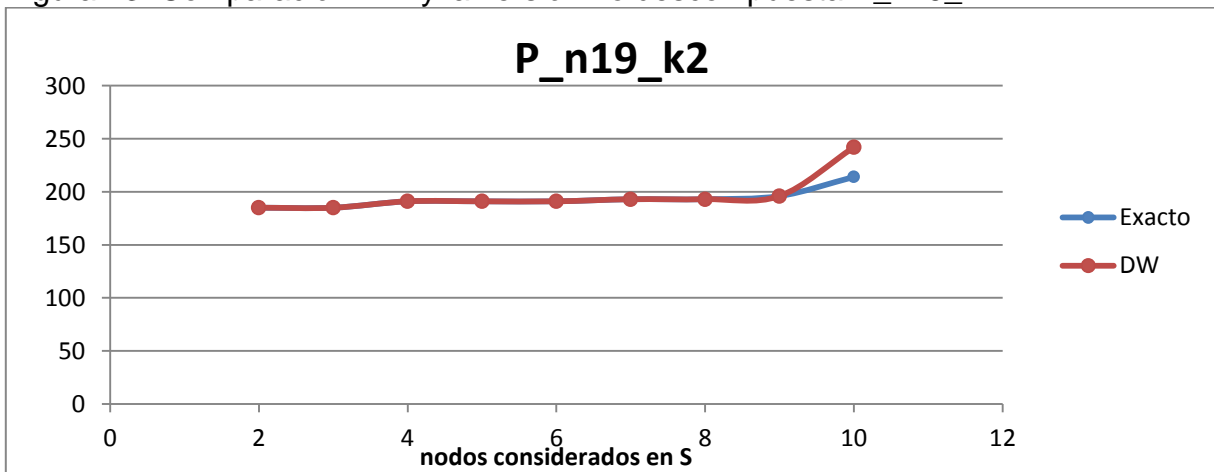


Figura 13. Comparación DW y la versión no descompuesta P\_n19\_k2



Dentro de las aplicaciones estudiadas, se encontró que el tiempo requerido para solucionar el problema de ruteo de vehículos con restricciones de capacidad se incrementaba dramáticamente con incrementos de los tamaños permitidos para los conjuntos de eliminación de sub-tours. No es de asombrar ya que para un

CVRP completo  $[\forall S] = 2^{n-1}$  siendo  $n$  el número de nodos asociados a la red de estudio, así si se considera un computador con una memoria ram 3GB y de 32 bits Matlab limita la memoria disponible para un arreglo es 432.700.000 bytes es decir 54.087.500 elementos de un arreglo de naturaleza double, aproximadamente enumeraría un conjunto máximo de 24 clientes para el caso de Matlab la operación más grande es la enumeración del conjunto que corresponde a la cantidad de conjuntos  $S$  a enumerar.

En GAMS no solo se tiene en cuenta la cantidad de conjuntos  $S$  necesarios el tamaño real del arreglo de restricciones de sub-tours depende también del número de variables tomadas en cuenta que estaría dado por  $(n \times n) - n$  aumentando el espacio necesario a un arreglo de  $2^{n-1} \times (n \times n) - n$  y reduciendo la cantidad de clientes posibles a considerar, además GAMS solo utiliza la memoria física disponible que para este caso es de 1.389 MB equivalentes a 1.389.000.000 bytes y 173.625.000 espacios en un arreglo. Al considerar las demás restricciones del sub-problema la dimensión del arreglo aumenta a  $(2^{n-1} + 2n + 2) \times (n^2 - n)$  espacios en la versión relajada que serán limitados por la memoria física sin contar la variables no cero o restricciones adicionadas con la estrategia de ramificación seleccionada y así alcanzaría para formular un sub-problema de un CVRP máximo hasta 19 clientes en la versión relajada.

Para seccionar el conjunto  $S$ , se toma la máxima cardinalidad que se desea considerar. En la literatura, a las soluciones generadas se les conoce como rutas sin  $s$ -ciclos, estas respuestas son rutas restringidas a no formar ciclos hamiltonianos con conjuntos de clientes  $\leq s$  es decir se enumera  $\sum_{i=1}^s \binom{n-1}{i}$  es fácil demostrar que cuando  $s = n - 1$   $\sum_{i=1}^s \binom{n-1}{i} + 1 = 2^{n-1}$  en el caso de enumerar todos los conjuntos posibles incluidos en las restricciones de eliminación de sub-tours.

De esta forma, para la formulación (109)-(114), se encuentra que es necesario  $\sum_{i=1}^s \binom{n-1}{i} + 2n + 2 - n$  restricciones, sólo en la versión relajada para la aplicación de estrategias como BB y BC. Se deben considerar el ingreso de variables y restricciones al tablero para garantizar encontrar una solución entera. Las formulaciones (116)-(119) que corresponden al problema maestro y las (120)-(124) que corresponden al sub-problema, requieren 3 restricciones y  $\sum_{i=1}^s \binom{n-1}{i} + 2n - n$  restricciones, respectivamente. Para el sub problema, no sólo se disminuye la cantidad de restricciones sino también el número de posibles restricciones a agregar para encontrar una solución entera.

Por tanto, en la aplicaciones mostradas no se pudo obtener soluciones para un problema de 19 nodos para conjuntos con cardinalidades superiores a 10 que es menor a 18, ya que ese es el límite suministrado por GAMS, se necesitaban agregar más cortes de lo que permitía esta plataforma en Windows de 32bits, pero es de resaltar que el algoritmo de descomposición logró generar una solución no factible, la cual no se puede graficar porque no es entera.

De forma similar, un equipo de Windows de 64 bits con 8GB de Ram Matlab limita la memoria disponible para un arreglo a 13.490.000.000 Bytes, es decir que un arreglo con 1.686.300.000 elementos de un arreglo double que aproximadamente enumeraría completamente hasta un conjunto del número de sub-tours de 30 clientes, pero, al igual que en el caso de Windows de 32bits, sólo se tienen en cuenta los límites de Matlab, porque los límites de GAMS dependen de las variables no ceros y restricciones adicionales que agregue al tablero según la estrategia de ramificación que se utilice para encontrar las soluciones enteras. Pero si sólo se considera la versión relajada y una memoria física disponible de 3.976 MB equivalentes a 3.976.000.000 bytes y 497.000.000 espacios de un arreglo doblé, que equivale a formular un sub-problema de máximo 21 nodos, los cálculos realizados presentados son netamente teóricos en aplicaciones reales pueden variar en algunas unidades.

## 8. CONCLUSIONES

Dentro de las instancias estudiadas, se encontró que la dificultad de los métodos exactos aplicados al problema de ruteo de vehículos con restricciones de capacidad no sólo radica en el tiempo sino también en el consumo de memoria requerido para la aplicación de estos métodos.

El problema de ruteo de vehículos con restricciones de capacidad no necesita formularse completo para llegar a una solución óptima, aunque está sujeto a las condiciones del problema.

Una estructura descompuesta presenta ventajas a la hora de enumerar los sub-tours, ya que converge a una solución óptima antes que en la versión no descompuesta.

Se encuentran mayores espacios de memoria en sistemas operativos de 64 bits comparadas con los de 32 bits, por lo que se sugiere este tipo de sistemas operativos para aplicaciones futuras.

El consumo de memoria del CVRP se incrementa dramáticamente con la cantidad de nodos considerados para las restricciones de eliminación de sub-tours.

Dentro de la literatura encontrada, se resalta como un campo de estudio prometedor la descomposición de Dantzig Wolfe para problemas de optimización combinatoria.

## 9. RECOMENDACIONES

Como recomendación para aplicaciones futuras, se propone no aplicar la descomposición de Dantzig-Wolfe netamente exacta, dado que los requerimientos de tiempo y memoria son bastante altos e imposibles para computadores comunes. En lugar de esto, se sugiere un enfoque de descomposición de Dantzig-Wolfe junto con generación de columnas mixto, es decir una combinación con métodos estocásticos. Para lo anterior, se propone modelar el problema maestro y el sub-problema así:

### Maestro

$$\min \sum_{p \in P} \sum_{i \in V} \sum_{j \in V} c_{ij} \lambda_p x_{pij} \quad (125)$$

$$\sum_{p \in P} \lambda_p \sum_{i \in V} x_{pi0} = K \quad (126)$$

$$\sum_{p \in P} \lambda_p \sum_{j \in V} x_{p0j} = K \quad (127)$$

$$\sum_{p \in P} \lambda_p \sum_{i \in V} x_{pij} = 1 \quad (128)$$

$$\sum_{p \in P} \lambda_p \sum_{j \in V} x_{pij} = 1 \quad (129)$$

$$\sum_{p \in P} \lambda_p \sum_{i \in S} \sum_{j \in S} x_{pij} \leq |S| - r(S) \quad \forall S \subseteq V \setminus 0, S \neq \emptyset \quad (130)$$

$$\sum_{p \in P} \lambda_p = 1 \quad (131)$$

### Sub-problema

$$\min \bar{c} = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} - \pi_1 \sum_{i \in V} x_{i0} - \pi_2 \sum_{j \in V} x_{0j} - \pi_3 \sum_{i \in V} x_{ij} - \pi_4 \sum_{j \in V} x_{ij} - \pi_{5S} \sum_{i \in S} \sum_{j \in S} x_{ij} - \pi_0 \quad (132)$$

La formulación (125)-(132) propone generar soluciones de un grafo modificado por las variables duales del máster, es decir  $\min \bar{c} = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} - \pi_1 \sum_{i \in V} x_{i0} - \pi_2 \sum_{j \in V} x_{0j} - \pi_3 \sum_{i \in V} x_{ij} - \pi_4 \sum_{j \in V} x_{ij} - \pi_{5S} \sum_{i \in S} \sum_{j \in S} x_{ij}$ , ya que al ser  $\pi_0$  una constante no es relevante al objetivo de minimizar, es de aclarar que se debe evaluar el costo reducido de la nueva columna en (132) y el método para generar dichas respuestas se sugiere que sea estocástico, en pocas palabras que el

máster sea el medio para que el método estocástico converja a una solución óptima.

Ya que después del análisis de resultados se encontró que en un enfoque de descomposición puede brindar ventajas cuando se enumera parcialmente los conjuntos de sub-tours, se recomienda que para efectos de memoria no se enumere por completo el conjunto S en el problema maestro.

Finalmente, después de haber realizado los experimentos con ayuda de archivos GDX se recomienda avanzar más en aplicaciones con esta herramienta, ya que para efectos de modelar y desarrollar el método presenta un manejo sencillo, el cual se puede explotar aún más incluyendo el uso de librerías de Matlab como *optim* y *parallel.pool* que ayudan a resolver problemas de optimización con varias técnicas predefinidas y permite hacer prácticas de computación en paralelo en un computador de procesadores con más de un núcleo o un clúster de computadores respectivamente.

## BIBLIOGRAFÍA

ABBASBANDY, S., EZZATI, R. and JAFARIAN, A. LU decomposition method for solving fuzzy system of linear equations. Applied Mathematics and Computation [online]. 2006. Vol. 172, no. 1, [Cited 13 December 2013]. p. 633–643. Available from: <http://www.sciencedirect.com/science/article/pii/S0096300305002286>

AGIN, Norman. Optimum Seeking with Branch and Bound. Management Science [online]. 1966. Vol. 13, no. 4, [Cited 23 March 2014]. p. B176–B185.

AKKER, Jm van den, HURKENS, Caj and SAVELSBERGH, Mwp. Time-Indexed Formulations for Machine Scheduling Problems: Column Generation. [online]. 1996. [Cited 26 January 2014]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.3335&rep=rep1&type=pdf>

BARNHART, Cynthia, JOHNSON, Ellis L., NEMHAUSER, George L., SAVELSBERGH, Martin W. P. and VANCE, Pamela H. Branch-and-Price: Column Generation for Solving Huge Integer Programs. [online]. 1 June 1998. [Cited 26 January 2014]. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.46.3.316>

BARTELS, Richard H. and GOLUB, Gene H. The simplex method of linear programming using LU decomposition. Communications of the ACM [online]. 1 May 1969. Vol. 12, no. 5, [Cited 10 December 2013]. , p. 266–268. Available from: [http://dl.acm.org/citation.cfm?id=362946.362974\\_DOI 10.1145/362946.362974](http://dl.acm.org/citation.cfm?id=362946.362974_DOI 10.1145/362946.362974).

BAZARAA, Mokhtar S., JARVIS, John J. and SHERALI, Hanif D. Programación Lineal y Flujo de Redes. Segunda Ed. Mexico, 1998. ISBN 968-18-4867-5.

BEASLEY, J. E. Advances in linear and integer programming. [online]. 1 March 1996. [Cited 27 January 2014]. Available from: <http://dl.acm.org/citation.cfm?id=247975>

BERTIER, P. and ROY, B. Procédure de Résolution pour une Classe de Problems Pouvant Avoir un Caractère Combinatoire. In : Cahiers Cent. d'Etudes Recherche Operationelle 6. 1964. p. 202–208.

BIXBY, Robert E. Implementing the Simplex Method: The Initial Basis. [online]. 1 September 1991. [Cited 27 January 2014]. Available from: <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA453079>

\_\_\_\_\_. Progress in Linear Programming. [online]. 1993. [Cited 27 January 2014]. Available from: <http://www.caam.rice.edu/caam/trs/93/TR93-40.pdf>

COLOMBIANO.COM. Camioneros tendrán líneas de crédito por 120.000 millones de pesos - El Colombiano. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.elcolombiano.com/BancoConocimiento/C/camioneros\\_tendran\\_lineas\\_de\\_credito\\_por\\_120000\\_millones\\_de\\_pesos/camioneros\\_tendran\\_lineas\\_de\\_credito\\_por\\_120000\\_millones\\_de\\_pesos.asp](http://www.elcolombiano.com/BancoConocimiento/C/camioneros_tendran_lineas_de_credito_por_120000_millones_de_pesos/camioneros_tendran_lineas_de_credito_por_120000_millones_de_pesos.asp)

CONPES 3489. Política Nacional de Transporte Público Automotor de Carga [online]. 2007. Available from: <https://www.mintransporte.gov.co/descargar.php?id=3128>

CONPES 3706. Importancia Estratégica del Programa “Corredores Prioritarios para la Prosperidad” [online]. 2011. [Cited 21 February 2014]. Available from: <https://www.dnp.gov.co/LinkClick.aspx?fileticket=johYvpSM6Mc=&tabid=1260>

CORNUEJOLS, Gerard and HARCHE, Farid. Polyhedral study of the capacitated vehicle routing problem. *Mathematical Programming* [online]. June 1993. Vol. 60, no. 1-3, [Cited 14 November 2013]. p. 21–52. DOI 10.1007/BF01580599.

CHRISTIANSEN, Christian H. and LYSGAARD, Jens. *A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands* [online]. 2007. [Cited 27 January 2014].

CLARKE, G. and WRIGHT, J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* [online]. 1 August 1964. Vol. 12, no. 4, [Cited 26 January 2014]. p. 568–581. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.12.4.568>, DOI 10.1287/opre.12.4.568.

DANE. Metodología Índice de Costos del Transporte de Carga por Carretera [online]. 2010. [Cited 21 February 2014]. Available from: <http://www.dane.gov.co/files/investigaciones/fichas/ICTC.pdf>.

DANTZIG, George B. and FULKERSON, D. R. On a Linear-Programming, Combinatorial Approach to the Traveling-Salesman Problem. *Operations Research* [online]. 1959. Vol. 7, no. 1, [Cited 23 March 2014]. p. 58 – 66.

DANTZIG, G. B. and RAMSER, J. H. The Truck Dispatching Problem. *Management Science* [online]. 1 October 1959. Vol. 6, no. 1, [Cited 16 January 2014]. p. 80–91. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.6.1.80>, DOI 10.1287/mnsc.6.1.80.

DANTZIG, George B. and WOLFE, Philip. Decomposition Principle for Linear Programs. *Operations Research* [online]. 1 February 1960. Vol. 8, no. 1, [Cited 27 January 2014]. p. 101–111. Available from:

<http://pubsonline.informs.org/doi/abs/10.1287/opre.8.1.101>,  
10.1287/opre.8.1.101.

DOI

DELL'AMICO, Mauro, RIGHINI, Giovanni and SALANI, Matteo. A Branch-and-Price Approach to the Vehicle Routing Problem with Simultaneous Distribution and Collection. [online]. 1 May 2006. [Cited 27 January 2014]. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.1050.0118>.

EL-TIEMPO.CO. Avanzan obras para recuperación vial en Sogamoso - Colombia Noticias: Actualidad Nacional - ELTIEMPO.COM. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.eltiempo.com/colombia/boyaca/ARTICULO-WEB-NEW\\_NOTA\\_INTERIOR-13378439.html](http://www.eltiempo.com/colombia/boyaca/ARTICULO-WEB-NEW_NOTA_INTERIOR-13378439.html)

ELTIEMPO.COM. Balance sobre el Sistema Integrado. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.eltiempo.com/colombia/bogota/balance-sobre-el-sistema-integrado\\_13411237-4](http://www.eltiempo.com/colombia/bogota/balance-sobre-el-sistema-integrado_13411237-4)

ENTRIKEN, Robert. The Parallel of Decomposition of Linear Programs. [online]. 1 November 1989. [Cited 27 January 2014]. Available from: <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA216100>

FISCHETTI, Matteo, TOTH, Paolo and VIGO, Daniele. A Branch-and-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs. [online]. 1 October 1994. [Cited 14 November 2013]. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.42.5.846>

FISHER, Marshall L. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. [online]. 1 August 1994. [Cited 14 November 2013]. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.42.4.626>

FOURER, Robert. Solving staircase linear programs by the simplex method, 1: Inversion. *Mathematical Programming* [online]. December 1982. Vol. 23, no. 1, [Accessed 2 April 2014]. p. 274–313. Available from: <http://link.springer.com/10.1007/BF01583795>. DOI 10.1007/BF01583795.

FUKASAWA, Ricardo, LONGO, Humberto, LYSGAARD, Jens, ARAGÃO, Marcus Poggi de, REIS, Marcelo, UCHOA, Eduardo and WERNECK, Renato F. Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. *Mathematical Programming* [online]. 12 October 2005. Vol. 106, no. 3 [Cited 20 December 2013]. p. 491–511. Available from: <http://link.springer.com/10.1007/s10107-005-0644-x>, DOI 10.1007/s10107-005-0644-x.

FUKASAWA, Ricardo, POGGI DE ARAGAO, Marcus and UCHOA, Eduardo. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. In *Proceedings Of The International Network Optimization Conference* [online]. 2003. [Cited 27 January 2014]. p. 231–236. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.8142>

GARFINKEL, ROBERT S. NEMHAUSER, George L. *Integer Programing. I*. United States of America : JOHN WILEY & SONS, 1972. ISBN 0-471-29196-1.

GOMORY, Ralph E. and BAUMOL, William J. Integer Programming and Pricing. *Econometrica* [online]. 1960. Vol. 28, no. 3, [Cited 23 March 2014]. p. 521–550. Available from: [http://www.ime.unicamp.br/~moretti/ms901\\_ms915/1910130.pdf](http://www.ime.unicamp.br/~moretti/ms901_ms915/1910130.pdf)

GOMORY, R. E., HOFFMAN, A. J. and WATSON, Thomas J. On the convergence of an integer-programming process. *Naval Research Logistics Quarterly* [online]. March 1963. Vol. 10, no. 1, [Cited 23 March 2014]. p. 121–123. Available from: <http://doi.wiley.com/10.1002/nav.3800100111>, DOI 10.1002/nav.3800100111.

GOMORY, R. E., HOFFMAN, A. J. and WATSON, Thomas J. On the convergence of an integer-programming process. *Naval Research Logistics Quarterly* [online]. March 1963. Vol. 10, no. 1, [Cited 23 March 2014]. p. 121–123. Available from: <http://doi.wiley.com/10.1002/nav.3800100111>. DOI 10.1002/nav.3800100111.

GUNNERUD, Vidar, FOSS, Bjarne and TORGNES, Erlend. Parallel Dantzig–Wolfe decomposition for real-time optimization—Applied to a complex oil field. *Journal of Process Control* [online]. October 2010. Vol. 20, no. 9, [Cited 17 February 2014]. p. 1019–1026. Available from: <http://www.sciencedirect.com/science/article/pii/S0959152410001149>. DOI 10.1016/j.jprocont.2010.06.003.

HADJICONSTANTINO, Eleni, CHRISTOFIDES, Nicos and MINGOZZI, Aristide. A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations. *Annals of Operations Research* [online]. December 1995. Vol. 61, no. 1, [Cited 20 December 2013]. p. 21–43. Available from: <http://link.springer.com/10.1007/BF02098280>. DOI 10.1007/BF02098280.

HILLER, Frederick S. and LIBERMAN, Gerald J. *Introducción a la Investigación de Operaciones*. 9. Mexico, 2010. ISBN 970-10-5621-3

HO, James K. and LOU, Etienne. A Comparative Study of Two Methods for Staircase Linear Programs. *ACM Transactions on Mathematical Software* [online]. 1 March 1980. Vol. 6, no. 1, [Accessed 2 April 2014]. p. 17–30. Available from: <http://dl.acm.org/citation.cfm?id=355873.355875>. DOI 10.1145/355873.355875.

\_\_\_\_\_, James K. Note—Nested Decomposition of a Dynamic Energy Model. *Management Science* [online]. 1 May 1977. Vol. 23, no. 9, [Accessed 3 April 2014]. p. 1022–1026. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.23.9.1022>. DOI 10.1287/mnsc.23.9.1022.

JEPSEN, Mads, SPOORENDONK, Simon and ROPKE, Stefan. A Branch-and-Cut Algorithm for the Symmetric Two-Echelon Capacitated Vehicle Routing Problem. *Transportation Science* [online]. February 2013. Vol. 47, no. 1, [Cited 20 December 2013]. p. 23–37. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84873596215&partnerID=tZOtx3y1>. DOI 10.1287/trsc.1110.0399.

KIM, Daeki, BARNHART, Cynthia, WARE, Keith and REINHARDT, Gregory. Multimodal Express Package Delivery: A Service Network Design Application. [online]. 1 November 1999. [Cited 26 January 2014]. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.33.4.391>

KOHL, Niklas, DESROSIERS, Jacques, MADSEN, Oli B. G., SOLOMON, Marius M. and SOUMIS, François. 2-Path Cuts for the Vehicle Routing Problem with Time Windows. [online]. 1 February 1999. [Cited 26 January 2014]. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.33.1.101>

KULKARNI, R.V. and BHAVE, P.R. Integer programming formulations of vehicle routing problems. *European Journal of Operational Research* [online]. 1985. Vol. 20, no. 1, [Cited 26 January 2014]. p. 58–67. Available from: <http://www.sciencedirect.com/science/article/pii/037722178590284X>

LAPORTE, Gilbert, MERCURE, Hélène and NOBERT, Yves. An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks* [online]. 1986. Vol. 16, no. 1, [Cited 26 January 2014]. p. 33–46. Available from: <http://doi.wiley.com/10.1002/net.3230160104>. DOI 10.1002/net.3230160104.

LA-REPUBLICA.COM. Concesión para malla vial de Bogotá se suscribió con empresa de papel: Cardozo | La República. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.larepublica.co/economia/concesión-para-malla-vial-de-bogotá-se-suscribió-con-empresa-de-papel-cardozo\\_99266](http://www.larepublica.co/economia/concesión-para-malla-vial-de-bogotá-se-suscribió-con-empresa-de-papel-cardozo_99266)

LAWLER, E. L. and WOOD, D. E. Branch and Bound Methods: A Survey. *Operational Research* [online]. 1966. Vol. 14, no. 4, [Cited 23 March 2014]. p. 699–719.

LETCHFORD, A N, LYSGAARD, J and EGGLESE, R W. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society* [online]. 13 December 2006. Vol. 58, no. 12, [Cited 14 November 2013]. p. 1642–1651. Available from: <http://www.palgrave-journals.com/jors/journal/v58/n12/abs/2602345a>. DOI 10.1057/palgrave.jors.2602345.

LETCHFORD, Adam N., EGGLESE, Richard W. and LYSGAARD, Jens. Multistars, partial multistars and the capacitated vehicle routing problem. *Mathematical Programming* [online]. 1 December 2002. Vol. 94, no. 1, [Cited 20 December 2013]. p. 21–40. Available from: <http://link.springer.com/10.1007/s10107-002-0336-8>. DOI 10.1007/s10107-002-0336-8.

LITTLE, John D. C., MURTY, Katta G., SWEENEY, Dura W. and KAREL, Caroline. An Algorithm for the Traveling Salesman Problem. *Operations Research* [online]. 1 December 1963. Vol. 11, no. 6, [Cited 23 March 2014]. p. 972–989. Available from: <http://pubsonline.informs.org/doi/abs/10.1287/opre.11.6.972A>. DOI 10.1287/opre.11.6.972.

JEROSLOW, R. G. Comments on Integer Hulls of Two Linear Constraints. *Operations Research* [online]. 1971. Vol. 19, no. 4, [Cited 24 March 2014]. p. 1061–1069.

MARTÍ, Rafael and REINELT, Gerhard. *The Linear Ordering Problem* [online]. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. [Cited 27 January 2014]. Applied Mathematical Sciences. Available from:

<http://www.springerlink.com/index/10.1007/978-3-642-16729-4>. ISBN 978-3-642-16728-7.

MILLER, C. E., TUCKER, A. W. and ZEMLIN, R. A. Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM* [online]. 1 October 1960. Vol. 7, no. 4, [Cited 26 January 2014]. p. 326–329 Available from: <http://dl.acm.org/citation.cfm?id=321043.321046>. DOI 10.1145/321043.321046.

MILLER, D L. A Marching based exact algorithm for capacitated vehicle routing problems. *ORSA journal on computing*. Vol. 7, no. 1 [online]. 1995. [Cited 26 January 2014]. Available from: <http://trid.trb.org/view.aspx?id=652565>

PRENSA-LATINA.CU. Camioneros colombianos y Gobierno en nueva mesa de diálogo. [online]. 2014. [Cited 27 January 2014]. Available from: [http://www.prensa-latina.cu/index.php?option=com\\_content&task=view&idioma=1&id=2315641&Itemid=1](http://www.prensa-latina.cu/index.php?option=com_content&task=view&idioma=1&id=2315641&Itemid=1)

POGGI DE ARAGAO, Marcus and UCHOA, Eduardo. Integer Program Reformulation for Robust Branch-and-Cut-and-Price Algorithms. In *Proceedings of the Conference Mathematical Program in Rio: A Conference in Honour of Nelson Maculan* [online]. 2003. [Cited 27 January 2014]. p. 56–61. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.1278>

PUCHINGER, Jakob and RAIDL, Günther R. Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research* [online]. December 2007. Vol. 183, no. 3, [Cited 6 February 2014]. p. 1304–1327. Available from: <http://www.sciencedirect.com/science/article/pii/S0377221706003067>. DOI 10.1016/j.ejor.2005.11.064.

RALPHS, T.K. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing* [online]. 2003. Vol. 29, no. 5, [Cited 14 November 2013]. p. 607–629. Available from: <http://www.sciencedirect.com/science/article/pii/S0167819103000450>

RESTREPO, Hernando José Gómez. Prosperidad Para Todos [online]. Bogotá DC, 2010. [Cited 17 February 2014]. Available from: <https://www.dnp.gov.co/LinkClick.aspx?fileticket=pfUiAGFRUug=&tabid=82>

RUBIN, David S. On the Unlimited Number of Faces in Integer Hulls of Linear Programs with a Single Constraint. *Operations Research* [online]. 1970. Vol. 18, no. 5, [Cited 24 March 2014]. p. 940–946.

SANTOS, Fernando Afonso, CUNHA, Alexandre Salles and MATEUS, Geraldo Robson. Branch-and-price algorithms for the Two-Echelon Capacitated Vehicle Routing Problem. *Optimization Letters* [online]. 17 October 2012. Vol. 7, no. 7, [Cited 20 December 2013]. p. 1537–1547. Available from: <http://link.springer.com/10.1007/s11590-012-0568-3>. DOI 10.1007/s11590-012-0568-3.

STRANG, Gilbert. *Linear Algebra*. Second. New York, NY, 1980.

SUBRAMANIAN, Anand, UCHOA, Eduardo, PESSOA, Artur Alves and OCHI, Luiz Satoru. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters* [online]. 21 October 2012. Vol. 7, no. 7, [Cited 20 December 2013]. p. 1569–1581 Available from: <http://link.springer.com/10.1007/s11590-012-0570-9>. DOI 10.1007/s11590-012-0570-9.

TANG, Lixin, WANG, Gongshu and LIU, Jiyin. A branch-and-price algorithm to solve the molten iron allocation problem in iron and steel industry. *Computers & Operations Research* [online]. October 2007. Vol. 34, no. 10, [Cited 17 February

2014]. p. 3001–3015. Available from:  
<http://www.sciencedirect.com/science/article/pii/S0305054805003576>. DOI  
10.1016/j.cor.2005.11.010.

TEBBOTH, James Richard. A Computational Study of Dantzig-Wolfe Decomposition. 2001.

TOLEDO, Sivan. Locality of Reference in LU Decomposition with Partial Pivoting. SIAM Journal on Matrix Analysis and Applications [online]. 31 October 1997. Vol. 18, no. 4, [Cited 13 December 2013]. , p. 1065–1081. Available from:  
<http://epubs.siam.org/doi/abs/10.1137/S0895479896297744>.  
DOI 10.1137/S0895479896297744.

TOTH, Paolo and VIGO, Daniele. Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics [online]. 2002. Vol. 123, no. 1,. [Cited 14 November 2013]. p. 487–512 Available from:  
<http://www.sciencedirect.com/science/article/pii/S0166218X01003511>

ULRICH BLASUM, Winfried Hochstättler. Application of the branch and cut method to the vehicle routing problem. [online]. 2002. [Cited 14 November 2013]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.163.222>

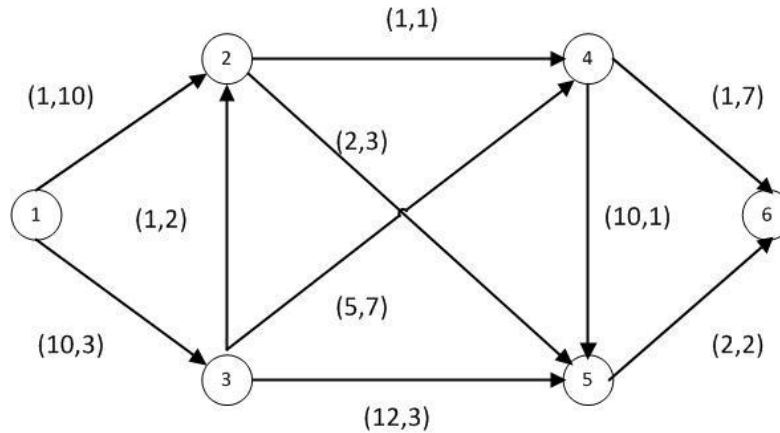
WINSTON, Wayne L. Investigación de Operaciones Aplicaciones y Algoritmos.4.

## ANEXOS

Anexo A. Solución aplicación Branch Bound and Pricing manual para un SPP

Para un problema de la ruta más corta en un grafo mostrado en la figura 14.

Figura 14. Red de Shorest Path Problem



Las etiquetas de cada arco que corresponde a la forma  $(c_{ij}, t_{ij})$  siendo  $c_{ij}$  el costo asociado a cada arco, y  $t_{ij}$  es el tiempo necesario para atravesar el arco  $(i, j)$ . El problema consiste en encontrar la ruta con el mínimo costo de transporte sin incurrir en un tiempo superior a 14 unidades de tiempo, inicialmente se mostrara la formulación sin descomponer.

$$\text{Min } z = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (133)$$

Sujeto a

$$\sum_{j:(1,j) \in A} x_{1j} = 1 \quad (134)$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = 0 \quad i = 2,3,4,5 \quad (135)$$

$$\sum_{i:(i,6) \in A} x_{i6} = 1 \quad (136)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq 14 \quad (137)$$

$$x_{ij} \text{ binarias } (i,j) \in A \quad (138)$$

Como la idea de este ejemplo es ilustrar una aplicación manual se muestra la formulación anterior expandida; el término  $x_{ij}$  representa el uso de cualquier arco que conecta un nodo  $i$  y  $j$ , es decir en el caso de una aplicación computacional es necesario considerar todas las posibles parejas de nodos  $(i,j)$  y en el caso de arcos no conectados se asocia un elevado costo usualmente se conoce como  $M$ .

En este ejemplo no se enumerara todas las posibles parejas de nodos solo se tendrán en cuenta las que se consideran en el grafo de la Figura 4.

$$z = x_{12} + 12x_{13} + x_{32} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46} \quad (139)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (140)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (141)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (142)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (143)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (144)$$

$$x_{56} + x_{46} = 1 \quad (145)$$

$$10x_{12} + 3x_{13} + 2x_{32} + x_{24} + 3x_{35} + 7x_{34} + 3x_{25} + x_{45} + 2x_{56} + 7x_{46} \leq 14 \quad (146)$$

$$x_{ij} \text{ binarias} \quad (147)$$

Ya expandida se observa que es un tablero con 7 restricciones y 17 variables sujetas 10 variables a que sean binarias. La anterior formulación no se va a solucionar solo es para compararlo con la formulación de descomposición que se presenta a continuación.

En el contexto de la descomposición se recomienda que el problema tenga una estructura especial como la mostrada en la figura 1 en esta ocasión el problema no la tiene pero esto no quiere decir que no se pueda aplicar una descomposición, las restricciones (140) – (145) corresponden a las restricciones seleccionadas para incluirlas en los sub-problemas la restricción (146) se asocia al máster.

La siguiente división se obtiene el siguiente máster aumentado

$$\text{Min } z = 0h_1 + MA_1 \quad (148)$$

$$\text{Sujeto a:} \quad (149)$$

$$h_1 = 14 \quad (150)$$

$$A_1 = 1 \quad (151)$$

Para esta aplicación numérica se asume  $M = 100$  con esto la solución del máster es  $h_1 = 14$  ,  $A_1 = 1$  y  $z = 100$ . Las variables duales asociadas a las restricciones son  $\pi_1 = 0$  y  $\pi_0 = 100$  con estos resultados se puede alimentar el sub-problema de la iteración en curso.

$$\bar{c} = (x_{12} + 10x_{13} + x_{32} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46}) - 100 \quad (152)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (153)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (154)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (155)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (156)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (157)$$

$$x_{56} + x_{46} = 1 \quad (158)$$

$$x_{ij} \text{ binarias} \quad (159)$$

La solución del sub-problema sugiere que se agregue una columna al máster que corresponde a un costo reducido  $\bar{c} = -97$  la ruta 1246 con un costo asociado  $c = 3$  y un tiempo  $t = 18$  como es de notar al separar las restricciones en dos problemas distintos las soluciones de los sub-problemas no necesariamente son factibles para el problema original pero al incluir la columna correspondiente generará un mayor costo a la restricción asociada al tiempo en el máster aumentando su peso en el siguiente sub-problema.

Se organiza la nueva columna generada para poder determinar que variable básica debe salir de la base actual del máster que en este momento posee una base de dimensiones  $[2 \times 2]$  muy diferente a la base original que es  $[7 \times 7]$  generando ventajas en el espacio utilizado por iteración.

Tabla 6. Columna 1

C	3	B	R
H	18	14	0.77777778
A	1	1	1

La columna r de la tabla 6 representa un coeficiente que indica que variable debe salir de la base actual como es la primera iteración la base solo está constituida por las variables básicas  $h_1$  que corresponde a una variable de holgura de la primera restricción y  $A_1$  que corresponde a una variable artificial de la segunda restricción, según el coeficiente r la variable que debe salir  $h_1$ , de esta forma el máster se actualiza de la siguiente manera

$$\text{Min } z = 3\lambda_{1246} + 0h_1 + MA_1 \quad (160)$$

Sujeto a

$$18\lambda_{1246} + h_1 = 14 \quad (161)$$

$$\lambda_{1246} + A_1 = 1 \tag{162}$$

Es de resaltar que la base solo está constituida por las variables  $A_1$  y  $\lambda_{1246}$  que resolviendo esta base se obtiene que  $A_1 = 0.22$  ,  $\lambda_{1246} = 0.778$  y  $z = 24.6$ , esta solución corresponde a las variables duales  $\pi_1 = -5.389$  y  $\pi_0 = 100$ . Los valores de las variables duales definen el nuevo sub-problema así

$$\bar{c} = (54.89x_{12} + 26.167x_{13} + 11.778x_{32} + 6.389x_{24} + 28.167x_{35} + 42.723x_{34} + 18.167x_{25} + 15.389x_{45} + 12.778x_{56} + 38.723x_{46}) - 100 \tag{163}$$

Sujeto a

$$x_{12} + x_{13} = 1 \tag{164}$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \tag{165}$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \tag{166}$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \tag{167}$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \tag{168}$$

$$x_{56} + x_{46} = 1 \tag{169}$$

$$x_{ij} \text{ binarias} \tag{170}$$

La solución óptima de este problema con ruta 1356 y  $\bar{c} = -32.888$  esto sugiere que se debe iterar más, el costo de la ruta  $c = 24$  y tiempo  $t = 8$  es de resaltar este resultado ya que puede que su costo sea mayor al de la ruta 1246 pero la ruta 1356 cumple con la restricción de tiempo del problema original.

Como paso siguiente se procede a generar la nueva columna así

Tabla 7. Columna 2

C	24	b	r
$\lambda_{1246}$	8	14	1.75
A	1	1	1

El coeficiente  $r$  de la tabla 7 sugiere que la variable que debe abandonar la base es la artificial de la segunda restricción, de esta manera el máster se define

$$\text{Min } z = 24\lambda_{1356} + 3\lambda_{1246} + 0h_1 + MA_1 \quad (171)$$

Sujeto a

$$8\lambda_{1356} + 18\lambda_{1246} + h_1 = 14 \quad (172)$$

$$\lambda_{1356} + \lambda_{1246} + A_1 = 1 \quad (173)$$

La solución del anterior problema es  $\lambda_{1246} = 0.6$  ,  $\lambda_{1356} = 0.4$  y  $z = 11.4$ , este resultado tiene asociado las variables duales  $\pi_1 = -2.1$  y  $\pi_0 = 40.8$  alimentando el nuevo sub-problema a resolver.

$$\bar{c} = (22x_{12} + 16.3x_{13} + 5.2x_{32} + 3.1x_{24} + 18.3x_{35} + 19.7x_{34} + 8.3x_{25} + 12.1x_{45} + 6.2x_{56} + 15.7x_{46}) - 40.8 \quad (174)$$

Sujeto a:

$$x_{12} + x_{13} = 1 \quad (175)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (176)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (177)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (178)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (179)$$

$$x_{56} + x_{46} = 1 \quad (180)$$

$$x_{ij} \text{ binarias} \quad (181)$$

La solución de este sub-problema corresponde a la ruta 13256 con costo reducido  $\bar{c} = -4.8$  con un costo de transporte  $c = 15$ , un tiempo de transporte  $t = 10$  y así se establece la nueva columna.

Tabla 8. Columna 3

C	15	B	r
$\lambda_{1246}$	10	14	1.4
$\lambda_{1356}$	1	1	1

El coeficiente r de la tabla 8 sugiere que la variable  $\lambda_{1356}$  debe abandonar la base y debe ingresar una nueva variable en su lugar  $\lambda_{13256}$  modificando el máster.

$$\text{Min } z = 15\lambda_{13256} + 24\lambda_{1356} + 3\lambda_{1246} + 0h_1 + MA_1 \quad (182)$$

Sujeto a

$$10\lambda_{13256} + 8\lambda_{1356} + 18\lambda_{1246} + h_1 = 14 \quad (183)$$

$$\lambda_{13256} + \lambda_{1356} + \lambda_{1246} + A_1 = 1 \quad (184)$$

La solución de este problema es  $z = 9$ ,  $\lambda_{13256} = \lambda_{1246} = 0.5$  esta solución tiene asociadas dos variables duales  $\pi_1 = -1.5$  y  $\pi_0 = 30$  las cuales alimentan el siguiente sub-problema.

$$\bar{c} = (16x_{12} + 14.5x_{13} + 4x_{32} + 2.5x_{24} + 16.5x_{35} + 15.5x_{34} + 6.5x_{25} + 11.5x_{45} + 5x_{56} + 11.5x_{46}) - 30 \quad (185)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (186)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (187)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (188)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (189)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (190)$$

$$x_{56} + x_{46} = 1 \quad (191)$$

$$x_{ij} \text{ binarias} \quad (192)$$

Se propone una nueva ruta 1256 con un costo reducido  $\bar{c} = -2.5$ , un costo de transporte de  $c = 5$  y un tiempo  $t = 15$ . Este resultado genera una nueva columna de la forma.

Tabla 9. Columna 4

c	5	b	r
$\lambda_{1246}$	15	14	0.93333333
$\lambda_{13256}$	1	1	1

El coeficiente r de la columna 4 sugiere que la variable que debe abandonar la base es  $\lambda_{1246}$  esta es remplazada por  $\lambda_{1256}$  cambiando nuevamente el máster.

$$\text{Min } z = 5\lambda_{1256} + 15\lambda_{13256} + 24\lambda_{1356} + 3\lambda_{1246} + 0h_1 + MA_1 \quad (193)$$

Sujeto a

$$15\lambda_{1256} + 10\lambda_{13256} + 8\lambda_{1356} + 18\lambda_{1246} + h_1 = 14 \quad (194)$$

$$\lambda_{1256} + \lambda_{13256} + \lambda_{1356} + \lambda_{1246} + A_1 = 1 \quad (195)$$

Se obtienen como resultados  $z = 7$ ,  $\lambda_{1256} = 0.8$  y  $\lambda_{13256} = 0.2$  con variables duales asociada  $\pi_1 = -2$  y  $\pi_0 = 35$  que modifican el sub-problema así

$$\bar{c} = (21x_{12} + 16x_{13} + 5x_{32} + 3x_{24} + 18x_{35} + 19x_{34} + 8x_{25} + 12x_{45} + 6x_{56} + 15x_{46}) - 35 \quad (196)$$

Sujeto a:

$$x_{12} + x_{13} = 1 \quad (197)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (198)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (199)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (200)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (201)$$

$$x_{56} + x_{46} = 1 \quad (202)$$

$$x_{ij} \text{ binarias} \quad (203)$$

El costo reducido óptimo para este arreglo es  $\bar{c} = 0$  lo cual indica que no se puede mejorar más. En términos de  $x_{ij}$  se da a continuación.

$$x_{12} = 0.8$$

$$x_{13} = 0.2$$

$$x_{32} = 0.2$$

$$x_{25} = 1$$

$$x_{56} = 1$$

El problema de la ruta más corta al igual que los problema de ruteo de transporte hacen parte de los problemas de programación lineal entera, hasta este momento solo se ha solucionado la versión relajada del problema.

Este ejemplo propone que sea solucionado utilizando Branch and Bound de igual manera se puede solucionar usando Branch and Cut, solo que como es una aplicación manual se utiliza BB con una estrategia de ramificación que disminuye la cantidad de nodos a ramificar.

Estrategia de Ramificación: Por la forma de la solución solo se debe ramificar las variables  $x_{12}, x_{13}, x_{32}$  ya que son las variables que no cumplen con la condición de enteras pero por el grafo se puede notar que  $x_{13}, x_{32}$  dependen entre si y es independiente de  $x_{12}$ , por esto se establecen dos árboles de BB el primero se ramifica con respecto a  $x_{13} + x_{32}$  este nodo tiene dos ramas la primera es  $x_{13} + x_{32} = 0$  y la segunda  $x_{13} + x_{32} \geq 1$ .

Para el segundo árbol se ramifica a partir de  $x_{12}$  con dos ramas, la primera es  $x_{12} = 0$  y la segunda  $x_{12} = 1$ .

Rama  $x_{13} + x_{32} = 0$

En el Branch and cut como en el Branch and bound tanto como los cortes como las ramas se agregan al máster no a los sub-problemas, pero en este caso para que se cumpla el criterio de la rama tanto  $x_{13}$  como  $x_{32}$  deben ser iguales a 0, por esto se pueden eliminar del sub-problema pero normalmente no se recomienda este tipo de prácticas para instancias computacionales.

Máster

$$\text{Min } z = 0h_1 + MA_1 + MA_2 \quad (204)$$

Sujeto a:

$$h_1 = 14 \quad (205)$$

$$A_1 = 1 \quad (206)$$

$$A_2 = 0 \quad (207)$$

Solución Máster

$z = 1000$  esta solución tiene las variables duales  $\pi_1 = 0$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$  asociadas a las restricciones, es de resaltar que tanto en esta iteración como en las posteriores se aumenta el valor de  $M = 1000$ .

Sub- problema

$$\bar{c} = (x_{12} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46}) - 1000 \quad (208)$$

Sujeto a

$$x_{12} = 1 \quad (209)$$

$$x_{12} - x_{24} - x_{25} = 0 \quad (210)$$

$$-x_{35} - x_{34} = 0 \quad (211)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (212)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (213)$$

$$x_{56} + x_{46} = 1 \quad (214)$$

$$x_{ij} \text{ binarias} \quad (215)$$

### Solución Sub-problema

La ruta correspondiente es la 1246 con un costo reducido  $\bar{c} = -997$  un costo asociado  $c = 3$  y un tiempo  $t = 18$  con estos datos se genera la nueva columna:

Tabla 10. Columna 1 rama 1

C	3	B	R
H	18	14	0.77777778
A1	1	1	1
A2	0	0	-

Esto sugiere que la variable que debe abandonar la base es la variable de holgura que corresponde a la primera restricción.

### Máster

$$\text{Min } z = 3\lambda_{1246} + 0h_1 + MA_1 + MA_2 \quad (216)$$

### Sujeto a

$$18\lambda_{1246} + h_1 = 14 \quad (217)$$

$$\lambda_{1246} + A_1 = 1 \quad (218)$$

$$0\lambda_{1246} + A_2 = 0 \quad (219)$$

### Solución Máster

$z = 224,556$  este valor corresponde a la solución  $\lambda_{1246} = 0.778$ ,  $A_1 = 0.22$  y  $A_2 = 0$ , con variables duales asociadas  $\pi_1 = -55.389$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$ .

Sub-problema

$$\bar{c} = (554.89x_{12} + 56.389x_{24} + 178.167x_{35} + 392.723x_{34} + 168.167x_{25} + 65.389x_{45} + 112.778x_{56} + 388.723x_{46}) - 1000 \quad (220)$$

Sujeto a

$$x_{12} = 1 \quad (221)$$

$$x_{12} - x_{24} - x_{25} = 0 \quad (222)$$

$$-x_{35} - x_{34} = 0 \quad (223)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (224)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (225)$$

$$x_{56} + x_{46} = 1 \quad (226)$$

$$x_{ij} \text{ binarias} \quad (227)$$

Solución Sub-problema

La ruta 1256 con un costo reducido  $\bar{c} = -164.165$  con un costo de transporte  $c = 5$  y un tiempo  $t = 15$  con estos valores se genera la nueva columna.

Tabla 11. Columna 2 rama 1

C	5	B	R
$\lambda_{1246}$	15	14	0.93333333
A1	1	1	1
A2	0	0	-

El coeficiente r sugiere que la variable que debe abandonar la base es  $\lambda_{1246}$ .

Máster

$$\text{Min } z = 5\lambda_{1256} + 3\lambda_{1246} + 0h_1 + MA_1 + MA_2 \quad (228)$$

Sujeto a

$$15\lambda_{1256} + 18\lambda_{1246} + h_1 = 14 \quad (229)$$

$$\lambda_{1256} + \lambda_{1246} + A_1 = 1 \quad (230)$$

$$0\lambda_{1256} + 0\lambda_{1246} + A_2 = 0 \quad (231)$$

### Solución Máster

$z = 71.33$  con  $\lambda_{1246} = 0.933$  y  $A_1 = 0.067$  relacionadas con las variables duales  $\pi_1 = -66.333$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$ .

### Sub-problema

$$\bar{c} = (664.33x_{12} + 67.333x_{24} + 210.999x_{35} + 469.331x_{34} + 200.999x_{25} + 76.333x_{45} + 134.666x_{56} + 465.331x_{46}) - 1000 \quad (232)$$

### Sujeto a

$$x_{12} = 1 \quad (233)$$

$$x_{12} - x_{24} - x_{25} = 0 \quad (234)$$

$$-x_{35} - x_{34} = 0 \quad (235)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (236)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (237)$$

$$x_{56} + x_{46} = 1 \quad (238)$$

$$x_{ij} \text{ binarias} \quad (239)$$

### Solución Sub-problema

$\bar{c} = -57.338$  con la ruta 12456 con un costo asociado  $c = 14$  y un tiempo  $t = 14$ .

Tabla 12. Columna 3 rama 1

C	14	B	r
$\lambda_{1256}$	14	14	1
A1	1	1	1
A2	0	0	-

### Máster

$$\text{Min } z = 14\lambda_{12456} + 5\lambda_{1256} + 3\lambda_{1246} + 0h_1 + MA_1 + MA_2 \quad (240)$$

Sujeto a

$$14\lambda_{12456} + 15\lambda_{1256} + 18\lambda_{1246} + h_1 = 14 \quad (241)$$

$$\lambda_{12456} + \lambda_{1256} + \lambda_{1246} + A_1 = 1 \quad (242)$$

$$0\lambda_{12456} + 0\lambda_{1256} + 0\lambda_{1246} + A_2 = 0 \quad (243)$$

Solución Máster

$z = 14$  con  $\lambda_{12456} = 1$  y variables asociadas  $\pi_1 = -9$ ,  $\pi_0 = 140$  y  $\pi_2 = 1000$ .

Sub-problema

$$\bar{c} = (91x_{12} + 10x_{24} + 39x_{35} + 68x_{34} + 29x_{25} + 19x_{45} + 20x_{56} + 64x_{46}) - 140$$

Sujeto a

$$x_{12} = 1 \quad (244)$$

$$x_{12} - x_{24} - x_{25} = 0 \quad (245)$$

$$-x_{35} - x_{34} = 0 \quad (246)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (247)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (248)$$

$$x_{56} + x_{46} = 1 \quad (249)$$

$$x_{ij} \text{ binarias} \quad (250)$$

Solución Sub-problema

$\bar{c} = 0$  con esto se termina las iteraciones en la rama con solución en términos de  $x_{ij}$  así  $x_{12} = x_{24} = x_{45} = x_{56} = 1$  cumpliendo la condición de entera, con un costo asociado  $c = 14$  y un tiempo  $t = 14$ .

Rama  $x_{13} + x_{32} \geq 1$

Máster

$$\text{Min } z = 0h_1 + MA_1 + 0h_2 \quad (251)$$

Sujeto a

$$h_1 = 14 \quad (252)$$

$$A_1 = 1 \quad (253)$$

$$h_2 \geq 1 \quad (254)$$

Solución Máster

$z = 1000$  con variables duales asociadas  $\pi_1 = 0$ ,  $\pi_0 = 1000$  y  $\pi_2 = 0$ .

Sub-problema

$$\bar{c} = (x_{12} + 10x_{13} + x_{32} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46}) - 1000 - (0 * (x_{13} + x_{32})) \quad (255)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (256)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (257)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (258)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (259)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (260)$$

$$x_{56} + x_{46} = 1 \quad (261)$$

$$x_{ij} \text{ binarias} \quad (262)$$

Solución Sub-problema

$\bar{c} = -997$  con la ruta 1246 con un costo asociado  $c = 3$  y tiempo  $t = 18$ .

Tabla 13. Columna 1 Rama 2

C	3	B	r
H	18	14	0.77777778
A1	1	1	1
h2	0	1	-

Máster

$$\text{Min } z = 3\lambda_{1246} + 0h_1 + MA_1 + 0h_2 \quad (263)$$

Sujeto a

$$18\lambda_{1246} + h_1 = 14 \quad (264)$$

$$\lambda_{1246} + A_1 = 1 \quad (265)$$

$$h_2 \geq 1 \quad (266)$$

Solución del Máster

$z = 224.556$  con  $\lambda_{1246} = 0.778$   $h_2 = 1$   $A_1 = 0.222$  y variables duales asociadas  $\pi_1 = -55.389$ ,  $\pi_0 = 1000$  y  $\pi_2 = 0$ .

Sub-problema

$$\begin{aligned} \bar{c} = & (554.89x_{12} + 176.167x_{13} + 111.778x_{32} + 56.389x_{24} + 178.167x_{35} + \\ & 392.723x_{34} + 168.167x_{25} + 65.389x_{45} + 112.778x_{56} + 388.723x_{46}) - 1000 - (0 * \\ & (x_{13} + x_{32})) \end{aligned} \quad (267)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (268)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (269)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (270)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (271)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (272)$$

$$x_{56} + x_{46} = 1 \quad (273)$$

$$x_{ij} \text{ binarias} \quad (274)$$

Solución Sub-problema

$\bar{c} = -532.888$  corresponde a la ruta 1356 con un costo asociado  $c = 24$  y un tiempo  $t = 8$ .

Tabla 14. Columna 2 rama 2

C	24	b	R
$\lambda_{1246}$	8	14	1.75
A1	1	1	1
h2	1	1	1

Máster

$$\text{Min } z = 24\lambda_{1356} + 3\lambda_{1246} + 0h_1 + MA_1 + 0h_2 \quad (275)$$

Sujeto a

$$8\lambda_{1356} + 18\lambda_{1246} + h_1 = 14 \quad (276)$$

$$\lambda_{1356} + \lambda_{1246} + A_1 = 1 \quad (277)$$

$$\lambda_{1356} + h_2 \geq 1 \quad (278)$$

Solución Máster

$$z = 11.4 \text{ con } \lambda_{1356} = 0.6 \lambda_{1246} = 0.4 h_2 = 0.6 \text{ y variables duales asociadas } \pi_1 = -2.1, \pi_0 = 40.8 \text{ y } \pi_2 = 0$$

Sub-problema

$$\bar{c} = (22x_{12} + 16.3x_{13} + 5.2x_{32} + 3.1x_{24} + 18.3x_{35} + 19.7x_{34} + 8.3x_{25} + 12.1x_{45} + 6.2x_{56} + 15.7x_{46}) - 40.8 - (0 * (x_{13} + x_{32})) \quad (279)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (280)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (281)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (282)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (283)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (284)$$

$$x_{56} + x_{46} = 1 \quad (285)$$

$$x_{ij} \text{ binarias} \quad (286)$$

### Solución Sub-problema

$\bar{c} = -4.8$  con la ruta 13256 con un costo asociado  $c = 15$  y un tiempo  $t = 10$

Tabla 15. Columna 3 rama 2

C	15	B	r
$\lambda_{1246}$	10	14	1.4
$\lambda_{1356}$	1	1	1
h2	2	1	0.5

### Máster

$$\text{Min } z = 15\lambda_{13256} + 24\lambda_{1356} + 3\lambda_{1246} + 0h_1 + MA_1 + 0h_2 \quad (287)$$

### Sujeto a

$$10\lambda_{13256} + 8\lambda_{1356} + 18\lambda_{1246} + h_1 = 14 \quad (288)$$

$$\lambda_{13256} + \lambda_{1356} + \lambda_{1246} + A_1 = 1 \quad (289)$$

$$2\lambda_{13256} + \lambda_{1356} + h_2 \geq 1 \quad (290)$$

### Solución del Máster

$z = 9$  con  $\lambda_{13256} = 0.5$   $\lambda_{1246} = 0.5$  con variables duales asociadas  $\pi_1 = -1.5$ ,  $\pi_0 = 30$  y  $\pi_2 = 0$

### Sub-problema

$$\bar{c} = (16x_{12} + 14.5x_{13} + 4x_{32} + 2.5x_{24} + 16.5x_{35} + 15.5x_{34} + 6.5x_{25} + 11.5x_{45} + 5x_{56} + 11.5x_{46}) - 30 - (0 * (x_{13} + x_{32})) \quad (291)$$

### Sujeto a

$$x_{12} + x_{13} = 1 \quad (292)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (293)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (294)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (295)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (296)$$

$$x_{56} + x_{46} = 1 \quad (297)$$

$$x_{ij} \text{ binarias} \quad (298)$$

Solución Sub-problema

$\bar{c} = -2.5$  corresponde a la ruta 1256 con costo asociado  $c = 5$  y tiempo  $t = 15$ .

Tabla 16. Columna 4 rama 2

C	5	B	r
$\lambda_{1246}$	15	14	0.93333333
$\lambda_{1356}$	1	1	1
$\lambda_{13256}$	0	1	-

Máster

$$\text{Min } z = 5\lambda_{1256} + 15\lambda_{13256} + 24\lambda_{1356} + 3\lambda_{1246} + 0h_1 + MA_1 + 0h_2 \quad (299)$$

Sujeto a

$$15\lambda_{1256} + 10\lambda_{13256} + 8\lambda_{1356} + 18\lambda_{1246} + h_1 = 14 \quad (300)$$

$$\lambda_{1256} + \lambda_{13256} + \lambda_{1356} + \lambda_{1246} + A_1 = 1 \quad (301)$$

$$2\lambda_{13256} + \lambda_{1356} + h_2 \geq 1 \quad (302)$$

Solución Máster

$z = 5.33$  con  $\lambda_{1256} = 0.667$   $\lambda_{13256} = 0.667$   $\lambda_{1356} = -0.333$  y variables duales asociadas  $\pi_1 = -3.111$ ,  $\pi_0 = 51.667$  y  $\pi_2 = -2.778$ .

Sub-problema

$$\begin{aligned} \bar{c} = & (32.11x_{12} + 19.333x_{13} + 7.222x_{32} + 4.111x_{24} + 21.333x_{35} + 26.777x_{34} + \\ & 11.333x_{25} + 13.111x_{45} + 8.222x_{56} + 22.777x_{46}) - 51.667 - (-2.778 * \\ & (x_{13} + x_{32})) \end{aligned} \quad (303)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (304)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (305)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (306)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (307)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (308)$$

$$x_{56} + x_{46} = 1 \quad (309)$$

$$x_{ij} \text{ binarias} \quad (310)$$

Solución Sub-problema

$\bar{c} = -0.002$  este resultado se puede considerar como 0 ya que es producto de errores en el redondeo de decimales.

Rama  $x_{12} = 0$

Máster

$$\text{Min } z = 0h_1 + MA_1 + MA_2 \quad (311)$$

Sujeto a

$$h_1 = 14 \quad (312)$$

$$A_1 = 1 \quad (313)$$

$$A_2 = 0 \quad (314)$$

Solución del Máster

$z = 1000$  con las variables duales asociadas  $\pi_1 = 0$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$

Sub-problema

$$\bar{c} = (x_{12} + 10x_{13} + x_{32} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46}) - 1000 - 1000(x_{12}) \quad (315)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (316)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (317)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (318)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (319)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (320)$$

$$x_{56} + x_{46} = 1 \quad (321)$$

$$x_{ij} \text{ binarias} \quad (322)$$

Solución del Sub-problema

$\bar{c} = -1987$  con la ruta 13246 con un costo asociado  $c = 13$  y un tiempo  $t = 13$ .

Tabla 17. Columna 1 rama 3

C	13	B	R
H	13	14	1.07692308
A1	1	1	1
A2	0	0	-

Máster

$$\text{Min } z = 13\lambda_{13246} + 0h_1 + MA_1 + MA_2 \quad (323)$$

Sujeto a

$$13\lambda_{13246} + h_1 = 14 \quad (324)$$

$$1\lambda_{13246} + A_1 = 1 \quad (325)$$

$$0\lambda_{13246} + A_2 = 0 \quad (326)$$

Solución del Máster

$z = 13$  con  $\lambda_{13246} = 1$  y las variables duales asociadas  $\pi_1 = 0$ ,  $\pi_0 = 13$  y  $\pi_2 = 1000$ .

Sub-problema

$$\bar{c} = (x_{12} + 10x_{13} + x_{32} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46}) - 13 - 1000(x_{12}) \quad (327)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (328)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (329)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (330)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (331)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (332)$$

$$x_{56} + x_{46} = 1 \quad (334)$$

$$x_{ij} \text{ binarias} \quad (335)$$

Solución del Sub-problema

$\bar{c} = 0$  lo que indica que no se puede mejorar más la solución actual que es  $x_{13} = x_{32} = x_{24} = x_{46} = 1$  con un costo asociado  $c = 13$  y un tiempo  $t = 13$ .

Rama  $x_{12} = 1$

Máster

$$\text{Min } z = 0h_1 + MA_1 + MA_2 \quad (336)$$

Sujeto a

$$h_1 = 14 \quad (337)$$

$$A_1 = 1 \quad (338)$$

$$A_2 = 1 \quad (339)$$

Solución del máster

$z = 2000$  con variables duales asociadas  $\pi_1 = 0$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$ .

Sub-problema:

$$\bar{c} = (x_{12} + 10x_{13} + x_{32} + x_{24} + 12x_{35} + 5x_{34} + 2x_{25} + 10x_{45} + 2x_{56} + x_{46}) - 1000 - 1000(x_{12}) \quad (340)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (341)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (342)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (343)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (344)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (345)$$

$$x_{56} + x_{46} = 1 \quad (346)$$

$$x_{ij} \text{ binarias} \quad (347)$$

Solución Sub-problema

$\bar{c} = -1997$  correspondiente a la ruta 1246 con un costo asociado  $c = 3$  y un tiempo  $t = 18$ .

Tabla 18. Columna 1 rama 4

C	3	b	R
H	18	14	0.77777778
A1	1	1	1
A2	1	1	1

Máster:

$$\text{Min } z = 3\lambda_{1246} + 0h_1 + MA_1 + MA_2 \quad (348)$$

Sujeto a

$$18\lambda_{1246} + h_1 = 14 \quad (349)$$

$$\lambda_{1246} + A_1 = 1 \quad (350)$$

$$\lambda_{1246} + A_2 = 1 \quad (351)$$

Solución del Máster

$z = 440.778$  con  $\lambda_{1246} = 0.778$ ,  $A_1 = 0.22$ ,  $A_2 = 0.22$  y variables duales asociadas  $\pi_1 = -110.94$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$ .

Sub-problema

$$\bar{c} = (110.4x_{12} + 342.82x_{13} + 222.88x_{32} + 111.94x_{24} + 344.82x_{35} + 781.58x_{34} + 334.82x_{25} + 120.94x_{45} + 223.88x_{56} + 777.58x_{46}) - 1000 - 1000(x_{12}) \quad (352)$$

Sujeto a

$$x_{12} + x_{13} = 1 \quad (353)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (354)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (355)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (356)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (357)$$

$$x_{56} + x_{46} = 1 \quad (358)$$

$$x_{ij} \text{ binarias} \quad (359)$$

Solución Sub-problema

$\bar{c} = -330.840$  correspondiente a la ruta 1256 con costo  $c = 5$  y tiempo  $t = 15$ .

Tabla 19. Columna 2 rama 4

C	5	b	R
$\lambda_{1246}$	15	14	0.93333333
A1	1	1	1
A2	1	1	1

Máster

$$\text{Min } z = 5\lambda_{1256} + 3\lambda_{1246} + 0h_1 + MA_1 + MA_2 \quad (360)$$

Sujeto a

$$15\lambda_{1256} + 18\lambda_{1246} + h_1 = 14 \quad (361)$$

$$\lambda_{1256} + \lambda_{1246} + A_1 = 1 \quad (362)$$

$$\lambda_{1256} + \lambda_{1246} + A_2 = 1 \quad (363)$$

### Solución del Máster

$z = 138$  con  $\lambda_{1256} = 0.933$ ,  $A_1 = 0.067$ ,  $A_2 = 0.067$  y variables duales asociadas  $\pi_1 = -133$ ,  $\pi_0 = 1000$  y  $\pi_2 = 1000$ .

### Sub-problema

$$\bar{c} = (1331x_{12} + 409x_{13} + 267x_{32} + 134x_{24} + 411x_{35} + 936x_{34} + 401x_{25} + 143x_{45} + 268x_{56} + 932x_{46}) - 1000 - 1000(x_{12}) \quad (364)$$

### Sujeto a

$$x_{12} + x_{13} = 1 \quad (365)$$

$$x_{12} + x_{32} - x_{24} - x_{25} = 0 \quad (366)$$

$$x_{13} - x_{32} - x_{35} - x_{34} = 0 \quad (367)$$

$$x_{24} + x_{34} - x_{45} - x_{46} = 0 \quad (368)$$

$$x_{35} + x_{25} + x_{45} - x_{56} = 0 \quad (369)$$

$$x_{56} + x_{46} = 1 \quad (370)$$

$$x_{ij} \text{ binarias} \quad (371)$$

### Solución del Sub-problema

$\bar{c} = 0$  indica que la solución actual no puede ser mejorada la cual no es entera.

## Anexo B. Guía de Ejecución de códigos

Dentro de este documento se encuentra la aplicación métodos exactos en los cuales se utilizaron librerías GDX en Matlab, para utilizar los códigos presentados en este trabajo se recomienda:

Disponer de las licencias tanto de GAMS como de Matlab.

Las versiones de los programas deben ser para el mismo tipo de sistema operativo es decir si se tiene un sistema operativo de 64 bits se recomienda que las versiones tanto de GAMS como de Matlab sean de 64 bits, dado que si no es así las librerías presentan errores cuando se ejecutan.

Se recomienda utilizar versiones de Matlab del 2010 o posteriores a esta.

Agregar la ruta de la carpeta de la versión de GAMS que disponga.

Los códigos presentados están compuestos un archivo .m que corresponde a una script y uno o más archivos .gms que corresponden a sub-rutinas de GAMS para poder ejecutar los códigos presentados se sugiere el siguiente procedimiento:

- Copiar los archivos .gms correspondientes al código presentado en el GAMSdir/projdir.
- Copiar la ruta de esta carpeta.
- Abrir el script.
- Dentro del script modificar la primera línea de código que tiene la sentencia cd, borrando la ruta anterior y colocando la ruta copiada en el paso 2.

## Anexo C. Script del método exacto completo sin descomponer

```
%Sub-rutina correspondiente a la solución de un problema de ruteo de vehículos
%con restricciones de capacidad dentro de un algoritmo exacto
%enumerando parcialmente el conjunto de sub-tours
%Autor: DANIEL ORLANDO MARTÍNEZ QUEZADA
tic;
cd C:\Users\EEIE\Documents\GAMSdir\projdir;
%DEFINIR UBICACION DEL GAMDIR
clear all;
%LIMPIAR VARIABLES ANTERIORES
%DEFINIR PARAMETROS DEL METODO
CARMAX=15;
%DEFINIR LAS ENTRADAS DEL PROBLEMA
CAMIONES=8;
NODOS=16;
POSICION=[ 30 40; 37 52; 49 49; 52 64; 31 62; 52 33; 42 41; 52 41; 57 58; 62 42;
42 57;27 68; 43 67; 58 48; 58 27;37 69];
DEMANDAS=[1 0;2 19;3 30;4 16;5 23;6 11;7 31;8 15;9 28;10 8;11 8;12 7;13 14;14
6;15 19;16 11];
CAPACIDAD=35;

NS=1;
for C=1:CARMAX
NS=NS+(nchoosek(NODOS-1,C));
End
%CREAR EL CONJUNTO
for C=1:NS
S(C,1)=C;
end
```

```

I(1)=1;
for C=2:NODOS
I(C,1)=C;
V(C-1,1)=C;
end
A=0;
for C=1:NODOS
for D=1:2
A=A+1;
P(A,1)=C;
P(A,2)=D;
P(A,3)=POSICION(C,D);
end
end

writegdx('CVRP_ENTRADAS.gdx','set','i',I,'set','v',V,'set','n',S,'parameter','p',P,'parameter','d',DEMANDAS,'parameter','k',CAMIONES,'parameter','c',CAPACIDAD,'parameter','cor',CARMAX);
GAMS('CVRP_SOLVE.gms');
Xo=readgdx('SOLVE_CVRP.gdx','xo');
Zo=readgdx('SOLVE_CVRP.gdx','zo');
toc;

```

## Anexo D. GMS del método exacto sin descomponer

\*Solucion del CVRP en una formulación entera completa

\*Autor: DANIEL ORLANDO MARTÍNEZ QUEZADA

set i;

set v(i);

set d /1\*2/;

set n;

parameter c ;

parameter p(i,d);

parameter de;

parameter k;

parameter cor ;

\$gdxin CVRP\_ENTRADAS.gdx

\$load i=i

\$load v=v

\$load n=n

\$load p=p

\$load de=d

\$load k=k

\$load c=c

\$load cor=cor

set e(i,i);

alias(i,j);

e(i,j)\$(not sameas(i,j))=yes;

\*conjuntos auxiliares

set S(n,v) conjunto de subtours;

\*inicializar conjunto S

set vi(v) cliente inicial;

vi(v)\$(ord(v)=1)=yes;

```

set w(v) clientes sin el inicial;
w(v)= not vi(v);
*primera iteracion
S("1",vi)=no;
S("2",vi)=yes;
set actual(n) sub-conjunto actual de S;
actual("2")=yes;
set hecho(n) todo n-esimo S recorrido;
hecho("1")=yes;
hecho("2")=yes;
set lo(n) "control de ciclo'hecho";
parameter cora;
*Bucle
loop(w,
lo(n)=hecho(n);
loop(lo,
cora=sum(S(lo,v),1);
if((cora<cor),
actual(n)= actual(n-1);
S(actual,v)=S(lo,v);
S(actual,w)=yes;
hecho(actual)=yes;
else
hecho(actual)=no;
);
);
);
parameter Sciclo(n);
Sciclo(n)=sum(S(n,v),1);
set sub(n);

```

```

sub(n)$(Sciclo(n)>=2)=yes;
alias(v,nv);
set nS(n,i,j);
nS(sub(n),e(v,nv))$(S(n,v) and S(n,nv))=yes;
*fin de conjuntos auxiliares
parameter l(i,j);
l(e(i,j))= sqrt(power((p(i,"1")-p(j,"1")),2)+power((p(i,"2")-p(j,"2")),2));
binary variable x(i,j);
parameter r(n);
r(n)=((sum((v)$(S(n,v)),de(v)))/c);

variable Z;
equations
fo función objetivo
flujo(i) restriccion de unica entrada y unica salida
flujo2(i) restriccion de unica entrada y unica salida
camiones1 salida de camiones del depot
camiones2 entrada de camiones al depot
subtour(n) restriccion de subtours;

fo.. Z=e=sum(e(i,j),l(i,j)*x(i,j));
flujo(i)$v(i).. sum{j$(ord(i) ne ord(j)),x[i,j]}=e= 1;
flujo2(i)$v(i).. sum{j$(ord(i) ne ord(j)),x[j,i]}=e= 1;
camiones1..sum(i$v(i),x('1',i))=e=k;
camiones2..sum(i$v(i),x(i,'1'))=e=k;
subtour(n).. sum(nS(n,i,j),x(i,j))=l=Sciclo(n)-r(n) ;
model CVRP /all/;
solve CVRP minimizing Z using MIP;
$set matout "'SOLVE_CVRP.gdx', Xo, Zo";
variable Xo(i,j);

```

```
variable Zo;  
Xo.l(i,j)=X.l(i,j);  
Zo.l=Z.l;  
execute_unload %matout%;
```

## Anexo E. Script del método de descomposición D-W

```
%Rutina para aplicar un algoritmo de descomposición de Dantzig-Wolfe
%a un problema de ruteo de vehiculos con restricciones de capacidad
%Autor: DANIEL ORLANDO MARTINEZ QUEZADA

tic;
cd C:\Users\EEIE\Documents\GAMSdir\projdir;
%DEFINIR UBICACION DEL GAMDIR
clear all;
%LIMPIAR VARIABLES ANTERIORES
%DEFINIR PARAMETROS DEL METODO
CARMAX=2;
%DEFINIR LAS ENTRADAS DEL PROBLEMA
CAMIONES=2;
NODOS=19;
POSICION=[ 30 40; 37 52; 49 43; 52 64; 31 62; 52 33; 42 41; 52 41; 57 58; 62 42;
42 57; 27 68; 43 67; 58 27; 37 69; 61 33; 62 63; 63 69; 45 35 ];

DEMANDAS=[1 0;2 19;3 30;4 16;5 23;6 11;7 31;8 15;9 28;10 14;11 8;12 7;13
14;14 19;15 11;16 26;17 17;18 6;19 15];
CAPACIDAD=160;
%ESTABLECER LAS CONDICIONES DEL MÁSTER
LD=[CAMIONES;CAMIONES;1];
NB=3;
BASE=eye(NB);
M=999;
OC=[M M M];
%ADECUAR PARA FORMATO GDX
for C=1:NB
NI(C,1)=C;
```

```

NC(C,1)=C;
NC(C,2)=OC(C);
NLD(C,1)=C;
NLD(C,2)=LD(C);
end
A=0;
for C=1:NB
for D=1:NB
A=A+1;
NBA(A,1)=C;
NBA(A,2)=D;
NBA(A,3)=BASE(C,D);
end
end

%INCLUIR EN GDX
writegdx('MásterA.gdx','set','i',NI,'parameter','ladod',NLD);
writegdx('MásterB.gdx','parameter','base',NBA,'parameter','c',NC);
GAMS('MÁSTER');
DUALES =readgdx('CVRP_MÁSTER.gdx','dua');
%EJECUTAR MÁSTER
NS=1;
for C=1:CARMAX
NS=NS+(nchoosek(NODOS-1,C));
end
%CREAR EL CONJUNTO
for C=1:NS
    S(C,1)=C;
end

```

```

I(1,1)=1;
for C=2:NODOS

I(C,1)=C;
V(C-1,1)=C;

end
A=0;
for C=1:NODOS
for D=1:2
A=A+1;
P(A,1)=C;
P(A,2)=D;
P(A,3)=POSICION(C,D);
end
end

```

```

%EXTRAER DUALES PROPORCIONADAS DEL MÁSTER

```

```

DUAL1=0;
DUAL2=0;
DUALEST=size(DUALES);
DUAL0=0;
for C=1:DUALEST(1,1)
NARUT=DUALES(C,1);
if(NARUT<2)
DUAL1=DUALES(C,2);
else
if(NARUT<3)
DUAL2=DUALES(C,2);
else

```

```
DUAL0=DUALES(C,2);
```

```
end
```

```
end
```

```
end
```

```
% INCLUIR DUALES A GDX
```

```
writgdx('PricingB.gdx','parameter','dual1',DUAL1,'parameter','dual2',DUAL2,'parameter','dual0',DUAL0);
```

```
%INCLUIR DATOS A GDX
```

```
writgdx('PricingA.gdx','set','i',I,'set','v',V,'set','n',S,'parameter','d',DEMANDAS,'parameter','c',CAPACIDAD,'parameter','car',CARMAX,'parameter','p',P);
```

```
GAMS('Pricing.gms');
```

```
FULL=readgdx('CVRP-PRICING','full');
```

```
MET= readgdx('CVRP-PRICING','met');
```

```
COSTOP= readgdx('CVRP-PRICING','costop');
```

```
CRO=readgdx('CVRP-PRICING','cro');
```

```
Xo=readgdx('CVRP-PRICING','xo');
```

```
iter=0;
```

```
IL=zeros(NODOS*2,3);
```

```
JL=zeros(NODOS*2,3);
```

```
XL=zeros(NODOS*2,3);
```

```
while(CRO<0)
```

```
% ESTABLECER LA NUEVA COLUMNA ENTRANTE A LA BASE DEL MÁSTER
```

```
RNCOLUMNA=zeros(NB,1);
```

```

RNCOLUMNA(1,1)=FULL;
RNCOLUMNA(2,1)=MET;
RNCOLUMNA(NB,1)=1;

RATIO=LD./RNCOLUMNA;
[MINI LRATIO]=min(RATIO(RATIO>=0));
BASE(:,LRATIO)=RNCOLUMNA(:,1);
OC(1, LRATIO)=COSTOP;
XoS=size(Xo);
for C=1:XoS(1,1)
    IL(C,LRATIO)=Xo(C,1);
    JL(C,LRATIO)=Xo(C,2);
    XL(C,LRATIO)=Xo(C,3);
end
ILS=size(IL);
if(XoS(1,1)<ILS(1,1))
for C=XoS(1,1)+1:ILS(1,1)
    IL(C,LRATIO)=0;
    JL(C,LRATIO)=0;
    XL(C,LRATIO)=0;
end
end

%MODIFICAR EL MÁSTER
%ADECUAR PARA FORMATO GDX
for C=1:NB
    NC(C,1)=C;
    NC(C,2)=OC(C);

end

```

```

A=0;
for C=1:NB
for D=1:NB
A=A+1;
NBA(A,1)=C;
NBA(A,2)=D;
NBA(A,3)=BASE(C,D);
end
end

```

```

%INCLUIR EN GDX

```

```

writegdx('MásterB.gdx','parameter','base',NBA,'parameter','c',NC);

```

```

%EJECUTAR MÁSTER

```

```

GAMS('Máster');

```

```

%EXTRAER DUALES PROPORCIONADAS DEL MÁSTER

```

```

DUALES=readgdx('CVRP_MÁSTER.gdx','dua');

```

```

DUAL1=0;

```

```

DUAL2=0;

```

```

DUALEST=size(DUALES);

```

```

DUAL0=0;

```

```

for C=1:DUALEST(1,1)

```

```

NARUT=DUALES(C,1);

```

```

if(NARUT<2)

```

```

DUAL1=DUALES(C,2);

```

```

else

```

```

if(NARUT<3)

```

```

DUAL2=DUALES(C,2);

```

```

else

```

```

DUAL0=DUALES(C,2);

```

```

end
end

end
% INCLUIR DUALES A GDY

writegdx('PricingB.gdx','parameter','dual1',DUAL1,'parameter','dual2',DUAL2,'parameter','dual0',DUAL0);
GAMS('Pricing.gms');
iter=iter+1;
FULL=readgdx('CVRP-PRICING','full');
MET= readgdx('CVRP-PRICING','met');
COSTOP= readgdx('CVRP-PRICING','costop');
CRO=readgdx('CVRP-PRICING','cro');

Xo=readgdx('CVRP-PRICING.gdx','xo');
Zo=readgdx('CVRP_MÁSTER.gdx','zo');
LO=readgdx('CVRP_MÁSTER','lo');
LL=zeros(3,1);
LOS=size(LO);

for C=1:LOS(1,1)
LL(LO(C,1),1)=LO(C,2);
end
end
Xo=[IL*LL JL*LL XL*LL];
toc;

```

## Anexo F. GMS del problema Maestro de la descomposición D-W

\$ontext

Sub-rutina correspondiente a la solución de un problema de ruteo de vehículos con restricciones de capacidad dentro de un algoritmo de descomposición de Dantzig-Wolfe donde se presenta el máster solo incluyendo en este las restricciones

de indegree y outdegree de flujo y las restricciones de ingreso y salida de camiones del depot.

Autor: DANIEL ORLANDO MARTÍNEZ QUEZADA

\$offtext

\$set matout "'CVRP\_MÁSTER.gdx', dua, lo, zo";

set i restricciones;

parameter ladod(i);

\*Carga de GDX-A

\$gdxin MásterA.gdx

\$load i=i

\$load ladod=ladod

\*Fin carga de GDX-A

alias (i,j);

parameter base (i,j);

parameter costo(j);

\*Carga de GDX-B

\$gdxin MásterB.gdx

\$load base=base

\$load costo=c

\*Fin carga de GDX-B

variable Z distancia total requerida;

variable l(j) eleccion del conjunto de rutas solucion;

```
equations
fun funcion objetivo
res(i) restricciones del máster;
fun.. Z=e= sum(j,l(j)*costo(j));
res(i).. sum(j, base(i,j)*l(j))=e=ladod(i);
model máster /all/;
solve máster minimizing Z using LP;
parameter dua(i);
parameter lo(j);
parameter zo;
lo(j)=l.l(j);
zo=z.l;
dua(i)=res.m(i);
execute_unload %matout%;
```

## Anexo G. GMS del sub-problema de la descomposición D-W

\$ontext

Sub-rutina correspondiente a la solución de un problema de ruteo de vehículos con restricciones de capacidad dentro de un algoritmo de descomposición de Dantzig-Wolfe donde se presenta el pricing solo considerando la restricción de eliminación de sub-tours y de flujo.

Autor: DANIEL ORLANDO MARTÍNEZ QUEZADA

\$offtext

```
set i  nodos;
set v(i)  clientes;
set di  dimensiones /1*2/;
set n  sub-tours considerados;
parameter c  capacidad;
parameter d(i)  demanda;
parameter p(i,di)  posicion;
parameter dual1  dual1;
parameter dual2  dual2;
parameter dual0  dual0;
parameter car  limite de cardinalidad;
```

\*Carga de GDX-A

```
$gdxin PricingA.gdx
```

```
$load i=i
```

```
$load v=v
```

```
$load n=n
```

```
$load car=car
```

```
$load c=c
```

```
$load d=d
```

\$load p=p

\*Fin carga de GDX-A

\*Carga de GDX-B

\$gdxin PricingB.gdx

\$load dual1=dual1

\$load dual2=dual2

\$load dual0=dual0

\*Fin carga de GDX-B

set e(i,i);

alias(i,j);

e(i,j)\$ (not sameas(i,j))=yes;

set S(n,v) conjunto de subtours;

set vi(v) cliente inicial;

vi(v)\$ (ord(v)=1)=yes;

set w(v) clientes sin el inicial;

w(v)= not vi(v);

S("1",vi)=no;

S("2",vi)=yes;

set actual(n) sub-conjunto actual de S;

actual("2")=yes;

set hecho(n) todo n-esimo S recorrido;

hecho("1")=yes;

hecho("2")=yes;

set lo(n) "control de ciclo'hecho";

parameter cor;

loop(w,

    lo(n)=hecho(n);

```

loop(lo,
  cor=sum(S(lo,v),1);
if((cor<car),
  actual(n)= actual(n-1);
  S(actual,v)=S(lo,v);
  S(actual,w)=yes;
  hecho(actual)=yes;
else
  hecho(actual)=no;
);
);
);

```

```

parameter Sciclo(n);
  Sciclo(n)=sum(S(n,v),1);
set sub(n);
  sub(n)$(Sciclo(n)>=2)=yes;
alias(v,nv);
set nS(n,i,j);
nS(sub(n),e(v,nv))$(s(n,v) and s(n,nv))=yes;

```

\*fin de conjuntos auxiliares

```

parameter l(i,j);
  l(e(i,j))=(sqrt(power((p(i,"1")-p(j,"1")),2)+power((p(i,"2")-p(j,"2")),2)));
binary variable x(i,j);
parameter r(n);
  r(n)=((sum((v)$(S(n,v)),d(v)))/c);
variable CR;
equations
costo costo reducido asociado

```

flujoi(i) restriccion de unica entrada y unica salida  
 flujoo(i) restriccion de unica entrada y unica salida  
 subtour(n) restricciones de sub-tours considerados;

```
costo..      CR=e=      (sum(e(i,j),l(i,j)*x(i,j)))-      dual1*(sum(nv,      x(nv,'1')))-
dual2*(sum(nv,x('1',nv)))-(dual0);
flujoi(i)$v(i).. sum{j$(ord(i) ne ord(j)),x[i,j]}=e= 1;
flujoo(i)$v(i).. sum{j$(ord(i) ne ord(j)),x[j,i]}=e= 1;
subtour(n).. sum(nS(n,e(i,j)),x(i,j))=l=Sciclo(n)-r(n) ;
```

```
model pricing /all/;
solve pricing minimizing CR using MIP;
$set matout "CVRP-PRICING.gdx', full, met,costop, cro, xo";
parameter full;
parameter met;
parameter costop;
parameter cro;
parameter xo(i,j);
  full=sum(nv, x.l(nv,'1'));
  met =sum(nv, x.l('1',nv));
  costop =sum(e(i,j), l(i,j)*x.l(i,j));
  cro =CR.l;
  xo(i,j)=x.l(i,j);
execute_unload %matout%;
```

## Anexo H. Script para generar gráficas de las soluciones

```
%Grafica las soluciones de instancias del CVRP
%Autor: DANIEL ORLANDO MARTÍNEZ QUEZADA
X=POSICION(:,1);
Y=POSICION(:,2);
[a b]=size(Xo);
plot(X,Y,'LineStyle','none','marker','.')
hold on
for j=1:NODOS
text(X(j),Y(j),int2str(j));
end
plot(X(1),Y(1),'*r')
for i=1:a
X1=[X(Xo(i,1)) X(Xo(i,2))];
Y1=[Y(Xo(i,1)) Y(Xo(i,2))];
plot(X1,Y1,'-.k');
end
title(int2str(Zo));
hold off
```

Anexo I. Instancia P\_n16\_k8

CAMIONES=8;

NODOS=16;

POSICION=[ 30 40; 37 52; 49 49; 52 64; 31 62; 52 33; 42 41; 52 41; 57 58; 62 42;  
42 57;27 68; 43 67; 58 48; 58 27;37 69];

DEMANDAS=[1 0;2 19;3 30;4 16;5 23;6 11;7 31;8 15;9 28;10 8;11 8;12 7;13 14;14  
6;15 19;16 11];

CAPACIDAD=35;

Anexo J. Instancia P\_n19\_k2

CAMIONES=2;

NODOS=19;

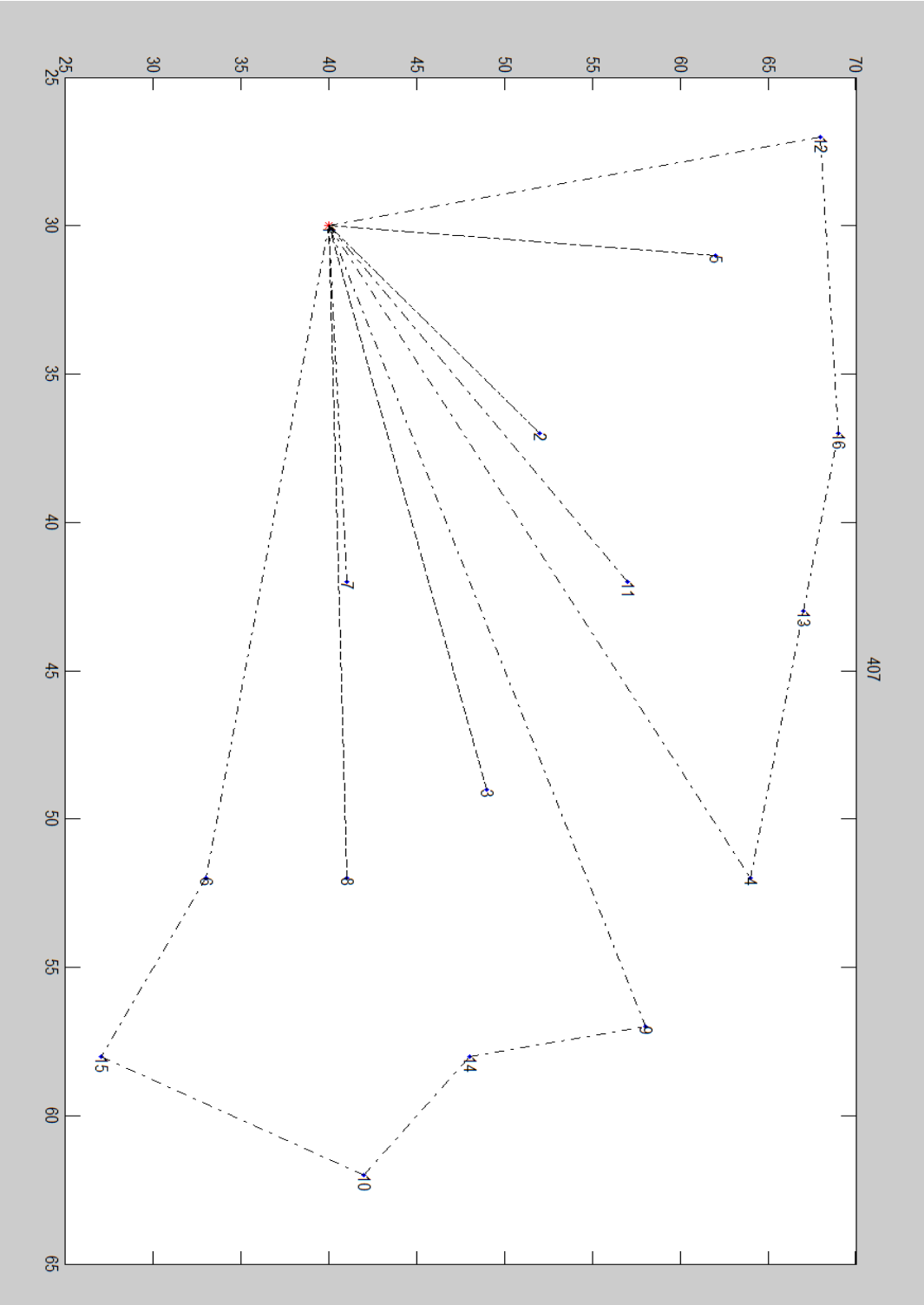
POSICION=[ 30 40; 37 52; 49 43; 52 64; 31 62; 52 33; 42 41; 52 41; 57 58; 62 42;  
42 57; 27 68; 43 67; 58 27; 37 69; 61 33; 62 63; 63 69; 45 35 ];

DEMANDAS=[1 0;2 19;3 30;4 16;5 23;6 11;7 31;8 15;9 28;10 14;11 8;12 7;13  
14;14 19;15 11;16 26;17 17;18 6;19 15];

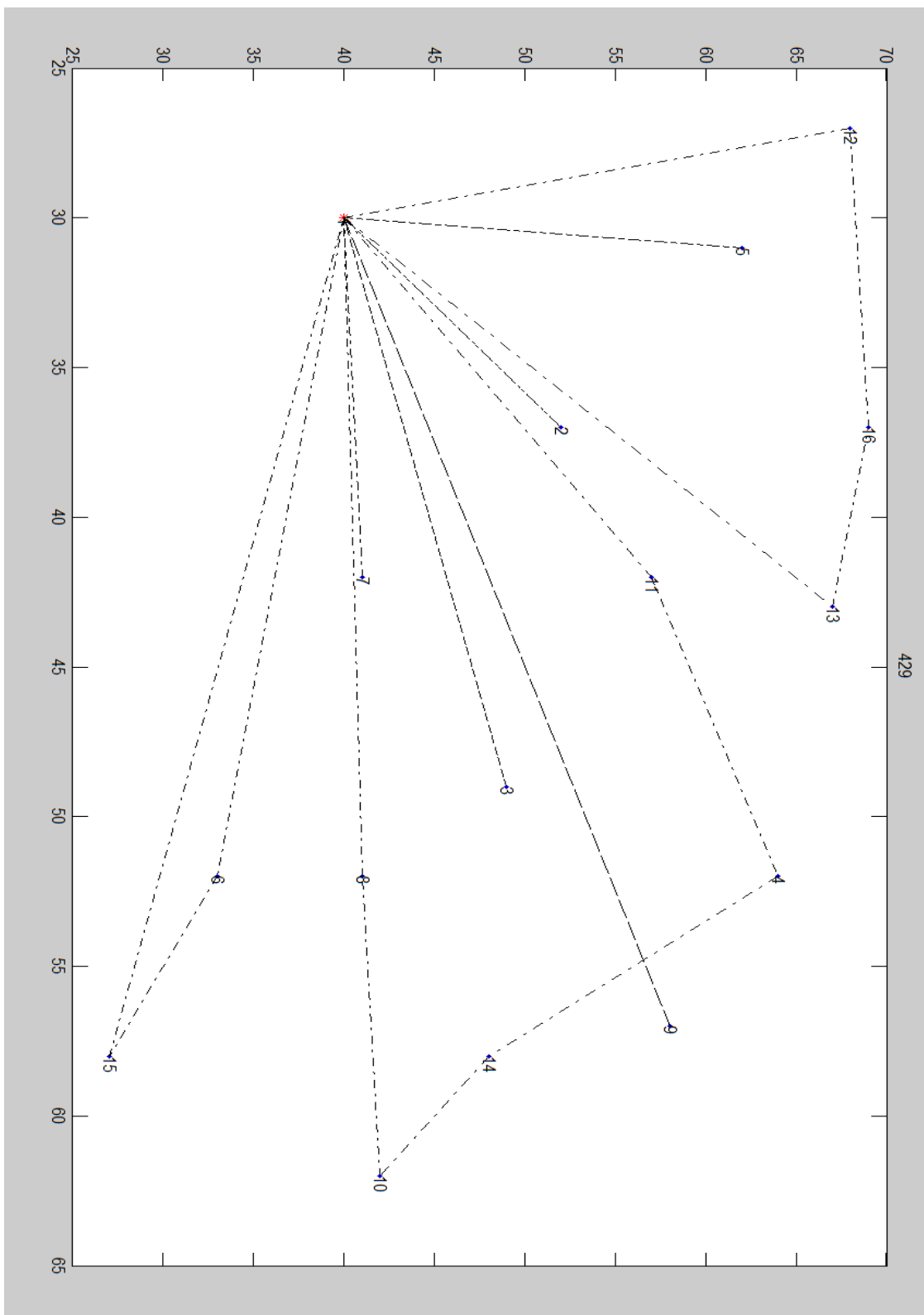
CAPACIDAD=160;

Anexo K. Resultados P\_n16\_k8 sin descomponer

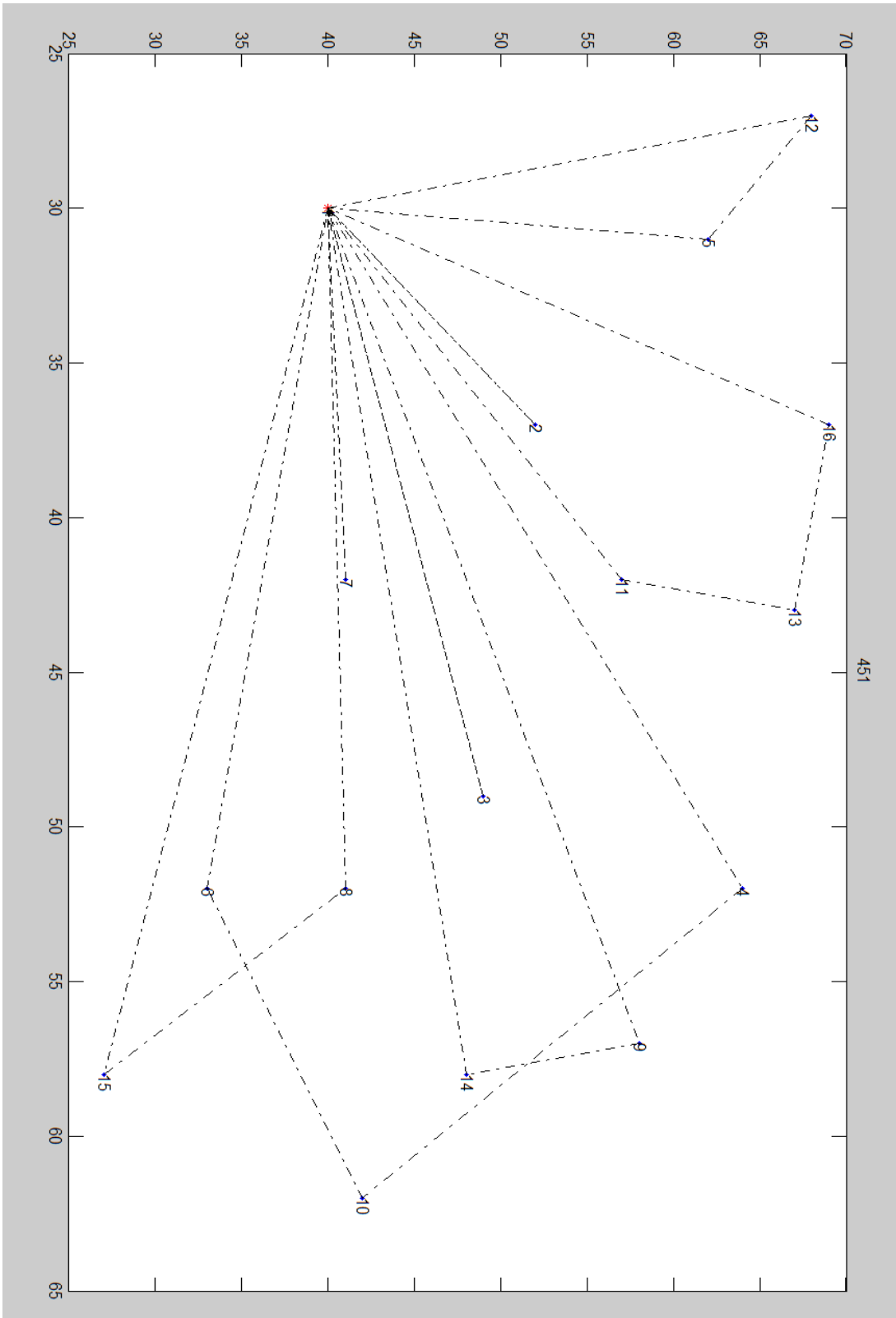
Elapsed time is 0.807612 seconds, Infactibel S=2



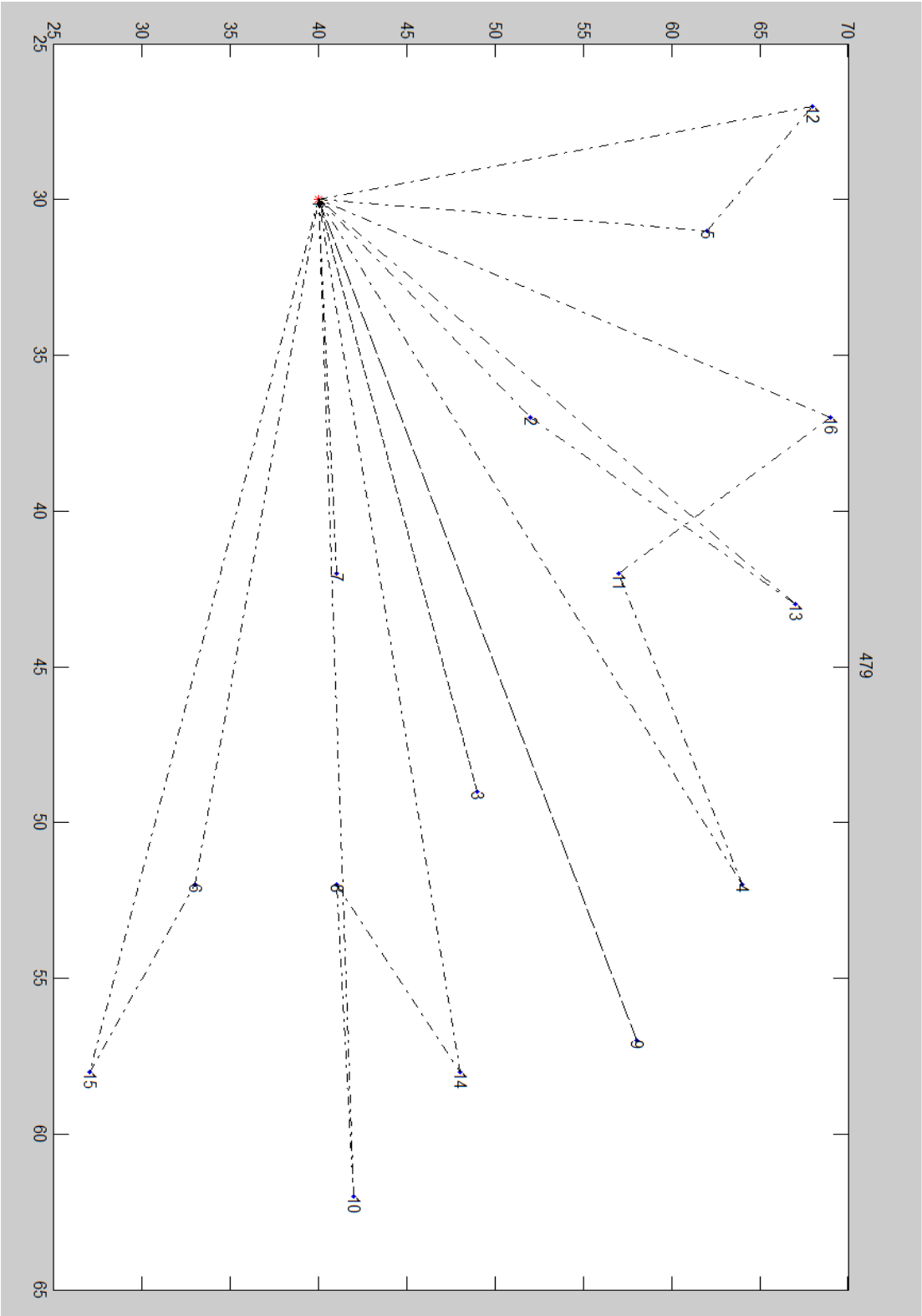
Elapsed time is 0.243615 seconds. Infeasible S=3



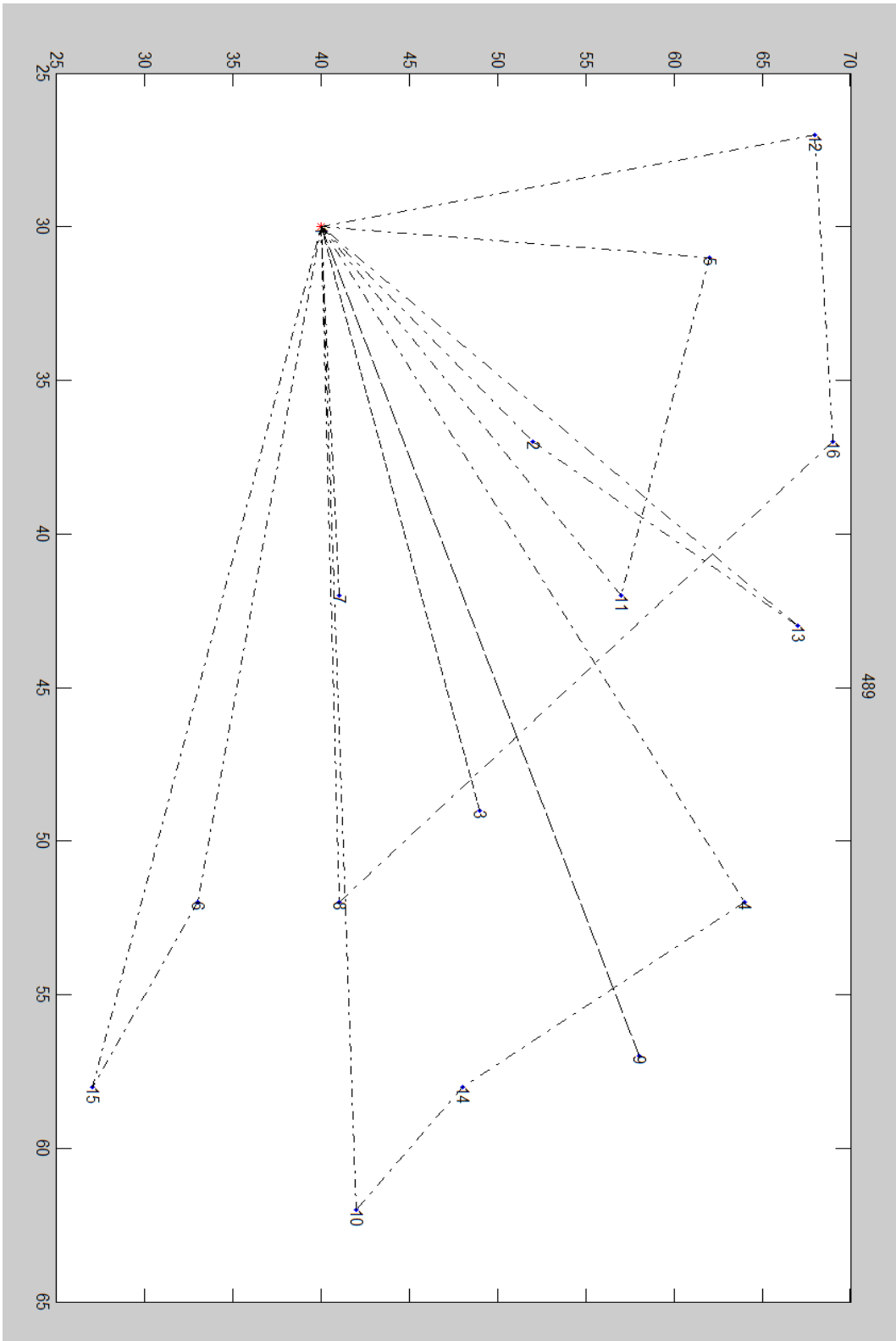
Elapsed time is 0.443328 seconds. Infeasible S=4



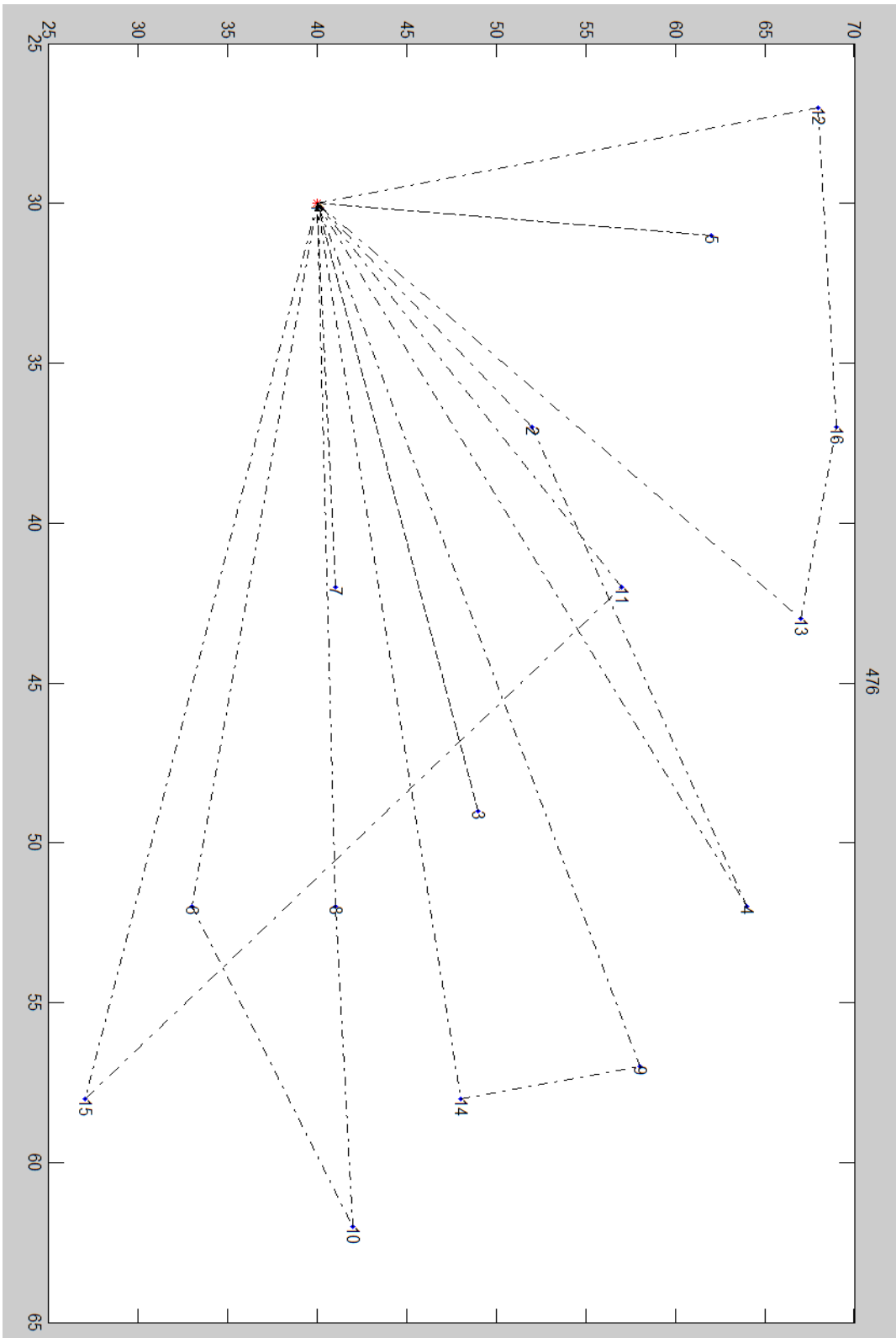
Elapsed time is 1.947178 seconds, factible no optima S=5.



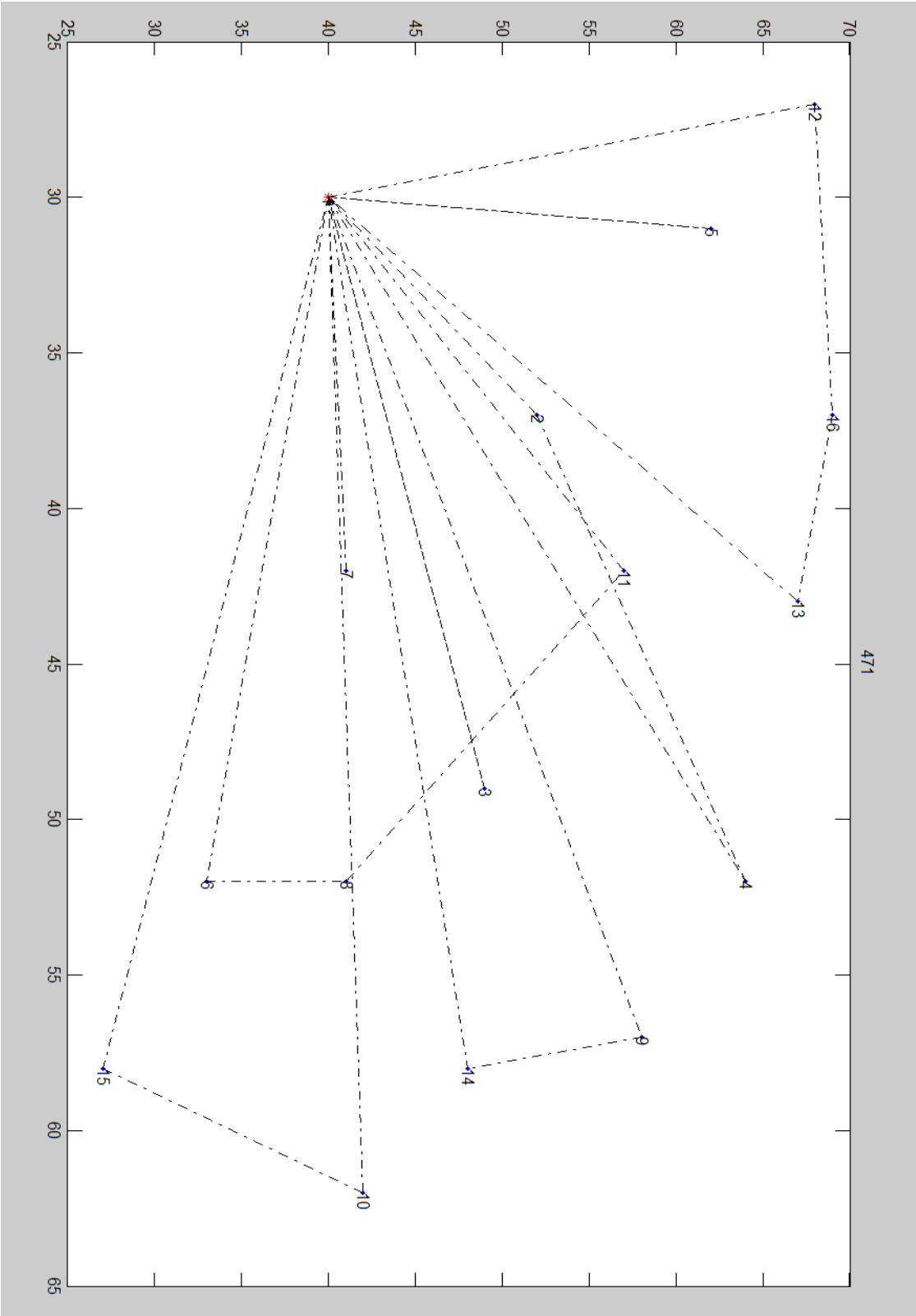
Elapsed time is 7.518216 seconds. Factible no optima S=6.



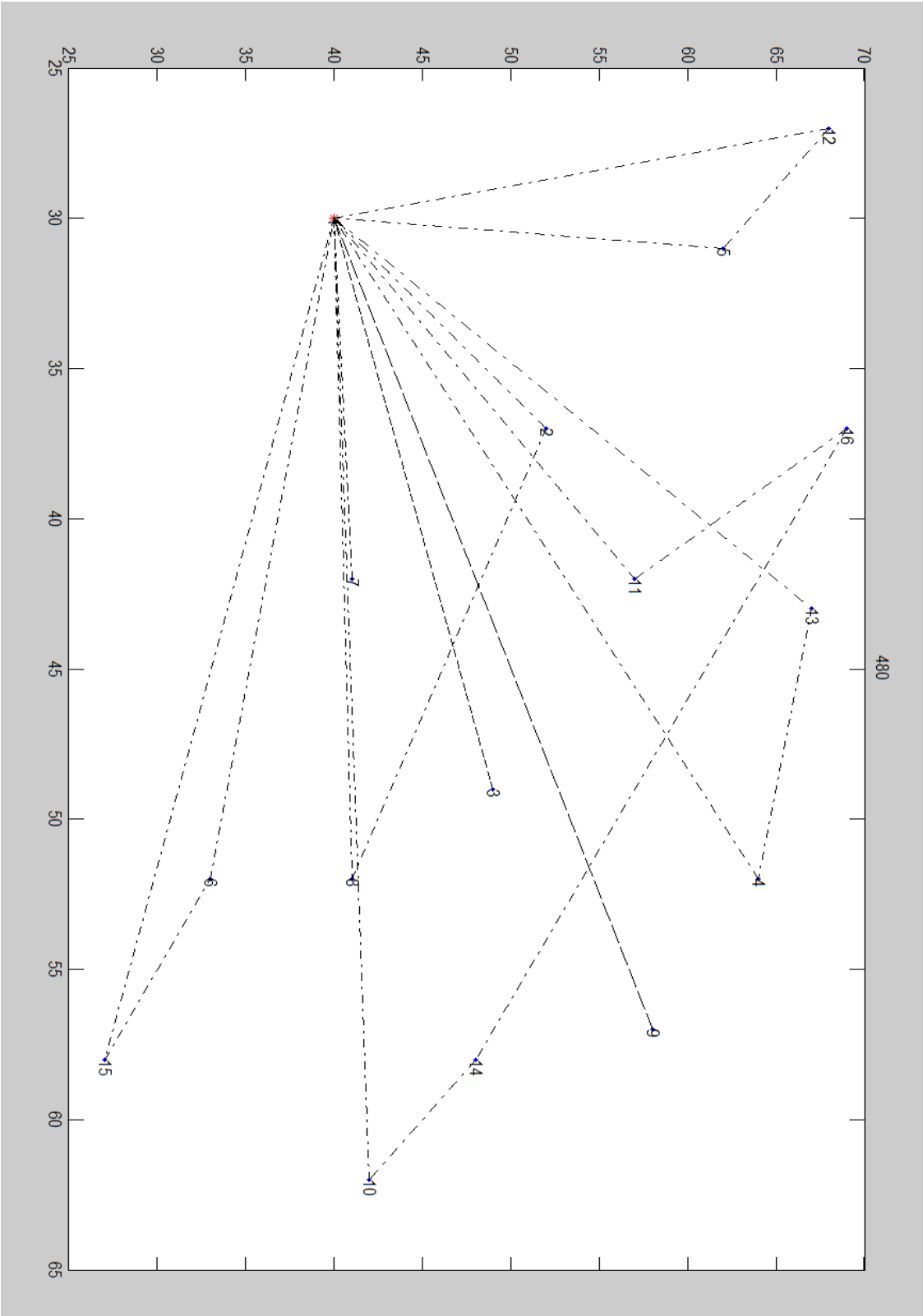
Elapsed time is 21.082206 seconds. Feasible no optima S=7.



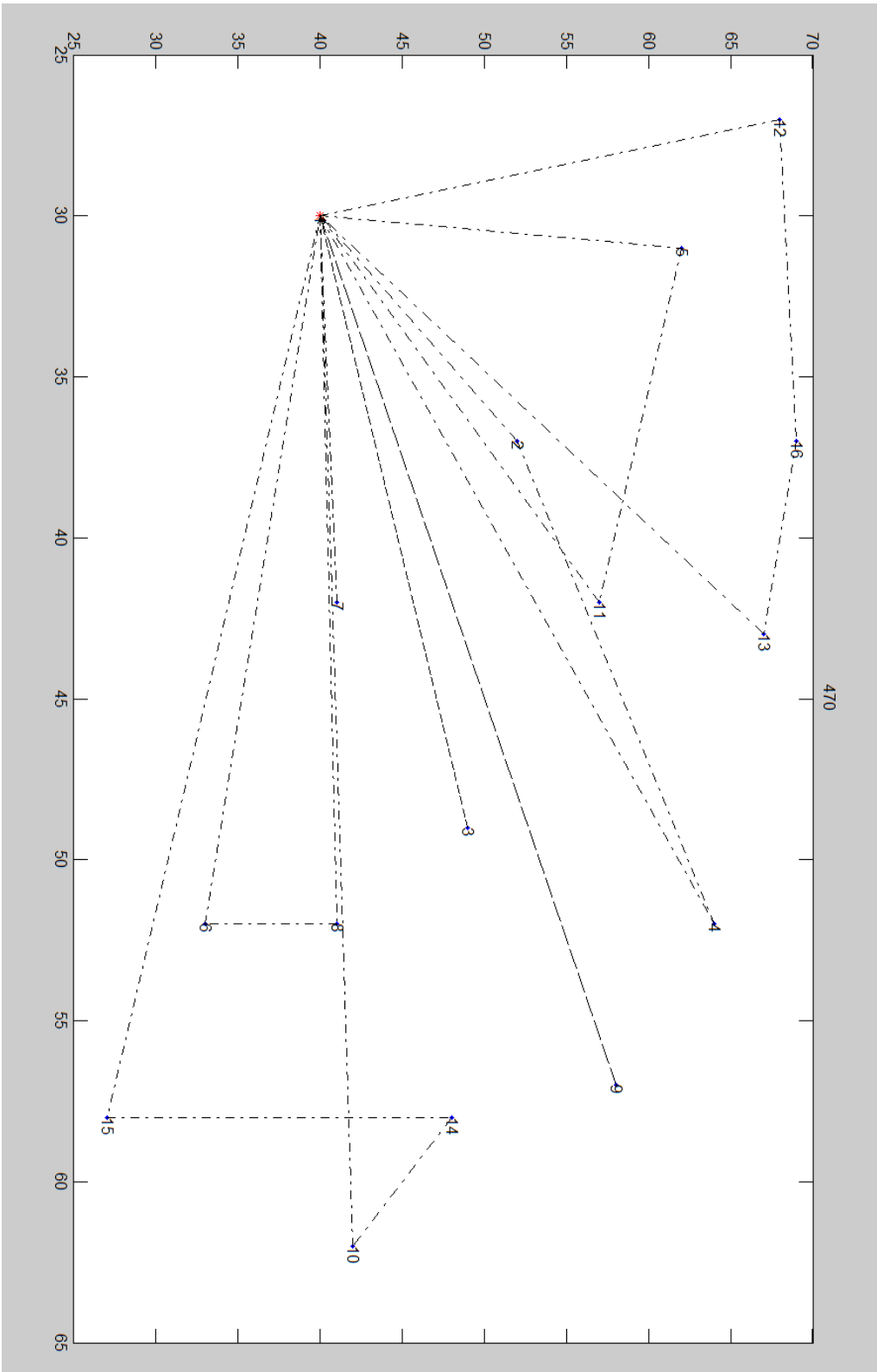
Elapsed time is 42.849806 seconds. Feasible no optima S=8.



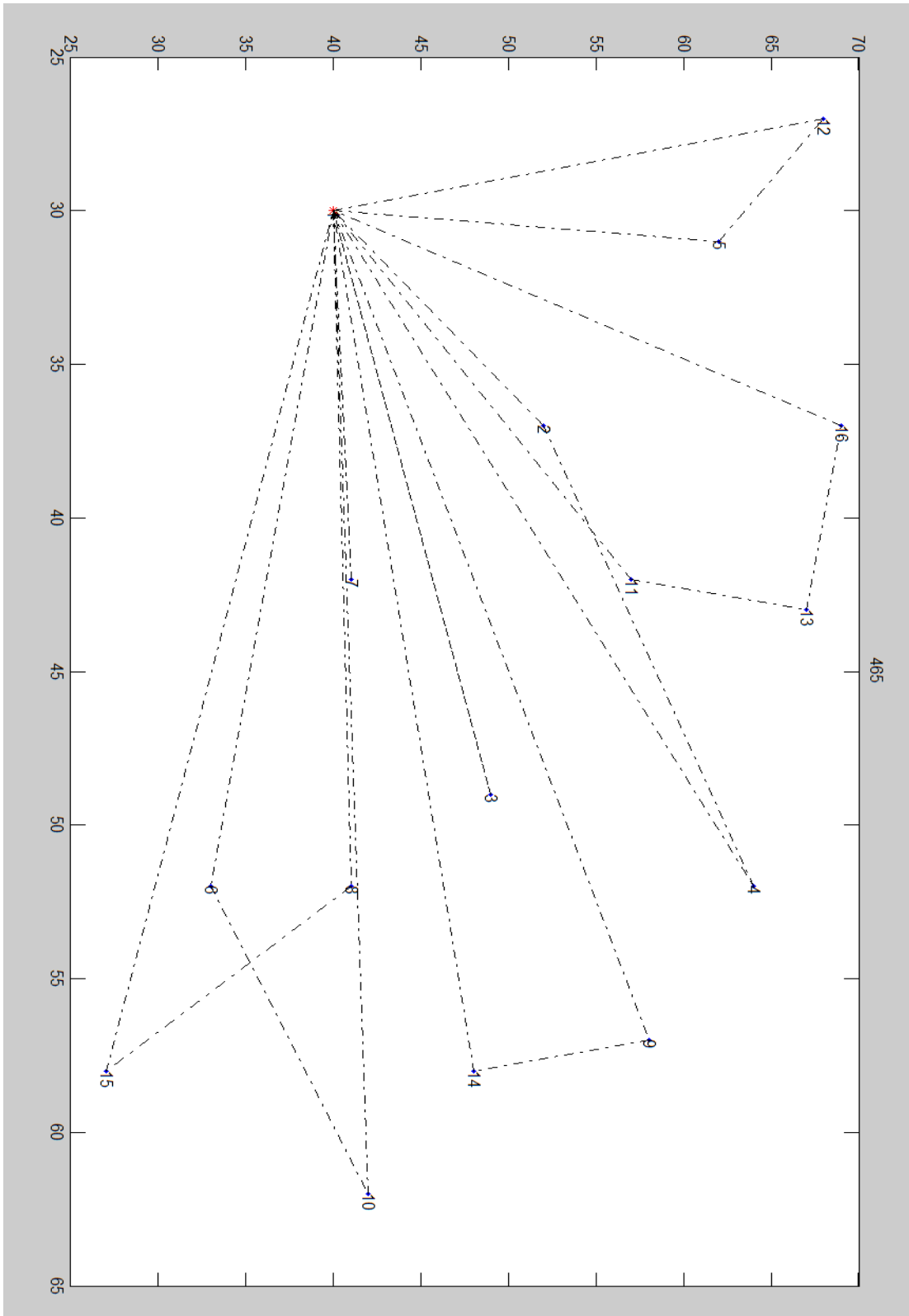
Elapsed time is 71.430450 seconds. Factible no optima S=9



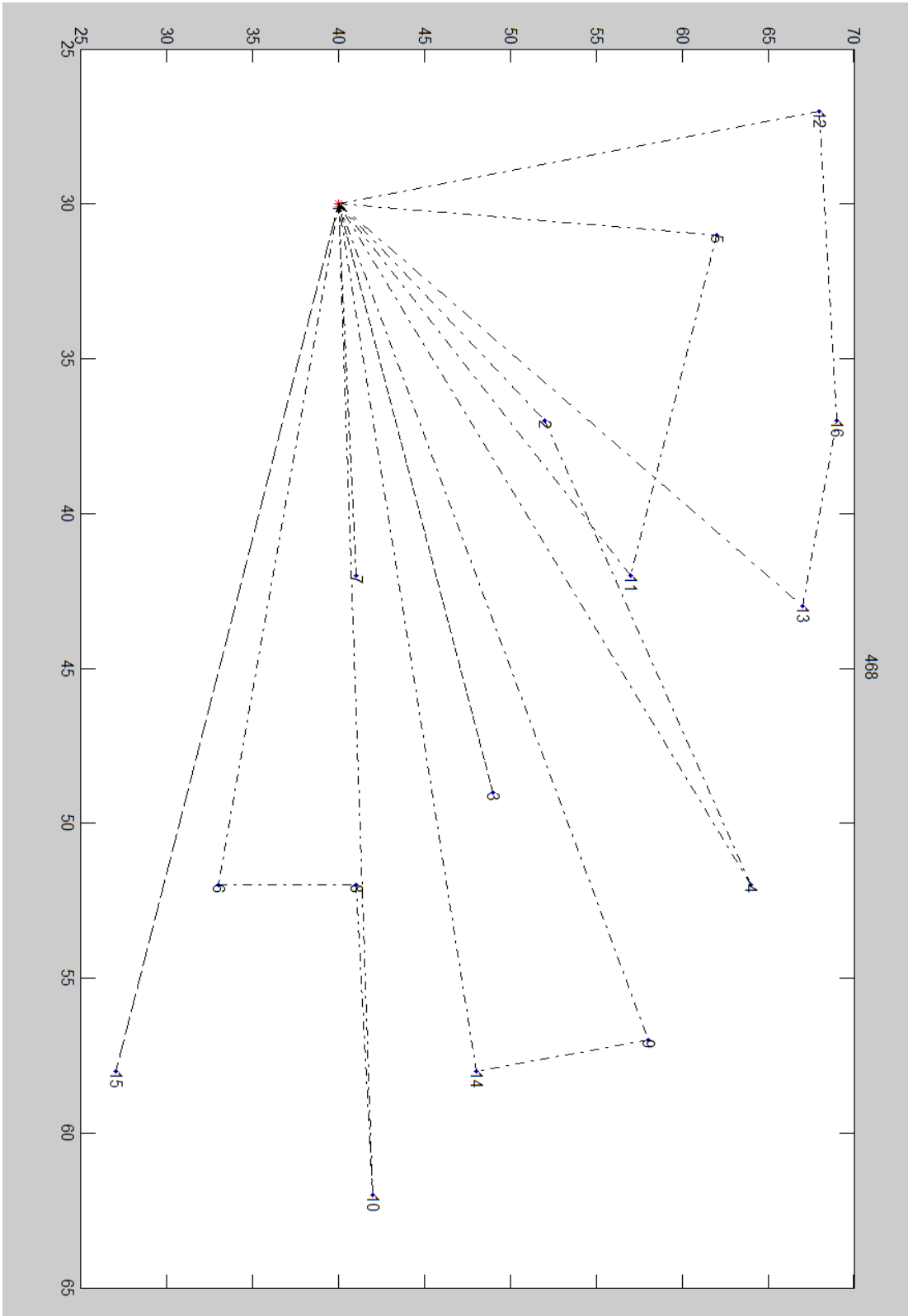
Elapsed time is 93.435029 seconds. Factible no optima S=10



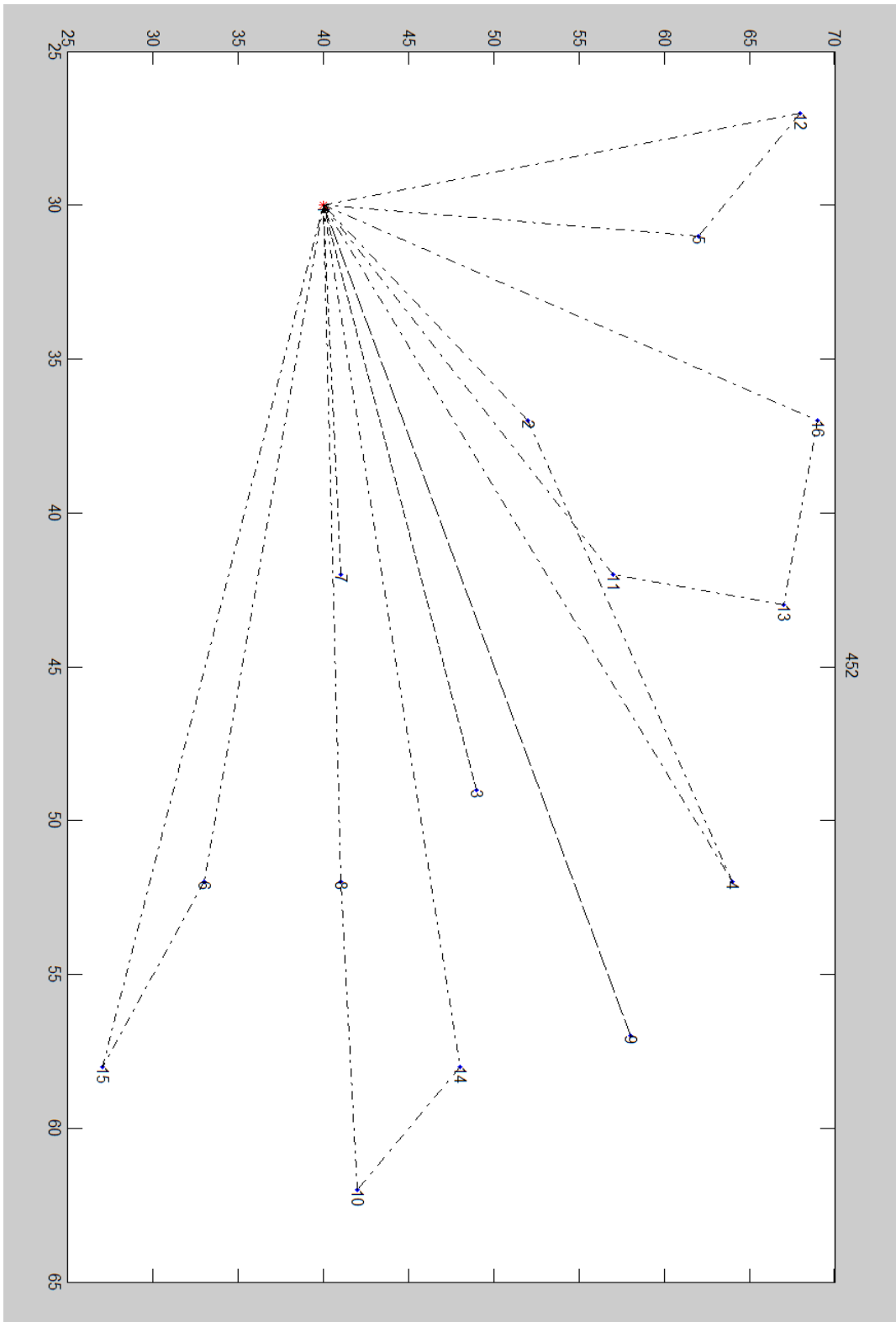
Elapsed time is 104.254440 seconds. Factible no optima S=11.



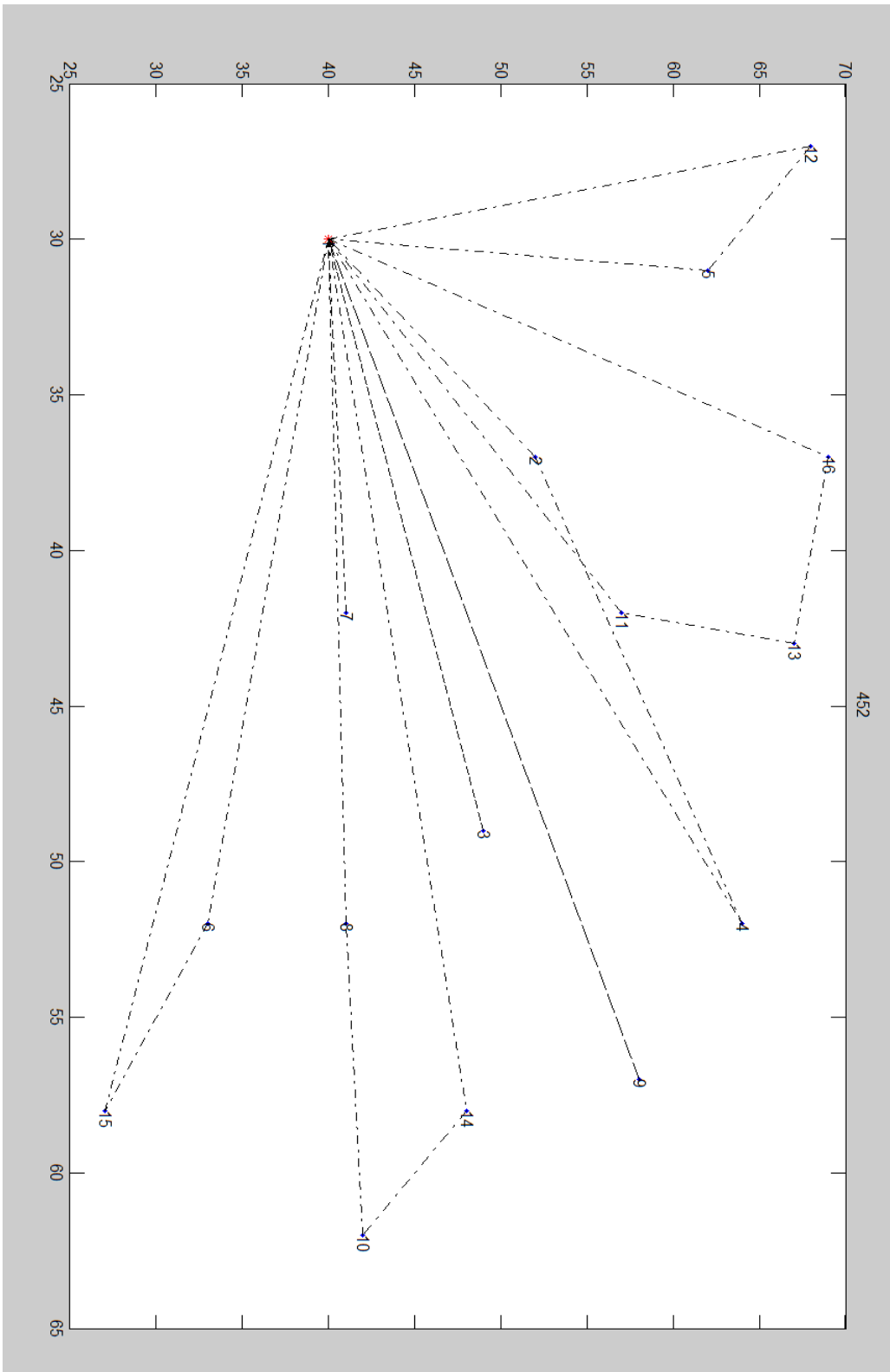
Elapsed time is 109.464665 seconds. Factible no optima S=12.



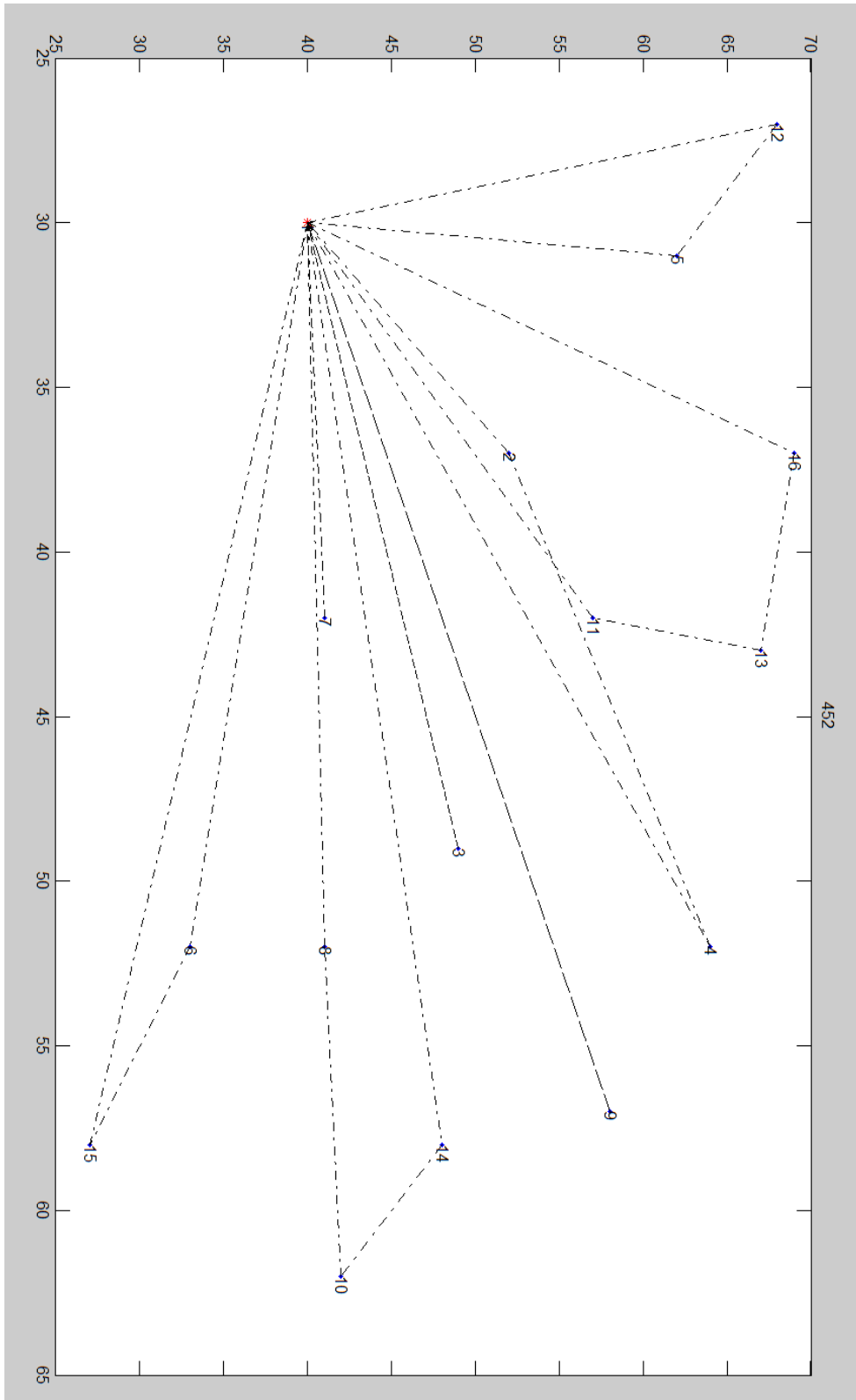
Elapsed time is 112.199789 seconds. Solución optima S=13.



Elapsed time is 113.280404 seconds. Optima S=14

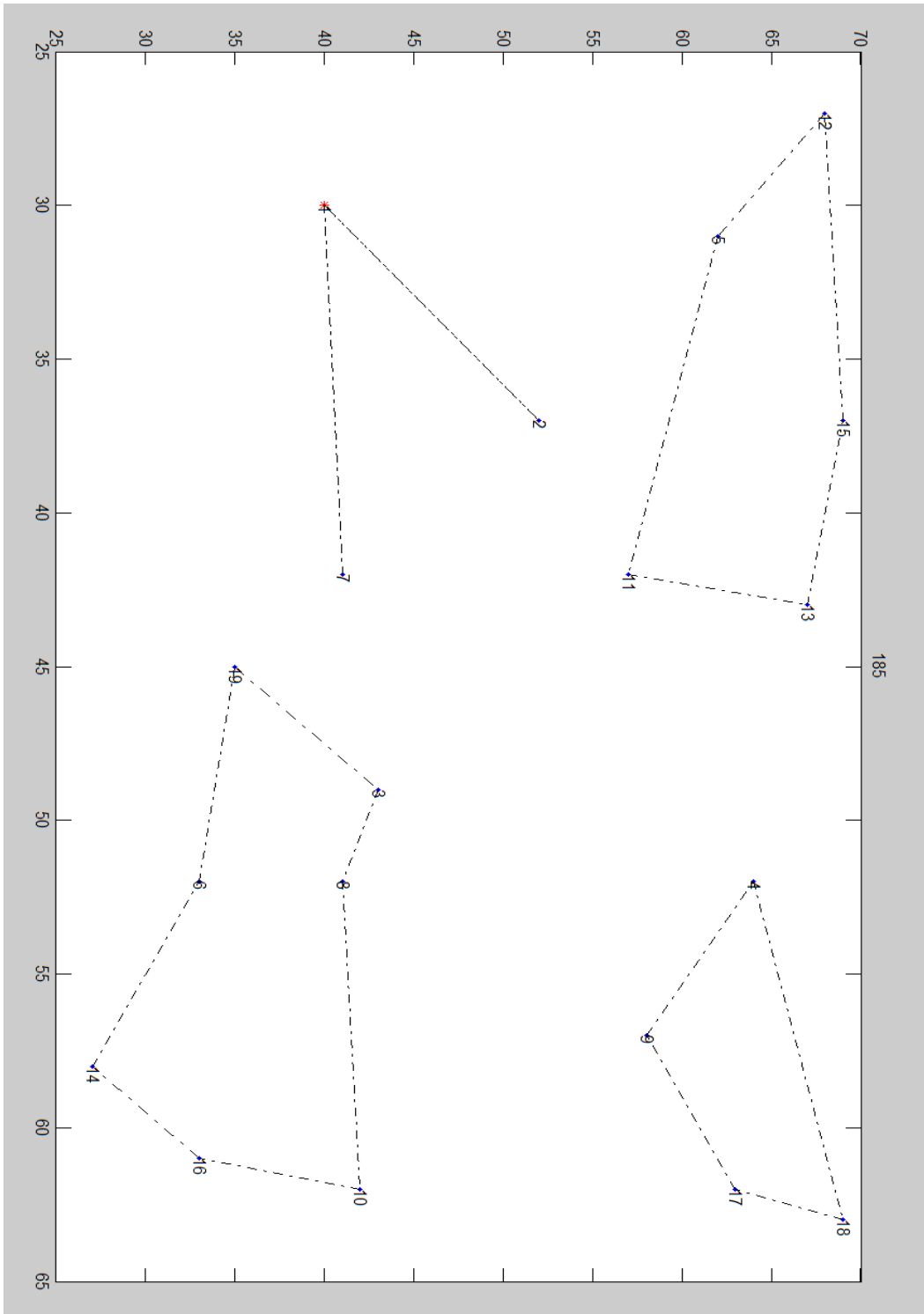


Elapsed time is 111.947630 seconds. Optima S=15.

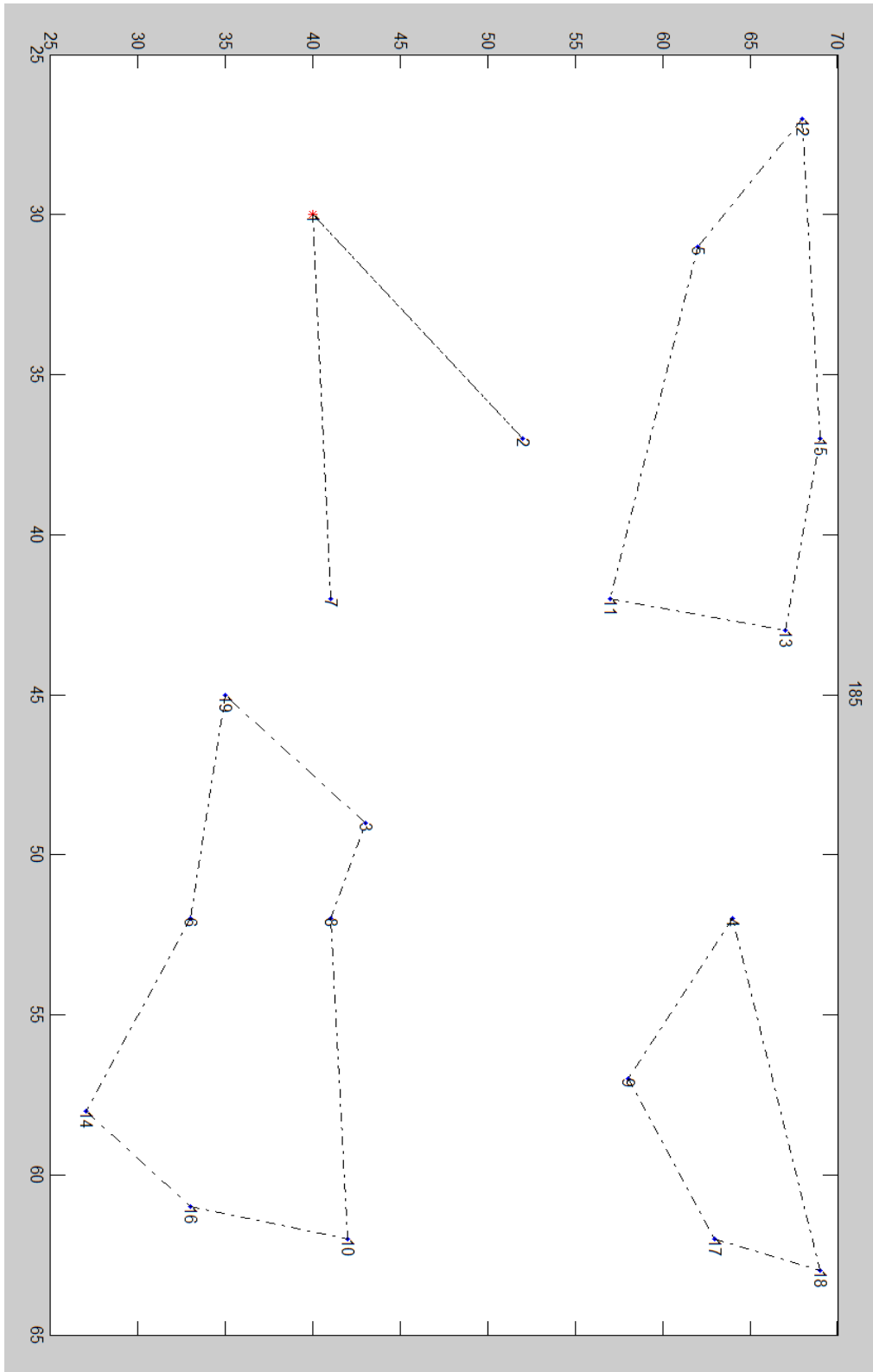


Anexo L. Resultados P\_n19\_k2 sin descomponer

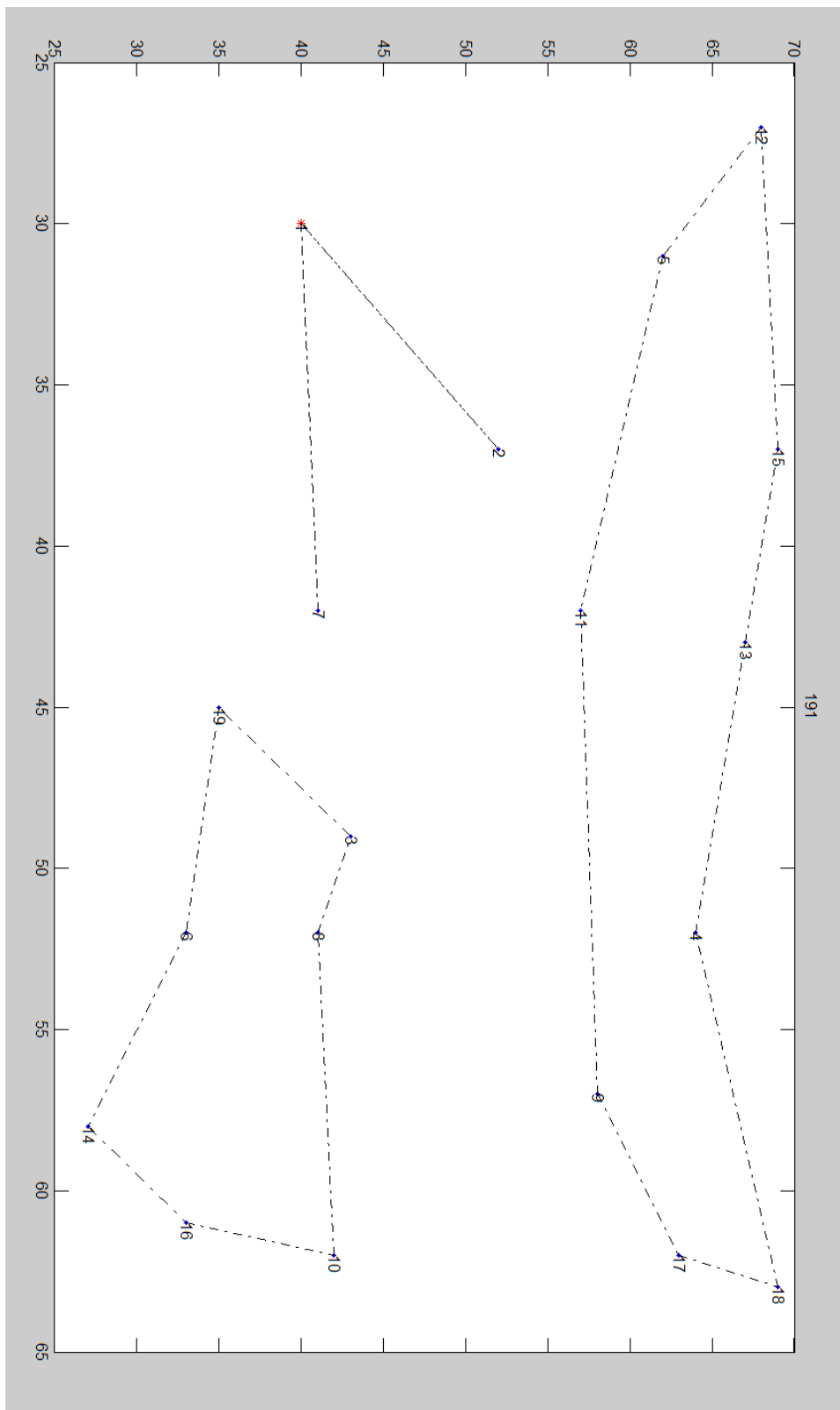
Elapsed time is 0.178770 seconds. Infeasible S=2.



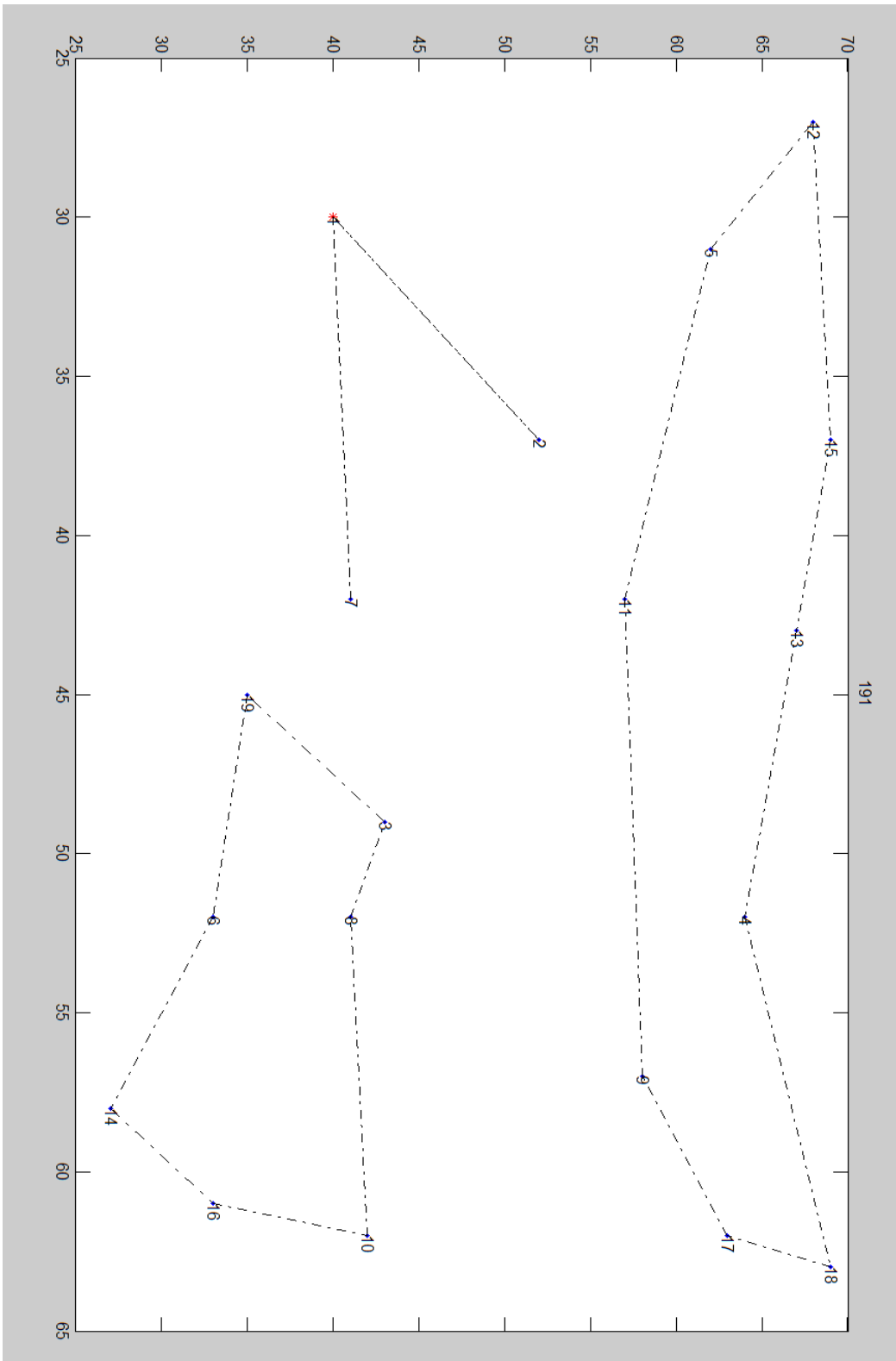
Elapsed time is 0.248763 seconds. Infeasible S=3.



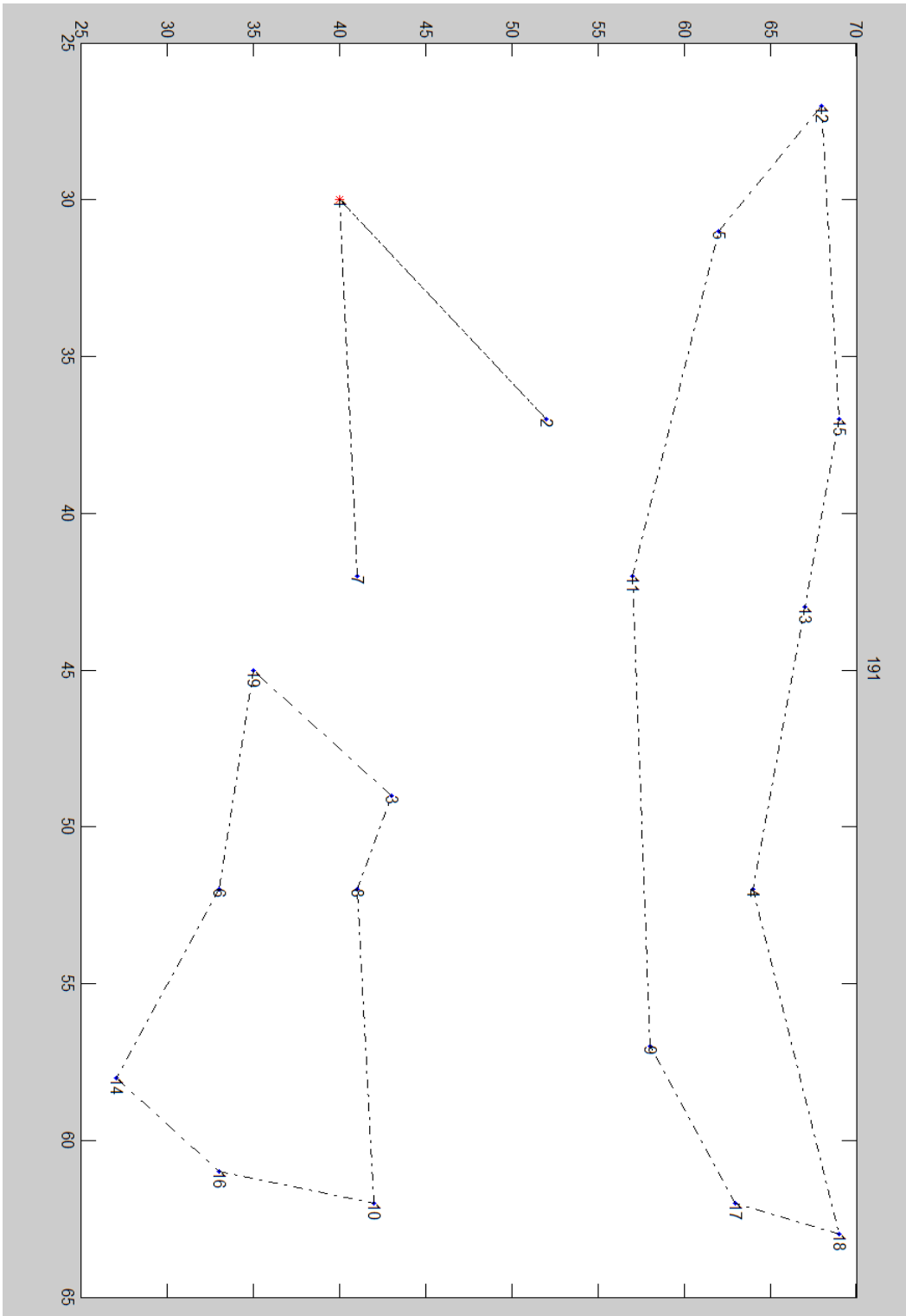
Elapsed time is 1.271368 seconds. Infeasible S=4



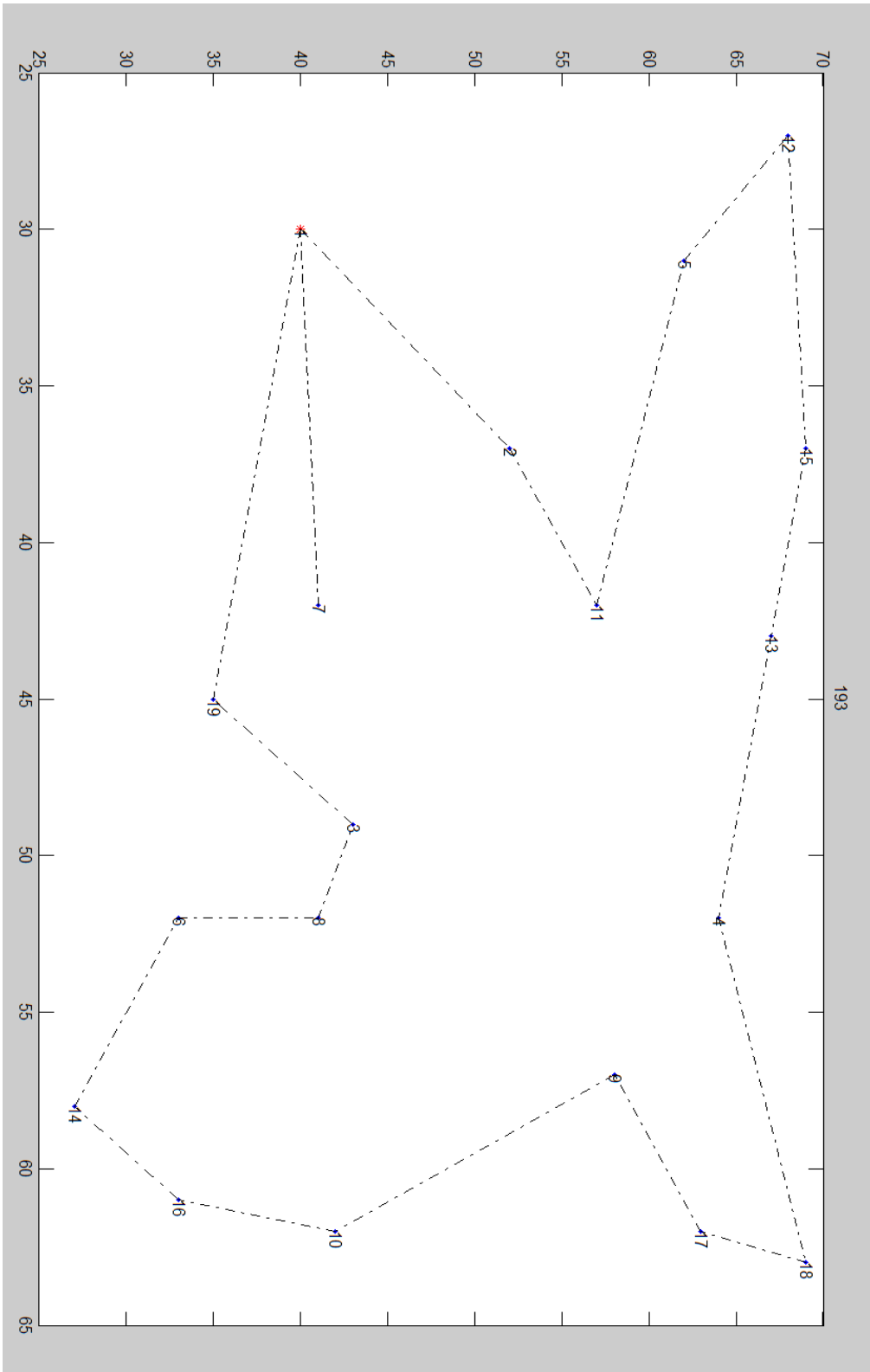
Elapsed time is 10.840778 seconds. Infeasible S=5.



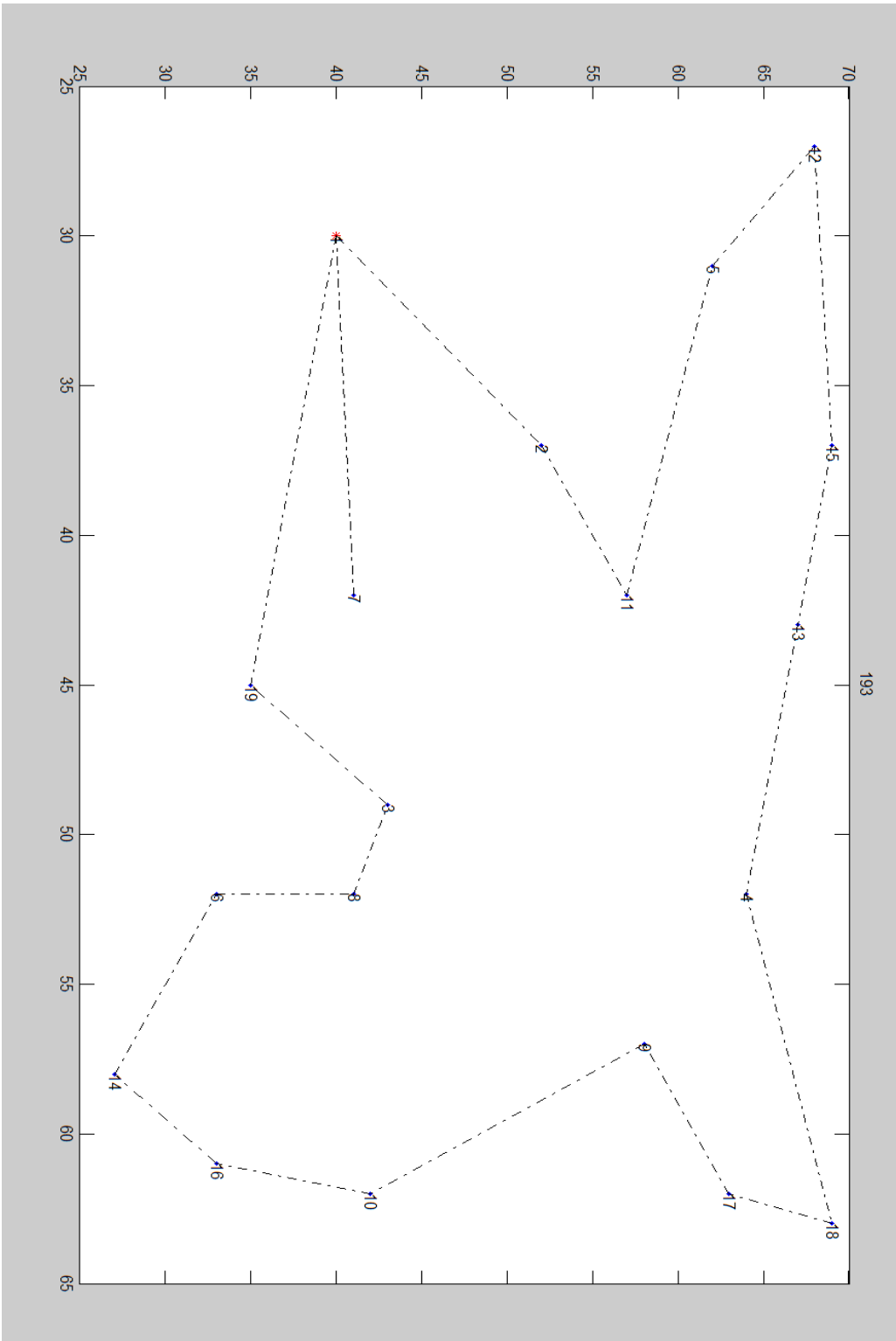
Elapsed time is 72.561255 seconds. Infeasible S=6.



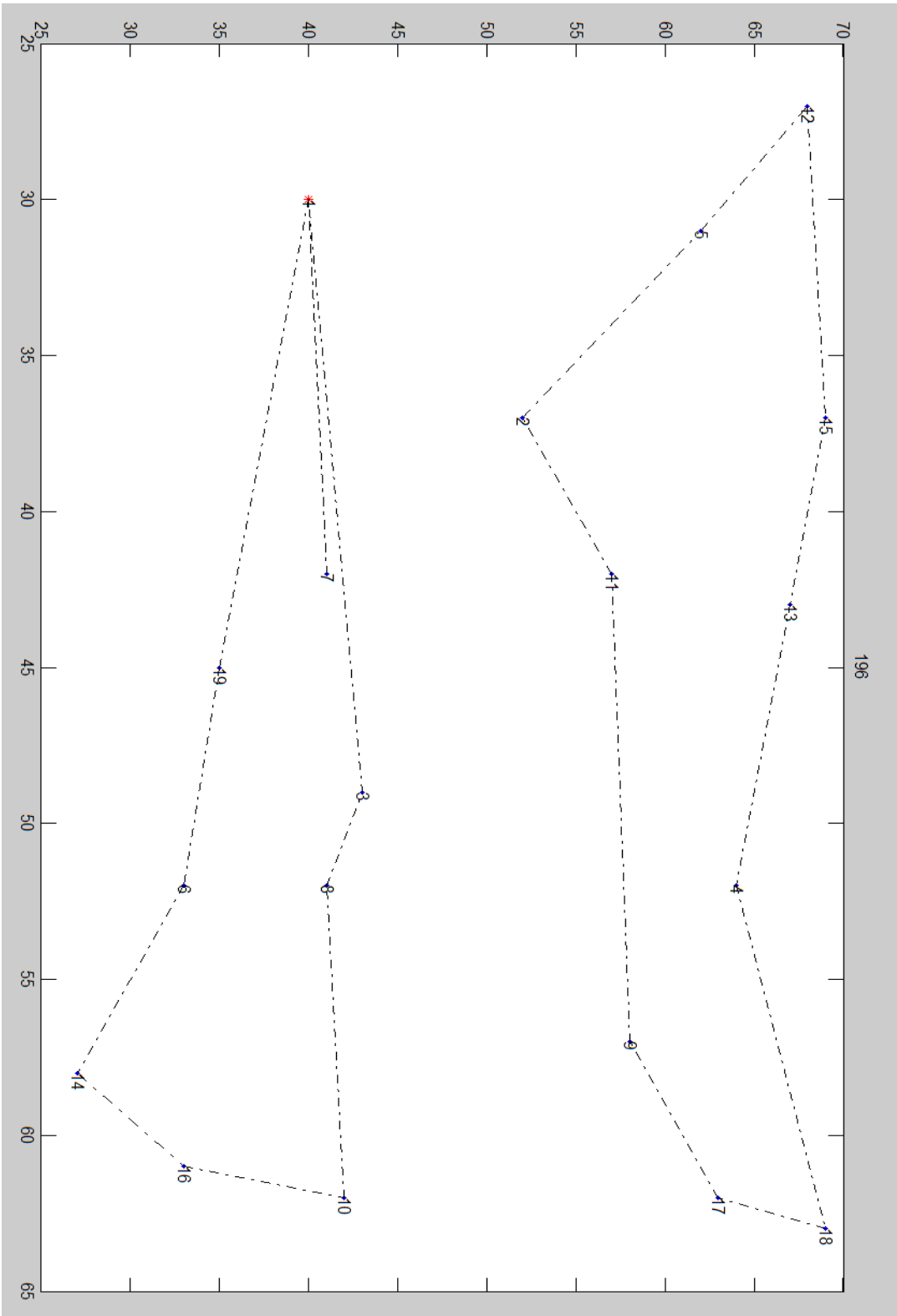
Elapsed time is 591.183741 seconds. Infeasible S=7.



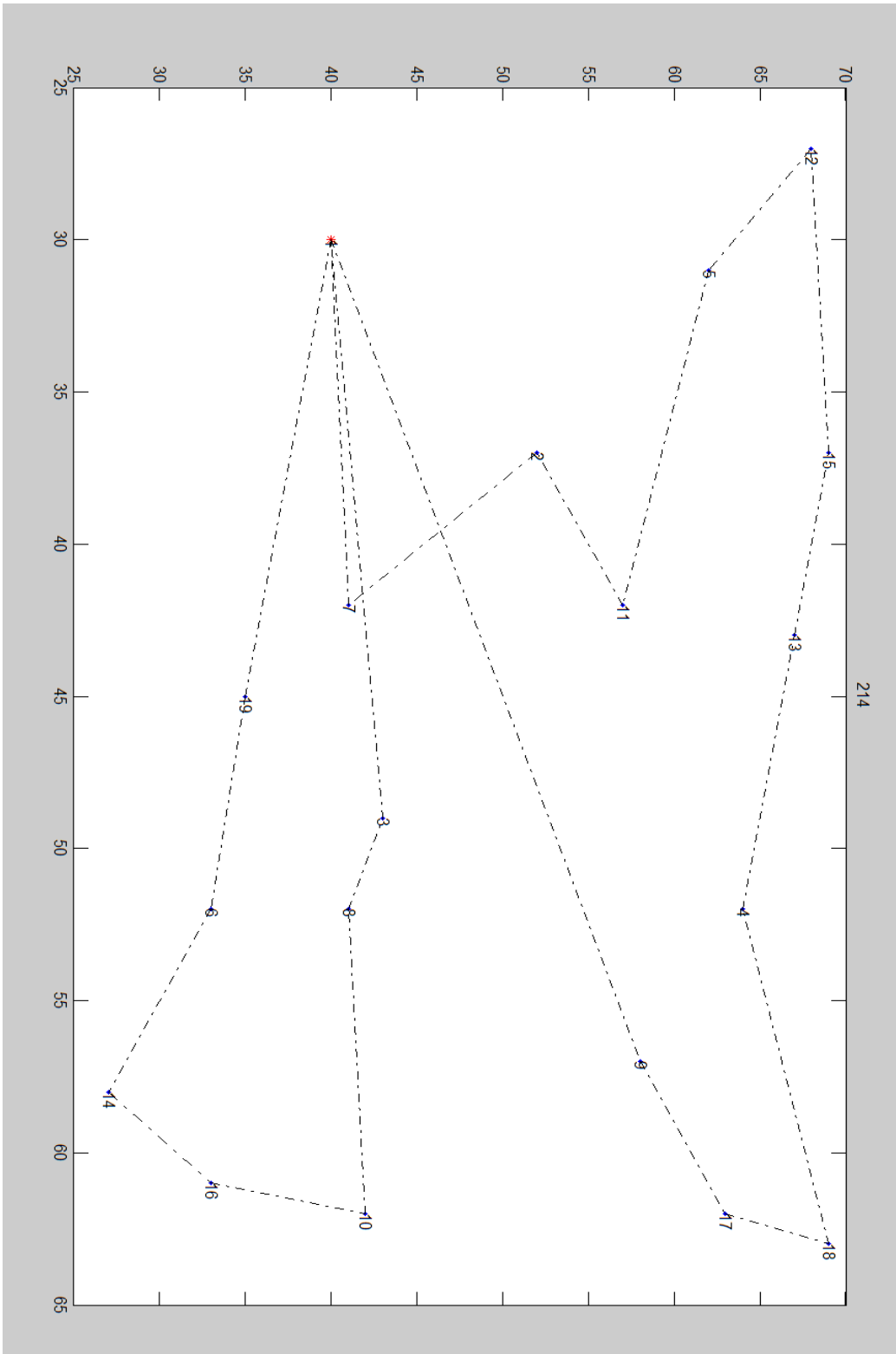
Elapsed time is 2385.724682 seconds Infactible S=8.



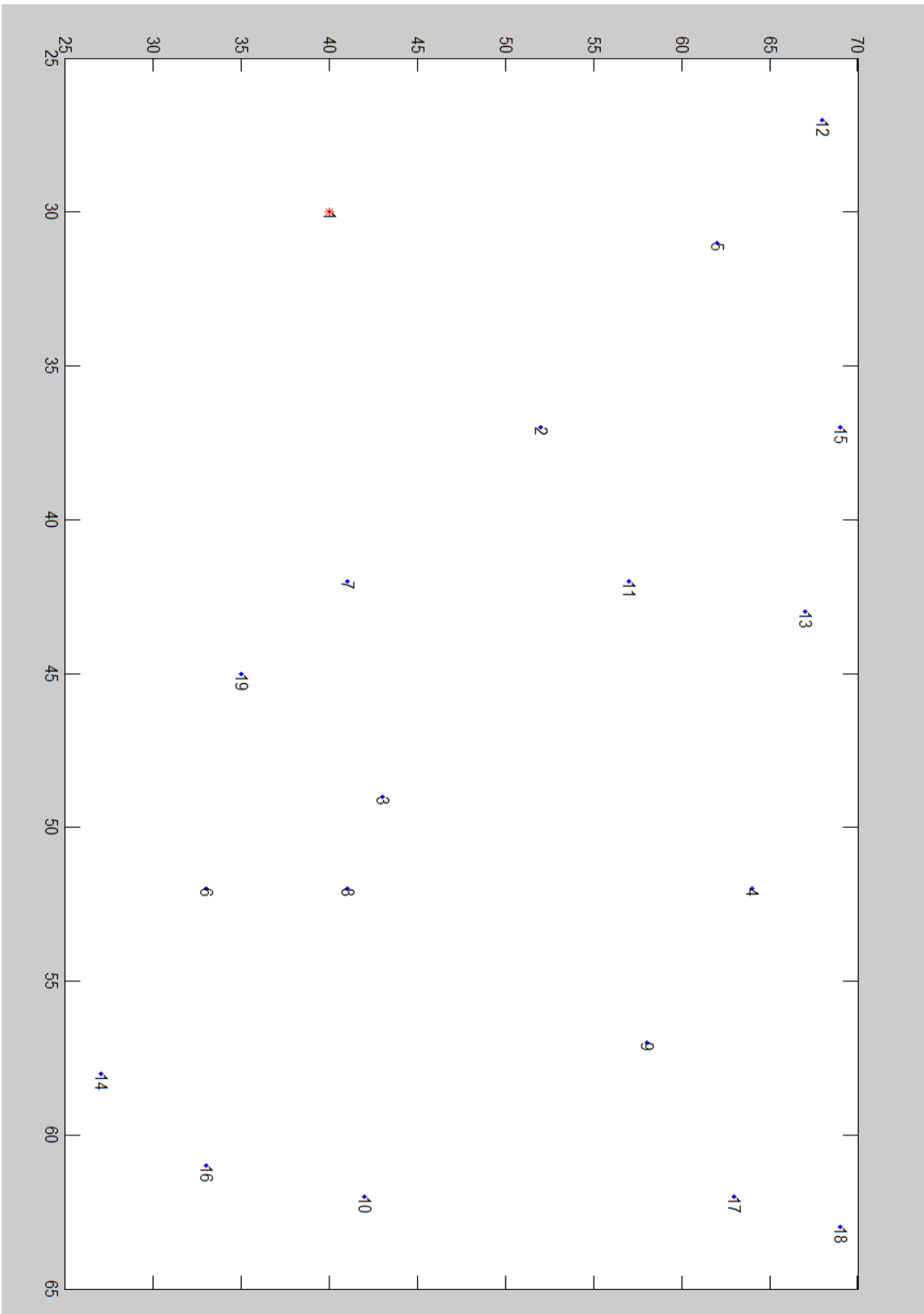
Elapsed time is 5860.173304 seconds. Infeasible S=9.



Elapsed time is 10420.807945 seconds. Infeasible, S=10.

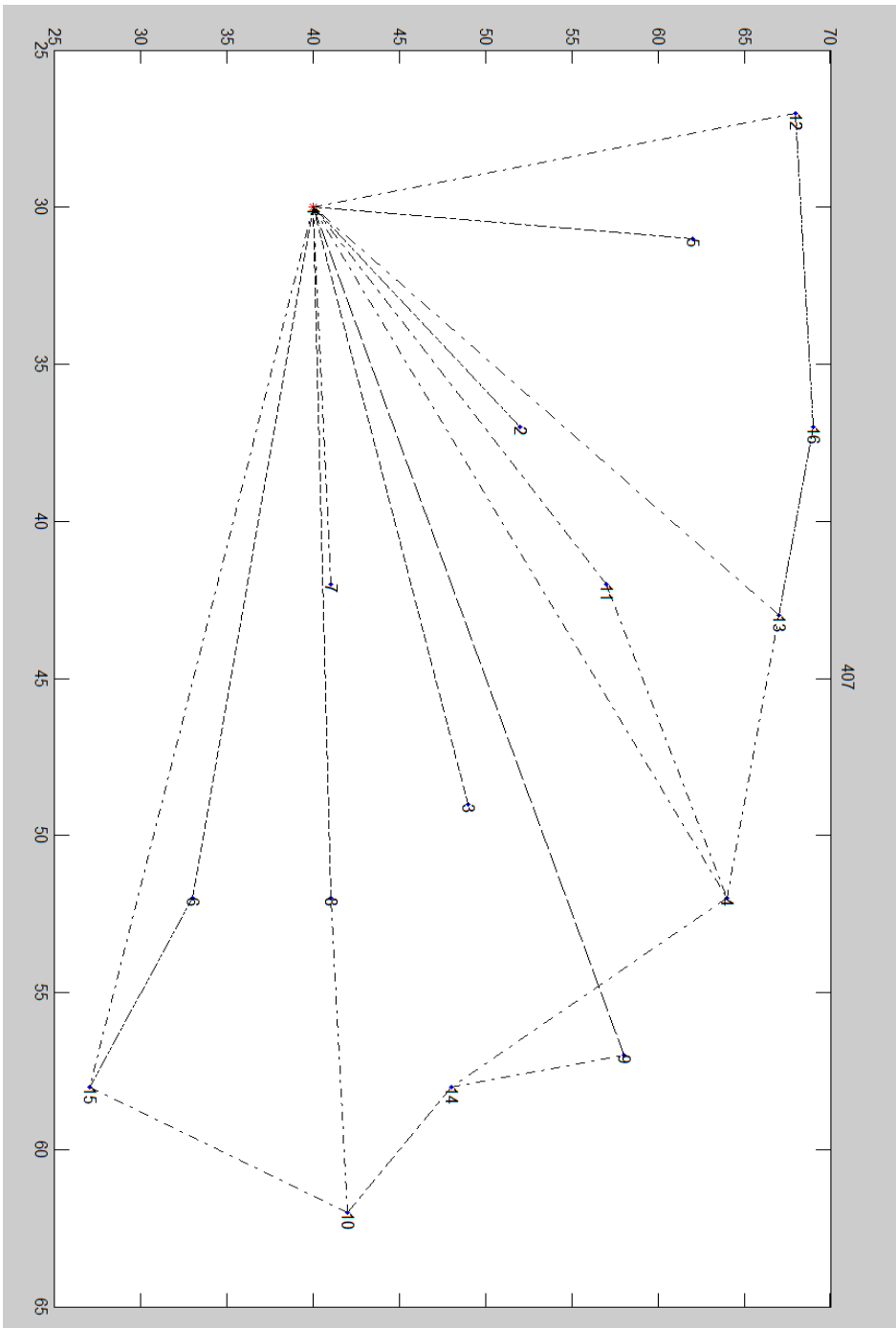


Elapsed time is 14615.931227 seconds. No retorna solución excede los límites de memoria en 32-bits S=11.

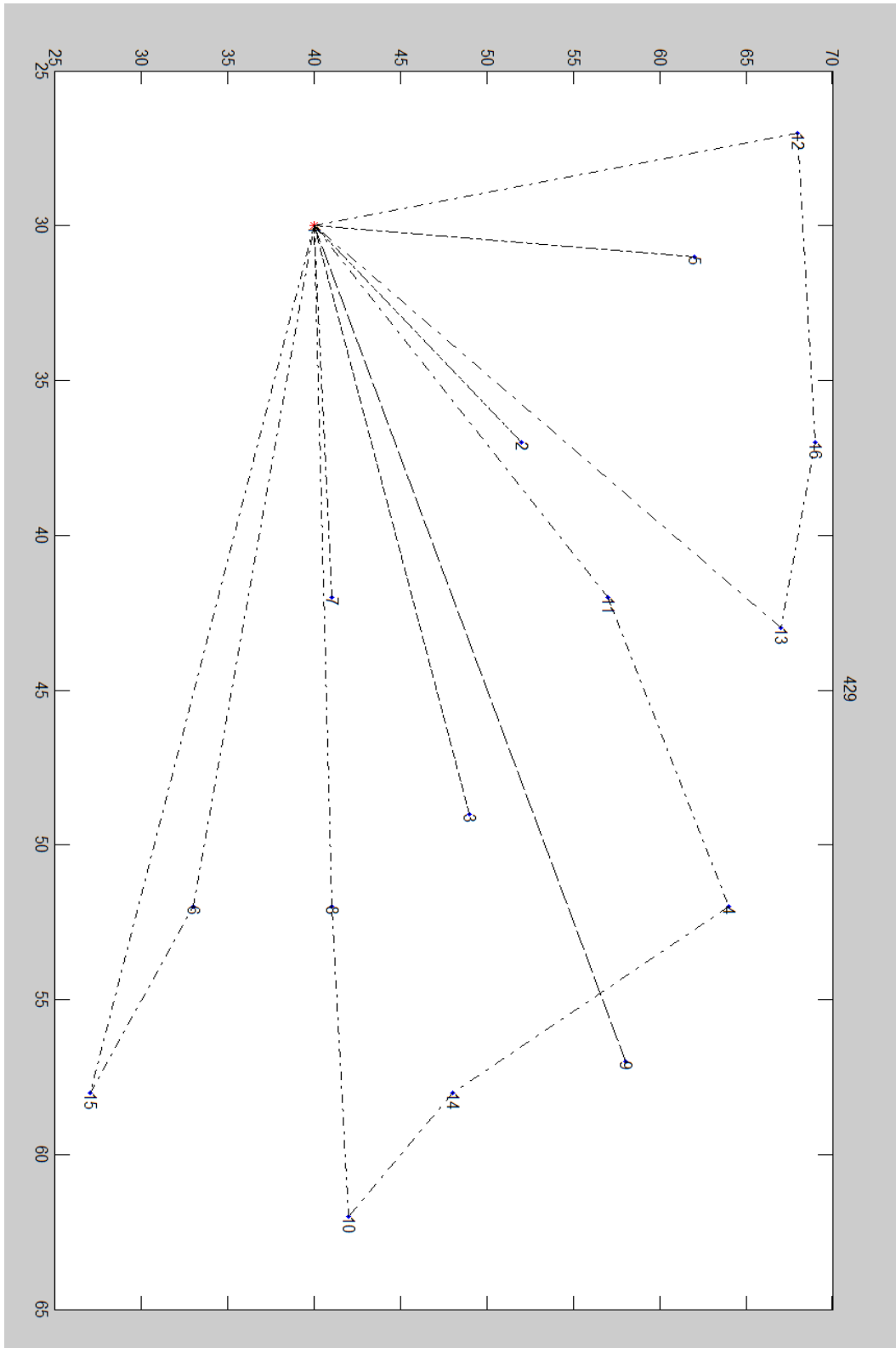


Anexo M. Resultados P\_n16\_k8 descompuesto

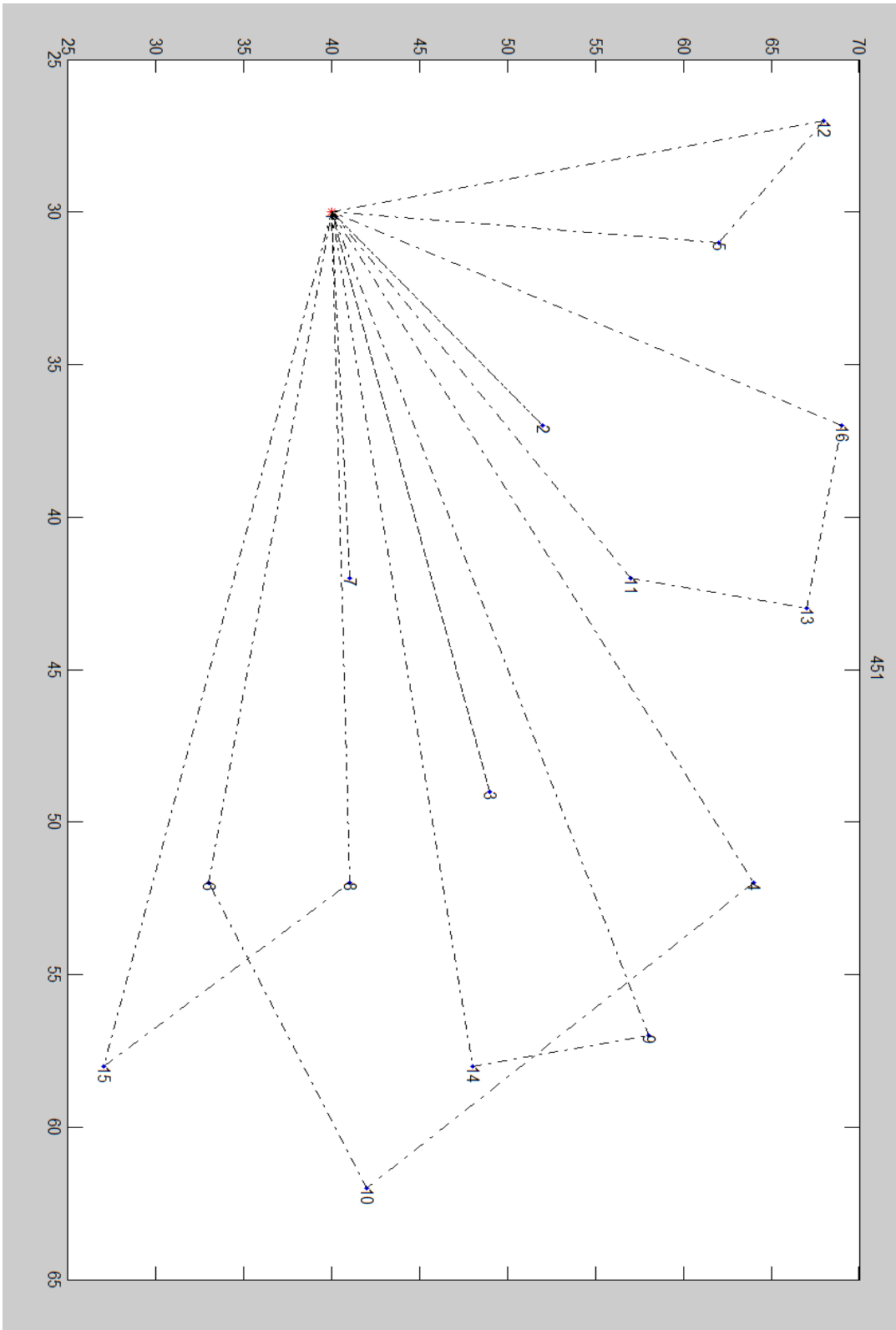
Elapsed time is 1.241541 seconds. Solución infactible S=2.



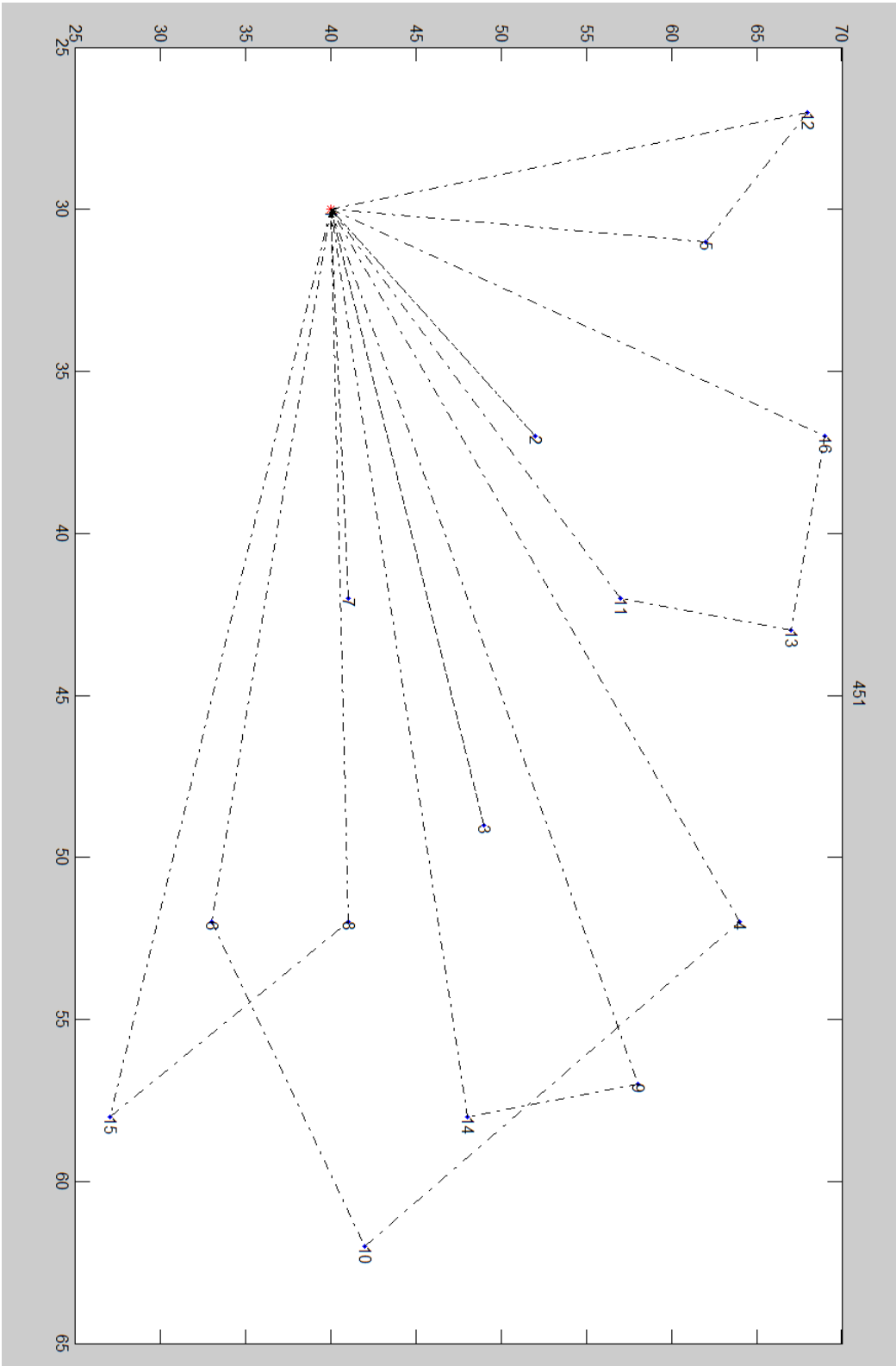
Elapsed time is 1.317767 seconds. Solución Infactible S=3



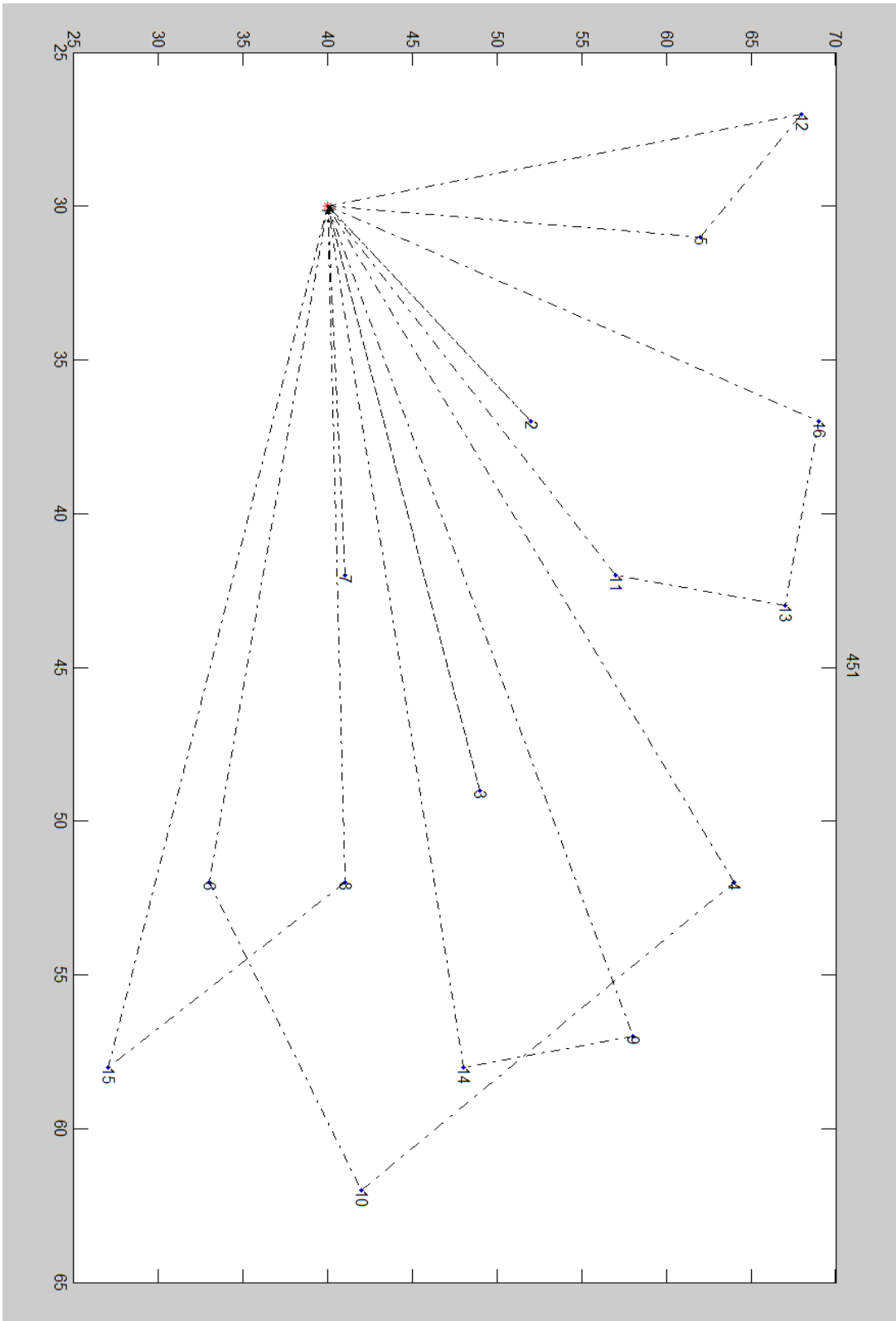
Elapsed time is 18.120266 seconds. Solución factible óptima S=4.



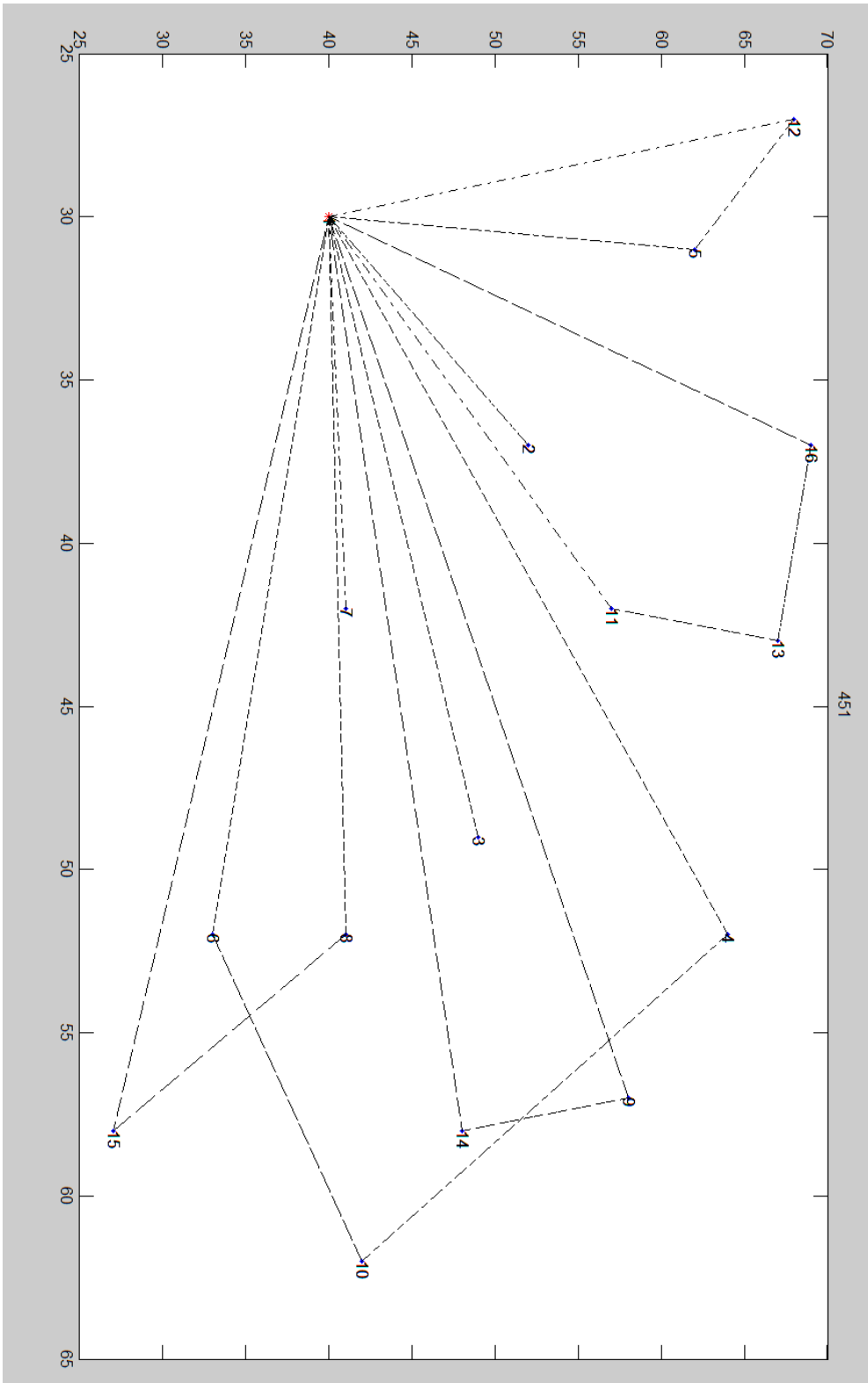
Elapsed time is 70.879267 seconds. Solution factible optima S=5.



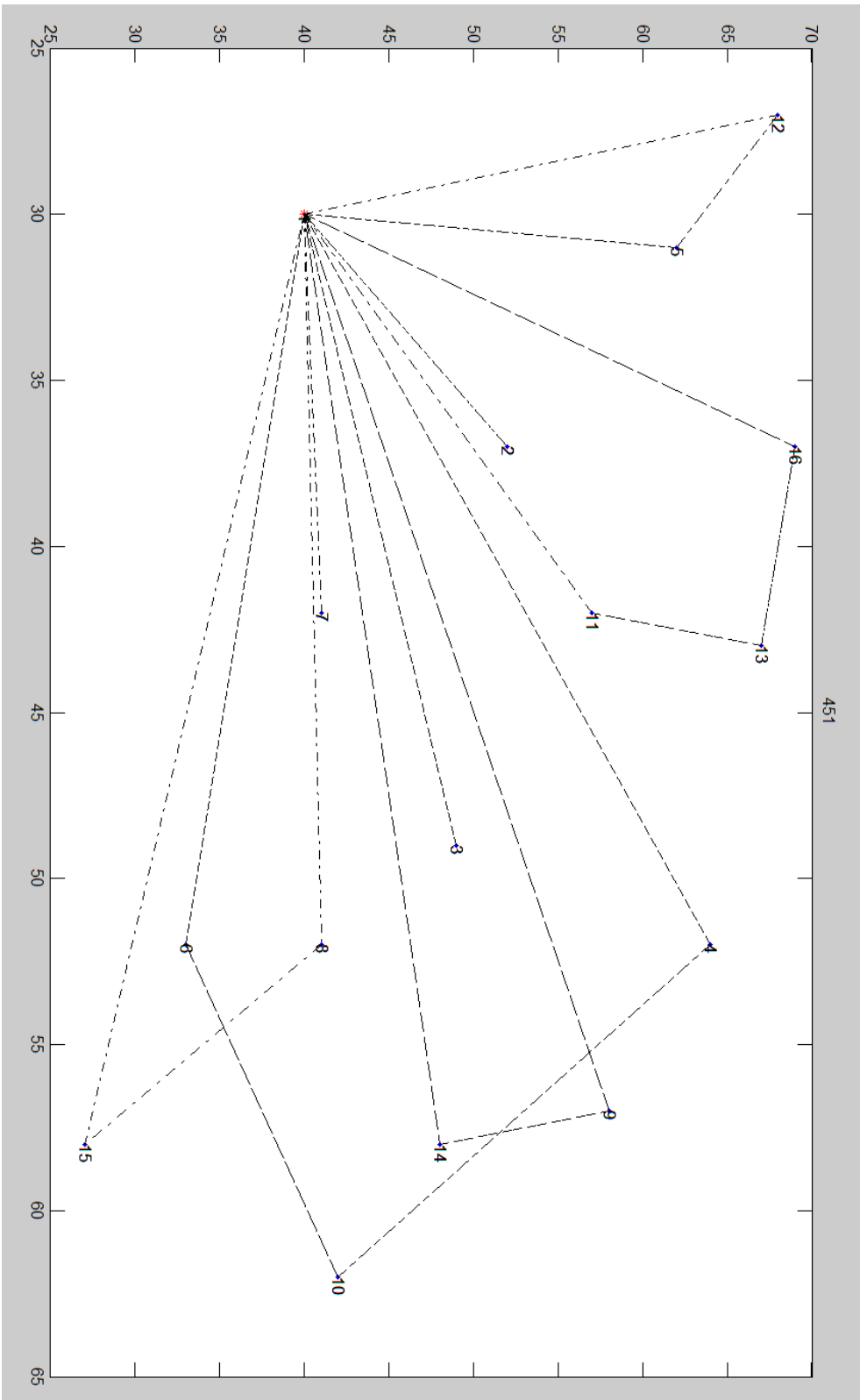
Elapsed time is 98.940068 seconds. Solution factible optima S=6.



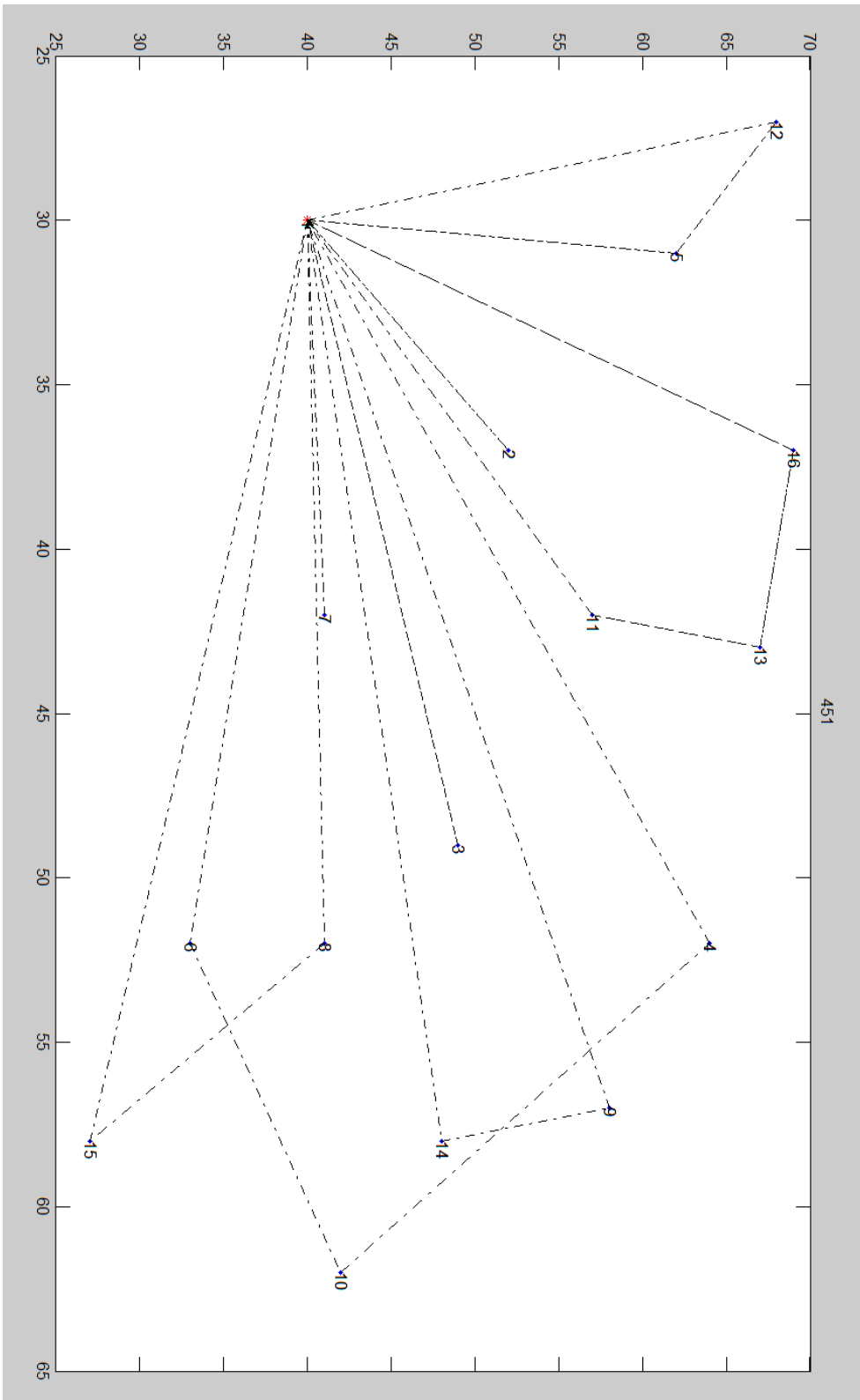
Elapsed time is 184.069090 seconds. Solución optima factible S=7.



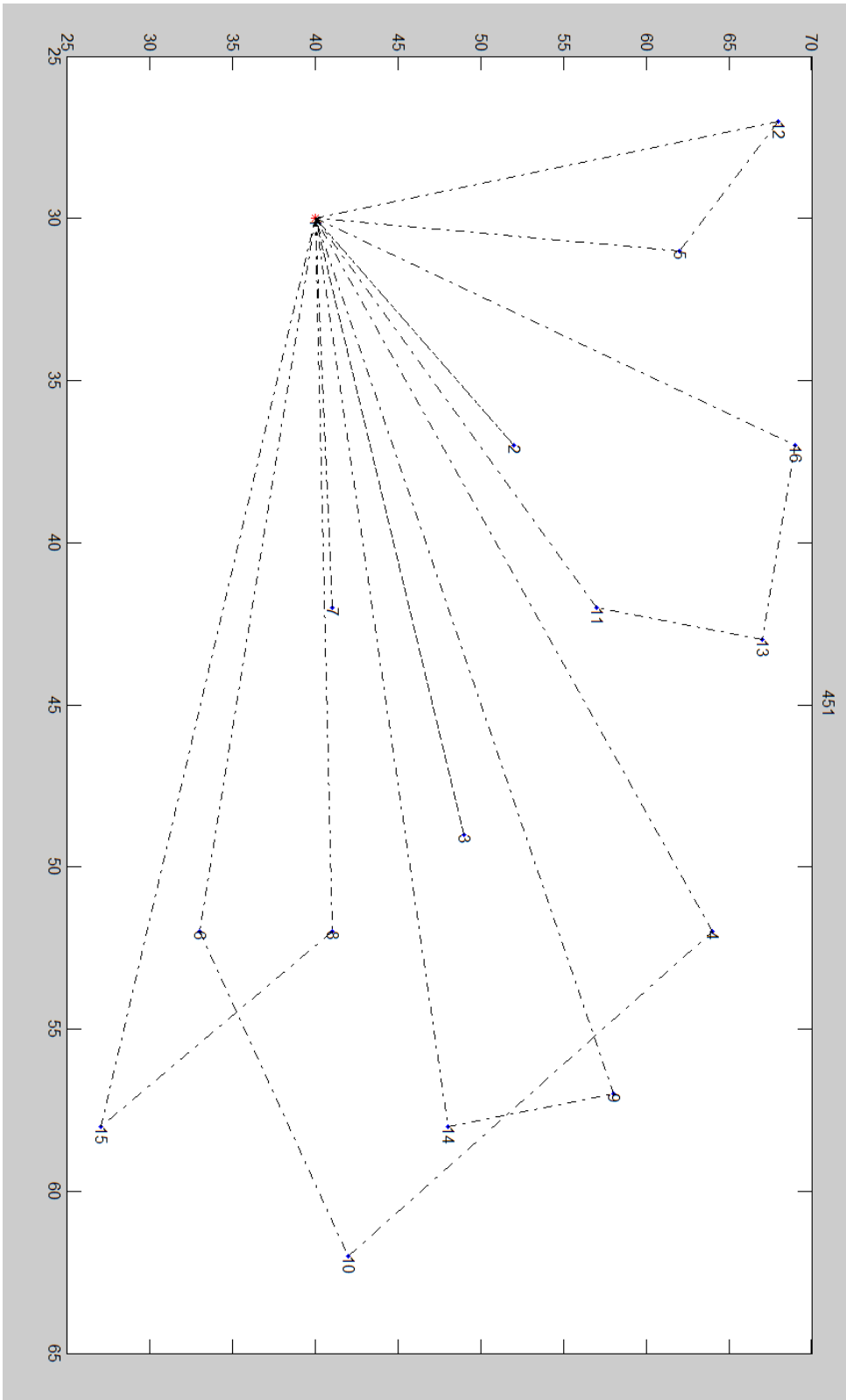
Elapsed time is 309.443216 seconds. Solución factible optima S=8.



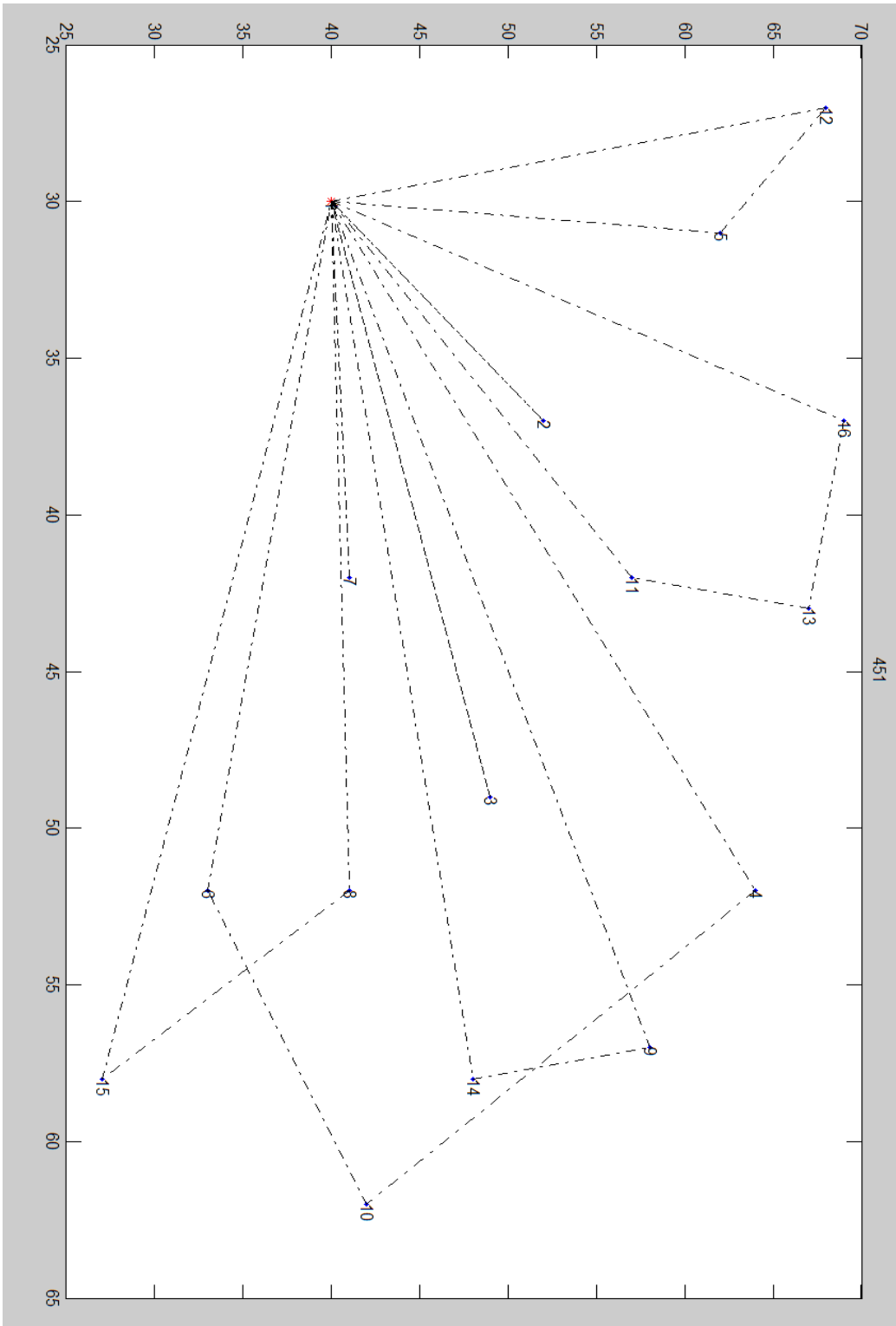
Elapsed time is 372.416432 seconds. Solución factible optima S=9.



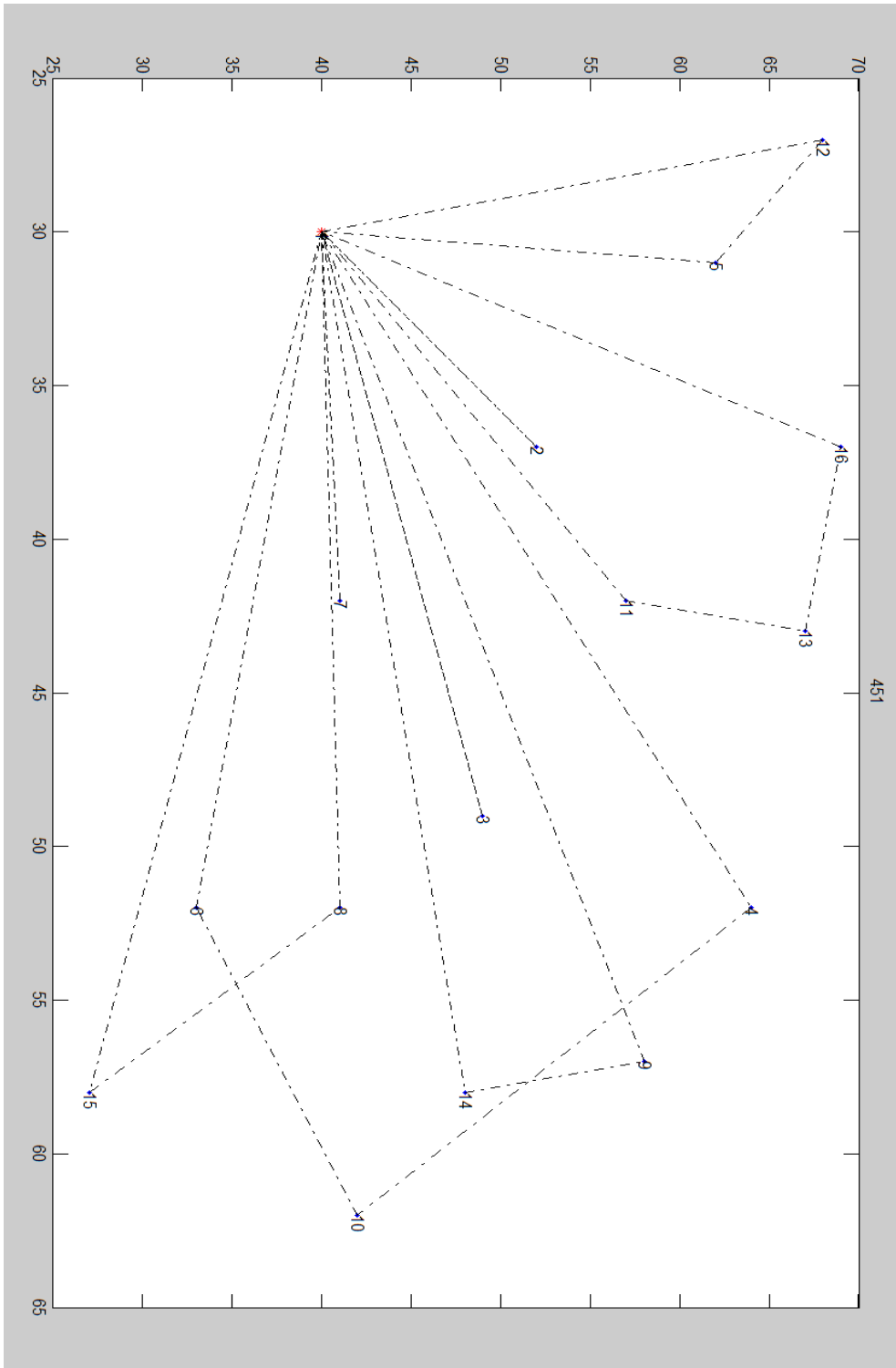
Elapsed time is 441.013554 seconds. Solución factible óptima S=10.



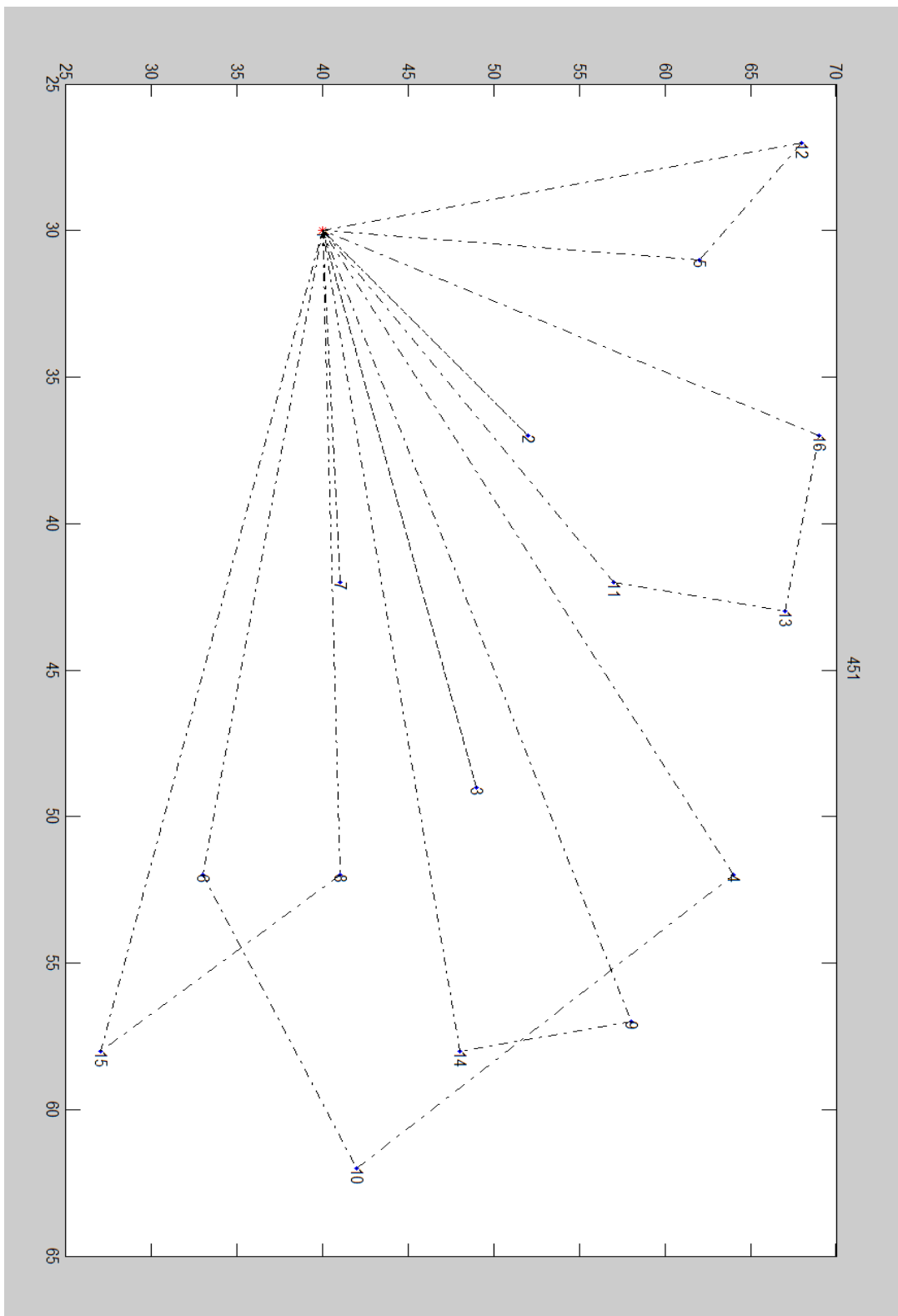
Elapsed time is 449.188572 seconds. Solución factible óptima S=11.



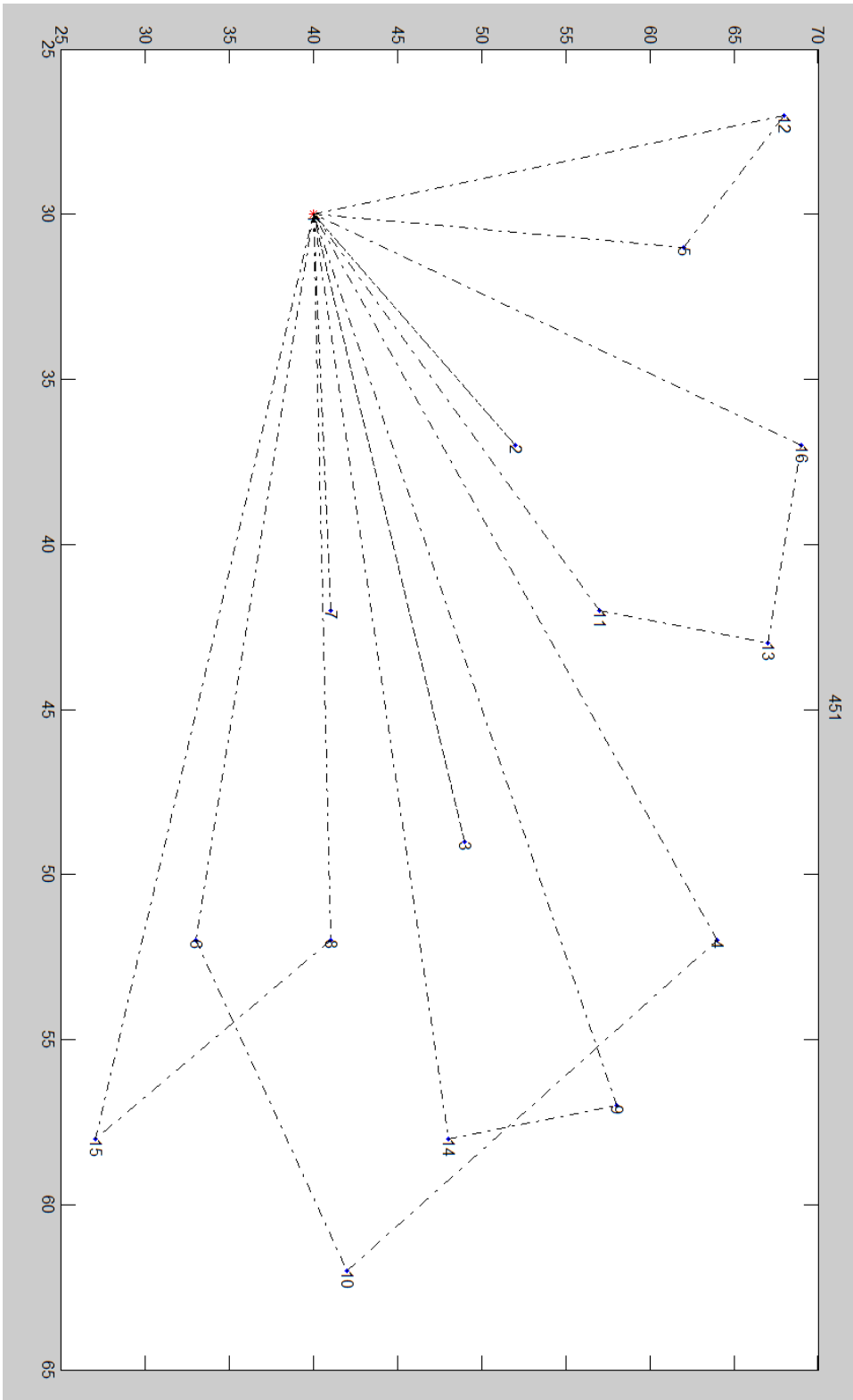
Elapsed time is 454.490892 seconds. Solución factible óptima S=12.



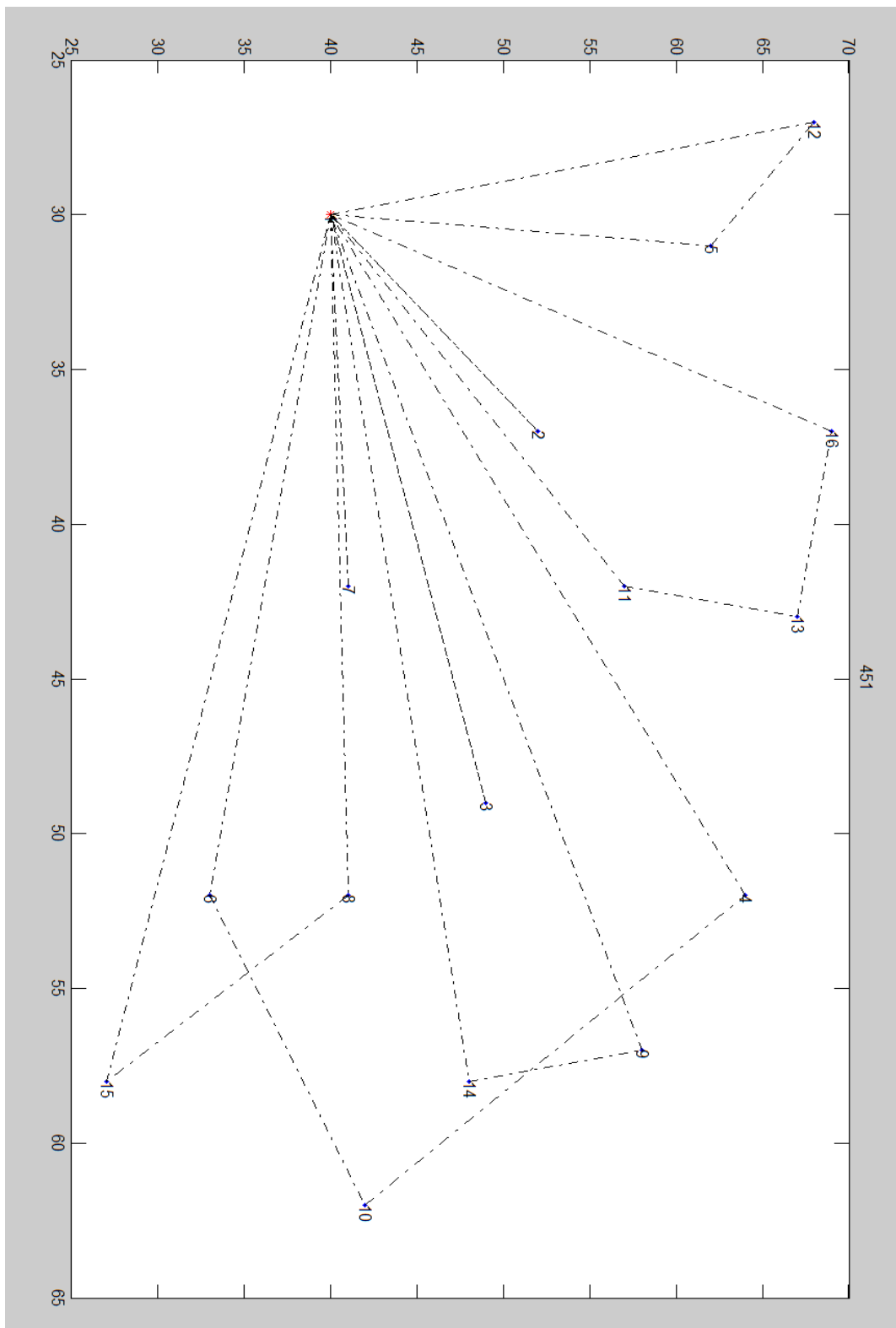
Elapsed time is 338.853525 seconds. Solución factible óptima S=13



Elapsed time is 480.120406 seconds. Solución Factible optima, S=14

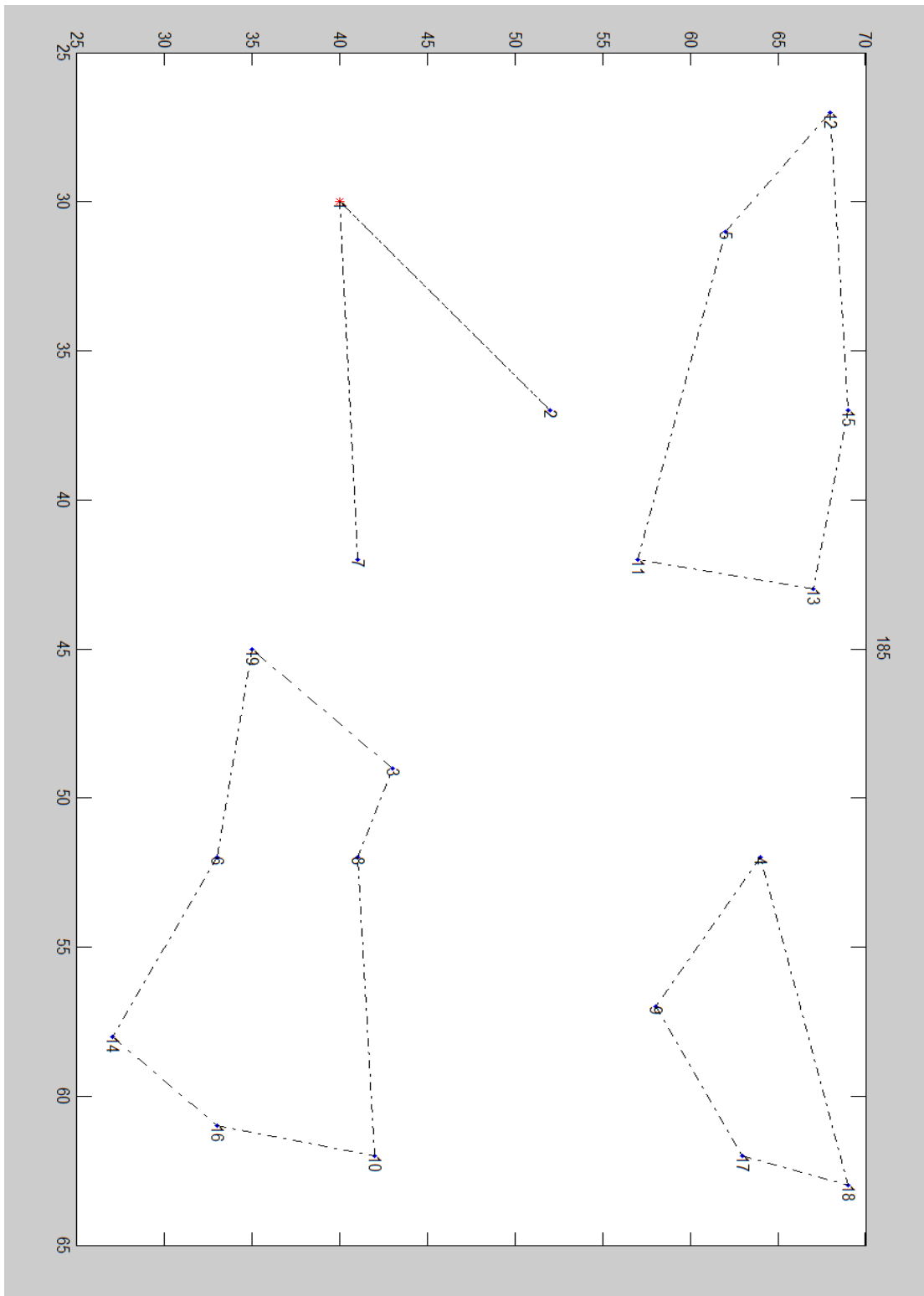


Elapsed time is 441.558046 seconds. Factible optima S=15

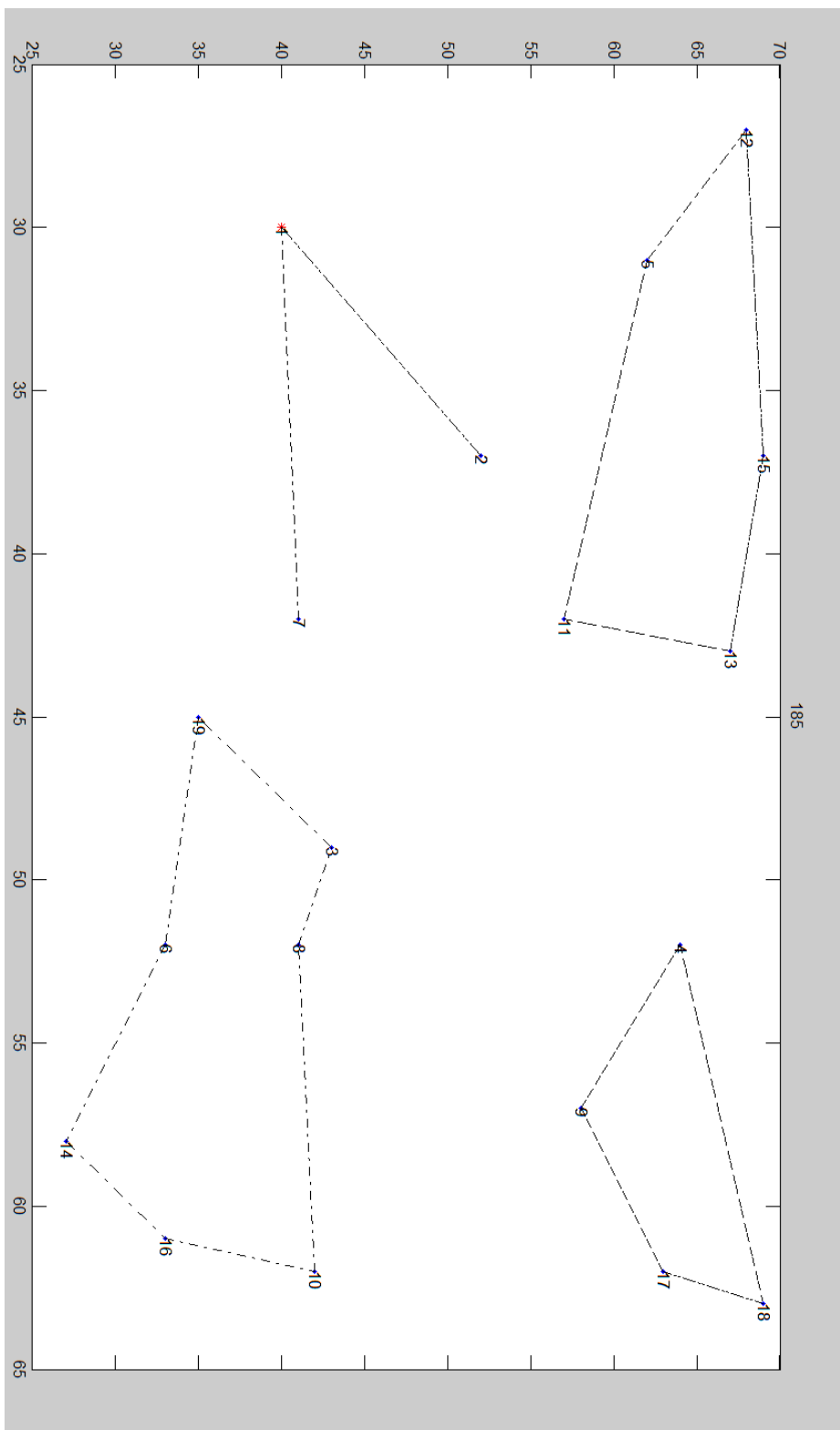


Anexo N. Resultados P\_n19\_k2 descompuesto

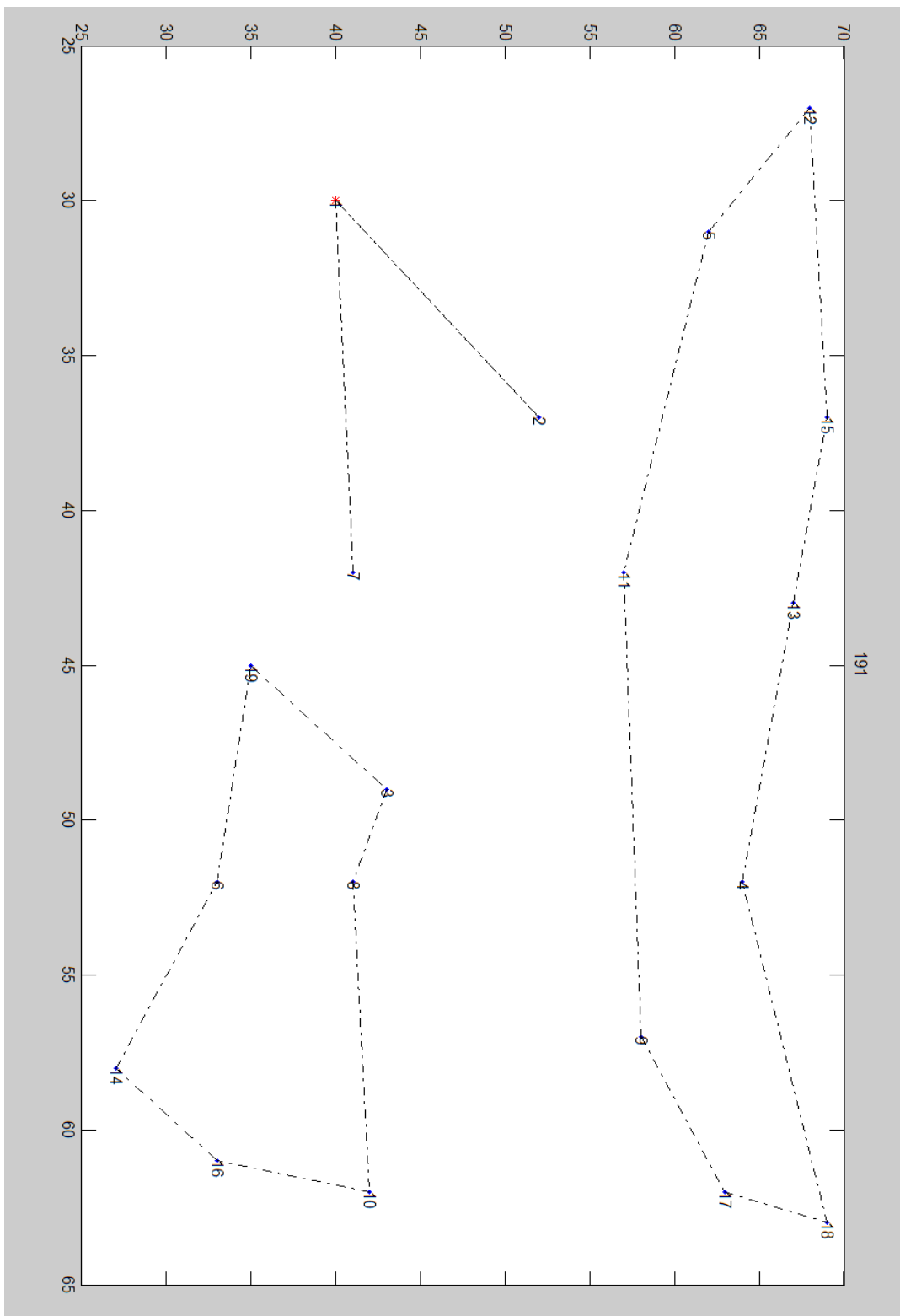
Elapsed time is 1.258734 seconds. Infeasible S=2.



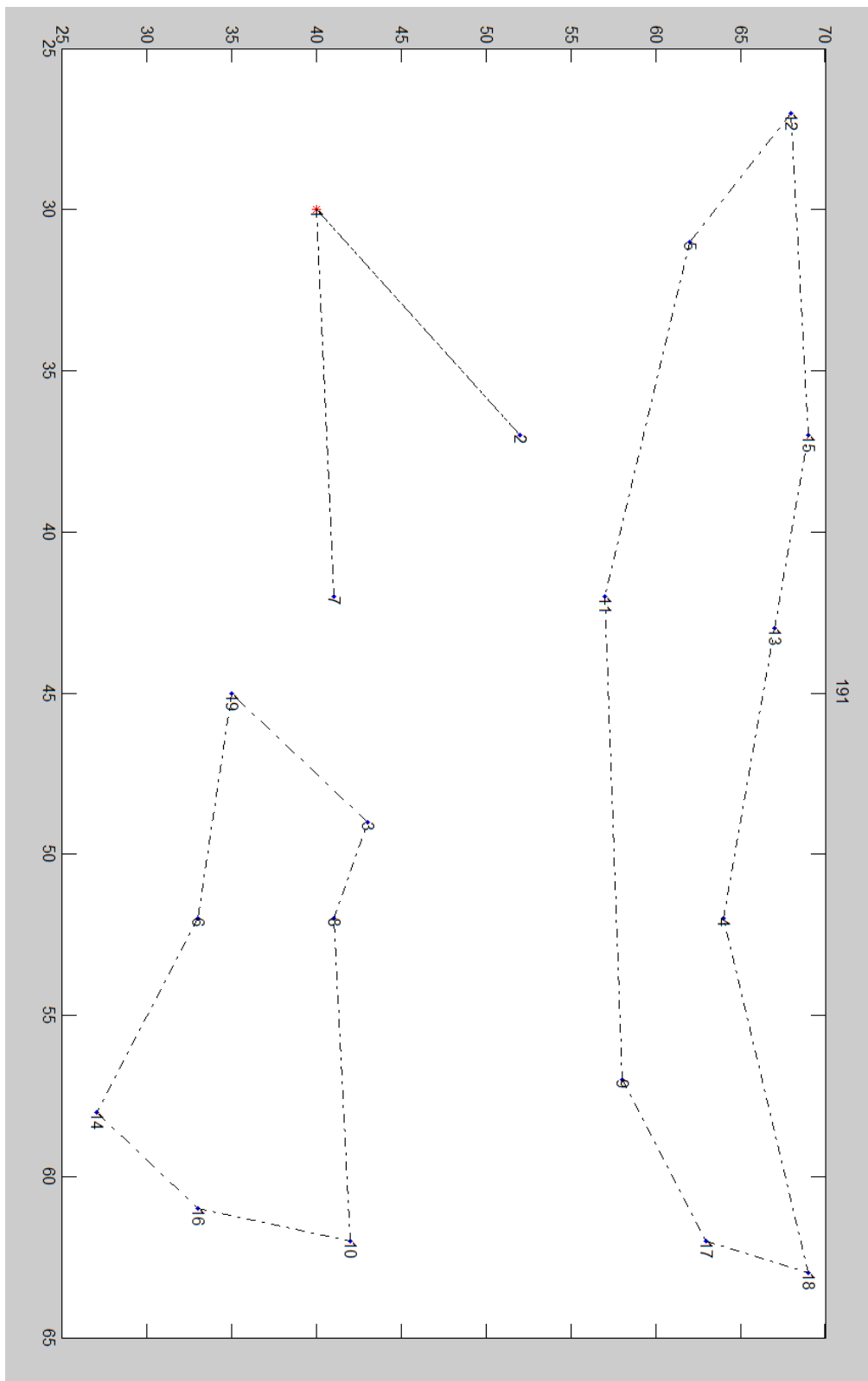
Elapsed time is 1.770973 seconds. Solución infactible S=3.



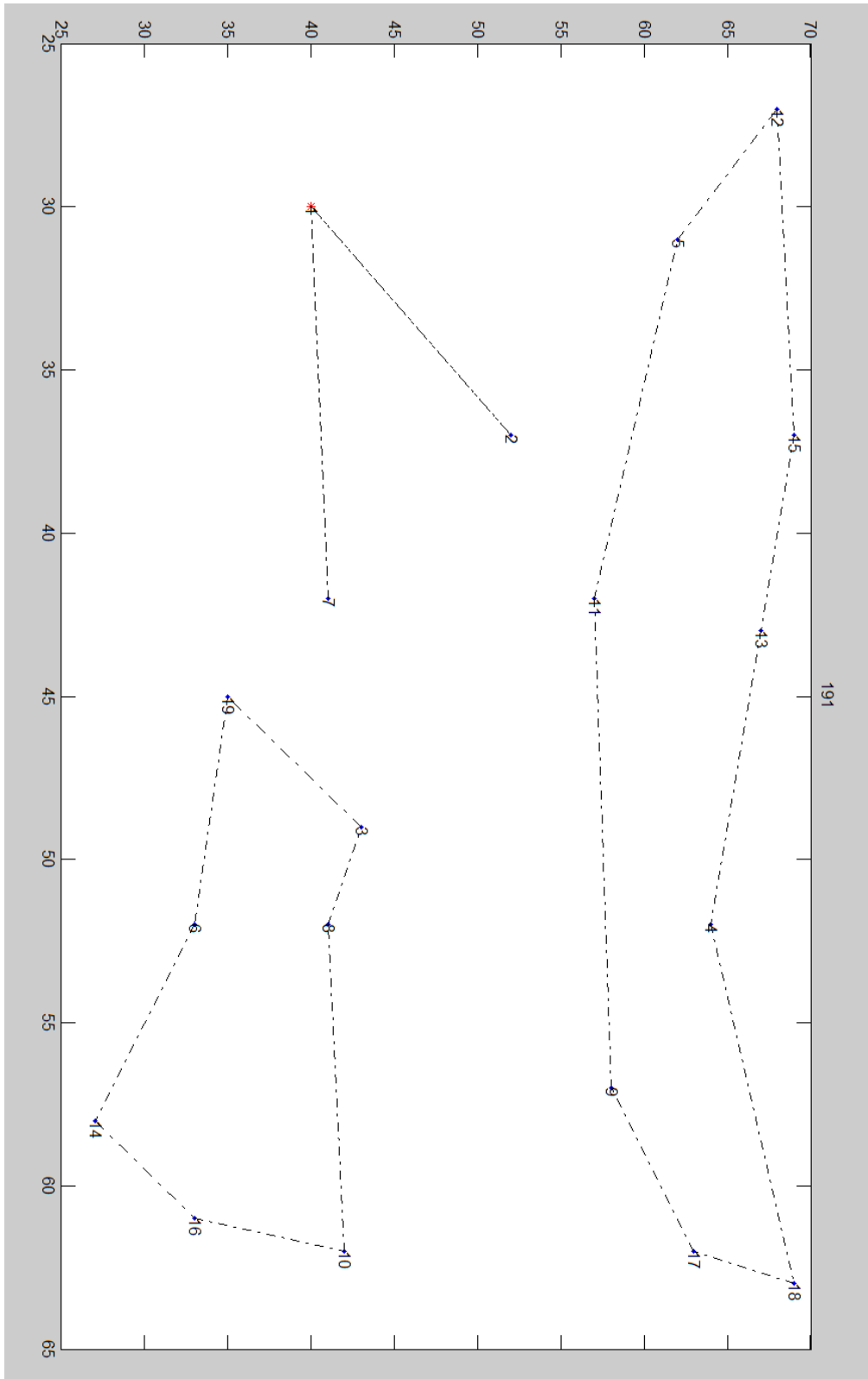
Elapsed time is 8.469487 seconds. Solución infactible S=4.



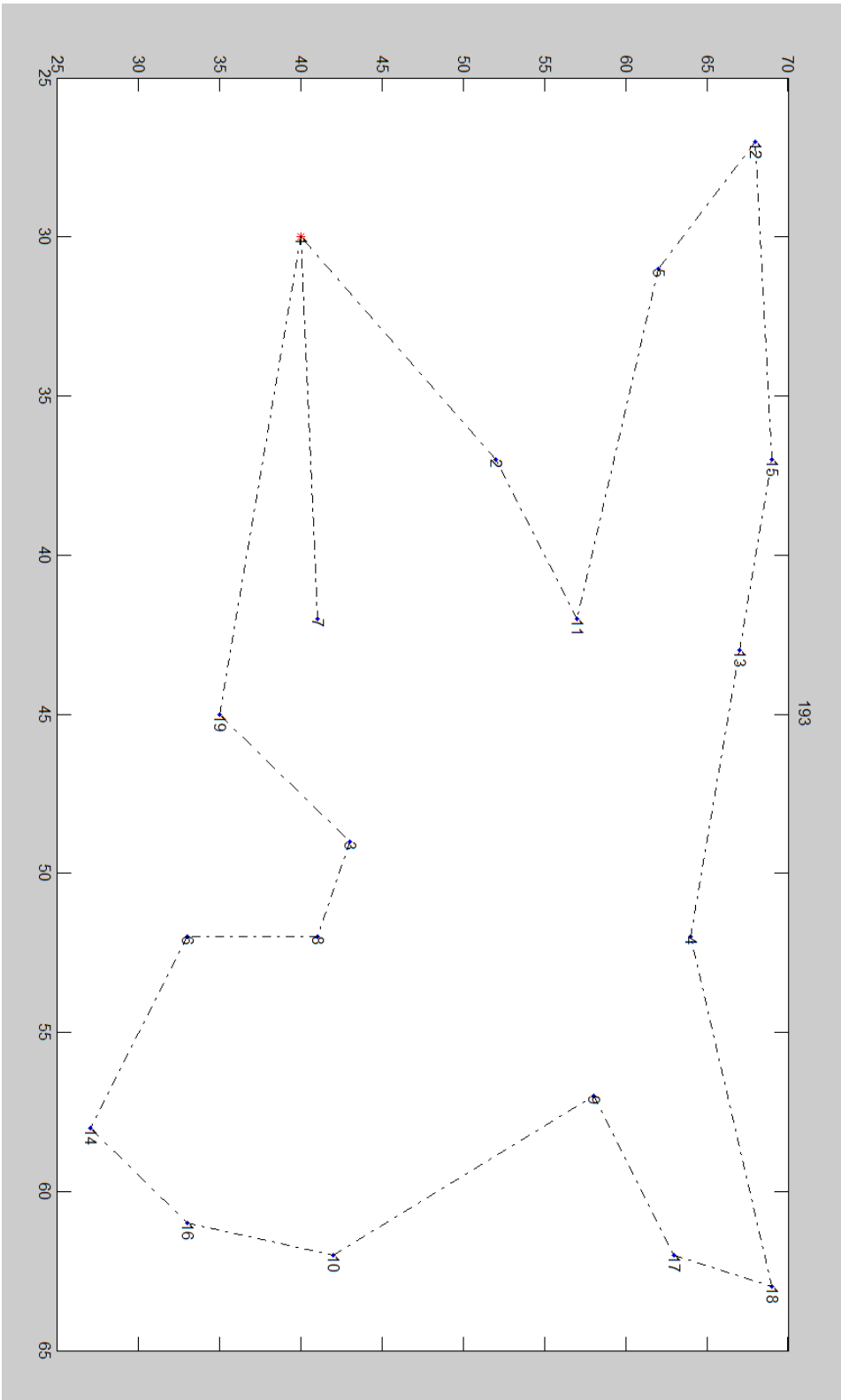
Elapsed time is 74.268784 seconds. Solución infactible S=5.



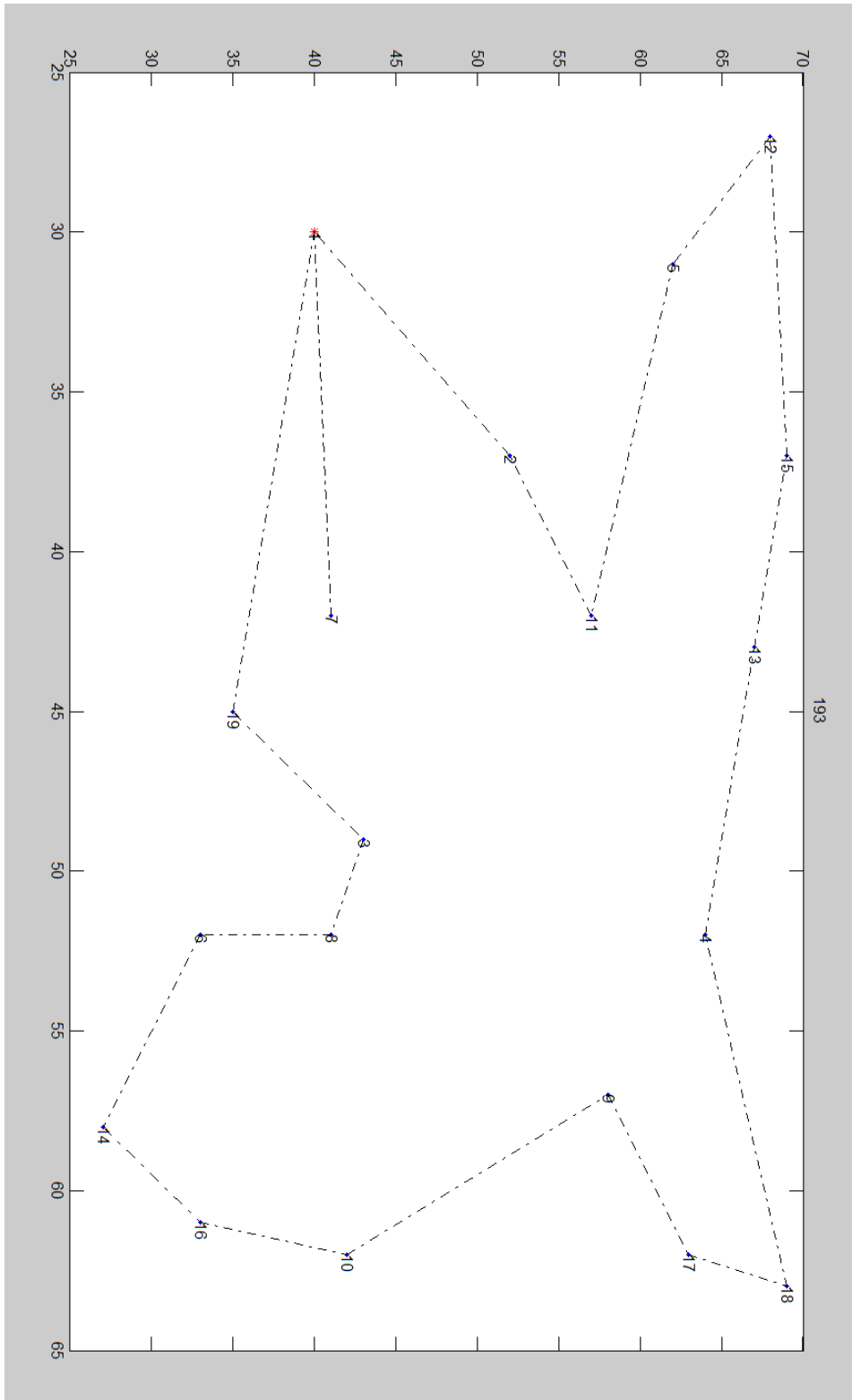
Elapsed time is 516.770890 seconds. Solución infactible S=6.



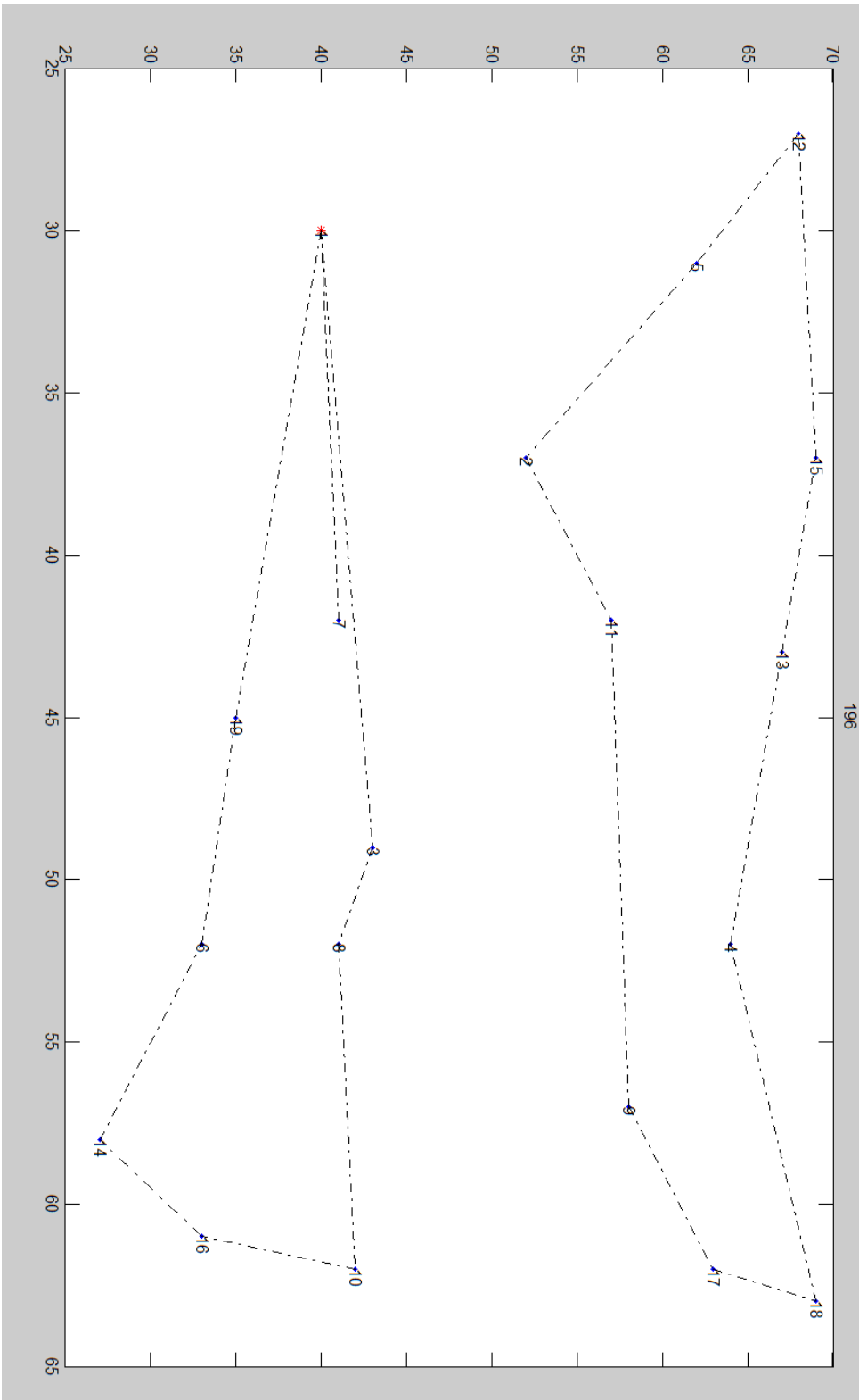
Elapsed time is 4051.570484 seconds. Infeasible S=7



Elapsed time is 16539.025680 seconds. Solución Infactible S=8.



Elapsed time is 46850.504278 seconds. Infeasible S=9.



Elapsed time is 31131.042500 seconds. Solución infactible S=10.

