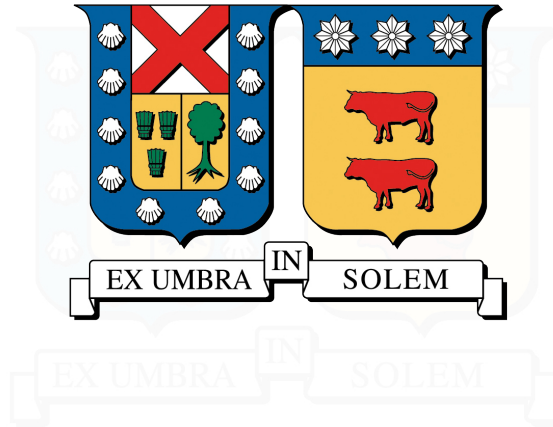


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INDUSTRIAS
VALPARAISO - CHILE



**UN PROBLEMA DE WEBER CON UNA FUNCIÓN DE COSTOS CONTINUA
POR PARTES**

GABRIELA PAULINA IRIARTE KAMP

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL INDUSTRIAL

Y AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INDUSTRIAL

PROFESOR GUÍA : DR. PABLO ESCALONA R.
PROFESOR CORREFERENTE 1 : DR. ALEJANDRO ANGULO C.
PROFESOR CORREFERENTE 2 : SR. RAÚL STEGMAIER B.

ENERO 2017



A mis Padres

AGRADECIMIENTOS

Quiero agradecer a mis padres por darme la oportunidad de estudiar, y no echarme de la casa aún, a mis hermanas por acompañarme, a mis sobrinas por nunca preguntarme como va la tesis, y a todos los que han estado por allí.



RESUMEN EJECUTIVO

Este trabajo analiza la localización de un centro de distribución en un área urbana usando el problema single-source de Weber con costos fijos por partes y continuos para encontrar la localización óptima global. El costo fijo es caracterizado por medio de un método de interpolación de Kriging. Para hacer el costo fijo tratable, aproximamos esta interpolación con una función por partes y continua que sea convexa en cada parte, a través del uso la triangulación de Delaunay. Presentamos tres propuestas de solución para el problema: un método de descomposición, un método cónico de descomposición y una reformulación monolítica cónica para resolver a optimalidad el problema de localización de una única instalación considerando una función de costos fijos por partes y continuos. A pesar que este enfoque continuo no garantiza la localización óptima global factible, permite delimitar una zona donde se puede intensificar la búsqueda de puntos factibles.

Para las instancias que fueron testeadas, los resultados computacionales muestran que el enfoque continuo encuentra una mejor localización que el enfoque discreto en el 23,25 % de las instancias y que el método de descomposición es el mejor, en términos de tiempo CPU, con un speedup promedio de 7.98x y 7.72x sobre el método de descomposición cónica y la reformulación monolítica cónica, respectivamente. Además, se presenta una comparación entre dos métodos de solución para el problema de Weber sin considerar los costos fijos, donde los resultados computacionales muestran que es mejor el algoritmo de Weiszfeld por sobre una reformulación cónica.

Keywords: Problema de Weber, Costos Fijos, Triangulación de Delaunay, Interpolación de Kriging.



ABSTRACT

This paper analyze the location of a distribution center in an urban area using a single-source Weber problem with continuous piecewise fixed cost to find a global optimal location. The fixed cost is characterized by a Kriging interpolation method and, to make the fixed cost tractable, we approximate this interpolation with a continuous piecewise function that is convex in each piece, using Delaunay triangulation. We present a *decomposition formulation*, a *decomposition conic formulation* and a *conic logarithmic disaggregated convex combination model* to optimally solve the single-source Weber problem with continuous piecewise fixed cost. Although our continuous approach does not guarantee the global feasible optimal location, it allow us to delimit a zone where we can intensify the search of feasible points.

For instances we tested, computational results show that our continuous approach found better locations than the discrete approach in 23,25 % of the instances and that the decomposition formulation is the best one, in terms of CPU time.

Keywords: Weber Problem, Fixed Costs, Delaunay Triangulation, Kriging Interpolation.



Índice de Contenidos

1. Introducción y Objetivos	1
1.1. Objetivos	2
1.1.1. Objetivo General	2
1.1.2. Objetivos Específicos	2
1.2. Alcance	2
1.3. Metodología	2
2. Marco Teórico	5
2.1. Revisión Bibliográfica	5
2.2. Problema de Weber	6
2.3. Algoritmo de Weiszfeld	7
2.3.1. Pseudocódigo del Algoritmo de Weiszfeld	7
2.4. Métodos de Interpolación de Kriging	8
2.4.1. Simple Kriging	8
2.4.2. Ordinary y Universal Kriging	9
2.4.3. Intrinsic Kriging	11
2.5. Triangulación de Delaunay	12
2.5.1. Relación de la triangulación de Delaunay con el diagrama de Voronoi	12
2.5.2. Propiedades	12
2.5.3. Construcción del Algoritmo	13
3. Formulación del Modelo	15
3.1. Problema Single-Source de Weber con Función de Costos Fijos Continuos por Partes	15
4. Propuesta de Solución	17
4.1. Logarithmic Disaggregated Convex Combination Model con Conos	17
4.2. Método de Descomposición	18
4.3. Método de Descomposición Cónica	19
5. Estudios Computacionales	21
5.1. Problema Single-Source de Weber sin Costos Fijos	21
5.2. Conjunto de Prueba	22
5.3. Modelo Discreto: Uncapacitated Facility Location Problem	27
5.4. Ejemplo Ilustrativo para un Caso Industrial	28
6. Conclusiones	31
Bibliografía	33
A. Glosario	37
B. Data del Ejemplo Ilustrativo	39

C. Códigos de Propuesta de Solución	43
C.1. Código Logarithmic Disaggregated Convex Combination Model con Conos	43
C.2. Código Método de Descomposición	45
C.3. Código Método de Descomposición Cónica	46
D. Problema Cónico de Weber	49
E. Paper Realizado	51
E.1. INTRODUCTION	51
E.2. RELATED WORK	52
E.3. MODEL FORMULATION	53
E.3.1. Single-Source Weber Problem with Continuous Piecewise Fixed Costs	53
E.4. SOLUTION APPROACH	54
E.4.1. Conic Logarithmic Disaggregated Convex Combination Model	54
E.4.2. Decomposition Formulation	55
E.4.3. Decomposition Conic Formulation	56
E.5. COMPUTATIONAL STUDY	56
E.5.1. Test Set	57
E.5.2. Discrete Model: Uncapacitated Facility Location Problem	59
E.6. CONCLUSIONS	60

Índice de Tablas

5.1. Resultados del Problema de Weber sin costos fijos	21
5.2. Speedup Promedio del tiempo CPU de $\min_{k \in K}\{(SP0(k))\}$ sobre $(DlogCP0)$	27
5.3. Porcentaje de mejora del modelo continuo sobre el UFLP.	27
5.4. Tiempos CPU del Ejemplo Ilustrativo.	28
B.1. Data de los nodos de información	39
B.2. Data de los nodos clientes	40
E.1. Average Speedup in CPU time of $\min_{k \in K}\{(SP0(k))\}$ over $(DlogCP0)$	58
E.2. Percentage of improvement for the continuous model over the UFLP.	59



Índice de Figuras

5.1. Primer refinamiento.	22
5.2. Segundo refinamiento.	23
5.3. Tercer refinamiento.	23
5.4. Cuarto refinamiento.	24
5.5. Perfil de las Performance.	25
5.6. Performance ratio vs Número de la instancia, para el primer refinamiento.	25
5.7. Performance ratio vs Número de la instancia, para el segundo refinamiento.	26
5.8. Performance ratio vs Número de la instancia, para el tercer refinamiento.	26
5.9. Performance ratio vs Número de la instancia, para el cuarto refinamiento.	27
5.10. Localización del DC.	28
E.1. First refinement.	57
E.2. Fourth refinement.	57
E.3. Performance Profile.	58



1 | Introducción y Objetivos

La localización de un centro de distribución (DC) dentro de un área urbana, considerando los costos de transporte y de instalación, puede ser tratada como un problema Uncapacitated Facility Location Problem (UFLP), considerando un enfoque discreto, o como un problema de Weber con costo fijo, considerando un enfoque continuo. Se sabe que la solución de un problema UFLP es factible pero no necesariamente óptimo, debido al uso de un conjunto incompleto de posibles localizaciones. Por el otro lado, el problema de Weber con costos fijos entrega una localización óptima que es altamente probable no factible.

Este trabajo analiza la instalación de un único DC dentro de un área urbana, utilizando el problema single-source de Weber con costo fijo para encontrar una localización óptima que nos permita delimitar una zona, alrededor de la localización óptima encontrada anteriormente, que sea más pequeña que la zona original. De esta manera, se puede focalizar la búsqueda de puntos factibles, obteniendo así un conjunto de posibles locaciones que sea más exacto y completo, para que cuando sea aplicado un problema UFLP se encuentre la *localización óptima factible*.

Según nuestro conocimiento, son pocos los papers que incluyen el costo fijo al problema de Weber. En estos trabajos, el costo fijo se ha incluido considerando diferentes aproximaciones, teniendo un costo fijo como un costo constante en todo el plano (Brimberg et al., 2004), como dependiente del lugar con un costo constante dentro de un polígono convexo específico (Brimberg y Salhi, 2005), (Hosseini-zhad et al., 2015), o como una proporción entre el costo fijo de dos zonas y de su distancia relativa (Luis et al., 2015). Considerar que un área urbana puede ser particionada en un conjunto finito de polígonos convexos, cada uno con un costo fijo constante, es considerada una buena primera aproximación para caracterizar la naturaleza variable de sus costos fijos. En este trabajo se propone que el costo de instalación en cada polígono convexo sea una función de sus vértices, permitiendo así modelar de mejor forma la naturaleza variable de los costos de instalación dentro de un área urbana.

Es este trabajo, se considera la instalación de un único centro de distribución porque se sabe que, dentro de un área urbana, el costo de instalar un centro de distribución extra tiende a ser mayor que los ahorros generados por un menor costo de transporte.

El objetivo de este trabajo es encontrar la mejor formulación para localizar un solo centro de distribución en un área urbana, donde los costos fijos dependen de la localización en una forma continua. El costo fijo es caracterizado a través de un método de interpolación de Kriging usando un conjunto de puntos donde el costo fijo es conocido. Para que la formulación sea tratable, se aproxima la interpolación con una función continua por partes donde cada parte sea convexa. Esto es generado a través de una combinación convexa de los vértices de una malla creada con la triangulación de Delaunay. El problema single-source de Weber con costos fijos continuos por partes es formulado como un problema Mixed Integer Non Linear Problem (MINLP). Aprovechando la estructura del problema se proponen tres enfoques de solución. El primer enfoque considera una reformulación cónica del problema single-source de Weber con costos fijos continuos por partes usando el modelo *logarithmic disaggregated convex combination model*. El segundo enfoque consiste en un método de descomposición, donde se resuelve un problema convexo no-lineal por cada triángulo de Delaunay, y se determina la solución óptima usando enumeración completa. El último enfoque considera una reformulación cónica de cada sub-problema del método de descomposición anterior. Considerando los diferentes enfoques, se implementan una serie de experimentos para comparar el desempeño de los tres modelos basándose en el tiempo CPU. Además, en este trabajo son analizadas dos formas de solucionar el

problema de Weber sin considerar los costos de instalación: el algoritmo de Weiszfeld y una reformulación cónica del problema de Weber.

Las principales contribuciones de este trabajo son (i) una nueva forma de representar la naturaleza variable de los costos fijos en un área urbana e (ii) identificar la mejor propuesta de solución para el problema single-source de Weber con costos fijos continuos por partes.

1.1. Objetivos

1.1.1. Objetivo General

“Encontrar la mejor formulación para localizar un único DC en un área urbana, donde los costos fijos dependen de la localización de una forma continua. Para caracterizar los costos de instalación, se utilizará un método de interpolación de Kriging. Debido a que la función entregada por la interpolación es compleja y no necesariamente convexa, se aproxima ésta mediante una combinación convexa de los vértices de una malla formulada a partir de la triangulación de Delaunay.”

1.1.2. Objetivos Específicos

- Recopilar modelos existentes del problema de Weber.
- Analizar modelos existentes del problema de Weber que consideren costos fijos.
- Investigar sobre los modelos existentes para caracterizar el valor de la tierra.
- Analizar enfoques para formular funciones por partes.
- Generar una función de costos fijos continuos por partes donde cada parte sea convexa.
- Analizar diferentes enfoques para resolver el problema single-source de Weber.
- Formular diferentes enfoques para resolver el problema single-source de Weber con costos fijos.
- Evaluar computacionalmente los modelos con y sin costos fijos.
- Comparar el desempeño de los modelos basados en su tiempo CPU.

1.2. Alcance

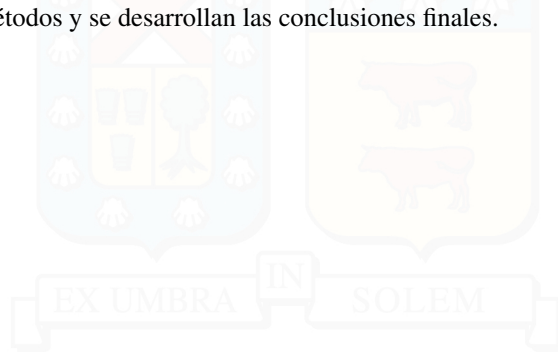
El tema de investigación posee un alcance de carácter descriptivo-exploratorio. Las razones de la naturaleza de este alcance radican en que, por una parte, la investigación explica y entrega una amplia descripción tanto de la forma en que se han abordado los modelos del problema de Weber a través de los años, como de los diversos enfoques para considerar la inclusión de costos fijos en el problema y las diferentes propuestas de solución, tanto algorítmicas como exactas, del problema de Weber general y con costos fijos. Por otra parte, constituye una investigación original que busca mejorar lo propuesto en investigaciones anteriores.

1.3. Metodología

La metodología de este trabajo corresponde, en una primera instancia, a la revisión bibliográfica de investigaciones asociadas al problema de Weber, a la inclusión de los costos fijos en dicho problema y a modelos geoestadísticos aplicados en aproximar el valor de la tierra, donde los objetivos son (i) generar

una fuente de información relevante, relacionada con los modelos propuestos, y (ii) analizar los diversos enfoques de valoración de la tierra para aplicar alguno de ellos en la generación de la función de costos fijos, considerando su variabilidad.

Posteriormente, se analiza el problema de Weber sin considerar los costos de instalación para establecer un método de solución como el mejor. Se propone una función de costos continua por partes que es convexa en cada parte. Luego se investigan tres propuestas distintas de solución para el problema de Weber con costos fijos. A continuación, se realizan estudios computacionales que permitirán establecer la propuesta de solución con mejor performance. Finalmente se obtienen los resultados computacionales, con los cuales se analizan los distintos métodos y se desarrollan las conclusiones finales.





2 | Marco Teórico

2.1. Revisión Bibliográfica

El problema de localización continua para ubicar una instalación es conocido como el problema single-source de Weber, descrito en [Weber y Friedrich \(1929\)](#), y ha sido ampliamente estudiado. Se ha demostrado que el problema single-source de Weber posee un único mínimo ([Palermo, 1961](#)) y que la localización óptima está contenida dentro de la envoltura convexa del conjunto de los nodos clientes ([Wendell y Hurter, 1973](#)). Para encontrar la solución hay diferentes métodos: un método iterativo *one-point* mejor conocido como el algoritmo de Weiszfeld ([Weiszfeld y Plastria, 2009](#)), un método unificado de planos cortantes ([Plastria, 1987](#)), un método dual ([Planchart y Hurter, 1975](#)), un algoritmo primal-dual que involucra normas mixtas ([Michelot y Lefebvre, 1987](#)), un algoritmo usando un sistema de coordenadas polares con origen cambiante para un set convexo con demanda uniforme ([Cavalier y Sherali, 1986](#)) con una extensión a un algoritmo multi-source, o un algoritmo de reducción potencial primal-dual con el problema formulado como un problema de programación convexa estándar en forma cónica ([Xue y Ye, 1997](#)). El algoritmo de Weiszfeld, el más popular de los algoritmos para resolver el problema de Weber, ha sido extensamente estudiado en la literatura como en [Kuhn \(1973\)](#), [Eckhardt \(1980\)](#), [Vardi y Zhang \(2001\)](#), [Brimberg \(1995\)](#), [Brimberg et al. \(1998\)](#), [Kuhn y Kuenne \(1962\)](#), ext. Una revisión exhaustiva del problema de Weber puede ser encontrada en [Drezner et al. \(2002\)](#).

El problema multi-source de Weber, o problema de localización-asignación, es un problema NP-completo ([Megiddo y Supowit, 1984](#)). Existen pocas heurísticas que lo resuelven a optimalidad y estas funcionan solamente en problemas pequeños. Como en [Cooper \(1972\)](#), donde es propuesto un método de solución exacto que está basado en una enumeración explícita de todas las asignaciones básicas factibles; En [Sherali et al. \(2002\)](#) se propone un algoritmo branch-and-bound basado en una partición del espacio de asignación, el cual converge finitamente a un óptimo global dentro de un porcentaje de tolerancia especificado; Y en [Chen et al. \(1998\)](#) es desarrollado un enfoque exacto usando una diferencia de funciones convexas y un constructo de programación concava. Para el enfoque heurístico donde el problema se resuelve a cuasi-óptimamente, existen mas publicaciones: En [Cooper \(1964\)](#) son explorados diferentes algoritmos para resolver el problema de localización-asignación con experimentos computacionales. Una heurística de branch-and-bound es propuesta en [Kuenne y Soland \(1972\)](#), describiendo sus límites superiores e inferiores. Una heurística alternante de localización-asignación es desarrollada en [Cooper \(1972\)](#). En [Murtagh y Niwatitsyawong \(1982\)](#) es sugerido que un procedimiento de búsqueda local tal como el código MINOS puede ser aplicado. [Avriel y Zang \(1980\)](#) propuso una técnica geométrica de programación basada en la estructura especial del problema multi-source de Weber. El método usado en [Bongartz et al. \(1994\)](#) relaja las restricciones binarias de la asignación, y resuelve para ambos localización y asignación simultáneamente. [Taillard \(2003\)](#) resuelve problemas de centroide muy grandes, como el problema multi-source de Weber, usando técnicas eficientes de clustering. Un enfoque primal-dual de continuación para el problema multi-source de Weber con capacidad basado en su reformulación no lineal de programación de conos de segundo orden se encuentra en [Chen et al. \(2011\)](#). El enfoque de utilizar los modelos discretos para resolver el problema de localización-asignación continuo es ampliamente aplicado ([Cooper, 1964](#)), ([Rosing, 1992](#)), ([Hansen et al., 1998](#)), ([Aras et al., 2007](#)), ([Brimberg et al., 2014](#)). Por esta razón, un estudio de los problemas de p-mediana con un foco en los procedimientos basados en reglas de metaheurísticas ([Mladenovic et al., 2007](#)) es conveniente. Para

una revisión del problema multi-source de Weber vea [Brimberg et al. \(2000\)](#) y [Brimberg et al. \(2008\)](#), y para una revisión en problemas representativos de localización vea [Brandeau y Chiu \(1989\)](#).

El problema multi-source ha sido estudiado considerando además una capacidad de las instalaciones y con un costo fijo de instalación. El problema con capacidad ha sido más estudiado, pero el problema multi-source de Weber con costos fijos tiene poco análisis, según nuestro conocimiento, existen cuatro papers sobre el tema. Primero los costos fijos son incluidos como un costo constante para todo el plano en [Brimberg et al. \(2004\)](#). Luego, en [Brimberg y Salhi \(2005\)](#), fue extendido a un costo fijo que depende de la zona a instalar, donde las zonas son polígonos convexos no sobrepuestos con un costo fijo constante en cada zona. En [Hosseini-zhad et al. \(2015\)](#) es desarrollada una metaheurística de entropía cruzada para un problema de localización continuo, con costos de instalación dependiente de la zona y del tipo de instalación a localizar. Y en [Luis et al. \(2015\)](#), es propuesto un problema multi-source de Weber con capacidad y con costos fijos dependientes de la zona a instalar usando regiones de segundo orden de Voronoi.

En general, el recopilamiento de los datos es costoso en terminos monetarios y en consumo de tiempo ([Helbich et al., 2013](#)). Por esta razón, hay una necesidad de estimar el valor de la tierra en los lugares no visitados, para realizar estas aproximaciones existen diferentes métodos, como adaptive weights smoothing ([Kolbe et al., 2015](#)), métodos hedónicos ([Larson et al., 2015](#)), o métodos geoestadísticos [Luo \(2004\)](#), [Cellmer et al. \(2014\)](#). En nuestro trabajo, utilizamos un método de interpolación de Kriging ([Oliver y Webster, 1990](#)). Este método de interpolación fue recomendado sobre otros métodos de interpolación en [Anselin y Le Gallo \(2006\)](#) y [Fernández-Avilés et al. \(2012\)](#), en un estudio de calidad del aire y de polución, respectivamente. Las posibilidades y limitaciones de los métodos geoestadísticos para aproximar el valor de la tierra es discutido en [Cellmer \(2014\)](#). Una comparación entre los diferentes métodos de Kriging para el mercado inmobiliario es discutido en [Kuntz y Helbich \(2014\)](#). Una extensión a espacio-tiempo en la interpolación de Kriging se puede encontrar en [Heuvelink y Griffith \(2010\)](#). La interpolación de Kriging es utilizada para encontrar el valor de la tierra en diferentes estudios en variadas ciudades del mundo [Martínez et al. \(2000\)](#), [Chica-Olmo et al. \(2007\)](#), [Montero-Lorenzo y Larraz-Iribas \(2012\)](#), [Cellmer et al. \(2012\)](#), [Liang y Yi \(2012\)](#), [Hu et al. \(2015\)](#), [Larraz y Población \(2013\)](#).

En resumen, existe poco estudio previo sobre los problemas single-source y multi-source de Weber que incluyan formulaciones con conos de segundo orden y, según nuestro conocimiento, solamente un paper contiene una propuesta de solución del problema. Existen pocos papers que incluyen el costo fijo, y los pocos que lo hacen, representan los costos fijos de una manera simplista que no refleja sus variaciones dentro de una zona urbana. A diferencia de ellos, nosotros hacemos una representación más realista de los costos de instalación, considerando diferentes posibles métodos de solución para el problema single-source de Weber con costos fijos.

2.2. Problema de Weber

El problema de Weber, o el problema single-source de Weber, consiste en encontrar las coordenadas (x, y) de una única instalación X que minimice la suma de las distancias hacia n clientes o puntos de demanda conocidos, con coordenadas (x_j, y_j) y pesos w_j , con $j = 1, 2, \dots, n$, donde todos los puntos se encuentran ubicados dentro de un mismo plano. Este es un problema de optimización no lineal sin restricciones, y su formulación general es:

$$\min z(X) = \sum_{j=1}^n w_j d(X, P_j),$$

donde $d(X, P_j)$ es la distancia entre la instalación y el punto de demanda j . Si es utilizada una distancia Euclidiana, esta distancia es $d(X, P_j) = \sqrt{(x - x_j)^2 + (y - y_j)^2}$.

Una aplicación de este problema es localizar un único centro de distribución (la instalación) que abastece diferentes cantidades (pesos), de tal manera que el costo total de transporte es minimizado, asumiendo que el costo depende de la distancia y la cantidad de producto transportado.

La solución de este problema es fácilmente computada usando calculo diferencial. Sin embargo, las coordenadas de la localización óptima de la instalación termina siendo una función dependiente de la distancia $d(X, P_j)$, que es desconocida. La solución práctica del problema, encontrada a través de Weiszfeld, involucra el uso de un algoritmo iterativo que toma como solución inicial el punto que minimiza la suma de los cuadrados de las distancias.

Una generalización natural del problema de Weber es la versión multi-source. Como se busca localizar más de una instalación, un problema de asignación debe ser resuelto conjuntamente con el problema de localización, lo que genera un problema mucho más complejo. En el caso más sencillo, se asume que cada demanda es asignada a la instalación más cercana.

2.3. Algoritmo de Weiszfeld

El promedio geométrico de un conjunto de puntos en un plano Euclidiano es el punto (no necesariamente incluido dentro del conjunto) que minimiza la suma de las distancias hacia los puntos dados. La solución para tres puntos fue dada por Evangelista Torricelli, luego de ser desafiado con dicho problema por Pierre de Fermat en el siglo 17. Un algoritmo para el problema más generalizado con una gran cantidad arbitraria de puntos, publicada por Weiszfeld en 1937 (Weiszfeld y Plastria, 2009), resuelve el problema numéricamente usando algoritmo hill climbing que repetidamente encuentra un punto que mejora la suma de las distancias hasta que no se puedan generar mayores mejoras. Cada paso de este algoritmo asigna pesos a los puntos, inversamente proporcional a la distancia de la solución actual, y luego encuentra el promedio ponderado de los puntos, el cual es el punto que minimiza la suma de los cuadrados de las distancias ponderadas.

El algoritmo ha sido frecuentemente redescubierto, y a pesar que existen otros métodos para encontrar el promedio geométrico, el algoritmo de Weiszfeld continua siendo ampliamente utilizado debido a su simplicidad y rápida convergencia.

2.3.1. Pseudocódigo del Algoritmo de Weiszfeld

Considerando el problema de Weber descrito anteriormente, el pseudocódigo del algoritmo de Weiszfeld es el siguiente:

1. (Inicialización)

$$x^0 = \frac{\sum_{j=1}^n w_j x d_j}{\sum_{j=1}^n w_j}$$

$$y^0 = \frac{\sum_{j=1}^n w_j y d_j}{\sum_{j=1}^n w_j}$$

y Costo óptimo $CO = \infty$

2. Calcular la distancia $d(X, P_j)^{(k)}$ entre los clientes y $(x^{(k)}, y^{(k)})$.
3. Calcular el costo:

$$Costo^{(k)} = \sum_{j=1}^n w_j d(X, P_j)^{(k)}$$

Si $CO > Costo^{(k)}$, entonces $(\bar{x}, \bar{y}) = (x^{(k)}, y^{(k)})$ y $CO = Costo^{(k)}$.

4. Actualizar $(x^{(k+1)}, y^{(k+1)})$:

$$x^{(k+1)} = \frac{\sum_{j=1}^n \frac{w_j x d_j}{d(X, P_j)^{(k)}}}{\sum_{j=1}^n \frac{w_j}{d(X, P_j)^{(k)}}}$$

$$y^0 = \frac{\sum_{j=1}^n \frac{w_j y_j}{d(X, P_j)^{(k)}}}{\sum_{j=1}^n \frac{w_j}{d(X, P_j)^{(k)}}}$$

Volver al paso 2

5. (Criterio de parada) Si $(x^{(k+1)}, y^{(k+1)}) - (x^{(k)}, y^{(k)}) \leq \epsilon$, con $\epsilon > 0$ o si se llega a la iteración 1000

2.4. Métodos de Interpolación de Kriging

En estadística, originalmente en geoestadística, Kriging o proceso de regresión Gaussiana es un método de interpolación para el cual los valores interpolados son modelados mediante un proceso Gaussiano regido por covarianzas anteriores. Bajo supuestos adecuados en lo anterior, Kriging entrega el mejor estimador lineal insesgado de los valores intermedios. Este método es ampliamente utilizado en las áreas de análisis espacial y de experimentos computacionales.

La idea básica detrás de Kriging es predecir el valor de una función en un punto dado por medio de la estimación de un promedio con pesos de los valores conocidos de dicha función en el vecindario del punto dado. El método está relacionado matemáticamente al análisis de regresión. Ambas teorías derivan un mejor estimador lineal insesgado, basando sus suposiciones en covarianzas, haciendo uso del teorema Gauss-Markov para probar independencia del estimador y del error.

Con el método de interpolación de Kriging se busca reconstruir una función $f : T \rightarrow \mathbb{R}$, donde T es un dominio perteneciente a $\mathbb{R}^d$, basándose en los valores de f en un conjunto finito de puntos de datos $X := \{x_1, \dots, x_n\} \subset T$.

Kriging se basa en la hipótesis de modelamiento de que f es una realización de una variable aleatoria Z , que es un conjunto $\{Z(x) : x \in T\}$ de variables aleatorias sobre el mismo espacio probabilístico (Ω, A, P) , indexado por T . Las observaciones $f(x_1), \dots, f(x_n)$ son entonces realizaciones de las variables aleatorias $Z(x_1), \dots, Z(x_n)$. Para predecir Z en algún punto $x_0 \in T$, se consideran todos los predictores lineales de la forma:

$$Z_u(x_0) = \sum_{i=1}^n u_i(x_0) f(x_i). \quad (2.1)$$

los cuales son a su vez variables aleatorias. El predictor de $f(x_0)$ dados $f(x_1), \dots, f(x_n)$ es entonces:

$$s_{f,X}(x_0) = \sum_{i=1}^n u_i(x_0) f(x_i). \quad (2.2)$$

Para determinar los pesos óptimos $u_1^*(x_0), \dots, u_n^*(x_0)$, se necesitan hipótesis estructurales adicionales, y dependiendo de estas hipótesis se distinguen diferentes tipos de Kriging: simple, ordinary, universal e intrinsic Kriging. Existen otras formas de Kriging, pero no serán discutidas aquí.

2.4.1. Simple Kriging

La hipótesis adicional con simple Kriging es que $Z(x)$ está centrada, i.e., $E(Z(x)) = 0$, y que el segundo momento existe para cada $x \in T$. Entonces la función de covarianza

$$K(x, y) := \text{Cov}(Z(x), Z(y)) \stackrel{Z \text{ centrado}}{=} E(Z(x)Z(y)), \quad x, y \in T \quad (2.3)$$

puede ser definida y es deducida de algunas propiedades básicas de la (co)varianza en que K es siempre simétrica y semi-definida positiva. En este contexto, K describe la estructura probabilística de Z , pero ciertas “propiedades determinísticas”, tal como suavizamiento de las realizaciones, son controladas a

su vez por K . Sin embargo, se tiene en cuenta que una completa caracterización de la estructura probabilística de un campo aleatorio requiere hipótesis adicionales sobre su distribución.

La noción geoestadística de optimalidad es considerar como el “mejor” predictor lineal $Z_{u^*}(x_0)$ la variable aleatoria con mínima desviación cuadrada esperada para $Z(x_0)$, i.e.

$$E((Z(x_0) - Z_{u^*}(x_0))^2) \leq E((Z(x_0) - Z_u(x_0))^2)$$

para todo $Z_u(x_0)$ de la forma (2.1).

Los pesos óptimos son obtenidos minimizando

$$E((Z(x_0) - Z_u(x_0))^2) = \underbrace{E(Z(x_0)Z(x_0))}_{=K(x_0, x_0)} - 2 \sum_{i=1}^n u_i(x_0) \underbrace{E(Z(x_0)Z(x_i))}_{=K(x_0, x_i)} + \sum_{i=1}^n \sum_{j=1}^n u_i(x_0) u_j(x_0) \underbrace{E(Z(x_i)Z(x_j))}_{=K(x_i, x_j)}. \quad (2.4)$$

$$Q(u(x_0)) = K(x_0, x_0) - 2 \sum_{i=1}^n u_i(x_0) K(x_0, x_i) + \sum_{i=1}^n \sum_{j=1}^n u_i(x_0) u_j(x_0) K(x_i, x_j) \quad (2.5)$$

y sigue que los coeficientes óptimos $u_1^*(x_0), \dots, u_n^*(x_n)$ que minimizan Q deben satisfacer:

$$\sum_{j=1}^n u_j^*(x_0) K(x_i, x_j) = K(x_0, x_i), \quad i = 1, \dots, n. \quad (2.6)$$

Cabe mencionar brevemente otra perspectiva que es, en cierta manera, aún más probabilística: un enfoque Bayesiano. En el marco del simple Kriging, es esencialmente la hipótesis adicional que debe hacerse es de que Z es Gaussiana. Como es mencionado anteriormente, una vez que K es fijado la distribución del espacio aleatorio Gaussiano está completamente determinado, y el enfoque Bayesiano lo utiliza como una *infinite-dimensional prior distribution* de la función desconocida f . La posterior distribución de f en x_0 dada las observaciones $f(x_1), \dots, f(x_n)$ es entonces una distribución Gaussiana con promedio $s_{f,x}(x_0)$ (con pesos $u_1^*(x_0), \dots, u_n^*(x_0)$ descritas anteriormente) y varianza $Q(u^*(x_0))$. A diferencia del simple Kriging, este resultado no está basado en una función de pérdida particular. Es una consecuencia inmediata de la completa especificación de una distribución previa para f .

2.4.2. Ordinary y Universal Kriging

La hipótesis de que Z es centrada, especialmente, parece inapropiada en la mayoría de las aplicaciones de geoestadística. Una primera generalización del simple Kriging es entonces permitir una función no-cero del promedio $m(x) := E(Z(x))$ mientras se mantiene la hipótesis de que Z tiene segundos momentos. La función del promedio es usualmente desconocida en la práctica, pero este problema puede ser evitado mediante el requerimiento de que los predictores potenciales Z_u de Z sean insesgados, i.e.

$$E(Z(x)) = E(Z_u(x)), \quad \forall x \in T. \quad (2.7)$$

Esto tiene la ventaja adicional de sistemáticamente prevenir la sobre-estimación o sub-estimación de $Z(x_0)$. Note que cualquier predictor así es automáticamente insesgado si Z es centrado. Usando ésta restricción sobre el insesgamiento para recalcular la función objetivo (2.4) se obtiene:

$$E((Z(x_0) - Z_u(x_0))^2) = E((Z(x_0) - E(Z(x_0)) + E(Z_u(x_0)) - Z_u(x_0))^2) \quad (2.8)$$

$$= E((Z(x_0) - E(Z(x_0)) - \sum_{i=1}^n u_i(x_0)(Z(x_i) - E(Z(x_i))))^2) \quad (2.9)$$

$$= \underbrace{Cov(Z(x_0), Z(x_0))}_{=K(x_0, x_0)} - 2 \sum_{i=1}^n u_i(x_0) \underbrace{Cov(Z(x_0), Z(x_i))}_{=K(x_0, x_i)} + \sum_{i=1}^n \sum_{j=1}^n u_i(x_0) u_j(x_0) \underbrace{Cov(Z(x_i), Z(x_j))}_{=K(x_i, x_j)}, \quad (2.10)$$

donde tiene la forma Q en (2.5), dependiendo únicamente en K y no en m . Sin embargo, su minimizador es, en general, no el mismo que anteriormente debido a la restricción adicional sobre insesgamiento

$$m(x) = \sum_{i=1}^n u_i(x) m(x_i), \quad \forall x \in T, \quad (2.11)$$

que restringe la elección de los pesos. Para asegurar que la condición (2.11) pueda ser satisfecha, no se puede dejar ser a la función del promedio completamente general, pero se debe suponer una suficiente simplicidad, un modelo dimensional finito. El modelo más simple es una constante (pero desconocida) como la función del promedio, y esta suposición conlleva a lo que es conocido como ordinary Kriging. Esto es, sin embargo, solamente un caso especial del universal Kriging donde la función del modelo es modelada por:

$$m(x) := \sum_{k=1}^q \beta_k p_k(x), \quad x \in T \quad (2.12)$$

con funciones $p_1, \dots, p_q$ lineales independientes conocidas y coeficientes $\beta_1, \dots, \beta_q$ desconocidos. Una función del promedio así es también llamada *tendencia*, y la condición (2.11) se convierte en:

$$\sum_{k=1}^q \beta_k p_k(x) = \sum_{k=1}^q \beta_k \sum_{i=1}^n u_i(x) p_k(x_i), \quad \forall x \in T.$$

Esta condición debe cumplirse para cualquier conjunto de coeficientes $\beta_1, \dots, \beta_q$, y así cuando se predice en x_0 se está de vuelta con la condición

$$p_k(x_0) = \sum_{i=1}^n u_i(x_0) p_k(x_i), \quad k = 1, \dots, q. \quad (2.13)$$

restringiendo los pesos $u_1(x_0), \dots, u_n(x_0)$. La representación

$$s_{f,X} = \sum_{j=1}^n \alpha_j K(\cdot, x_j) + \sum_{k=1}^q \beta_k p_k. \quad (2.14)$$

ha sido ya notada por [Matheron \(1971\)](#), y su sistema de ecuaciones correspondiente es conocido como *dual Kriging*.

El interpolante del universal Kriging puede ser, también, derivado dentro del marco Bayesiano. Como una distribución previa para f se asume un campo aleatorio Gaussiano con una función de covarianza K y función del promedio m como en (2.12). Para una completa especificación de la función previa para f , es necesario hacer, a su vez, una suposición de la distribución para $\beta_1, \dots, \beta_q$. Después, la distribución posterior de f puede ser resuelta, pero dependerá tanto de K como de la distribución previa de los coeficientes de tendencia. En el caso especial de una distribución plana poco informativa, sin embargo, se ha demostrado que la distribución posterior de f en x_0 es una distribución Gaussiana con promedio $s_{f,X}(x_0)$ y varianza $Q(u^*(x_0))$, ambos calculados con los pesos óptimos del método universal Kriging.

2.4.3. Intrinsic Kriging

La idea con los campos aleatorios intrínsecos es que uno tiene que especificar completamente la estructura de segundo orden de Z , pero sólo la estructura de dependencia de ciertos incrementos. Más específicamente, sea $\mathcal{P}$ un espacio de dimensión finita con funciones en T , y sea $L_{\mathcal{P}}(T)$ el espacio de funcionales de la forma $\lambda_X = \sum_{i=1}^n a_i \delta_{x_i}$ con $\lambda_X(p) = 0$ para todo $p \in \mathcal{P}$. Para cada $\lambda_X \in L_{\mathcal{P}}(T)$ que

$$Z_{\lambda,X} := \lambda_X(Z) = \sum_{i=1}^n a_i Z(x_i)$$

es llamado una *combinación lineal permitida de Z con respecto a $\mathcal{P}$* . Se asume que toda combinación lineal permitida de Z tiene segundos momentos y son centrados, i.e. $E(Z_{\lambda,X}) = 0$. La función $K : T \rightarrow \mathbb{R}$ es entonces llamada una *función de covarianzas generalizada* de Z si

$$E(Z_{\lambda,X} Z_{\lambda',X'}) = \sum_{i=1}^n \sum_{j=1}^{n'} a_i a'_j K(x_i, x'_j) \quad \forall \lambda_X \in L_{\mathcal{P}}(T).$$

Es posible notar que Z y $Z + p$ tienen la misma función de covarianza generalizada para cualquier $p \in \mathcal{P}$. Además, desde la expectativa de que cualquier variable cuadrada aleatoria es no-negativa, K debe ser condicionada semi-definida positiva con respecto a $\mathcal{P}$.

El caso más importante en la practica es el caso donde $\mathcal{P} = \pi_m(\mathbb{R}^d)$, el espacio de polinomios de orden de a lo más m , y donde Z es *intrinsically (weakly) stationary de orden m* , i.e. $K(x, y) = \Phi(y - x)$ para alguna función $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ que es condicionalmente semi-definida positiva con respecto a $\pi_m(\mathbb{R}^d)$. Asumir que K es invariante ante traslaciones es frecuentemente razonable debido a que estructuras de dependencia generales son usualmente demasiado complejas para una inferencia confiable. Note que $\lambda_X \in L_{\pi_m}(\mathbb{R}^d)$ implica que $\lambda_{X+x} \in L_{\pi_m}(\mathbb{R}^d)$ para todo $x \in \mathbb{R}^d$, debiendo ser una formula binomial, y entonces todas las variables aleatorias

$$Z_{\lambda, X+x} = \sum_{i=1}^n a_i Z(x_i + x), \quad x \in \mathbb{R}^d$$

son permitidas solamente si $\lambda_X \in L_{\pi_m}(\mathbb{R}^d)$. Sigue que el campo aleatorio $\{Z_{\lambda, X+x} : x \in \mathbb{R}^d\}$ con incrementos del orden m -ésimo es *weakly stationary*, i.e. centrado con segundos momentos y función de covarianza invariante a la traslación, con

$$E(Z_{\lambda, X+x} Z_{\lambda', X'+x}) = \sum_{i=1}^n \sum_{j=1}^{n'} a_i a'_j \Phi(y - x + x_j - x_i) =: \Phi_{\lambda, \lambda'}(y - x).$$

Cuando en la práctica se enfrenta a una situación donde el mismo Z no aparenta ser weakly stationary, existe aún una posibilidad de que esto sea plausible para algún incremento de orden superior, y entonces esto motiva a un modelamiento con campos aleatorios intrínsecamente estacionarios.

Para predecir $Z(x_0)$ se considera el error funcional $\lambda_{err} = \delta_{x_0} - \lambda_{u(x_0)}$, pero en el marco estocástico se interesa en el error esperado cuadrado de la predicción. Cuando K es la función generalizada de la covarianza, se tiene por definición

$$E((\lambda_{err}(Z))^2) = K(x_0, x_0) - 2 \sum_{i=1}^n u_i(x_0) K(x_0, x_i) + \sum_{i=1}^n \sum_{j=1}^n u_i(x_0) u_j(x_0) K(x_i, x_j),$$

el cual es nuevamente de la forma cuadrática Q en (2.5). El requerimiento que $\lambda_{err} \in L_{\mathcal{P}}(T)$ nuevamente implica la condición (2.13).

La diferencia con el universal Kriging se encuentra en la interpretación de la representación (2.14). Mientras es legítimo en universal Kriging el interpretar el segundo término en (2.14) como una función del promedio aproximada y el primer término como una desviación aproximada de este, tal interpretación estaría siendo errónea para el intrinsic Kriging donde la función del promedio no está siquiera definida.

2.5. Triangulación de Delaunay

En matemática y en geometría computacional, una triangulación de Delaunay para un conjunto de puntos en un plano es una triangulación tal que no hay un punto del conjunto en el interior de la circunferencia de ningún triángulo. La triangulación de Delaunay maximiza el mínimo ángulo de todos los ángulos de los triángulos dentro de la triangulación. La triangulación obtiene su nombre por Boris Delaunay y su trabajo en este tema desde 1934.

Para un conjunto de puntos en la misma línea, no existe una triangulación de Delaunay, dado que la noción de triangulación está degenerada en ese caso. Para cuatro o más puntos en el mismo círculo, la triangulación de Delaunay no es única: cada una de las dos posibilidades de dividir el cuadrilátero satisface la "condición de Delaunay", i.e., el requerimiento de que los círculos circunscritos de todos los triángulos deben tener un interior vacío.

Si se consideran esferas circunscritas, la noción de la triangulación de Delaunay puede extenderse a tres o más dimensiones. Generalizaciones son posibles para métricas diferentes a la Euclidiana. Sin embargo, en estos casos la triangulación de Delaunay no puede garantizarse de ser única o que exista.

2.5.1. Relación de la triangulación de Delaunay con el diagrama de Voronoi

La triangulación de Delaunay de un conjunto discreto de puntos en una posición general corresponde al grafo dual del diagrama de Voronoi para el mismo conjunto de puntos. Casos especiales incluyen la existencia de tres puntos en una línea y de cuatro puntos dentro de un círculo.

2.5.2. Propiedades

Sea n el número de puntos y d el número de dimensiones.

1. La unión de todos los simplex en la triangulación es una envoltura convexa de los puntos.
2. La triangulación de Delaunay contiene $O(n^{\lceil d/2 \rceil})$ simplex.
3. En el plano, $d=2$, si hay b vértices en la envoltura convexa, entonces cualquier triangulación de los puntos tiene a lo mas $2n - 2 - b$ triángulos, más una cara exterior.
4. En el plano, cada vértice tiene en promedio seis triángulos a su alrededor.
5. En el plano, la triangulación de Delaunay maximiza el ángulo mínimo. Comparado con cualquier otra triangulación de los puntos, el menor ángulo en la triangulación de Delaunay es al menos tan grande como el menor ángulo en cualquier otra. Sin embargo, La triangulación de Delaunay no minimiza necesariamente el máximo ángulo. Además, la triangulación de Delaunay no necesariamente minimiza el largo de las aristas.
6. Un círculo circunscribiendo cualquier triángulo de Delaunay no contiene ningún otro punto en su interior.
7. Si un círculo que atraviesa dos puntos no contiene ningún otro punto en su interior, entonces es el segmento que conecta estos dos puntos es una arista de la triangulación de Delaunay para los puntos dados.
8. Cada triángulo de la triangulación de Delaunay para un conjunto de puntos en un espacio d -dimensional corresponde a la faceta de la envoltura convexa de la proyección de los puntos en una paraboloides $(d + 1)$ -dimensional, y vice versa.

2.5.3. Construcción del Algoritmo

Hay varios algoritmos que sirven para crear una triangulación de Delaunay a partir de un conjunto de puntos. En todos los algoritmos hay que inspeccionar si un vértice está dentro de una circunferencia circunscrita o no, así que este test tiene que ser muy eficiente. Por supuesto es posible computar el circuncentro y la circunferencia circunscrita y, después, examinar si el vértice está dentro del círculo, pero hay un test más simple y eficiente que usa el determinante de una matriz.

Considerándose dos dimensiones: Si los tres puntos A, B y C forman un triángulo (con los puntos denominados en sentido contrario al de las agujas del reloj), el punto D está dentro de su circunferencia circunscrita si:

$$\begin{bmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{bmatrix} = \begin{bmatrix} A_x - D_x & A_y - D_y & (A_x - D_x)^2 + (A_y - D_y)^2 \\ B_x - D_x & B_y - D_y & (B_x - D_x)^2 + (B_y - D_y)^2 \\ C_x - D_x & C_y - D_y & (C_x - D_x)^2 + (C_y - D_y)^2 \end{bmatrix} > 0$$

Es decir, si el determinante de esta matriz es mayor que 0. En este caso es suficiente conocer el signo aritmético, así que este cómputo puede ser acelerado fácilmente.

Incremental construction

El algoritmo *Incremental construction* realiza la siguiente idea: añadir un vértice a una triangulación de Delaunay y corregir la red hasta que todos los triángulos cumplan de nuevo la condición de Delaunay.

Hay varias posibilidades para seleccionar el vértice siguiente, incidentalmente, ordenado por una coordenada o usando un árbol. La selección del vértice siguiente tiene una gran influencia en el tiempo de ejecución del algoritmo.

Divide and conquer

Este algoritmo usa el principio conocido como *divide and conquer*: dividir el conjunto de puntos en dos partes de igual tamaño, calcular la triangulación de Delaunay para cada parte individualmente y después reunir las dos triangulaciones corrigiendo los errores.

La idea de usar el principio *divide and conquer* para computar un diagrama de Voronoi en dos dimensiones fue introducida en 1975. La idea fue revisada en 1980 para computar la triangulación de Delaunay y fue mejorada por Guibas y Stolfi in 1985. En 1986 Dwyer presentó una modificación que mejoró la cota ajustada asintótica y en 1992 Leach presentó otra aceleración del método. En 1997 Cigoni, Montani y Scopigno presentaron el algoritmo DeWall, que hace posible el cálculo de una triangulación de Delaunay usando divide and conquer para cualquier número de dimensiones.

Usando el algoritmo más avanzado, ese método resulta en una cota superior asintótica de $O(n \log(n))$ y una cota ajustada asintótica de $O(n \log(\log(n)))$; en algunos casos especiales es posible reducir la cota ajustada a la cota superior.

Sweepline

El algoritmo *sweepline* se basa en un principio similar a la construcción incremental: construir una pequeña parte de la triangulación final y después seguir añadiendo vértices hasta que la triangulación esté completa. La diferencia estriba en que no hay que corregir ningún de los errores que pudieran presentarse.



3 | Formulación del Modelo

3.1. Problema Single-Source de Weber con Función de Costos Fijos Continuos por Partes

El problema single-source de Weber con costos fijos generalizado considera la localización de una única instalación con coordenadas $(\bar{x}, \bar{y}) \in \mathbb{R}^2$. Esta instalación debe abastecer a un conjunto J de clientes. Se asume que las coordenadas de los clientes son conocidas. Sea $f(\bar{x}, \bar{y})$ la función que representa los costos fijos incurridos cuando se instala el emplazamiento en $(\bar{x}, \bar{y})$. El problema consiste en determinar la localización óptima para una única instalación tal que el costo de transporte y el costo fijo sean minimizados. El problema single-source de Weber con costos fijos generalizado puede ser expresado de la siguiente forma:

$$\min_{(\bar{x}, \bar{y})} \sum_{j \in J} w_j \|(\bar{x}, \bar{y}) - (x_j, y_j)\|_2 + f(\bar{x}, \bar{y}) \quad (3.1)$$

$$\text{s.t. } (\bar{x}, \bar{y}) \in \mathbb{R}^2 \quad (3.2)$$

Se considera un conjunto conveniente I de nodos donde se conoce la información sobre sus costos fijos, $\mathbf{C}$, y sus coordenadas, $(\mathbf{x}_i, \mathbf{y}_i)$. En este trabajo, la forma de abordar los costos fijos es mediante la aplicación de un método de interpolación de Kriging y, así, definir una función continua de los costos en cada punto de la envoltura convexa de I . Esta función de costos no es simple y puede no ser convexa. Para hacer que la función de costos continua sea manejable, se aproxima la interpolación de Kriging con una función por partes que sea convexa en cada parte. Para esto, se particiona la envoltura convexa de I a través de una malla poliedral y se define una función de costos continua por partes a través de una combinación convexa de los vértices de la malla. Es sabido que es mejor usar subconjuntos con el menor número de nodos de información posible que tengan su interior vacío para crear los poliedros, i.e. usando la triangulación de Delaunay.

Aplicando la triangulación de Delaunay sobre el conjunto I se obtiene un conjunto K de triángulos, donde cada triángulo k -ésimo será denotado como P_k , con $P_k = \{(x, y) \in \mathbb{R}^2 | (x, y) = (\mathbf{x}\mathbf{k}_k, \mathbf{y}\mathbf{k}_k)^T \cdot \boldsymbol{\lambda}^k, \mathbf{1}^T \boldsymbol{\lambda}^k = 1, \forall \lambda^k \geq 0\}$. Se tiene $\boldsymbol{\lambda}^k \in \mathbb{R}^3$ como el vector de la combinación convexa de los vértices del triángulo $k \in K$. El conjunto de todas las posibles localizaciones de la instalación, $\bigcup_{k \in K} P_k$, puede no ser convexo si las áreas que no interesa instalar, como una laguna o una zona estrictamente residencial, son excluidas de la triangulación.

Dado lo anterior, la localización de la instalación puede ser expresada como $(\bar{x}, \bar{y}) = \sum_{k \in K} (\mathbf{x}\mathbf{k}_k, \mathbf{y}\mathbf{k}_k)^T \boldsymbol{\lambda}^k$ y su costo fijo como la combinación convexa de los costos fijos de los vértices de los triángulos de Delaunay, $\sum_{k \in K} \mathbf{C}\mathbf{k}_k^T \boldsymbol{\lambda}^k$. Considerando una variable binaria, Z_k , tal que se fuerce que la instalación puede estar incluida solamente dentro de un triángulo, siendo 1 si es instalada en el k -ésimo triángulo o 0 si es instalada en otro triángulo.

Se puede formular el problema single-source de Weber con costo fijo continuo por partes de la siguiente forma:

Problema (P0):

$$\min_{\mathbf{Z}, \lambda} \sum_{k \in K} (Z_k \sum_{j \in J} w_j \|(\mathbf{xk}_k, \mathbf{yk}_k)^T \lambda^k - (xc_j, yc_j)\|_2 + \mathbf{CK}_k^T \lambda^k) \quad (3.3)$$

$$\text{s.t. } \mathbb{1}^T \lambda^k = Z_k, \quad \forall k \in K \quad (3.4)$$

$$\mathbb{1}^T \mathbf{Z} = 1 \quad (3.5)$$

$$\lambda_i^k \geq 0, \quad \forall i \in \{1, 2, 3\}, k \in K \quad (3.6)$$

$$Z_k \in \{0, 1\}, \quad \forall k \in K \quad (3.7)$$

A continuación, se presentarán diferentes formas de resolver el problema (P0).

4 | Propuesta de Solución

Se considerarán tres propuestas de solución diferentes para resolver el problema ($P0$). Para la primera propuesta, se usará una formulación monolítica utilizando conos de ($P0$). La segunda propuesta considera una descomposición del problema ($P0$) a través de fijar la variable $\mathbf{Z}$ para luego resolver el sub-problema generado, y se evalúan todos los posibles valores que puede tomar $\mathbf{Z}$. La última propuesta considera una reformulación cónica de los sub-problemas generados en la propuesta de solución anterior.

4.1. Logarithmic Disaggregated Convex Combination Model con Conos

El problema ($P0$) es reformulado siguiendo las siguientes dos etapas. Primero, se formula el problema como un problema no-lineal con conos de segundo orden. Luego, usando el modelo *logarithmic disaggregated convex combination model* (Vielma et al., 2010), se puede resolver eficientemente la función de costos continua por partes.

En lo siguiente, se reformula el problema ($P0$) como un problema no-lineal con conos de segundo orden para poder eliminar los términos con raíces cuadradas. Primero se introduce un conjunto de variables continuas no negativas, $\mathbf{d}$, para representar los términos de las raíces cuadradas en:

$$d_j = \|(\mathbf{x}\mathbf{k}_k, \mathbf{y}\mathbf{k}_k)^T \lambda^k - (xc_j, yc_j)\|_2, \quad \forall j \in J \quad (4.1)$$

$$d_j \geq 0, \quad \forall j \in J \quad (4.2)$$

Por simplicidad, se agregan dos conjuntos extras de variables auxiliares, $\mathbf{v}$ y $\mathbf{r}$, para transformar (4.1) en:

$$d_j^2 = z_j^2 + w_j^2, \quad \forall j \in J \quad (4.3)$$

$$v_j = \mathbf{x}\mathbf{k}_k^T \lambda^k - xc_j, \quad \forall j \in J \quad (4.4)$$

$$r_j = \mathbf{y}\mathbf{k}_k^T \lambda^k - yc_j, \quad \forall j \in J \quad (4.5)$$

Debido a que las variables no negativas $\mathbf{d}$ son introducidas en la función objetivo del problema ($P0$) con coeficientes positivos y este problema es un problema de minimización, la ecuación puede ser más lejos relajada como las siguientes desigualdades:

$$d_j^2 \geq v_j^2 + r_j^2, \quad \forall j \in J \quad (4.6)$$

Se puede notar que las restricciones (4.2) y (4.6) definen las restricciones de un cono de segundo orden. El problema ($P0$) puede ser expresado como el siguiente problema cónico:

Problema (CP0):

$$\begin{aligned} \min_{\mathbf{Z}, \lambda, \mathbf{d}, \mathbf{v}, \mathbf{r}} \quad & \sum_{k \in K} (Z_k \sum_{j \in J} w_j d_j + \mathbf{C}\mathbf{k}_k^T \lambda^k) \\ \text{s.t.} \quad & (3,4), (3,5), (3,6), (3,7), (4,2), (4,4), (4,5), (4,6) \end{aligned} \quad (4.7)$$

El modelo *logarithmic disaggregated convex combination model* consiste en reemplazar la función por partes $f(\bar{x}, \bar{y})$ por su epígrafe $epi(f)$ y ajustar las coordenadas $(\bar{x}, \bar{y})$ para ser restringidas por uno y sólo uno de los dominios de f . Para una minimización, resolver la función f es equivalente a resolver su epígrafe $epi(f)$. Para construir un modelo con el menor número de variables binarias y de restricciones posible, se identifica cada triángulo a través de un vector binario que pertenezca a $\{0, 1\}^{\lceil \log_2 |K| \rceil}$ a través de una función biyectiva $B : K \rightarrow \{0, 1\}^{\lceil \log_2 |K| \rceil}$ y se genera una cantidad de $\lceil \log_2 |K| \rceil$ variables binarias, $\mathbf{m} \in \{0, 1\}^{\lceil \log_2 |K| \rceil}$, para asegurar que las coordenadas estén dentro de un único triángulo. Sea Q el epígrafe $epi(f)$. La variable λ utilizada en esta formulación es la misma variable que fue utilizada para definir P_k , λ^k , con la única diferencia de que en esta formulación se utiliza de una manera agregada.

Usando el modelo *logarithmic disaggregated convex combination model* y una formulación de conos de segundo orden para reformular el problema (P0), se puede expresar el problema single-source de Weber con costos fijos continuos por partes de la siguiente forma:

Problema (DlogCP0):

$$\min_{\lambda, \mathbf{m}, \mathbf{d}, Q, \mathbf{v}, \mathbf{r}} \sum_{j \in J} w_j d_j + Q \quad (4.8)$$

$$\text{s.t. } tr(\mathbf{CK}^T \lambda) \leq Q \quad (4.9)$$

$$\sum_{k \in K} \sum_{l=1}^3 \lambda_l^k = 1 \quad (4.10)$$

$$\sum_{k \in K^+(B, t)} \sum_{l=1}^3 \lambda_l^k \leq m_t, \quad \forall t \in T(K) \quad (4.11)$$

$$\sum_{k \in K^0(B, t)} \sum_{l=1}^3 \lambda_l^k \leq (1 - m_t), \quad \forall t \in T(K) \quad (4.12)$$

$$\lambda_l^k \geq 0 \quad \forall l \in \{1, 2, 3\}, k \in K \quad (4.13)$$

$$m_t \in \{0, 1\} \quad \forall t \in T(K) \quad (4.14)$$

$$(4.2), (4.4), (4.5), (4.6)$$

donde $B : K \rightarrow \{0, 1\}^{\lceil \log_2 |K| \rceil}$ es una función inyectiva cualquiera, y donde $K^+(B, t) = \{k \in K : B(k)_t = 1\}$, $K^0(B, t) = \{k \in K : B(k)_t = 0\}$ y $T(K) = \{1, \dots, \lceil \log_2 |K| \rceil\}$. Este problema es un problema mixed integer conic quadratic non linear con una función objetivo lineal y que puede ser resuelto mediante solvers como GUROBI, CPLEX o MOSEK.

4.2. Método de Descomposición

Considerando el problema (P0), se puede observar que las variables λ y $\mathbf{Z}$ están relacionadas mediante una única restricción. Si se fija la variable $\mathbf{Z}$, el problema se vuelve separable en $|K|$ sub-problemas donde se fuerza que la localización esté dentro del triángulo de Delaunay k -ésimo, i.e. forzando que $Z_k = 1$ y $Z_{k'} = 0$, para todo triángulo $k' \in K \setminus k$. Entonces el k -ésimo sub-problema puede ser escrito de la siguiente forma:

Sub-Problema (SP0(k)):

$$\min_{\lambda^k} \sum_{j \in J} w_j \|(\mathbf{xk}_k, \mathbf{yk}_k)^T \lambda^k - (x_{c_j}, y_{c_j})\|_2 + \mathbf{CK}_k^T \lambda^k \quad (4.15)$$

$$\text{s.t. } \mathbf{1}^T \lambda^k = 1 \quad (4.16)$$

$$\lambda_i^k \geq 0, \quad \forall i \in \{1, 2, 3\} \quad (4.17)$$

Este sub-problema (SP0(k)) es un problema convexo no lineal con restricciones lineales y que puede ser resuelto eficientemente por solvers como MINOS o IPOPT.

Sea λ^{k*} la solución óptima del problema $(SP0(k))$; Sea $FO_{(SP0(k))}(\lambda^{k*})$ el costo óptimo de la función objetivo del problema $(SP0(k))$, y sea $(\bar{\lambda}, \bar{\mathbf{Z}})$ la solución óptima para el problema $(P0)$. La solución óptima del problema $(P0)$ es la mejor solución considerando todos los sub-problemas $(SP0(k))$, i.e. $\bar{\lambda} = \lambda^{k^*}$, donde $k^* = \operatorname{argmin}_{k \in K} \{FO_{(SP0(k))}(\lambda^{k*})\}$. Para $\bar{\mathbf{Z}}$, el valor de $\bar{Z}_k = 1$ para $k = k^*$ y $\bar{Z}_k = 0$ para cualquier otro k .

4.3. Método de Descomposición Cónica

El término de la norma Euclidiana en la función objetivo del sub-problema $(SP0(k))$ puede dar origen a dificultades en el procedimiento de optimización. Siguiendo la lógica expuesta para la primera propuesta de solución, se reformula el sub-problema $(SP0(k))$ como un problema no lineal con conos de segundo orden, dejando el siguiente problema cónico:

Sub-Problema $(SCP0(k))$:

$$\begin{aligned} \min_{\lambda^k, \mathbf{d}, \mathbf{v}, \mathbf{r}} \quad & \sum_{j \in J} w_j d_j + \mathbf{C} \mathbf{K}_k^T \lambda^k \\ \text{s.t.} \quad & (4,16), (4,17), (4,2), (4,4), (4,5), (4,6) \end{aligned} \quad (4.18)$$

Puede demostrarse trivialmente que el sub-problema $(SCP0(k))$ es equivalente al sub-problema $(SP0(k))$, pero se tiene ahora restricciones cónicas no-lineales con una función objetivo lineal bastante más sencilla. La solución óptima del problema $(P0)$ es la mejor solución considerando todos los sub-problemas $(SCP0(k))$, equivalentemente a la propuesta de solución anterior.

La ventaja de una formulación de conos de segundo orden es que puede ser resuelta directamente usando paquetes de software estándares de optimización tales como CPLEX, GUROBI o MOSEK.



5 | Estudios Computacionales

En este capítulo se presentan los estudios computacionales y sus resultados. Los objetivos principales de estos estudios computacionales es mostrar cual de las propuestas de solución presentadas tiene la mejor performance en términos de tiempo CPU, ilustrar la aplicabilidad industrial del problema single-source de Weber con costos fijos y comparar el problema single-source de Weber con costo fijo con un problema UFLP. Además, se busca caracterizar diferentes propuestas de solución para el problema single-source de Weber sin considerar los costos fijos, las cuales serán descritas a continuación. Para determinar cual propuesta tiene la mejor performance, se llevan a cabo 400 instancias que se denotan *conjunto de prueba*, y para mostrar la aplicabilidad en la industria se considera un ejemplo ilustrativo de una compañía que manufactura productos derivados de las frutas y verduras, encontrado en [Escalona et al. \(2015\)](#). Junto con lo anterior, se corrobora la instalación de una única instalación en cada instancia con el UFLP.

Todos los problemas single-source de Weber con y sin costos fijos, desarrollados con cada una de las diferentes propuestas de solución, fueron programados usando AMPL. Para resolver el método de descomposición se utiliza el solver MINOS. El problema (*DlogCP0*) y el método de descomposición cónica se resuelven a través del solver CPLEX. La resolución del problema de Weber sin costos fijos reformulado cónicamente fue llevada a cabo con el solver GUROBI. El método de interpolación de Kriging, la triangulación de Delaunay y el algoritmo de Weiszfeld fueron programados utilizando MATLAB. El *conjunto de prueba* fue llevado a cabo en un PC con AMD FX 4,00 GHz processor y 12 GB RAM. El ejemplo ilustrativo, el problema de UFLP y los problemas single-source de Weber sin considerar los costos fijos fueron llevados a cabo en un PC con Intel i3 2,10 Ghz y 4 GB RAM.

5.1. Problema Single-Source de Weber sin Costos Fijos

En esta sección, se resuelve el problema single-source de Weber sin considerar los costos de instalación. Este problema es descrito en el capítulo 2 y se plantean dos métodos para solucionarlo a optimalidad: (i) el algoritmo de Weiszfeld, descrito en el capítulo 2, y (ii) una reformulación del problema en un problema cónico, descrito en el anexo D.

Para comparar los dos métodos y poder determinar cual de los dos tiene un mejor desempeño computacional, se utilizar la base de datos de 49, 88 y 250 nodos, descrita en [Daskin \(2011\)](#).

Tabla 5.1: Resultados del Problema de Weber sin costos fijos

Algoritmo		Problema Cónico
N	T[s]	T[s]
49	0.002137	0.172
88	0.002843	0.234
250	0.00502	1.062

Los resultados se muestran en la tabla 5.1. Se puede observar en ella que el algoritmo de Weiszfeld tiene una mejor performance, en base al tiempo CPU, sobre el problema cónico. El speedup promedio

del algoritmo sobre el problema cónico es de $124,7x$. Por esta razón, de poder utilizarse, es recomendable resolver el problema single-source de Weber con el algoritmo de Weiszfeld.

5.2. Conjunto de Prueba

Para comparar las diferentes propuestas y poder determinar cual de las tres propuestas de solución tiene la mejor performance en términos del tiempo CPU, se generan 100 experimentos. En cada experimento se fija el número de nodos clientes y se generan 4 refinamientos a la malla generada por la triangulación de Delaunay. Es decir, se tienen 400 instancias. Por simplicidad, se considera que los pesos de la demanda esperada por el ratio de transporte para todos los clientes es el mismo, i.e. que $w_j = 1$, para cualquier $j \in J$.

Cada experimento tiene un mismo conjunto inicial de 100 nodos de información, generados aleatoriamente. Entonces, para una mejor aproximación convexa por partes de la función continua de los costos fijos, es decir, de la interpolación de Kriging, se propone el siguiente refinamiento de la malla. Se considera como el primer refinamiento la triangulación de Delaunay del conjunto inicial de nodos de información, mostrado en la figura 5.1. El segundo refinamiento es generado a través de la creación de nodos de información adicionales donde su localización es en el centro de cada arista de todos los triángulos y su costo de instalación es determinado evaluando el punto en la función continua de la interpolación de Kriging de los costos fijos. Luego, es generada una nueva triangulación de Delaunay sobre el conjunto I más el conjunto de nodos de información adicionales. Se muestra el segundo refinamiento en la figura 5.2. El tercer y cuarto refinamiento son aplicados sobre la segunda y tercera triangulación, respectivamente. En las figuras 5.3 y 5.4 se muestran el tercer y cuarto refinamiento, respectivamente.

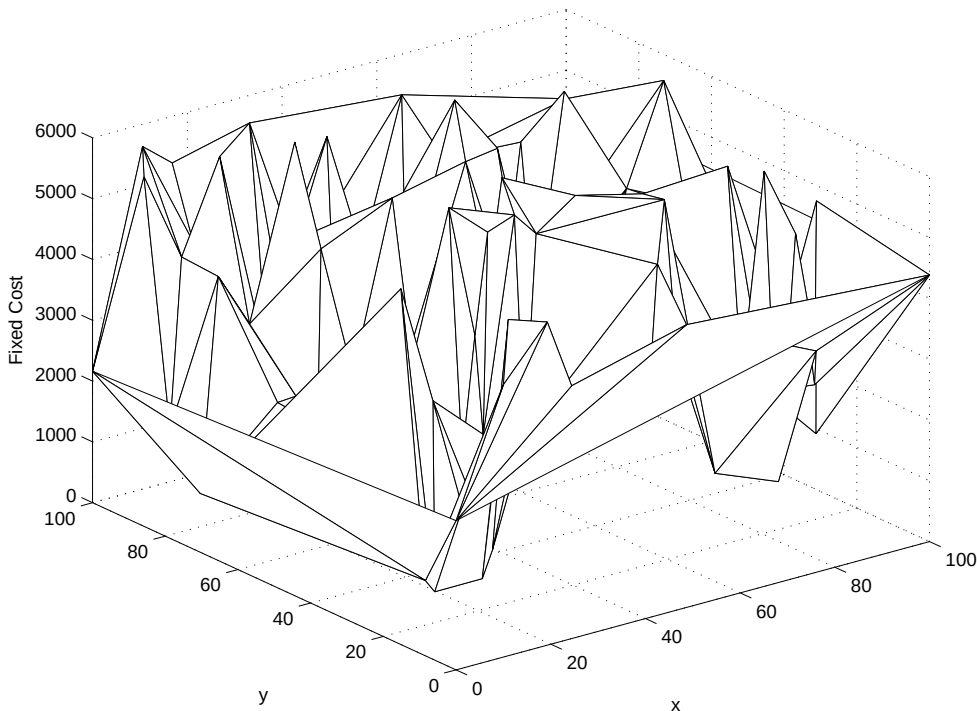


Figura 5.1: Primer refinamiento.

La cantidad de nodos clientes varía desde 100 nodos clientes hasta 1000 nodos clientes, i.e. los primeros 10 experimentos tienen 100 nodos clientes, los siguientes 10 experimentos tienen 200 nodos clientes, y así consecutivamente, llegando a que los últimos 10 experimentos tienen 1000 nodos clientes. El punto de

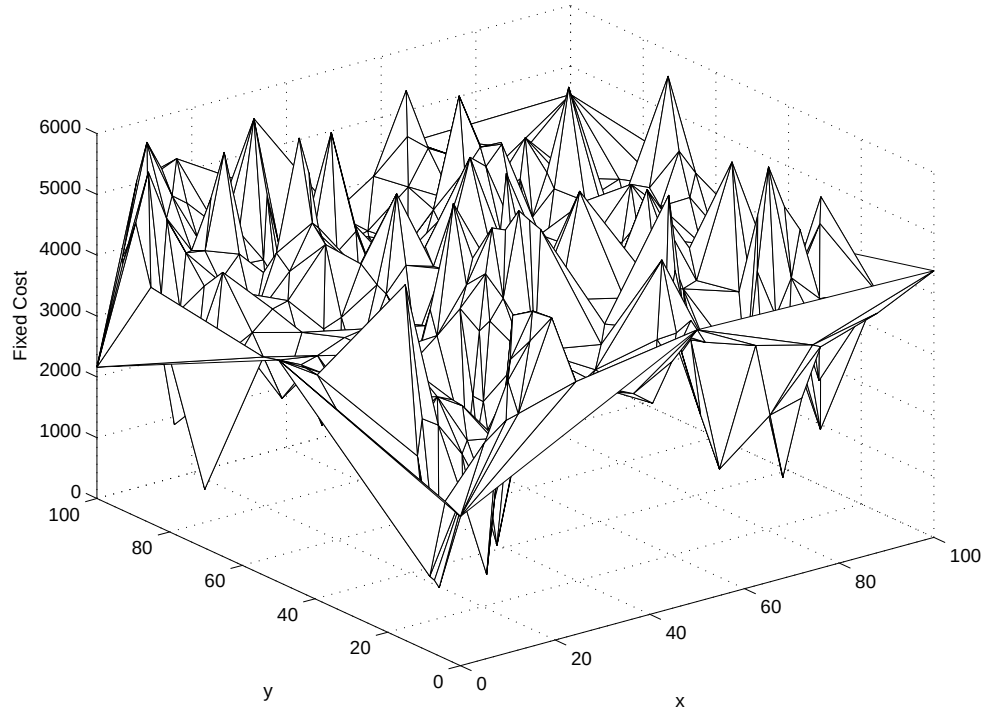


Figura 5.2: Segundo refinamiento.

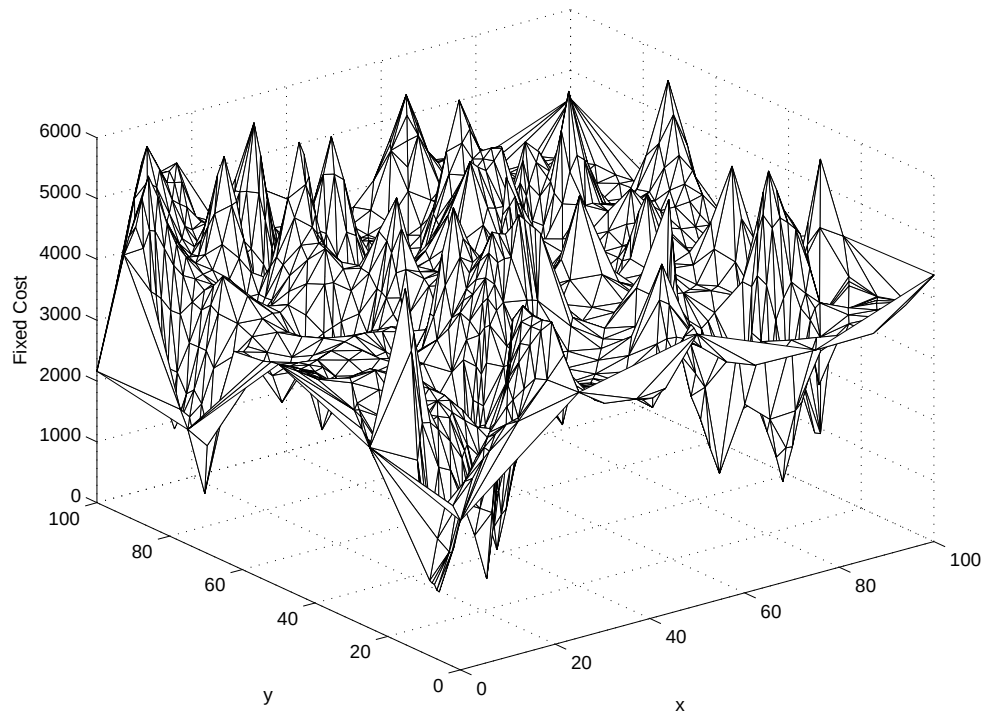


Figura 5.3: Tercer refinamiento.

localización de cada cliente es obtenido generando locaciones aleatorias, i.e. donde cada cliente j se ubica en $(xc_j, yc_j) \in ([0, 100], [0, 100])$.

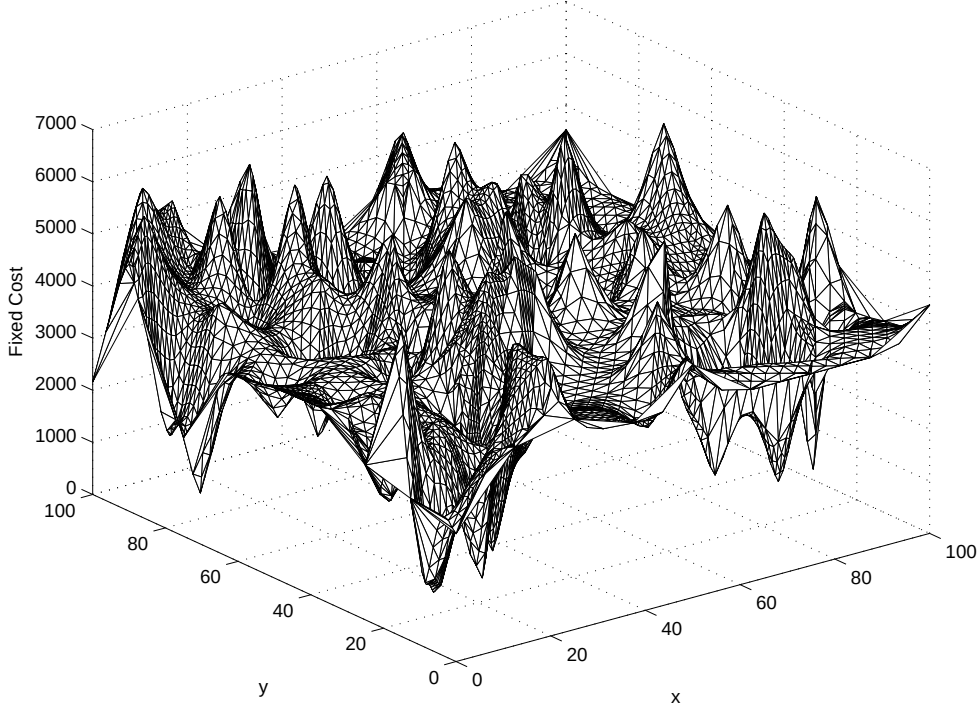


Figura 5.4: Cuarto refinamiento.

En la figura 5.5 se muestra el perfil de las performance basado en el *performance ratio* del tiempo CPU de cada propuesta de solución (Dolan y Moré, 2002). Considerando que t_{pm} es el tiempo CPU para resolver la instancia p mediante la propuesta de solución m , se tiene que el *performance ratio* es:

$$r_{pm} = \frac{t_{pm}}{\min\{t_{pm} : m \in M\}},$$

donde $m = \{DlogCP0, \min_{k \in K}\{(SP0(k))\}, \min_{k \in K}\{(SCP0(k))\}\}$.

Se puede observar en la figura 5.5 que la mejor performance de las propuestas es la de el método de descomposición, i.e., $\min_{k \in K}\{(SP0(k))\}$, porque en un 80 % de las instancias se obtiene el tiempo menor, sobrepasado por (*DlogCP0*), en menos del 20 % de las instancias. El método de descomposición tiene la mejor performance con la mayor eficiencia, resolviendo todas las instancias con un $r \leq 5$.

En las figuras 5.6, 5.7, 5.8 y 5.9 se muestran los *performance ratio* contra el número de la instancia considerando de forma separada para el primer, segundo, tercer y cuarto refinamiento, respectivamente. Se puede observar un claro patrón de comportamiento en cada refinamiento donde (*DlogCP0*) tiene una mejor performance en las instancias con un conjunto menor de nodos clientes, y es superado por el método de descomposición en el resto de las instancias. Este patrón es consistente con la tabla 5.2, donde se muestra el promedio de los speedup del tiempo CPU del método de descomposición sobre (*DlogCP0*) es superior a 1x para todos los refinamientos en las instancias con $|J| = 100$. Considerando el segundo y tercer refinamiento, (*DlogCP0*) es superior, en promedio, en las instancias con $|J| \leq 200$. Para el cuarto refinamiento, (*DlogCP0*) es mejor en todas las instancias con $|J| \leq 300$ y con un promedio del speedup de sobre 4x cuando se considera que $|J| = 100$.

Los resultados computacionales muestran que la performance de las formulaciones cónicas ((*DlogCP0*) y del método de descomposición cónica) son sensibles al tamaño del conjunto de nodos cliente. Esto se debe a que una formulación cónica crea $|J|$ conos y $3|J|$ nuevas variables, entonces el problema crece más rápidamente que el número de clientes. Por esta razón, a pesar que (*DlogCP0*) puede manejar mejor un conjunto mayor de nodos de información, esto se puede observar solamente cuando se tiene un conjunto de clientes

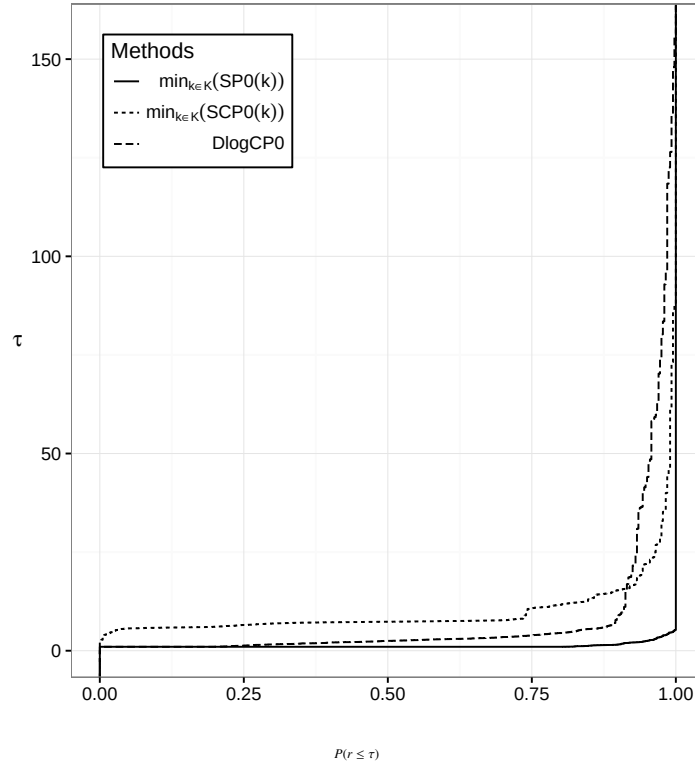


Figura 5.5: Perfil de las Performance.

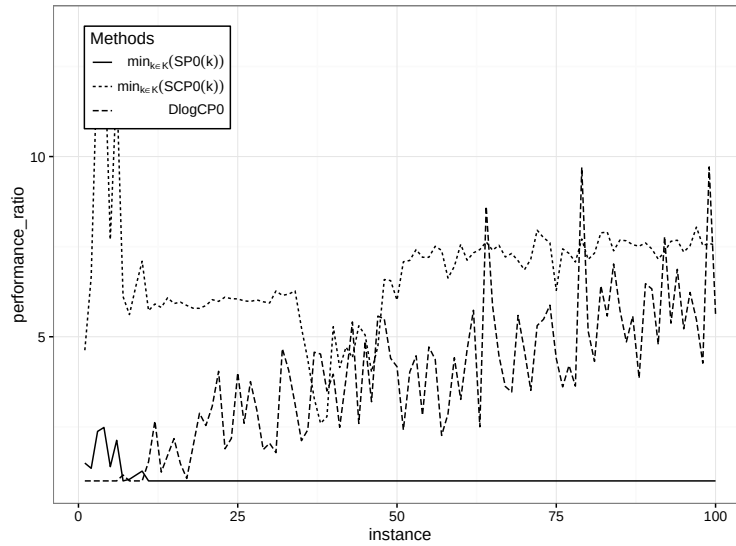


Figura 5.6: Performance ratio vs Número de la instancia, para el primer refinamiento.

pequeño.

La performance del método de descomposición, que se muestra en las figuras 5.5, 5.6, 5.7, 5.8 y 5.9, posee la performance más estable de entre las tres propuestas de solución, i.e., con menor diferencia entre los valores extremos de sus *performance ratio*. Esto indica que si ocurre que el método de descomposición no tiene la mejor performance en una instancia dada, su tiempo CPU está cercano al mejor tiempo CPU para dicha instancia.

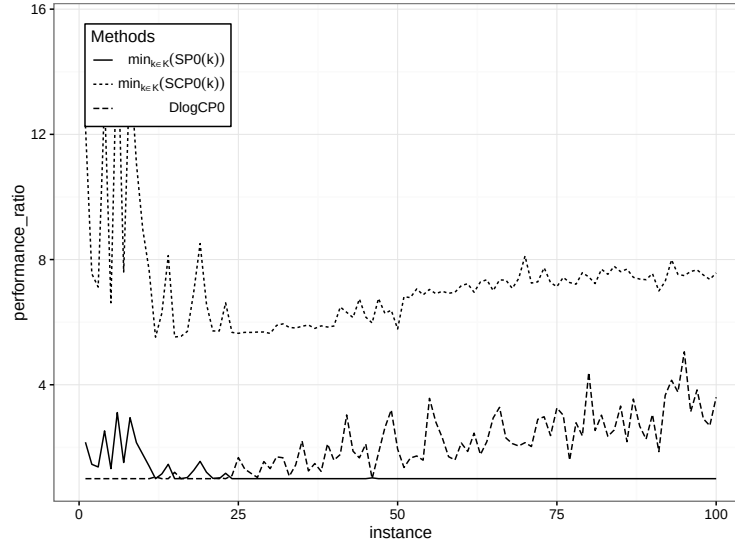


Figura 5.7: Performance ratio vs Número de la instancia, para el segundo refinamiento.

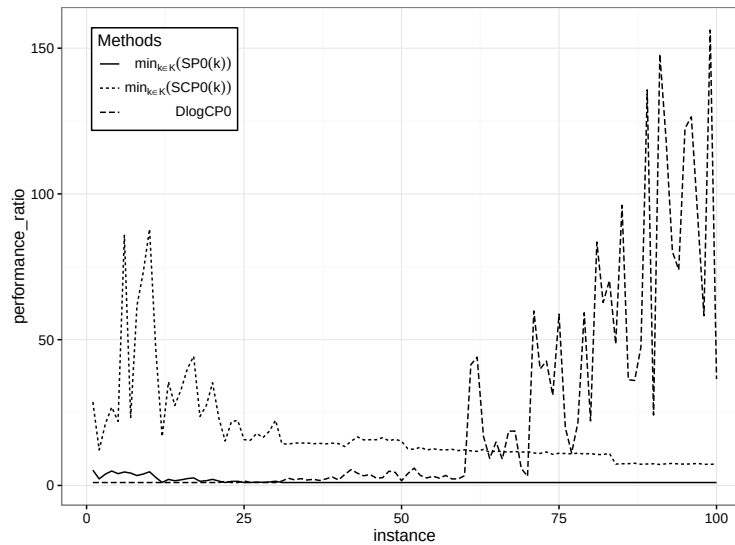


Figura 5.8: Performance ratio vs Número de la instancia, para el tercer refinamiento.

El promedio de las mejoras en la función objetivo usando los diferentes refinamientos, comparados con el primer refinamiento, son: 0,08 % para el segundo, 0,74 % para el tercero, y 1,29 % para el cuarto refinamiento.

Se puede observar en la tabla 5.2 que para las instancias con un número pequeño de clientes es mejor utilizar (*DlogCP0*), considerando que puede tener un speedup sobre 4x comparándose con el método de descomposición, pero esto ocurre cuando los tiempos CPU son más pequeños. Por ejemplo, en todas las instancias con $|J| = 100$, a pesar de obtenerse un mejor promedio del tiempo CPU con (*DlogCP0*), el peor caso para el método de descomposición no logra superar los 250 segundos. Considerando que el método de descomposición posee la performance más estable de las distintas propuestas y con un mejor promedio general en términos de tiempo CPU, y debido a que este problema resuelve una decisión estratégica, es recomendable para resolver el problema single-source de Weber con costos fijos utilizar la propuesta de solución del método de descomposición.

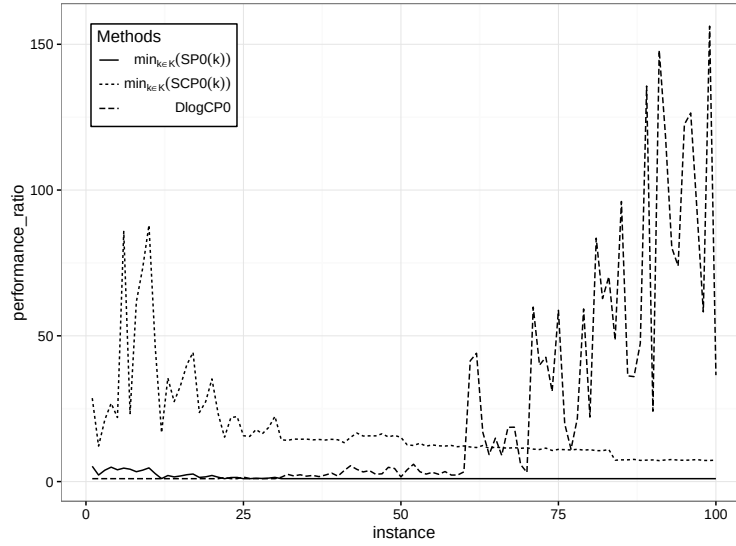


Figura 5.9: Performance ratio vs Número de la instancia, para el cuarto refinamiento.

Tabla 5.2: Speedup Promedio del tiempo CPU de $\min_{k \in K}\{(SP0(k))\}$ sobre $(DlogCP0)$.

J	Refinamiento			
	Primero	Segundo	Tercero	Cuarto
100	1.556x	2.035x	2.808x	4.125x
200	0.574x	1.185x	1.548x	1.944x
300	0.382x	0.865x	0.840x	1.149x
400	0.320x	0.665x	0.697x	0.494x
500	0.258x	0.528x	0.470x	0.308x
600	0.301x	0.530x	0.410x	0.330x
700	0.226x	0.447x	0.416x	0.099x
800	0.214x	0.387x	0.385x	0.036x
900	0.184x	0.373x	0.367x	0.020x
1000	0.172x	0.309x	0.312x	0.012x

5.3. Modelo Discreto: Uncapacitated Facility Location Problem

Los siguientes experimentos fueron generados resolviendo las instancias previamente descritas en el *conjunto de prueba*, considerando el conjunto de nodos de información sin los refinamientos. Se consideran los nodos de información como el conjunto discreto de posibles lugares de instalación, modelado a través de un UFLP.

En la tabla 5.3 se muestra el promedio y el máximo porcentaje de mejora considerando la disminución del valor de la función objetivo del problema single-source de Weber con costos por partes y continua sobre el UFLP.

Tabla 5.3: Porcentaje de mejora del modelo continuo sobre el UFLP.

	Refinamiento			
	Primero	Segundo	Tercero	Cuarto
Promedio	0.13 %	0.21 %	0.86 %	1.42 %
Máximo	1.62 %	5.80 %	16.83 %	18.58 %
N° Casos	27	8	25	33

La tabla 5.3 muestra, para el cuarto refinamiento un promedio de mejoría de un 1,42 %. Del total de los experimentos resueltos con el cuarto refinamiento, el 67 % de las instancias tiene el mismo resultado que el UFLP. Pero en el 33 % donde si hay diferencia, el promedio de mejora es de un 4,297 %.

Una mejor solución fue encontrada en el 23,25 % de las instancias utilizando el problema single-source de Weber con costos fijos contra la solución encontrada con el problema UFLP. Esto es debido a que el problema UFLP solamente considera los nodos de información como posibles localizaciones de la instalación y no siempre se considera al óptimo global en dicho conjunto. Considerando la inclusión de más nodos de información, es decir, más cercano a la realidad, el porcentaje de mejora y el número de mejores casos encontrados tiende a crecer.

Se puede observar a demás que en todas las instancias es instalada solamente una instalación. Esto está en concordancia con decir que, dentro de un área urbana, el costo de instalar un centro de distribución extra tiende a ser mayor que los ahorros en transporte.

5.4. Ejemplo Ilustrativo para un Caso Industrial

Se considera el caso de un mayorista que distribuye productos derivados de frutas y verduras el cual requiere localizar un centro de distribución en la ciudad de Santiago, Chile. Como es descrito en [Escalona et al. \(2015\)](#), se tienen 38 clientes y 38 nodos de información. Se aplican los cuatro refinamientos a la malla que fueron descritos y aplicados en el *conjunto de prueba*. En la tabla 5.4 se muestran los tiempos CPU para cada refinamiento de cada uno de las tres propuestas de solución.

Tabla 5.4: Tiempos CPU del Ejemplo Ilustrativo.

Refinamiento	Tiempo CPU (s)		
	$\min_{k \in K} \{(SP0(k))\}$	$\min_{k \in K} \{(SCP0(k))\}$	$DlogCP0$
Primero	0.844	1.67	0.266
Segundo	3.504	6.33	0.687
Tercero	15.153	25.163	3.109
Cuarto	63.607	107.938	11.797

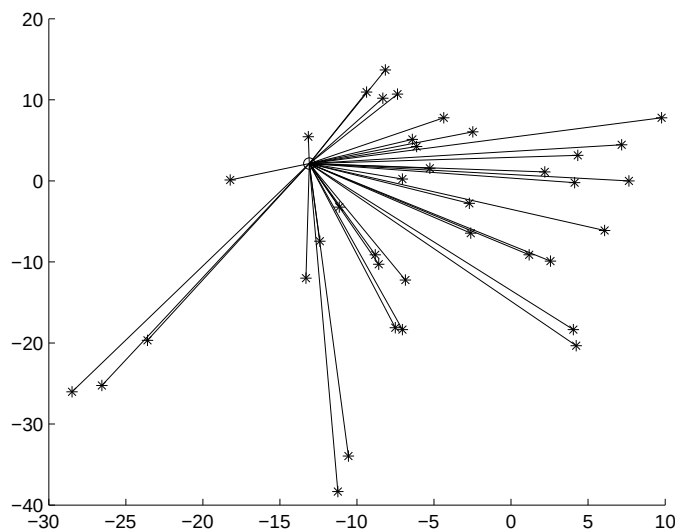
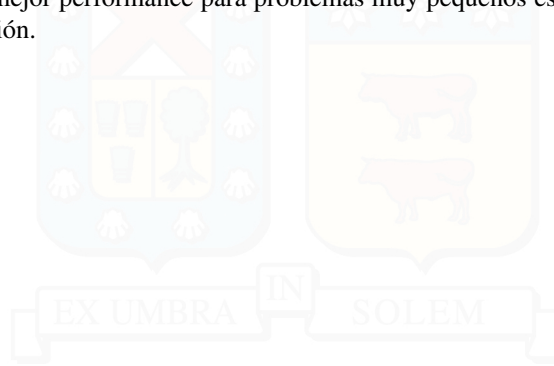


Figura 5.10: Localización del DC.

La localización óptima del centro de distribución está en las coordenadas $(-13,09, 2,14)$ con un costo total de 235,77. El costo está conformado en un 83 % por los costos de instalación y en un 17 % por los costos de transporte. En la figura 5.10 se muestra en un plano coordenado la localización del centro de distribución y de los clientes, con una línea que une a cada cliente con el centro de distribución.

Como puede ser observado en los resultados, el mejor tiempo CPU es del modelo (*DlogCP0*), debido a que este experimento considera un muy reducido y fijo número de nodos de clientes y de nodos de información, i.e. un número reducido de conos y de variables auxiliares son generadas. Como se muestra en el *conjunto de prueba*, la mejor performance para problemas muy pequeños es de (*DlogCP0*), seguido por el método de descomposición.





6 | Conclusiones

Este trabajo analiza el problema de localizar un único centro de distribución dentro de un área urbana considerando los costos fijos y de transporte. El problema es formulado como un problema single-source de Weber con costos fijos continuos por partes. Los costos fijos son caracterizados mediante un método de interpolación de Kriging usando un conjunto de nodos donde los costos son conocidos. Para hacer la formulación convexa y tratable, se utiliza una triangulación de Delaunay. Se proponen y evalúan tres propuestas de solución para resolver optimamente el problema single-source de Weber con costos fijos continuos por parte. La primera propuesta de solución considera una reformulación cónica del problema single-source de Weber con costos fijos continuos por partes y el uso de un método conocido como *logarithmic disaggregated convex combination model*. La segunda propuesta consiste en un método de descomposición, donde se resuelve un problema convexo no-lineal por cada triángulo de Delaunay, y usando enumeración completa se determina la solución óptima. La última propuesta considera una reformulación cónica para cada subproblema de la anterior descomposición.

En las instancias testeadas, en el 23,25 % de las veces, se obtiene una mejor solución con el problema single-source de Weber con costos fijos continuos por partes que con el UFLP. Se observan dos posibles explicaciones: (i) el conjunto I está completo y, entonces, la solución encontrada por el problema de Weber es infactible, y (ii) el conjunto I está incompleto y requiere una búsqueda más exhaustiva de locaciones factibles sobre el área urbana, es decir, el UFLP puede haber encontrado una solución sub-óptima. Poder asegurar que el un conjunto I está completo dentro de un área urbana es costoso y cercano a imposible. Es posible mejorar el conjunto de localizaciones factibles utilizando el enfoque continuo, focalizando la búsqueda de posibles localizaciones en una sección reducida del área urbana alrededor de la localización óptima encontrada con el problema single-source de Weber con costos fijos continuos por partes, donde es mas probable encontrar la *localización óptima y factible*, reduciendo así los esfuerzos de búsqueda de puntos factibles. Con esto, se puede aplicar un UFLP sobre el nuevo conjunto I eficientemente completado.

Los resultados computacionales muestran que la mejor propuesta de solución para el problema single-source de Weber con costos fijos continuos por parte, en términos del tiempo CPU promedio, es el método de descomposición, con un speedup promedio de $7,98x$ y $7,72x$ sobre el método de descomposición cónico y la reformulación monolítica cónica, respectivamente. La primera propuesta de solución tiene, en instancias pequeñas, la mejor performance y puede manejar mejor un conjunto de información más grande solamente si se tiene, a su vez, un conjunto pequeño de clientes. Lo anterior ocurre en las instancias donde la diferencia entre el tiempo CPU es más pequeño. Además, para el problema single-source de Weber sin considerar los costos fijos, el mejor método de solución es el algoritmo de Weiszfeld.

Quedan aún un número de preguntas y problemas dejados para investigaciones futuras, tales como: (i) aplicar algún algoritmo semejante a Weiszfeld para mejorar la performance de la formulación de descomposición, dado que fue la mejor para el problema de Weber sin costo fijo, (ii) usar una formulación de un modelo estocástico del problema single-source de Weber con costo fijo usando la varianza del método de interpolación de Kriging, (iii) considerar un problema de localización y ruteo, y (iv) generar una extensión a un problema multi-source de Weber con costos fijos continuos y dependientes de la localización considerando la mejor propuesta de solución encontrada en este trabajo.



Bibliografía

- Anselin, Luc y Le Gallo, Julie (2006). Interpolation of air quality measures in hedonic house price models: spatial aspects. *Spatial Economic Analysis*, 1(1), 31–52. 2.1, E.2
- Aras, Necati; Altinel, Kuban; y Orbay, Metin (2007). New heuristic methods for the capacitated multi-facility weber problem. *Naval Research Logistics*, 54(1), 21–32. 2.1
- Avriel, M. y Zang, I. (1980). Generalized arcwise-connected functions and characterizations of local-global minimum properties. *Journal of Optimization Theory and Applications*, 32(4), 407–425. 2.1
- Bongartz, Ingrid; Calamai, Paul H.; y Conn, Andrew R. (1994). A projection method for p norm location-allocation problems. *Mathematical Programming*, 66(1), 283–312. 2.1, E.2
- Brandeau, Margaret L. y Chiu, Samuel S. (1989). An overview of representative problems in location research. *Management Science*, 35(6), 645–674. 2.1
- Brimberg, Jack (1995). The fermat-weber location problem revisited. *Mathematical Programming*, 71(1), 71–76. 2.1
- Brimberg, Jack; Chen, Reuven; y Chen, Doron (1998). Accelerating convergence in the fermatweber location problem. *Operational Research Letters*, 22(4-5), 151–157. 2.1
- Brimberg, Jack; Drezner, Zvi; Mladenovic, Nenad; y Salhi, Said (2014). A new local search for continuous location problems. *European Journal of Operational Research*, 232(2), 256–265. 2.1, E.2
- Brimberg, Jack; Hansen, Pierre; Mladenovic, Nenad; y Salhi, Said (2008). Survey of solution methods for the continuous location-allocation problem. *International Journal of Operations Research*, 5(1), 1–12. 2.1, E.2
- Brimberg, Jack; Hansen, Pierre; Mladenovic, Nenad; y Taillard, Eric D (2000). Improvements and comparison of heuristics for solving the uncapacitated multisource weber problem. *Operations Research*, 48(3), 444–460. 2.1, E.2
- Brimberg, Jack; Mladenovic, Nenad; y Salhi, Said (2004). The multi-source weber problem with constant opening cost. *Journal of the Operational Research Society*, 55(6), 640–646. 1, 2.1, E.1, E.2
- Brimberg, Jack y Salhi, Said (2005). A continuous location-allocation problem with zone-dependent fixed cost. *Annals of Operations Research*, 136(1), 99–115. 1, 2.1, E.1, E.2
- Cavalier, Tom M. y Sherali, Hanif D. (1986). Euclidean distance location-allocation problems with uniform demands over convex polygons. *Transportation Science*, 20(2), 107–116. 2.1
- Cellmer, Radosław (2014). The possibilities and limitations of geostatistical methods in real estate market analyses. *Real Estate Management and Valuation*, 22(3), 54–62. 2.1, E.2
- Cellmer, Radoslaw; Belej, Miroslaw; Zrobek, Sabina; y Šubic-Kovač, Maruška (2014). Urban land value maps a methodological approach. *Geodetski vestnik*, 58(3), 535–551. 2.1, E.2

- Cellmer, Radosław; Senetra, Adam; y Szczepańska, Agnieszka (2012). Land value maps of naturally valuable areas. *Geomatics and Environmental Engineering*, 6(3), 15–24. 2.1
- Chen, Jein-Shan; Pan, Shaohua; y Ko, Chun-Hsu (2011). A continuation approach for the capacitated multi-facility weber problem based on nonlinear socp reformulation. *Journal of Global Optimization*, 50(4), 713–728. 2.1, E.2
- Chen, Pey-Chun; Hansen, Pierre; Jaumard, Brigitte; y Tuy, Hoang (1998). Solution of the multisource weber and conditional weber problems by d.-c. programming. *Operations Research*, 46(4), 548–562. 2.1, E.2
- Chica-Olmo, Jorge; Cano, Rafael; y Chica-Olmo, Mario (2007). Modelo hedónico espacio-temporal y análisis variográfico del precio de la vivienda. *GeoFocus. Revista Internacional de Ciencia y Tecnología de la Información Geográfica*, (7), 56–72. 2.1
- Cooper, Leon (1964). Heuristic methods for location-allocation problems. *SIAM Review*, 6(1), 37–53. 2.1, E.2
- Cooper, Leon (1972). The transportation-location problem. *Operations Research*, 20(1), 94–108. 2.1, E.2
- Daskin, Mark S (2011). *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons. 5.1
- Dolan, Elizabeth D y Moré, Jorge J (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2), 201–213. 5.2, E.5.1
- Drezner, Zvi; Klamroth, Kathrin; Schöbel, Anita; y Wesolowsky, George O (2002). 1 the weber problem. 2.1, E.2
- Eckhardt, Ulrich (1980). Weber's problem and weiszfeld's algorithm in general spaces. *Mathematical Programming*, 18(1), 186–196. 2.1
- Escalona, P.; Ordóñez, F.; y Marianov, V. (2015). Joint location-inventory problem with differentiated service levels using critical level policy. *Transportation Research Part E: Logistics and Transportation Review*, 83, 141–157. 5, 5.4
- Fernández-Avilés, Gema; Minguez, Roman; y Montero, Jose-Maria (2012). Geostatistical air pollution indexes in spatial hedonic models: the case of madrid, spain. *Journal of Real Estate Research*. 2.1, E.2
- Hansen, Pierre; Mladenovic, Nenad; y Taillard, Eric (1998). Heuristic solution of the multisource weber problem as a pmedian problem. *Operations Research Letters*, 22(2-3), 55–62. 2.1, E.2
- Helbich, Marco; Jochem, Andreas; Mücke, Werner; y Höfle, Bernhard (2013). Boosting the predictive accuracy of urban hedonic house price models through airborne laser scanning. *Computers, environment and urban systems*, 39, 81–92. 2.1, E.2
- Heuvelink, Gerard y Griffith, Daniel A (2010). Space-time geostatistics for geography: A case study of radiation monitoring across parts of germany. *Geographical Analysis*, 42(2), 161–179. 2.1
- Hosseininezhad, Sayed Javad; Salhi, Said; y Jabalameli, Mohammad Saeed (2015). A cross entropy-based heuristic for the capacitated multi-source weber problem with facility fixed cost. *Computers & Industrial Engineering*, 83, 151–158. 1, 2.1, E.1, E.2
- Hu, Shougeng; Tong, Luyi; Frazier, Amy E; y Liu, Yansui (2015). Urban boundary extraction and sprawl analysis using landsat images: A case study in wuhan, china. *Habitat International*, 47, 183–195. 2.1, E.2
- Kolbe, Jens; Schulz, Rainer; Wersing, Martin; y Werwatz, Axel (2015). Identifying berlins land value map using adaptive weights smoothing. *Computational Statistics*, 30(3), 767–790. 2.1
- Kuenne, Robert E. y Soland, Richard (1972). Exact and approximate solutions to the multisource weber problem. *Mathematical Programming*, 3(1), 193–209. 2.1

- Kuhn, Harold (1973). A note on fermat's problem. *Mathematical Programming*, 4(1), 98–107. 2.1
- Kuhn, Harold y Kuenne, Robert (1962). An efficient algorithm for the numerical solution of the generalized weber problem in spatial economics. *Journal of Regional Science*, 4(2), 21–33. 2.1
- Kuntz, Michael y Helbich, Marco (2014). Geostatistical mapping of real estate prices: an empirical comparison of kriging and cokriging. *International Journal of Geographical Information Science*, 28(9), 1904–1921. 2.1, E.2
- Larraz, B y Población, J (2013). An online real estate valuation model for control risk taking: A spatial approach. *Investment Analysts Journal*, 42(78), 83–96. 2.1, E.2
- Larson, William et al. (2015). *New Estimates of Value of Land of the United States*. Technical report, Bureau of Economic Analysis. 2.1
- Liang, Hua y Yi, Wen (2012). Effect of rent spatial distribution to urban business district planning. In *Advanced Materials Research*, volume 374 (pp. 2001–2008).: Trans Tech Publ. 2.1, E.2
- Luis, Martino; Salhi, Said; y Nagy, Gabor (2015). A constructive method and a guided hybrid grasp for the capacitated multi-source weber problem in the presence of fixed cost. *Journal of Algorithms & Computational Technology*, 9(2), 215–232. 1, 2.1, E.1, E.2
- Luo, Jun (2004). Modeling urban land values in a gis environment. *University of Wisconsin. Milwaukee, USA*. 2.1, E.2
- Martínez, Matías Gámez; Lorenzo, José María Montero; y Rubio, Noela García (2000). Kriging methodology for regional economic analysis: Estimating the housing price in albacete. *International Advances in Economic Research*, 6(3), 438–450. 2.1
- Matheron, Georges (1971). *The theory of regionalized variables and its applications*, volume 5. École nationale supérieure des mines. 2.4.2
- Megiddo, Nimrod y Supowit, Kenneth (1984). On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1), 182–196. 2.1, E.2
- Michelot, C. y Lefebvre, O. (1987). A primal-dual algorithm for the fermat-weber problem involving mixed gauges. *Mathematical Programming*, 39(3), 319–335. 2.1, E.2
- Mladenovic, Nenad; Brimberg, Jack; Hansen, Pierre; y Moreno-Perez, Jose A. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3), 927–939. 2.1, E.2
- Montero-Lorenzo, José-María y Larraz-Iribas, Beatriz (2012). Space-time approach to commercial property prices valuation. *Applied Economics*, 44(28), 3705–3715. 2.1
- Murtagh, B. A. y Niwattisayawong, S. R. (1982). An efficient method for the multi-depot location-allocation problem. *The Journal of the Operational Research Society*, 33(7), 629–634. 2.1
- Oliver, Margaret A y Webster, Richard (1990). Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information System*, 4(3), 313–332. 2.1, E.2
- Palermo, F.P. (1961). A network minimization problem. *IBM Journal of Research and Development*, 5(4), 335–337. 2.1
- Planchart, Alejo y Hurter, Arthur P. Jr. (1975). An efficient algorithm for the solution of the weber problem with mixed norms. *SIAM Journal on Control*, 13(3), 650–665. 2.1
- Plastria, Frank (1987). Solving general continuous single facility location problems by cutting planes. *European Journal of Operational Research*, 29(1), 98–110. 2.1, E.2

- Rosing, K.E. (1992). An optimal method for solving the (generalized) multi-weber problem. *European Journal of Operational Research*, 58(3), 414–426. 2.1
- Sherali, Hanif D.; Al-Loughani, Intesar; y Subramanian, Shivaram (2002). Global optimization procedures for the capacitated euclidean and l_p distance multifacility location-allocation problems. *Operations Research*, 50(3), 433–448. 2.1, E.2
- Taillard, Eric D. (2003). Heuristic methods for large centroid clustering problems. *Journal of Heuristics*, 9(1), 51–73. 2.1
- Vardi, Yehuda y Zhang, Cun-Hui (2001). A modified weiszfeld algorithm for the fermat-weber location problem. *Mathematical Programming*, 90(3), 559–566. 2.1
- Vielma, Juan Pablo; Ahmed, Shabbir; y Nemhauser, George (2010). Mixed-integer models for nonseparable piecewise linear optimization: Unifying framework and extensions. *Operations Research*, 58(2), 303–315. 4.1, E.4.1
- Weber, Alfred y Friedrich, Carl Joachim (1929). *Alfred Weber's Theory of the Location of Industries*. Chicago, Illinois: The University of Chicago Press. 2.1, E.2
- Weiszfeld, Endre y Plastria, Frank (2009). On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167(1), 4–41. 2.1, 2.3, E.2
- Wendell, Richard y Hurter, Arthur (1973). Location theory, dominance, and convexity. *Operations Research*, 21(1), 314–320. 2.1
- Xue, Guoliang y Ye, Yinyu (1997). An efficient algorithm for minimizing a sum of euclidean norms with applications. *SIAM Journal on Optimization*, 7(4), 1017–1036. 2.1, E.2

A | Glosario

La siguiente notación y parámetros serán utilizados a lo largo de este trabajo:

J : conjunto de nodos clientes.

I : conjunto de nodos de información.

K : conjunto de triangulos de Delaunay.

$\mathbf{w}$: vector de los pesos de la demanda esperada por los ratios de los costos de transporte, donde $\mathbf{w} \in \mathbb{R}^{|J|}$.

$(\bar{x}, \bar{y})$: coordenadas de la localización óptima.

$(\mathbf{x}, \mathbf{y})$: vector de la localización de los clientes, con dimensión $\mathbb{R}^{|J| \times 2}$.

$(\mathbf{x}_i, \mathbf{y}_i)$: vector de la localización de los nodos de información, con dimensión $\mathbb{R}^{|I| \times 2}$.

$\mathbf{C}$: matriz de los costos de instalación en los nodos de información, con dimensión $\mathbb{R}^{|I|}$.

$(\mathbf{x}_k, \mathbf{y}_k)$: vector de los vertices de los triangulos de Delaunay, con dimensión $\mathbb{R}^{|K| \times 3 \times 2}$.

$\mathbf{CK}$: matriz de los vertices de los triangulos, con dimensión $\mathbb{R}^{|K| \times 3}$.

λ : vector de la combinación convexa de los vertices de los triangulos de Delaunay, con dimensión $\mathbb{R}_0^+|^K| \times 3$.

$\mathbf{Z}$: vector binario que denota en qué triangulo está localizada la instalación.

Sea $\mathbf{a}, \mathbf{b} \in \mathbb{R}^t$ dos vectores cualesquiera y $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{t \times r}$ sea una matriz cualquiera, con $t, r \in \mathbb{Z}_+$. Sea $\mathbf{a} \cdot \mathbf{b}$ el producto interno de $\mathbf{a}$ y $\mathbf{b}$, $\|\cdot\|_2$ sea la norma euclidiana y $tr(\cdot)$ sea la traza de una matriz, con $tr(\mathbf{A}'\mathbf{B}) = \sum_{i,j=1}^{t,r} A_{ij}B_{ji}$.



B | Data del Ejemplo Ilustrativo

Tabla B.1: Data de los nodos de información

N°	x_i	y_i	Costo (US\$/día)
1	-9.47	21.02	191
2	-6.22	15.53	200
3	-6.67	15.92	195
4	-7.98	16.39	206
5	-8.6	16.71	206
6	-7.59	16.18	202
7	-7.56	15.23	211
8	-7.1	14.9	206
9	-6.45	14.93	211
10	-6.89	14.03	206
11	-5.97	13.62	244
12	-4.91	11.62	237
13	-3.56	12.01	233
14	-0.89	10.31	233
15	-4.49	9.56	255
16	-4.35	9.29	255
17	-3.64	5.38	233
18	-6.87	10.81	211
19	-9.55	10.16	222
20	-10.27	10.08	211
21	-10.64	10.15	206
22	-10.65	9.79	217
23	-11.08	9.89	211
24	-11.87	9.55	203
25	-12.34	8.76	208
26	-22.13	2.04	200
27	-12.88	2.72	222
28	-13.07	2.92	222
29	-13.1	2.53	211

N°	x_i	y_i	Costo (US\$/día)
30	-13.09	2.14	195
31	-12.32	2.31	211
32	-9.31	1.59	228
33	-8.72	1.51	228
34	-6.93	-6.22	228
35	-5.07	-3.42	217
36	-6.47	-8.86	222
37	-6.98	-9.36	208
38	-6.49	-9.47	208

Tabla B.2: Data de los nodos clientes

N°	x_i	y_i	w
1	-7.37	10.71	0.55768
2	-18.23	0.11	0.194
3	-13.31	-12.01	0.15304
4	-9.38	10.96	0.075701
5	-11.15	-3.24	0.0645
6	-5.27	1.55	0.10584
7	-6.13	4.23	0.0372
8	2.19	1.11	0.5856
9	4.13	-0.22	0.058001
10	-8.16	13.68	0.1508
11	-13.15	5.45	0.1772
12	-2.48	6.04	0.0264
13	-8.32	10.19	0.08408
14	7.18	4.45	0.0345
15	-6.4	5.12	0.10472
16	-8.81	-9.12	0.0264
17	-7.05	0.22	0.0582
18	1.18	-9.12	0.0264
19	-11.24	-38.33	0.068501
20	-6.86	-12.23	0.0171
21	-2.61	-6.45	0.0234
22	-7.05	-18.35	0.037601
23	7.65	0	0.021701
24	-7.51	-18.12	0.0482
25	-10.55	-33.96	0.029201



N°	x_i	y_i	w
26	4.04	-18.35	0.026201
27	-23.59	-19.68	0.0252
28	-2.71	-2.78	0.0297
29	-28.49	-26.02	0.022601
30	-12.41	-7.45	0.0167
31	9.77	7.78	0.015
32	6.07	-6.12	0.013301
33	-8.59	-10.3	0.0279
34	-4.37	7.78	0.0231
35	-26.55	-25.24	0.039701
36	2.56	-9.9	0.035
37	4.33	3.14	0.0333
38	4.23	-20.35	0.0164



C | Códigos de Propuesta de Solución

C.1. Código Logarithmic Disaggregated Convex Combination Model con Conos

.mod

```

param ni;
param nd;
param b;
param k;

set I={1..ni};
set J={1..nd};
set K={1..k};
set L={1..b};

set Sa{L};
set Sb{L};

param xi{I};
param yi{I};
param ci{I};
param xd{J};
param yd{J};
param vr;

param k1{K};
param k2{K};
param k3{K};

var Q;
var lambda{1..3,K}, >=0;
var m{L}, binary;
var d{J}, >=0;
var z{J};
var w{J};

minimize costo:
sum {j in J} vr*d[j]+Q;

```

```

subject to dist{j in J}:
d[j]^2>=(z[j])^2+(w[j])^2;

subject to igualdad1{j in J}:
z[j]=sum {i in K}(lambda[1,i]*xi[k1[i]]+lambda[2,i]*xi[k2[i]]+lambda[3,i]*xi[k3[i]])-xd[j];

subject to igualdad2{j in J}:
w[j]=sum {i in K}(lambda[1,i]*yi[k1[i]]+lambda[2,i]*yi[k2[i]]+lambda[3,i]*yi[k3[i]])-yd[j];

subject to sumQ:
sum{j in K}(lambda[1,j]*ci[k1[j]]+lambda[2,j]*ci[k2[j]]+lambda[3,j]*ci[k3[j]])<=Q;

subject to sumLambda:
sum{j in K}(lambda[1,j]+lambda[2,j]+lambda[3,j])=1;

subject to sumSa{i in L}:
sum{j in Sa[i]}(lambda[1,j]+lambda[2,j]+lambda[3,j])<=m[i];

subject to sumSb{i in L}:
sum{j in Sb[i]}(lambda[1,j]+lambda[2,j]+lambda[3,j])<=(1-m[i]);

```

.run

```

#LOAD THE MODEL
model CDlogP1.mod;

#Change the solver
option solver cplex;

set NODO:={100,200,300,400,500,600,700,800,900,1000};
param xp;
param yp;
param lambdaopt{1..3};

#####

for{qt in 4..4}{
printf "%s ; %s ; %s ; %s ; %s ; %s ; %s \n",
"Numero Nodo", "Numero Caso", "Costo", "X ", "Y", "Tiempo Solver",
"Lambda optimo 1; 2; 3","Resultado optimo" >(sprintf("resultadosCDlogP1%.0f.txt",qt));

for{dta in 1000..1000}{
for{cso in 8..10}{

reset data;
data (sprintf("%sinfo%01.0fn%04.0fc%02.0f.dat", "...\", qt, dta , cso));

let vr:=1;

#solve
solve;
let xp:=sum {i in K}(lambda[1,i]*xi[k1[i]]+lambda[2,i]*xi[k2[i]]+lambda[3,i]*xi[k3[i]]);
let yp:=sum {i in K}(lambda[1,i]*yi[k1[i]]+lambda[2,i]*yi[k2[i]]+lambda[3,i]*yi[k3[i]]);

```

```

let lambdaopt[1]:=sum {i in K}lambda[1,i];
let lambdaopt[2]:=sum {i in K}lambda[2,i];
let lambdaopt[3]:=sum {i in K}lambda[3,i];
#Show results

printf " %.0f ; %.0f ; %f ; %f ; %f ; %f ; %f ; %f ; %f ; %f \n",
dta, cso, costo, xp, yp, _solve_elapsed_time, lambdaopt[1], lambdaopt[2], lambdaopt[3],
solve_result >>(sprintf("resultadosCDlogP1%.0f.txt",qt));
}}}
```

C.2. Código Método de Descomposición

.mod

```

param ni;
param nd;
param k;
set I={1..ni};
set J={1..nd};
set K={1..k};

param xi{I};
param yi{I};
param ci{I};
param xd{J};
param yd{J};
param vr;
param l;

param k1{K};
param k2{K};
param k3{K};

var lambda{1..3}, >=0;

minimize costo:
sum {j in J} vr*sqrt((lambda[1]*xi[k1[l]]+lambda[2]*xi[k2[l]]+lambda[3]*xi[k3[l]]
- xd[j])^2+(lambda[1]*yi[k1[l]]+lambda[2]*yi[k2[l]]+lambda[3]*yi[k3[l]]-yd[j])^2)
+lambda[1]*ci[k1[l]]+lambda[2]*ci[k2[l]]+lambda[3]*ci[k3[l]];

subject to sumlambda:
sum{i in 1..3} lambda[i] =1;
```

.run

```

#LOAD THE MODEL
model P1k.mod;

#Change the solver
option solver minos;
```

```

param costopt;
param yopt;
param xopt;

set NODO:={ 100,200,300,400,500,600,700,800,900,1000};

for{qt in 1..4}{

printf "%s ; %s ; %s ; %s ; %s ; %s \n",
"Numero Nodo", "Numero Caso" , "Tiempo k", "lambda k1 ", "lambda k2 ", "lambda k3 ..."
>(sprintf("resultadosP1k%.0f.txt",qt));

for{dta in NODO}{
for{cso in 1..10}{
reset data;

data (sprintf("%sinfo%01.0fn%04.0fc%02.0f.dat","...\",qt , dta , cso));
let vr:=1;

let costopt:=99999999;
printf " %.0f ; %.0f ",dta, cso >>(sprintf("resultadosP1k%.0f.txt",qt));

for{a in K}{
let l:=a; #cambiar l dependiendo de que k se este
#solve
solve;
printf "; %.0f ; %f ; %f ; %f ; %f \n ;
",l, _solve_elapsed_time, lambda[1], lambda[2], lambda[3]
>>(sprintf("resultadosP1k%.0f.txt",qt));

if costopt > costo then {
let costopt:=costo;
let xopt:=lambda[1]*xi[k1[l]]+lambda[2]*xi[k2[l]]+lambda[3]*xi[k3[l]];
let yopt:=lambda[1]*yi[k1[l]]+lambda[2]*yi[k2[l]]+lambda[3]*yi[k3[l]];
}
}

#Show results
printf " %s ; %f ; %f ; %f \n","Optimo", costopt, xopt, yopt
>>(sprintf("resultadosP1k%.0f.txt",qt));
}}
close (sprintf("resultadosP1k%.0f.txt",qt));
}

```

C.3. Código Método de Descomposición Cónica

.mod

```

param ni;
param nd;
param k;
set I={1..ni};

```

```

set J={1..nd};
set K={1..k};

param xi{I};
param yi{I};
param ci{I};
param xd{J};
param yd{J};
param vr;
param l;

param k1{K};
param k2{K};
param k3{K};

var lambda{1..3}, >=0;
var d{J}, >=0;
var z{J};
var w{J};

minimize costo:
sum {j in J} vr*d[j] +lambda[1]*ci[k1[l]]+lambda[2]*ci[k2[l]]+lambda[3]*ci[k3[l]];

subject to dist{j in J}:
d[j]^2>=(z[j])^2+(w[j])^2;

subject to igualdad1{j in J}:
z[j]=lambda[1]*xi[k1[l]]+lambda[2]*xi[k2[l]]+lambda[3]*xi[k3[l]]-xd[j];

subject to igualdad2{j in J}:
w[j]=lambda[1]*yi[k1[l]]+lambda[2]*yi[k2[l]]+lambda[3]*yi[k3[l]]-yd[j];

subject to sumlambda:
sum{i in 1..3} lambda[i] =1;

.run

#LOAD THE MODEL
model CP1k.mod;

#Change the solver
option solver cplex;

param costopt;
param xopt;
param yopt;

set NODO:={ 100,200,300,400,500,600,700,800,900,1000};

for{qt in 1..4}{

printf "%s ; %s ; %s ; %s ; %s ; %s \n",
"Numero Nodo", "Numero Caso" , "Tiempo k", "lambda k1 ", "lambda k2 ", "lambda k3 ..."

```

```

>(sprintf("resultadosCP1k%.0f.txt",qt));

for{dta in NODO}{
for{cso in 1..10}{
reset data;

data (sprintf("%sinfo%01.0fn%04.0fc%02.0f.dat", "...\",qt, dta , cso));
let vr:=1;

let costopt:=99999999;

printf " %.0f ; %.0f ",dta, cso >>(sprintf("resultadosCP1k%.0f.txt",qt));

for{a in K}{
let l:=a; #cambiar l dependiendo de que k se este
#solve
solve;
printf " ; %.0f ; %f ; %f ; %f ; %f \n ;
",l, _solve_elapsed_time, lambda[1], lambda[2], lambda[3]
>>(sprintf("resultadosCP1k%.0f.txt",qt));

if costopt > costo then {
let costopt:=costo;
let xopt:=lambda[1]*xi[k1[1]]+lambda[2]*xi[k2[1]]+lambda[3]*xi[k3[1]];
let yopt:=lambda[1]*yi[k1[1]]+lambda[2]*yi[k2[1]]+lambda[3]*yi[k3[1]];
}

}

#Show results
printf " %s ; %f ; %f ; %f \n","Optimo", costopt, xopt, yopt
>>(sprintf("resultadosCP1k%.0f.txt",qt));
}}
close (sprintf("resultadosCP1k%.0f.txt",qt));
}

```

D | Problema Cónico de Weber

El problema single-source de Weber considera la localización de una sola instalación con coordenadas $(\bar{x}, \bar{y}) \in \mathbb{R}^2$. Esta instalación debe abastecer a un conjunto J de clientes. Se asume que las coordenadas de los clientes son conocidas. El problema es determinar la localización óptima para la única instalación tal que solamente el costo de transporte sea minimizado. El problema single-source de Weber puede ser expresado de la siguiente forma:

$$\min_{(\bar{x}, \bar{y})} \sum_{j \in J} w_j \|(\bar{x}, \bar{y}) - (x c_j, y c_j)\|_2 \quad (\text{D.1})$$

$$\text{s.t.} \quad (\bar{x}, \bar{y}) \in \mathbb{R}^2 \quad (\text{D.2})$$

El problema de Weber puede ser reformulado utilizando conos de segundo orden. Siguiendo los pasos descritos en la sección E.4 para la primera propuesta de solución. Por consiguiente, el problema cónico de Weber es expresado de la siguiente manera:

$$\min_{(\bar{x}, \bar{y}), \mathbf{d}, \mathbf{v}, \mathbf{r}} \sum_{j \in J} w_j d_j \quad (\text{D.3})$$

$$\text{s.t.} \quad d_j^2 \geq v_j^2 + r_j^2, \quad \forall j \in J \quad (\text{D.4})$$

$$v_j = \mathbf{x} \mathbf{k}_k^T \lambda^k - x c_j, \quad \forall j \in J \quad (\text{D.5})$$

$$r_j = \mathbf{y} \mathbf{k}_k^T \lambda^k - y c_j, \quad \forall j \in J \quad (\text{D.6})$$

$$\mathbf{d} \geq 0, \quad (\text{D.7})$$

$$(\bar{x}, \bar{y}) \in \mathbb{R}^2 \quad (\text{D.8})$$



E | Paper Realizado

A Single-Source Weber Problem with Continuous Piecewise Fixed Cost

Abstract

This paper analyzes the location of a distribution center in an urban area using a single-source Weber problem with continuous piecewise fixed cost to find a global optimal location. The fixed cost is characterized by a Kriging interpolation method. To make the fixed cost tractable, we approximate this interpolation with a continuous piecewise function that is convex in each piece, using Delaunay triangulation. We present a decomposition formulation, a decomposition conic formulation and a conic logarithmic disaggregated convex combination model to optimally solve the single-source Weber problem with continuous piecewise fixed cost. Although our continuous approach does not guarantee the global optimal feasible location, it allows us to delimit a zone where we can intensify the search of feasible points. For instances we tested, computational results show that our continuous approach found better locations than the discrete approach in 23,25 % of the instances and that the decomposition formulation is the best one, in terms of CPU time.

E.1. INTRODUCTION

The location of a distribution center (DC) in an urban area, considering the transportation and installation costs, can be treated as an uncapacitated facility location problem (UFLP) or as a Weber problem with fixed cost. It is known that the solution of the UFLP is feasible but not necessarily optimal, due to the use of an incomplete set of possible locations. On the other hand, the Weber problem with fixed cost gives an optimal location probably not feasible.

This paper analyzes the installation of a single DC in an urban area, using the single-source Weber problem with fixed cost to find an optimal location that allows us to delimit a zone around the optimal location previously found, but smaller than the original one. This way, we can focus the search of feasible points, obtaining a more reliable and complete set of possible locations such that, when an UFLP is applied, we find the *optimal feasible location*.

To the best of our knowledge, few papers deal with the inclusion of the fixed costs into the Weber problem. Fixed costs have been considered as a constant cost for all the plane (Brimberg et al., 2004), as zone dependent with a constant cost in a specific convex polygon (Brimberg y Salhi, 2005), (Hosseininezhad et al., 2015), or as a proportion between the fixed cost of two zones and their relative distance, (Luis et al., 2015). To consider that a plane can be partitioned in a finite set of convex polygons, each one with constant fixed costs, is considered a good first approximation to characterize the varying nature of this cost. In this paper we propose that the fixed cost on each convex polygon is a function of its vertices, allowing us to better model the fixed costs in an urban area.

The objective of this paper is to find the best formulation to locate a single DC in an urban area, where the fixed costs depend on the location in a continuous way. The fixed cost function is characterized by

a Kriging interpolation method using a set of nodes where the cost is known. To make the formulation tractable, we approximate the interpolation with a continuous piecewise function that is convex in each piece. This is constructed through a convex combination of the vertices of a mesh created with a Delaunay triangulation. The single-source Weber problem with continuous piecewise fixed cost is formulated as an MINLP problem. We take advantage of the problem's structure to propose three solution approaches. The first approach considers a conic reformulation of the single-source Weber problem with continuous piecewise fixed cost using a *logarithmic disaggregated convex combination model*. The second consists of a decomposition method, where we solve a non-linear convex problem for each Delaunay triangle, and using complete enumeration we determine the optimal solution. The last one, consider a conic reformulation for each sub-problem of the subsequent decomposition formulation. Each approach was implemented in a series of experiments to compare their performance in CPU time.

The main contributions of this paper are: (i) a new way to represent the fixed costs in an urban area and (ii) to identify the best solution approach for the single-source Weber problem with continuous piecewise fixed cost.

The paper is organized as follows: Section E.2 presents our related work. Section E.3 presents a single-source Weber problem with continuous piecewise fixed cost. Section E.4 presents three different approaches to solve the problem defined in the previous section. Section E.5 presents some experimental results for all the different approaches. Our conclusions and highlights are presented in section E.6.

E.2. RELATED WORK

The continuous location problem for a single-source or single-source Weber problem, described in [Weber y Friedrich \(1929\)](#), has been extensively studied. To find the solution there are different approaches: a one-point iterative method better known as the Weiszfeld algorithm ([Weiszfeld y Plastria, 2009](#)), a unified cutting plane method ([Plastria, 1987](#)), a dual method (?), a primal-dual algorithm involving mixed norms ([Michélot y Lefebvre, 1987](#)), or a primal-dual potential reduction algorithms with the problem formulated in conic form ([Xue y Ye, 1997](#)). A comprehensive review of the Weber problem can be found in [Drezner et al. \(2002\)](#).

The multi-source Weber problem, or location-allocation problem, is an NP-hard problem ([Megiddo y Supowit, 1984](#)). There are few heuristics that solve it to optimality but they work only in small problems ([Cooper, 1972](#)), ([Sherali et al., 2002](#)), ([Chen et al., 1998](#)). For the heuristic approach to solve the problem to near optimum, there are more publications: [Cooper \(1964\)](#) explored different algorithms with computational experiments. The alternating location-allocation heuristic is used by [Cooper \(1972\)](#). The method used by [Bongartz et al. \(1994\)](#) relaxes the binary constraints on the allocations, and solves both location and allocation simultaneously. An approach based on a nonlinear second-order cone program reformulation is found in [Chen et al. \(2011\)](#). The approach to use the discrete models in solving the continuous location-allocation problems is widely used by [Hansen et al. \(1998\)](#), [Brimberg et al. \(2014\)](#), and others. For this, a survey in the p-median problem with the aim in procedures based on metaheuristics rules ([Mladenovic et al., 2007](#)) is useful. For a survey on the multi-source Weber problem there is [Brimberg et al. \(2000\)](#) and [Brimberg et al. \(2008\)](#).

The inclusion of the fixed cost to the Weber problem has little reviews, there are four papers to the best of our knowledge. First it is included as a constant cost for all plane in [Brimberg et al. \(2004\)](#). Later, in [Brimberg y Salhi \(2005\)](#), it was extended to a zone-dependent fixed cost, where zones are non-overlapping convex polygons with a constant fixed cost for each zone. In [Hosseini-zhad et al. \(2015\)](#) is developed a metaheuristic Cross Entropy for a continuous location problem, with an fixed cost depending on the zone and on the facility to install. And [Luis et al. \(2015\)](#), proposed a multi-source Weber problem with capacity and zone-dependent fixed cost using the second-order Voronoi regions.

In general, data gathering is expensive in terms of monetary and time-consuming costs ([Helbich et al., 2013](#)). Therefore, there is a necessity to estimate the land values in unvisited locations, as geostatistical methods [Luo \(2004\)](#), [Cellmer et al. \(2014\)](#). Here, we use a Kriging method of interpolation ([Oliver y Webster,](#)

1990). This method was recommended over other interpolation approaches in [Anselin y Le Gallo \(2006\)](#) and [Fernández-Avilés et al. \(2012\)](#), in an air quality and pollution studies, respectively. The possibilities and limitations of geostatistical methods to approximate the land values are discussed in [Cellmer \(2014\)](#). A comparison between Kriging methods for the real estate market is discussed in [Kuntz y Helbich \(2014\)](#). The Kriging interpolation is used to find the value of land for different cities by [Liang y Yi \(2012\)](#), [Hu et al. \(2015\)](#), [Larraz y Población \(2013\)](#).

In summary, there are few previous works on single-source and multi-source Weber problem that include a second order cone formulation and, to the best of our knowledge, only one paper presents a solution approach. The few papers that include fixed costs make a simplistic representation of them that do not reflect their variations in an urban area. Unlike them, we make a more realistic representation of the fixed costs, considering different possible approaches for the Weber problem with fixed costs.

E.3. MODEL FORMULATION

E.3.1. Single-Source Weber Problem with Continuous Piecewise Fixed Costs

The generalized single-source Weber problem with fixed costs considers the localization of a single source with coordinates $(\bar{x}, \bar{y}) \in \mathbb{R}^2$. This source must supply a set J of customers with known coordinates, (x_j, y_j) for every $j \in J$. Let $f(\bar{x}, \bar{y})$ be the fixed cost incurred when the source is installed in $(\bar{x}, \bar{y})$. Let w_j be the expected demand weighted by the transportation ratios, for all $j \in J$. The problem is to determine the optimal location for the single source such that the transportation and the fixed costs are minimized. The generalized single-source Weber problem with fixed costs can be expressed as follows:

$$\min_{(\bar{x}, \bar{y})} \sum_{j \in J} w_j \sqrt{(\bar{x} - x_j)^2 + (\bar{y} - y_j)^2} + f(\bar{x}, \bar{y}) \quad (\text{E.1})$$

$$\text{s.t. } (\bar{x}, \bar{y}) \in \mathbb{R}^2 \quad (\text{E.2})$$

We consider a convenient set I of nodes with known information of their fixed costs, C_i , and their coordinates, (x_i, y_i) , for every $i \in I$. In our paper, the way to address the fixed costs is by applying a Kriging interpolation method and defining a continuous function for the cost in every point of the convex hull of I . This cost function is not simple and could not be convex. To make the continuous fixed cost function tractable, we are going to approximate the Kriging interpolation with a piecewise function that is convex in each piece. For this, we partition the convex hull of I through a polyhedral mesh and defined the continuous piecewise fixed cost function as the convex combination of the vertices of the mesh. To the best of our knowledge, it is better to use the smallest subset of information nodes possible with empty interior to create the polyhedra, i.e, using Delaunay triangulation.

We applied a Delaunay triangulation over the set I obtaining a set K of triangles; each triangle k -th will be denoted as P_k , with $P_k = \{(x, y) \in \mathbb{R}^2 | (x, y) = \sum_{l=1}^3 \lambda^{k_l} (x_{k_l}, y_{k_l}), \sum_{l=1}^3 \lambda^{k_l} = 1, \forall \lambda^{k_l} : \lambda^{k_l} \geq 0\}$, where (x_{k_1}, y_{k_1}) , (x_{k_2}, y_{k_2}) and (x_{k_3}, y_{k_3}) are the vertices of the k -th triangle and C_{k_1} , C_{k_2} , C_{k_3} their fixed cost. We have λ^{k_l} as the convex combination vector for the vertices of the triangle $k \in K$ and $l = 1, 2, 3$ the vertices of the triangle. The set of all possible locations, $\bigcup_{k \in K} P_k$, can be non-convex if we clean the areas where we can not install, as a lake or a strictly residential area.

Given the above, the facility's location can be expressed as $(\bar{x}, \bar{y}) = \sum_{k \in K} \sum_{l=1}^3 \lambda^{k_l} (x_{k_l}, y_{k_l})$ and its fixed cost as a convex combination of the vertices of the triangles's costs, $\sum_{k \in K} \sum_{l=1}^3 \mathbf{C}_{k_l}^T \lambda^{k_l}$. Let Z_k be a binary variable that forces the installation to be in only one triangle, being 1 if it is installed in the k -th triangle and 0 if it is not.

We can formulate the single-source Weber problem with continuous piecewise fixed cost as follows:

Problem (P0):

$$\min_{\mathbf{Z}, \lambda} \sum_{k \in K} \left(\sum_{l=1}^3 \mathbf{C}_{k_l} \lambda^{k_l} + Z_k \sum_{j \in J} w_j \sqrt{\left(\sum_{k \in K} \sum_{l=1}^3 \lambda^{k_l} x_{k_l} - x c_j \right)^2 + \left(\sum_{k \in K} \sum_{l=1}^3 \lambda^{k_l} y_{k_l} - y c_j \right)^2} \right) \quad (\text{E.3})$$

$$\text{s.t. } \sum_{l=1}^3 \lambda^{k_l} = Z_k, \quad \forall k \in K \quad (\text{E.4})$$

$$\sum_{k \in K} Z_k = 1 \quad (\text{E.5})$$

$$\lambda^{k_l} \geq 0, \quad \forall l \in \{1, 2, 3\}, k \in K \quad (\text{E.6})$$

$$Z_k \in \{0, 1\}, \quad \forall k \in K \quad (\text{E.7})$$

In what follows, we present different ways to solve the problem (P0).

E.4. SOLUTION APPROACH

We consider three distinct solution approaches for (P0). For the first approach, we use a monolithic reformulation of (P0). The second approach considers a decomposition of (P0) by fixing the variable $\mathbf{Z}$ and solving the sub-problem generated; we evaluated all the possible values of $\mathbf{Z}$. The last approach considers a conic reformulation of the previous sub-problems.

E.4.1. Conic Logarithmic Disaggregated Convex Combination Model

Now, we reformulate (P0) in two steps. First, we formulate the problem as a Conic Quadratic Non Linear problem (CQNLP). Afterwards, using the *logarithmic disaggregated convex combination model* (Vielma et al., 2010), we efficiently solve the continuous piecewise fixed cost function.

Next, we formulate the problem (P0) as a CQNLP in order to eliminate the square root terms. First we introduce one set of nonnegative continuous variables, d_j , to represent the square root term in:

$$d_j = \sqrt{\left(\sum_{k \in K} \sum_{l=1}^3 \lambda^{k_l} x_{k_l} - x c_j \right)^2 + \left(\sum_{k \in K} \sum_{l=1}^3 \lambda^{k_l} y_{k_l} - y c_j \right)^2}, \quad \forall j \in J \quad (\text{E.8})$$

$$d_j \geq 0, \quad \forall j \in J \quad (\text{E.9})$$

For simplicity, we can add two more sets of auxiliary variables, v_j and r_j , leaving (E.8) as:

$$d_j^2 = v_j^2 + w_j^2, \quad \forall j \in J \quad (\text{E.10})$$

$$v_j = \sum_{l=1}^3 x_{k_l} \lambda^{k_l} - x c_j, \quad \forall j \in J \quad (\text{E.11})$$

$$r_j = \sum_{l=1}^3 y_{k_l} \lambda^{k_l} - y c_j, \quad \forall j \in J \quad (\text{E.12})$$

Because the nonnegative variables d_j are introduced in the objective function of (P0) with positive coefficients, and this problem is a minimization problem, the equation can be further relaxed as the following inequalities:

$$d_j^2 \geq v_i^2 + r_i^2, \quad \forall j \in J \quad (\text{E.13})$$

Note that the constraints (E.9) and (E.13) define second-order cone constraints. The problem (P0) can be expressed as the following conic problem:

Problem (CP0):

$$\begin{aligned} \min_{\mathbf{Z}, \lambda, \mathbf{d}, \mathbf{v}, \mathbf{r}} \quad & \sum_{k \in K} (Z_k \sum_{j \in J} w_j d_j + \sum_{l=1}^3 C_{k_l} \lambda^{k_l}) \\ \text{s.t.} \quad & (E.4), (E.5), (E.6), (E.7), (E.9), (E.11), (E.12), (E.13) \end{aligned} \quad (\text{E.14})$$

The *logarithmic disaggregated convex combination model* consists in replacing the piecewise function $f(\bar{x}, \bar{y})$ for its epigraph $\text{epi}(f)$ and setting the coordinate $(\bar{x}, \bar{y})$ to be contained by one and only one of the domains of f . For a minimization, solving the function f is equivalent to solving $\text{epi}(f)$. To construct a model with the least number of binary variables and constraints, we identify each triangle with a binary vector in $\{0, 1\}^{\lceil \log_2 |K| \rceil}$ through an injective function $B : K \rightarrow \{0, 1\}^{\lceil \log_2 |K| \rceil}$ and use $\lceil \log_2 |K| \rceil$ binary variables, $\mathbf{m} \in \{0, 1\}^{\lceil \log_2 |K| \rceil}$, to ensure that the coordinates are in only one triangle. Let Q be $\text{epi}(f)$.

Using the *logarithmic disaggregated convex combination model* and a second order cone formulation to reformulate (P0), leaves the following:

Problem (DlogCP0):

$$\min_{\lambda, \mathbf{m}, \mathbf{d}, Q, \mathbf{v}, \mathbf{r}} \quad \sum_{j \in J} w_j d_j + Q \quad (\text{E.15})$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{l=1}^3 C_{k_l} \lambda^{k_l} \leq Q \quad (\text{E.16})$$

$$\sum_{k \in K} \sum_{l=1}^3 \lambda^{k_l} = 1 \quad (\text{E.17})$$

$$\sum_{k \in K^+(B, t)} \sum_{l=1}^3 \lambda^{k_l} \leq m_t, \quad \forall t \in T(K) \quad (\text{E.18})$$

$$\sum_{k \in K^0(B, t)} \sum_{l=1}^3 \lambda^{k_l} \leq (1 - m_t), \quad \forall t \in T(K) \quad (\text{E.19})$$

$$\lambda^{k_l} \geq 0 \quad \forall l = 1, 2, 3, k \in K \quad (\text{E.20})$$

$$m_t \in \{0, 1\} \quad \forall t \in T(K) \quad (\text{E.21})$$

$$(E.9), (E.11), (E.12), (E.13)$$

where $B : K \rightarrow \{0, 1\}^{\lceil \log_2 |K| \rceil}$ is any injective function, $K^+(B, t) = \{k \in K : B(k)_t = 1\}$, $K^0(B, t) = \{k \in K : B(k)_t = 0\}$ and $T(K) = \{1, \dots, \lceil \log_2 |K| \rceil\}$. This problem is a mixed integer conic quadratic nonlinear problem with a linear objective function and can be solved by solvers like GUROBI, CPLEX or MOSEK.

E.4.2. Decomposition Formulation

From the problem (P0), we can observe that the variables λ and $\mathbf{Z}$ are related in only one constraint. And, fixing the variable $\mathbf{Z}$, the problem is separable in $|K|$ sub-problems where we force the localization of the

DC to be in the k -th Delaunay triangle, i.e., forcing $Z_k = 1$ and $Z_{k'} = 0$ for all $k' \in K \setminus k$. Then the k -th sub-problem can be written as:

Sub-Problem (SP0(k)):

$$\min_{\lambda^k} \sum_{j \in J} w_j \sqrt{(\bar{x} - xc_j)^2 + (\bar{y} - yc_j)^2} + \sum_{l=1}^3 \mathbf{C}_{k_l} \lambda^{k_l} \quad (\text{E.22})$$

$$\text{s.t.} \quad \sum_{l=1}^3 \lambda^{k_l} = 1 \quad (\text{E.23})$$

$$\lambda^{k_l} \geq 0, \quad \forall l \in \{1, 2, 3\} \quad (\text{E.24})$$

This sub-problem (SP0(k)) is a convex nonlinear problem with linear constraint and can be efficiently solved by MINOS or IPOPT solvers.

Let λ^{k^*} be the optimal solution of the problem (SP0(k)); $FO_{(SP0(k))}(\lambda^{k^*})$ be the optimal cost of the objective function in the problem (SP0(k)), and let $(\bar{\lambda}, \bar{Z})$ be the optimal solution of the problem (P0). The optimal solution for the (P0) problem is the best solution for all of the sub-problems (SP0(k)), i.e. $\bar{\lambda} = \lambda^{k_{\dagger}}$, where $k_{\dagger} = \operatorname{argmin}_{k \in K} \{FO_{(SP0(k))}(\lambda^{k^*})\}$. For $\bar{Z}$, the value of $\bar{Z}_k = 1$ for $k = k_{\dagger}$ and $\bar{Z}_k = 0$ for every other k .

E.4.3. Decomposition Conic Formulation

The squared root term in the objective function of problem (SP0(k)) can give rise to difficulties in the optimization procedure. Following the logic exposed for the first approach, we reformulate (SP0(k)) as a CQNLP, leaving the following conic problem:

Sub-Problem (SCP0(k)):

$$\min_{\lambda^k, \mathbf{d}, \mathbf{v}, \mathbf{r}} \sum_{j \in J} w_j d_j + \sum_{l=1}^3 \mathbf{C}_{k_l} \lambda^{k_l} \quad (\text{E.25})$$

$$\text{s.t. } (E,23), (E,24), (E,9), (E,11), (E,12), (E,13)$$

The problem (SCP0(k)) can be trivially shown to be equivalent to (SP0(k)), but it has now conic and nonlinear constraints with a more simple linear objective function. The optimal solution for (P0) is the best solution for all the sub-problems (SCP0(k)), equivalently to the decomposition formulation.

The advantage of the CQNLP formulation is that it can be solved directly using standard optimization software packages such as CPLEX, GUROBI or MOSEK.

E.5. COMPUTATIONAL STUDY

In this section, we present our numerical study and its results. The main objectives of this computational study is to show which solution approach has the best performance in terms of CPU time, and to compare them to an UFLP. To characterize the different approaches, we carried out 400 instances that we denote *test set*. We also corroborate the installation of a single DC in every instance with the UFLP.

All the problems were programmed using AMPL. To solve the decomposition formulation we use the solver MINOS. For (DlogCP0) and the decomposition conic formulation we solve it through CPLEX solver. The Kriging interpolation method and the Delaunay triangulation were made in MATLAB. The *test set* were run on a PC with AMD FX 4,00 GHz processor and 12 GB RAM, and the UFLP were run on a PC with Intel i3 2,10 Ghz and 4 GB RAM.

E.5.1. Test Set

In order to determine which one has the best performance in CPU time, we generated 100 experiments. In each experiment, we fixed the number of customer nodes and used 4 refinements of the triangulation. Therefore, we have 400 instances. For simplicity, we considered $w_j = 1$, for any $j \in J$.

Each experiment has the same initial set of 100 information nodes, generated randomly. For a better piecewise convex approximation of the continuous fixed cost function, we proposed the following refinement of the mesh. We consider the Delaunay triangulation of the initial set of information nodes as the first refinement, shown in figure E.1. The second refinement is generated by creating additional information nodes where their location is at the center of the edge of every triangle and their fixed cost is determined by the Kriging interpolation. Then the Delaunay triangulation is used over the original set I plus the additional information nodes. The third and fourth refinements are applied over the second and third triangulation, respectively. In figure E.2 the fourth refinement is shown.

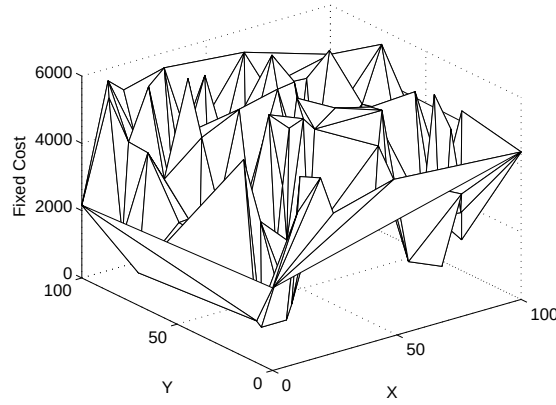


Figura E.1: First refinement.

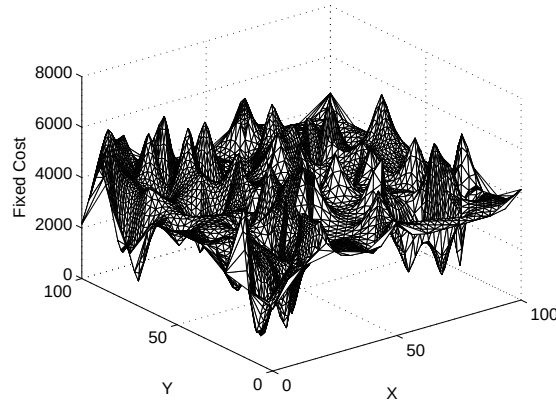


Figura E.2: Fourth refinement.

We modified the number of customer nodes from 100 to 1000 customers, i.e., the first 10 experiments have 100 customer nodes, the next 10 experiments have 200 customer nodes, and so on. Each customer location is obtained making random locations, i.e., where $(x_{c_j}, y_{c_j}) \in ([0, 100], [0, 100])$.

Figure E.3 shows the performance profile based on the *performance ratio* of the CPU time for each model (Dolan y Moré, 2002). Considering that t_{pm} is the CPU time for solving the instance p by the model

m , we have the *performance ratio*:

$$r_{pm} = \frac{t_{pm}}{\min\{t_{pm} : m \in M\}},$$

where $M = \{DlogCP0, \min_{k \in K}\{(SP0(k))\}, \min_{k \in K}\{(SCP0(k))\}\}$.

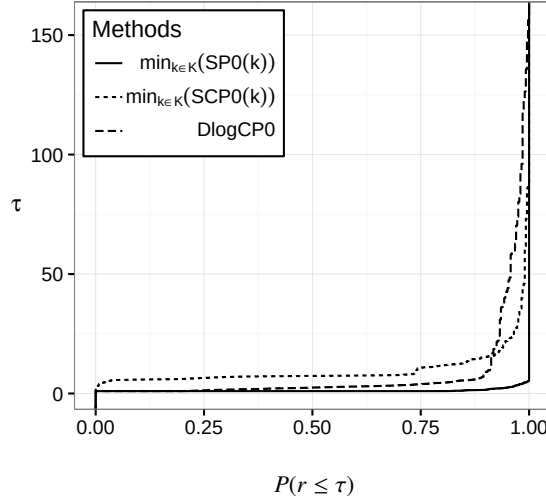


Figura E.3: Performance Profile.

We observe in figure E.3 that the best model performance is the decomposition formulation, i.e., $\min_{k \in K}\{(SP0(k))\}$, because in 80 % of the instances has the lowest time, overcome by (*DlogCP0*), in less than 20 % of the instance. The decomposition formulation has the best performance with the greater efficiency, solving all the instances with a $r \leq 5$.

There is a pattern in every refinement where (*DlogCP0*) has the best performance in the instances with a small set of customers nodes, and get outperformed by the decomposition formulation in the rest of the instances. This is shown in table E.1, where it shows that the average speedup in the CPU time of the decomposition formulation over (*DlogCP0*) is greater than 1x for all the refinements in the instances with $|J| = 100$. Considering the second and third refinement, (*DlogCP0*) is better, in average, for the instances with $|J| \leq 200$. For the fourth refinement, (*DlogCP0*) is better in instances with $|J| \leq 300$ and with an average speedup of over 4x when $|J| = 100$.

Tabla E.1: Average Speedup in CPU time of $\min_{k \in K}\{(SP0(k))\}$ over (*DlogCP0*).

$ J $	Refinement			
	First	Second	Third	Fourth
100	1.556x	2.035x	2.808x	4.125x
200	0.574x	1.185x	1.548x	1.944x
300	0.382x	0.865x	0.840x	1.149x
400	0.320x	0.665x	0.697x	0.494x
500	0.258x	0.528x	0.470x	0.308x
600	0.301x	0.530x	0.410x	0.330x
700	0.226x	0.447x	0.416x	0.099x
800	0.214x	0.387x	0.385x	0.036x
900	0.184x	0.373x	0.367x	0.020x
1000	0.172x	0.309x	0.312x	0.012x

We obtain an average speedup of 7.98x and 7.72x for the decomposition formulation over the decomposition conic formulation and (*DlogCP0*), respectively.

Our numerical results show that the performance from the conic formulations (*DlogCP0*) and the decomposition conic formulation) are sensible to the size of the customer set. This is because the conic formulations create $|J|$ cones and $3|J|$ new variables, so the problem grows faster than the number of customers. For this reason, even that (*DlogCP0*) can better handle a big set of information nodes, this only is seen with a small set of customers.

The performance of the decomposition formulation, shown in figure E.3, is the most stable of the performances of the three solution approaches, i.e., with less difference in the extremes values of its *performance ratio*. This indicates that if the decomposition formulation does not have the best performance in an instance, its CPU time is closer to the better one.

The average improvement in the objective function using the different refinements, compared with the first refinement, are: 0,08 % for the second, 0,74 % for the third, and 1,29 % for the fourth refinement.

We can observe in table E.1 that in instances with small number of customers is better to use (*DlogCP0*), considering that can have a speedup over 4x against the decomposition formulation, but it is when the CPU times are lower. For example, in all the instances with $|J| = 100$, although we have a better average of CPU time with (*DlogCP0*), the worst CPU time for the decomposition formulation does not get over 250 seconds. Considering that the decomposition formulation has a more stable performance with the better overall average in CPU time, and because this solves a strategic decision, we recommend to model the single source Weber problem with fixed cost with the decomposition formulation.

E.5.2. Discrete Model: Uncapacitated Facility Location Problem

The following experiments where made using the instances previously described in the *test set*, considering the set of information nodes without the refinements. We consider the information nodes as the discrete set of possible locations, modelled by an UFLP.

In table E.2 are the average and maximum percentage of the improvement in lowering the value of the objective function of the single-source Weber problem with continuous piecewise fixed cost over the UFLP, and the number of cases where this happened.

Tabla E.2: Percentage of improvement for the continuous model over the UFLP.

	Refinement			
	First	Second	Third	Fourth
Average	0.13 %	0.21 %	0.86 %	1.42 %
Max	1.62 %	5.80 %	16.83 %	18.58 %
N° Cases	27	8	25	33

Table E.2 shows, for the fourth refinement an average improvement of 1,42 %. From the total of experiments solved with the fourth refinement, the 67 % of the instances have the same result as the UFLP. But in the 33 % where they are different, the average improvement is of 4,297 %.

The better solutions found in 23,25 % of the instances with the single-source Weber problem over the UFLP is because the UFLP only consider the information nodes as possible locations and not always is consider the global optimum in that set. With the inclusion of more information nodes, i.e. closer to reality, the average savings and the number of better cases tend to grow.

We also observed that in all the instances only one facility is installed. This is in accordance to say that, in an urban area, the fixed cost of an extra DC tends to be bigger than the savings in transportation.

E.6. CONCLUSIONS

This paper analyses the problem of locating a single DC in an urban area considering the fixed and transportation costs using a single-source Weber problem with continuous piecewise fixed cost. The fixed cost is characterized by a Kriging interpolation method. Using a Delaunay triangulation, we make the fixed cost function convex and tractable. We propose and evaluate three solution approaches to optimally solve the single-source Weber problem with continuous piecewise fixed cost: (i) decomposition formulation, (ii) decomposition conic formulation, and (iii) logarithmic disaggregated convex combination model with a conic formulation.

In the instances we tested, in 23,25 % of the time, a better solution is found with the single-source Weber problem with continuous piecewise fixed cost than with the UFLP. We observe two possible explanations: (i) the set I is complete, and therefore, the solution of the Weber problem is unfeasible, and (ii) the set I is incomplete and requires a more thorough search of feasible locations over the urban area, i.e., the UFLP could have found a sub-optimal solution. To ensure a complete set I in an urban area is expensive and almost impossible. It is possible to improve with the continuous approach, the set of feasible locations focusing in a reduced section of the urban area around the optimal location found, where it is more probable to find the *optimal feasible location*, thus reducing the search effort of feasible points. With this, we can apply an UFLP over the new set I .

The computational results show that the best approach for the single-source Weber problem with continuous piecewise fixed cost, in terms of average CPU time, is the decomposition method, with an average speedup of 7.98x and 7.72x over the decomposition conic method and the conic monolithic reformulation, respectively. The first approach has the best performance and can better handle a bigger set of information nodes only with a small number of customers, but this happens in the instances where the difference between the CPU times are smaller.

There are a number of questions and issues left for future research, such as: (i) to apply some Weizfield-like algorithm to improve the performance of the decomposition formulation, given that that was the best one, (ii) to use the formulation of a stochastic model of the single-source Weber problem with fixed cost using the variance of the Kriging interpolation method, (iii) to consider a location-routing problem, and (iv) the extension to a multi-source Weber problem with continuous dependent fixed cost considering the best solution approach we obtain.