



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ciencias Matemáticas

Escuela Profesional de Computación Científica

**Simulación del uso del algoritmo de Karmarkar para
minimizar costos de una red de transporte mediante
Matlab**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Licenciado en Computación
Científica

AUTOR

Miguel Angel FERRER ADRIANO

ASESOR

Mg. Luis Javier VÁSQUEZ SERPA

Lima, Perú

2021



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Ferrer, M. (2021). *Simulación del uso del algoritmo de Karmarkar para minimizar costos de una red de transporte mediante Matlab*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Escuela Profesional de Computación Científica]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Miguel Angel Ferrer Adriano
Tipo de documento de identidad	DNI
Número de documento de identidad	43490993
URL de ORCID	No aplica.
Datos de asesor	
Nombres y apellidos	Luis Javier Vásquez Serpa
Tipo de documento de identidad	DNI
Número de documento de identidad	43389380
URL de ORCID	https://orcid.org/0000-0002-5414-6764
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	María Natividad Zegarra Garay
Tipo de documento	DNI
Número de documento de identidad	09206994
Miembro del jurado 1	
Nombres y apellidos	Hugo Froilan Vega Huerta
Tipo de documento	DNI
Número de documento de identidad	06147737
Datos de investigación	
Línea de investigación	No aplica.

Grupo de investigación	No aplica.
Agencia de financiamiento	Sin financiamiento.
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Latitud: -11.9752033 Longitud: -77.1335071
Año o rango de años en que se realizó la investigación	2020 - 2021
URL de disciplinas OCDE	Matemáticas aplicadas https://purl.org/pe-repo/ocde/ford#1.01.01



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América

FACULTAD DE CIENCIAS MATEMÁTICAS

ESCUELA PROFESIONAL DE COMPUTACIÓN CIENTÍFICA

ACTA DE SUSTENTACIÓN DEL INFORME DE TRABAJO DE SUFICIENCIA PROFESIONAL EN LA MODALIDAD VIRTUAL PARA LA OBTENCIÓN DEL TÍTULO PROFESIONAL DE LICENCIADO(A) EN COMPUTACIÓN CIENTÍFICA (PROGRAMA DE TITULACIÓN PROFESIONAL 2021-I)

En la UNMSM – Ciudad Universitaria – Facultad de Ciencias Matemáticas, siendo las 16:40 horas del jueves 21 de octubre del 2021, se reunieron los docentes designados como Miembros del Jurado Evaluador (PROGRAMA DE TITULACIÓN PROFESIONAL 2021-I): Dra. María Natividad Zegarra Garay (PRESIDENTE), Dr. Hugo Froilán Vega Huerta (MIEMBRO) y el Mg. Luis Javier Vásquez Serpa (MIEMBRO ASESOR), para la sustentación del Informe de Trabajo de Suficiencia Profesional titulado: “**SIMULACIÓN DEL USO DEL ALGORITMO DE KARMARKAR PARA MINIMIZAR COSTOS DE UNA RED DE TRANSPORTE MEDIANTE MATLAB**”, presentado por el señor **Bachiller Miguel Angel Ferrer Adriano**, para optar el Título Profesional de Licenciado en Computación Científica.

Luego de la exposición del Informe de Trabajo de suficiencia Profesional, la Presidente invitó al expositor a dar respuesta a las preguntas formuladas.

Realizada la evaluación correspondiente por los Miembros del Jurado Evaluador, el expositor mereció la aprobación **bueno**, con un calificativo promedio de **dieciséis (16)**.

A continuación, los Miembros del Jurado Evaluador dan manifiesto que el participante **Bachiller Miguel Angel Ferrer Adriano** en vista de haber aprobado la sustentación del Informe de Trabajo de Suficiencia Profesional, será propuesta para que se le otorgue el Título Profesional de Licenciado en Computación Científica.

Siendo las 17:20 horas se levantó la sesión firmando para constancia la presente Acta.



Firmado digitalmente por ZEGARRA
GARAY María Natividad FAU
20148092232 soft
Motivo: Soy el autor del documento
Fecha: 03.11.2021 17:27:07 -05:00

Dra. María Natividad Zegarra Garay
PRESIDENTE



Dr. Hugo Froilán Vega Huerta
MIEMBRO



Mg. Luis Javier Vásquez Serpa
MIEMBRO ASESOR

"Aquel que tiene un porqué para vivir, se puede enfrentar a todos los cómo"

Friedrich Nietzsche

Dedicatoria

Especialmente a mi Padre Amalio, que desde el cielo me guía en cada paso.
A mi madre Leonidas, por su gran amor, humildad y apoyo incondicional.
A mis hermanos Rafael, Walter, Alex y Gaby por la gran familia que somos.
A Rocio, por ser una buena referencia en mi vida y una muy noble persona.

Agradecimientos

Agradezco en primer lugar a Dios,
porque con él todo es posible.
Su voluntad ha hecho que concluya este trabajo.

Agradezco a mis padres Amalio y Leonidas,
por la formación de valores a todos sus hijos,
por hacernos personas de bien y por el profundo
amor que sienten hacia nosotros.

Agradezco a mis hermanos Rafael recuerda,
que con perseverancia cumplirás tus objetivos,
Walter eres y serás un ejemplo para mí, Gaby
de carácter complicado pero de gran corazón,
Alex noble persona con muchos sueños.

Agradezco a Rocio, por su aporte espiritual
a la culminación de este proyecto y por su deseo
desinteresado de verme cumplir mis sueños.

Mi más profundo agradecimiento al
Mg. Edinson Montoro Alegre, por sus enseñanzas
y a mi alma mater UNMSM por la formación.

Resumen

SIMULACIÓN DEL USO DEL ALGORITMO DE KARMARKAR PARA MINIMIZAR COSTOS DE UNA RED DE TRANSPORTE MEDIANTE MATLAB

MIGUEL ANGEL FERRER ADRIANO

2021

Asesor : Mg. Luis Javier Vásquez Serpa
Título obtenido : Licenciado en Computación Científica

En el presente trabajo, se elaboró un modelo matemático aplicado a una red de transporte, utilizando el algoritmo de modificado de Karmarkar, publicado en 1986 por R. Vanderbey para resolver programas lineales. La finalidad es obtener una óptima distribución de los viajes de una empresa de transporte interurbano que opera en el sur chico del Perú. Actualmente el proceso de programar los viajes se realiza a juicio de experto y en una hoja de cálculo, generando costos de operación muy altos, es por ello que nos vemos en la necesidad que crear una herramienta que minimice los costos bajo una óptima distribución de los viajes.

Finalmente se implementa computacionalmente una interfaz, bajo una data histórica almacenada en un archivo excel, mediante el lenguaje de programación Matlab.

Palabras Claves:

Programación Lineal
Algoritmo Karmarkar
Método gradiente proyectado
Puntos interiores
Matlab

Abstract

SIMULATION OF THE USE OF THE KARMARKAR ALGORITHM TO MINIMIZE COSTS OF A TRANSPORTATION NETWORK THROUGH MATLAB

MIGUEL ANGEL FERRER ADRIANO

2021

Adviser : Mg. Luis Javier Vásquez Serpa
Obtained title : Licentiate in scientific computing

In the present work, a mathematical model applied to a transport network was elaborated, using the Karmarkar modified algorithm, published in 1986 by R. Vanderbey to solve linear programs. The purpose is to obtain an optimal distribution of the trips of an interurban transport company that operates in the small south of Peru. Currently the process of scheduling trips is made by an expert judgment and in a spreadsheet, generating very high operating costs, that is why we see the need to create a tool that minimizes costs under an optimal distribution of the travels.

Finally, an interface is computationally implemented, under a historical data stored in an excel file, using the Matlab programming language.

Palabras Claves:

Linear programming
Karmarkar algorithm
Projected gradient method
Interior points
Matlab

Índice general

1 Preliminares	2
1.1 Empresa en estudio	2
1.2 Situación problemática	2
1.3 Planteamiento del problema	3
1.4 Objetivos	3
1.4.1 Objetivo general	3
1.4.2 Objetivo específico	3
1.5 Alcance y limitaciones	3
1.6 Motivación del proyecto	3
2 Marco Teórico	4
2.1 Antecedentes	4
2.2 Bases teóricas	5

2.2.1	Planificación del transporte de pasajeros	5
2.2.2	Programación lineal	6
2.2.3	Programación lineal en el Perú	6
2.2.4	Formulación de un modelo de Programación lineal	7
2.2.5	Método de punto interior	7
2.2.6	Desarrollo del método de punto interior	9
2.2.7	Modificación del algoritmo de Karmarkar	11
3	Metodología	17
3.0.1	Lugar de aplicación	17
3.0.2	Obtención de información	18
3.0.3	Explicación del modelo	20
4	Implementación Computacional	23
	Bibliografía	32

Índice de figuras

2.1	Etapas de planificación de operaciones en una empresa de transporte	5
2.2	Mapa del recorrido Lima-Ica (Fuente: copa del pacifico 2019)	5
2.3	Algunas aplicaciones de la programación lineal (Fuente: Elaboración propia) . . .	6
2.4	Se alcanza el optimo x^* siguiendo la trayectoria marcada. a) El método del Simplex, b) Método de Punto Interior (Fuente: Castro, 2000)	8
2.5	Punto inicial x^0 enmarcada dentro de las restricciones (Fuente: Castro, 2000) . .	10
2.6	Movimiento realizado en dirección del gradiente	11
2.7	proceso de escalamiento que se realiza en cada iteración	12
3.1	Programación de viajes ejecutadas en el tramo Lima e Ica y viceversa	18
3.2	Distribución de la demanda por bloque de hora y origen destino (Fuente: Elaboración propia)	19
3.3	Distribución de la demanda por bloque de hora y origen destino (Fuente: Elaboración propia)	20
3.4	Tabla de costos (Fuente: Elaboración propia)	21

4.1	ambiente de trabajo (Fuente: Matlab)	23
4.2	Ejecución en interfaces para dia 2 (día de alta demanda)	25
4.3	Ejecución en consola para dia 2 (día de alta demanda)	25

Introducción

En la actualidad, los costos de transporte de pasajeros vía terrestre se han incrementado, debido al alza en el precio del combustible, repuestos, además de lidiar diariamente con la informalidad que prevalece en este rubro. Optimizar la programación de viajes para minimizar los costos se ha convertido en una tarea necesaria para las empresas peruanas.

Hoy en día algunas empresas del sector realizan la programación de viajes basados en la experiencia, en el juicio de experto y de manera manual, sin seguir un modelo matemático y además sin tener en algunos casos una tecnología que les proporcione los datos de manera rápida y precisa.

El presente trabajo minimiza el costo producido en un día, optimizando la programación de los viajes, usando una modificación del algoritmo de Karmarkar propuesta por R. Vanderbey para solución de programas lineales.

El trabajo se ha dividido en 4 capítulos. El primero presentamos el contexto de estudio, así como el planteamiento del problema, los objetivos, los alcances y limitaciones que enfrentamos y la motivación que tuvimos para el desarrollo de este trabajo.

En el segundo capítulo detallaremos los conceptos previos y antecedentes en los que se generan la aplicación y desarrollo del problema mediante el algoritmo de Karmarkar. En el tercer capítulo nos centramos en la metodología utilizada, lugar donde se desarrolló la aplicación, la obtención de la información, así como la explicación del modelo. Finalmente se realizó el desarrollo computacional, en matlab.

Capítulo 1

Preliminares

1.1 Empresa en estudio

La empresa en estudio ofrece el servicio de transporte de pasajeros, carga y encomienda a nivel interregional y regional, Con más de 30 años de experiencia ha logrado consolidarse como una empresa líder en el corredor vial: Lima, Cañete, Chincha, Pisco, Ica y viceversa. Tiene un promedio de 40 mil pasajeros diarios y unos ingresos aproximados de 150 millones al año, entre los servicios que ofrecen: Servicio Vip, Express y Estándar. Para este caso de estudio nos centraremos en el Servicio Estándar que representa el 80 % de los ingresos de la empresa.

1.2 Situación problemática

Actualmente el proceso de programación de viajes del servicio normal se realiza con varios días de anticipación donde se actualiza una tabla dinámica que esta conectada a un gestor de base de datos, el cual muestra información de cantidad de boletos, cantidad de viajes, ingresos, fecha y hora de programación, el origen y destino de los pasajeros, entre otros.

Los cambios en la distribución de los viajes por cada día y bloque de hora se realizan teniendo con consideración los factores económicos y sociales, tecnológicos, fechas festivas en las ciudades del corredor vial, políticos y fenómenos naturales que han tenido incidencia directa en la programación. La empresa cuenta con 220 vehículos con una capacidad de 50 asientos, se realiza la distribución de viajes tratando de cubrir toda la demanda sin tener una herramienta o metodología para optimizar, una mala distribución aumenta los viajes innecesariamente y genera que el costo operativo aumente.

1.3 Planteamiento del problema

Para solucionar el proceso de programación de viajes realizado de manera manual en una hoja excel, se utilizara el método de programación lineal de Karmarkar, el cual optimizara el tiempo y esfuerzo del personal de la empresa.

1.4 Objetivos

1.4.1 Objetivo general

Minimizar los costos producidos por programación de viajes en el corredor rumbo sur que comprende: Lima, Cañete, Chincha, Pisco e Ica.

1.4.2 Objetivo específico

Determinar la optima distribución de viajes, satisfaciendo la afluencia de pasajeros en días de alta y baja demanda.

1.5 Alcance y limitaciones

Este trabajo esta enfocado en empresas de transporte de pasajeros con puntos intermedios donde pueden bajar y subir pasajeros. Nos limitaremos a realizar la simulación mediante el método de Karmarkar para el rumbo sur, tomando como concepto importante la satisfacción de la demanda.

1.6 Motivación del proyecto

Los años de trabajo, la experiencia en el rubro de transporte así como la formación científica, motivaron la optimización de este proceso, ademas la futura construcción del Tren de Cercanías que conectara Lima e Ica, nos motiva a tener un sistema optimo, bajo una metodología que se sostenga en el tiempo.

Capítulo 2

Marco Teórico

2.1 Antecedentes

Existen diversas literaturas que concentran sus fines en la parte teórica y practica aplicando técnicas heurísticas y metaheurística para resolver problemas de transporte, utilizando distintos métodos, uno de los más comunes es el método Simplex y algunos casos el método Karmarkar que ha tenido varias versiones propuestas por diferentes autores que han tratado de explicar los aspectos más complejos de este método.

Mauttone (2003) afirma que el uso de una herramienta de apoyo a la toma de decisiones para la planificación de turnos cobra cada vez más importancia en los países. Utilizando técnicas de algoritmos genéticos.

Uribe (2016) manifiesta que la asignación del recurso humano en la operación de los viajes representa en todo los sistemas una buena parte del costo operacional, es por ello que tener una optima programación, Para la asignación de turnos se plantea una solución en varias etapas secuenciales con técnicas heurísticas y metaheurísticas.

Vanderbei (1986) publica su trabajo en cual presenta una modificación al algoritmo original de Karmarkar para resolver programas lineales, entre sus características más importantes sobresalen que utiliza la gradiente proyectado, obvia el conocimiento apriori del valor optimo y su convergencia es más rápido que el método simplex.

Amstreicher (1986) realiza proyecciones sobre el espacio nulo de las restricciones, se introduce un nuevo concepto llamado dirección admisible, por ello se obtiene problemas relajados del inicialmente planteado.

2.2 Bases teóricas

2.2.1 Planificación del transporte de pasajeros

Ceder (2007) planifica las operaciones de una empresa de transporte en 4 etapas, la cual se puede apreciar mejor en la figura 2.1 donde se puede observar un proceso secuencial.

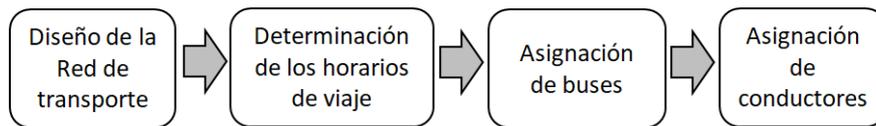


Figura 2.1: Etapas de planificación de operaciones en una empresa de transporte

La primera etapa nos habla del diseño de recorridos, en los cuales existe puntos donde pueden y subir y bajar pasajeros como se puede apreciar en la figura 2.2 tenemos los puntos Lima, Cañete, Chincha, Pisco e Ica a considerar. En la segunda etapa se debe decidir la hora de inicio en que el bus saldrá a ruta, en la tercera etapa se debe asignar los buses a los viajes de manera de cumplir todos y buscar la forma de minimizar los costos. Una vez establecido los itinerarios la ultima etapa asignar a los conductores, de acuerdo a las distintas restricciones contractuales. En este trabajo nos ocuparemos en minimizar los costos de operación, tratando de satisfacer la demanda, sujeta a los días de semana y bloques de hora.



Figura 2.2: Mapa del recorrido Lima-Ica (Fuente: copa del pacifico 2019)

2.2.2 Programación lineal

La programación lineal hace referencia a varias técnicas matemáticas usadas para la asignación óptima de recursos limitados a distintas demandas y competencias (Chase, Jacobs y Aquilano, 2009). La programación lineal debe su fama principalmente a los siguientes dos puntos: El primero a la gran variedad de problemas en diversos campos, estos pueden ser formulados o aproximados como modelos lineales, en efectos este tipo de problemas surgen con mucha frecuencia en la industria, en el sistema de transporte, en la comunicación, en la educación, entre otras (Hernandez, 2007).

Lo anterior no sería una ventaja si, además de poder formular los modelos, no existiera una buena manera de resolverlos. El segundo puntos consiste en que han sido desarrollados algoritmos eficientes para resolver modelos lineales. La programación lineal y sus extensiones han adquirido en los últimos años gran importancia dentro del mundo de las personas que toman decisiones a nivel empresa. La gran variedad de problemas lineales, esto unido al desarrollo tecnológico experimentado por los ordenadores cada vez más potentes y de menor precio, ha hecho que la programación lineal sea cada vez más difundida, puesto que ya es posible resolver grandes problemas que años atrás no se podían por la dificultad de la tecnología (Hernandez, 2007).

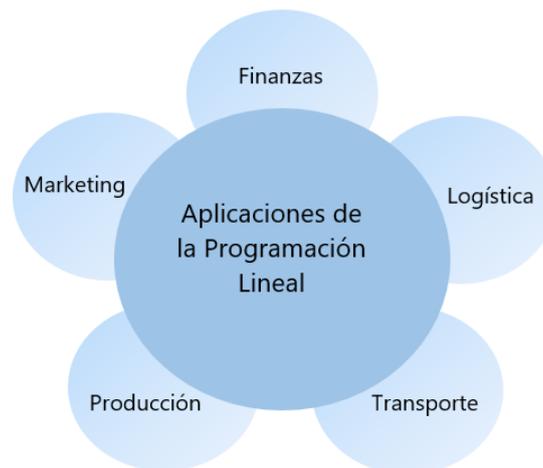


Figura 2.3: Algunas aplicaciones de la programación lineal (Fuente: Elaboración propia)

2.2.3 Programación lineal en el Perú

Esta ultima década se ha notado que diversos investigadores han solucionado diferentes problemas de las entidades industriales y empresariales aplicando la programación lineal dando soporte a los encargados en la toma de decisiones. Se han sistematizado los procesos en algunos casos. Cabe señalar también que existen filiales de empresas extranjeras que operan en nuestro país, emiten soluciones en su sede central a los problemas que afrontan sus operaciones en el Perú, desarrollando también en cierta medida la programación Lineal. Dentro de las instituciones que se conocen en el medio, mencionaremos a Dialnet, Empresa de Transporte Peru Bus, Instituto Nacional de Planificación, Ministerio de Agricultura, entre otros.

2.2.4 Formulación de un modelo de Programación lineal

En un modelo de programación lineal donde las variables de decisión poseen un comportamiento lineal, existe una relación sistemática entre función objetivo y las variables de decisión del problema. Es por ello que la programación lineal es la herramienta más utilizada en la investigación de operaciones, debido a su naturaleza facilitan los cálculos y tienen una buena capacidad de aproximación.

Se quiere solucionar el programa lineal que se puede observar en 2.1 esta dada en su forma estándar. Para su formulación debemos considerar lo siguiente: La función objetivo, es aquella que vamos a minimizar o maximizar. Determinación de las variables de decisión y de las restricciones.

$$\begin{aligned} \min_{\Omega} \quad & c^T x \\ \text{s.a} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{2.1}$$

Si $x \in \mathbb{R}^n$ es llamado vector de variables, $A \in \mathbb{R}^{m,n}$ matriz formada por las restricciones del programa (donde $m < n$), $b \in \mathbb{R}^m$ se conoce como términos independientes, y por ultimo $c \in \mathbb{R}^n$ el vector de costos. Los puntos que se encuentran dentro de $\Omega = \{x : Ax = b, x \geq 0\}$ son denominados factibles (o puntos admisibles) este conjunto debe ser acotado y cerrado.

Existen diferentes métodos para resolver programas de programación lineal, una de las más conocidas es el método Simplex publicada por Dantzig en el año 1949 y que actualmente es utilizado por su gran ductilidad sobre distintos problemas. Para este trabajo se tomara el método de puntos interiores de Karmarkar con una modificación planteado por Vanderbei en el año 1986.

2.2.5 Método de punto interior

Antiguamente este método era considerado solo para resolver programas no lineales, en 1984 Narendra Karmarkar lo utilizo en el algoritmo que lleva su nombre.

Son llamados métodos de punto interior ya que son generados dentro de la region factible, son algoritmos iterativos. Aquí radica la diferencia con respecto al método del Simplex, ya que esta avanza por el interior de la región y no por su frontera, haciendo más rápida la convergencia al punto optimo.

La Fig. 2.4 muestra gráficamente ambos comportamientos. Puede observarse como en el método de punto interior es necesario que el punto inicial x^0 sea también un punto interior de la región factible.

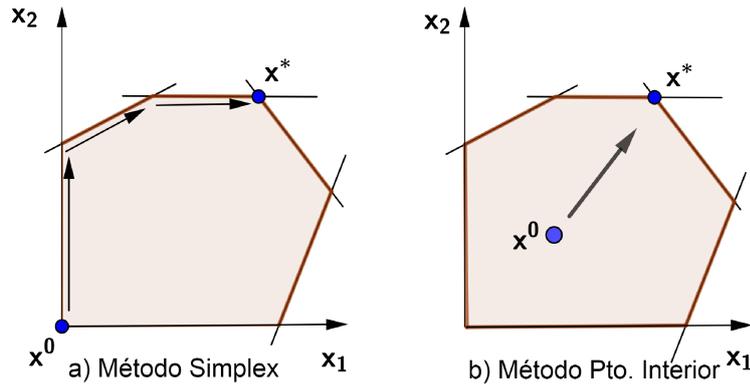


Figura 2.4: Se alcanza el óptimo x^* siguiendo la trayectoria marcada. a) El método del Simplex, b) Método de Punto Interior (Fuente: Castro, 2000)

Otras diferencias resaltantes respecto al método del Simplex es que existen algoritmos de punto interior polinómicos. Esto quiere decir que obtienen el punto óptimo en un número de iteraciones que es función polinómica del tamaño del problema. El método del simplex, en el peor de los casos, tiene un costo exponencial. En la actualidad los métodos de punto interior más eficientes tienen una complejidad de $\mathcal{O}(\sqrt{n}L)$, donde n es el número de variables del problema y L una medida del tamaño del problema (a saber, el número de bits necesarios para representar los datos del problema), (Castro, 2000).

Podemos clasificar en 4 grandes categorías los métodos de punto interior:

1. Métodos de escalado afín

De todos los métodos de punto interior estos son catalogados como los más sencillos. Sin embargo a pesar de esta calificación posee un rendimiento que sobresale entre las demás, fue sugerido por Dikin matemático Ruso en 1967, sin embargo no tomó mayor importancia hasta que un par de décadas más tarde el método fue presentado en occidente de forma independiente por diferentes investigadores de la época como una variante del algoritmo de las transformaciones proyectivas de Karmarkar (Castro, 2000).

2. Métodos basados en transformaciones proyectivas

Este método transforma el programa lineal original bajo una transformación proyectiva. Teniendo como resultado una forma más compleja. Estas transformaciones complejas dadas Tienen la siguiente forma:

$$\tilde{x} = \frac{D^{-1}x}{e^T D^{-1}x}$$

donde $x \in \mathbb{R}^n$ es el punto actual, \tilde{x} representa la imagen de x en el nuevo espacio de variables, D es una matriz diagonal que sus componentes son los del vector x , ($D = \text{diag}(x_1, \dots, x_n)$), y e^T es el vector n -dimensional de unos $e^T = (1, \dots, 1_n)$. Este es el famoso primer algoritmo de punto interior, el matemático Hindú Karmarkar lo desarrolló, pertenece a esta categoría. De allí nace la inquietud de la comunidad científica por promover nuevas soluciones y modificaciones a este método.

3. Los métodos de ruta a seguir

Estos utilizan una técnica para problemas no lineales, llamada barrera logarítmica. Solucionan una secuencia de programas en vez de solucionar 2.1 directamente, viene dado de la forma

$$\begin{aligned} \min_x \quad & c^T x - \mu \sum \ln(x_j) \\ \text{s.a.} \quad & Ax = b \end{aligned}$$

Puede probarse que, a medida que $\mu \rightarrow 0$ la secuencia de soluciones de la ecuación anterior, se acerca a la solución del problema 2.1 original.

4. Los métodos de reducción de potencial

Mediante uso de artificios matemáticos se reduce una función de potencia.

Al considerar los métodos de punto interior para programación lineal; uno inmediatamente piensa en ir en dirección de la gradiente. Sin embargo, esto dará una reducción sustancial en la función objetivo solo si el punto factible actual se centra tanto en el politopo de manera que todas las paredes son suficientemente distante, por lo tanto nos lleva a considerar un método que alterne entre centrar el punto factible y dando un paso en la dirección del gradiente. Muchos métodos interiores existentes son de este tipo, la eficiencia y la elegancia de cualquier método particular depende de la elección del esquema de centrado. El resto de la tesis se considerara un esquema de centrado especial.

2.2.6 Desarrollo del método de punto interior

El objetivo es solucionar el programas lineal formulado en 2.1 (nuestras dimensiones son n variables y m restricciones). Para la ejecución del algoritmo consideremos que se tiene un punto que se encuentra en el interior de la región factible $x^0 \in \mathbb{R}^n$. Esto significa que satisface las restricciones del problema. Que sea interior quiere decir que no sobrepase las fronteras de la región factible, esto implica que, $x > 0$. Por tanto, nuestro x^0 punto interior inicial en resumen debe cumplir con lo siguiente:

$$\begin{aligned} Ax^0 &= b \\ x^0 &> 0 \end{aligned}$$

En la figura 2.5 podemos apreciar el punto x^0 dentro de la región factible, reflejado de color azul. Las líneas punteadas de color rojo muestran la frontera de la región factible.

El algoritmo básico consiste en, tomar como punto de inicio el x^0 , de genera una secuencia de puntos $\{x^k\}$ hasta la trayectoria del punto más óptimo x^* del programa. Así, inicializando en x^0 seguimos hasta x^1 , de éste hasta x^2 , y así secuencialmente, hasta que se tenga alguna condición que nos indique que el punto actual x^k es solución óptima para el problema planteado. Cada punto nuevo que se halle, se obtendrá en el interior, determinado por el siguiente proceso iterativo:

$$x^{k+1} = x^k + \alpha \Delta x \tag{2.2}$$

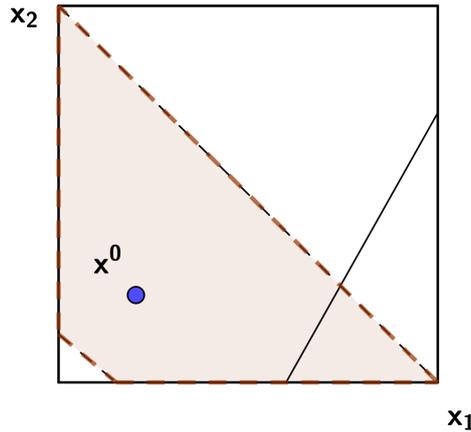


Figura 2.5: Punto inicial x^0 enmarcada dentro de las restricciones (Fuente: Castro, 2000)

Aquí podemos determinar que $\Delta x \in \mathbb{R}^n$ representa nuestro vector de movimiento en dirección del nuevo punto, mientras que el escalar α , denominado longitud de paso, es positivo y nos indica cuanto nos alejamos de x^k a lo largo de la dirección calculada. Este esquema es común a la mayoría de métodos de optimización.

El método de punto interior se centra en dos cuestiones, obtener la dirección de movimiento y calcular la longitud de paso. Las ideas detrás de los métodos de punto interior son las siguientes, suponiendo que un punto interior factible inicial este disponible y que todos los movimientos satisfagan $Ax = b$. Una forma de evitar que los puntos sobrepasen la frontera es transformar la región factible para que el punto interior factible actual esté en el centro de la región factible transformada. Una vez que se ha realizado el movimiento, el nuevo punto interior se transforma nuevamente en el espacio original, y todo proceso se repite con el nuevo punto como centro.

Manteniendo la viabilidad

Está claro que el punto factible debe satisfacer $Ax = b$ por lo tanto, los únicos puntos interiores factibles $x = x^k$ son aquellos que satisfacen $Ax^k = b$ y $x^k > 0$. Como resultado, cada vez que nos movemos desde un punto x^k (en el espacio original o espacio transformado) a $x^{k+1} = x^k + \alpha\Delta x$, debemos satisfacer $Ax^{k+1} = b$ y $x^{k+1} > 0$, esto implica que: $Ax^{k+1} = A(x^k + \alpha\Delta x) = b + \alpha A\Delta x$

Dirección de descenso

Lo siguiente a tener en cuenta es que la dirección de disminución máxima o descenso más pronunciado es el gradiente negativo de la función objetivo $c^T x$. Por lo tanto, $-c$. Teniendo claro todos estos elementos, nosotros utilizaremos el algoritmo modificado de Karmarkar, propuesta por R. Vanderbi para buscar solución a un problema de transporte, mediante la minimización de sus costos, tras una optima distribución de sus viajes.

2.2.7 Modificación del algoritmo de Karmarkar

En esta sección se muestra una modificación del Algoritmo Original de Karmarkar, para programas lineales, utilizando el método de punto interior, el cual requiere de un esquema de centrado.

Esquema de centrado

La mejor dirección esta dado por la gradiente proyectado, esto debido a que entre todas las direcciones factibles es la que tiene máxima disminución de $c^T x$ por unidad de longitud de paso, a pesar de ello el proceso secuencial del método. Si nos encontramos muy cerca de la frontera de la región factible definidas por las restricciones $x \geq 0$, la longitud de paso α que podremos realizar sera pequeña (sino cumplimos con esto las variables podrían caer en el lado negativo).

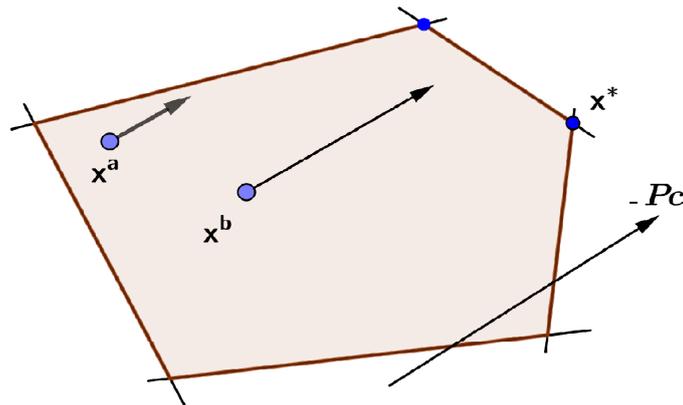


Figura 2.6: Movimiento realizado en dirección del gradiente

Tiene sentido cambiar las unidades de cada variable de manera que el punto factible actual x^0 se convierta en un vector de todos los unos. En estas coordenadas, la matriz A de restricciones y el vector de costos c cambian, pero las restricciones de desigualdad no, por lo tanto en estas coordenadas, nuestro punto es factible al menos una unidad de distancia de todas las paredes del politopo.

Desde el punto de vista de un físico, el esquema de centrado anterior debe sonar particularmente agradable. Después de todo, tomar el gradiente de una función donde las diversas coordenadas tienen unidades diferentes no es razonable. Pero el esquema de centrado anterior hace que coordinen las direcciones, por lo tanto juegan un papel importante.

Ahora escribimos los detalles, dado un punto factible x^0 hacemos el siguiente cambio de

unidades.

$$\tilde{x}_i = \frac{x_i}{x_i^0} \quad i = 1, 2 \dots n$$

x^0 punto actual de iteración

$$\begin{aligned} x_i &= \tilde{x}_i x_i^0 \quad i = 1, 2 \dots n \\ \tilde{x} &= D_{x^0}^{-1} x \end{aligned} \quad (2.3)$$

Donde:

D_x matriz diagonal que contiene los componentes de x , es evidente que x^0 se le asigna a $\tilde{x}^0 = \mathbf{1}$ en términos de coordenadas, Además mediante un artificio matemático podemos ver que el escalado esta bien definido, ya que siempre nos caminamos por el interior de la región factible, con lo que $x_i^0 > 0$, se define D_x no singular y positiva, así obtenemos un nuevo programa lineal con A y c sustituidos por:

$$\tilde{A} = A D_{x^0} \quad (2.4)$$

$$\tilde{c} = D_{x^0} c \quad (2.5)$$

Es decir transforma el programa lineal 2.1 en un programa escalado

$$\begin{aligned} \text{mín} \quad & \tilde{c}^T \tilde{x} \\ \text{s.a} \quad & \tilde{A} \tilde{x} = b \\ & \tilde{x} \geq 0 \end{aligned}$$

El punto x^0 que tenemos inicialmente, se transforma en el punto $\tilde{x}^0 = (1, 1 \dots, 1)^T$ que tiene una posición centrada en el nuevo programa lineal. en la figura 2.2.7 podemos apreciar geoméricamente como se realiza el proceso de escalamiento.

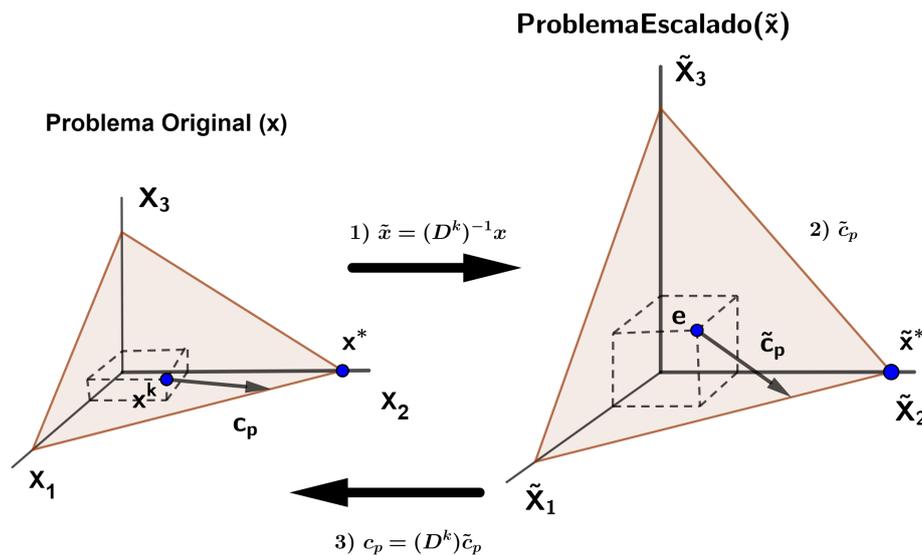


Figura 2.7: proceso de escalamiento que se realiza en cada iteración

Para obtener el máximo porcentaje de disminución en la función objetivo transformada, debemos movernos en la dirección negativa del gradiente \tilde{c} , sin embargo para conservar las restricciones de igualdad debemos proyectar \tilde{c} sobre el espacio nulo de \tilde{A} .

$$\tilde{c}_p = \tilde{P}_{x^0} \tilde{c} \quad (2.6)$$

Donde \tilde{P}_{x^0} denota la matriz proyección sobre el espacio nulo de $AD_x = \tilde{A}$. Luego nos movemos de \tilde{x}^0 hasta \tilde{x}^1 en dirección de $-\tilde{c}_p$ y se elige la longitud α de paso de manera que $\tilde{x}^1 > 0$.

$$\tilde{x}^1 = \tilde{x}^0 - \alpha \frac{\tilde{c}_p}{\text{máx}(e_i \cdot \tilde{c}_p)} \quad (2.7)$$

Donde $\alpha \in (0, 1)$ es la longitud de paso, $e_i = (0, 0, \dots, 1, 0, 0, \dots, 0)^T$ es el vector de i -ésimo componente. Si $\tilde{c}_p \leq 0$, el programa no será acotado esto se verá más adelante.

Finalmente hacemos un nuevo mapeo, para conseguir un nuevo punto factible interior. Utilizando la Ecuación 2.1, podemos obtener:

$$\begin{aligned} x^1 &= D_{x^0} \tilde{x}^1 \\ x^0 &= D_{x^0} \tilde{x}^0 \end{aligned}$$

Además de la Ecuación 2.7 se tiene que:

$$\begin{aligned} \tilde{x}^1 &= \tilde{x}^0 - \alpha \frac{\tilde{c}_p}{\text{máx}(e_i \cdot \tilde{c}_p)} \\ D_{x^0} \tilde{x}^1 &= D_{x^0} \tilde{x}^0 - \alpha \frac{D_{x^0} \tilde{c}_p}{\text{máx}(e_i \cdot \tilde{c}_p)} \end{aligned}$$

Combinando con los resultados anteriores, se tiene:

$$x^1 = x^0 - \alpha \frac{D_{x^0} P_{x^0} D_{x^0} c}{\text{máx}(e_i \cdot P_{x^0} D_{x^0} c)}$$

Ahora denotamos $x^1 = T(x^0)$ Entonces:

$$T(x) = x - \frac{\alpha}{\gamma} D_{x^0} P_{x^0} D_{x^0} c \quad (2.8)$$

Donde:

$$\gamma = \text{máx}(e_i \cdot P_{x^0} D_{x^0} c) \quad (2.9)$$

De esto se tiene:

$$x^{k+1} = x^k - \frac{\alpha}{\gamma} D_{x^k} P_{x^k} D_{x^k} c$$

El algoritmo consiste en generar puntos de la forma que se muestra a continuación.

Por lo tanto cada iteración k cambiamos de escala y avanzamos en la dirección de la gradiente negativa. Finalmente volvemos al escalar del espacio original, luego repetimos el proceso iterativo bajo una regla de parada.

$$x^{k+1} = T(x^k) \quad (2.10)$$

Debemos tener en consideración, que el programa lineal solucionado bajo el método de Karmarkar debe cumplir las siguientes proposiciones:

Proposición 2.1 Sabiendo que $\tilde{c}_p(x) = P_x D_x c$, se cumplen las siguientes propiedades.

1. Si la función objetivo no es constante en Ω , entonces la secuencia cx^k es estrictamente decreciente.
2. Si para algún $x \in \Omega^0$, tenemos que $\tilde{c}_p(x) \leq 0$ y $\tilde{c}_p(x) \neq 0$, entonces el problema no es acotado.
3. Si para algún $x \in \Omega^0$, tenemos que $\tilde{c}_p(x) = 0$, entonces todo punto factible es óptimo.

Convergencia

En esta sección tomaremos, bajo ciertos supuestos, que el algoritmo converge a la solución óptima. Para mostrar esto introduciremos variables duales e implícitamente reconstruiremos la teoría de dualidad. Teniendo nuestro programa 2.1 podemos obtener su dual, que estará dado por.

$$\begin{aligned} & \max_{y,z} b^T y \\ \text{s.a. } & A^T y + z = c \\ & z \geq 0 \end{aligned}$$

Asumimos las siguientes propiedades.

1. El programa 2.1 es acotado y factible.
2. El programa 2.1 es no degenerado.
3. El programa dual es no degenerado.

Ahora que ya tenemos el escalamiento, el proceso iterativo y la convergencia del algoritmo, nos falta imponer reglas que determinen el óptimo de un punto.

Reglas de parada

En la sección anterior descubrimos el comportamiento del algoritmo, ya que se aproxima a un punto límite. En esta sección vamos analizar con más detalle el comportamiento en una pequeña vecindad entorno a un punto límite. Esta información se utiliza para establecer reglas de parada para nuestro algoritmo.

Para $\gamma \geq 0$, $\mathbf{1} = (1, 1, \dots, 1)$

Donde,

$$\gamma = \max(e_i P_x D_x c)$$

Se tiene,

$$S_\gamma = \{x \in \Omega : D_x r \leq \gamma \mathbf{1}\} \quad (2.11)$$

Decimos que los puntos en S_γ satisfacen la γ -Holgura complementaria si $x \in S_\gamma$ ahora sabemos que:

$$\begin{aligned} \mathbf{1} - \frac{\alpha}{\gamma} D_x^2 r &\geq 0 \\ \mathbf{1} &\geq \frac{\alpha}{\gamma} D_x^2 r \\ \gamma \mathbf{1} &\geq \alpha D_x^2 r \geq D_x r \\ \gamma \mathbf{1} &\geq D_x r > 0 \end{aligned} \quad (2.12)$$

De esto se tiene que:

$$F_\delta = \{x \in \Omega : c \geq A^T w - \delta \mathbf{1}\} \quad (2.13)$$

Ahora combinando las ecuaciones 2.12 y 2.13 vemos que si $x \in S_\gamma \cap F_\delta$, entonces

$$\begin{aligned} c^T x &\leq w^T b + \gamma n \\ w^T(x).b - \delta n \max \bar{z} &\leq \min c^T z \end{aligned}$$

De esto se puede inferir que:

$$c^T x \leq \min c^T z + \gamma n + \delta n M \quad (2.14)$$

Donde: $M \geq \bar{z}$, Si Ω es limitado, entonces M es finito.

Para cualquier $x \in \Omega$, podemos medir el grado de holgura complementaria y dual feasible poniendo

$$\gamma(x) = \text{máx } x_i r_i(x)$$

$$\delta(x) = - \text{mín } r_i(x)$$

Gracias a que esta definición de γ esta de acuerdo con la definición dada anteriormente. Ahora podemos formular la regla de parada efectiva.

$$\gamma = \text{máx}(e_i P_x D_x c)$$

Entonces

$$\tilde{c}_p = P_x \tilde{c}$$

$$\tilde{c} = D_x c$$

$$c_p = D_x r$$

Por lo que se tiene

$$\delta = \text{máx}(e_i D_x r)$$

$$\gamma = \text{máx}(x_i r_i)$$

Proposición 2.2 Fijo $\varepsilon > 0$, si M es una cota superior de \bar{z} , $z \in \Omega$ y si nos detenemos en la primera iteración en la que

$$\gamma(x^k) + \delta(x^k)M \leq \frac{\varepsilon}{n}$$

Entonces

$$c^T x^k \leq \text{mín } c^T z + \varepsilon$$

Para implementar esta regla de parada, necesitamos un valor de M . Un enfoque conservador sería elegir un numero muy grande, sin embargo puede ser mejor elegir M de forma dinámico, es decir simplemente definiendo $M(x) = \bar{x}$.

Esto hace que la desigualdad sea precisa en la proposición 2.2 que es una desigualdad aproximada.

De lo anterior vemos $w(x).b$ es una estimación del valor óptimo de la función objetivo. Para el resto de esta sección se muestra que esto tiende a ser una mejor estimación que $c^T x$.

A efectos prácticos del algoritmo, puede ser útil para controlar el gap-dual $c^T x - b^T w$ junto con el grado de dual viable δ , se detendrá cuando ambos son suficientemente pequeños.

Capítulo 3

Metodología

En la metodología a seguir, mostraremos el lugar de aplicación del modelo de Karmarkar, seguido de la obtención de los datos y finalmente el procedimiento para su implementación.

3.0.1 Lugar de aplicación

Se aplicara en una empresa de transportes de pasajeros interurbano, que ofrece viajes diarios al sur chico del País, trasladando personas por diferentes razones tales como: educación, turismo, comercio, trabajo entre otros. La empresa preocupado por sus clientes trata de dar el mejor servicio, es por ello que ofrece seguridad, rapidez y limpieza en todos sus servicios disponibles.

La empresa cuenta con una flota de alta gama de 250 buses, distribuidos en 2 servicios, el primero es el servicio económico con 220 buses y el segundo el servicio diferenciado o también llamado servicio Vip que cuenta con 30 buses. La empresa cubre la ruta sur a norte y viceversa del tramo comprendido entre Lima e Ica, con paradas en Cañete, Chincha y Pisco, según sea el servicio tomado por el pasajero.

La empresa tiene como visión ser líder en el tramo Lima e Ica dando la mayor satisfacción a sus clientes nacionales e internacionales. Tiene a futuro un plan ambicioso de expansión a diferentes zonas del país, el servicio de pasajeros, transporte de personal y carga y encomienda.

Como se comento anteriormente la empresa cuenta con los servicios estándar y diferenciado, el servicio estándar se caracteriza por tener salidas frecuentes en días de alta demanda puede llegar a tener una frecuencia de hasta 5 minutos. Este servicio realiza paradas en cada agencia a lo largo del tramo principal.

En el caso del servicio diferenciado, son servicios directos de origen Lima con destino a Ica y viceversa. Las bondades del bus que realiza este servicio son: asientos con inclinación de 140°, baño a bordo, servicio de WIFI, Snack, climatización ecológica, cargador para celulares y asientos con pantalla táctil. Además cuenta con personal altamente calificado que hará su viaje más placentero.

También podemos mencionar que la empresa brinda el servicio de carga y encomienda, envío de sobre, giros con la rapidez y seguridad que nos diferencia de la competencia.

3.0.2 Obtención de información

Se cuenta con información registrada mediante ERP, gestionan la información mediante el motor de base datos SQL y los tiempos de recorrido de los buses se monitorea mediante un GPS instalado en cada bus. La distribución de viajes se realiza en una hoja excel, cargado la información de manera sincrónica a una tabla dinámica. En la figura 3.1 se muestra la programación ejecutada, la parte en blanco muestra los viajes en cada punto de origen y la parte coloreada muestra el promedio de pasajeros por cada viaje.

Existen 2 rumbos bien definidos en la el tramo Lima e Ica y viceversa.

BloqHoraProy	SUR										BloqHoraProy	NOR												
	OCUPACION.					TURNOS.						OCUPACION.					TURNOS.							
	LIMA	CHILCA	ASIA	CANETE	CHINCHA	PISCO	LIMA	CHILCA	ASIA	CANETE		CHINCHA	PISCO	ICA	PISCO	CHINCHA	CANETE	ASIA	CHILCA					
00	28	28	27	29	31	42	6	6	6	6	6	6	27	34	43	46	46	46	4	4	4	4	4	4
01	42	42	41	39	39	51	3	3	3	4	6	7	29	36	34	40	40	39	4	4	6	6	6	6
02	48	45	46	63	37	45	3	3	3	3	7	7	23	27	28	38	39	41	4	4	6	6	6	6
03	36	38	38	56	30	46	3	3	3	3	7	8	30	35	34	33	33	36	4	4	8	8	8	8
04	39	38	33	44	23	40	7	7	7	8	10	11	23	25	41	28	30	31	6	5	9	9	9	9
05	41	38	34	41	35	47	12	12	12	11	8	8	30	26	37	37	37	39	6	6	9	9	9	9
06	34	31	28	25	24	39	14	14	14	11	8	8	31	29	32	25	26	31	8	7	8	8	8	8
07	47	42	40	39	30	35	15	15	15	12	10	10	35	30	33	35	35	39	7	7	7	7	7	7
08	44	43	40	54	45	40	13	13	13	10	8	8	36	26	37	41	44	48	8	8	8	8	8	8
09	49	48	45	48	49	67	12	12	12	9	7	7	44	35	34	39	41	47	8	8	8	8	8	8
10	50	49	47	46	43	58	13	13	13	11	9	10	32	25	29	40	42	46	9	9	9	9	9	9
11	48	48	46	45	38	47	13	13	13	11	11	11	26	29	31	45	47	49	8	8	9	9	9	9
12	55	53	51	44	40	41	15	15	15	15	12	13	36	31	29	50	53	56	11	11	11	11	11	11
13	53	53	49	41	40	42	15	15	15	15	12	13	53	44	40	60	62	64	15	15	16	18	18	18
14	50	49	46	33	27	32	17	17	17	17	14	14	63	49	32	46	47	52	14	12	14	13	13	13
15	51	50	46	34	32	43	13	13	13	13	9	9	68	57	45	60	61	61	15	13	14	15	15	15
16	56	52	50	39	32	33	15	15	15	15	12	12	62	47	46	49	49	50	15	15	15	17	17	17
17	53	52	49	34	27	24	15	15	15	15	12	12	54	38	36	45	45	46	15	15	15	15	15	15
18	48	46	42	30	19	17	13	13	13	13	11	11	57	38	49	56	56	55	12	12	8	8	8	8
19	57	52	49	33	27	16	11	11	11	11	9	9	48	40	49	56	56	56	10	9	6	6	6	6
20	43	41	40	24	22	17	9	9	9	9	7	7	43	38	46	42	42	41	8	7	5	5	5	5
21	41	36	35	25	18	13	8	8	8	8	7	7	43	50	65	69	67	64	7	6	4	4	4	4
22	43	36	35	23	19	16	6	6	6	6	5	5	51	53	56	52	52	50	6	5	4	4	4	4
23	35	30	26	19	19	19	6	6	6	6	5	5	51	47	43	43	43	42	5	5	5	5	5	5

Figura 3.1: Programación de viajes ejecutadas en el tramo Lima e Ica y viceversa

Para leer esta tabla se debe considerar que la hora de bloque proyectada es del origen, es decir donde empieza la programación ya sea Lima o Ica, por ejemplo de hacemos la lectura de los vehículos que parten de Lima rumbo sur, si en Lima son las 00:00 horas cuando pase por cañete serán las 02:40 aproximadamente, al pasar por Chincha serán 03:40 aproximadamente, cuando pase por Pisco serán las 04:20 y al llegar a Ica punto final del recorrido serán las 05:20 aproximadamente.

Además podemos obtener la demanda potencial por día y bloque de hora de fechas donde queremos realizar algún cambio en la programación, actualmente estos cambios se realizan de forma empírica a juicio de experto. En la figura 3.2 muestra la distribución de la demanda en los diferentes puntos.

DEMANDA DE PASAJEROS POR ORIGEN DESTINO										
BLOQ HORA	LIMA				CANETE			CHINCHA		PISCO
	CANETE	CHINCHA	PISCO	ICA	CHINCHA	PISCO	ICA	PISCO	ICA	ICA
00:00	24	38	22	94	3	3	6	18	13	22
01:00	14	16	12	52	10	1	5	12	15	37
02:00	2	12	7	17	25	8	6	11	16	55
03:00	23	22	11	26	29	7	20	10	16	45
04:00	86	42	25	36	71	10	10	20	86	128
05:00	169	87	28	74	152	13	20	82	238	291
06:00	117	86	21	49	93	6	22	72	197	296
07:00	131	95	38	65	61	4	17	25	102	152
08:00	141	113	62	93	141	9	11	19	57	269
09:00	227	153	62	129	86	15	19	54	131	251
10:00	167	78	54	101	167	11	29	22	60	218
11:00	129	84	41	69	103	8	38	13	75	195
12:00	115	99	57	82	111	10	20	34	66	212
13:00	150	160	50	84	94	15	19	33	53	227
14:00	221	156	94	150	165	13	35	52	103	319
15:00	246	173	83	116	131	19	50	40	57	189
16:00	217	234	85	124	155	15	45	53	80	260
17:00	265	258	86	121	140	13	29	69	68	174
18:00	185	188	58	81	103	20	37	60	44	121
19:00	146	197	101	153	60	3	23	28	58	108
20:00	127	146	65	116	75	13	23	43	60	44
21:00	54	101	28	95	32	4	24	24	17	35
22:00	47	95	24	48	13	3	5	22	16	52
23:00	49	60	31	82	40	11	3	16	12	35

Figura 3.2: Distribución de la demanda por bloque de hora y origen destino (Fuente: Elaboración propia)

Con todos estos datos mostrados y la organización de la información, podemos generar una interfaces para poder minimizar los costos en una programación de buses. En este caso solo abordaremos el modelo en el rumbo Sur del recorrido y nos enfocaremos en satisfacer la demanda, por ahora no se considera en tiempo y la disponibilidad de buses en cada origen destino.

A continuación, explicamos el modelo así como la definición de las variables.

3.0.3 Explicación del modelo

La propuesta es realizar un modelo de programación lineal que satisfaga la demanda por bloque de hora y dividido en días de alta demanda y días de baja demanda, esta data utilizada se muestra en la figura 3.3 donde se puede apreciar las ciudades de Lima, Cañete, Chincha, Pisco e Ica.

DIA	HORA	LIMA	CANETE	CHINCHA	PISCO
1	0	88	119	207	306
1	1	44	98	214	234
1	2	42	96	210	232
1	3	167	240	272	340
1	4	91	122	210	309
1	5	244	298	414	434
1	6	775	465	286	221
1	7	990	675	468	384
1	8	616	504	350	520
1	9	585	456	310	410
1	10	566	454	300	470
1	11	599	304	231	351
1	12	584	289	216	336
1	13	291	162	55	141
1	14	379	343	287	411
1	15	584	289	216	336
1	16	506	394	240	410
1	17	456	344	190	360
1	18	344	398	314	334
1	19	177	220	282	350
1	20	165	225	272	315
1	21	167	240	272	340
1	22	139	235	248	340
1	23	114	210	223	315

Figura 3.3: Distribución de la demanda por bloque de hora y origen destino (Fuente: Elaboración propia)

VARIABLES DE DECISIÓN

x_{ij} : Cantidad de viajes programados del origen i al destino j

VARIABLES DE DEMANDA

D_{ij} : Demanda de pasajero del origen i al destino j

Variables de costo

c_{ij} : Costo de un viaje del origen i al destino j

Definición de la función objetivo

Como ya se explico anteriormente nuestro objetivo es minimizar los costos producidos por la programación de buses, pero satisfaciendo la demanda por bloque hora, entonces nuestra función objetivo estará dado por:

$$\min c^T x = c_{ij} * x_{ij}$$

Donde ij corresponden a los orígenes destinos donde se detecta la demanda. los costos son valores fijos calculados por la empresa, tomando en consideración gastos operacionales del servicio. Hay que tener en claro que esta aplicación se ejecuta en el servicio estándar que ya se hablo con anterioridad.

Origen-Destino	Costos
Lima-Cañete	288.12
Lima-Chincha	392.59
Lima-Ica	615.44
Cañete-Ica	328.3
Chincha-Ica	223.83
Piso-Ica	141.71

Figura 3.4: Tabla de costos (Fuente: Elaboración propia)

Restricciones

Debemos considerar como restricción que: la capacidad de un bus es de 50 pasajeros, pero por políticas de la empresa el aforo máximo que debería tener cada bus en recorrido es de 45.

Para las restricciones de no negatividad se debe tomar en cuenta que los viajes son números positivos, es decir $x_{ij} \geq 0$ con todo estos apuntes podemos generar nuestra ecuación para cada bloque de hora.

Formulación de las ecuaciones

Para los buses que partes de Lima, se tiene los destinos a Cañete, Chincha e Ica.

$$45 * x_{Lima-Canete} + 45 * x_{Lima-Chincha} + 45 * x_{Lima-Ica} \geq D_{Lima}$$

Para los buses que partes de Cañete, se tiene los destinos a Ica, pero tener en cuenta que los buses que vienen del anterior origen también puede satisfacer la demanda del actual punto, es decir:

$$45 * x_{Lima-Chincha} + 45 * x_{Lima-Ica} + 45 * x_{Canete-Ica} \geq D_{Canete}$$

Para el origen Chincha, se tiene los destinos a Ica y los buses que vienen de los orígenes anteriores, esto lo podemos plantear de la siguiente manera:

$$45 * x_{Lima-Ica} + 45 * x_{Canete-Ica} + 45 * x_{Chincha-Ica} \geq D_{Chincha}$$

Para el origen Pisco se tiene el destino Ica, de igual manera se tomara los bues que vienen de los orígenes anteriores, de esta forma:

$$45 * x_{Lima-Ica} + 45 * x_{Canete-Ica} + 45 * x_{Chincha-Ica} + 45 * x_{Pisco-Ica} \geq D_{Pisco}$$

De aqui se forma la matriz de restricciones y esto se aplicara para cada bloque de hora y día donde hagamos el análisis, entonces ya estamos en la capacidad de realizar la interfaces con el método de Karmarkar.

Capítulo 4

Implementación Computacional

Se utilizar el software Matlab, debido a su sencillez y rapidez en los cálculos matemáticos, como ya se vio en capítulos anteriores la formación del método de Karmarkar, mostraremos un resumen paso a paso para la formulación del algoritmo y sea más sencillo su programación computacional.

En la figura 4.1 se muestra en ambiente de trabajo de Matlab, tanto en la salida de consola y como el archivo m, donde realizamos nuestra programación conectadas a una base de datos en archivo excel.

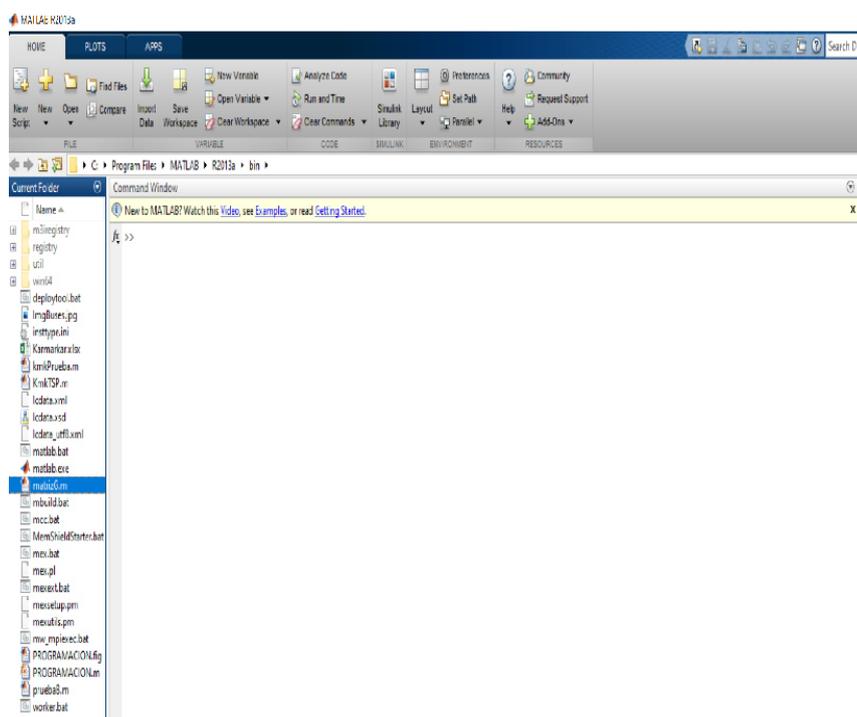


Figura 4.1: ambiente de trabajo (Fuente: Matlab)

Ahora presentamos el algoritmo modificado de Karmarkar, utilizando el escalado Afín a sus variables. Dejando de lado la transformación proyectiva más elaborada.

Se tiene $x^0 > 0$, $\alpha \in]0, 1[$, $\varepsilon \in]0, 1[$

Paso 1. Hacer $k = 0$

Paso 2. Definimos

$$D_k = \begin{pmatrix} x^k_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x^k_n \end{pmatrix}, \quad \tilde{A}_k = (AD_k)$$

$$\tilde{P}_k = I - \tilde{A}_k^T (\tilde{A}_k \tilde{A}_k^T)^{-1} \tilde{A}_k$$

$$\tilde{c}_{p_k} = \tilde{P}_k D_k c$$

$$d^k = \frac{\tilde{c}_{p_k}}{\max(e_i \cdot \tilde{c}_{p_k})}$$

$$w^k = (AD_k^2 A^T)^{-1} AD_k^2 c$$

Paso 3.

$$\tilde{x}^{k+1} = e - \alpha d^k$$

Donde $e \in \mathbb{R}^n$ vector de unos.

Paso 4. Regresando al espacio original

$$x^{k+1} = D \tilde{x}^{k+1}$$

Paso 5.

$$\text{si, } c^T x^{k+1} - b^T w^k < \varepsilon \quad \text{Paramos } x^{k+1} \text{ es solucion}$$

Paso 6.

Caso contrario, $k = k + 1$ y volvemos al Paso 2.

El anexo se muestra la codificación computacional, además mostraremos la interfaces que hace más dinámico la interacción con el usuario en este caso los analistas de la empresa en estudio.



Figura 4.2: Ejecución en interfaces para día 2 (día de alta demanda)

Aquí se nota la ejecución de una óptima distribución de buses por origen en bloques de hora y el costo total por enviar esa programación asciende a S/. 161'459.2113 por razones de confidencialidad no se puede dar el monto producido por un día de operación en el rumbo sur, pero podemos asegurar que sobrepasa convincentemente el gasto.

```
solo ingresar los numeros 1 ó 2
1-(dia baja demanada) 2-(dia alta demanda)
Ingrese el día de analisis: 2
  Hora      Lima      Cañete      Chincha      Pisco      Costo(s/.)
ans =
      0         6         4         3         5         3264
      1         7         4         1         3         2798
      2         4         5         6         8         3189
      3         3         3         5         7         2548
      4         6         4         1         3         2782
      5        13         7         5         8         6063
      6        13         6         5         7         5858
      7        17        10         6         6         7456
      8        22        19        14        14        11383
      9        22        19        16        19        12294
     10        22        20        15        18        12220
     11        22        15        10        10        10223
```

Figura 4.3: Ejecución en consola para día 2 (día de alta demanda)

También podemos apreciar la salida en consola que nos calcula los mismos montos. Es decir, se mantiene la buena distribución de viajes en ambos casos.

Conclusiones

1. Para cuestiones de la implementación del Algoritmo de Karmarkar se ha tomado una solución inicial factible, pero esto no siempre es trivial, para solventar esta dificultad se puede aplicar el método de la Gran M, esto nos dará un punto de inicio que cumpla con las condiciones necesarias para la aplicación.
2. Los métodos de punto interior a diferencia del símplex avanzan hacia la solución por el interior de la región factible, esto hace que su convergencia sea más rápido. Para programas de gran tamaño estos métodos encuentran la solución óptima realizando menor costo computacional.
3. A pesar de tener sencillos razonamientos matemáticos entre su estructura, este algoritmo presenta un excelente rendimiento en comparación con el Símplex.
4. El aplicativo sirve de gran ayuda para optimizar los procesos manuales que se usan, la empresa en estudio puede ganar horas hombre en otras actividades.

Recomendaciones

1. Es recomendable este trabajo de tesis para aquellos estudiantes e investigadores que inicien sus estudios en la línea Optimización Matemática.
2. Actualmente existen muchas variante al método de Karmarkar original, el cálculo de dirección de búsqueda es uno de los puntos más estudiados, se recomienda aplicar alguna de estas variantes bajo la modificación expuesta en este trabajo.
3. Se recomienda que antes de aplicar el algoritmo de Karmarkar verificar la data de ingresos, ya que debe cumplir con todas las consideraciones para su optima ejecución.
4. Se deja al lector la como mejora al problema, incluir la variable de tiempo y cantidad de recursos, que seria el siguiente paso de este trabajo.

Anexos

La programación del algoritmo de Karmarkar se desarrollo en Matlab 2013, el cual esta guardado con el nombre de PROGRAMACION.m, ademas para calcular el valor inicial se programo el método de la Gran M, que esta incluido en el código principal.

Archivo PROGRAMACION

```
function varargout = PROGRAMACION(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @PROGRAMACION_OpeningFcn, ...
                  'gui_OutputFcn',  @PROGRAMACION_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function PROGRAMACION_OpeningFcn(hObject, eventdata, handles, varargin)

%Colocar Imagen de Fondo
axes('Units','Normalized','Position',[0 0 1 1]);
x=imread('ImgBuses.jpg');
```

```

image(x),axis off

% Choose default command line output for PROGRAMACION
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = PROGRAMACION_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function edit2_Callback(hObject, eventdata, handles)

%Ingreso de Valor d
Val=get(hObject,'String');%almacenar el valor ingresado
NewVal = str2double(Val); %transformar el formato double
handles.edit2=NewVal;      %almacenar en identificador
guidata(hObject,handles);%salvar datos de la aplicación

function edit2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
c = [288.12;392.59;615.44;328.3;223.83;141.71;0;0;0;0];
A = [45 45 45 0 0 0 -1 0 0 0;0 45 45 45 0 0 0 -1 0 0;0 0 45 45 45 0 0 0 -1 0;
      0 0 45 45 45 45 0 0 0 -1];

d = handles.edit2;
if d == 1
    bb = xlsread('Karmarkar','Demanda','C2:F25');
elseif d == 2
    bb = xlsread('Karmarkar','Demanda','C26:F49');
end

MMM = zeros(24,6);
h =1;
while (h <= 24)
    b = bb(h,:);

    %calcular las dimensiones de los datos ingresados
    [m n] = size(A);
    [mc nc] = size(c);
    [mb nb] = size(b);
    %disp(' ')

%comprobar si los vectores c y b son vectores columna
if((nc ~= 1) | (nb ~= 1))
    fprintf('!!!REVISAR.. b y c deben ser vectores columnas\n\n')
elseif((m ~= mb)|(n ~= mc))
    fprintf('!!!REVISAR.. Dimensiones incompatibles para b, c y la matriz A\n\n')
else
    if(rank(A) ~= m)
        fprintf('!!!REVISAR.. La matriz no es de rango completo')
    else
        alpha = 0.5;
        epsilon = 10^-7;
        k = 0;

%Metodo del GranM para Obtener el Punto Inicial
M = 1000*max(abs(c));
c = [c ;M]; %ampliar el vector c
en = ones(n,1);
r = b - A*en; %calculo de infactibilidades
A = [A r]; %ampliar la columna A
x = [en ;1];
D = eye((n+1));
y = inv(A*D^2*A')*A*D^2*c;
%proceso de iteracion Karmarkar

```

```

while (abs(c'*x-b'*y)/(1 + abs(c'*x))) > epsilon
    z = c - A'*y;
    dx= D^2*z;
    ey = max(ones((n+1),1).*(D*z));
    x = x - (alpha/ey)*dx;
    k = k + 1;
    D= diag(x);
    y= inv(A*D^2*A')*A*D^2*c;
    %fprintf('%8.0f\t\t %8.1f\t %8.1f\t %8.1f\t %8.1f\t
    %8.1f\t %8.1f\t %8.1f\t %8.1f\t %8.1f\t %8.1f\t %8.1f\t
    %8.1f\n',k,x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),x(9),x(10),
    (abs(c'*x-b'*y)/(1 + abs(c'*x))),c'*x)
end
if(x(n+1) > 10^-7)
    fprintf('El problema es infactible\n')
else
    %fprintf('\n\n')
    %fprintf('%8.0f\t\t %8.0f\t %8.0f\t %8.0f\t %8.0f\t %8.2f\n',
    h-1,x(1)+x(2)+x(3),x(2)+x(3)+x(4),x(3)+x(4)+x(5),x(3)+x(4)+x(5)+x(6),
    c'*x)

    fila = h;
    for col = 1:6
        if col == 1
            MMM(fila,col) = h-1;
        elseif col == 2
            MMM(fila,col) = x(1)+x(2)+x(3);
        elseif col == 3
            MMM(fila,col) = x(2)+x(3)+x(4);
        elseif col == 4
            MMM(fila,col) = x(3)+x(4)+x(5);
        elseif col == 5
            MMM(fila,col) = x(3)+x(4)+x(5)+x(6);
        else
            MMM(fila,col) = c'*x;
        end
    end
end

end
end
end
h = h +1;
end
format shortg
set(handles.text6,'String',num2str(round(MMM)))
format bank
set(handles.text5,'String',num2str(sum(MMM(:,6))))

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

```

Bibliografía

- [1] ANSTREICHER, K. M.(1986). *A monotonic projective Algorithm for fractional linear programming.*
- [2] BARNES, E.R.(1985), *A variation on Karlmakar's algorithm for solving linear programming problems, Yorktown Heights*
- [3] BERTSIMAS D. AND TSITSIKLIS J.(1997),
Introduction to Linear Optimization, Athena Scientific, Belmont, USA
- [4] CASTRO, J.(2000). *Una Introducción Al Método Del Escalado Afín Para Programación Lineal, España, Statistics Operations Research Dept.UPC*
- [5] DANTZIG G. B.(1951), *Application of the Simplex Method to a Transportation Problem, Activity Analysis of Production and Allocation, Koopmans, T.C., Ed., John Wiley and Sons,New York*
- [6] GONZALES, M.(1991). *El Método De Karmarkar: Un Estudio De Sus Variantes. España: Universidad de la Laguna*
- [7] GUTIERREZ, M.(1986), *Estudio hidrogeológico detallado del valle Caplina - La Yarada, Universidad Nacional Agraria, Facultad de Ingeniería Agrícola, tesis para optar el título de Ingeniero Agrícola. Lima.*
- [8] KHACHIVAN, L. G.(1980), *Polinomial Algorithm i Linear Programing, U.R.S.S. Computational Mathematics and Mathematical Physics.*
- [9] KARMARKAR, N.(1984), *A new polynomial-time algorithm for linear programming, Laboratories Bell, USA*
- [10] SALINAS J. F.(2010), *Estudio Del Algoritmo Proyectivo De Karmarkar Y Sus Aplicaciones En La Ingeniería, Universidad Nacional De Ingeniería, Facultad de Ciencias, Perú*
- [11] VANDERBEI, R.J.,MEKETON M.S.,FREEDMAN B.A.(1985),
A Modificación Of Karmarkar Linear Programming Algorithm, Laboratories, Holmdel, USA
- [12] VANDERBEI R.J., *Linear Programming: Foundations and Extensions Kluwer Academic Publishers, Boston, USA.*