



UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
FACULTAD DE INGENIERÍA ELÉCTRICA

“SOLUCIÓN AL PROBLEMA DEL DESPACHO ECONÓMICO EN
SISTEMAS ELÉCTRICOS DE POTENCIA USANDO EL MÉTODO
SIMPLEX”

TESIS

Que para obtener el título de Ingeniero en Computación

Presenta
EDUARDO TAPIA LÓPEZ

Director de Tesis
DR. ANTONIO RAMOZ PAZ

Codirector de Tesis
DRA. VIOLETA MEDINA RIOS

Agosto de 2018

Agradecimientos

Quiero dar las gracias a mis asesores que me apoyaron para la realización de esta tesis a la Dra. Violeta Medina Ríos y al Dr. Antonio Ramos Paz que me han apoyado a lo largo del desarrollo de esta tesis. A mis padres Juan Tapia García y Ma. Guadalupe López Tapia que siempre han estado para apoyarme en todo momento a lo largo de mi travesía en la universidad. Dándome su soporte incondicional en todos los momentos, difíciles y fáciles para poder llegar a obtener mi título. Les doy las gracias por todo su apoyo y paciencia para poder llegar hasta este punto en mi vida.

Dedicatoria

Quiero dedicar esta tesis a mis padres Juan Tapia García y Ma. Guadalupe López Tapia quienes fueron un gran soporte durante el tiempo en que escribía esta tesis, quienes me apoyaron todo el tiempo y me animaban a continuar sin dar un paso atrás.

A mis amigos quienes me apoyaron y alentaron para continuar, cuando parecía que me iba a rendir.

A mis maestros quienes nunca desistieron al enseñarme, aún sin importar que me costaba trabajo en comprender algunos temas, a ellos, a los que me enseñaron y me incitaron a continuar con mi aprendizaje.

A mis asesores la Dra. Violeta Medina Ríos y al Dr. Antonio Ramos Paz, que depositando su esperanza y confianza y por todo su apoyo a pesar de tener muchas labores se dieron el tiempo para que pudiera terminar esta tesis.

A todos aquellos que me apoyaron para escribir y concluir esta tesis.

Para todos ellos es esta dedicatoria.

Índice

Agradecimientos	ii
Dedicatoria.....	ii
Índice.....	iii
Resumen	v
Abstract.....	v
Lista de Símbolos y Abreviaturas	vi
Lista de Figuras.....	vii
Lista de Tablas.....	viii
Palabras Clave	ix
Capítulo 1. Introducción.	1
1 Introducción	1
1.1 Antecedentes	3
1.2 Objetivos	6
1.2.1 General	6
1.2.2 Particulares.....	6
1.3 Justificación	7
1.4 Metodología.....	8
1.5 Descripción de los capítulos.....	8
Capítulo 2. Despacho económico.	10
2.1 Introducción al problema de despacho económico	10
2.2 Métodos de solución del problema de despacho económico	14
2.2.1 Búsqueda Binaria.....	14
2.2.2 Iteración Lambda	16
2.2.3 MÉTODO PL	19
2.3 Conclusiones	21
Capítulo 3. Método simplex.....	22
3.1 Programación lineal	22
3.2 Forma estándar	22
3.3 Solución Gráfica.....	24

3.4 El método simplex.....	29
3.5 Implementación del método simplex	39
3.6 Conclusiones	43
Capítulo 4. Aplicación del método simplex en la solución de un despacho económico.....	44
4.1 Propuesta de solución al problema de DE usando programación lineal y MATLAB...	44
4.1.1 Propuesta de solución al problema de DE usando programación lineal y MATLAB	46
4.2 Propuesta de solución al problema de DE usando búsqueda binaria y MATLAB.....	49
4.3. Propuesta de solución al problema de DE usando la iteración lambda y MATLAB...	54
4.5 Conclusiones	59
Capítulo 5. Casos de Estudio.....	60
5.1 Introducción	60
5.2 Caso de Estudio No. 1.....	60
5.2 Caso de Estudio No. 2.....	64
5.3 Conclusiones	69
Capítulo 6. Conclusiones y recomendaciones para trabajos de investigación futuros.	70
6.1 Conclusiones	70
6.2 Recomendaciones para trabajos futuros	71
Referencias	73

Resumen

En la presente tesis se describirá de manera detallada el modelo de despacho económico en sistemas eléctricos de potencia usando el método simplex y a su vez siendo comparando con dos métodos más que son la búsqueda binaria y la iteración lambda. Todos estos métodos están programados en Matlab y se muestra cual lleva a una mejor solución, en los siguientes capítulos se presenta el modelo de despacho económico, se ejemplificara así como también cada uno de los métodos antes mencionados. Se muestra la formulación, sus procedimientos y que pasos se siguen en cada uno de los métodos. En los ejemplos y casos de estudio se ha observado que el método de iteración lambda llega una mejor solución con respecto a los otros dos métodos. Tiene una buena precisión para un mejor resultado así como también es muy rápido en milésimas de segundo en su tiempo de ejecución, llevando a cabo muy pocas iteraciones en la resolución del problema de Despacho Económico (DE). La iteración lambda demostró ser un el mejor método en esta presente tesis. No fue el mejor en los dos casos, la búsqueda binaria, sin embargo, entrega buenos resultados, pero ejecutando un mayor número de iteraciones y su tiempo de ejecución es mayor que el de la iteración lambda. Los resultados obtenidos fueron buenos, se llegó a una solución óptima para cada caso en esta presente tesis, tomando a la iteración lambda como el mejor método.

Abstract

In this thesis we will describe in detail the model of economic dispatch in electric power systems using the simplex method and in turn being compared with two more methods that are the binary search and the lambda iteration. all these methods are programmed in Matlab and it is shown which leads to a better solution, in the following chapters the model of economic dispatch is presented, it will be exemplified as well as each of the aforementioned methods. The formulation, its procedures and what steps are followed in each of the methods are shown. In the examples and case studies it has been observed that the lambda iteration method arrives at a better solution with respect to the other two methods. It has a good precision for a better result as well as it is very fast in thousandths of a second in its execution time, carrying out very few iterations in the resolution of the Economic Dispatch (DE) problem. The lambda iteration proved to be the best method in this thesis. It was not the best in both cases, the binary search, however, delivers good results, but executing a greater number of iterations and its execution time is greater than that of the lambda iteration. The results obtained were good, an optimal solution was reached for each case in this thesis, taking the lambda iteration as the best method.

Lista de Símbolos y Abreviaturas

Despacho Económico	DE
<i>Economic Load Dispatch</i>	ELD
<i>Estimation of Distribution Algorithms</i>	EDA
Flujo de Potencia Óptimo	FPO
<i>Genetic Algorithms</i>	GA
<i>Particle Swarm Optimization</i>	PSO
Programación Lineal	PL
Sistema Eléctrico de Potencia	SEP
<i>Split-On-Demand</i>	SOD

Lista de Figuras

Figura 2.1. N unidades térmicas comprometidas para servir una carga de P_{carga}	10
Figura 2.2. Diagrama búsqueda binaria.	15
Figura 2.3. Despacho económico por el método de iteración lambda.	16
Figura 2.4. Solución gráfica al despacho económico.	17
Figura 2.5. Proyecciones lambda.	18
Figura 2.6. Característica de función de costo no lineal.	19
Figura 2.7. Función de costo no lineal aproximada por segmentos de línea recta.	19
Figura 3.1. Solución gráfica del problema de programación lineal.	26
Figura 3.2. Alternativa óptima.	28
Figura 3.3. Solución no factible.	28
Figura 3.4. Resultado no acotado.	29
Figura 3.5. Diagrama del método simplex.	35
Figura 3.6. Gráfica para llegar al óptimo de manera eficiente.	40

Lista de Tablas

Tabla 3.1. Recursos Disponibles	23
Tabla 3.2. Tipos de desigualdad.....	33
Tabla 3.3. Tabla del método simplex construcción.	34
Tabla 3.4. Valores de punto extremo.	38
Tabla 3.5. Valores iniciales.....	40
Tabla 3.6. Renglón pivote.	41
Tabla 3.7. Eliminación de X1.	42
Tabla 3.8. Nuevos valores.....	42
Tabla 3.9. Eliminación Gauss-Jordán.	42
Tabla 4.1. Límites max y min.....	45
Tabla 4.2. Segmentos costo valor.	45
Tabla 4.3. Código del método simplex en MATLAB.....	46
Tabla 4.4. Función ms5 del código de MATLAB.....	48
Tabla 4.5. Resultados obtenidos en Matlab para PL por partes.....	48
Tabla 4.6. Costos incrementales.....	49
Tabla 4.7. Código de la búsqueda binaria en MATLAB.....	52
Tabla 4.8. Resultados de la búsqueda binaria Ejemplo 2.1 con MATLAB.....	53
Tabla 4.9. Código de la iteración lambda en MATLAB.....	57
Tabla 4.10. Resultado de la iteración lambda del Ejemplo 2.1 con MATLAB.....	58
Tabla 4.11. Resultados óptimos de los tres métodos.....	58
Tabla 5.1. Costos incrementales para el Caso de Estudio 1.	61
Tabla 5.2. Resultado de la búsqueda binaria para el caso de estudio 1.....	62
Tabla 5.3. Resultado de la iteración lambda para el Caso de estudio 1.....	63
Tabla 5.4. Resultados con Método Simplex para Caso de Estudio 1.....	63
Tabla 5.5. Resultados de los tres métodos para el Caso de Estudio 1.	64
Tabla 5.6. Costos incrementales para el Caso de Estudio 2.	66
Tabla 5.7. Resultados con búsqueda binaria para Caso de Estudio 2.	66
Tabla 5.8. Resultados con iteración lambda para Caso de Estudio 2.	67
Tabla 5.9. Resultados con Método simplex para Caso de Estudio 2.	68
Tabla 5.10. Resultados de los 3 métodos para el Caso de Estudio 2.....	68

Palabras Clave

DE: Despacho Económico.

SEP: Sistemas Eléctricos de Potencia.

PL: Programación Lineal.

Método Simplex

Capítulo 1. Introducción.

1 Introducción

Debido a que la crisis de energía es un problema previsible para el futuro, en especial por la escasez de petróleo y otros recursos naturales, los estudios de Sistemas Eléctricos de potencia (SEP) se han vuelto más y más importantes. Entre todos los aspectos de sistemas de potencia, que incluyen generación, reserva, distribución, transmisión, etc., el problema de Despacho Económico (DE), es una función importante en la operación de los sistemas de potencia. El despacho económico se refiere al problema de asignar apropiadamente entre los generadores para satisfacer las restricciones específicas, tales como la mínima y máxima salida de cada generador, así como satisfacer una demanda de potencia dada [Chao, 2007] y producir electricidad con los costos financieros y ambientales más bajos [Ying-ping, 2010]. El despacho económico es un problema central en la operación de micro-redes, que apunta a programar efectivamente varias fuentes de energía para minimizar el costo de operación mientras se satisface la demanda de electricidad [Keles, 2007].

El problema de generación de energía con hidrocarburos y con energía renovable es un problema bien conocido y estudiado por la ciencia. Las generaciones de energía distribuidas que se alimentan de combustibles económicos limpios están emergiendo fuentes prometedoras de energía para sistemas de cómputo de escala extrema [Yiqing, 2015].

El despacho económico es de gran ayuda para solventar los compromisos de una unidad generadora [Callaway, 2011]. La demanda de electricidad en un sistema de energía varía a lo largo del día, siguiendo patrones que dependen, entre otras cosas, de las características regionales, la temperatura, la hora del día, el día de la semana y la estación del año. Las decisiones de cambiar la salida del generador para acomodar la variación en las escalas de tiempo por hora generalmente se hacen por procesos de compromiso de unidad y despacho económico. El compromiso de la unidad establece los cronogramas de operación del generador antes del tiempo de operación y toma en cuenta las capacidades de rampa del generador y costos de arranque y de paro.

Las soluciones se obtienen mediante un proceso de optimización de varios períodos, como programación dinámica, relajación de Lagrange o programación de enteros mixtos. El compromiso de la unidad determina establecer generadores en línea y fuera de línea, y por lo general se ejecuta con un día de anticipación. El despacho económico es el proceso de elegir los niveles de salida para los generadores que ya están en línea, con el objetivo de minimizar el costo total de satisfacer la demanda.

El despacho económico tiende a ser bastante rápido y se puede ejecutar en minutos del tiempo de operación. Ambos procesos requieren pronósticos de demanda [Wordpress, 2013].

El modelo de despacho económico resuelto a través del método simplex es una técnica que ya ha sido implementada en trabajos anteriores. Sin embargo en la presente Tesis se resuelve el problema del despacho económico por medio de la búsqueda binaria, la iteración lambda y el método simplex para llegar a una solución más óptima con los recursos disponibles, tomando en consideración variables de restricción o si existen márgenes de error. Los problemas de despacho y coordinación en un Sistema Eléctrico de Potencia (SEP) buscan determinar los niveles de producción de las centrales del sistema que permitan abastecer la demanda al menor costo posible, respetando las limitaciones técnicas impuestas por el modelo eléctrico utilizado. Este modelo es multinodal cuando representa explícitamente la red de transmisión, incorporando sus restricciones. Al resolverlo, se obtiene una solución más apropiada para el problema real e información relevante desde el punto de vista económico. Una vez que se obtiene los datos tendremos los valores más óptimos para tener un menor costo en el precio de la energía. Una adecuada cuantificación de las pérdidas de transmisión con sus magnitudes y su consecuencia en las congestiones en la red. En los modelos de despachos económicos se debe tomar en cuenta el análisis de los costos, que la operación del sistema requiere, una supervisión del servicio para obtener un menor costo posible. Estos costos pueden ser controlados por los operadores del SEP y depende de la confiabilidad de todos los componentes del SEP, el tipo de combustible usado en el control de las pérdidas. El modelo del despacho económico puede ser resuelto a través de diferentes métodos. Los resultados de dichos métodos pueden ser comparados entre ellos y así saber cuál es más eficiente para ser elegido como la mejor opción para resolver el problema del despacho económico.

La optimización en la planificación y operación de los sistemas eléctricos, repercute inmediatamente en la economía de su funcionamiento. El uso eficiente del combustible disponible crece día a día en importancia ya que la mayoría de los combustibles empleados son del tipo no renovables.

El desarrollo de los despachos económicos se basa en el análisis de los condicionamientos económicos que afectan a un determinado sistema eléctrico para definir la combinación óptima de generadores en un instante concreto para una demanda particular. Se prescindirá de los datos e influencia de la red que se obtengan en cada caso para el cual se implemente [Wordpress, 2013].

El despacho económico consiste, por tanto, en que para una demanda dada consideramos qué generación tenemos, atendiendo o no a las pérdidas del sistema, declarando unos costes de producción de las centrales o los generadores y, a partir de

ello, perseguimos la mejor configuración de la generación posible desde el punto de vista económico para así obtener los mejores resultados o el mejor resultado positivo.

Por todo ello, se trata de una optimización donde se considera una central única dentro de la red la que reparte energía a una serie de subestaciones, que son absorbidas en cada una de ellas por las demandas que solicita cada subestación sin desperdiciar recursos.

1.1 Antecedentes

Los modelos de despacho económico son utilizados esencialmente en mercados eléctricos para obtener una mejor optimización y distribución de la energía manteniendo bajos los costos de los recursos que se utilizan y así tener una mayor producción, para llegar a la meta de tener menores costos y una mayor producción se realizan por diferentes métodos como es el del método simplex. A través de los años se han analizado los diferentes casos de despacho económicos que se han publicado en su mayoría para sistemas eléctricos los siguientes casos:

Método de solución para el despacho económico en línea considerando restricciones y reglas de un mercado eléctrico, se presentan dos métodos para resolver un despacho económico que considere restricciones de seguridad, con el objetivo de resolver un despacho económico en ventanas de tiempo cercanas a tiempo real. El primer método resuelve un despacho económico para un solo período de tiempo. Este método es muy rápido y se utiliza principalmente el método simplex y los multiplicadores de LaGrange para su solución, incluyendo restricciones de sobrecarga de líneas y transformadores y contingencia de los mismos. El segundo método resuelve un despacho económico para cuatro períodos de tiempo, cada uno de 10 minutos. Se resuelve en menos de dos minutos e involucra restricciones de sobrecarga de líneas, transformadores, contingencias, arranque de plantas térmicas, rampas entre otras. Se utiliza un método iterativo para el ajuste de las restricciones [Berrio Kevin, 2016].

El despacho económico de un sistema termo eléctrico a corto plazo, comprende la problemática actual del suministro eléctrico, así como las diferentes técnicas de solución para el problema de despacho de generación. Emplear una metodología que nos permite a resolver el problema de despacho de generación para unidades termoelectricas con el fin de minimizar costos de operaciones de estas centrales, la solución al despacho económico resolviéndolo por los multiplicadores de Lagrange y realizando un programa en lenguaje fortran [Flores Silva et al., 2008].

Despacho económico para sistemas medianos se confeccionará un cuestionario para levantar los requerimientos técnicos y operacionales del sistema, definir soporte

tecnológico para implementar modelo de despacho económico e implementar un modelo de despacho económico en software [Garrido Yerco, 2016].

Despacho de energía en mercados competitivos. La reestructuración de la industria eléctrica en muchos países ha propiciado el desarrollo de mercados competitivos con el objetivo de reducir precios de la energía eléctrica e introducir innovación en tecnología, el trabajo detalla y evalúa una formulación del problema de despacho de energía considerando agentes intermediarios. El trabajo desarrollado utiliza el Toolbox de Optimización de Matlab, un paquete de optimización muy versátil ya que puede manejar cualquier problema de Programación Lineal (PL) o no lineal. También se presenta el análisis de un mercado competitivo donde los participantes proponen ofertas para la compra y venta de energía [Larez Adán, 2003].

Despacho económico de cargas en sistemas eléctricos de potencia. Se plantea la puesta en marcha de un modelo de red eléctrica optimizando su funcionamiento desde el punto de vista de los costes de generación y explotación. Se acota al estudio de aquellos sistemas de generación eléctrica mediante centrales térmicas exclusivamente, se diseña y pone en marcha un modelo de un sistema eléctrico con múltiples grupos de generación [Díaz Carla, 2017].

Despacho económico de carga considerando restricciones en la red de transporte con el uso de técnicas de programación lineal. En la operación de un sistema eléctrico es deseable minimizar el costo de generación, lo cual requiere información de las plantas generadoras que relacione el costo de la fuente de energía primaria con la potencia neta de salida del generador. Utilizando un sistema de ecuaciones lineales se puede obtener una solución con un número finito de pasos y no se tienen problemas de convergencia de las formulaciones no-lineales, se utiliza una hoja electrónica de Excel y se apoya con una de las herramientas de optimización solver que se encuentra en Excel así se resuelve el despacho económico con apoyo de esta herramienta [Leal Elvis, 2005].

La programación de generación inteligente (incluyendo compromiso de la unidad y despacho económico) es el proceso de programar diferentes recursos de generación para minimizar costo mientras se satisfacen las restricciones físicas de un sistema eléctrico. En muchas ocasiones es un problema no lineal, y usualmente resuelto usando programación genética u otras técnicas de optimización no-convexa. El despacho económico convencional, una metodología muy bien investigada, es típicamente conducida con 24 horas de anticipación (un día adelante sin conexión) y se usa el hecho de que la carga del sistema puede ser razonablemente bien predicha un día antes. Sin embargo en una micro red con altos niveles de penetración de viento, esto ya no es válido debido a la naturaleza intermitente e impredecible de la energía del viento (la cual puede ser predicha con confiabilidad solo unos minutos antes [Monteiro, 2009]).

El problema del despacho económico ha sido resuelto por diferentes métodos. Un método de discretización adaptativa, llamado split-on-demand (split-on-demand, SoD) [Ying-ping, 2010], permite la estimación de algoritmos de distribución (Estimation of Distribution Algorithms, EDA) para obtener variables discretas para resolver problemas continuos de optimización. SoD divide aleatoriamente un intervalo continuo si el número de puntos de búsqueda dentro del intervalo excede un umbral, que se reduce en cada iteración. Después de la operación de división, a los intervalos no vacíos se les asignan códigos enteros, y los puntos de búsqueda se discretizan en de acuerdo a lo anterior. En [Ying-ping, 2010] se concluye que ECGA con SoD funciona bastante bien en el problema del despacho económico y ofrece soluciones mejores que otros métodos existentes. En [Thanos, 2013], se propone un marco de simulación de escala múltiple adaptativo basado en datos dinámicos (RT-DDDAMS) para el despacho eficiente de electricidad en tiempo real. El marco incluye: 1) Un procedimiento de descubrimiento donde la red se divide en subredes y se identifican posibles fidelidades, 2) Una plataforma RT-DDDAMS que incluye algoritmos para estimar el estado, selección de fidelidad y optimización multiobjetivo junto con un sistema simulación; y 3) Bases de datos para almacenar topologías de subred, fidelidades y mediciones selectivas. En [Lubin, 2011] se presenta una solución híbrida paralela PIPS que se basa en métodos de punto interior y utiliza una técnica complementaria de Schur para obtener una descomposición basada en el escenario del álgebra lineal. El PIPS se aplica a un problema de envío económico estocástico que utiliza pronósticos de viento por hora y un modelo detallado de flujo de potencia física. La solución de este problema es necesaria para la integración eficiente de la energía eólica con la red eléctrica de Illinois y el mercado de energía en tiempo real. En [Zwe-Lee, 2003] se propone un método de optimización de enjambre de partículas (Particle Swarm Optimization, PSO) para resolver el problema del despacho económico (DE) en los sistemas de energía. Muchas características no lineales del generador, tales como los límites de la tasa de incremento, la zona de operación prohibida y las funciones de costo no uniforme se consideran utilizando el método propuesto en la operación práctica del generador. La viabilidad del método propuesto se demuestra para tres sistemas diferentes, y se compara con el método Algoritmos Genéticos (Genetic Algorithms, GA) en términos de la calidad de la solución y la eficiencia de la computación.

La programación evolutiva se ha convertido en una herramienta de optimización útil para manejar problemas de programación no lineal. En [Sinha, 2003] se propone la aplicación de esta herramienta para solucionar el problema del despacho económico de carga (Economic Load Dispatch, ELD).

Los métodos que a continuación se mencionan serán los usados a lo largo de esta tesis.

Método simplex

Este método se basa en la suposición de que la solución óptima se encuentra en un punto extremo. Es un método numérico que busca un mínimo o máximo local de una función cualquiera examinando en cada paso los vértices de un polígono. Este procedimiento debe ser capaz de discriminar si durante la solución del problema se presenta un punto extremo. Para esto las ecuaciones con restricciones se formulan como igualdades y se introducen las variables de holgura.

Iteración Lambda

Se recomienda utilizar una lambda " λ " inicial que sea el promedio de los lambdas de los involucrados calculados en base a valores asignados de tal manera que estos suplan la demanda, esto con el objeto de iniciar con un lambda cercana a la lambda óptima. Se repite hasta que se cumpla la condición de equilibrio de potencia para una tolerancia especificada, sino se cumple esta condición ajustar lambda.

Búsqueda binaria

Esta búsqueda es similar a la iteración lambda. El valor delta lambda cambia a la mitad en cada iteración. La búsqueda binaria funciona con funciones de costo incrementales lineales por partes e incluso funciones de costo incremental lineal por partes que tienen saltos entre segmentos lineales.

1.2 Objetivos

Los objetivos principales para esta tesis a nivel licenciatura son los siguientes:

1.2.1 General

Resolver el despacho económico por el método simplex, observar que ventajas y desventajas tiene en comparación a los otros métodos, si es eficaz medir tiempos de ejecución, verificar cumplimiento de y cálculo de costos de producción.

1.2.2 Particulares

- Resolver el modelo de despacho económico, aplicado a los diferentes problemas del presente utilizando el método simplex.
- Programar el despacho económico utilizando la programación lineal, la búsqueda binaria y la iteración lambda.

- Comparar los diferentes métodos para resolver el despacho económico y determinar cuál es el que proporcionara una solución óptima en menor tiempo y menos costo.

1.3 Justificación

Hay una necesidad de tener una energía eléctrica eficiente confiable y que tenga un bajo costo, se ha tenido que buscar nuevas maneras de generar más cantidades de energía con nuevas tecnologías a un alto precio, pero para eso hay varios métodos diferentes para saber que método utilizar y tener energía de calidad y confiable a un bajo costo de producción, así como por medio de algunas herramientas computacionales programar modelos para minimizar aún más los costos y compararlos.

En esta tesis se analiza el modelo de despacho económico solucionado por el método simplex y comparando resultados con el método de búsqueda binaria y con el método de iteración lambda.

Se elige la solución del método simplex porque tiene una gran facilidad para comprenderse y pueda ser aplicado para la solución de problemas del mundo real. Este trabajo es relevante porque se resolverá un problema del despacho económico con el método simplex y este pueda ser de gran utilidad en el futuro para la solución de problemas similares. Se podrá analizar cuál es el método más eficiente, si es menos costoso y rápido en comparación con otros métodos y así elegir el que nos dé una solución más óptima aplicada de una red eléctrica, podría servir para investigaciones futuras y no se inicien desde cero, ya que tengan un punto de partida.

Con la creciente conciencia ambiental y con las nuevas directivas que el gobierno de varios países ha instaurado en cuanto a generación de energía, se ha preparado el escenario para un incremento en la fracción de energía suministrada usando recursos renovables [Narayanaswamy, 2012].

A medida que aumenta la demanda de electricidad; su generación es más costosa e ineficiente, los costos de generación pueden llegar a ser tan altos que los costos de oferta exceden el precio minorista en un orden de magnitud o más. Por lo tanto, durante varias décadas, muchas empresas de servicios han mantenido la infraestructura para reducir la carga de electricidad (especialmente acondicionadores de aire y calentadores de agua) para reducir la carga en lugar de despachar generación adicional durante los períodos de demanda muy alta [Callaway, 2011]. Lo anterior, con la intención de reducir los costos operativos, manteniendo el sistema en un estado operativo aceptable.

Los desafíos centrales asociados con controlar directamente las cargas eléctricas para contener los costos de generación son los siguientes:

1) La potencia total y la energía disponible para el control es limitado por la necesidad obvia de servir a la función primaria de uso final de la carga;

2) Podría haber un arranque posterior al control (o pico de recuperación) en la carga que resulta de la operación continua de cargas previamente controladas a medida que recuperan su estado operativo deseado (por ejemplo, punto de ajuste de temperatura). En algunas circunstancias, el pico de recuperación puede causar que la carga total exceda la que se hubiera producido en ausencia de control.

1.4 Metodología

La metodología que se seguirá para el desarrollo de esta Tesis es de la siguiente manera:

1. Revisión al problema de despacho económico. Revisión literaria del tema, se dará solución del problema del despacho económico utilizando diferentes métodos.

2. Análisis de los diferentes métodos empleados para resolver DE. Los pasos a seguir por cada método.

3. Programación de diferentes métodos para resolver el problema de DE. Se programara en MATLAB los diferentes métodos de programación lineal, búsqueda binaria e interacción lambda y método simplex para el problema de despacho económico.

4. Se analizara, se compararan y se determinara cuál es el método que entregue los mejores resultados obtenidos y que cumplen las necesidades de nuestro problema para obtener la solución más óptima.

1.5 Descripción de los capítulos

En el capítulo 1, se describen, antecedentes así como los objetivos generales y particulares de la presente tesis, la justificación de por qué se eligió el tema y que alcance puede llegar a tener, así como una descripción los métodos que se utilizaran.

En el capítulo 2. Se hace la introducción al problema del despacho económico, los métodos de solución que se utilizan como son la búsqueda binaria y la interacción lambda.

En el capítulo 3. En este capítulo se abordará el tema del método simplex, su descripción, los pasos a seguir que se deben tener en cuenta a la hora de abordar un problema y algunos ejemplos.

En el capítulo 4. Se presenta la solución al despacho económico por los diferentes métodos de programación lineal, propuestos como: búsqueda binaria e interacción lambda. Todos ellos programados en MATLAB y comparados entre sí.

En el capítulo 5. Se hace una descripción detallada las observaciones y resultados que se obtuvieron en cada uno de los casos de estudio.

En el capítulo 6. Se presentan los resultados obtenidos con relación a los métodos utilizados y el alcance que puede a llegar a tener este trabajo en el futuro.

Capítulo 2. Despacho económico.

En este Capítulo se describirán cada uno de métodos que se usaron en la presente tesis se podrán ver las ecuaciones que se utilizan así como gráficas y diagramas de cada uno de los métodos.

2.1 Introducción al problema de despacho económico

La Figura 2.1 muestra un sistema consta de N unidades generadoras de calor conectadas a una barra de bus única que sirve una carga eléctrica recibida P_{carga} . La entrada a cada unidad, que se muestra como F_i , representa la tasa de costo de la unidad. La salida de cada unidad, P_i , es la potencia eléctrica generada por esa unidad en particular. La tasa de costo total de este sistema es, por supuesto, la suma de los costos de cada una de las unidades individuales. La restricción esencial para el funcionamiento de este sistema es que la suma de las potencias de salida debe ser igual a la demanda de carga.

Matemáticamente hablando, el problema puede expresarse de manera muy concisa. Es decir, una función objetivo, F_T , es igual al costo total para suministrar la carga indicada. El problema es minimizar F_T sujeto a la restricción de que la suma de las potencias generadas debe ser igual a la carga recibida. No toman se tienen en cuenta las pérdidas de transmisión y que no se establecen explícitamente los límites operativos al formular este problema. Es decir:

$$F_T = F_1 + F_2 + F_3 + \dots + F_N$$
$$\sum_{i=1}^N F_i(P_i) \quad (2.1)$$

$$\emptyset = 0 = P_{carga} - \sum_{i=1}^N P_i \quad (2.2)$$

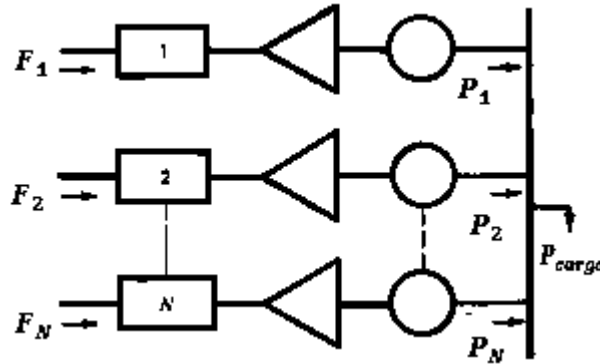


Figura 2.1. N unidades térmicas comprometidas para servir una carga de P_{carga}

Este es un problema de optimización restringido que puede ser atacado formalmente usando métodos avanzados de cálculo que involucran la función de Lagrange. Para establecer las condiciones necesarias para un valor extremo de la función objetivo, agregar la función de restricción a la función objetivo después de que la función de restricción se haya multiplicado por un multiplicador indeterminado. Esto se conoce como la función de Lagrange y se muestra en Ecuación. 2.3

$$\mathcal{L} = F_T + \lambda \Phi \quad (2.3)$$

Las condiciones necesarias para un valor extremo de la función objetivo resultan cuando tomamos la primera derivada de la función de Lagrange con respecto a cada una de las variables independientes y establecemos que las derivadas son iguales a cero. En este caso, hay $N + 1$ variables, los N valores de potencia de salida, P_i , más el multiplicador de Lagrange indeterminado, λ . La derivada de la función de Lagrange con respecto al multiplicador indeterminado simplemente devuelve la ecuación de restricción. Por otro lado, las ecuaciones N que resultan cuando se toman la derivada parcial de la función de Lagrange con respecto a los valores de salida de potencia uno a la vez dan el conjunto de ecuaciones que se muestran como en la Ecuación (2.4).

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial P_1} &= \frac{dF_1(P_1)}{dP_1} - \lambda = 0 \\ 0 &= \frac{dF_1}{dP_1} - \lambda \end{aligned} \quad (2.4)$$

Es decir, la condición necesaria para la existencia de una condición de operación de costo mínimo para el sistema de energía térmica es que las tasas de costo incremental de todas las unidades sean iguales a algún valor indeterminado λ . Por supuesto, a esta condición necesaria se debe agregar la ecuación de restricción que la suma de las salidas de potencia debe ser igual a la potencia demandada por la carga. Además, hay dos desigualdades que deben cumplirse para cada una de las unidades. Es decir, la potencia de salida de cada unidad debe ser mayor o igual que la potencia mínima permitida y también debe ser menor o igual a la potencia máxima permitida en esa unidad en particular. Estas condiciones y desigualdades se pueden resumir como se muestra en el conjunto de ecuaciones que componen la Ecuación. (2.5).

$$\begin{aligned} \frac{dF_1}{dP_1} &= \lambda \quad N_{gen} \text{ ecuaciones} \\ P_{imin} &\leq P_i \leq P_{imax} \quad 2N_{gen} \text{ desigualdades} \\ \sum_{i=1}^N P_i &= P_{carga} \quad 1 \text{ restricción} \end{aligned} \quad (2.5)$$

Cuando reconocemos las restricciones de desigualdad, entonces las condiciones necesarias se pueden expandir ligeramente como se muestra en el conjunto de ecuaciones que componen la Ecuación (2.6).

$$\begin{aligned}
\frac{dF_i}{dP_i} &= \lambda & \text{para } P_{i,min} < P_i < P_{i,max} \\
\frac{dF_i}{dP_i} &\leq \lambda & \text{para } P_i = P_{i,max} \\
\frac{dF_i}{dP_i} &\geq \lambda & \text{para } P_i = P_{i,min}
\end{aligned} \tag{2.6}$$

Las siguientes tres unidades generadoras se usarán de ejemplo en futuros capítulos.

Unidad 1: unidad de vapor de carbón:

Salida Max = 600 MW

Salida Min = 150 MW

Curva de entrada-salida

$$H_1 \left(\frac{MBtu}{h} \right) = 510 + 7.2P_1 + 0.00142P_1^2$$

Unidad 2: unidad de vapor con aceite:

Salida Max = 400 MW

Salida Min = 100 MW

Curva de entrada-salida

$$H_2 \left(\frac{MBtu}{h} \right) = 310 + 7.85P_2 + 0.00194P_2^2$$

Unidad 3: Unidad de vapor con aceite:

Max output = 200 MW

Min output = 50 MW

Curva de entrada-salida

$$H_3 \left(\frac{MBtu}{h} \right) = 78 + 7.97P_3 + 0.00482P_3^2$$

Ejemplo 2.1 Supongamos que deseamos determinar el punto de operación económico para estas tres unidades cuando se entregan un total de 850 MW. Antes de que este problema pueda ser resuelto el costo de combustible de cada unidad debe ser especificado.

Unidad 1: Costo del combustible= 1.1\$/MBtu

Unidad 2: Costo del combustible= 1.0\$/MBtu

Unidad 3: Costo del combustible= 1.0\$/MBtu

Entonces, las ecuaciones de costo de producción son $F = H * costo$:

$$F_1(P_1) = H_1(P_1) * 1.1 = 561 + 7.92P_1 + 0.001562P_1^2 \$/h$$

$$F_2(P_2) = H_2(P_2) * 1.0 = 310 + 7.85P_2 + 0.00194P_2^2 \$/h$$

$$F_3(P_3) = H_3(P_3) * 1.0 = 78 + 7.97P_3 + 0.00482P_3^2 \$/h$$

Usando la ecuación. (2.4), las condiciones para un despacho óptimo son

$$\frac{dF_1}{dP_1} = 7.92 + 0.003124P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 7.85 + 0.00388P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 7.97 + 0.00964P_3 = \lambda$$

Y

$$P_1 + P_2 + P_3 = 850MW$$

Resolviendo por λ , uno obtiene

$$\lambda = 9.148 \$/MWh$$

Y luego resolviendo para P_1 , P_2 y P_3 .

$$P_1 = 393.2MW$$

$$P_2 = 334.6MW$$

$$P_3 = 122.2MW$$

Hay que tener en cuenta que se cumplen todas las restricciones; es decir, cada unidad está dentro de su límite alto y bajo, y la salida cuando se suma en las tres unidades cumple con el total deseado de 850 MW

2.2 Métodos de solución del problema de despacho económico

2.2.1 Búsqueda Binaria

Un algoritmo de iteración lambda es muy útil a veces llamado búsqueda binaria evita oscilaciones y siempre tiene éxito en encontrar el despacho económico óptimo. Primero, el usuario debe calcular el costo incremental en el máximo de generación y el rendimiento mínimo para cada generador. A continuación se establece λ_{min} al valor más pequeño entre los costos incrementales en los valores P_{min} del generador. si $\lambda = \lambda_{min}$, entonces el algoritmo de búsqueda lambda establecería todos los generadores en $P = P_{min}$, y si $\lambda = \lambda_{max}$, entonces el algoritmo de búsqueda lambda establecería todos los generadores en $P = P_{min}$, la búsqueda binaria comienza con:

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2}$$
$$\lambda_i = \lambda_{min} + \Delta\lambda$$

Que está en medio de cada extremo.

Después se calcula la salida del generador que corresponde a cada generador que tiene este costo incremental. Si el valor λ_1 es menor que el costo incremental en P_{min} . Entonces simplemente se configura la salida del generador en P_{min} , y si el valor λ_1 es mayor que el costo incremental en P_{max} , entonces simplemente se configura la salida del generador a P_{max} : de lo contrario, calcule el valor P para el generador con la función de costo incremental. Ahora se agrega todas las salidas del generador:

Si $\sum_{i=1}^{N_{gen}} P_i > P_{carga}$, se debe reducir lambda, entonces

$$\Delta\lambda = \frac{\Delta\lambda}{2}$$
$$\lambda_{i+1} = \lambda_i - \Delta\lambda$$

Si $\sum_{i=1}^{N_{gen}} P_i < P_{carga}$, se debe incrementar lambda, entonces

$$\Delta\lambda = \frac{\Delta\lambda}{2}$$
$$\lambda_{i+1} = \lambda_i + \Delta\lambda$$

Si $\text{abs}(\sum_{i=1}^{N_{gen}} P_i - P_{carga}) \leq \text{tolerancia}$, el algoritmo termina.

El valor delta lambda se reduce a la mitad en cada iteración. La búsqueda binaria funcionará con funciones de costo incrementales lineales por partes e incluso funciones de costo incremental lineal por partes que tienen saltos entre segmentos lineales.

En la Figura 2.2 se muestra el diagrama de la búsqueda binaria.

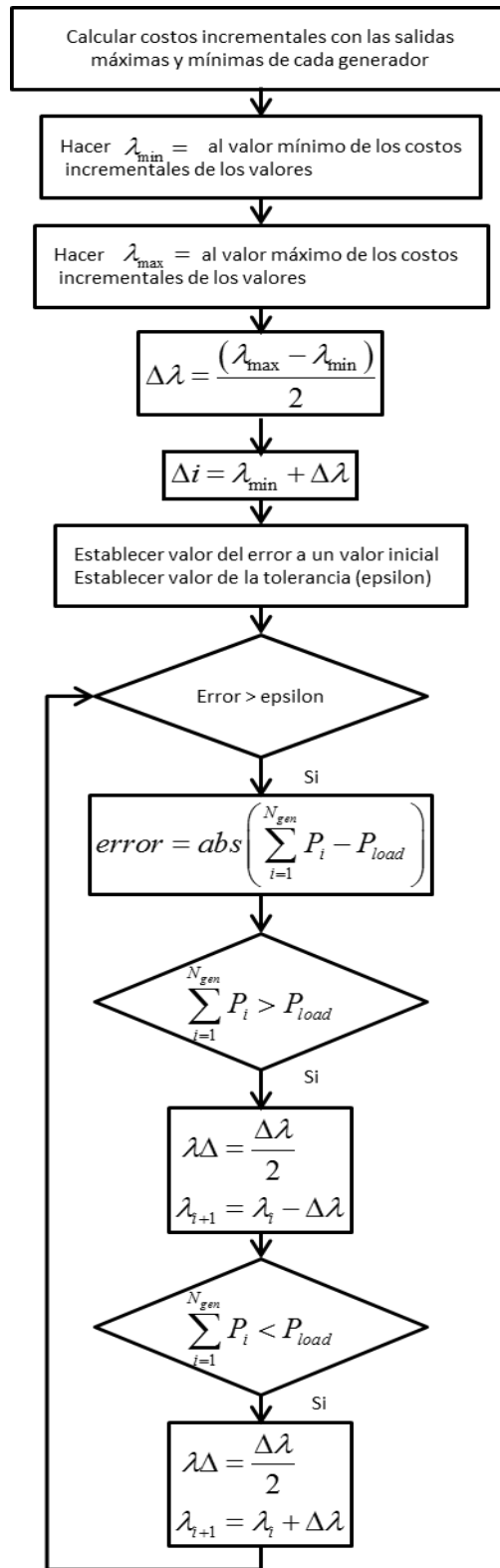


Figura 2.2. Diagrama búsqueda binaria.

2.2.2 Iteración Lambda

En la Figura 2.3 es un diagrama de bloques del método de solución iteración lambda para las pérdidas de desperdicio. Podemos abordar la solución a este problema considerando una técnica gráfica para resolver el problema y luego extenderlo al área de los algoritmos informáticos. Supongamos que tenemos un sistema de tres máquinas y deseamos encontrar el punto óptimo de operación económica.

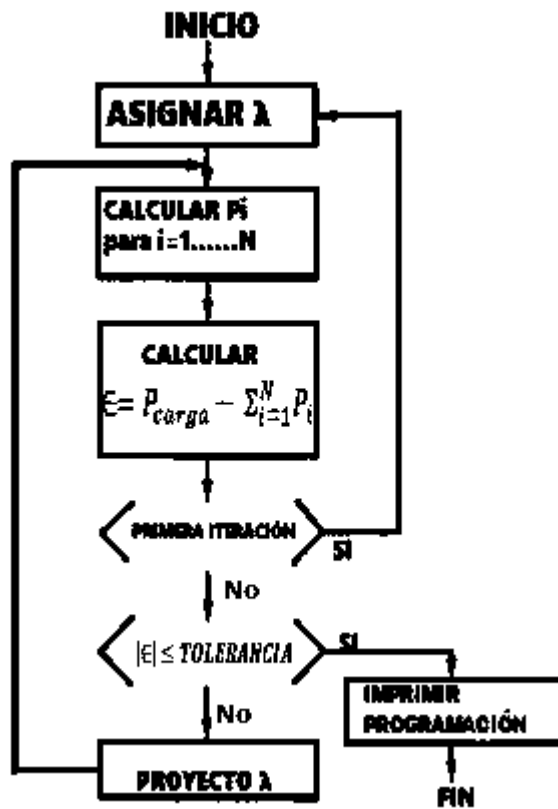


Figura 2.3. Despacho económico por el método de iteración lambda.

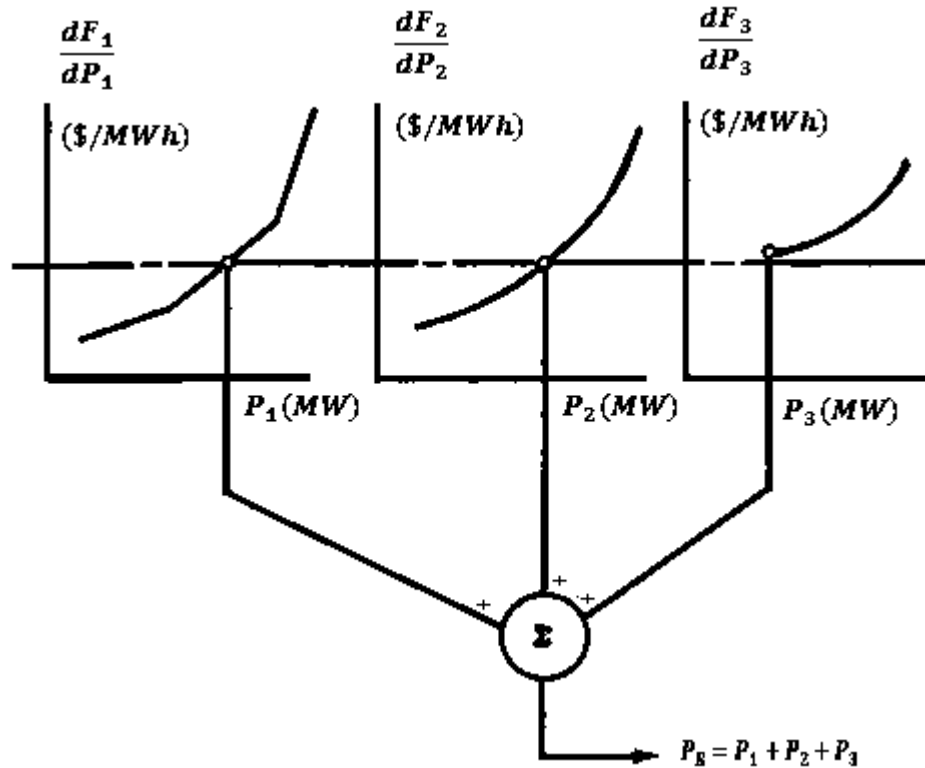


Figura 2.4. Solución gráfica al despacho económico.

Un enfoque sería graficar las características de costo incremental para cada una de estas tres unidades en el mismo gráfico, tal como se esboza en la Figura 2.4. Con el fin de establecer los puntos de operación de cada una de estas tres unidades de manera que tenga un costo mínimo y al mismo tiempo satisfacer la demanda especificada, podría usarse este boceto y una regla para encontrar la solución. Es decir, se puede asumir una tasa de costo incremental (λ) y encontrar las salidas de potencia de cada una de las tres unidades para este valor de costo incremental.

Por supuesto, la primera estimación será correcta. Si se asume el valor del costo incremental de tal manera que la producción de potencia total es demasiado baja, se debe valor de λ . Y probar con otra solución. Con dos soluciones, se puede extrapolar (o interpolar) las dos soluciones para acercarnos al valor deseado de la potencia total recibida Tal y como se puede apreciar en la Figura 2.5.

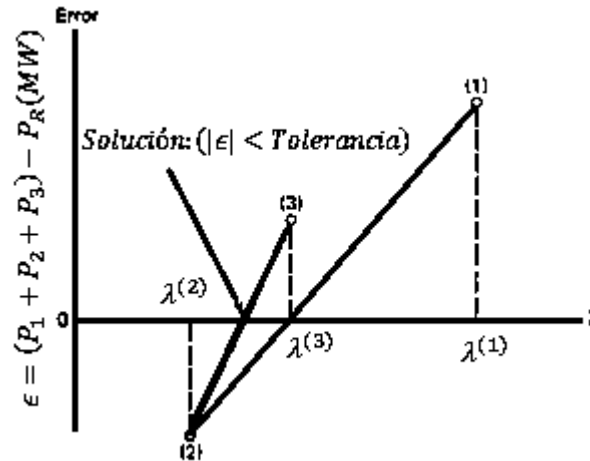


Figura 2.5. Proyecciones lambda.

Al hacer un seguimiento de la demanda total versus el costo incremental, se puede encontrar rápidamente el punto de operación deseado. Si se requiere, se puede fabricar una serie completa de tablas que mostrarían la potencia total suministrada para diferentes niveles de costos incrementales y combinaciones de unidades.

Este mismo procedimiento se puede adoptar para una implementación de computadora como se muestra en la Figura 2.3. Es decir, ahora estableceremos un conjunto de reglas lógicas que permitan lograr el mismo objetivo, que con la regla y el papel cuadriculado. Los detalles reales de cómo se establece la salida de potencia en función de la tasa de costo incremental son de muy poca importancia.

Podríamos, por ejemplo, almacenar tablas de datos dentro de la computadora e interpolar entre los puntos de energía almacenados para encontrar la salida de potencia exacta para un valor específico de tasa de costo incremental. Otro enfoque sería desarrollar una función analítica para la salida de potencia en función de la tasa de costo incremental, almacenar esta función (o sus coeficientes) en la computadora, y usar esto para establecer la salida de cada una de las unidades individuales.

Este procedimiento es un tipo iterativo de cálculo, y debemos establecer reglas de parada. Dos formas generales de reglas de paro parecen apropiadas para esta aplicación. La primera se muestra en la Figura 2.3 y es esencialmente una regla basada en encontrar el punto de operación apropiado dentro de una tolerancia especificada. La otra, que no se muestra en la Figura 2.3, implica contar el número de veces a través del ciclo iterativo y detenerse cuando se excede un número máximo.

El procedimiento de iteración lambda converge muy rápidamente para este tipo particular de problema de optimización. El procedimiento computacional real es un poco más complejo que el indicado en la Figura 2.3, ya que es necesario observar los límites operativos en cada una de las unidades durante el transcurso del cálculo. El bien conocido

método de Newton-Raphson se puede usar para proyectar el valor de costo incremental para conducir el error entre la generación calculada y la deseada a cero.

2.2.3 MÉTODO PL

Funciones de costos lineales por partes

PL es muy conveniente para manejar las restricciones de desigualdad siempre que el problema a resolver sea tal que pueda linealizarse sin pérdida de precisión.

En la formulación que sigue, se muestra cómo el problema del despacho económico se puede estructurar como PL. Primero, se aborda el problema de expresar las funciones de entrada-salida o costo no lineales como un conjunto de funciones lineales.

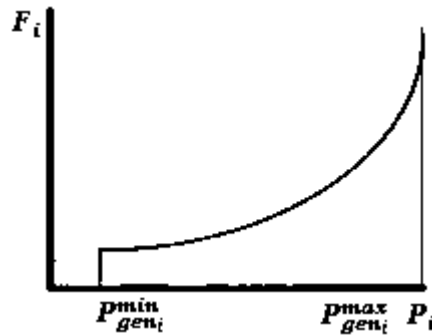


Figura 2.6. Característica de función de costo no lineal.

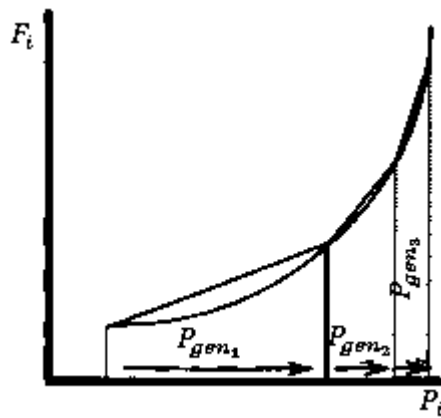


Figura 2.7. Función de costo no lineal aproximada por segmentos de línea recta.

Comenzamos con una reducción de costos no lineales que se muestra en la Figura 2.6.

Podemos aproximar esta función no lineal como una serie de segmentos de línea recta como se muestra en la Figura 2.7.

Los tres segmentos para el generador i mostrado se representarán como i_1, i_2 e i_3 . La variable P_i se reemplaza con tres nuevas variables $P_{gen_{i1}}, P_{gen_{i2}}$ y $P_{gen_{i3}}$. Cada segmento tendrá una pendiente designada S_{i1}, S_{i2}, S_{i3} (donde $S_{i1} < S_{i2} < S_{i3}$); entonces la función de costo en sí se representa ahora como la suma del costo en P_i^{min} más la suma del costo lineal para cada segmento, que es simplemente su pendiente multiplicado por la variable P_{ij} . Entonces

$$F_1(P_{gen_i}) = F_i(P_{gen_i}^{min}) + S_{i1}P_{gen_{i1}} + S_{i2}P_{gen_{i2}} + S_{i3}P_{gen_{i3}}$$

Donde

$$0 \leq P_{gen_{ik}} \leq P_{gen_{ik}}^{min} \text{ para } k = 1, 2, 3$$

Y finalmente

$$P_{gen_i} = P_{gen_i}^{min} + P_{gen_{i1}} + P_{gen_{i2}} + P_{gen_{i3}}$$

Y

$$S_{ik} = \frac{F_i(P_{gen_{ik+1}}) - F_i(P_{gen_{ik}})}{(P_{gen_{ik+1}}) - (P_{gen_{ik}})}$$

La función de costo ahora se compone de una expresión lineal en las tres variables $P_{gen_{i1}}, P_{gen_{i2}}$ y $P_{gen_{i3}}$.

Porque las pendientes aumentan el valor m , el programa lineal hará que $P_{gen_{ik}}$ esté en su límite $P_{gen_{ik}}^{max}$ antes de que $P_{gen_{i(k+1)}}$ aumente más allá de 0.

Despacho económico con Programación Lineal PL

Ahora la solución al método PL del despacho económico puede escribirse como

$$\text{Minimizar } \sum_{i=1}^{N_{gen}} (F_1(P_{gen_i}^{min}) + S_{i1}P_{gen_{i1}} + S_{i2}P_{gen_{i2}} + S_{i3}P_{gen_{i3}})$$

$$0 \leq P_{gen_{ik}} \leq P_{gen_{ik}}^{min} \text{ para } k = 1, 2, 3 \dots \text{ para todos los generadores } i = 1 \dots N_{gen}$$

Y finalmente

$$P_i = P_i^{min} + P_{gen_{i1}} + P_{gen_{i2}} + P_{gen_{i3}} \text{ para todos los generadores } i = 1 \dots N_{gen}$$

Sujeto a

$$\sum_{i=1}^{N_{gen}} P_i = P_{load}$$

Ahora mostraremos cómo PL resuelve el mismo despacho económico de tres generadores dado en el ejemplo anterior. Antes de investigar las soluciones que brinda el método de PL, es interesante ver con qué variables se enfrentará el método de PL. Aquí hay una lista de todas las variables para los tres generadores, problemas con tres segmentos usados para cada generador:

Variables del generador:

$$P_{gen_1}, P_{gen_{11}}, P_{gen_{12}}, P_{gen_{13}}, P_{gen_2}, P_{gen_{21}}, P_{gen_{22}}, P_{gen_{23}}, P_{gen_3}, P_{gen_{31}}, P_{gen_{32}}, P_{gen_{33}}$$

Junto con la función de costo lineal PL, habrá tres ecuaciones de generación de la línea $P_{gen_i} = P_{gen_i}^{min} + P_{gen_1} + P_{gen_{i2}} + P_{gen_{i3}}$ y habrá una ecuación para forzar la generación total a igual la carga.

2.3 Conclusiones

En este Capítulo, del despacho económico, observamos la problemática para poder suministrar la carga indicada respetando las restricciones y así la suma de las potencias generadas se igual a la carga conectada. Es un problema de optimización restringida y para poder llegar a una solución se aborda a través de tres métodos: búsqueda binaria, iteración lambda y el método simplex. En cada uno se respetan sus fórmulas, y se proporcionan los datos necesarios para llegar a una solución óptima.

Capítulo 3. Método simplex.

En este capítulo analizaremos método Simplex es un procedimiento iterativo que permite resolver problemas de Programación Lineal (PL) y mejoran la solución de la función objetivo en cada paso. El proceso concluye cuando no es posible continuar mejorando dicho valor, es decir, se ha alcanzado la solución óptima. El mayor o menor valor posible, según el caso, para el que se satisfacen todas las restricciones.

3.1 Programación lineal

Es un método de optimización que se ocupa del cumplimiento de un determinado objetivo como maximizar las utilidades o minimizar el costo, en presencia de restricciones como recursos limitados. El término lineal denota que las funciones matemáticas que representan el objetivo y las restricciones son lineales. El termino programar denota no hace referencia a la programación de computadoras esto hace referencia más a programar o fijar una fecha [Revelle y *et al*; 1997].

3.2 Forma estándar

Uno de los problemas básicos de la programación lineal consiste en dos partes principales una es la función objetivo y la segunda es un conjunto de restricciones. En un problema de maximización, la función objetivo se expresa como

$$\text{Maximizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (3.1)$$

Donde c_j = la contribución de cada unidad de la j -ésima actividad realizada y x_j = magnitud de j -ésima actividad. Así el valor de la función objetivo, Z , es la combinación total debida al número total de actividades, n .

Las restricciones se representan, en forma general, como

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad (3.2)$$

Donde a_{ij} = cantidad del i -ésimo recurso que se consume por cada unidad de la j -ésima actividad, y la b_i = cantidad del i -ésimo recurso que está disponible. Es decir, los recursos son limitados.

El segundo tipo general de restricción, especifica que todas las actividades deben de tener un valor positivo:

$$x_i \geq 0 \quad (3.3)$$

En este contexto, lo anterior expresa la noción realista de que en algunos problemas, la actividad negativa es físicamente imposible.

En conjunto, la función objetivo y las restricciones, especifican el problema de programación lineal. Estas indican que se trata de maximizar la contribución de varias actividades, bajo la restricción de que esta actividad utiliza cantidades finitas de recursos.

Planteamiento de un problema de la programación lineal

Supongamos que una planta procesadora de gas recibe cada semana una cantidad fija de materia prima. Esta última se procesa para dar dos tipos de gas calidad regular y Premium.

Estas clases de gas son de gran demanda y dan diferentes utilidades a la compañía. Sin embargo, su producción involucra restricciones de tiempo y almacenamiento por ejemplo, no se puede producir las dos clases a la vez, y las instalaciones están disponibles solamente 80 horas por semana. Además, existe un límite de almacenamiento para cada uno de los productos. Todos estos factores se enlistan en la Tabla 3.1.

Tabla 3.1. Recursos Disponibles

Recurso	Producto		Disponibilidad del recurso
	Regular	Premium	
Materia prima	7m ³ /ton	11m ³ /ton	77m ³ /semana
Tiempo de producción	10hr/ton	8hr/Ton	80hr/semana
Almacenamiento	9 ton	6 ton	
Aprovechamiento	150/ton	175/ton	

Solución: si las cantidades producidas cada semana de gas regular y Premium se designan como x_1 y x_2 respectivamente, la ganancia total se calcula mediante

$$\text{Ganancia total} = 150x_1 + 175x_2 \quad (3.4)$$

O se escribe como una función objetivo en programación lineal:

$$\text{Maximizar } Z = 150x_1 + 175x_2 \quad (3.5)$$

Las restricciones se desarrollan de una forma similar.

$$\text{Total de gas utilizado} = 7x_1 + 11x_2 \quad (3.6)$$

Este total no puede exceder el abastecimiento disponible de 77m³/semana, así que la restricción se representa como:

$$7x_1 + 11x_2 \leq 77 \quad (3.7)$$

Las restricciones restantes se desarrollan en una forma similar: la formación completa resultante de PL está dada por

$$\text{Maximizar } Z = 150x_1 + 175x_2 \quad (\text{Maximizar la ganancia}) \quad (3.8)$$

Sujeta a

$$7x_1 + 11x_2 \leq 77 \quad (\text{Restricciones de material}) \quad (3.9)$$

$$10x_1 + 8x_2 \leq 80 \quad (\text{Restricción de tiempo}) \quad (3.10)$$

$$x_1 \leq 9 \quad (\text{Restricción de almacenaje de gas "regular"}) \quad (3.11)$$

$$x_2 \leq 6 \quad (\text{Restricción de almacenaje de gas "Premium"}) \quad (3.12)$$

$$x_1, x_2 \geq 0 \quad (\text{Restricciones positivas}) \quad (3.13)$$

3.3 Solución Gráfica

Debido a que las soluciones gráficas están limitadas a dos o tres dimensiones, tienen una utilidad práctica limitada. Sin embargo son muy útiles para demostrar algunos conceptos básicos de las técnicas algebraicas generales, utilizadas para resolver problemas multidimensionales en la computadora.

En un problema bidimensional, como el del ejemplo anterior del gas regular y el Premium, el espacio de solución se define como un plano con x_1 medida a lo largo de la abscisa; y x_2 a lo largo de la ordenada. Como las restricciones son lineales, se trazan sobre este plano como líneas rectas. Si el problema de PL se formula adecuadamente estas

líneas restrictivas describen una región, llamada espacio de solución factible, que abarca todas las posibles combinaciones para x_1 y x_2 las cuales obedecen las restricciones y, por lo tanto, representan soluciones factibles. La función objetivo de un valor particular de Z se puede trazar como otra línea recta y sobreponerse en este espacio. El valor de Z , entonces se ajusta hasta que esté en el valor máximo, y toque aun el espacio factible. Este valor de Z representa la solución óptima. Los valores correspondientes de x_1 y x_2 donde Z toca el espacio de solución factible, representa los valores óptimos de las actividades.

Ejemplo usando una solución gráfica para el problema anterior

$$\text{Maximizar } Z = 150x_1 + 175x_2 \quad (3.14)$$

$$7x_1 + 11x_2 \leq 77 \quad (1)$$

$$10 + 8x_2 \leq 80 \quad (2)$$

$$x_1 \leq 9 \quad (3)$$

$$x_2 \leq 6 \quad (4)$$

$$x_1 \geq 0 \quad (5)$$

$$x_2 \geq 0 \quad (6)$$

Se numeran las restricciones para identificarlas en la solución gráfica.

Se reformula la primera restricción como una línea al remplazar la desigualdad por un signo igual, y se despeja x_2 :

$$x_2 = -\frac{7}{11}x_1 + 7 \quad (3.15)$$

Las otras restricciones se evalúan en forma similar, se sobreponen en la Figura 3.1 inciso a).

Se verá como estas encierran una región donde todas se satisfacen, eso se conoce como el espacio solución factible.

Después se agrega la función objetivo a la gráfica. Para hacerlo, se debe escoger un valor de Z . Por ejemplo, para $Z = 0$, la función objetivo es ahora

$$0 = 150x_1 + 175x_2 \quad (3.16)$$

O, despejando x_2 , se obtiene la línea recta

$$x_2 = -\frac{150}{175}x_1 \quad (3.17)$$

Ahora digamos que como queremos maximizar Z , esta se aumenta a digamos 600, y la función objetivo es

$$x_2 = \frac{600}{175} - \frac{150}{175}x_1 \quad (3.18)$$

Así al incrementar el valor de la función objetivo. La línea se aleja del origen. Como la línea todavía se encuentra dentro del espacio solución, nuestro resultado es aun factible. No obstante, por la misma razón aún hay espacio para mejorarlo. Por lo tanto Z continua aumentando hasta que un incremento adicional lleve la función objetivo más allá de la región factible como se muestra en la Figura 3.1 inciso b), el valor máximo de Z corresponde aproximadamente a 1400. En este punto x_1 y x_2 son, de manera aproximada, iguales a 4.9 y 3.9, respectivamente. Así, la solución grafica indica que si se producen estas cantidades de gas regular y Premium. Se alcanzara una máxima utilidad de aproximadamente 1400.

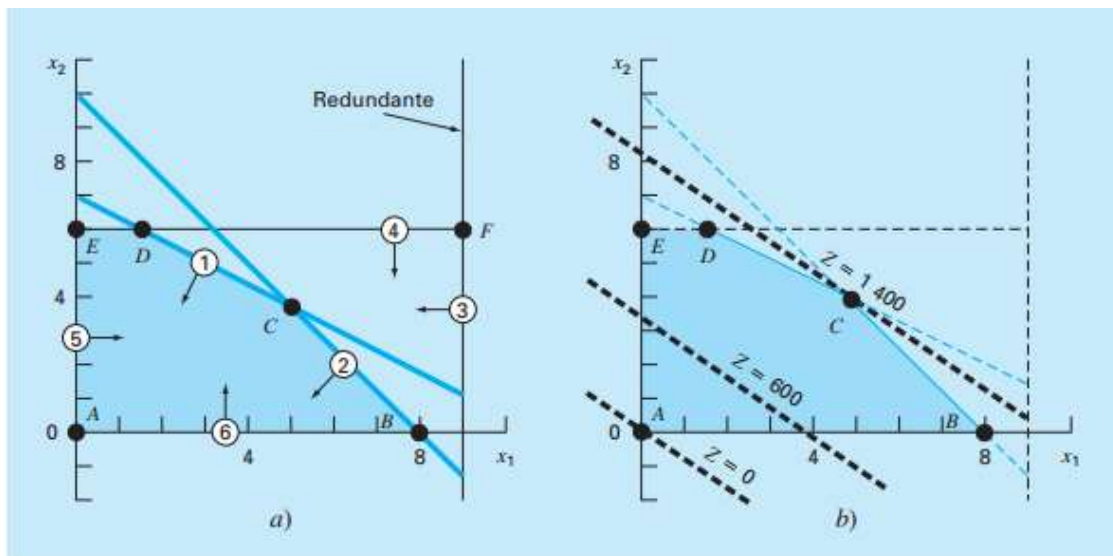


Figura 3.1. Solución gráfica del problema de programación lineal.

El método grafico ofrece una mejor comprensión del problema. Esto se aprecia al sustituir de nuevo las soluciones en las ecuaciones restrictivas:

$$7(4.9) + 11(3.9) = 77 \quad (3.19)$$

$$10(4.9) + 8(3.9) = 80 \quad (3.20)$$

$$4.9 \leq 9 \quad (3.21)$$

$$3.9 \leq 6 \quad (3.22)$$

En consecuencia, como se ve claramente en la gráfica, producir la cantidad óptima de cada producto nos lleva directamente al punto donde se satisfacen las restricciones de los recursos (1) y del tiempo (2). Estas restricciones se dicen que son obligatorias. Además la gráfica también hace evidente que ninguna de las restricciones de almacenamiento (3) ni (4) actúan como una limitante. Tales restricciones se conocen como no obligatorias. Esto nos lleva a la conclusión práctica de que, en caso, se puede aumentar las utilidades, ya sea con un incremento en el abastecimiento de recursos el gas o crudo o en tiempo de producción. Esto nos indica que el aumento del almacenamiento podría no tener impacto sobre las utilidades.

El resultado obtenido en el ejemplo anterior es uno de los cuatro posibles resultados que, por lo general, se obtienen en un problema de programación lineal. Estas son:

1. Solución única. Como en el ejemplo, la función objetivo máxima intersecta un solo punto.

2. Soluciones alternativas. Suponga que los coeficientes de la función objetivo del ejemplo fueran paralelos precisamente a una de las restricciones. En nuestro ejemplo, una forma en la cual esto podría ocurrir sería que las utilidades se modificaran a \$140/ton y \$220/ton. Entonces en lugar de un solo punto, el problema podría tener un número infinito de óptimos correspondientes a un segmento de línea como en la Figura 3.2.

3. Solución no factible. Como en la Figura 3.3, es posible que el problema este formulado de tal manera que no haya una solución factible. Esto puede deberse a que se trata de un problema sin solución o a errores en la formulación del problema. Lo último ocurre si el problema está sobre restringido, y ninguna solución satisface todas las restricciones.

4. Problemas no acotado. Como en la Figura 3.4, esto usualmente significa que el problema está subrestringido y, por lo tanto, tiene límites abiertos. Como en el caso de la solución no factible, esto a menudo ocurre debido a errores cometidos durante la especificación del problema.

Ahora supongamos que el problema tiene una solución única. El procedimiento gráfico podría sugerir una estrategia numérica para dar el máximo. Observar en las primeras gráficas a) y b) de la Figura 3.1 deberá quedar claro que siempre se presenta el óptimo en uno de los puntos esquina, donde se presentan dos restricciones. Tales puntos se conocen de manera formal como puntos extremos. Así como un número de posibilidades en el espacio de decisiones. Al enfocarse en los extremos, se reducen claramente las opciones posibles.

Es posible reconocer que no todo extremo es factible; es decir que satisfacen todas las restricciones. Observemos que el punto F en la Figura 3.1 a) es un punto extremo, pero

no es factible. Al limitarse a puntos extremos factibles, se reduce todavía más el campo que es factible.

Por último una vez que se han identificado todos los puntos extremos factibles, el que ofrezcan el mejor valor de la función objetivo representara la solución óptima, para encontrar la solución óptima se tendrá que evaluar el valor de la función objetivo en cada punto extremo factible.

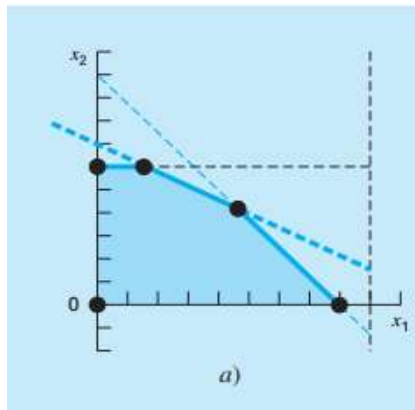


Figura 3.2. Alternativa óptima.

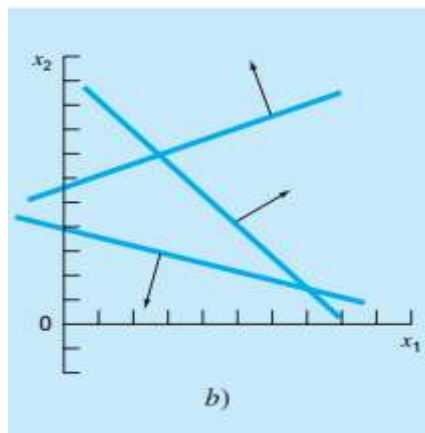


Figura 3.3. Solución no factible.

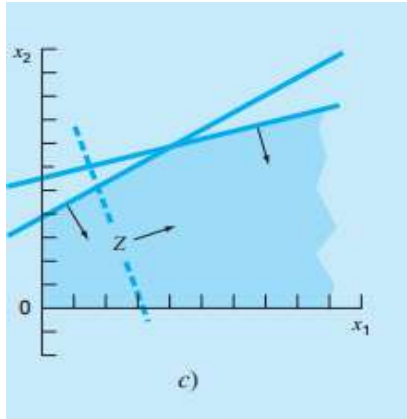


Figura 3.4. Resultado no acotado.

3.4 El método simplex

Partiendo del valor de la función objetivo en un punto cualquiera, el procedimiento consiste en buscar otro punto que mejore el valor anterior. Como se vio en el método Gráfico, dichos puntos son los vértices del polígono (o poliedro o polícoro, si el número de variables es mayor de 2) que constituye la región determinada por las restricciones a las que se encuentra sujeto el problema (llamada región factible). La búsqueda se realiza mediante desplazamientos por las aristas del polígono, desde el vértice actual hasta uno adyacente que mejore el valor de la función objetivo. Siempre que exista región factible, como su número de vértices y de aristas es finito, será posible encontrar la solución.

El método Simplex se basa en la siguiente propiedad: si la función objetivo Z no toma su valor máximo en el vértice A , entonces existe una arista que parte de A y a lo largo de la cual el valor de Z aumenta.

Será necesario tener en cuenta que el método Simplex únicamente trabaja con restricciones del problema cuyas inecuaciones sean del tipo " $\leq$ " (menor o igual) y sus coeficientes independientes sean mayores o iguales a 0. Por lo tanto, habrá que estandarizar las restricciones para que cumplan estos requisitos antes de iniciar el algoritmo del método Simplex. En caso de que después de éste proceso aparezcan restricciones del tipo " $\geq$ " (mayor o igual) o " $=$ " (igualdad), o no se puedan cambiar, será necesario emplear otros métodos de resolución, siendo el más común el método de las Dos Fases.

Preparando el modelo para adaptarlo al método Simplex

La forma estándar del modelo de problema consta de una función objetivo sujeta a determinadas restricciones:

$$\text{Función objetivo:} \quad c_1 \cdot x_1 + c_2 x_2 + \cdots + c_n x_n$$

Sujeto a:

$$\begin{aligned}
 a_{11} \cdot x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
 a_{21} \cdot x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
 &\vdots \\
 a_{m1} \cdot x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \\
 x_1, \dots, x_n &\geq 0
 \end{aligned}
 \tag{3.23}$$

El modelo debe cumplir las siguientes condiciones:

El objetivo consistirá en maximizar o minimizar el valor de la función objetivo (por ejemplo, incrementar ganancias o reducir pérdidas, respectivamente).

Todas las restricciones deben ser ecuaciones de igualdad (identidades matemáticas).

Todas las variables (x_i) deben tener valor positivo o nulo (condición de no negatividad).

Los términos independientes (b_i) de cada ecuación deben ser no negativos.

Hay que adaptar el problema modelado a la forma estándar para poder aplicar el algoritmo del Simplex.

Tipo de optimización.

Como se ha comentado, el objetivo del método consistirá en optimizar el valor de la función objetivo. Sin embargo, se presentan dos opciones: obtener el valor óptimo mayor (maximizar) u obtener el valor óptimo menor (minimizar).

Además existen diferencias en el algoritmo entre el objetivo de maximización y el de minimización en cuanto al criterio de condición de parada para finalizar las iteraciones y a las condiciones de entrada y salida de la base.

Objetivo de maximización

Condición de parada: cuando en la fila Z no aparece ningún valor negativo.

Condición de entrada a la base (base o variables básicas esto es, aquellas que toman valor para proporcionar una solución): el menor valor negativo en la fila Z (o el de mayor valor absoluto entre los negativos) indica la variable P_j que entra a la base.

Condición de salida de la base: una vez obtenida la variable entrante, la variable que sale se determina mediante el menor cociente $\frac{P_0}{P_j}$ de los estrictamente positivos.

Objetivo de minimización

Condición de parada: cuando en la fila Z no aparece ningún valor positivo (Z es la última fila de la tabla que recoge el valor la función objetivo y los costes reducidos).

Condición de entrada a la base (base o variables básicas esto es, aquellas que toman valor para proporcionar una solución): el mayor valor positivo en la fila Z indica la variable P_j que entra a la base.

Condición de salida de la base: una vez obtenida la variable entrante, la variable que sale se determina mediante el menor cociente $\frac{P_0}{P_j}$ de los estrictamente negativos.

No obstante, es posible normalizar el objetivo del problema con el fin de aplicar siempre los mismos criterios en lo referente a la condición de parada del algoritmo y a las condiciones de entrada y salida de las variables de la base. De esta forma, si el objetivo es minimizar la solución, se puede cambiar el problema a otro equivalente de maximización simplemente multiplicando la función objetivo por "-1". Es decir, el problema de minimizar Z es equivalente al problema de maximizar $(-1) * z$. Una vez obtenida la solución será necesario multiplicarla también por (-1) .

Ventajas: No hay que preocuparse por nuevos criterios de parada, condición de entrada y salida de la base ya que se mantienen.

Inconvenientes: En el caso de que la función tenga todos los coeficientes de sus variables básicas positivos, y además las restricciones sean del tipo de desigualdad " $\leq$ ", al hacer el cambio dichos coeficientes quedan negativos cumpliéndose la condición de parada en la primera iteración (en la fila del valor de la función objetivo todos los valores son positivos o cero). Obteniéndose en este caso por defecto un valor óptimo para la función igual a 0.

Solución: Realmente no existe este problema dado que para que la solución sea superior a 0 es necesario que alguna restricción tenga impuesta la condición " $\geq$ " (y se trataría de un modelo para el método de las Dos Fases). En el caso planteado, la solución real debe ser cero.

Cambio de signo de los términos independientes

También se ha dicho que los términos independientes (b_i) de cada ecuación deben ser no negativos para poder emplear el método Simplex. A tal fin, si alguna de las restricciones presenta un término independiente menor que 0 habrá que multiplicar por "-1" ambos lados de la inequación (teniendo en cuenta que esta operación también afecta al tipo de restricción).

Ventajas: Con ésta simple modificación de signos en las restricciones correspondientes se posibilita la aplicación del método Simplex al problema modelado.

Inconvenientes: Puede resultar que en las restricciones donde se tenga que modificar los signos de las constantes, los tipos de desigualdad fueran " $\leq$ " (quedando tras la operación del tipo " $\geq$ ") siendo necesario desarrollar el método de las Dos Fases. Este inconveniente no es controlable, aunque podría ocurrir el caso contrario y resultar beneficioso si los términos independientes negativos se presentan en todas aquellas restricciones con desigualdad de tipo " $\geq$ ". Si existe alguna restricción del tipo "=" no supondría ninguna ventaja ni desventaja puesto que siempre sería de necesaria aplicación el método de las Dos Fases.

Normalización de las restricciones

Otra de las condiciones del modelo estándar del problema es que todas las restricciones sean ecuaciones de igualdad (también llamadas restricciones de igualdad), por lo que hay que convertir las restricciones de desigualdad o inecuaciones en dichas identidades matemáticas.

La condición de no negatividad de las variables ($x_1, \dots, x_n \geq 0$) es la única excepción y se mantiene tal cual.

Restricción de tipo " $\leq$ "

Para normalizar una restricción con una desigualdad del tipo " $\leq$ ", hay que añadir una nueva variable, llamada variable de holgura x_s (con la condición de no negatividad: $x_s \geq 0$). Esta nueva variable aparece con coeficiente cero en la función objetivo, y sumando en la ecuación correspondiente (que ahora sí será una identidad matemática o ecuación de igualdad).

$$a_{11} * x_1 + a_{12} * x_2 \leq b_1 \rightarrow a_{11} * x_1 + a_{12} * x_2 + 1 * x_s = b_1$$

Restricción de tipo " $\geq$ "

En caso de una desigualdad del tipo " $\geq$ ", también hay que añadir una nueva variable llamada variable de exceso x_s (con la condición de no negatividad: $x_s \geq 0$). Esta nueva variable aparece con coeficiente cero en la función objetivo, y restando en la ecuación correspondiente.

Surge ahora un problema con la condición de no negatividad con esta nueva variable del problema. Las inecuaciones que contengan una desigualdad de tipo " $\geq$ " quedarían:

$$a_{11} * x_1 + a_{12} * x_2 \geq b_1 \rightarrow a_{11} * x_1 + a_{12} * x_2 - 1 * x_s = b_1$$

Al realizar la primera iteración con el método Simplex, las variables básicas no estarán en la base y tomarán valor cero. En este caso la nueva variable x_s , tras hacer cero a x_1 y x_2 , tomará el valor $-b_1$ y no cumpliría la condición de no negatividad. Es necesario añadir otra nueva variable x_r , llamada variable artificial, que también aparecerá con

coeficiente cero en la función objetivo y sumando en la restricción correspondiente. Quedando entonces de la siguiente manera:

$$a_{11} * x_1 + a_{12} * x_2 \geq b_1 \rightarrow a_{11} * x_1 + a_{12} * x_2 - 1 * x_s + 1 * x_r = b_1$$

Restricción de tipo "="

Al contrario de lo que cabría pensar, para las restricciones de tipo "=" (aunque ya son identidades) también es necesario agregar variables artificiales x_r . Como en el caso anterior, su coeficiente será cero en la función objetivo y aparecerá sumando en la restricción correspondiente.

$$a_{11} * x_1 + a_{12} * x_2 = b_1 \rightarrow a_{11} * x_1 + a_{12} * x_2 + 1 * x_r = b_1$$

En el último caso se hace patente que las variables artificiales suponen una violación de las leyes del álgebra, por lo que será necesario asegurar que dichas variables artificiales tengan un valor 0 en la solución final. En la Tabla 3.2 se muestra el tipo de desigualdad y el tipo de variable que aparece, de esto se encarga el método de las Dos Fases y por ello siempre que aparezcan este tipo de variables habrá que realizarlo.

Tabla 3.2. Tipos de desigualdad.

Tipo de desigualdad	Tipo de variable que aparece
$\geq$	- exceso + artificial
$=$	+ artificial
$\leq$	+ holgura

Desarrollando el método Simplex

Una vez estandarizado el modelo puede ocurrir que sea necesario aplicar el método Simplex o el método de las Dos Fases. En la Figura 3.5 se muestra la forma de actuación para llegar a la solución del problema modelado.

Método Simplex construcción de la primera tabla:

Las columnas de la tabla están dispuestas de la siguiente forma: la primera columna de la Tabla 3.3 contiene las variables que se encuentran en la base (o variables básicas), esto es, aquellas que toman valor para proporcionar una solución; la segunda columna recoge los coeficientes que dichas variables básicas tienen en la función objetivo (esta columna es llamada c_b); la tercera muestra el término independiente de cada restricción (P_0); a partir

de ésta aparece una columna por cada una de las variables de decisión y holgura presentes en la función objetivo (P_j). Para tener una visión más clara de la tabla, se incluye una fila que contiene los títulos de cada una de las columnas.

Sobre esta tabla se agregan dos nuevas filas: una de ellas, que lidera la tabla, donde aparecen los coeficientes de las variables de la función objetivo, y una última fila que recoge el valor la función objetivo y los costes reducidos $Z_j - C_j$.

Los costes reducidos muestran la posibilidad de mejora en la solución Z_0 . Por este motivo también son llamados valores indicadores.

En la Tabla 3.3 se muestra el aspecto general de la tabla del método Simplex:

Tabla 3.3. Tabla del método simplex construcción.

			C_1	C_2	...	C_n
Base	C_b	P_0	P_1	P_2	...	P_n
P_1	C_{b1}	b_1	a_{11}	a_{12}	...	a_{1n}
P_2	C_{b2}	b_2	a_{21}	a_{22}	...	a_{2n}
...	...	...	...	...	...	...
P_m	C_{bm}	b_m	a_{m1}	a_{m2}	...	A_{mn}
Z		Z_0	$Z_1 - C_1$	$Z_2 - C_2$	...	$Z_n - C_n$

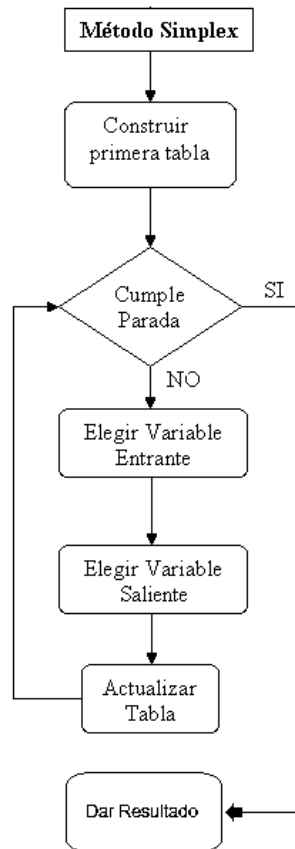


Figura 3.5. Diagrama del método simplex.

* Condición de parada: cuando la fila indicadora no contiene ningún valor negativo

Todos los valores incluidos en la tabla vendrán dados por el modelo del problema salvo los valores de la fila Z (o fila indicadora). Estos se obtienen de la siguiente forma: $Z_j = \sum (C_{bi} * P_j)$ para $i = 1..m$, donde si $J = 0$, $P_0 = b_i$ y $C_0 = 0$, y en caso contrario $P_j = a_{ij}$.

Se observa, al realizar el método Simplex, que en esta primera tabla ocupan la base todas las variables de holgura y por ello (todos los coeficientes de las variables de holgura son 0 en la función objetivo) el valor inicial de Z es cero.

Por este mismo motivo tampoco es necesario realizar los cálculos de los costes reducidos en la primera tabla, pudiéndose determinar directamente como el cambio de signo de los coeficientes de cada variable en la función objetivo, esto es, $-C_j$.

Condición de parada:

Se cumple la condición de parada cuando la fila indicadora no contiene ningún valor negativo entre los costes reducidos (cuando el objetivo es la maximización), esto es, no existe posibilidad de mejora.

Una vez cumplida la condición de parada, el valor de cada variable que logra la solución óptima se encuentra en la columna P_0 , indicándose en la base a qué variable corresponde dicho valor. Si una variable no aparece en la base, significa que su valor es cero. De la misma forma el valor óptimo de la función objetivo (Z) se encuentra en la columna P_0 , fila Z . Si no se cumple la condición de parada es necesario realizar una iteración más del algoritmo, esto es, determinar la variable que se vuelve básica y la que deja de serlo, encontrar el elemento pivote, actualizar los valores de la tabla y comprobar si se cumple nuevamente la condición de parada.

Es también posible determinar que el problema no se encuentra acotado y su solución siempre resultará mejorable. En tal caso no es necesario continuar iterando indefinidamente y se puede finalizar el algoritmo. Esta situación ocurre cuando en la columna de la variable entrante a la base todos los valores son negativos o nulos.

Elección de la variable que entra a la base:

Cuando una variable se vuelve básica, es decir, entra en la base, comienza a formar parte de la solución. Observando los costes reducidos en la fila Z , se decide que entra a la base la variable de la columna en la que éste sea el de menor valor (o de mayor valor absoluto) entre los negativos.

Elección de la variable que sale de la base:

Una vez obtenida la variable entrante, se determina que sale de la base la variable que se encuentre en aquella fila cuyo cociente $\frac{P_0}{P_j}$ sea el menor de los estrictamente positivos (teniendo en cuenta que esta operación se hará únicamente cuando P_j sea superior a 0).

Elemento pivote:

El elemento pivote de la tabla queda marcado por la intersección entre la columna de la variable entrante y la fila de la variable saliente.

Actualización de la tabla:

Las filas correspondientes a la función objetivo y a los títulos permanecerán inalteradas en la nueva tabla. El resto de valores deberán calcularse como se explica a continuación:

En la fila del elemento pivote cada nuevo elemento se calcula como:

Nuevo Elemento Fila Pivote = Anterior Elemento Fila Pivote / Pivote.

En el resto de las filas cada elemento se calcula:

Nuevo Elemento Fila = Anterior Elemento Fila - (Anterior Elemento Fila en Columna Pivote * Nuevo Elemento Fila Pivote).

De esta forma se consigue que todos los elementos de la columna de la variable entrante sean nulos salvo el de la fila de la variable saliente cuyo valor será 1. (Es análogo a utilizar el método de Gauss-Jordan para resolver sistemas de ecuaciones lineales).

Recordando el planteamiento del problema, supongamos que una planta procesadora de gas recibe cada semana una cantidad fija de materia prima. Esta última se procesa para dar dos tipos de gas calidad regular y Premium.

Estas clases de gas son de gran demanda y dan diferentes utilidades a la compañía. Sin embargo, su producción involucra restricciones de tiempo y almacenamiento por ejemplo, no se puede producir las dos clases a la vez, y las instalaciones están disponibles solamente 80 horas por semana. Además, existe un límite de almacenamiento para cada uno de los productos. Todos estos factores se enlistaron en la Tabla 3.1.

$$7x_1 + 11x_2 \leq 77 \quad (3.24)$$

Se define una variable de holgura S_1 como la cantidad de gas crudo que no se usa para un nivel de producción específico (x_1, x_2) esta cantidad se suma al lado izquierdo de nuestra restricción, esto vuelve exacta a la relación:

$$7x_1 + 11x_2 + S_1 = 77 \quad (3.25)$$

La variable de holgura indica si es positiva, significa que tiene algo de “holgura” en esta restricción. Es decir, se cuenta con un excedente de recursos que no se está utilizando por completo. Si es negativa, nos indica que hemos sobrepasado la restricción. Finalmente, si es cero, denota que la restricción se satisface con precisión.

Una variable de holgura diferente se desarrolla para cada ecuación restringida, lo cual resulta en lo que se conoce como la versión aumentada completamente

$$\text{Maximizar } Z = 150x_1 + 175x_2 \quad (3.26)$$

Sujeta a

$$7x_1 + 11x_2 + S_1 = 77 \quad (3.27)$$

$$10x_1 + 8x_2 + S_2 = 80 \quad (3.28)$$

$$x_1 + S_3 = 9 \quad (3.29)$$

$$x_2 + S_4 = 6 \quad (3.30)$$

$$x_1, x_2, S_1, S_2, S_3, S_4 \geq 0 \quad (3.31)$$

Se puede apreciar que las incógnitas quedaron acomodadas en columnas quedan alineadas en columnas. Esto se hace para resaltar que ahora se trata de un sistema de ecuaciones lineales.

Solución algebraica. Antes se tenían n ecuaciones con m incógnitas, en el sistema del ejemplo ecuaciones (desde la EC 3.26 hasta la EC 3.31) esta subespecificado, que tienes más incógnitas que ecuaciones. En términos generales, hay n variables estructurales “las incógnitas originales, m variables de holgura o excedentes una por cada restricción y $n + m$ variables en total. En el problema de la producción de gas se tienen 2 variables estructurales, 4 variables de holgura y 6 variables en total. Así, el problema consiste en resolver 4 ecuaciones con 6 incógnitas. La diferencia entre el número de incógnitas y el de ecuaciones igual a 2 en el problema está directamente relacionada con la forma en que se distingue un punto extremo factible. Específicamente cada punto factible tiene 2 de las 6 variables igualadas a cero. Por ejemplo los cinco puntos en las esquinas del área ABCDE tienen los siguientes valores cero como se muestra en la Tabla 3.4:

Tabla 3.4. Valores de punto extremo.

Punto extremo	Variables cero
A	x_1, x_2
B	x_2, S_2
C	S_1, S_2
D	S_1, S_4
E	x_1, S_4

Esto nos lleva a concluir que los puntos extremos se determinan a partir de la forma estándar igualando dos de las variables cero. Se aplica al problema reduce el problema a resolver 4 ecuaciones con 4 incógnitas. Para el punto E si $x_1=S_4=0$, la forma estándar se reduce a:

$$11x_2 + S_1 = 77 \quad (3.32)$$

$$8x_2 + S_2 = 80 \quad (3.33)$$

$$+S_3 = 9 \quad (3.34)$$

$$x_2 = 6 \quad (3.35)$$

Donde se obtiene $x_2=6$, $S_1=11$, $S_2=32$ y $S_3=9$. Junto con $x_1= S_4=0$, estos valores definen al punto E.

Generalizando una ecuación básica de m ecuaciones lineales con n incógnitas se obtiene al igualar a cero las variables $n - m$ y resolver las m ecuaciones para las m incógnitas restantes. Las variables igualadas a cero se conocen como variables no básicas y a las m variables restantes se les llaman básicas. Si todas las variables no son negativas al resultado se le llama una solución factible básica. Un procedimiento más directo para

determinar la solución óptima será calcular todas las soluciones básicas, determinar cuáles son factibles ver cuales tienen el valor mayor de Z. Pero este no es un procedimiento recomendable por dos razones:

La primera, aún para problemas de tamaños moderados, se necesita resolver una gran cantidad de ecuaciones. Para m ecuaciones con n incógnitas, se tendrá que resolver ecuaciones simultáneas.

$$C^n_m = \frac{n!}{m!(n-m)!} \quad (3.36)$$

Segundo, quizás una porción significativa de estas no sea factible. Por ejemplo, en el problema del gas crudo de los $C^4_6=15$ puntos extremos, solo 5 son factibles. Claramente, si se pudiese evitar resolver todos estos sistemas innecesarios, se tendría un algoritmo más eficiente.

3.5 Implementación del método simplex

Evita las ineficiencias descritas en la sección anterior 3.4 de este mismo Capítulo 3 que para problemas de tamaño moderado, se necesita resolver una gran cantidad de ecuaciones y puede que una gran porción de esas ecuaciones no sea factible, al comenzar con una solución factible básica. Luego se mueve a través de una secuencia de otras soluciones factibles básicas que mejoran sucesivamente el valor de la función objetivo. En forma eventual, se alcanza el valor óptimo y se termina el método.

Se ilustrara el procedimiento con el problema del procesamiento de gas. El primer paso consiste en empezar en una solución factible básica. Para casos como este un punto de inicio obvio podría ser el punto A; esto es, $x_1=x_2=0$. Las 6 ecuaciones originales en 4 incógnitas se convierten en

$$S_1 = 77 \quad (3.37)$$

$$S_2 = 80 \quad (3.38)$$

$$S_3 = 9 \quad (3.39)$$

$$S_4 = 6 \quad (3.40)$$

Así, los valores iniciales de las variables básicas se dan automáticamente siendo iguales a los lados derechos de las restricciones.

Como se muestra a continuación, la Tabla 3.5 proporciona un resumen de la información clave que constituye el problema de la programación lineal.

Tabla 3.5. Valores iniciales.

Básica	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solución	Intersección
Z	1	-150	-175	0	0	0	0	0	
S_1	0	7	11	1	0	0	0	77	11
S_2	0	10	8	0	1	0	0	80	8
S_3	0	1	0	0	0	1	0	9	9
S_4	0	0	1	0	0	0	1	6	∞

Se observa que para propósitos de la tabla, la función objetivo se expresa como

$$Z - 150x_1 - 175x_2 - 0S_1 - 0S_2 - 0S_3 - 0S_4 = 0 \quad (3.41)$$

El siguiente paso consiste en moverse a una nueva solución factible básica que lleve a mejorar la función objetivo. Incrementado una variable actual no básica por arriba de cero para que Z aumente.

Resumiendo este paso una de las variables no básicas actuales debe hacerse básica no cero. Esta variable se llama variable de entrada. En el proceso, una de las variables básicas actuales se vuelve no básica. Esta variable se llama variable de salida.

Haciendo el procedimiento para seleccionar las variables de entrada y de salida. A causa de la convención de cómo escribir la función objetivo, la variable de entrada puede ser cualquier variable de la función objetivo que tenga un coeficiente negativo. La variable con el valor negativo más grande se elige de manera convencional porque usualmente nos lleva al incremento mayor en Z . En nuestro caso, x_2 será la variable entrante puesto que su coeficiente, -175, es más negativo que el coeficiente de x_1 : -150.

Se puede ver la solución gráfica. Se comienza en el punto A considerando su coeficiente, se escogerá x_2 como entrada pero para más rápido seleccionamos x_1 puesto que en la gráfica nos llevara más rápido al máximo.

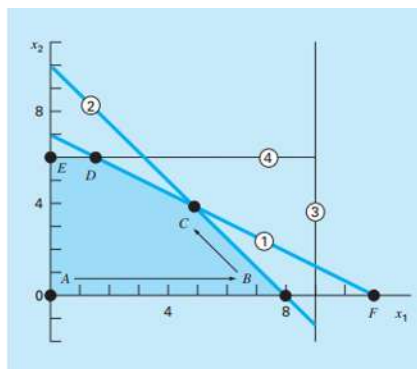


Figura 3.6. Gráfica para llegar al óptimo de manera eficiente.

Elegimos la variable de la salida entre las variables actuales. Se observó que hay dos posibilidades en la Figura 3.6. Moviendo al punto B se tendrá S_2 igual a cero, mientras que al mover al punto F tendremos S_1 igual a cero. Sin embargo, en la gráfica también nos queda claro que F no es posible, ya que queda fuera del espacio solución factible. Así, decide moverse de A a B.

Una manera de calcular valores en los que las líneas de restricción intersectan el eje o la línea que corresponde a la variable saliente. Es posible calcular este valor como la razón del lado derecho de la restricción entre el coeficiente correspondiente de x_1 . Para la primera variable de holgura restrictiva S_1 , el resultado es

$$\text{Intersección} = \frac{77}{7} = 11 \quad (3.42)$$

Las intersecciones restantes se pueden calcular y enlistar como la última columna de la Tabla 3.5. Debido a que 8 es el menor entero positivo. Significa que la segunda línea de restricción se alcanza primero conforme se incrementa x_1 , por lo tanto, S_2 será la variable de entrada.

De esta manera, se ha movido al punto B ($x_2 = S_2 = 0$), y la nueva solución básica es ahora

$$7x_1 + S_1 = 77 \quad (3.43)$$

$$10x_1 = 80 \quad (3.44)$$

$$+S_3 = 9 \quad (3.45)$$

$$+S_4 = 6 \quad (3.46)$$

La solución de este sistema de ecuaciones define efectivamente los valores de las variables básicas en el punto B: $x_1=8$, $S_1=21$, $S_3=1$ y $S_4=6$.

Se hace una tabla para realizar los cálculos usando el método de gauss Jordán.

En este ejemplo, el renglón pivote es S_2 y el elemento pivote es el 10, al dividir el renglón entre 10 y reemplazar S_2 por x_1 se tiene.

Tabla 3.6. Renglón pivote.

Básica	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solución
Z	1	-150	-175	0	0	0	0	0
S_1	0	7	11	1	0	0	0	77
x_1	0	1	0.8	0	0.1	0	0	8
S_3	0	1	0	0	0	1	0	9
S_4	0	0	1	0	0	0	1	6

Ahora eliminamos los coeficientes de x_1 en los otros renglones. Para el renglón de la función objetivo, el renglón pivote se multiplica por -150 y el resultado se resta del primer renglón para obtener

Tabla 3.7. Eliminación de x_1 .

Z	x_1	x_2	S_1	S_2	S_3	S_4	solución
1	-150	-175	0	0	0	0	0
-0	-(-150)	-(-120)	-0	-(-15)	0	0	-(-1200)
1	0	-55	0	15	0	0	1200

Es posible utilizar operaciones similares en los renglones restantes para obtener la nueva tabla 3.8.

Tabla 3.8. Nuevos valores.

Básica	Z	x_1	x_2	S_1	S_2	S_3	S_4	solución	intersección
Z	1	0	-55	0	15	0	0	1200	
S_1	0	0	5.4	1	-0.7	0	0	21	3.889
x_1	0	1	0.8	0	0.1	0	0	8	10
S_3	0	0	-0.8	0	-0.1	1	0	1	-1.25
S_4	0	0	1	0	0	0	1	6	6

Así la nueva Tabla 3.8 resume la información del punto B. esto incluye el hecho de que el movimiento ha aumentado la función objetivo a $Z=1200$.

La tabla se utiliza después para representar el próximo y, en este caso, último paso. Solo una variable más, x_2 tiene un valor negativo en la función objetivo, y se elige, por lo tanto, como la variable de salida. De acuerdo con los valores de la intersección, la primera restricción tiene el valor positivo más pequeño y por lo tanto, se selecciona S_1 como la variable de entrada. Así el método simplex nos mueve del punto B al C.

Por último, la eliminación de gauss-jordan se utiliza para resolver las ecuaciones simultáneas. El resultado es la tabla final 3.9.

Tabla 3.9. Eliminación Gauss-Jordán.

Básica	Z	x_1	x_2	S_1	S_2	S_3	S_4	solución
Z	1	0	0	10.1852	7.8704	0	0	1413.889
x_2	0	0	1	0.1852	-0.1296	0	0	3.889
x_1	0	1	0	-0.1481	0.2037	0	0	4.889
S_3	0	0	0	0.1481	-0.2037	1	0	4.111
S_4	0	0	0	-0.1852	0.1296	0	1	2.111

Este resultado final es porque no quedan coeficientes negativo en la fila de la función objetivo. La solución final se tabula como $x_1=3.889$ y $x_2=4.889$, que dan una función objetivo máxima $Z=1413.889$. Además, como S_3 y S_4 están todavía en la base, sabemos que la solución está limitada por la primera y las segunda restricciones.

3.6 Conclusiones

El método simplex es un algoritmo eficiente y confiable para resolver problemas de programación lineal mediante aproximaciones sucesivas que van mejorando con cada iteración de la función objetivo. Para que esto se cumpla, el objetivo consistirá de maximizar o minimizar el valor de la función objetivo, para poder optimizar esta función objetivo es necesario respetar las restricciones, con cada iteración la solución se mueve de la solución básica factible actual a una adyacente mejor eligiendo tanto la variable básica entrante como la que sale y después usando la eliminación de Gauss para resolver el sistema de ecuaciones lineales. Una vez que ya no se pueda mejor el resultado anterior habre llegado a nuestra solución óptima.

Capítulo 4. Aplicación del método simplex en la solución de un despacho económico.

En este Capítulo se describirá mediante la resolución de un ejercicio práctico como se soluciona el problema de DE aplicando tres métodos: Método simplex para resolver un problema de PL, el método de búsqueda binaria y el método de iteración lambda.

4.1 Propuesta de solución al problema de DE usando programación lineal y MATLAB

Solución usando PL:

Si tomamos el ejemplo de tres generadores presentado en la sección 2.1.

Unidad 1: unidad de vapor de carbón:

Salida Max = 600 MW

Salida Min = 150 MW

Curva de entrada-salida

$$H_1 \left(\frac{MBtu}{h} \right) = 510 + 7.2P_1 + 0.00142P_1^2$$

Unidad 2: unidad de vapor con aceite:

Salida Max = 400 MW

Salida Min = 100 MW

Curva de entrada-salida

$$H_2 \left(\frac{MBtu}{h} \right) = 310 + 7.85P_2 + 0.00194P_2^2$$

Unidad 3: Unidad de vapor con aceite:

Max output = 200 MW

Min output = 50 MW

Curva de entrada-salida

$$H_3 \left(\frac{MBtu}{h} \right) = 78 + 7.97P_3 + 0.00482P_3^2$$

Unidad 1: Costo del combustible= 1.1\$/MBtu

Unidad 2: Costo del combustible= 1.0\$/MBtu

Unidad 3: Costo del combustible= 1.0\$/MBtu

Entonces

$$F_1(P_1) = H_1(P_1) * 1.1 = 561 + 7.92P_1 + 0.001562P_1^2 \$/h$$

$$F_2(P_2) = H_2(P_2) * 1.0 = 310 + 7.85P_2 + 0.00194P_2^2 \$/h$$

$$F_3(P_3) = H_3(P_3) * 1.0 = 78 + 7.97P_3 + 0.00482P_3^2 \$/h$$

$$F_1(P_{gen_1}) = 561 + 7.92P_{gen_1} + 0.00562P_{gen_1}^2$$

$$F_2(P_{gen_2}) = 310 + 7.85P_{gen_2} + 0.00194P_{gen_2}^2$$

$$F_3(P_{gen_3}) = 78 + 7.97P_{gen_3} + 0.00482P_{gen_3}^2$$

Con los límites de potencia mínima y máxima del generador de

Tabla 4.1. Límites max y min.

Generador	P_{gen}^{min}	P_{gen}^{max}
1	150	600
2	100	400
3	50	200

Y con una carga de 850 MW en total, podremos calcular el Flujo de Potencia Óptimo (FPO) utilizando PL. La única diferencia es que, según la cantidad de segmentos utilizados, la solución será diferente de la obtenida con el método estándar. En la Tabla 4.2, utilizamos 1, 3, 5, 10 y 50 segmentos para cada función de costo, y se puede observar que la solución se cierra con la misma solución a medida que aumenta el número de segmentos.

Tabla 4.2. Segmentos costo valor.

Numero de segmentos	Generador 1 MW	Generador 2 MW	Generador 3 MW	Costo total (\$/h)
1	400	400	50	8227.870
2	375	350	125	8195.369
3	450	300	100	8204.105
5	400	340	110	8195.206
10	385	340	125	8194.554
50	393	335	122	8194.357

Hay que tener en cuenta que aumentar la cantidad de segmentos no necesariamente es la solución más cercana a la solución exacta. Cuando se pasa de dos segmentos a tres segmentos, la solución en realidad decae levemente en términos de costo total. Esto es simplemente porque los puntos de corte con tres segmentos se alejan más de la solución verdadera que el caso de dos segmentos. A medida que el número de segmentos aumenta a cinco, diez e incluso cincuenta, la solución se acerca mucho a la solución exacta.

4.1.1 Propuesta de solución al problema de DE usando programación lineal y MATLAB

En el problema de DE, intervienen curvas de costo, las cuales no son lineales. Sin embargo, para evitar este problema se realizará una aproximación lineal por partes de las curvas de costo.

En la Tabla 4.3 se muestra el código en Matlab para solucionar el problema de DE usando Matlab. Los datos mostrados corresponden al problema descrito al inicio de la sección 4.1.

Tabla 4.3. Código del método simplex en MATLAB.

```
1 function ms5(n)
2
3 pmin1 = 150;
4 pmin2 = 100;
5 pmin3 = 50;
6
7 pmax1=600;
8 pmax2=400;
9 pmax3=200;
10
11
12 % DATOS DEL PROBLEMA
13 load=850;
14 tic
15 pgmin=pmin1+pmin2+pmin3;
16
17 paso1 = pmax1-pmin1;
18 paso2 = pmax2-pmin2;
19 paso3 = pmax3-pmin3;
20
21 % Tamaño del segmento
22 delta1 = paso1/n;
23 delta2 = paso2/n;
24 delta3 = paso3/n;
25
26 potencias = zeros(3,n+1);
27 potencias(1,1) = pmin1;
28 potencias(2,1) = pmin2;
29 potencias(3,1) = pmin3;
30
31 for i=2:n+1
```

```

32 potencias(1,i) = potencias(1,i-1) + delta1;
33 potencias(2,i) = potencias(2,i-1) + delta2;
34 potencias(3,i) = potencias(3,i-1) + delta3;
35
36 end
37
38 % Ciclo para encontrar las pendientes de las características de E/S de los
39 % generadores
40
41 for i=1:n
42     s1(i) = (c1(potencias(1,i+1))-c1(potencias(1,i)))/delta1;
43     s2(i) = (c2(potencias(2,i+1))-c2(potencias(2,i)))/delta2;
44     s3(i) = (c3(potencias(3,i+1))-c3(potencias(3,i)))/delta3;
45 end
46
47 c=[ s1 , s2 , s3 ] ; % FUNCION OBJETIVO
48
49 % Minimizar F1(P1)+F2(P2)+F3(P3)
50 % Sujeto a las siguientes restricciones:
51
52 % Restricciones de desigualdad
53 A=[ ] ;
54 b=[ ] ;
55
56 % Restricciones de Igualdad
57 Aeq=zeros ( length ( c ) ) ;
58 Aeq ( 1 , : )=1 ;
59 Beq=zeros ( length ( c ) , 1 ) ;
60 Beq(1)=load-pgmin ;
61
62 % Limites (Limites superiores e inferiores)
63 LB=zeros ( length ( c ) , 1 ) ;
64 UB=zeros ( length ( c ) , 1 ) ;
65 for i=1:n
66     UB(i,1) = delta1;
67     UB(i+n,1) = delta2;
68     UB(i+2*n,1) = delta3;
69 end
70
71 % Programación lineal para encontrar la solución optima
72 % El vector x da los valores de generación (3 generadores)
73 % Los valores de generación minimizar el costo de producirá una
74 % carga de 850 MW
75 x = linprog(c ,A, b ,Aeq , Beq ,LB,UB) ;
76
77 % Ciclo para encontrar la respuesta a la generación pérdidas
78 suma1 = 0;
79 suma2 = 0;
80 suma3 = 0;
81 for i=1:n
82     suma1 = suma1 + x(i);
83     suma2 = suma2 + x(i+n);
84     suma3 = suma3 + x(i+2*n);
85
86 end
87 P1 = suma1+pmin1;
88 P2 = suma2+pmin2;
89 P3 = suma3+pmin3;
90 % GENERACION TOTAL
91 costo = c1(P1)+c2(P2)+c3(P3);
92
93 [toc n P1 P2 P3 costo]

```


En la Tabla 4.4 se muestra el código manda llamar la función ms5 la cual revisa una lista y verifica el número de segmentos con los que realizara operaciones para el método simplex

Tabla 4.4. Función ms5 del código de MATLAB.

```
1 lista = [1 2 3 5 10 50 75 100 150 200 250 300 500 1000 1500 2000 2500]
2 for i=1:length(lista)
3   ms5(lista(i))
4 end
```

En la Tabla 4.5, se muestran los resultados generados con el código anterior usando el método simplex para resolver el ejercicio con tres generadores planteado al inicio de esta sección. En la Tabla 4.5, en la columna 1 se presentó el número de segmentos en los cuales se divide cada curva; las columnas P1, P2 y P3 muestran la potencia de cada una de las unidades citadas; la columna de Costo total muestra el costo que implicaría usar las potencias P1, P2 y P3 asociadas y la columna de Tiempo indica el tiempo de ejecución para Matlab para el código de PL por partes.

Tabla 4.5. Resultados obtenidos en Matlab para PL por partes.

No. Segmentos	P1 MW	P2 MW	P3 MW	Costo Total (\$/h)	Tiempo(s) En seg
1	400.00	400.00	50.00	8227.86999947	0.01730080
2	375.00	350.00	125.00	8195.36874981	0.01511429
3	450.00	300.00	100.00	8204.10500000	0.01678852
5	400.00	340.00	110.00	8195.20599987	0.01695138
10	385.00	340.00	125.00	8194.55394995	0.01752999
25	388.00	340.00	122.00	8194.45460798	0.02073810
50	393.00	335.00	122.00	8194.35671800	0.02563050
75	392.00	336.00	122.00	8194.36228799	0.02565123
100	393.00	335.00	122.00	8194.35671800	0.03509966
200	393.00	335.00	122.00	8194.35671800	0.05032291
250	393.00	335.00	122.00	8194.35671800	0.05672847
500	393.10	334.60	122.30	8194.35615502	0.12333473
750	393.00	334.80	122.20	8194.35624440	0.19994729
1000	393.10	334.60	122.30	8194.35615502	0.32091784
1250	393.04	334.72	122.24	8194.35617471	0.45203919
1500	393.20	334.60	122.20	8194.35612608	0.54885448
1750	393.25	334.51	122.23	8194.35614777	0.68359436
2000	393.18	334.60	122.22	8194.3561213	0.76095077

La mejora que tuvo el método simplex con respecto al costo total desde el uso de un solo segmento a 2000 segmentos, fue una mejora del 0.4%.

4.2 Propuesta de solución al problema de DE usando búsqueda binaria y MATLAB

Solución al problema de DE usando Búsqueda binaria:

Usando el ejemplo de tres generadores de la sección 2.1

Usando la ecuacion.2.4, las condiciones para un despacho óptimo son

$$\frac{dF_1}{dP_1} = 7.92 + 0.003124P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 7.85 + 0.00388P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 7.97 + 0.00964P_3 = \lambda$$

PL = 850 MW

Gen 1 150 a 600 MW

Gen 2 100 a 400 MW

Gen 3 50 a 200 MW

Tabla 4.6. Costos incrementales.

Min	Max
8.3886	9.7944
8.238	9.402
8.452	9.898

Calculamos los costos incrementales máximos y mínimos

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} = \frac{9.898 - 8.238}{2} =$$
$$\Delta\lambda = 0.83$$

Se calcula delta lambda con los costos incrementales máximo y mínimo.

$$\lambda_1 = \lambda_{min} + \Delta\lambda$$

$$\lambda_1 = 8.238 + 0.83 = 9.068$$

Con el delta lambda se calcula la lambda inicial.

$$P_1 = \frac{\lambda - 7.92}{0.003124}$$

$$P_1 = \frac{9.068 - 7.92}{0.003124} = 367.477592$$

$$P_2 = \frac{\lambda - 7.85}{0.00388}$$

$$P_2 = \frac{9.068 - 7.85}{0.00388} = 313.917525$$

$$P_3 = \frac{\lambda - 7.97}{0.00964}$$

$$P_3 = \frac{9.068 - 7.97}{0.00964} = 113.90041494$$

Una vez obtenida la lambda inicial se sustituye para obtener P_1, P_2, P_3 .

$$P_T = P_1 + P_2 + P_3$$

Se suma P_1, P_2, P_3 para obtener P_T .

$$P_T = 795.29553124$$

$$\text{Si } P_T > P_L$$

Entonces

$$\Delta\lambda = \frac{\Delta\lambda}{2}$$

$$\lambda_2 = \lambda_1 - \Delta\lambda$$

$$\text{Si } P_T < P_L$$

Entonces

$$\lambda_2 = \lambda_1 + \Delta\lambda$$

Se obtiene una nueva lambda

$$\lambda_2 = 9.068 + \frac{0.83}{2}$$

$$\lambda_2 = 9.483$$

Con la nueva lambda se vuelve a sustituir para obtener los nuevos P_1, P_2, P_3 .

$$\text{Error} = |P_T - P_L|$$

$$\text{While (error} > \text{tol)}$$

$$\text{Tol} = 1 * 10^{-6}$$

El proceso se repite hasta que la tolerancia sea mayor que el error

Código de la búsqueda binaria en Matlab

A continuación, se presenta el código en Matlab para solucionar el problema de DE mediante búsqueda binaria. De la línea 1 a la 10 del código se declara las variables y sus valores. En la línea 4 se inicializa que es la carga máxima, en la línea 4lmin nuestra lambda mínima, en línea 5 se muestra la lmax que lambda máxima, en la línea 6 muestra dl que es delta lambda, en la línea 7 muestra la ln que es nueva lambda, en la línea 8 se tiene el error, en la línea 9 se tiene épsilon que es la tolerancia y en la línea 10 un contador el valor de las iteraciones.

```
1 %método busqueda binaria
2 %Datos del problema
3 pl = 850;
4 lmin = 8.238;
5 lmax = 9.898;
6 dl = (lmax-lmin)/2;
7 ln = lmin + dl;
8 error = 1.0;
9 epsilon = 1e-6;
10 cont = 0;
```

El siguiente párrafo muestra el código de la línea 11 a la 36, en la linea13 del código a la 16 es donde se calculan p1, p2, p3 se hace la suma de estas para estimar pt (carga total) se verifica el error. Si no termina continua con las condiciones para calcular los nuevos valores para dl y también para ln. El contador de iteraciones se va incrementando y el ciclo se repite hasta que épsilon sea mayor o igual que el error

```
11 tic
12 while (error > epsilon)
13   p1 = (ln - 7.92)/0.003124;
14   p2 = (ln - 7.85)/0.00388;
15   p3 = (ln - 7.97)/0.00964;
16   pt = p1+p2+p3;
17   error = abs(pt-pl);
18   if (pt > pl)
19     dl = dl/2;
20     ln = ln -dl;
21   end
22   if (pt < pl)
23     dl = dl/2;
24     ln = ln +dl;
25   end
26   error
27   error
28   cont = cont + 1;
```

```

29 ct1 = 561 + 7.92*(p1) + 0.001562*(p1)^2;
30 ct2 = 310 + 7.85*(p2) + 0.00194*(p2)^2;
31 ct3 = 78 + 7.97*(p3) + 0.00482*(p3)^2;
32 ct=ct1+ct2+ct3
33 toc
34 [toc cont error p1 p2 p3 ln pt ct]
35
36 end

```

En la parte anterior se muestra la ejecución del programa tenemos las funciones tic y toc que permite calcular el tiempo de ejecución de cada iteración e imprimirlo en terminal junto con la impresión del contador, el error, los valores de p1, p2, p3, la nueva lambda, la potencia total pt y el costo total.

En la Tabla 4.7 se muestra el código en Matlab para solucionar el problema de DE usando Matlab. Los datos mostrados corresponden al problema descrito al inicio de la sección 4.1.

Código de la búsqueda binaria

Tabla 4.7. Código de la búsqueda binaria en MATLAB.

```

1 % metodo de busqueda binaria
2 % Datos del problema
3 pl = 850; % carga maxima
4 lmin = 8.238; % lambda minima
5 lmax = 9.898; % lambda maxima
6 dl = (lmax-lmin)/2; % delta lambda
7 ln = lmin + dl; % nueva lambda
8 error = 1.0; % error
9 epsilon = 1e-6; % tolerancia
10 cont = 0;
11 tic % medir el tiempo de ejecución
12 % ciclo que verifica que el error sea mayor que la tolerancia
13 while (error > epsilon)
14     p1 = (ln - 7.92)/0.003124;% valores para nuestro p1
15     p2 = (ln - 7.85)/0.00388;% valores para nuestro p2
16     p3 = (ln - 7.97)/0.00964;% valores para nuestro p3
17     pt = p1+p2+p3; % nuestra pt la suma de nuestras p1, p2, p3
18     error = abs(pt-pl); % error
19     if (pt > pl) % verifica si nuestra Pt es mayor que pl para cumplir la condición
20         dl = dl/2;
21         ln = ln -dl;
22     end
23     if (pt < pl) % verifica si nuestra Pt es menor que pl para cumplir la condición
24         dl = dl/2;
25         ln = ln +dl;
26     end
27     error
28     cont = cont + 1;
29     ct1 = 561 + 7.92*(p1) + 0.001562*(p1)^2;% los costos de p1
30     ct2 = 310 + 7.85*(p2) + 0.00194*(p2)^2;% los costos de p2
31     ct3 = 78 + 7.97*(p3) + 0.00482*(p3)^2;% los costos de p3
32     ct=ct1+ct2+ct3 % el costo total
33 [toc cont error p1 p2 p3 ln pt ct] % el orden que los muestra impresos

```

A continuación en la Tabla 4.8 se observará los resultados que se obtuvieron al ejecutar el código de la búsqueda binaria. Binaria al ejecutar el programa en MATLAB para el ejemplo 2.1:

Tabla 4.8. Resultados de la búsqueda binaria Ejemplo 2.1 con MATLAB

cont	Error	P1 MW	P2 MW	P3 MW	ln	Pt MW	Costo Total (\$/h)	Tiempo en seg
1	54.704466	367.477593	313.917526	113.90042	9.483	795.29553	7696.100659	0.0005478
2	228.146599	500.320102	420.876289	156.95021	9.2755	1078.1466	10319.68571	0.0013577
3	86.721066	433.898848	367.396907	135.42531	9.17175	936.72107	8993.220287	0.0019478
4	16.0083	400.68822	340.657216	124.66286	9.119875	866.0083	8340.992248	0.0025479
5	19.348083	384.082907	327.287371	119.28164	9.145812	830.65192	8017.629397	0.0031828
6	1.669892	392.385563	333.972294	121.97225	9.158781	848.33011	8179.081559	0.0037629
7	7.169204	396.536892	337.314755	123.31756	9.152297	857.1692	8259.979587	0.0043749
8	2.749656	394.461228	335.643524	122.6449	9.149055	852.74966	8219.516244	0.0046849
9	0.539882	393.423395	334.807909	122.30858	9.147434	850.53988	8199.295319	0.0049607
10	0.565005	392.904479	334.390101	122.14041	9.148244	849.435	8189.187543	0.0052099
11	0.012561	393.163937	334.599005	122.2245	9.148649	849.98744	8194.241207	0.0054788
12	0.26366	393.293666	334.703457	122.26654	9.148447	850.26366	8196.768207	0.0057618
13	0.12555	393.228802	334.651231	122.24552	9.148345	850.12555	8195.504693	0.0060529
14	0.056494	393.19637	334.625118	122.23501	9.148295	850.05649	8194.872947	0.0063698
15	0.021966	393.180154	334.612062	122.22975	9.148269	850.02197	8194.557076	0.0066859
16	0.004703	393.172046	334.605534	122.22712	9.148257	850.0047	8194.399141	0.0069959
17	0.003929	393.167991	334.602269	122.22581	9.148263	849.99607	8194.320174	0.0073118
18	0.000387	393.170019	334.603901	122.22647	9.14826	850.00039	8194.359658	0.007622
19	0.001771	393.169005	334.603085	122.22614	9.148262	849.99823	8194.339916	0.0079348
20	0.000692	393.169512	334.603493	122.2263	9.148262	849.99931	8194.349787	0.0082659
21	0.000153	393.169765	334.603697	122.22638	9.148263	849.99985	8194.354722	0.0085799
22	0.000117	393.169892	334.603799	122.22643	9.148263	850.00012	8194.35719	0.0088789
23	0.000018	393.169828	334.603748	122.22641	9.148263	849.99998	8194.355956	0.0091838
24	0.000049	393.16986	334.603774	122.22642	9.148263	850.00005	8194.356573	0.0094878
25	0.000016	393.169844	334.603761	122.22641	9.148263	850.00002	8194.356265	0.0097918
26	0.000001	393.169836	334.603755	122.22641	9.148263	850	8194.356111	0.0100979
27	0.000007	393.16984	334.603758	122.22641	9.148263	850.00001	8194.356188	0.0104188
28	0.000003	393.169838	334.603756	122.22641	9.148263	850	8194.356149	0.0107309
29	0.000001	393.169837	334.603756	122.22641	9.148263	850	8194.35613	0.0110468

En la Tabla 4.8 se puede ver que se aplica 29 iteraciones para llegar a la solución más óptima su tiempo de ejecución fue muy rápido de 0.01104689 y su costo total fue de 8194.35613 cabe señalar que en las primeras iteraciones, el método no fue tan preciso, la

mejora tomando en cuenta de la iteración 9 a la 29 tuvo una mejora en el costo de 0.0006%.

4.3. Propuesta de solución al problema de DE usando la iteración lambda y MATLAB

Propuesta de solución al problema de DE con los datos de la sección usando iteración lambda:

Usando el ejemplo de tres generadores de la sección 2.1.

Usando la ecuación 2.4, las condiciones para un despacho óptimo son:

$$\frac{dF_1}{dP_1} = 7.92 + 0.003124P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 7.85 + 0.00388P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 7.97 + 0.00964P_3 = \lambda$$

Se tienen las ecuaciones derivadas

$$\lambda_1 = 8$$

Se tiene una lambda inicial de 8

$$P_1 = \frac{\lambda_1 - 7.92}{0.003124}$$

$$P_1 = \frac{8 - 7.92}{0.003124} = 25.6081$$

$$P_2 = \frac{\lambda_1 - 7.85}{0.00388}$$

$$P_2 = \frac{8 - 7.85}{0.00388} = 38.6597$$

$$P_3 = \frac{\lambda_1 - 7.97}{0.00964}$$

$$P_3 = \frac{8 - 7.97}{0.00964} = 3.1120$$

Se sustituye la lambda inicial para obtener nuestros valores de P_1, P_2, P_3 .

$$P_T = P_1 + P_2 + P_3$$

$$P_T = 67.3798$$

Se suma P_1, P_2, P_3 para tener nuestra P_T .

$$\varepsilon_1 = P_L - P_T = 850 - 67.3798 = 782.6202$$

Se verifica si P_L es mayor o menor que P_T .

$$\text{Si } P_L > P_T$$

$$\lambda = \lambda * 1.1 \quad \lambda_2 = 8.8$$

Multiplicamos lambda por el valor de 1.1

$$\text{Si } P_L < P_T$$

$$\lambda = \lambda * 0.9$$

$$P_1 = \frac{\lambda_2 - 7.92}{0.003124}$$

$$P_1 = \frac{8.8 - 7.92}{0.003124} = 281.6901$$

$$P_2 = \frac{\lambda_2 - 7.85}{0.00388}$$

$$P_2 = \frac{8.8 - 7.85}{0.00388} = 244.8453$$

$$P_3 = \frac{\lambda_2 - 7.97}{0.00964}$$

$$P_3 = \frac{8.8 - 7.97}{0.00964} = 86.0995$$

Se calculamos los valores de P_1, P_2, P_3 con la nueva lambda

$$P_T = P_1 + P_2 + P_3$$

$$P_T = 612.6349$$

Se suma P_1, P_2, P_3 para tener P_T .

$$\varepsilon_2 = P_L - P_T = 850 - 612.6349 = 237.3651$$

$$y - y_1 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right) (x - x_1)$$

El procedimiento se repite mientras la tolerancia sea mayor que el error.

El código en MATLAB de la iteración lambda para solucionar DE

En la primera parte del código [Wood Allen. J, Wollenberg Bruce F Sheblé Gerald 2014] de la línea 1 a la 4, se declara las variables iniciando por el error, ϵ que es la tolerancia, p_1 que es la carga máxima, l_1 que es el valor de la lambda que la inicial.

```
1 error = 1.0;  
2 epsilon = 1e-6;  
3 p1 = 850;  
4 l1 = 8;
```

En esta siguiente del código que va de la línea 5 a la 7 se calculan los valores de p_1 , p_2 y p_3 sustituyendo el valor de lambda.

```
5 p1 = (l1 - 7.92)/0.003124;  
6 p2 = (l1 - 7.85)/0.00388;  
7 p3 = (l1 - 7.97)/0.00964;
```

En la línea 8 se hace la suma de p_1 , p_2 y p_3 y obtenemos la potencia total.

```
8 pt = p1+p2+p3;
```

De la línea 10 a la 17 se verifica si p_1 es mayor o menor que pt dependiendo de si es mayor o menor la lambda se multiplicara ya sea por 1.1 o 0.9

```
10 e1 = p1 - pt;  
11  
12 if (p1 > pt)  
13   l2 = l1*1.1;  
14 end  
15 if (p1 < pt)  
16   l2 = l1*0.9;  
17 end
```

De la línea 18 a la 23 se obtiene una nueva lambda y se reemplaza para obtener nuevas p_1, p_2 y p_3 al igual que una pt (potencia total)

```
18 p1 = (l2 - 7.92)/0.003124;  
19 p2 = (l2 - 7.85)/0.00388;  
20 p3 = (l2 - 7.97)/0.00964;  
21 pt = p1+p2+p3;  
22 e2 = p1 - pt;  
23 tic
```

De la línea 24 a 43 de la siguiente parte del código, se verifica que cuando el error sea menor que la tolerancia la ejecución se detenga.

```
24 while (error > epsilon)
```

```

25 l3 = -e1*(l2-l1)/(e2-e1)+l1;
26 l3
27 p1 = (l3 - 7.92)/0.003124;
28 p2 = (l3 - 7.85)/0.00388;
29 p3 = (l3 - 7.97)/0.00964;
30 pt = p1+p2+p3;
31 e3 = pl - pt;
32 error = abs(e3);
33 l1 = l2;
34 l2 = l3;
35 e1 = e2;
36 e2 = e3;
37 ct1 = 561 + 7.92*(p1) + 0.001562*(p1)^2;
38 ct2 = 310 + 7.85*(p2) + 0.00194*(p2)^2;
39 ct3 = 78 + 7.97*(p3) + 0.00482*(p3)^2;
40 ct=ct1+ct2+ct3
41 toc
42 end
43 [toc cont p1 p2 p3 pt ct]

```

El programa mide el tiempo de ejecución y lo imprime junto con el contador de iteraciones los valores de p1, p2, p3 (potencias de cada unidad) y pt (potencia total) y el costo total.

Código de iteración lambda

En la Tabla 4.9 se muestra el código en Matlab para solucionar el problema de DE usando Matlab. Los datos mostrados corresponden al problema descrito al inicio de la sección 4.1.

Tabla 4.9. Código de la iteración lambda en MATLAB.

```

1 % iteración lambda
2 error = 1.0; % error
4 epsilon = 1e-6; % tolerancia
5 pl = 850; % carga máxima
6 l1 = 8; % lambda inicial
7 cont = 0;
8 p1 = (l1 - 7.92)/0.003124; % valor de generador p1
9 p2 = (l1 - 7.85)/0.00388; % valor de generador p2
10 p3 = (l1 - 7.97)/0.00964; % valor de generador p3
11 pt = p1+p2+p3; % suma de nuestros generadores
12 e1 = pl - pt;
13 if (pl > pt) % condición de si pl es mayor que pt se multiplica por 1.1
14 l2 = l1*1.1;
15 end
16 if (pl < pt) % condición de si pl es menor que pt se multiplica por 0.9
17 l2 = l1*0.9;
18 end
19 % l2 es el valor de la nueva lambda que se calculo
20 p1 = (l2 - 7.92)/0.003124; % nuevo valor del generador P1 calculados con el valor de la nueva lambda
21 p2 = (l2 - 7.85)/0.00388; % nuevo valor del generador P2 calculados con el valor de la nueva lambda
22 p3 = (l2 - 7.97)/0.00964; % nuevo valor del generador P3 calculados con el valor de la nueva lambda
23 pt = p1+p2+p3;

```

```

24 e2 = pl - pt;
25 tic%medir el tiempo de ejecución
26 while (error > epsilon) % se verifica que el error no sea mayor que la tolerancia
27     l3 = -e1*(l2-l1)/(e2-e1)+l1;% se obtiene una nueva lambda
28     p1 = (l3 - 7.92)/0.003124; % nuevo valor del generador P1 calculados con el valor de la nueva lambda
29     p2 = (l3 - 7.85)/0.00388;% nuevo valor del generador P2 calculados con el valor de la nueva lambda
30     p3 = (l3 - 7.97)/0.00964; % nuevo valor del generador P3 calculados con el valor de la nueva lambda
31     pt = p1+p2+p3; % la suma de nuestros generadores p1, p2 y p3
32     e3 = pl - pt; % error
33     error = abs(e3); verifica el valor absoluto
34     cont = cont + 1;
35     l1 = l2; % iguala l1 con l2
36     l2 = l3; % iguala l2 con l3
37     e1 = e2; % iguala e1 con e2
38     e2 = e3; % iguala e2 con e3
39     ct1 = 561 + 7.92*(p1) + 0.001562*(p1)^2;% se calcula el costo del generador p1
40     ct2 = 310 + 7.85*(p2) + 0.00194*(p2)^2;% se calcula el costo del generador p2
41     ct3 = 78 + 7.97*(p3) + 0.00482*(p3)^2;% se calcula el costo del generador p3
42     ct=ct1+ct2+ct3 % el costo total de los 3 generadores
43 end
44 [toc cont p1 p2 p3 pt ct] % lo que nos imprime al ejecutarlo

```

En la Tabla 4.10 se observa la solución usando iteración lambda al ejecutar el programa para el ejemplo 2.1 utilizando una lambda inicial de 8:

Tabla 4.10. Resultado de la iteración lambda del Ejemplo 2.1 con MATLAB.

Contador	P1 MW	P2 MW	P3 MW	Pt MW	Costo total (\$/h)	Tiempo (seg)
1	393.169837	334.603755	122.22640	850.0000	8194.3561212	0.000422

En la Tabla 4.11 se observan los resultados obtenidos comparando el método simplex la búsqueda binaria y la iteración lambda

Tabla 4.11. Resultados óptimos de los tres métodos.

Método	Segmentos o contador	P1 MW	P2 MW	P3 MW	Costo total (\$/h)	Tiempo En seg
simplex	2000	393.18	334.60	122.22	8194.3561213	0.76095077
Búsqueda binaria	29	393.169837	334.603756	122.22641	8194.35613	0.0110468
Iteración lambda	1	393.169837	334.603755	122.2264	8194.3561212	0.000422

*El método simplex es el único que tiene segmentos en lugar de contador de iteraciones.

En la Tabla 4.11 se puede observar los resultados a los que llegaron los tres métodos siendo la iteración lambda el que nos entrega mejores valores, solución más óptima por una pequeña diferencia con respecto al costo, pero en cuanto a precisión y

respetar las condiciones el simplex y la iteración lambda fueron muy precisos pero con una mejor optimización el método de la iteración lambda.

4.5 Conclusiones

Al resolver el ejemplo 2.1 por los tres métodos que se están utilizando en la presente Tesis la iteración lambda ha sido la que tiene una mejor optimización en comparación al método simplex y al método de la búsqueda binaria. La búsqueda binaria en comparación con la iteración lambda tiene un tiempo de ejecución mayor en dar la solución al problema de DE. La búsqueda binaria tiene una menor precisión en comparación con la iteración lambda al determinar el costo total que es de 0.000009, apenas se nota la diferencia, es mínima, pero es un ahorro y es una mejor optimización. El método simplex fue el segundo mejor en cuanto al costo solo por una diferencia de 0.00000001. En cuanto a tiempo tiene un mayor tiempo de ejecución con un segmento de 2000. Si el segmento hubiera sido aún mucho más grande podría ser que tuviera un mejor costo y también un mayor tiempo de ejecución. Es decir, el costo se reduce y las potencias entregadas por los generadores tendrían una variación no muy perceptible, ya que solo se modifican las cifras correspondientes a las diezmilésimas en los valores de las potencias, incrementándose el tiempo de ejecución. El mejor método fue la iteración lambda seguido por el método simplex y en último lugar, la búsqueda binaria en cuanto a la optimización del costo total.

Capítulo 5. Casos de Estudio.

5.1 Introducción

En este capítulo se verá los casos de estudio que fueron seleccionados para ser analizados, se observará el comportamiento que tiene cada uno al ser comparado con cada caso de los métodos que son el método simplex la búsqueda binaria y la iteración lambda. Se obtendrá cual es el método más eficiente para cada caso dependiendo de los parámetros tomados para valorar cual llega a la solución óptima de una manera más eficiente.

5.2 Caso de Estudio No. 1

Considerar un sistema de tres unidades con las siguientes curvas de costos expresadas como cuadráticos, y los rangos de generación mínimos y máximos asociados. Se convierte cada curva de costo en una expresión lineal por partes utilizando tres variables por curva.

Curva de entrada-salida

$$H_1 \left(\frac{MBtu}{h} \right) = 213.1 + 11.669P_1 + 0.00533P_1^2$$

Curva de entrada-salida

$$H_1 \left(\frac{MBtu}{h} \right) = 200 + 10.333P_2 + 0.00889P_2^2$$

Curva de entrada-salida

$$H_1 \left(\frac{MBtu}{h} \right) = 240 + 10.833P_3 + 0.00741P_3^2$$

Unidad 1: Costo del combustible= 1.0\$/MBtu

Unidad 2: Costo del combustible= 1.0\$/MBtu

Unidad 3: Costo del combustible= 1.0\$/MBtu

Enseguida se presenta las curvas de costo H*Costo

$$F_1(P_1) = H_1(P_1) * 1.0 = 213.1 + 11.669P_1 + 0.00533P_1^2 \$ / h$$

$$F_2(P_2) = H_2(P_2) * 1.0 = 200 + 10.333P_2 + 0.00889P_2^2 \$ / h$$

$$F_3(P_3) = H_3(P_3) * 1.0 = 240 + 10.833P_3 + 0.00741P_3^2 \$/h$$

$$F_1(P_1) = 213.1 + 11.669P_1 + 0.00533P_1^2$$

$$50 \leq P_1 \leq 200$$

$$F_2(P_2) = 200 + 10.333P_2 + 0.00889P_2^2$$

$$37.5 \leq P_2 \leq 150$$

$$F_3(P_3) = 240 + 10.833P_3 + 0.00741P_3^2$$

$$45 \leq P_3 \leq 180$$

$$PL=217.87$$

Entonces

$$F_1(P_1) = 213.1 + 11.669P_1 + 0.00533P_1^2$$

$$F_2(P_2) = 200 + 10.333P_2 + 0.00889P_2^2$$

$$F_3(P_3) = 240 + 10.833P_3 + 0.00741P_3^2$$

Usando la Ecuación 2.4, las condiciones para un despacho óptimo son

$$\frac{dF_1}{dP_1} = 11.669 + 0.0106P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 10.333 + 0.01778P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 10.833 + 0.01482P_3 = \lambda$$

$$PL = 217.87 \text{ MW}$$

Potencias máximas y mínimas de cada generador

Gen 1 50 a 200 MW

Gen 2 37.5 a 150 MW

Gen 3 45 a 180 MW

La Tabla 5.1 muestra los costos incrementales:

Tabla 5.1. Costos incrementales para el Caso de Estudio 1.

Min	Max
12.19900	13.7890
10.99975	13.0000
11.49990	13.5006

En la Tabla 5.2 se muestra la solución usando búsqueda binaria al ejecutar el programa para el caso 1, se puede observar el número de iteraciones que realiza, el tiempo, la potencia total, así como la de cada generador, el error la nueva lambda y el costo total.

Tabla 5.2. Resultado de la búsqueda binaria para el caso de estudio 1.

Cont	Error	P1 MW	P2 MW	P3 MW	Ln	Pt MW	Costo Total (\$/h)	Tiempo en (seg)
1	71.855393	68.431604	115.937852	105.355938	11.697063	289.725393	4017.640612	0.00055814
2	80.199856	2.647406	76.718926	58.303812	12.045719	137.670144	2185.885567	0.00134397
3	4.172231	35.539505	96.328389	81.829875	12.220047	213.697769	3088.476879	0.00194812
4	33.841581	51.985554	106.13312	93.592907	12.132883	251.711581	3549.737193	0.00231194
5	14.834675	43.762529	101.230754	87.711391	12.089301	232.704675	3318.276648	0.00267816
6	5.331222	39.651017	98.779571	84.770633	12.06751	223.201222	3203.169167	0.00303411
7	0.579495	37.595261	97.55398	83.300254	12.056614	218.449495	3145.771124	0.00338411
8	1.796368	36.567383	96.941184	82.565065	12.062062	216.073632	3117.111027	0.00374699
9	0.608437	37.081322	97.247582	82.932659	12.064786	217.261563	3131.437831	0.00410414
10	0.014471	37.338291	97.400781	83.116457	12.066148	217.855529	3138.603667	0.00445604
11	0.282512	37.466776	97.477381	83.208355	12.065467	218.152512	3142.187192	0.00480795
12	0.134021	37.402534	97.439081	83.162406	12.065126	218.004021	3140.395379	0.00515199
13	0.059775	37.370413	97.419931	83.139431	12.064956	217.929775	3139.49951	0.00533795
14	0.022652	37.354352	97.410356	83.127944	12.064871	217.892652	3139.051585	0.00552702
15	0.004091	37.346322	97.405569	83.1222	12.064828	217.874091	3138.827625	0.00571704
16	0.00519	37.342307	97.403175	83.119329	12.06485	217.86481	3138.715646	0.0059011
17	0.00055	37.344314	97.404372	83.120764	12.06486	217.86945	3138.771635	0.00608516
18	0.001771	37.345318	97.40497	83.121482	12.064855	217.871771	3138.79963	0.00626707
19	0.00061	37.344816	97.404671	83.121123	12.064852	217.87061	3138.785633	0.00644994
20	0.00003	37.344565	97.404521	83.120944	12.064851	217.87003	3138.778634	0.00663304
21	0.00026	37.34444	97.404447	83.120854	12.064852	217.86974	3138.775135	0.00681496
22	0.000115	37.344502	97.404484	83.120899	12.064852	217.869885	3138.776884	0.00699902
23	0.000042	37.344534	97.404503	83.120922	12.064852	217.869958	3138.777759	0.00717998
24	0.000006	37.344549	97.404512	83.120933	12.064852	217.869994	3138.778197	0.00736308
25	0.000012	37.344557	97.404517	83.120938	12.064852	217.870012	3138.778415	0.00754595
26	0.000003	37.344553	97.404514	83.120936	12.064852	217.870003	3138.778306	0.0077281
27	0.000001	37.344551	97.404513	83.120934	12.064852	217.869999	3138.778251	0.00791812
28	0.000001	37.344552	97.404514	83.120935	12.064852	217.870001	3138.778279	0.00810599

En la Tabla 5.2 se observa como en la primera iteración no se cumplen las restricciones, se empezaron a cumplir a partir de la iteración número 13, después de esta iteración se puede observar como los valores va siendo casi idénticos pero varían en sus milésimas tardo 28 iteraciones en llegar al resultado más óptimo y tuvo una mejora de 0.0002 de la iteración 13 a la 28.

En la Tabla 5.3 se muestra la solución usando iteración lambda al ejecutar nuestro programa para el caso 1. Con una lambda inicial de 8, Se puede observar que nos entrega el tiempo, un contador, los valores de cada generador, la potencia total y el costo total.

Tabla 5.3. Resultado de la iteración lambda para el Caso de estudio 1.

Contador	P1 MW	P2 MW	P3 MW	Pt MW	Costo total (\$/h)	Tiempo en (seg)
2	37.3445519	97.4045135	83.1209346	217.87	3138.778268	0.0007608

En la Tabla 5.4, se muestran los resultados obtenidos con Matlab usando el método simplex. Como puede observarse, la solución óptima se encuentra con un costo total $Z = 3140.30755207$ con un número de 2000 segmentos. Dicha solución se encuentra muy cerca de las obtenidas con los métodos de búsqueda binaria con $Z = 3138.778279$ y con el método de Iteración lambda con $Z = 3138.778268$.

Tabla 5.4. Resultados con Método Simplex para Caso de Estudio 1.

No. Segmentos	P1 MW	P2 MW	P3 MW	Costo Total (\$/h)	Tiempo(s) En seg
1	50.00	122.87	45.00	3156.19361804	0.02033301
2	50.00	93.75	74.12	3140.37934401	0.01723210
3	50.00	77.87	90.00	3143.40333104	0.01793389
5	50.00	95.87	72.00	3140.59764584	0.01670146
10	50.00	93.75	74.12	3140.37934393	0.01977513
25	50.00	91.50	76.37	3140.30792536	0.02148253
50	50.00	91.50	76.37	3140.30792544	0.02415585
75	50.00	91.50	76.37	3140.30792533	0.03006097
100	50.00	91.82	76.05	3140.30801591	0.03292914
200	50.00	91.82	76.05	3140.30801576	0.04909108
250	50.00	91.55	76.32	3140.30771941	0.05875389
500	50.00	91.72	76.14	3140.30764065	0.10639460
750	50.00	91.65	76.22	3140.30755207	0.19133865
1000	50.00	91.68	76.18	3140.30757116	0.29333416
1250	50.00	91.66	76.21	3140.30755337	0.45053730
1500	50.00	91.65	76.22	3140.30755207	0.60621071
1750	50.00	91.63	76.24	3140.30756050	0.73164489
2000	50.00	91.67	76.20	3140.30755699	0.82671724

En la Tabla 5.4 del segmento numero 1 al 2000 tuvo una mejora del 5% con respecto al costo.

A continuación se muestra las soluciones óptimas para el caso 1 por medio del método simplex, búsqueda binaria e iteración lambda.

Tabla 5.5. Resultados de los tres métodos para el Caso de Estudio 1.

Método	Segmento o contador	P1 MW	P2 MW	P3 MW	Costo total (\$/h)	Tiempo en seg
Simplex	2000	50.00	91.67	76.20	3140.30755699	0.82671724
Búsqueda binaria	28	37.344552	97.404514	83.120935	3138.778279	0.00810599
Iteración lambda	2	37.3445519	97.4045135	83.1209346	3138.778268	0.0007608

En la Tabla 5.5 podemos observar un comparativo con los mejores valores a los que pudieron llegar los tres métodos y el mejor para el caso uno fue el método de la iteración lambda. Tuvo los mejores valores entregados por los generadores con el mejor costo total. En un tiempo muy rápido en comparación a los otros dos métodos, la mejora que tuvo con respecto al valor del costo total fue de 0.0004%.

5.2 Caso de Estudio No. 2

Considerar un sistema de cinco unidades con las siguientes curvas de costos expresadas como cuadráticos, y los rangos de generación mínimos y máximos asociados. En seguida se convierte cada curva de costo en una expresión lineal por partes utilizando tres variables por curva.

Curva de entrada-salida

$$H_1 \left(\frac{MBtu}{h} \right) = 610 + 7.3P_1 + 0.00144P_1^2$$

Curva de entrada-salida

$$H_2 \left(\frac{MBtu}{h} \right) = 300 + 7.8P_2 + 0.00195P_2^2$$

Curva de entrada-salida

$$H_3 \left(\frac{MBtu}{h} \right) = 100 + 7.9P_3 + 0.0048P_3^2$$

Curva de entrada-salida

$$H_4 \left(\frac{MBtu}{h} \right) = 200 + 8.1P_4 + 0.00125P_4^2$$

Curva de entrada-salida

$$H_5 \left(\frac{MBtu}{h} \right) = 180 + 8.4P_5 + 0.00160P_5^2$$

Unidad 1: Costo del combustible= 1.0\$/MBtu

Unidad 2: Costo del combustible= 1.0\$/MBtu

Unidad 3: Costo del combustible= 1.0\$/MBtu

Unidad 4: Costo del combustible= 1.0\$/MBtu

Unidad 5: Costo del combustible= 1.0\$/MBtu

Entonces

$$F_1(P_1) = H_1(P_1) * 1.0 = 610 + 7.3P_1 + 0.00144P_1^2 \$/h$$

$$F_2(P_2) = H_2(P_2) * 1.0 = 300 + 7.8P_2 + 0.00195P_2^2 \$/h$$

$$F_3(P_3) = H_3(P_3) * 1.0 = 100 + 7.9P_3 + 0.0048P_3^2 \$/h$$

$$F_4(P_4) = H_4(P_4) * 1.0 = 200 + 8.1P_4 + 0.00125P_4^2 \$/h$$

$$F_5(P_5) = H_5(P_5) * 1.0 = 180 + 8.4P_5 + 0.00160P_5^2 \$/h$$

Realizando la multiplicación

$$F_1(P_1) = 610 + 7.3P_1 + 0.00144P_1^2$$

$$100 \leq P_1 \leq 600$$

$$F_2(P_2) = 300 + 7.8P_2 + 0.00195P_2^2$$

$$150 \leq P_2 \leq 700$$

$$F_3(P_3) = 100 + 7.9P_3 + 0.0048P_3^2$$

$$125 \leq P_3 \leq 500$$

$$F_4(P_4) = 200 + 8.1P_4 + 0.00125P_4^2$$

$$200 \leq P_4 \leq 800$$

$$F_5(P_5) = 180 + 8.4P_5 + 0.00160P_5^2$$

$$80 \leq P_5 \leq 300$$

$$PL=1600$$

Usando la ecuación 2.4, las condiciones para un despacho óptimo son:

$$\frac{dF_1}{dP_1} = 7.3 + 0.00288P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 7.8 + 0.0039P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 7.9 + 0.0096P_3 = \lambda$$

$$\frac{dF_4}{dP_4} = 8.1 + 0.0025P_1 = \lambda$$

$$\frac{dF_5}{dP_5} = 8.4 + 0.0032P_1 = \lambda$$

Tabla 5.6. Costos incrementales para el Caso de Estudio 2.

Min	Max
7.588	9.028
8.385	10.530
9.100	12.700
8.600	10.100
8.656	9.360

En la Tabla 5.7 se muestra la solución usando búsqueda binaria al ejecutar nuestro programa para el caso de estudio 2, para este caso se observa que hay 5 generadores, tiempo, contador, potencia total y el costo total.

Tabla 5.7. Resultados con búsqueda binaria para Caso de Estudio 2.

Cont	P1 MW	P2 MW	P3 MW	P4 MW	P5 MW	Pt MW	Costo total (\$/h)	Tiempo en (seg)
1	987.5	601.025641	233.75	817.6	545	3184.875641	30015.65675	0.00055313
2	543.75	273.333333	100.625	306.4	145.625	1369.733333	12762.72912	0.00137115
3	765.625	437.179487	167.1875	562	345.3125	2277.304487	21099.22395	0.00194693
4	654.6875	355.25641	133.90625	434.2	245.46875	1823.51891	16858.48429	0.002424
5	599.21875	314.294872	117.265625	370.3	195.546875	1596.626122	14792.48364	0.00265312
6	626.953125	334.775641	125.585937	402.25	220.507812	1710.072516	15820.9532	0.00287604
7	613.085937	324.535256	121.425781	386.275	208.027344	1653.349319	15305.58573	0.00310612
8	606.152344	319.415064	119.345703	378.2875	201.787109	1624.98772	15048.75151	0.00333095
9	602.685547	316.854968	118.305664	374.29375	198.666992	1610.806921	14920.54678	0.00355506
10	600.952148	315.57492	117.785645	372.296875	197.106934	1603.716521	14856.49751	0.00377703
11	600.085449	314.934896	117.525635	371.298437	196.326904	1600.171322	14824.48615	0.00399899
12	599.6521	314.614884	117.39563	370.799219	195.93689	1598.398722	14808.48379	0.00422215
13	599.868774	314.77489	117.460632	371.048828	196.131897	1599.285022	14816.4847	0.00445414
14	599.977112	314.854893	117.493134	371.173633	196.229401	1599.728172	14820.48536	0.00467896
15	600.031281	314.894894	117.509384	371.236035	196.278152	1599.949747	14822.48574	0.00490093
16	600.058365	314.914895	117.517509	371.267236	196.302528	1600.060534	14823.48594	0.00512409

Cont	P1 MW	P2 MW	P3 MW	P4 MW	P5 MW	Pt MW	Costo total (\$/h)	Tiempo en (seg)
17	600.044823	314.904895	117.513447	371.251636	196.29034	1600.00514	14822.98584	0.00534606
18	600.038052	314.899895	117.511415	371.243835	196.284246	1599.977443	14822.73579	0.00556803
19	600.041437	314.902395	117.512431	371.247736	196.287293	1599.991292	14822.86081	0.00579214
20	600.04313	314.903645	117.512939	371.249686	196.288817	1599.998216	14822.92332	0.00601506
21	600.043976	314.90427	117.513193	371.250661	196.289579	1600.001678	14822.95458	0.00626707
22	600.043553	314.903957	117.513066	371.250173	196.289198	1599.999947	14822.93895	0.00650597
23	600.043765	314.904113	117.513129	371.250417	196.289388	1600.000813	14822.94677	0.00673103
24	600.043659	314.904035	117.513098	371.250295	196.289293	1600.00038	14822.94286	0.00697207
25	600.043606	314.903996	117.513082	371.250234	196.289245	1600.000164	14822.94091	0.00720716
26	600.04358	314.903977	117.513074	371.250204	196.289222	1600.000055	14822.93993	0.00743294
27	600.043566	314.903967	117.51307	371.250188	196.28921	1600.000001	14822.93944	0.007658
28	600.04356	314.903962	117.513068	371.250181	196.289204	1599.999974	14822.9392	0.00788307
29	600.043563	314.903964	117.513069	371.250185	196.289207	1599.999988	14822.93932	0.00810695
30	600.043565	314.903966	117.513069	371.250187	196.289208	1599.999995	14822.93938	0.00833011
31	600.043566	314.903966	117.51307	371.250187	196.289209	1599.999998	14822.93941	0.00855303
32	600.043566	314.903967	117.51307	371.250188	196.289209	1600	14822.93942	0.00877595

En la Tabla 5.7 se observa que en las primeras iteraciones el método no respeta las restricciones hasta la 23 iteración los valores cumplen las condiciones este método para el caso dos tardo 32 iteraciones en entregar la solución más óptima, con una mejora de 0.0000004 de la iteración 23 a la 39.

En la Tabla 5.8 solución usando iteración lambda al ejecutar nuestro programa para el caso de estudio 2 con el valor inicial de nuestra lambda de 8 y nos muestra, el tiempo el contador, 5 generadores la potencia total y el costo total.

Tabla 5.8. Resultados con iteración lambda para Caso de Estudio 2.

cont	P1 MW	P2 MW	P3 MW	P4 MW	P5 MW	Pt MW	Costo total (\$/h)	Tiempo en (seg)
1	600.043566	314.903967	117.51307	371.25018	196.28929	1600	14822.9394291284	0.0003550

Solución al Caso de Estudio 2 mediante el Método Simplex.

Usando el Método Simplex para solucionar el Caso de Estudio 2, se obtuvieron los resultados que se presentan en la Tabla 30. Como puede observarse, la solución óptima se encuentra con un costo total $Z = 14823.22985113$ con 2000 números de segmentos. Debe mencionarse que esta solución también se encuentra muy cercana a los resultados obtenidos con los otros dos métodos. Tal es el caso de la de búsqueda binaria con $Z = 14822.9394256454$ y el método de Iteración lambda con $Z = 14822.9394291284$.

Tabla 5.9. Resultados con Método simplex para Caso de Estudio 2.

No. Segmentos	P1 MW	P2 MW	P3 MW	P4 MW	P5 MW	Costo Total (\$/h)	Tiempo(s) En seg
1	600.00	150.00	125.00	425.00	300.00	14897.05624812	0.02186333
2	600.00	425.00	125.00	260.00	190.00	14862.37875000	0.01821342
3	600.00	333.33	125.00	388.33	153.33	14827.18791667	0.02022937
5	600.00	343.00	125.00	320.00	212.00	14828.42595000	0.01895608
10	600.00	315.00	125.00	370.00	190.00	14823.27375000	0.02243187
25	600.00	312.60	125.00	368.00	194.40	14823.23775801	0.02289855
50	600.00	315.00	125.00	368.00	192.00	14823.25114262	0.03327026
75	600.00	311.33	125.00	368.00	195.67	14823.24717778	0.03624207
100	597.60	315.00	125.00	368.00	194.40	14823.23602040	0.04325466
200	597.50	315.00	125.00	368.06	194.44	14823.23600835	0.07502184
250	598.00	312.80	125.00	369.80	194.40	14823.23147416	0.09215503
500	598.00	313.40	125.00	369.20	194.40	14823.22987800	0.23648311
750	598.00	313.53	125.00	368.80	194.67	14823.22988246	0.40300616
1000	598.00	313.35	125.00	369.20	194.45	14823.22987804	0.59055868
1250	598.00	313.46	125.00	368.96	194.58	14823.22979827	0.98427631
1500	598.00	313.53	125.00	368.92	194.55	14823.22981149	1.44316227
1750	598.02	313.43	125.00	369.03	194.53	14823.22979236	1.77711863
2000	598.24	313.35	125.00	368.90	194.51	14823.22985113	2.21487206

En la Tabla 5.9 tuvo una mejora de 0.004% del segmento 1 al 2000 para el caso del método simplex.

A continuación se muestra los resultados más óptimos para el caso de estudio 2 que tuvieron el método simplex la búsqueda binaria y la iteración lambda.

Tabla 5.10. Resultados de los 3 métodos para el Caso de Estudio 2.

Método	Seg o con	P1 MW	P2 MW	P3 MW	P4 MW	P5 MW	Costo total (\$/h)	Tiempo en seg
Simplex	2000	598.24	313.35	125.00	368.90	194.51	14823.22985113	2.21487206
Búsqueda binaria	32	600.043566	314.903967	117.51307	371.250188	196.2892	14822.9394256454	0.00877595
Iteración lambda	1	600.043566	314.903967	117.51307	371.250188	196.2892	14822.9394291284	0.00035501

En la tabla 5.10 podemos observar que para el caso 2 el método de la búsqueda binaria fue más eficiente, el método de la iteración lambda casi obtuvo los mismos valores en los resultados entregados por nuestros generadores pero tuvo un mayor costo y un menor tiempo que la búsqueda binaria, la mejora entre el mayor y menor es de 0.00001%.

5.3 Conclusiones

En este Capítulo se analizaron dos casos de estudio de los cuales se resolvieron por medio de los métodos simplex, búsqueda binaria e iteración lambda. Para el primer caso de estudio, el método de iteración lambda tuvo la mejor optimización con el mejor costo de 3138.778268, el menor tiempo de ejecución de 0.0007608 segundos, con solo dos iteraciones. El método binario tuvo el segundo mejor desempeño en cuanto a costo y tiempo de ejecución, con un costo de 3138.778279 y un tiempo de 0.00810599 segundos y por último el método simplex con un costo de 3140.3075569 y un tiempo de 0.8267172 segundos. Así la iteración lambda, para el primer caso de estudio fue el método que llegó a una mejor optimización.

En el segundo caso el método de la búsqueda binaria y el método de la iteración lambda obtuvieron resultados muy similares al tener la mejor optimización. Para este caso, el mejor método con respecto al costo total fue el de la búsqueda binaria, con un costo 14822.9394256454 en comparación a la iteración lambda, cuyo costo total fue de 14822.9394291284 con una pequeña diferencia de 0.000003483 y con respecto al método simplex donde el costo fue de 14823.22985113. Donde hubo una diferencia más notable de 0.29043. Con respecto al tiempo de ejecución, la iteración lambda fue el método más rápido con 0.00035501 segundos, seguida la búsqueda binaria con 0.00877595 segundos y por último el método simplex con 2.21487206 segundos. En cuanto al costo de cada generador, la búsqueda binaria y la iteración lambda fueron iguales, pero lo que nos importa es tener una mejor optimización en costos, por tanto, se puede decir que el método de la búsqueda binaria fue el mejor para este segundo caso.

El método simplex para cualquiera de los dos casos obtuvo mayor tiempo de ejecución, pero entre más grande sea el número de segmentos el costo total va decreciendo y aumentando el tiempo de ejecución.

Capítulo 6. Conclusiones y recomendaciones para trabajos de investigación futuros.

6.1 Conclusiones

Los modelos de despacho económico son altamente necesarios para poder aprovechar al máximo los recursos disponibles con la menor pérdida de recursos y de dinero. Nos permite obtener las mejores maneras de distribución siempre llegando a un punto óptimo, que satisfaga las necesidades y sin romper con las restricciones que limitan a los diferentes modelos de despacho económico.

El método simplex es un algoritmo eficiente y confiable para resolver problemas de PL ya sea si necesite maximizar o minimizar la función objetivo. Para esta tesis, el método simplex llego a buenos resultados, pero no se pueden comparar a los que llego la iteración lambda y la búsqueda binaria, lo cual no significa que no sea un buen método. Para la presente Tesis los valores otorgados por el método simplex no fueron los mejores, pero si muy cercanos a los más óptimos. No se pudo acercar más, ya que se necesita una cantidad muy grande de segmentos. Se observa que a la computadora se le agotaba la memoria debido al número de operaciones que realizaba. Para realizar más cálculos se requeriría una computadora más potente, que permita llegar a una mejor solución para el método simplex. Además, cada vez que se incrementa el número de segmentos, el costo total decrece. Lo cual es favorable, pero el valor apenas se va diferenciado, tendría que ser un número grande de segmentos para poder apreciar un costo menor.

El ejemplo 2.1 fue resuelto por los tres métodos para ver de qué manera funciona cada método, que fórmulas están implicadas, las restricciones que conlleva cada uno y así poder observar y comparar los resultados entregados por los diferentes métodos planteados. Se resolvió el ejemplo 2.1 y se observa que los valores entregados por la iteración lambda y el método simplex fueron muy similares con respecto al costo total por una diferencia de 0.00000001. Los dos tuvieron un buen valor óptimo, pero la iteración lambda tuvo mejor resultado.

En los dos casos de estudio se probaron los tres métodos la búsqueda binaria, la iteración lambda y el método simplex. Para el primer Caso de Estudio hubo dos métodos que entregaban valores similares, solo con pequeñas diferencias que fueron el método de la búsqueda binaria y la iteración lambda, en cuanto a los valores entregados por los generadores fueron casi prácticamente iguales, el costo total solo tenía una diferencia de 0.00734519, pero aún así nos indica que un método llegó a la mejor optimización y para el Caso 1 fue el método de la iteración lambda. En cuestión de tiempo fue el más rápido en resolverlo, entregando los mejores resultados. El método simplex estuvo cerca de los

otros dos métodos. Sin embargo, presentó una variación de 2MW, siendo perceptible la diferencia. Y es más tardado en su tiempo de ejecución.

La mejor optimización del costo de producción la obtuvo el método de la iteración lambda seguido por la búsqueda binaria y el método simplex.

Para el Caso de Estudio 2, nuevamente, dos de los tres métodos estuvieron a la par sus resultados casi fueron idénticos, dichos métodos fueron la búsqueda binaria y la iteración lambda. Los sus resultados en cuestión a la potencia generada por cada generador fueron idénticos, diferenciándose en el costo, tiempo y número de iteraciones que realizaron. Para este Segundo Caso el método de la búsqueda binaria fue mejor, ya que tuvo una mejor optimización en cuestión al costo la diferencia fue 0.000003483 pero en cuanto a iteraciones y tiempo de ejecución fue mejor la iteración lambda. El mejor método fue la búsqueda binaria seguido de la iteración lambda y por último el método simplex.

Cabe resaltar que el método simplex podría tener mejores resultados, pero como se necesita tener muchos segmentos y en las computadoras comunes la memoria se desborda y llega a un límite y no puede hacer más operaciones. Con una computadora especial se podrían tener mejores resultados. Si se pudiera usar una computadora de más alto rendimiento el método simplex podría tener una mejor optimización. Con mayor número segmentos serian mejores los valores obtenidos por el método simplex, pero a su vez el tiempo de ejecución sería muy grande.

La conclusión de la presente Tesis, debido a la investigación realizada es que el método de la iteración lambda es el que otorga mejores resultados, en menor tiempo es el mejor método en esta Tesis, más que cualquiera de los otros dos métodos En el Ejemplo 2.2, se puede observar que entre el método simplex y la iteración lambda se obtuvieron valores de costo total muy cercanos, de tal manera que es muy difícil diferenciar los valores, ya que las variaciones en cifras se presentan en el orden de las diezmilésimas. La búsqueda binaria también entrego buenos resultados que en los casos era muy difícil de ver las pequeñas diferencias en los resultados en las diezmilésimas, es también un excelente método, pero la iteración lambda fue el que llego con un valor óptimo en los dos casos y el en el Ejemplo 2.1.

6.2 Recomendaciones para trabajos futuros

Este trabajo de Tesis tiene un gran potencial para seguir siendo desarrollada en un futuro ya que el despacho económico es algo que nos ayuda a tener una solución más óptima en el despacho de energía en una red. Los métodos que son utilizados son algunos de muchos que pueden ser escogidos. Para la solución del DE todos estos métodos pueden

ser programados en diferentes lenguajes así se podría comparar la velocidad de ejecución de cada uno y ver si nos llevan a una solución óptima más rápido teniendo en cuenta las restricciones del despacho económico así como la de los métodos que se puede usar para tener un mejor despacho económico.

Podrían realizar el método simplex con diferentes lenguajes de programación y con computadoras más potentes en cuestión de capacidad de memoria y así pueda realizar muchas operaciones y no tener una limitante de memoria, los resultados que podría entregar podrían ser comparados y verificar que tanto se podría optimizar los modelos de despacho económico usando el método simplex.

Para un trabajo futuro se podrían implementar la codificación de los métodos que se utilizaron en la presente tesis en el lenguaje de programación C y añadir técnicas en paralelo para el método simplex.

Referencias

- [Berrio kevin, 2016] Kevin Jesús Berrío Castro “Método de solución para el despacho económico en línea considerando restricciones y reglas de un mercado eléctrico” 2016.
- [Callaway, 2011] D. S. Callaway and I. A. Hiskens, "Achieving Controllability of Electric Loads," in Proceedings of the IEEE, vol. 99, no. 1, pp. 184-199, Jan. 2011.
- doi: 10.1109/JPROC.2010.2081652.
- [Chao, 2007] Chao-Hong Chen and Ying-ping Chen. 2007. Real-coded ECGA for economic dispatch. In Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07). ACM, New York, NY, USA, 1920-1927. DOI=<http://dx.doi.org/10.1145/1276958.1277343>.
- [Chapra Canale, 2006] C, Chapra Steven, P. Canale Raymond “Métodos numéricos para ingenieros” 2006.
- [Díaz Carla, 2017] Díaz Gonzales Carla “Despacho económico de cargas en sistemas eléctricos de potencia” 2017.
- [Flores silva et al., 2008] Flores Silva Ángel, Mendoza Monroy Manuel Alejandro y rosales Arroyo Fernando “El despacho económico en de un sistema termo eléctrico a corto plazo” 2008.
- [Garrido Yerco, 2016] Garrido Fuhrop Yerco Nicolás “Despacho económico para sistemas medianos” 2016.
- [Keles, 2007] Ali Keles. 2007. Binary differential evolution for the unit commitment problem. In Proceedings of the 9th annual conference companion on Genetic and evolutionary computation (GECCO '07). ACM, New York, NY, USA, 2765-2768. DOI=<http://dx.doi.org/10.1145/1274000.1274092>.
- [Larez Adan, 2003] Larez Córdova Adán Alberto “Despacho de energía en mercados competitivos” 2003.
- [Leal Elvis, 2005] Leal Cante Elvis Noé “Despacho económico de carga considerando restricciones en la red de transporte con el uso de técnicas de programación lineal” 2005.
- [Lubin, 2011] Miles Lubin, Cosmin G. Petra, Mihai Anitescu, and Victor Zavala. 2011. Scalable stochastic optimization of complex energy systems. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11). ACM, New York, NY, USA, Article 64, 64 pages. DOI: <https://doi.org/10.1145/2063384.2063470>

- [Monteiro, 2009] Monteiro C., Bessa R.J., Miranda,V., Botterud A., Wang J., Conzelmann G.,WindPowerForecasting:StateoftheArt2009<https://anl.box.com/s/v8nyd3smx1gt0t96x5gr1ziljrvqa2>.ArgonneNationalLaboratoryArgonne, IL USAEng.
- [Narayanaswamy, 2012] Balakrishnan Narayanaswamy, Vikas K. Garg, and T. S. Jayram. 2012. Online optimization for the smart (micro) grid. In Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy '12). ACM, New York, NY, USA, , Article 19 , 10 pages. DOI: <http://dx.doi.org/10.1145/2208828.2208847>
- [Núñez Alejandro, 2013] Núñez Jiménez Alejandro “Despacho económico y optimización de cargas” 2013.
- [Revelle y colaboradores, 1997] Revelle, C.S., E.E. Whitlatch y J.R. Wright, Civil and Environmental Systems Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1997.
- [Sinha, 2003] N. Sinha, R. Chakrabarti and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," in IEEE Transactions on Evolutionary Computation, vol. 7, no. 1, pp. 83-94, Feb 2003. doi: 10.1109/TEVC.2002.806788
- [Thanos, 2013] Aristotelis E. Thanos, Xiaoran Shi, Juan P. Sáenz, and Nurcin Celik. 2013. A DDDAMS framework for real-time load dispatching in power networks. In Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World (WSC '13). IEEE Press, Piscataway, NJ, USA, 1893-1904.
- [Wordpress, 2013]<https://ingenierosenapuros.files.wordpress.com/2013/12/a-despachos-econoc3b3micosmtee-002.pdf>
- [Wood Allen. J, Wollenberg Bruce F Sheblé Gerald 2014] Wood Allen. J, Wollenberg Bruce F Sheblé Gerald Power Geration, Operation, nda control, third Edition, 2015
- [Ying-ping, 2010] Ying-ping Chen and Chao-Hong Chen. 2010. Enabling the extended compact genetic algorithm for real-parameter optimization by using adaptive discretization. Evol. Comput. 18, 2 (June 2010), 199-228. DOI=<http://dx.doi.org/10.1162/evco.2010.18.2.18202>
- [Yiqing, 2015] Yiqing Hua, Chao Li, Weichao Tang, Li Jiang, and Xiaoyao Liang. 2015. Building Fuel Powered Supercomputing Data Center at Low Cost. In Proceedings of the 29th ACM on International Conference on Supercomputing (ICS '15). ACM, New York, NY, USA, 241-250. DOI: <https://doi.org/10.1145/2751205.2751215>.

- [Zwe-Lee, 2003] Zwe-Lee Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," in IEEE Transactions on Power Systems, vol. 18, no. 3, pp. 1187-1195, Aug. 2003. doi: 10.1109/TPWRS.2003.814889