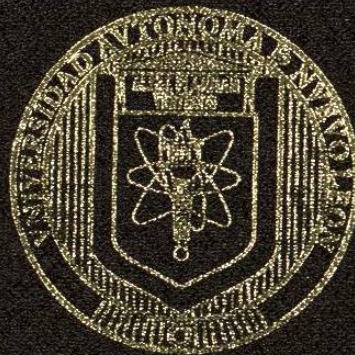


UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA
DIVISION DE ESTUDIOS DE POSTGRADO



MODELOS DE OPTIMIZACION ENTERA MIXTA NO LINEAL
EN SISTEMAS DE TRANSPORTE DE GAS NATURAL

POR:
ING. DIANA COBOS ZALETÁ

TESIS
EN OPCION AL GRADO DE MAESTRO EN
CIENCIAS EN INGENIERIA DE SISTEMAS

CD. UNIVERSITARIA

NOVIEMBRE 2003

MODELOS DE OPTIMIZACION ENTERA MIXTA NO LINEAL
SOLUCION DE TRANSPORTES DE GAS NATURAL
EN SISTEMAS DE TRANSPORTES DE GAS NATURAL

DCZ DCZ

TM
Z5853
.M2
FIME
2003
.C62



1020149292

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA
DIVISION DE ESTUDIOS DE POSTGRADO



MODELOS DE OPTIMIZACION ENTERA MIXTA NO LINEAL
EN SISTEMAS DE TRANSPORTE DE GAS NATURAL

POR:
ING. DIANA COBOS ZALETÁ

TESIS
EN OPCION AL GRADO DE MAESTRO EN
CIENCIAS EN INGENIERIA DE SISTEMAS

CD. UNIVERSITARIA

NOVIEMBRE 2003



980875

TH
Z 5853
.M2
FIME
2003
.C62

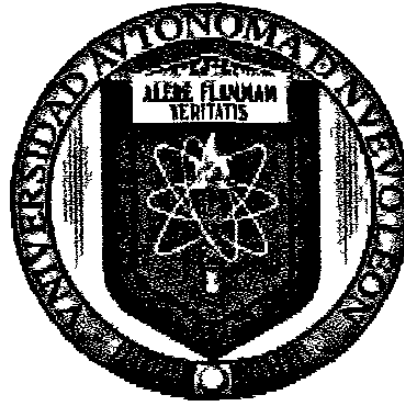


FONDO
TESIS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSTGRADO



**MODELOS DE OPTIMIZACIÓN ENTERA MIXTA NO LINEAL
EN SISTEMAS DE TRANSPORTE DE GAS NATURAL**

POR:

ING. DIANA COBOS ZALETA

TESIS

**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA DE SISTEMAS**

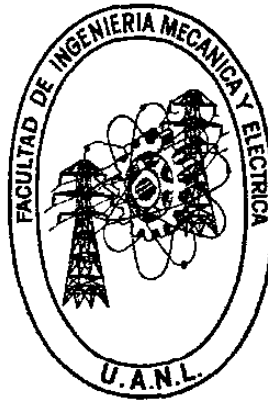
CIUDAD UNIVERSITARIA

NOVIEMBRE DE 2003

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSTGRADO



**MODELOS DE OPTIMIZACIÓN ENTERA MIXTA NO LINEAL
EN SISTEMAS DE TRANSPORTE DE GAS NATURAL**

POR:

ING. DIANA COBOS ZALETA

TESIS

**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA DE SISTEMAS**

CIUDAD UNIVERSITARIA

NOVIEMBRE DE 2003

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la tesis **“MODELOS DE OPTIMIZACIÓN ENTERA MIXTA NO LINEAL EN SISTEMAS DE TRANSPORTE DE GAS NATURAL”**, realizada por la alumna Diana Cobos Zaleta, matrícula 791729, sea aceptada para su defensa como opción al grado de Maestro en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



Dr. Roger Z. Ríos Mercado

Asesor



Dra. Ada M. Álvarez Socarrás

Revisor



Dr. Óscar L. Chacón Mondragón

Revisor

Vó. Bo.



Dr. G. Alan Castillo Rodríguez

Subdirector

División de Estudios de Posgrado

DEDICATORIA

Quiero dedicar este logro en mi vida profesional, a quienes siempre me han apoyado y me han brindado su amor y sabios consejos: mi familia.

En especial a mi madre y a mi insuperable hermana, que siempre han estado conmigo, dándome ánimos y por creer en mí. Este logro también es suyo.

Gracias Dios mío porque has permitido que se culmine este proyecto al que le he invertido tanto esfuerzo y dedicación.

También, quiero agradecerte a ti Roberto, por haberme acompañado a lo largo de mis estudios, por haberme escuchado y por estar siempre a mi lado.

AGRADECIMIENTOS

Primeramente quiero darle gracias a Dios que nunca me ha dejado de su mano, y ha caminado conmigo durante toda mi vida. Gracias por ayudarme a conquistar mis sueños y darme fuerzas para conseguir mis metas.

A mis profesores, mis compañeros y a todas las personas que me facilitaron el poder concluir mis estudios de maestría y por hacer mi estancia en el PISIS placentera.

También quiero externar mi agradecimiento a mi asesor, que más que un profesor ha sido un amigo, que ha tenido la paciencia para guiar esta investigación. Gracias porque siempre tuvo palabras de aliento cuando yo tenía vacilaciones.

Gracias a mis sinodales, dos maravillosas personas y profesores, por ayudarme a concluir esta tesis, por tomarse el tiempo para revisarla a pesar de todo el trabajo que tenían y por sus valiosos comentarios.

Quedo muy agradecida con mi alma mater, en la cual además de la maestría estudié la licenciatura. Pondré el nombre de FIME y de la UANL siempre en alto.

Agradezco la oportunidad al PISIS y CONACyT por haberme permitido realizar mis estudios de maestría, por la beca del proyecto J33187-A y por permitirme desempeñarme como Asistente de Investigación.

RESUMEN

Diana Cobos Zaleta

Candidato para el Grado de Maestro en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Título del Estudio:

MODELOS DE OPTIMIZACIÓN ENTERA MIXTA NO LINEAL EN SISTEMAS DE TRANSPORTE DE GAS NATURAL

Número de páginas: 79

OBJETIVO Y MÉTODO DE ESTUDIO

En esta tesis abordamos un problema de transporte de gas natural en una red de gasoductos, considerando simultáneamente variables de decisión enteras y continuas, con el objetivo de decidir la mejor forma de operar nuestra red, minimizando los costos en que se incurren para satisfacer las demandas del combustible. Al presentar además características de no linealidad, sabemos que nos enfrentamos a un problema no lineal entero mixto (MINLP por sus siglas en inglés que significan Mixed-Integer Non Linear Programming).

Al fluir el combustible a través de la red, la energía con la cual éste se mueve, se va perdiendo debido a la transferencia de calor entre el medio ambiente y el gas transportado, y debido a la fricción entre el fluido (gas natural) y las paredes internas de las tuberías por las cuáles es conducido. Para contrarrestar estas pérdidas de energía y mantener el gas en movimiento, a lo largo de la red, se instalan estaciones compresoras, las cuales tienen un número predeterminado de compresores centrífugos idénticos conectados en paralelo, y sobre los cuales debe decidirse cuáles utilizar. Además de las

pérdidas de presión antes mencionadas, se sabe que las estaciones compresoras son alimentadas del mismo combustible que transportan, lo cuál se ha estimado sea de un 3% a un 5% del gas natural transportado [44]. Esto cobra importancia debido a las grandes cantidades de gas natural que son transportadas diariamente (miles de millones de pies cúbicos) y por lo mismo es significativo encontrar la forma óptima de operar esta red.

La formulación de este problema es, en parte, como la de un problema de optimización de flujo en redes; sin embargo, al ser un problema MINLP, encontramos ciertos aspectos que lo hacen difícil de tratar. Por la naturaleza de los problemas MINLP, sabemos que al incrementarse el número de variables enteras (o binarias) nos enfrentamos a un problema combinatorio grande, y de acuerdo al análisis de complejidad encontramos que son problemas NP-completos [11]. A la vez, debido a la no linealidad de los problemas MINLP, generalmente son no convexos, lo cual implica la existencia potencial de varias soluciones óptimas locales. La determinación de una solución global para un problema MINLP no convexo es también NP-completo [26].

En nuestro caso particular, aunque las variables discretas no son demasiadas, nuestro problema es no convexo, tanto en la función objetivo, como en el conjunto de soluciones factibles definido por las restricciones tecnológicas, por lo que nos enfrentamos a un problema aún más difícil. Lo que se pretende con la metodología propuesta es encontrar soluciones óptimas locales y soluciones factibles a las topologías de red propuestas en nuestra base de datos, manejando simultáneamente de variables de decisión enteras (número de unidades compresoras encendidas en cada estación compresora) y continuas (flujo conducido en los arcos y las presiones del gas en cada nodo).

Para resolver nuestro problema empleamos una técnica de solución basada en un algoritmo de aproximación externa con relajamiento de igualdades y penalidad aumentada OA/ER/AP, propuesto por Grossmann [11]. Además empleamos tres técnicas de pre-procesamiento para ayudar a la convergencia del algoritmo.

Los objetivos centrales de esta tesis son:

1. Estudiar y comprender la estructura matemática del problema que nos permita emplear adecuadamente las técnicas de solución.
2. Evaluar computacionalmente el algoritmo y encontrar los valores de los parámetros algorítmicos que provean soluciones de mayor calidad y/o rapidez.

3. Mostrar que la técnica puede encontrar soluciones óptimas (locales) a una gran diversidad de instancias del problema.
4. Mostrar que las soluciones obtenidas al resolver el problema como MINLP son de mejor calidad que las obtenidas con otras variaciones del problema, por ejemplo, cuando se resuelve como programa no lineal (NLP) presuponiendo un número fijo de compresores en operación.

Para lograr estos objetivos se siguió la siguiente metodología. En primera instancia se llevó a cabo un estudio de la estructura matemática del problema. Esto permitió elaborar el modelo matemático como MINLP. Posteriormente, se empleó GAMS [5] para modelar el problema algebraicamente y la implementación DICOPT [15] del algoritmo OA/ER/AP antes mencionado. Las pruebas computacionales y de experimentación se realizaron en una estación de trabajo Sun ultra 10.

Cabe señalar que también se elaboró una amplia base de datos, con los tres diferentes tipos de topologías de redes. En total se construyeron tres instancias para el tipo sencillo, tres para el tipo de árbol y cinco para las redes con ciclos, con la finalidad de probar el método OA/ER/AP y evaluar su comportamiento.

Contribuciones y Conclusiones

Las contribuciones científicas de esta tesis fueron:

- Resolver este problema considerando simultáneamente variables discretas y continuas y que con sus demás características lo convierten en un MINLP, lo cual no había sido tratado anteriormente, hasta donde se tiene conocimiento.
- Evaluar el método OA/ER/AP y determinar bajo que condiciones en sus parámetros obtenemos los mejores resultados.
- Construir una amplia base de datos para evaluar el desempeño del método propuesto, con los tres tipos de topologías de redes definidas.
- Mostrar la validez de un modelo MINLP sobre uno NLP.

Desde el marco científico, se mostró que es posible resolver este problema como un MINLP y bajo que condiciones sus parámetros ayudan a la convergencia de soluciones de mejor calidad. Se compara la forma de resolver este problema como MINLP, NLP y como NLP con el número de compresores fijos en cierto valor, con lo que pretendemos subrayar la importancia del trabajo de investigación desarrollado. Estos resultados están avalados estadísticamente en un amplio y diverso número de instancias del problema de diversas topologías de red y usando datos de compresores reales tomados de la industria.

Profesor asesor:



Dr. Roger Z. Ríos Mercado

TABLA DE CONTENIDO

DEDICATORIA	iv
AGRADECIMIENTOS	v
RESUMEN	vi
CAPÍTULO 1. INTRODUCCIÓN	1
1.1 Descripción y Relevancia del Problema	1
1.2 Antecedentes Científicos	3
1.3 Objetivo	6
1.4 Descripción de la Tesis	7
CAPÍTULO 2. DESCRIPCIÓN DEL PROBLEMA	8
2.1 Definición	8
2.2 Modelo Matemático	10
2.3 Características del Compresor	12
CAPÍTULO 3. MÉTODO DE SOLUCIÓN PARA PROBLEMAS ENTEROS MIXTOS NO LINEALES	17
3.1 Modelos Enteros Mixtos No Lineales	17
3.2 Métodos de Solución para Resolver un MINLP	18
3.2.1 Formulación y Descripción Matemática de los Problemas MINLP	18
3.2.2 Retos y Dificultades en los Problemas MINLP	19
3.2.3 Descripción de Algoritmos para Resolver Problemas MINLP	20
3.3 Descripción del OA/ER/AP	21
3.3.1 Orígenes	22
3.3.2 Formulación de los los Problemas Resueltos por OA/ER/AP	22
3.3.3 Desarrollo del Algoritmo	24
3.3.4 Implementación del OA/ER/AP en GAMS: DICOPT	26
CAPÍTULO 4. EVALUACIÓN COMPUTACIONAL	28
4.1 Descripción de Base de Datos	28

	xi
4.2 Experimento A: Evaluación Algorítmica	31
4.2.1 Diseño Experimental	32
4.2.2 Resultados	35
4.2.3 Conclusiones	47
4.3 Experimento B: Comparación de Problemas Enteros Mixtos No Lineales y Problemas No Lineales con Variables Fijas	49
4.3.1 Diseño Experimental	50
4.3.2 Resultados	51
4.3.3 Conclusiones	53
CAPÍTULO 5. CONCLUSIONES	54
5.1 Conclusiones Finales	54
5.2 Aportaciones Científicas	56
5.3 Recomendaciones para Trabajos Posteriores	57
BIBLIOGRAFÍA	59
LISTADO DE TABLAS	64
LISTADO DE FIGURAS	65
APÉNDICE A: GAMS/DICOPT	66

CAPÍTULO 1

INTRODUCCIÓN

1.1 Descripción y Relevancia del Problema

En la industria del gas natural se requiere transportar el combustible desde sus yacimientos hasta los centros de distribución y demanda, que en muchas ocasiones, se encuentran muy alejados entre sí. Para satisfacer las demandas de los clientes, el fluido es transportado a través de una red de gasoductos, la cual se asume conocida, es decir, ya están diseñada y construida. En algunos casos algunos tramos se siguen añadiendo a sistemas existentes.

Al fluir el gas por las tuberías, hay una pérdida de energía originada por la fricción entre el gas y las paredes internas de los tubos y por la transferencia de calor entre el combustible y el medio ambiente. Para sobreponer esta disminución de presiones, a lo largo de toda la red se han instalado estaciones compresoras que consumen parte del combustible conducido, que se estima es alrededor del 3 al 5 % [44], lo cual es tremendamente significativo si observamos los miles de millones de pies cúbicos transportados diariamente.

Por otro lado, sabemos que con la optimización rigurosa de las operaciones se podría ahorrar al menos un 20% del combustible consumido en la transportación del gas natural [37]. Se sabe además que los costos de operación de las unidades compresoras encendidas representan del 25% al 50% del presupuesto de operación total de la compañía [23]. Esto hace al problema de minimización de consumo de combustible, en una red de gasoductos significativamente relevante.

La toma de decisiones para este problema consiste en encontrar la forma de operar esta red y sus compresores de modo tal que sean cubiertas las demandas de combustible, al mínimo costo posible. Es decir, con la solución a este problema se espera saber, dada una demanda y sabiendo donde se localizan las fuentes de combustible en la red, cuánto flujo pasar por cada arco, qué presiones necesitamos tener en cada nodo de la red (variables

continuas) y cuántos compresores deben estar encendidos en cada estación compresora (variable discreta).

Este problema ha sido tratado desde diferentes perspectivas. Por un lado, se han llevado a cabo trabajos de simulación determinística descriptiva basadas en modelos en estado estable y otras cuantas en estado transiente (dependientes del tiempo) con diferente grado de éxito. Por otra parte, desde la perspectiva de optimización, numerosos investigadores han tratado de resolver problemas similares al que aquí se presenta, haciendo diferentes consideraciones en los modelos que abordan (las características específicas del problema) y empleando diferentes técnicas de optimización (la más adecuada según las suposiciones hechas del modelo). Por ejemplo este problema se ha resuelto (ignorando las variables discretas) empleando técnicas de optimización no lineal (NLP por sus siglas en inglés) y de programación dinámica (DP por sus siglas en inglés) principalmente.

Como se ha trabajado anteriormente este problema era fijando el número de compresores y se resolvía encontrando el valor de los flujos y las presiones como un NLP, o se fijaba el flujo que pasaría por los arcos primeramente y posteriormente se determinaba el número óptimo de compresores que debían estar encendidos. Sin embargo, las metodologías empleadas, consideran en diferentes niveles las variables, por lo tanto, los resultados obtenidos son sub-óptimos del problema estudiado.

La contribución de mi trabajo es el considerar las variables de decisión continuas y discretas simultáneamente. Para tratar ambas variables de decisión al mismo nivel, nuestro modelo está elaborado como un problema de programación no lineal entero mixto conocido también como MINLP: Mixed-Integer Non Linear Problem, por sus siglas en inglés.

Para que el problema sea resuelto exitosamente debe enfrentar algunas dificultades, como por ejemplo, sabemos que por la naturaleza de los problemas MINLP, si se incrementa el número de variables enteras (o binarias) nos enfrentamos a un problema combinatorio grande, y de acuerdo al análisis de complejidad encontramos que son problemas NP-completos [27]. A la vez, debido a la no linealidad que se presenta, generalmente éstos problemas son no convexos, lo cual implica la existencia potencial de varias soluciones óptimas locales. La determinación de una solución global para un problema MINLP no convexo es también NP-completo [26].

En este caso en particular, las unidades compresoras son sofisticadas y complejas. En cada estación compresora hay un número determinado de compresores centrífugos con distintas configuraciones y características. El comportamiento de cada estación compresora es no lineal y por ello, las condiciones de operación factible de un compresor definen un dominio factible no lineal, al igual que las restricciones de la dinámica de flujo, y entre las dos constituyen un sistema complejo de restricciones no lineales (al igual que su dominio). Además, la posibilidad de que exista más de un óptimo local podría propiciar que el procedimiento de búsqueda de la solución, quede “atrapado” en un óptimo local. Todas estas características hacen que el problema sea difícil de resolver y dificulten el éxito de los algoritmos existentes.

En esta tesis se pretende evaluar computacionalmente la eficiencia de un algoritmo de aproximación externa con relajamiento de igualdades y penalidad aumentada (OA/ER/AP por sus siglas en inglés: Outer Approximation with Equality Relaxation and Augmented Penalty) en la solución de este problema al igual que los parámetros y técnicas que nos conduzcan a soluciones de calidad, tanto en valor de la función objetivo como el tiempo de cómputo empleado. Al modelar el problema como un MINLP, todas las variables de decisión se manejan al mismo nivel, lo cual implica que los resultados obtenidos serán soluciones factibles y óptimos locales. Hasta donde se tiene conocimiento, esta modelación no se había realizado. Un segundo objetivo es mostrar cómo con esta técnica se obtienen mejores resultados al compararla con algunos otros métodos, como al tratar de resolver este problema como NLP fijando algunas de las variables discretas [40].

1.2 Antecedentes Científicos

Entre los primeros trabajos desarrollados sobre algoritmos de optimización, para la minimización del costo de combustible en estado estable en redes de transmisión, se encuentra el realizado por Wong y Larson [43] en 1968, que mediante técnicas de DP resolvió problemas sencillos con estructuras de redes de línea recta (vea Figura 1.1).

Después, Lall y Percell [22] presentaron un algoritmo de DP que podía resolver problemas con diversas ramas o topología de árbol (vea Figura 1.2), e incorporaba al modelo

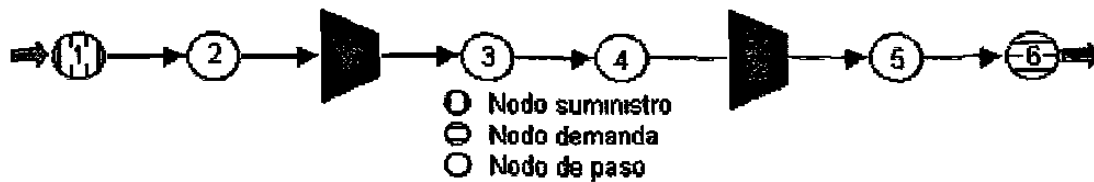


Figura 1.1: Ejemplo de red con topología sencilla ó gunbarrel

la variable de decisión para representar el número de unidades que serían operadas.

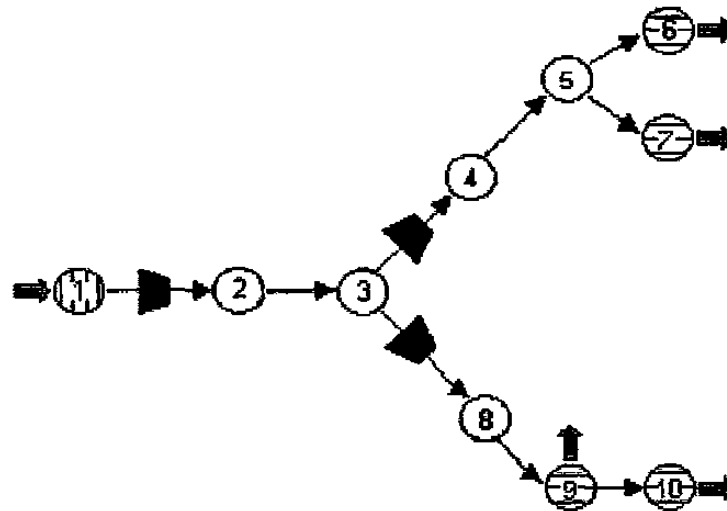


Figura 1.2: Ejemplo de red con topología con ramas ó de árbol

Más recientemente, Carter [6] desarrolló un algoritmo de DP no secuencial para manejar redes con ciclos (vea Figura 1.3), cuando las variables de tasa de flujo estaban fijas.

Las principales ventajas de DP son que está garantizado el encontrar el óptimo global, y el fácil manejo de la no convexidad. Las desventajas de usar DP son que su aplicación está prácticamente limitada a estructuras de redes sencillas, tales como estructuras de línea recta y de árbol, y que el esfuerzo computacional se incrementa exponencialmente según las dimensiones del problema.

Kim, Ríos-Mercado y Boyd [20] extendieron el trabajo de Carter proponiendo un algoritmo que iterativamente ajustara la variable flujo de forma heurística, para resolver el problema en topologías con ciclo, basado en un procedimiento iterativo de dos etapas. En

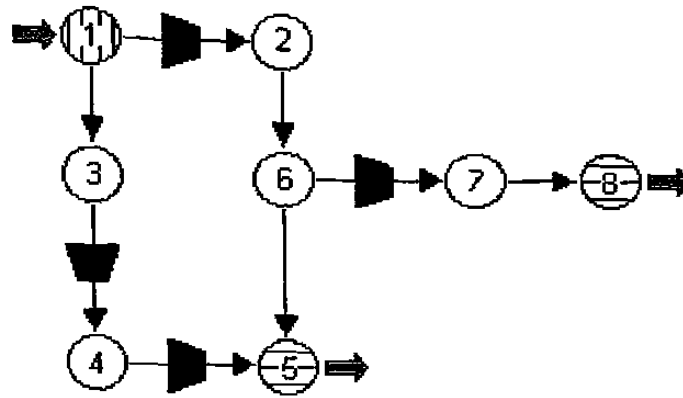


Figura 1.3: Ejemplo de red con topología con ciclos

la primera etapa, las variables de flujo de gas están fijas y las variables de presiones óptimas son encontradas usando DP. En la etapa posterior, las variables de presión son fijadas y se intenta encontrar un conjunto de variables de flujo que mejoren el valor de la función objetivo, explotando la estructura fundamental de la red.

Percell y Ryan [34] atacaron el problema usando el método del gradiente reducido generalizado (GRG por sus siglas en inglés Generalized Reduced Gradient) para la optimización no lineal. Las ventajas para el método GRG es que evita con el problema de dimensionalidad y que puede ser aplicado a redes con ciclos. Sin embargo, como el método GRG está basado en un método de búsqueda por gradiente, teóricamente no está garantizado que se encontrará el óptimo global, especialmente en la presencia de variables discretas de decisión, y puede quedar atrapado en un óptimo local.

Flores-Villarreal y Ríos-Mercado [10, 9] amplían este estudio mediante una extensa evaluación computacional del GRG en una diversa base de datos del problema.

En [45], Wu, Boyd y Scott presentaron un modelo matemático para la minimización del costo de combustible en una estación compresora con una única unidad. En [46], Wu et al. estudian el mismo problema pero ampliado para manejar las estaciones compresoras con múltiples unidades en cada estación compresora.

Por otro lado, muchas simulaciones numéricas de redes de tuberías de gas se han desarrollado desde el siglo pasado y los resultados son ahora más exactos con ayuda de las poderosas computadoras digitales. El libro de Osiadacz [29] es la mejor referencia en este tema.

Además, las técnicas de optimización también han sido aplicadas a modelos transientes (dependientes del tiempo) (ejemplo [30, 32]) y diseño de redes (por ejemplo [31]) con moderado éxito. Vea [36] para más referencias sobre técnicas de optimización aplicadas a problemas de tuberías de gas. Es importante mencionar que la metodología de optimización desarrollada hasta la fecha trabaja bien bajo ciertos supuestos; sin embargo, al volverse el problema más complejo, surgen necesidades para realizar más investigación y desarrollo efectivo de algoritmos desde la perspectiva de optimización.

1.3 Objetivo

Como se ha visto, el problema en estudio es un problema de optimización, cuyo objetivo es minimizar el consumo de combustible en el que incurren los compresores encendidos dentro de las estaciones compresoras en las redes de gas natural. Encontrar soluciones a este problema es de gran interés tanto para la industria del gas así como para el cliente final, pues es quien a final de cuentas, recibe el impacto económico generado en el transporte.

De acuerdo a las características de nuestro problema, sabemos que es un MINLP y para resolverlo se propone utilizar la técnica OA/ER/AP, porque de las técnicas existentes para solucionar estos problemas, es la única que puede operar adecuadamente con la no convexidad de la región factible. El trabajo consiste en evaluar computacionalmente el algoritmo y encontrar los parámetros que ayuden a encontrar soluciones de calidad, tanto en valor de la función objetivo, como en recursos computacionales empleados (tiempo, número de iteraciones, status final de la solución).

Además para ayudar a la convergencia del algoritmo, en la parte no lineal, hemos empleado técnicas de preprocesamiento, para transformar el problema original en uno equivalente que tenga mejores propiedades que le ayuden a ser resuelto fácilmente, desde el punto de vista computacional.

También se enfatiza el hecho de que al tratar este problema como un MINLP, se logran mejores soluciones que las obtenidas al tratarlo como un NLP, o prefijando alguna de las variables discretas para simplificar el problema.

Los objetivos de la tesis se resumen en los siguientes puntos:

1. Evaluar computacionalmente el algoritmo OA/ER/AP y los parámetros que ayuden a

encontrar las soluciones de mayor calidad.

2. Comparar las soluciones obtenidas al resolver el problema como MINLP contra otras variaciones del problema, por ejemplo, cuando se resuelve como NLP presuponiendo un número fijo de compresores de las estaciones compresoras.

1.4 Descripción de la Tesis

Este trabajo está organizado de la siguiente forma. En el Capítulo 2, se presenta la descripción del problema, se plantean las suposiciones efectuadas, así como el modelo matemático y se discuten las características del problema. En el Capítulo 3, se describe en que consisten los problemas MINLP, sus características y algunas de las técnicas de solución más populares. Finalmente se hace la descripción del OA/ER/AP, se exponen sus características y que lo hizo candidato para con este método resolver nuestro problema. En el Capítulo 4, se muestra el trabajo experimental. Se inicia describiendo la base de datos y posteriormente se describen los experimentos realizados (que incluyen: Evaluación algorítmica del método, comparación del modelo MINLP con un modelo NLP con variables fijas y el comportamiento general del método), sus resultados y las conclusiones. Finalmente, en el Capítulo 5, se concluye con los comentarios finales, las aportaciones científicas de nuestro trabajo y las recomendaciones para trabajos posteriores.

CAPÍTULO 2

DESCRIPCIÓN DEL PROBLEMA

2.1 Definición

En esta tesis abordamos el estudio de un problema de minimización de combustible en redes de gas natural, viéndolo como un problema en estado estable (independiente del tiempo), y que por sus características de no linealidad y el manejo de variables continuas y discretas al mismo nivel, nos referimos a él como un problema MINLP.

El objetivo de este problema es minimizar los costos que conlleva la transportación del combustible desde sus yacimientos hacia los puntos de demanda, que en muchas ocasiones se encuentran muy alejados entre sí, para lograr satisfacer las demandas de los clientes, a través de redes de tuberías.

Sabemos que esta transportación del combustible produce pérdidas de energía, originadas por la transferencia de calor entre el medio ambiente y el combustible, y por la fricción que surge entre las paredes internas de la tubería y el gas. Para reestablecer estas disminuciones de energía, se han instalado a lo largo de la red estaciones compresoras, cuyo objetivo es mantener al fluido en movimiento y las presiones en cierto nivel.

La industria del gas natural es una empresa próspera que ha ido desarrollándose y creciendo constantemente. Las redes que las constituyen hoy en día son muy complejas, y pueden constar de cientos de ductos, docenas de estaciones compresoras, y muchos otros accesorios como válvulas y reguladores. Además, dentro de cada estación compresora pueden existir varias unidades de compresores. Se considera que tales estaciones compresoras están siendo alimentadas con parte del combustible que está siendo transportado. De trabajos anteriores [44], sabemos que ese porcentaje varía entre un 3% a un 5%.

El problema de toma de decisión consiste en determinar la manera óptima de operar cada red de gasoductos, es decir, debemos determinar el flujo que pasará a través de cada

ducto y las presiones que deberán existir en cada nodo de nuestra red (variables continuas de decisión) así como cuántos y cuáles compresores deberán estar en operación (variable discreta de decisión), para transportar el combustible al menor costo y satisfacer las demandas de los clientes.

Para modelar el problema se efectúan las siguientes suposiciones:

1. El problema está en estado estable. Esto quiere decir que nuestro modelo es apto para sistemas que han estado operando por un período de tiempo relativamente largo. Un análisis transiente requeriría incrementar el número de variables y la complejidad de este problema significativamente.
2. La red está balanceada. Esto significa que la suma de todos los flujos en cada nodo de la red es igual a cero. En otras palabras, el flujo total suministrado cubre completamente toda la demanda, sin pérdidas. Sabemos que las estaciones compresoras están siendo alimentadas por el combustible que es transportado en la red de gasoductos, y para apoyar la suposición formulada, consideramos este costo de combustible como un costo extra llamado costo de oportunidad, que representa el costo que se generaría si alimentáramos a las estaciones compresoras con combustible comprado a otro proveedor.
3. La red es una red dirigida, lo cual quiere decir que cada arco en la red tiene una dirección pre-especificada.
4. La temperatura a la cual se transporta el combustible permanece constante a través del ducto. Es decir, se asume un proceso isotérmico. Esto se hace con la finalidad de hacer más tratable la restricción de flujo en los ductos.
5. Cada estación compresora está constituida por un número fijo de unidades compresoras idénticas, las cuáles se aumen del tipo centrífugos idénticos y están conectadas en paralelo (ver Figura 2.4). Esta suposición nos conduce a dividir el flujo que pasa por la estación compresora en " n " partes iguales, donde n representa el número de compresores encendidos. Cabe mencionar que la configuración en paralelo es de las más comunes en la industria. El tipo centrífugo de compresores es también de los más utilizados a nivel industrial.

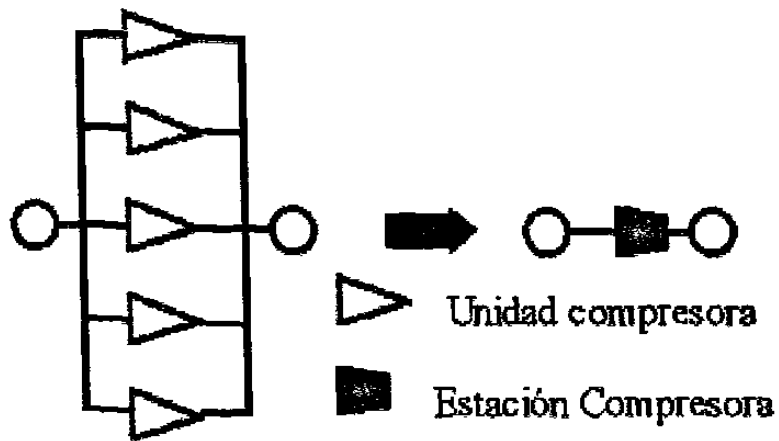


Figura 2.4: Estación Compresora configurada en paralelo

2.2 Modelo Matemático

Para facilitar la comprensión del modelo matemático, en este trabajo se denotarán por letras minúsculas a las variables y en letras mayúsculas se denotarán los datos y parámetros del problema.

Parámetros:

V_s Conjunto de nodos suministro

V_d Conjunto de nodos demanda

V Conjunto de todos los nodos en la red; $V = V_s \cup V_d$

A_p Conjunto de arcos ducto

A_c Conjunto de arcos de las estaciones compresoras

A Conjunto de todos los arcos en la red; $A = A_p \cup A_c$

U_{ij} Capacidad del arco ducto (i, j) ; $(i, j) \in A_p$

R_{ij} Resistencia del ducto (i, j) ; $(i, j) \in A_p$

N_{ij} Número de unidades compresoras disponibles en la estación (i, j) ; $(i, j) \in A_c$

Parámetros:

P_i^L, P_j^U Límites de presión superior e inferior, respectivamente en el nodo i ; $i \in V$

B_i Tasa de flujo neta en el nodo i ; $B_i > 0$ si $i \in V_s$, $B_i < 0$ si $i \in V_d$,

$B_i = 0$ de otro modo.

Variables:

x_{ij} Tasa de flujo másico en el arco (i, j) ; $(i, j) \in A$

p_i Presión en el nodo i ; $i \in V$

n_{ij} Número de unidades compresoras trabajando en la estación (i, j) ; $(i, j) \in A_c$

Formulación:

$$\min \sum_{(i,j) \in A_c} n_{ij} g_{ij}^1 \left(\frac{x_{ij}}{n_{ij}}, p_i, p_j \right) \quad (2.1)$$

sujeto a:

$$\sum_{\{j|(i,j) \in A\}} x_{ij} - \sum_{\{j|(j,i) \in A\}} x_{ji} = B_i \quad i \in V \quad (2.2)$$

$$x_{ij} \leq U_{ij} \quad (i, j) \in A \quad (2.3)$$

$$p_i^2 - p_j^2 = R_{ij} x_{ij}^2 \quad (i, j) \in A_p \quad (2.4)$$

$$P_i^L \leq p_i \leq P_i^U \quad i \in V \quad (2.5)$$

$$\left(\frac{x_{ij}}{n_{ij}}, p_i, p_j \right) \in D_{ij} \quad (i, j) \in A_c \quad (2.6)$$

$$x_{ij}, p_i \geq 0, n_{ij} \in \{1, 2, \dots, N_{ij}\} \quad (2.7)$$

La expresión (2.1) representa la función objetivo, donde $g_{i,j}^1$ mide el valor del combustible consumido en una unidad compresora individual. En otras palabras, deseamos que el consumo de combustible con que se alimentan las estaciones compresoras sea el mínimo, y como se puede apreciar, depende de las variables de decisión: flujos en cada arco de la red, presiones nodales (variables continuas de decisión) y del número de unidades centrífugas que se decida encender (variable discreta de decisión).

Las siguientes ecuaciones son las restricciones que formarán el dominio factible. La restricción (2.2) representa el balance de flujo en cada nodo de la red, donde un valor positivo de B_i indica que es un nodo fuente, un B_i negativo, indica que es un nodo demanda y si $B_i = 0$ nos dice que se trata de un nodo de paso. Además se tiene que $\sum_{i \in V} B_i = 0$, lo cual indica que la red está balanceada. La restricción (2.3) se refiere a que el flujo que pase por cada ducto de la red, no puede exceder la capacidad de la tubería. Estas dos restricciones son las características de cualquier problema típico de flujo en redes. La dificultad de este problema se presenta en las siguientes restricciones.

La restricción (2.4) se refiere a la dinámica del flujo del gas en cada ducto, y nos muestra las relaciones que hay entre la disminución de presiones y el flujo y se conoce como la

ecuación de Osiadacz [29], válida para gases de alta presión, como es el caso. Para un tratado más amplio, veáse la referencia [19]. La restricción (2.5) indican los límites permisibles de la presión en cada nodo.

La restricción (2.6) nos indica el dominio factible D_{ij} de operación para cada estación compresora. Aquí vemos que el flujo que pasa por cada arco compresor x_{ij} puede ser dividido igualmente entre el número n_{ij} de compresores centrífugos funcionando en esa estación compresora. Una descripción más detallada se puede encontrar en [44]. La restricción (2.7) se refiere a la no negatividad de las variables de flujo en cada arco y presiones en cada nodo, y al carácter entero de la variable n_{ij} , que puede tomar valores desde uno hasta su cota superior (N_{ij}).

2.3 Características del Compresor

La parte no lineal de este problema esta caracterizada en el objetivo (2.1) y en las restricciones (2.6) y (2.4). Sobre (2.4) podemos encontrar más información en la referencia [29]. En esta sección describimos las características de una estación compresora, lo cual da pauta a (2.1) y (2.6).

Comenzaremos por explicar el dominio D_{ij}^1 de operación de una unidad compresora. Se introducen las siguientes variables auxiliares:

q_{ij} Tasa de flujo volumétrico de entrada al compresor $(i, j); (i, j) \in A_c$

h_{ij} Cabeza adiabática del compresor $(i, j); (i, j) \in A_c$

s_{ij} Velocidad del compresor $(i, j); (i, j) \in A_c$

Se sabe de la referencia [46] que el dominio D_{ij}^1 de un compresor centrífugo está definido por las siguientes restricciones:

$$\frac{h_{ij}}{s_{ij}} = A_H + B_H \left(\frac{q_{ij}}{s_{ij}} \right) + C_H \left(\frac{q_{ij}}{s_{ij}} \right)^2 + D_H \left(\frac{q_{ij}}{s_{ij}} \right)^3$$

$$S_{ij}^L \leq s_{ij} \leq S_{ij}^U$$

$$R_{ij}^L \leq \frac{q_{ij}}{s_{ij}} \leq R_{ij}^U$$

Donde A_H, B_H, C_H, D_H son constantes que dependen del tipo de compresor y son típicamente estimados con el método de mínimos cuadrados; S_{ij}^L y S_{ij}^U son las velocidades conocidas mínimas y máximas del compresor respectivamente; y R_{ij}^L y R_{ij}^U son los límites inferior y superior de $\frac{q_{ij}^L}{s_{ij}^L}$ (surge) y $\frac{q_{ij}^U}{s_{ij}^U}$ (stonewall) respectivamente.

Ahora bien, existe un mapeo que relaciona el espacio (q_{ij}, h_{ij}, s_{ij}) con el espacio (x_{ij}, p_i, p_j) , con x_{ij} como el flujo másico, p_i como la presión de entrada (presión de succión) y p_j como la presión de salida (presión de descarga) en cada arco de la red. Este mapeo esta dado por las ecuaciones:

$$h_{ij} = \frac{Z R T_s}{m} \left[\left(\frac{p_i}{p_j} \right)^m - 1 \right]$$

$$q_{ij} = Z R T_s \frac{x_{ij}}{p_j}$$

donde los siguientes parámetros se asumen conocidos:

- T_s Temperatura del gas
- Z Factor de compresibilidad del gas
- R Constante del gas
- m Coeficiente específico de calor, $m = \frac{k-1}{k}$

Las variables q_{ij}, h_{ij} y s_{ij} las conoce el operador directamente (puede tomar mediciones); sin embargo, dado el mapeo de (h_{ij}, q_{ij}, s_{ij}) hacia (x_{ij}, p_i, p_j) , es preferible trabajar con el segundo espacio desde la perspectiva de la optimización de redes, ya que el flujo másico se conserva en cada nodo.

En la Figura 2.5 ilustra este dominio en el espacio (x_{ij}, p_i, p_j) para x_{ij} fijo. En la Figura 2.6 presentamos este mismo dominio, pero con tres diferentes niveles de flujo representados cada uno con un diferente símbolo.

Ahora bien, la función objetivo (2.1) mide la cantidad de combustible consumido en cada estación cuando n_{ij} unidades están operando. El consumo en una estación compresora está dado por:

$$g_{ij}^1(x_{ij}, p_i, p_j) = \left(\frac{\alpha_{ij} x_{ij}}{\eta(x_{ij}, p_i, p_j)} \right) \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right] \quad (x_{ij}, p_i, p_j) \in D_{ij} \quad (2.8)$$

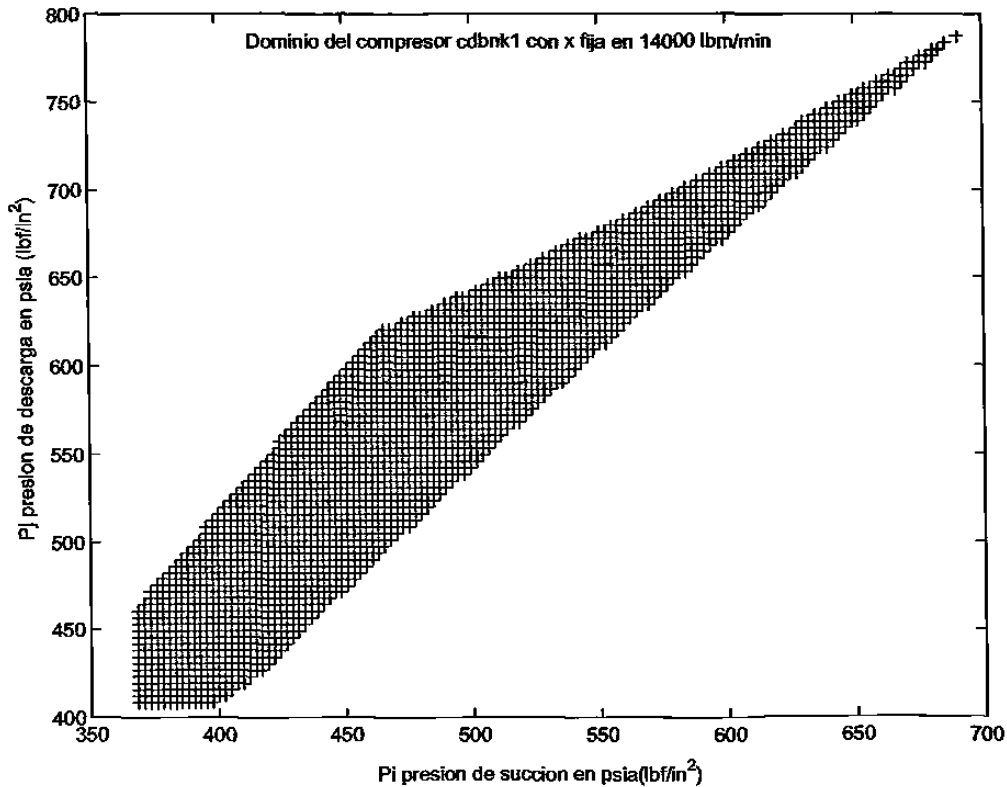


Figura 2.5: Gráfica del dominio de un compresor con la variable de flujo fija.

Donde α_{ij} es un parámetro cuyo valor es aproximadamente 1 y $\eta(x_{ij}, p_i, p_j)$ mide la eficiencia del compresor. D_{ij}^1 describe el dominio factible de operación del compresor y está definido para cada tipo de unidad centrífuga.

Ahora bien, en el caso de n_{ij} unidades en operación, como se asumen idénticas, es razonable suponer que el flujo del compresor se divide equitativamente entre cada unidad. Por tanto, el consumo total en la estación se da por:

$$g_{i,j}^n(x_{i,j}, p_i, p_j) = n_{ij} g_{ij}^1 \left(\frac{x_{ij}}{n_{ij}}, p_i, p_j \right) \quad \left(\frac{x_{ij}}{n_{ij}}, p_i, p_j \right) \in D_{ij}^1 \quad (2.9)$$

Como se aprecia, el flujo que pasa a través de cada unidad centrífuga debe satisfacer D_{ij}^1 , que es el mismo dominio de (2.8), donde sólo hay un compresor centrífugo funcionando. Para evaluar el costo de combustible de la estación compresora (representado por g_{ij}^n) se calcula el costo en cada unidad compresora g_{ij}^1 y se multiplica por los compresores centrífugos n_{ij} encendidos. Como se puede apreciar, (2.8) y (2.9) son no lineales. Además

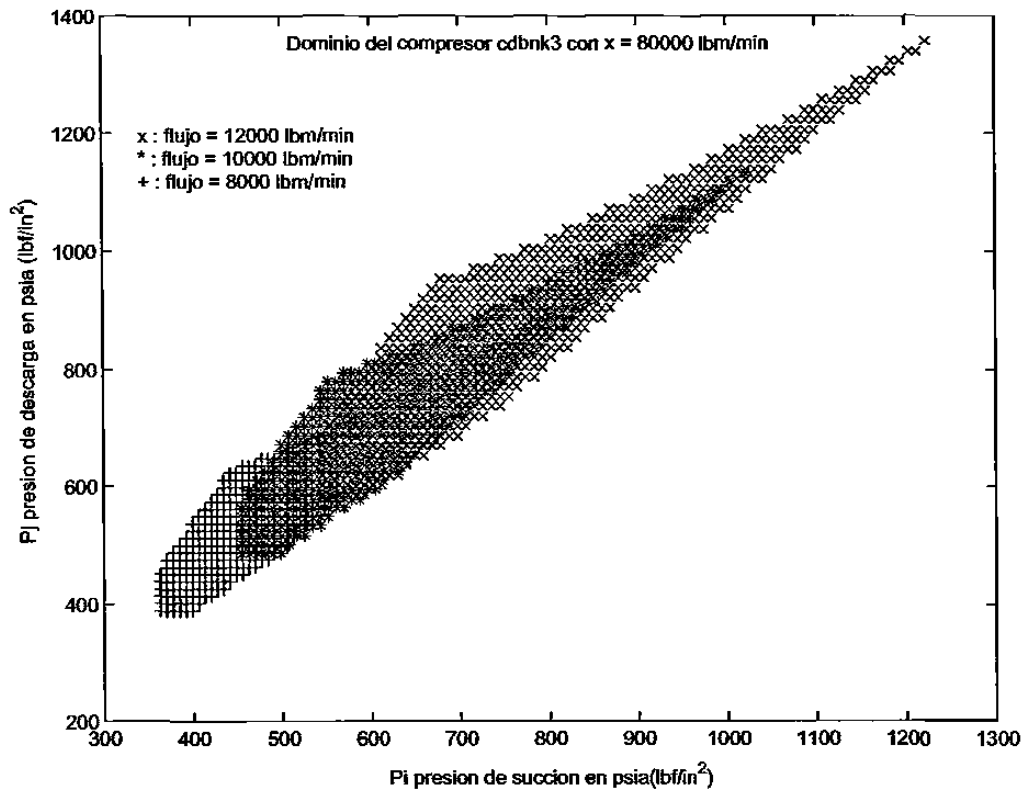


Figura 2.6: Gráfica del dominio de un compresor con variable flujo fija en tres niveles.

son no convexas, como se puede ver en la Figura 2.7. La parte entera se encuentra representada por el número de compresores trabajando en cada estación compresora (variable de decisión entera), que está representada por la variable n_{ij} presente en la ecuación dominio (2.6) y (2.9).

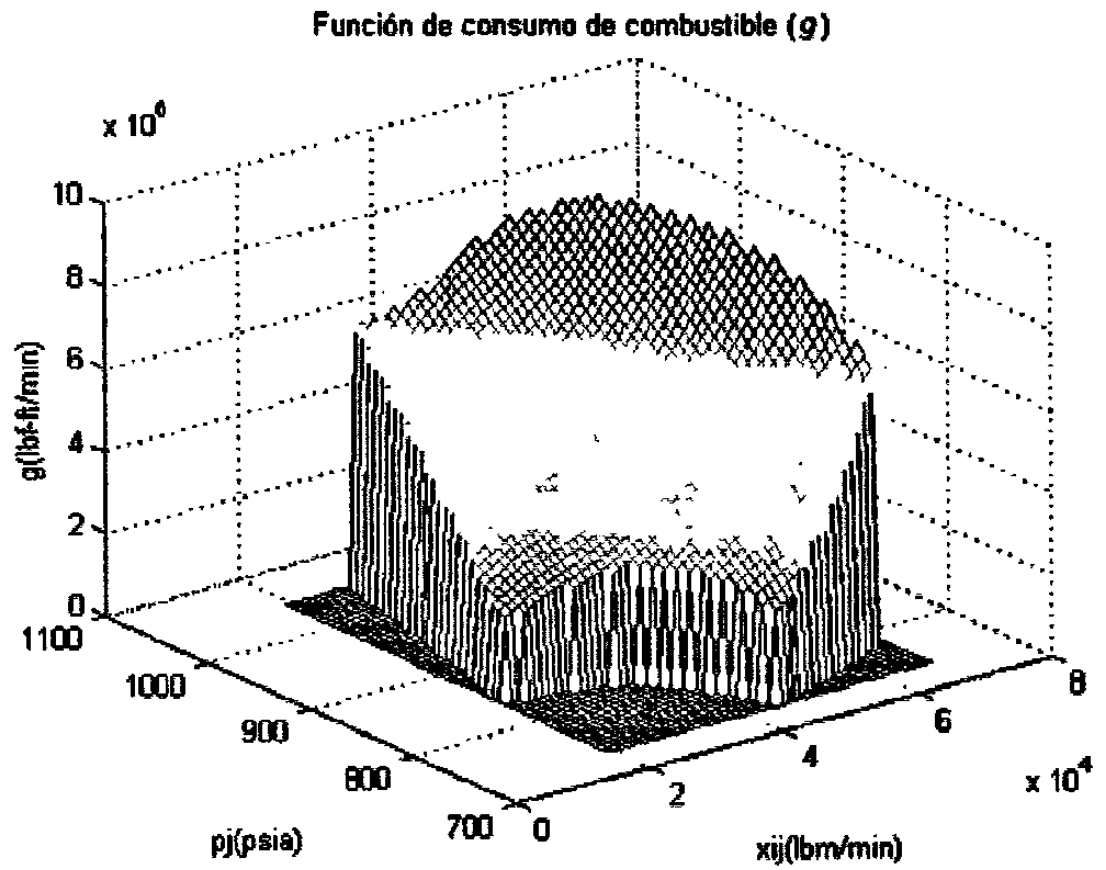


Figura 2.7: Gráfica del dominio de un compresor con la variable flujo fija.

CAPÍTULO 3

MÉTODO DE SOLUCIÓN PARA PROBLEMAS ENTEROS MIXTOS NO LINEALES

3.1 Modelos Enteros Mixtos No Lineales

Una gran parte de los problemas de optimización no lineal involucran variables enteras, al igual que variables continuas de decisión. Cuando el objetivo y las restricciones son todas lineales, se obtiene un problema de optimización entera mixta (lineal) o MIP (por sus siglas en inglés). Cuando existen restricciones u objetivo no lineal, el problema se denota como un problema de programación no lineal entera mixta o MINLP.

Las variables enteras pueden ser usadas para modelar, por ejemplo, secuencia de eventos, candidatos alternativos, existencia o no de unidades (representados como variables binarias 0-1), mientras que las variables discretas pueden modelar, por ejemplo, tamaños de equipo. Las variables continuas son usadas para representar la entrada-salida y relaciones de iteración entre unidades u operaciones individuales y diferentes sistemas interconectados.

En nuestro caso tenemos la variable discreta n_{ij} que representa el número de unidades compresoras funcionando dentro de cada estación compresora y las variables continuas de flujo a través de cada arco x_{ij} , y presiones p_i en cada nodo de nuestra red $p_i \ i \in V$.

La naturaleza no lineal de este tipo de problemas puede provenir de:

1. Relaciones no lineales en el dominio discreto únicamente,
2. Relaciones no lineales en el dominio continuo solamente, ó
3. Relaciones no lineales en la fusión del dominio entero-continuo.

En nuestro caso se nos presenta la tercera característica. La unión del dominio discreto y continuo asociado con características de no linealidad hacen a la clase de problemas MINLP desafiantes desde el punto de vista teórico, algorítmico y computacional. Además del reto que representan, hay una gran gama de aplicaciones que pueden ser modeladas como problemas MINLP. Algunas de sus aplicaciones están en el campo de la industria química, como el proceso de síntesis; en el área de diseño [2], programación de actividades en procesos [38] y planeación de procesos por lotes [35], entre los cuales podemos mencionar, el diseño de plantas multiproducto y el diseño y programación de plantas multiproducto [28]. Otro rubro donde se ha aplicado recientemente, es en el de diseño de aspectos moleculares para seleccionar los mejores solventes [1].

Además de la industria química, se encuentran aplicaciones en otros campos de ingeniería, tales como: la localización de edificios con espacios multiatributo [12], la ubicación óptima de unidades de edificios en un sistema de energía eléctrica [3], la planeación de construcciones de un generación de energía eléctrica [16] y la optimización de topologías en redes de transporte, por mencionar algunos.

A continuación explicaremos brevemente algunos métodos de solución y describiremos como funciona el algoritmo OA/ER/AP, que es el método de solución que empleamos en esta tesis.

3.2 Métodos de Solución para Resolver un MINLP

En esta sección se presenta la formulación de los problemas MINLP y se señalan en qué consisten las dificultades y los retos que enfrentamos al resolver estos problemas y finalmente, describimos brevemente en qué consisten algunas de las aproximaciones desarrolladas para resolver estos problemas.

3.2.1 Formulación y Descripción Matemática de los Problemas MINLP

La formulación general de un problema MINLP se enuncia como sigue:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{sujeto a:} \quad & \\ & h(x,y) = 0 \end{aligned} \tag{3.10}$$

$$\begin{aligned}
 g(x,y) &\leq \theta \\
 x &\in X \subseteq \mathfrak{R}^n \\
 y &\in Y \text{ entero}
 \end{aligned}$$

donde x representa un vector de n variables continuas, y es un vector de variables enteras; $h(x,y) = \theta$ representa m restricciones de igualdad (por ejemplo, balance de masa, balance de energía, relaciones de equilibrio); $g(x,y) \leq \theta$ son p restricciones de desigualdad (como especificaciones sobre la pureza de productos destilados, regulaciones ambientales, restricciones de factibilidad en sistemas de recuperación de calor, restricciones lógicas); $f(x,y)$ es la función objetivo (que puede ser el costo total anualizado, ganancias o criterio termodinámico, etc.).

Las variables enteras y con la cota inferior y la cota superior dadas, ($y^L \leq y \leq y^U$), pueden representarse en función de variables 0 – 1 (por ejemplo, binarias), y la formulación (3.10) se transforma en la siguiente:

$$\begin{aligned}
 \min_{x,y} \quad & f(x,y) && (3.11) \\
 \text{sujeto a:} & && \\
 & h(x,y) = \theta && \\
 & g(x,y) \leq \theta && \\
 & x \in X \subseteq \mathfrak{R}^n && \\
 & y \in Y = \{0, 1\}^q &&
 \end{aligned}$$

donde y ahora es un vector de q variables 0-1, que pueden representar la existencia de una unidad dentro de un proceso ($y_i = 1$) o la no existencia de la unidad ($y_i = 0$).

3.2.2 Retos y Dificultades en los Problemas MINLP

Tratar los problemas (3.10) ó (3.11) presenta principalmente dos grandes dificultades. Estas dificultades están asociadas con la naturaleza del problema (descritas en la subsección anterior), conocidas como el dominio combinatorio (el dominio de la variable y) y el dominio continuo (dominio x).

La existencia de variables binarias y en (3.11) implica un problema combinatorio grande, y el resultado del análisis de complejidad que caracteriza a estos problemas MINLP es que

son NP-completos [27].

A la vez, debido a las no linealidades de los problemas MINLP, éstos son generalmente no convexos lo cual implica la existencia potencial de varias soluciones locales. La determinación de un óptimo global de un problema MINLP no convexo es también NP-difícil [26] y la optimización global de problemas NLP no convexos es NP-difícil [33].

A pesar de las antes mencionadas dificultades derivadas del análisis de complejidad, las cuales son resultados del peor caso, se ha logrado un avance significativo en el área de los problemas MINLP desde las perspectivas teóricas, algorítmicas y computacionales. En consecuencia, se han propuesto varios algoritmos, de los que se han estudiado sus propiedades de convergencia; actualmente existen múltiples aplicaciones que cruzan los límites de varias disciplinas. A continuación presentaremos algunos de estos logros.

3.2.3 Descripción de Algoritmos para Resolver Problemas MINLP

Una colección representativa de algoritmos desarrollados para resolver los problemas MINLP de la forma (3.11) o clases restringidas, incluyen en orden cronológico de desarrollo, los siguientes:

- | | |
|---|------------|
| 1. Descomposición de Benders Generalizada | (GBD) |
| 2. Ramificación y Acotamiento | (BB) |
| 3. Aproximación Externa | (OA) |
| 4. Proximidad de Factibilidad | (FA) |
| 5. Aproximación Externa con Relajamiento de Igualdades | (OA/ER) |
| 6. Aproximación Externa con Relajamiento de Igualdades
y Penalidad Aumentada | (OA/ER/AP) |

En 1972, el trabajo pionero de Geoffrion [13, 18] sobre la Descomposición de Benders Generalizada (GBD), las secuencias de cotas superior (no incremento) e inferior (no decremento) actualizadas son creadas y convergen dentro de un rango ϵ en un número finito de iteraciones. La cota superior corresponde a la solución de subproblemas en la variable x cuando se fija la variable y , mientras que la cota inferior está basada en la teoría de dualidad.

En 1977, la metodología Ramificación y Acotamiento (BB) [4] empieza por resolver la relajación continua del problema MINLP y desarrolla subsecuentemente una enumeración

implícita donde un subconjunto de variables 0-1 se fijan en cada nodo. La cota inferior corresponde a la solución del problema NLP en cada nodo y se usa para expandir sobre el nodo con menor cota inferior (por ejemplo, enumeración primero en amplitud), o para eliminar nodos si la cota inferior excede a la actual cota superior (por ejemplo, enumeración primero en profundidad). Si la relajación continua NLP del problema MINLP obtiene un resultado 0-1 para las variables y , entonces el algoritmo BB termina en ese nodo. Con un argumento similar, si una relajación justa del problema NLP resulta en el primer nodo del árbol, entonces el número de nodos que necesitaríamos eliminar puede ser pequeño. Sin embargo, relajaciones holgadas al problema NLP pueden traer como consecuencia un gran número de subproblemas NLP que resolver.

La aproximación externa OA [8], resuelve el problema con desigualdades no lineales, y crea secuencias de cotas superiores e inferiores como el GBD pero con la característica distinta de usar información del primal para solucionar los problemas de cota superior, de forma tal que linealiza la función objetivo y restricciones alrededor de ese punto.

La aproximación de factibilidad FA [25], redondea la solución relajada de NLP a una solución entera con la mínima degradación local forzando a las variables superbásicas basadas en información de los costos reducidos.

El algoritmo OA/ER [21], (Outer Approximation with Equality Relaxation), extiende al OA de modo que puede manejar restricciones de igualdad no lineales relajándolas a desigualdades de acuerdo al signo de sus multiplicadores asociados.

El algoritmo OA/ER/AP [41], (Outer Approximation with Equality Relaxation and Augmented Penalty), introduce una función de penalidad aumentada en los subproblemas de la cota inferior de la OA/ER, el cual explicaremos en mayor detalle en la siguiente sección.

3.3 Descripción del OA/ER/AP

El método funciona como una Aproximación Externa (OA) para manejar las desigualdades no lineales, creando una secuencia de cotas superiores e inferiores como en la descomposición de Bender Generalizada, relajando las ecuaciones no lineales de igualdad a desigualdad de acuerdo al signo del multiplicador asociado (ER) e introduciendo una función

de penalidad aumentada en los subproblemas de las cotas inferiores (AP). La selección está basada en que este método tiene la siguiente ventaja: no está limitado por las suposiciones de convexidad que tiene el OA/ER. En otras palabras, maneja la no convexidad a diferencia de los demás algoritmos.

3.3.1 Orígenes

El método de OA/ER/AP fue desarrollado por Viswanathan y Grossmann [41] en el Centro de Investigación de Ingeniería y Diseño en la Universidad de Carnegie Mellon, EUA. El algoritmo resuelve una serie de subproblemas NLP y subproblemas MIP (problemas enteros mixtos). A pesar de que el algoritmo tiene previsto manejar no convexidad, no necesariamente obtiene un óptimo global. En nuestro trabajo, utilizamos la implementación en GAMS [5] del método llamado DICOPT [15].

3.3.2 Formulación de los los Problemas Resueltos por OA/ER/AP

El método AO/ER/AP resuelve problemas del siguiente tipo:

$$\begin{aligned} \min_{x,y} \quad & c^T y + f(x) & (3.12) \\ \text{sujeto a:} & & \\ & h(x) & = \theta \\ & g(x) & \leq \theta \\ & Cx + By & \leq d \\ & x \in X & = \{x : x \in \mathbb{R}^n, A_1 x \leq a_1\} \subseteq \mathbb{R}^n \\ & y \in Y & = \{y : y \in \{0, 1\}^q, A_2 y \leq a_2\} \end{aligned}$$

Bajo las siguientes consideraciones:

Consideración 1: f , h y g son continuamente diferenciables.

Consideración 2: Una condición de restricción se cumple en la solución para cada problema de programación no lineal (3.12) al fijar la variable y .

Nota: Las consideraciones 1 y 2 también se asumen para el algoritmo OA/ER. Además, nótese que como no se imponen suposiciones de convexidad, entonces las equivalencias de las igualdades no lineales y las igualdades relajadas $T^K \mathbf{h}(\mathbf{x})$ ahora pueden no ser válidas.

Idea básica

Como las suposiciones de convexidad no son impuestas al algoritmo OA/ER/AP entonces,

1. Las equivalencias de $\mathbf{h}(\mathbf{x})$ y $T^k \mathbf{h}(\mathbf{x}) \leq \mathbf{0}$ pueden ya no sostenerse.
2. Las linealizaciones puede que no sean soportes válidos.
3. Las soluciones del problema maestro pueden ya no ser cotas inferiores válidas a la solución del problema (3.12).

La idea básica del OA/ER/AP es manejar las limitaciones 1, 2 y 3 mediante relajaciones hechas al problema maestro. Esto se logra permitiéndole violar y utilizar la aproximación del tipo penalidad que castiga estas violaciones de las funciones apoyadas. Las violaciones de las linealizaciones son permitidas al introducir variables de holgura, mientras que la penalidad de las violaciones se introducen como factores de pesos positivos. De esta forma, por la relajación de las restricciones, la región factible se expande, y por tanto la posibilidad de cortar parte de la región factible debido a linealizaciones inválidas se reduce.

Nota: Es importante notar que esta aproximación no garantiza teóricamente no eliminar parte de la región factible, y como resultado, la determinación de la solución óptima global no se puede garantizar.

El Problema Maestro

El problema maestro relajado del OA/ER/AP es el siguiente:

$$\mathbf{z}_L^K = \min \quad \mathbf{c}^T \mathbf{y} + \mu + \sum_k w_k^o s_k^o + \sum_{i,k} w_{i,k}^p p_{i,k} + \sum_{i,k} w_{i,k}^q q_{i,k} \quad (3.13)$$

sujeto a:

$$\mu + s_k^o \geq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) \quad k = 1, 2, \dots, K$$

$$\begin{aligned}
\mathbf{p}_k &\geq \mathbf{g}(\mathbf{x}^k) + \nabla \mathbf{g}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) & k = 1, 2, \dots, K \\
\mathbf{q}_k &\geq T^k[\mathbf{h}(\mathbf{x}^k) + \nabla \mathbf{h}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k)] & k = 1, 2, \dots, K \\
\mathbf{C}\mathbf{x} + \mathbf{B}\mathbf{y} &\leq \mathbf{d} \\
\mathbf{x} \in \mathbf{X} &= \{\mathbf{x} : \mathbf{x} \in \mathfrak{R}^n, A_1\mathbf{x} \leq a_1\} \subseteq \mathfrak{R}^n \\
\mathbf{y} \in \mathbf{Y} &= \{\mathbf{y} : \mathbf{y} \in \{0-1\}^q, A_2\mathbf{y} \leq a_2\} \\
\sum_{i \in \mathbf{B}^k} y_i - \sum_{i \in \mathbf{NB}^k} y_i &\leq |B^k| - 1, \quad k = 1, 2, \dots, K-1 \\
s_k^0, p_{i,k}, q_{i,k} &\geq 0, \quad k = 1, 2, \dots, K
\end{aligned}$$

donde $\mathbf{p}_k = \{p_{i,k}\}$ y $\mathbf{q}_k = \{q_{i,k}\}$ son vectores positivos de variables de holgura, s_k^0 son escalares de holgura positivos y w_k^0 , $w_{i,k}^p$, $w_{i,k}^q$ son los pesos de las variables de holgura s_k^0 , $p_{i,k}$, $q_{i,k}$ respectivamente, que satisfacen las siguientes relaciones:

$$\begin{aligned}
w_k^0 &> |\bar{\mu}_k|, \\
w_{i,k}^p &> |\lambda_{i,k}|, \\
w_{i,k}^q &> |\mu_{i,k}|,
\end{aligned}$$

con $\bar{\mu}_k$, $\lambda_{i,k}$ y $\mu_{i,k}$ siendo los multiplicadores lagrangeanos del problema primal para $\mathbf{y} = \mathbf{y}^k$ escrito como:

$$\min_{\mathbf{x}} \quad \mathbf{c}^T \mathbf{y}^k + \mu \quad (3.14)$$

sujeto a:

$$f(\mathbf{x}^k) - \mu \leq 0 \leftarrow \bar{\mu}_k$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \leftarrow \lambda_k$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \leftarrow \mu_k$$

$$\mathbf{C}\mathbf{x} + \mathbf{B}\mathbf{y}^k \leq \mathbf{d}$$

$$\mathbf{x} \in \mathbf{X} = \{\mathbf{x} : \mathbf{x} \in \mathfrak{R}^n, A_1\mathbf{x} \leq a_1\} \subseteq \mathfrak{R}^n$$

Propiedad: Si (\mathbf{x}^k, μ) es un punto KKT (Karush-Kuhn-Tucker) de (3.13), entonces es también un punto KKT del problema maestro en $\mathbf{y} = \mathbf{y}^k$.

3.3.3 Desarrollo del Algoritmo

A continuación se enumeran los pasos del algoritmo:

1. Resolver la relajación del problema no lineal (3.12). Esto se logra tratando la variable y como una variable continua con $0 \leq y \leq 1$, para obtener (x^0, y^0) . Si y^0 es una combinación entero 0-1, terminar (“óptimo entero encontrado”). De otro modo continuar en el paso 2.
2. Resolver el problema maestro relajado (3.13) para identificar y^l .
3. Resolver el problema primal $P y^l$ (como en (3.14)) para encontrar la cota superior $UBD = P(y^l)$ al igual que los multiplicadores lagrangeanos.
4. Definir la matriz T^k de dimensión $(m \times m)$.
5. Resolver el problema maestro relajado (3.13) para determinar y^{k+1} y la cota inferior denotada como Z_L^K .
6. Repetir los pasos 3, 4 y 5 hasta que haya un incremento en el valor óptimo factible de los problemas NLP primales, punto en el cual terminamos.

Nota: En el paso 1 del algoritmo OA/ER/AP, no se requiere una combinación inicial 0-1, pero en lugar de ello se puede generar una, resolviendo una relajación NLP y subsecuentemente se resuelve el problema maestro relajado. Tal esquema puede ser aplicado en los algoritmos GBD, OA y OA/ER.

Observaciones:

1. El algoritmo puede terminar en el paso 1 si una combinación 0-1 se determina mediante la relajación NLP. Sin embargo, debe ser enfatizado que como la relajación NLP no satisface ninguna propiedad de convexidad en la variable x , la solución obtenida puede ser una solución local aunque haya sido identificada una combinación 0-1 para las variables y .
2. En el caso de que el problema maestro relajado produzca una combinación 0-1 en el punto y^{k+1} para el cual el problema primal es infactible, Viswanathan y Grossmann [41] sugieren las siguientes alternativas:
 1. Ignorar la infactibilidad de la solución primal x^{k+1} e introducir un corte entero que excluya la combinación y^{k+1} y resolver el problema maestro relajado nuevamente hasta generar una y para la cual el problema primal sea factible.

2. Agregar al problema maestro relajado las linealizaciones alrededor del punto infactible continuo. Note que para tratar al problema maestro relajado necesitamos tener información de los multiplicadores lagrangeanos. Para obtener dicha información, un problema factible necesita ser resuelto, y Viswanathan y Grossmann propusieron una formulación del problema factible, la cual es,

$$\min_{x, \alpha} \quad c^T y^k + f(x) + p\alpha \quad (3.15)$$

sujeto a:

$$\begin{aligned} h(x) &= 0 \\ g(x) &\leq \alpha \\ Cx + By^k - d &\leq \alpha \\ x \in X &= \{x : x \in \mathbb{R}^n, A_1x \leq a_1\} \subseteq \mathbb{R}^n \\ \alpha &\geq 0 \end{aligned}$$

Donde p es el parámetro de penalización y α es el vector de la variable de holgura aplicada solamente a las restricciones de desigualdad.

Si la solución al problema (3.15) tiene a $\alpha = 0$, entonces el primal tiene solución factible. Note que (3.15) no contiene suposiciones de convexidad y por ello solamente puede obtener soluciones locales.

Nota: El criterio de terminación en el paso 6 está basado en obtener un valor grande en los problemas primales para iteraciones consecutivas. Sin embargo, no existe razón teórica del porqué este debería ser el criterio de terminación dado que el problema primal por su definición no necesita satisfacer alguna propiedad de monotonía (por ejemplo, no necesita ser no creciente). Basados en lo anterior, este criterio de terminación puede únicamente ser visto como una heurística y como consecuencia puede llevar a la terminación prematura del OA/ER/AP. Además, el algoritmo OA/ER/AP puede fallar al identificar la solución global del problema MINLP.

3.3.4 Implementación del OA/ER/AP en GAMS: DICOPT

GAMS [5] es un modelador algebraico del cual nos auxiliamos para reproducir la formulación de nuestro problema. En este software se modela el MINLP algebraicamente

y se especifica que sea resuelto con el optimizador DICOPT. DICOPT [15] es una implementación del algoritmo OA/ER/AP desarrollada por Viswanathan, Grossmann (autores del mencionado algoritmo) y Vecchietti en colaboración con GAMS Development Corporation.

DICOPT comienza resolviendo el problema NLP en el cual la condición de integralidad de las variables enteras 0-1 está relajada. Si la solución a este problema contiene respuesta con variables enteras, la búsqueda termina (Paso 1 del algoritmo OA/ER/AP). De otra forma, continúa con una secuencia alternada de problemas no lineales (NLP) llamados subproblemas y programas lineales enteros-mixtos (MIP) llamados problemas maestros. Los subproblemas NLP son resueltos para variables fijas en valores 0-1 que son identificados por el problema maestro MIP en cada iteración (pasos 2, 3, 4 y 5). Un criterio de parada que suele funcionar muy bien en la práctica para problemas no convexos (como el nuestro) está basado en la regla heurística: detenerse tan pronto como los subproblemas NLP comiencen a empeorar (paso 6). Esto es, por ejemplo, el actual subproblema NLP tiene una función objetivo óptima que es peor que la del previo subproblema NLP.

Como se sabe, el desempeño de este tipo de métodos es sensible a una diversidad de condiciones algorítmicas. Por tal motivo, uno de los objetivos de esta tesis es realizar una evaluación algorítmica del método OA/ER/AP, en función de los siguientes parámetros.

1. Pre-procesamiento vs. no pre-procesamiento.
2. Criterio de parada, definido en el paso 6 del algoritmo OA/ER/AP.
3. Parámetro de penalidad aumentada. En el algoritmo es la variable p en la formulación (3.15).
4. Evaluación del valor del punto inicial en que se evalúa (2.1).

Estas pruebas se describen con más detalle en el siguiente capítulo.

CAPÍTULO 4

EVALUACIÓN COMPUTACIONAL

En este Capítulo, se muestra el trabajo experimental. Primeramente se describe la base de datos y posteriormente se explican los experimentos realizados:

- Evaluación algorítmica del método.
- Comparación del modelo MINLP con NLP con variables fijas.

Finalmente se muestran y discuten los resultados obtenidos y se presentan las conclusiones.

4.1 Descripción de Base de Datos

El propósito de diseñar y configurar una base de datos con instancias de problemas tiene dos objetivos:

1. Probar los algoritmos propuestos.
2. Crear un punto de referencia para hacer más confiables y eficiente las pruebas entre los algoritmos.

El primer paso para configurar una base de datos es definir las tareas que nuestro sistema debe efectuar. Para definir nuestra base de datos necesitamos especificar:

- (a) **Los datos del sistema.** La información que introduciremos a la computadora.
- (b) **Las funciones que procesará el sistema.** Los cálculos que la computadora debe realizar y la forma en que la información se transfiere entre los archivos de la base de datos.
- (c) **Las salidas del sistema.** La forma en la cual los datos serán desplegados en la pantalla, guardados en los archivos e impresos.

Como ya mencionamos en el Capítulo 1, en este estudio abordamos tres clases diferentes de topologías de red.

1. Sencilla ó gunbarrel, denominada tipo A (ver Figura 1.1),
2. Árbol o de ramas, llamada tipo B (ver Figura 1.2), y
3. Cíclica, designada tipo C, (ver Figura 1.3).

En [39], encontramos una pequeña base de datos (BD), la cual está compuesta por una instancia para cada tipo de red. Por lo tanto, una de las contribuciones de esta tesis es la de rediseñar y ampliar esa BD a problemas tipo MINLP.

Dada la precedencia del trabajo antes mencionado, decidimos adoptar la nomenclatura que en ese trabajo se utilizó, para extenderla a instancias más grandes. Nuestra BD y su nomenclatura se explican a continuación.

Por ejemplo, en las Figuras 1.1, 1.2 y 1.3 de red antes mencionadas, distinguimos tres tipos de vértices o nodos. Los nodos con líneas horizontales representan los nodos suministro u oferta, los nodos con líneas verticales, son los nodos demanda y los nodos grises sin líneas representan los nodos de paso. Una tubería está denotada por una flecha que une a cualesquiera dos nodos, y son conocidas como los arcos de la red. Estos arcos pueden ser de dos tipos: los arcos *ductos*, que son simplemente las flechas que indican la dirección del flujo y los arcos *estaciones compresoras*, que tienen un trapecoide gris entre los nodos que lo forman colocado sobre la flecha que indica la dirección del flujo.

Cada una de estas topologías toma su nombre de las características de la red y del tipo de compresor centrífugo que las compone. La primera parte es el vocablo *net* (que traducido del inglés significa red), el cual hace referencia a que nuestro problema está modelado como un problema de redes. La siguiente letra puede ser una *a*, si es una topología sencilla; una *b*, si se trata de una instancia de árbol; ó una *c* si en el problema hay algún ciclo.

El siguiente número es la cantidad de nodos en la red, seguido por una letra *c* (haciendo referencia a los compresores centrífugos), el próximo dígito nos indica el número de estaciones compresoras a lo largo de toda la red. Finalmente, el último número nos dice el tipo de compresor centrífugo utilizado en esa instancia. En esta tesis trabajamos con un total de 9 compresores centrífugos diferentes, por ello, el último dígito puede ser desde un número 1 hasta un número 9.

En la Tabla 4.1, enumeramos los nueve compresores centrífugos. Como ya explicamos anteriormente, este es el último dígito que indica el tipo de compresor centrífugo en cada instancia. También mostramos la relación entre el número asignado a estos compresores y sus nombres comerciales. Las topologías de red que componen nuestra base de datos se muestran en la Tabla 4.2 y se encuentran agrupadas de acuerdo al tipo de configuración a la que pertenecen. Son un total de 11 topologías distintas.

Número de compresor	Nombre comercial	Número de compresor	Nombre comercial	Número de compresor	Nombre comercial
1	bellvan-k1	4	rakeey-k1	7	bethany-k1
2	bellvan-k3	5	snarlin-k1	8	hamper-k1
3	bethany-k2	6	bellvan-k2	9	rakeey-k2

Tabla 4.1: Relación del número de compresor y su nombre comercial.

Topologías simples Tipo A	Topologías de árbol Tipo B	Topologías con ciclos Tipo C
net-a-5c1	net-b-10c3	net-c-6c2
net-a-6c2	net-b-11c4	net-c-8c3
net-a-8c3	net-b-14c6	net-c-9c3
		net-c-13c5
		net-c-15c5

Tabla 4.2: Listado de todas la topologías de nuestra base de datos.

Por ejemplo, la instancia *net-a-5c1-5* significa que es una topología de red del tipo simple (gunbarrel) con 5 nodos, una estación compresora y que los compresores centrífugos de la estación compresora son del tipo 5, es decir, está compuesta por compresores centrífugos snarlin-k1.

Las instancias totales se obtienen de la siguiente forma. Para cada una de las topologías de la Tabla 4.2 existen nueve instancias. Cada instancia se distingue por el tipo de configuración de la red además del tipo de compresor centrífugo usado. Sólo en la

topología *net-b-11c4* hay dos diferentes instancias para el problema *net-b-11c4-9* que difieren entre sí, por las ofertas y demandas en la red.

El número de instancias totales en nuestra BD se calcula como sigue. Son 3 topologías de red sencilla \times 9 compresores diferentes = 27 instancias de red simple. Más 3 topologías de red de árbol \times 9 compresores = 27 instancias + 1 instancia adicional llamada *net-c-11c4-9-b* (mismo compresor nueve, de la topología de red *net-b-11c4*, lo que diferencia a las instancias son la oferta y la demanda) = 28 instancias de árbol. Hay 5 topologías de ciclo \times 9 compresores = 45 instancias de ciclo. Esto da un total de 100 instancias.

4.2 Experimento A: Evaluación Algorítmica

Es esta parte describimos las pruebas algorítmicas hechas al método OA/ER/AP. Los parámetros probados fueron:

1. **Uso de pre-procesamiento.** Con esto pretendemos evaluar si las técnicas de pre-procesamiento (que incluyen el escalamiento, el acotamiento de las variables y la asignación del valor del punto inicial) nos ayudan a la pronta convergencia del algoritmo. Como referencia, para el caso NLP, se concluyó que emplear estas técnicas ayudan notablemente a la convergencia de una solución. Más detalles se pueden ver en las referencias [39, 40].
2. **Criterio de parada.** Aquí se prueban dos diferentes criterios de parada. Una es la regla heurística (descrita en el paso 6 del algoritmo OA/ER/AP). La otra tiene como criterio la infactibilidad en el problema maestro MIP ó límite de tiempo e iteraciones.
- 3 **Penalidad aumentada.** Aquí evaluamos tres valores diferentes del parámetro p en la formulación (3.15).
4. **El valor del punto inicial.** Sabemos este problema presenta características de no linealidad en las restricciones (2.4) y (2.6) y la función objetivo (2.1) y que el método de solución resuelve un subproblema no lineal. Es sabido que en los problemas NLP, al cambiar el valor del punto inicial, es posible que el método salga de alguna vecindad, donde haya quedado atrapado y cambie el valor de la

solución encontrada, es decir, las soluciones obtenidas son sensibles a la asignación realizada del punto inicial [39]. En (2.1), g_{ij}^1 es una función que depende del flujo másico x_{ij} , la presión de succión p_i y la presión de descarga p_j . Cuando introducimos el modelo MINLP en GAMS, la función g_{ij}^1 se traduce en una variable de decisión a la cual deseamos encontrar el valor mínimo posible y que sigue dependiendo de los valores de flujo por cada arco y de las presiones nodales obtenidas. De ahí que se ha decidido modificar el valor del punto inicial de g_{ij}^1 y realizar evaluaciones para decidir la importancia de este parámetro.

4.2.1 Diseño Experimental

En cada experimento se trata de evaluar el impacto de variar alguno de los cuatro parámetros explicados anteriormente, siendo esta variación lo único que distingue a las pruebas entre sí, para posteriormente comparar los resultados con estadística no paramétrica. Esta herramienta nos ayudará a decidir cuál valor de los parámetros evaluados es el mejor. En total se efectuaron 4 experimentos para realizar la evaluación algorítmica y en cada uno de ellos se evalúa el comportamiento del algoritmo según la modificación realizada.

Experimento 1

El primer experimento realizado fue para determinar si el utilizar las técnicas de pre-procesamiento ayuda a obtener mejores soluciones. Los criterios para determinar cuáles son mejores soluciones lo veremos más adelante.

El pre-procesamiento consiste básicamente en realizar una transformación del problema original en uno equivalente, antes de optimizar el problema. Esto se hace con el objetivo de obtener un problema equivalente con mejores propiedades para ser resuelto desde el punto de vista computacional.

A continuación explicamos las técnicas de pre-procesamiento que se utilizan en este trabajo.

1. *Escalamiento*. Consiste en proveer límites a las variables de decisión.
2. *Acotamiento de las variables*. Esto quiere decir que damos un intervalo de valores del cual cada variable no puede salirse, el menor valor recibe el nombre de cota inferior

y el mayor valor posible se denomina cota superior. Este se hace con la finalidad de reducir el espacio de búsqueda o región factible.

3. *Valor del punto inicial.* Aquí se asignan valores iniciales a las variables, de lo contrario, el algoritmo toma un punto arbitrario como punto inicial, lo cual puede ocasionar que el algoritmo no pueda encontrar una solución inicial factible para comenzar la búsqueda de una solución óptima. Es muy común que el valor inicial sea el θ , que en muchos casos es un muy mal valor inicial para una variable no lineal, como por ejemplo, si la variable en cuestión se encuentra en el denominador de alguna ecuación.

Las estadísticas que nos miden la calidad de las soluciones y/o desempeño del método son las siguientes:

- Intervalo de optimalidad relativa de la función objetivo (IOR),
- Número de iteraciones (NI),
- Recursos computacionales utilizados (T),
- Status Final del Modelo (SFM).

El intervalo de optimalidad relativa (IOR) se obtiene básicamente de la diferencia del valor de la función objetivo del problema MINLP y una cota inferior del problema. En este caso como cota inferior usamos el valor de la solución del problema relajado RMINLP con la aclaración que éste es un óptimo local y no global. El status final del modelo igual a 2, indica que la solución encontrada es un óptimo local, el status igual a 8, significa que se ha encontrado una solución factible, y el status final del modelo igual a 9 significa que el optimizador no fue capaz de encontrar una solución que satisficiera la restricción de integralidad de la restricción (2.7) del modelo de nuestro problema. También puede darse el caso que el optimizador (DICOPT) no sea capaz de encontrar alguna solución al problema, lo cual será reportado en la sección 4.2.2. (Para más detalles acerca de todos los status final del método ver referencia [5]). Los resultados del experimento se presentan en la siguiente sección.

En cuanto a NI y T, es deseable que sean lo menor posible y se traduce como el número de iteraciones necesarias y los recursos empleados por el optimizador para resolver cada instancia del problema MINLP.

Experimento 2

Aquí evaluamos dos diferentes reglas para la terminación del algoritmo AO/ER/AP, que denominamos:

1. **Regla S1:** Esta opción (stop 0 en DICOPT) le indica al algoritmo iterar hasta que:
 - (a) Se alcance el número máximo de iteraciones menores¹ en cada subproblema NLP y problema maestro MIP. El default es 1000;
 - (b) Se agoten los recursos computacionales. Esto se refiere al tiempo total en que DICOPT y los subproblemas se ejecutan. El default son 1000 segundos²;
 - (c) Se llegue al límite máximo de iteraciones principales³, es decir las iteraciones de DICOPT. El default son 20 iteraciones principales.
 - (d) Una infactibilidad del problema maestro MIP vuelva infactible la solución del problema.

Esta opción se usa para verificar que DICOPT no detiene la búsqueda muy pronto porque agota todas las posibilidades, a diferencia de otros criterios de parada.

2. **Regla S2:** El algoritmo itera (opción stop 2 en DICOPT) hasta que la solución del subproblema NLP deje de mejorar. Este criterio es en realidad una heurística. Para problemas no convexos en los cuales cotas válidas no pueden ser obtenidas, esta heurística funciona muy bien. Inclusive en problemas convexos en muchos casos termina la búsqueda demasiado pronto, entregando una solución óptima o de buena calidad.

¹Este parámetro se puede cambiar con la opción *INTERLIM*.

²Este valor se puede cambiar con la opción *RESLIM*.

³Este valor se puede modificar con la opción *MAXCYCLES*.

Experimento 3

Este experimento se realizó con la finalidad de evaluar el desempeño del algoritmo en función del parámetro de penalidad aumentada, representado por la variable p en la formulación (3.15). Se prueban tres diferentes valores (100, 1000 y 100,000). En la siguiente sección presentamos los resultados de la evaluación del parámetro penalidad aumentada.

Experimento 4

Este último experimento se hace para evaluar si el cambio en el valor del punto inicial del consumo de combustible total de las estaciones compresoras a lo largo de toda la red, repercute en la calidad de las soluciones encontradas. Esta variable está representada por $g_{i_j}^1$ en la ecuación (2.1).

Como ya hemos explicado, dentro de las técnicas de pre-procesamiento, si el valor inicial de la variable no se define, entonces al algoritmo puede dificultársele encontrar una solución óptima, o factible al menos.

También ya dijimos que es preferible acotar las variables, para reducir el espacio de búsqueda de las soluciones factibles, y para asegurarnos que la solución será un valor que cumpla con todas las restricciones del modelo matemático.

Los resultados de la comparación hecha entre las cuatro diferentes clases de programas se muestran en la siguiente sección.

4.2.2 Resultados

Experimento 1

Para presentar las observaciones obtenidas en el experimento, agrupamos los resultados de acuerdo a la topología de red a la que pertenecen las instancias.

Recordemos que las estadísticas que nos miden la calidad de las soluciones y/o desempeño del método, son:

IOR. Valor del intervalo de optimalidad relativa (porcentaje).

NI. Número de iteraciones que efectuó el optimizador,

T. Tiempo de CPU utilizado por el optimizador para resolver el problema,

SFM. Status final del modelo.

En la Tabla 4.3 se exponen los resultados de la comparación entre efectuar o no el pre-procesamiento. En la primer columna se encuentran los criterios de evaluación. En la segunda, cuarta y sexta columna aparecen los valores promedios de los estadísticos obtenidos al resolver los programas con el pre-procesamiento, en las topologías de red A, B y C, respectivamente. En la tercer, quinta y séptima columna están los valores promedios del resultado de la ejecución de los programas sin el pre-procesamiento, de las topologías de red A, B y C respectivamente.

	Red tipo A		Red tipo B		Red tipo C	
	Con	Sin	Con	Sin	Con	Sin
IOR	108.05%	108.02%	21.30%	22.34%	12.21%	13.44%
NI	878.66	107.49+	113.28	162.06+	232.88	221.19
T	2.46	251.49	665.59	515.18	0.661	0.666

Tabla 4.3: Evaluación del pre-pocesamiento. (+) $1 \times 10 + 03$.

Se puede observar en la Tabla 4.3 que cuando no se usa el pre-procesamiento, para las instancias de topología B, el valor del intervalo de optimalidad relativa (IOR) es menor y disminuye el tiempo empleado por el optimizador para resolver las instancias. El IOR para las instancias A con y sin pre-procesamiento es prácticamente el mismo. En cambio, utilizando el pre-procesamiento, el IOR es mejor para las instancias tipo B y C, número de iteraciones es menor para las instancias A y B y el tiempo requerido por el método es menor para las redes del tipo C, siendo aún menor para las del tipo A (disminuye alrededor del 99%). Si eliminamos la instancia net-a-7c3-3, cuyo valor de IOR es mucho mayor (1869%), el IOR para las instancias del tipo A sin usar el pre-procesamiento y usándolo disminuyen a 24.18%.

En la Tabla 4.4 se muestran cuántas instancias resultaron infactibles al aplicar el optimizador, usando el pre-procesamiento y sin utilizarlo y el número total de instancias probadas en cada topología de red.

Tipo de Topología	con	sin	Total de instancias
A	5	5	27
B	6	7	28
C	10	10	45

Tabla 4.4: Instancias sin solución al emplear ó no el pre-procesamiento.

Se puede observar que para las instancias de la topología de red tipo A y C, usando y sin usar el pre-procesamiento, el optimizador obtiene 5 y 10 instancias infactibles respectivamente. Para las topologías del tipo B usar el pre-procesamiento reduce el número de instancias infactibles de 7 (sin usarlo) a 6 (utilizándolo), respectivamente.

Con las instancias que lograron ser resueltas, se realizó una prueba de estadística no paramétrica [42] (prueba de los signos), para los criterios de evaluación (T, IOR y NI).

Para cada criterio se define a μ_1 como resultados cuando los programas se ejecutan sin el pre-procesamiento. μ_2 representa las evaluaciones cuando los programas utilizan el pre-procesamiento. La hipótesis nula H_0 afirma que las soluciones encontradas usando el pre-procesamiento se encuentran más alejadas del óptimo local del problema relajado y que emplean más iteraciones y tiempo, es decir, $H_0 : \mu_2 \geq \mu_1$. La hipótesis alternativa H_1 afirma que utilizar el pre-procesamiento reduce el tiempo y las iteraciones empleadas por el método, al igual que las soluciones encontradas están cerca del óptimo local del problema RMINLP, en otras palabras, $H_1 : \mu_2 < \mu_1$.

Se obtuvieron los siguientes resultados: Para las instancias del tipo A y B, en ninguno de los criterios se obtuvo evidencia suficiente para rechazar la H_0 en favor de H_1 , al igual que en las instancias tipo C para T. En cambio, para las instancias tipo C, para NI se obtuvo que se debía rechazar H_0 en favor de H_1 , es decir, usando el pre-procesamiento encendido reducimos el número de iteraciones y mejoramos la calidad (y cantidad) de las instancias solucionadas.

Cabe señalar que para las instancias tipo C, que con la prueba de los signos se obtiene un valor de probabilidad de significancia (o valor-p) de 0.06 y debido a que el nivel de significancia es del 5%, se concluyó que no se podía rechazar la hipótesis nula en favor de la

hipótesis alternativa. Pero si el nivel de significancia hubiera sido del 10%, se habría tenido que rechazar la H_0 en favor de H_1 , es decir, no usar el pre-procesamiento obtiene mejores resultados.

No fue posible encontrar una cota inferior a todas las instancias probadas aunque algunas veces estas instancias sí lograron ser resueltas como problemas MINLP, tal es el caso de las instancias: net-c-8c3-8, net-b-11c4-6, net-c-6c2-1 y net-c-6c2-8.

En seguida, en la Tabla 4.5, se presentan los resultados obtenidos en cuanto al tipo de solución obtenida (SFM), las cuales pueden ser: óptimo local, solución factible, solución infactible y sin solución encontrada por el optimizador, con el pre-procesamiento y sin utilizarlo.

Tipo de Topología	Número de Instancias	Óptimo Local		Solución Factible		Solución Infactible		Sin solución	
		con	sin	con	sin	con	sin	con	sin
A	27	2	3	20	19	5	5	0	0
B	28	2	2	20	19	4	5	2	2
C	45	10	11	25	24	7	7	3	3
Total	100	14	16	65	62	16	17	5	5

Tabla 4.5: Resultados del status final, utilizando y sin usar el pre-procesamiento.

En la Tabla 4.5 se puede ver que usando el pre-procesamiento, en las tres topologías de red, se obtienen 14 óptimos locales, 65 soluciones factibles, 16 soluciones no enteras y 5 instancias quedan sin resolverse. Cuando no se usa el pre-procesamiento, aumentan 2 instancias de óptimos locales, pero disminuyen 3 soluciones factibles, se incrementa en 1 las soluciones infactibles y el número de instancias sin solución permanece igual. Entre soluciones óptimas y factibles se obtienen 79 usando el pre-procesamiento, contra 78 sin utilizar el pre-procesamiento.

Debido a que usando el pre-procesamiento, se reducen tanto el número de iteraciones como el tiempo empleado por el optimizador, que obtenemos más soluciones óptimas locales y factibles y que el IOR es el mismo en las instancias A y es mejor en las del tipo C, emplearemos el pre-procesamiento en los siguientes experimentos.

Experimento 2

En la Tabla 4.6 se muestran los resultados de la comparación al usar el criterio de parada s1 y el criterio de parada s2, descritos en la sección 4.2.1. Los resultados están dispuestos de forma similar a la Tabla 4.3.

	Red tipo A		Red tipo B		Red tipo C	
	s1	s2	s1	s2	s1	s2
IOR	48.14%	108.02%	10.07%	21.64%	5.95%	12.31%
NI	1.10*	107.49+	1.37*	204.87+	1.37*	222.62
T	1.74+	284.15	3.70+	869.91	2.52+	0.68

Tabla 4.6: Evaluación del criterio de parada ($s1=stop 0$ y $s2=stop 2$). (*) $1 \times 10 + 06$, (+) $1 \times 10 + 03$.

Se puede observar de la Tabla 4.6 que al usar s1 se reduce el índice IOR en promedio para todas las instancias, pero si eliminamos del promedio algunas instancias que resultaron muy malas (valor mayor a 100%), para las instancias tipo A, usando s1 el IOR baja a un 6.14% y para s2 el IOR vale 5.25%. en las instancias del tipo B, usando s1 el IOR se reduce a 8.59% y con s2 el IOR vale 15.92%. Finalmente el IOR de las instancias tipo C se queda en 5.95% y usando s2 se reduce a 10.59%.

Para lograr un menor número iteraciones y tiempo empleado, es preferible usar s2 en las redes tipo A, B y C. Ambos criterios son significativamente mejores para las instancias del tipo C.

En la Tabla 4.7, se muestra el número de instancias en las que el método falló al encontrar una solución al usar los diferentes criterios de parada.

Se observa que usando s1 no se logran solucionar 5 instancias del tipo A y 8 del tipo B, al igual que cuando se reemplaza por s2. Para las instancias del tipo C, usando s2 incrementa en 1 instancia el número de instancias en las que el optimizador no encuentra soluciones ni óptimas ni factibles.

De las instancias que lograron ser resueltas, el resultado de SFM usando s1 y usando s2 se resume en la Tabla 4.8. Se observa que se obtiene el mismo número de óptimos locales para s1 y s2, pero las soluciones factibles disminuyen en 1 con s2, el número de soluciones

Tipo de Topología	s1	s2	Total de instancias
A	5	5	27
B	8	8	28
C	8	9	45

Tabla 4.7: Instancias donde el optimizador falló usando s1 y s2.

infactibles aumenta en 2 con s2 y disminuyen en 1 las instancias sin solución. Entre soluciones óptimas y factibles se obtienen 79 usando s1 contra 78 usando s2.

Tipo de Topología	Número de Instancias	Óptimo Local		Solución Factible		Solución Infactible		Sin solución	
		s1	s2	s1	s2	s1	s2	s1	s2
A	27	2	2	20	20	3	5	2	0
B	28	1	1	19	19	7	7	1	1
C	45	11	11	26	25	6	6	2	3
total	100	14	14	65	64	16	18	5	4

Tabla 4.8: Status final del modelo usando s1 y s2.

Posteriormente, a estas instancias se les realizaron pruebas de estadística no paramétrica, para los criterios de evaluación IOR, T y NI. Para cada criterio se define a μ_1 como los resultados de las evaluaciones utilizando como criterio de parada s1 y μ_2 como los resultados de las evaluaciones cuando cambiamos el criterio de parada por s2. Se prueba, la hipótesis nula $H_0 : \mu_2 \geq \mu_1$ contra la hipótesis alternativa $H_1 : \mu_2 < \mu_1$. H_0 representa la hipótesis de que los resultados obtenidos con s1 son mayores o iguales que cuando se usa s2 y H_1 indica que se obtienen mejores (menores) soluciones usando s2.

Para los casos de NI y T, la prueba de hipótesis arrojó por resultado que se rechazaba la hipótesis nula en favor de la alternativa, para todos los tipos de instancias, al igual que para el IOR de las topologías A y B. Para las instancias del tipo C, se obtiene de la prueba de hipótesis para el criterio IOR, que usando s1 obtenemos mejores valores (estamos más cerca de la cota inferior) que con s2. Sin embargo, no se lograron obtener cotas inferiores

(soluciones al problema RMINLP) para las instancias: net-c-6c2-1 y net-c-8c3-8, tanto para s1 como para s2.

En cuanto a los valores obtenidos para el status final del método observamos se obtuvieron igual número de óptimos locales, se incrementa el 1 las soluciones factibles usando s1 y aumentan en 2 las soluciones infactibles usando s2. Las instancias en las que no se encontró solución fueron 5 con s1 y 4 con s2. En total entre soluciones óptimas y factibles, se resolvieron 79 instancias usando s1 contra 78 usando s2.

El status final de modelo permaneció casi igual usando s1 ó s2, excepto por una instancia del tipo C. Pero el número de iteraciones y el tiempo empleado por el optimizador mejoran considerablemente cuando empleamos la heurística. En cuanto a las pruebas de hipótesis para el IOR, para las instancias del tipo A, después de eliminar las instancias críticas se prefiere usar s2, aunque no para las instancias del tipo B y C. Por consideraciones de tiempo, decidimos usar el criterio de parada s2 en las subsecuentes pruebas realizadas.

Experimento 3

En la Tabla 4.9, se exponen los resultados de la comparación cuando variamos el valor de p (penalidad aumentada) fijando el mismo en los valores: 100 (caso I), 1000 (valor default, caso II) y 100000 (caso III). En la primera columna aparecen los tres tipos de topologías existentes. En la segunda se muestran los tres diferentes casos. En la tercera se encuentran los resultados promedio del criterio IOR. En la cuarta columna están los resultados promedios de NI y finalmente en la quinta columna están los resultados promedio de T.

En todos estos casos están activos el pre-procesamiento y el criterio de parada s2, que determinamos en los experimentos previos como los mejores valores de estos parámetros.

Se observa que se obtienen los mismos resultados para IOR en las tres topologías de red, sin importar el valor de p utilizado. Si eliminamos la instancia crítica net-a-7c3-3 en los tres casos, el promedio del IOR en los tres casos se reduce a un 24.18%. Para NI en las instancias del tipo C tampoco importa el valor de p , pues en los 3 casos el número de iteraciones es el mismo, pero en las del tipo A el mejor resultado se obtiene en el caso II. Para el criterio NI, los casos II y II obtienen las mejores evaluaciones. En cuanto a T, el caso II reporta mejores

Tipo de Topología	Caso	IOR	NI	T
A	I	108.02%	125.16+	314.22
	II	108.02%	107.49+	287.01
	III	128.46%	124.33+	331.87
B	I	21.64%	162.06+	757.02
	II	21.64%	162.05+	710.59
	III	21.64%	162.05+	726.56
C	I	12.31%	222.28	0.74
	II	12.31%	222.28	0.68
	III	12.31%	222.28	0.69

Tabla 4.9: Evaluación del parámetro p (penalidad aumentada). (+) $1 \times 10 + 03$.

soluciones para todas las instancias existentes.

En la Tabla 4.10 se muestran cuántas instancias no lograron ser resueltas por el optimizador, en cada uno de los 3 casos para cada topología de red. Se puede observar que en todos los casos 5 instancias del tipo A son infactibles, para las redes tipo B, el optimizador no puede resolver 8 instancias y para las topologías de red tipo C, el optimizador falla en 9 instancias, sin importar el valor de p que se fije.

Tipo de Topología	I	II	III	Total de instancias
A	5	5	5	27
B	8	8	8	28
C	9	9	9	45

Tabla 4.10: Instancias que no lograron ser resueltas por el optimizador, al variar el valor de p .

Se presentaron instancias en que el el problema RMINLP (problema relajado) resultó infactible, pero el problema MINLP se resolvió existosamente. Tales instancias son:

net-c-6c2-1 y net-c-8c3-8 en los tres casos.

A las instancias que sí lograron resolverse, se les aplicó una prueba de hipótesis con estadística no paramétrica (Kruskal-Wallis), para los criterios de evaluación NI y T. Para IOR no fue necesario realizar las pruebas de hipótesis, pues los resultados en cada instancia fueron los mismos, al igual que para las instancias del tipo C del criterio NI. Para ello, se define a μ_1 como los valores obtenidos a los resultados del caso I ($p=100$), μ_2 son los resultados obtenidos del caso II ($p=1000$) y μ_3 son las evaluaciones en el caso III ($p=100000$). Se desea probar la hipótesis nula de que los resultados obtenidos son iguales independientemente del valor de p , es decir, $H_0 : \mu_1 = \mu_2 = \mu_3$, contra la hipótesis alternativa $H_1 : \text{al menos un valor es diferente}$, que significa que al menos uno de los resultados obtenidos para cada caso es diferente a las demás mediciones.

Para todas las instancias del criterio T, al igual que para las instancias tipo A y B del criterio NI los resultados obtenidos de las pruebas de hipótesis indican que no hay evidencia suficiente para rechazar H_0 en favor de H_1 , ningún caso. Es decir, las pruebas de hipótesis no pueden demostrar que el número de iteraciones de las instancias tipo A, ni el tiempo empleado para todas las instancias, varía si cambiamos el parámetro p .

En cuanto al status final del modelo, podemos ver en la Tabla 4.11 los resultados obtenidos al asignar al parámetro p los tres diferentes valores, según cada caso. Para todos los casos se obtienen 14 óptimos locales, 64 soluciones factibles, 17 soluciones infactibles y 5 instancias sin solución, independientemente del valor de p usado.

Tipo de Topología	Número de Instancias	Óptimo Local			Solución Factible			Solución Infactible			Sin solución		
		I	II	III	I	II	III	I	II	III	I	II	III
A	27	2	2	2	20	20	20	5	5	5	0	0	0
B	28	1	1	1	19	19	19	6	6	6	2	2	2
C	45	11	11	11	25	25	25	6	6	6	3	3	3
Total	100	14	14	14	64	64	64	17	17	17	5	5	5

Tabla 4.11: Evaluación de p de acuerdo al status final del optimizador.

La suma de soluciones óptimas y soluciones factibles, en los tres casos es 78.

En consecuencia decidimos efectuar la siguiente evaluación al algoritmo usando fijo el valor del parámetro p (penalidad aumentada) en 1000 para el siguiente y último experimento de los parámetros del algoritmo OA/ER/AP.

Experimento 4

Para probar el impacto de asignar valores diferentes al punto inicial, se proponen cuatro casos. En cada caso se varía el porcentaje del valor inicial de la variable g_{ij}^1 de (2.1) de su cota inferior. Se define el valor de la variable al 1% de la cota inferior (caso i), después al 5% (caso ii), posteriormente al 10% (caso iii) y finalmente en el punto medio, es decir, al 50% (caso iv). Los resultados están dispuestos en la Tabla 4.12.

Tipo de Topología	Caso	IOR	NI	T
A	i	125.11%	63.28+	178.64
	ii	103.69%	56.85+	183.30
	iii	120.36%	124.33+	290.71
	iv	108.02%	107.49+	264.84
B	i	21.70%	144.91+	584.27
	ii	23.33%	157.96+	704.70
	iii	24.05%	202.62+	834.45
	iv	21.64%	162.05+	704.70
C	i	11.94%	233.70	0.68
	ii	12.85%	464.33	0.74
	iii	11.99%	240.14	0.67
	iv	12.31%	222.62	0.73

Tabla 4.12: Evaluación del parámetro punto inicial de g_{ij}^1 . (+) $1 \times 10 + 03$.

En la primer columna se encuentran los tres tipos de topologías de redes. En la segunda columna se encuentran los cuatro casos. En la tercer columna se encuentran los resultados

promedio del criterio IOR. En la cuarta columna estan los resultados promedios de NI y finalmente en la quinta columna estan los resultados promedio de T.

En todos estos casos están activos el pre-procesamiento, el criterio de parada s2 y el valor del parámetro penalidad aumentada = 1000, que se determinaron como los mejores valores de estos parámetros en los experimentos anteriores.

Cuando eliminamos de los cuatro casos la instancia crítica para IOR (net-a-7c3-3 con un valor de IOR igual a 1868.76). El valor de IOR en todas las topologías del tipo A se reducen. Para el caso i se obtiene 24.18, para el caso ii se logra 19.64, para el caso iii el nuevo IOR vale 28.31 y para el caso iv el IOR se reduce a 24.18.

En la Tabla 4.12 se observa que el caso i obtiene los mejores resultados para IOR de las instancias C, para NI de las topologías B y para T de las topologías de red A y B. El caso ii resulta mejor para las topologías A de los criterios IOR y NI. El caso iii obtiene los mejores resultados para las instancias tipo C del criterio T. Finalmente, el caso iv reporta los mejores resultados para las instancias B en cuanto a IOR y para las instancias tipo C para NI.

Tipo de Topología	i	ii	iii	iv	Total de instancias
A	8	5	7	5	27
B	9	8	8	8	28
C	8	10	9	9	45
Total	25	23	24	22	100

Tabla 4.13: Instancias con soluciones infactibles al evaluar el punto inicial.

En la Tabla 4.13 se muestra cuántas instancias no lograron ser resueltas por el optimizador, para cada caso y en cada topología de red. Para el caso i se encontró que el optimizador falló al encontrar soluciones para 8 instancias del tipo A y C y 9 instancias del tipo B (total de 25). En el caso ii el optimizador falla en 5 instancias del tipo A, 8 del tipo B y 10 del tipo C (total de 23). En el caso iii, se encuentran soluciones infactibles para 7 instancias del tipo A, 8 del tipo B y 9 del tipo C (total de 24). Finalmente, no se obtienen soluciones óptimas o factibles en 5 instancias del tipo A, 8 del tipo B y 9 del tipo C (total de 22). El menor número de instancias infactibles se obtiene en el caso iv (cuando el punto

Tipo de solución	Topología de red	Caso				Total de instancias
		i	ii	iii	iv	
Óptimo Local	A	3	16	5	3	27
	B	1	18	8	1	28
	C	9	28	6	2	45
	Total	13	11	11	14	100
Solución Factible	A	3	19	5	0	27
	B	2	18	7	1	28
	C	6	29	8	2	45
	Total	62	66	65	64	100
Solución Infactible	A	3	17	4	3	27
	B	3	17	7	1	28
	C	5	31	7	2	45
	Total	19	20	18	17	100
Sin Solución	A	2	22	5	0	27
	B	1	19	6	2	28
	C	11	25	6	3	45
	Total	6	3	6	5	100

Tabla 4.14: Resultados del status final cuando se modifica el valor del punto inicial de g_{ij}^1 .

inicial es punto medio de las cotas).

Los resultados del status final del modelo, para las instancias que lograron ser resueltas, se exponen en la Tabla 4.14. Se observa que para el caso i se obtienen 13 óptimos locales, 62 soluciones factibles, 19 soluciones no factibles y 6 instancias sin solución. Para el caso ii se obtienen 11 óptimos locales, 66 soluciones factibles, 20 soluciones no factibles y 3 instancias sin solución. Para el caso iii se obtienen 11 óptimos locales, 65 soluciones factibles, 18 soluciones no factibles y 6 instancias sin solución. Para el caso iv se obtienen 14 óptimos locales, 64 soluciones factibles, 17 soluciones no factibles y 5 instancias sin solución. Agrupando las soluciones óptimas locales y las soluciones factibles, para el caso i

se obtienen 75 instancias, para el caso ii 76 instancias, para el caso iii se obtienen 76 instancias y para el caso iv, 78 instancias.

Finalmente, se efectúan las pruebas de hipótesis realizadas con estadística no paramétrica (Kruskal-Wallis), para cada caso en cada topología de red existente, para los criterios IOR, T y NI. Se define a μ_1 a los resultados obtenidos de las instancias en el caso i, μ_2 son los resultados de las instancias con el caso ii, μ_3 representan las soluciones para el caso iii y μ_4 son las soluciones obtenidas para el caso iv. Se desea probar la hipótesis nula $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$, contra la hipótesis alternativa $H_1 : \mu_1 \neq \mu_2 \neq \mu_3 \neq \mu_4$. Para el criterio IOR, H_0 se interpreta como: ‘los valores de las soluciones en los criterios de evaluación son los mismos en los cuatro casos’ y H_1 significa ‘los valores de las soluciones obtenidas en cada uno de los casos son diferentes entre sí’.

Los resultados de las 12 pruebas estadísticas arrojan que no se encontró evidencia suficiente para rechazar H_0 en favor de H_1 , es decir, no hay suficiente evidencia para decir que los resultados de los cuatro diferentes puntos iniciales al menos alguno es diferente de los demás.

En consecuencia obtuvimos que el valor del punto inicial que arrojó mejores resultados fue para IOR, NI y T depende de la topología de red de la que se trate, habiendo resultado el caso i el más frecuente. Pero el mayor número de soluciones óptimas y factibles se obtiene con el caso iv (78), al igual que para el menor número de instancias sin solución (22). Por lo cual se recomienda emplear un valor inicial para la variable g_{ij}^1 alejada un 1%, un 5% ó un 50% de la cota inferior, según lo que se desea encontrar (mejorar el IOR, aumentar las soluciones factibles o reducir el tiempo y las iteraciones).

4.2.3 Conclusiones

Como resultado de las comparaciones hechas en el experimento 1 (usar o no usar el pre-procesamiento) de las pruebas de hipótesis concluimos que aunque no se encuentra evidencia suficiente para afirmar que usando el pre-procesamiento se obtienen en promedio mejores resultados, para el criterio número de iteraciones se reducen sustancialmente las iteraciones para las topologías del tipo C. En promedio el tiempo empleado se reduce usando el pre-procesamiento para instancias del tipo A (hasta el

99%) y C. Además empleando el pre-procesamiento solo a una instancia no se logra encontrar su cota inferior (problema RMINLP). Del criterio IOR, se obtuvieron mejores resultados para las instancias tipo B sin usar el pre-procesamiento, para las instancias del tipo C se lograron mejores resultados usando el pre-procesamiento y para las instancias del tipo A fue indistinto. Sin utilizar el pre-procesamiento aumenta en 1 instancia sin solución para las topologías del tipo B. Por lo tal concluimos que usando el pre-procesamiento se obtienen más beneficios pues los resultados obtenidos en promedio indican que reduce drásticamente el número de iteraciones necesarias y el tiempo empleado por el optimizador para resolver las instancias y que obtiene buenos resultados para el IOR en instancias del A (24.18% cuando eliminamos la instancia crítica) y mejores para las del tipo C (12.21% en promedio).

Como resultado de las comparaciones hechas en el experimento 2 (decidir entre los dos diferentes criterios de parada), las pruebas de hipótesis en los tres tipos de instancias para NI y T indican que, es preferible usar s_2 , pues entrega soluciones con mayor rapidez, al igual que obtiene soluciones de mejor calidad para las instancias del tipo A en el criterio IOR. Se observa que para las instancias del tipo B si el nivel de significancia hubiera sido del 1%, también se hubiera preferido s_2 sobre s_1 . El número de instancias que no logran resolver el problema RMINLP pero si el MINLP son el mismo número y las mismas instancias (2). Se señala que se obtienen en promedio menores resultados en cuanto al tiempo y el número de iteraciones con la heurística (s_2), y por el factor tiempo decidimos que emplear como criterio de parada s_2 .

Cuando se prueban los 3 diferentes valores para el parámetro de penalidad aumentada (experimento 3), no fue necesario efectuar las pruebas de hipótesis en todos los criterios, ya que las evaluaciones para el criterio IOR en todas las topologías de red y para el criterio número de iteraciones de las instancias del tipo C fueron idénticas. En los demás casos, resultó que no había evidencia suficiente para decir que los resultados eran diferentes entre ellos. Aunque los valores promedio de las instancias solucionadas indican que para el criterio T y NI, es preferible usar el valor de p igual a 1000 (en promedio, no surte algún efecto el cambiar el valor default). En los tres casos probados se encontraron igual número de soluciones con soluciones infactibles (5 tipo A, 8 tipo B y 9 tipo C) y las mismas instancias sin cota inferior (un total de 2). Por lo tanto decidimos emplear el valor default (1000).

Para evaluar el mejor valor para el parámetro del punto inicial de g_{ij}^1 , las pruebas de hipótesis realizadas, indican que no hay diferencia en la asignación del punto inicial para IOR, NI y T. El caso que se repite con más frecuencia es el de las evaluaciones hechas alejados un 1% de la cota inferior. En promedio el mejor IOR fue diferente en cada instancia evaluada, para las instancias del tipo A, el mejor resultado se obtuvo cuando asignamos el valor inicial al 5% de la cota inferior, para las instancias del tipo B resultó mejor la asignación del punto inicial igual al 50% y para las instancias del tipo C, la mejor asignación fue al 1% de la cota inferior. En suma, el menor número de soluciones infactibles encontrados resultan con la aplicación del punto inicial igual al punto medio de sus cotas (inferior y superior), al igual que se obtuvieron el mayor número de soluciones factibles.

En resumen las conclusiones del experimento A:

1. El auto-pre-procesamiento encendido ayudó al optimizador a converger más rápido.
2. La heurística como criterio de parada ayudó a disminuir el número de iteraciones y los recursos computacionales requeridos por el algoritmo (tiempo).
3. El mejor valor encontrado para la penalidad aumentada fue 1000, es decir el default.
4. Encontramos que el valor del punto inicial podía ser diferente del punto medio entre las cotas. Si deseamos mejorar la evaluación de la función objetivo (IOR), debemos usar un valor inicial del 1% (topologías tipo C), del 5% (topología A) y 50% para las tipo B de la cota inferior de la variable g_{ij}^1 .

4.3 Experimento B: Comparación de Problemas Enteros Mixtos No Lineales y Problemas No Lineales con Variables Fijas

Como ya mencionamos en el Capítulo 3, los problemas enteros mixtos no lineales como el nuestro, han sido tratados no propiamente como problemas MINLP, sino que se han tratado de resolver fijando alguna variable.

Por ejemplo, tenemos conocimiento que el trabajo de la referencia [46], trabajan con el número de compresores en cada estación compresora como una variable de decisión. Sin

embargo, el modelo no es un MINLP. Ellos determinan en un primer plano la cantidad de flujo a través de la estación compresoras, y después, en un segundo plano, encuentran el número óptimo de unidades para ese flujo particular. Esta aproximación desde luego, limita la búsqueda de un óptimo global.

En otros trabajos como [39], el número de estaciones centrífugas no es considerado variable de decisión, sino que es tratado como un parámetro al cual se le asigna un valor (en este trabajo fue 1) a las unidades centrífugas en cada estación compresora y el problema se trabaja como un problema MINLP.

Otra alternativa podría ser resolver el problema anterior, fijando de antemano en diferentes valores el valor del número de estaciones compresoras encendidas y verificar cuál de esos valores arroja una solución factible.

Con el experimento B queremos resaltar la importancia de tratar a este problema como un MINLP, en el cual, todas las variables son tratadas al mismo nivel y comparar estos resultados con las metodologías expuestas anteriormente.

También para realizar las ejecuciones de los programas *instancia.gms* emplearemos las conclusiones que obtuvimos del experimento A (auto-pre-procesamiento encendido, criterio de parada s_2 , penalidad aumentada igual a 1000 y punto inicial fijo en el 1% del valor de la cota inferior de la variable g_{ij}^1).

4.3.1 Diseño Experimental

Este experimento se desarrolló para algunas instancias para reafirmar la importancia de este trabajo. Primeramente, creamos un programa *instanciar.gms* en el cual la variable número de compresores encendidos dentro de cada estación compresora no es una variable entera, sino continua. Esta relajación convierte al problema original (MINLP) en un problema relajado MINLP (RMINLP), que es básicamente un NLP.

Después creamos programas llamados *instanciaf.gms*, donde f es el valor en que la variable número de compresores funcionando en la estación compresora toma el valor f (puede ser un 1, 2 o 3) y se convierte en parámetro, por lo tanto, trabajaremos nuevamente con programas NLP.

Finalmente, resolveremos los problemas cuando el problema es un MINLP como ya describimos anteriormente.

4.3.2 Resultados

Instancia	RMINLP	Número de compresores	NLP-2	NLP-3	MINLP	Número de compresores
net-a-5c1-1	2.102*	4.3	2.566*	2.102*	2.102*	3
net-a-5c1-3	1.592*	3.6	1.592*	1.592*	1.592*	2
net-a-5c1-5	2.817*	3.2	2.817*	2.817*	2.817*	3
net-b-10c3-1	5.886*	2.1, 1.9, 1.2	6.500*	8.737*	6.380*	2, 1, 1
net-b-11c4-2	4.835*	3.1, 1.4, 2.05, 2.8	7.201*		4.942*	2, 1, 1, 1
net-c-13c5-6	3.900*	2.1, 2.3, 1, 2.6, 1.7	5.738*		4.516*	2, 1, 1, 1, 1

Tabla 4.15: Evaluación de los programas RMINLP, NLP-f y los MINLP. (*) $1 \times 10 + 06$.

En la Tabla 4.15 vemos los resultados de las evaluaciones hechas en la función objetivo. En la primer columna está el nombre de la instancia, en la siguiente columna, el resultado de la evaluación hecha del problema MINLP relajado, conocido como RMINLP. En la tercer columna se encuentra la parte de la solución al problema RMINLP referente al número de compresores encendidos en cada estación compresora n_{ij} (variable continua en el RMINLP). En la cuarta columna está el resultado obtenido de la evaluación de la función objetivo, cuando convertimos la variable n_{ij} en un parámetro al asignarle un valor de 1, de igual manera en la quinta columna está el resultado de la función objetivo, cuando $n_{ij} = 2$ y en la sexta columna esta el resultado de la evaluación de la función objetivo cuando $n_{ij} = 3$. Finalmente en la séptima y octava columna estan los resultados de la evaluación de la función objetivo cuando se resuelve el problema MINLP y la parte de la solución referente al número de compresores trabajando en cada estación compresora, respectivamente.

Las instancias net-b-11c4-2 y net-c-13c5-6 no lograron ser solucionadas ya que la solución del problema MINLP de esas instancias de acuerdo a la Tabla 4.15 no es la misma cantidad de unidades en todas las estaciones compresoras.

Ahora bien, de las instancias que sí pudieron ser resueltas observemos que en la mayoría de ellas, la solución de los problemas RMINLP son las mínimas, o iguales a las obtenidas en los MINLP (tal es el caso de las instancias net-a-5c1-3 y net-a-5c1-5) y en los NLP con los valores fijos de n_{ij} (como en net-a-5c1-1-1). Esto ocurre porque una relajación del

problema original, usualmente marca una cota inferior⁴, pues tiene menos restricciones que cumplir. Implementar las soluciones de los problemas relajados sería entonces lo ideal, pero no podemos encender 4.3 compresores, ni 3.2 compresores.

Pongamos atención a los problemas NLP con n_{ij} fijo en los valores 2 y 3 y en los problemas MINLP. En todos los casos la menor evaluación de la función objetivo de los problemas NLP es siempre mayor o igual que la solución ofrecida por el MINLP, es decir, la respuesta de los problemas MINLP es mejor (la evaluación numérica es menor) que las respuestas ofrecidas cuando se fija el valor del número de compresores y n_{ij} se trata como parámetro del problema.

Instancia	RMINLP		NLP-2			NLP-3			MINLP		
	T	NI	T	NI	IOR	T	NI	IOR	T	NI	IOR
net-a-5c1-1	0.026	14	0.018	22	0.21	0.016	16	0	0.508	236	0
net-a-5c1-3	0.018	28	0.016	14	0	0.026	14	0	0.366	200	0
net-a-5c1-5	0.027	17	0.028	20	0	0.030	21	0	0.430	206	0
net-b-10c3-1	0.034	22	0.047	44	0.10	0.025	35	0.48	3.876	1011	0.08
net-b-11c4-2	0.041	31	0.043	25	0.49				2.688	887	0.02
net-c-13c5-6	0.041	38	0.052	35	0.46				2.044	806	0.15

Tabla 4.16: Comparación de IOR, T y NI de los programas RMINLP, NLP-f y los MINLP.

De la Tabla 4.16 se observa que el IOR en todos los casos fue menor para las instancias resueltas como MINLP que el de las demás instancias probadas, es decir estamos muy cerca de la cota inferior.

En cuanto al número de iteraciones y recursos computacionales usados, resolver un problema MINLP, en general, siempre requerirá más tiempo, debido a que en cada iteración mayor hay dos iteraciones menores, cada una resuelve una parte del problema que a su vez requiere resolverse muchas veces. En cambio resolver un problema NLP, con n_{ij} como parámetro del problema, no requiere tanto esfuerzo, porque tiene una variable menos que encontrar, y el problema RMINLP también requiere menor esfuerzo, debido a que la variable n_{ij} no es entera, sino continua, lo cual hace menos complicado el problema.

⁴Lo cual es completamente cierto cuando el problema es convexo, y el nuestro no lo es.

Del status final del optimizador, en los problemas RMINLP y NLP con $n_{ij} \in 1, 2, 3$, las instancias que fueron resueltas tienen status = 2 (óptimo local), pero para los problemas MINLP, el status fue 8 (solución factible). Aún así, esta solución factible es mejor que el óptimo local del NLP.

4.3.3 Conclusiones

Como conclusión del experimento B, diremos que resolver el problema que planteamos en esta tesis como un MINLP, es mejor que resolverlo como un problema NLP con n_{ij} tratado como parámetro donde será asignado en primera instancia que valor deberá asumir, pues como vimos en la Tabla 4.15 la evaluación de la función objetivo es igual o menor en todas las instancias analizadas.

CAPÍTULO 5

CONCLUSIONES

En este Capítulo, daremos a conocer las conclusiones finales de nuestro trabajo de tesis. Primeramente se resumen las conclusiones obtenidas de los experimentos realizados. Posteriormente enumeramos las aportaciones científicas de nuestro trabajo de investigación. Finalmente, emitimos algunas recomendaciones para trabajos posteriores.

5.1 Conclusiones Finales

El objeto de estudio en esta investigación es un problema en redes de tuberías que transportan gas natural. Este problema de optimización fue modelado como un programa no lineal entero mixto, debido a las características de no linealidad de la función objetivo y restricciones y a la naturaleza continua y discreta de sus variables de decisión. El objetivo es resolver este problema minimizando los costos en que se incurre al transportar el combustible. Como hemos mencionado, para que el el combustible se mantenga en movimiento, se instalan a lo largo de las red estaciones compresoras que consumen alrededor del 3% al 5% del gas natural transportado, lo cual representa una cantidad gigantesca si consideramos que las cantidades transportadas son miles de millones de pies cúbicos diarios.

Debido a las características de nuestro problema, se decidió emplear el algoritmo OA/ER/AP, que a diferencia de otros métodos puede trabajar con problemas no convexos.

En este trabajo de tesis principalmente buscamos cumplir cuatro objetivos:

- Entendimiento de la estructura matemática del problema que nos permita emplear adecuadamente las técnicas de solución.
- Efectuar una evaluación algorítmica de los parámetros del algoritmo en estudio.

- Obtener soluciones (factibles y óptimas locales) a un número importante de instancias del problema.
- Mostrar la superioridad en la solución de un modelo MINLP sobre uno NLP.

Primeramente resolvimos las instancias creadas cuando construimos nuestra base de datos sin efectuar cambios a los parámetros algorítmicos. El resultado fue que no muchas de las instancias pudieron ser resueltas, y de las que sí pudieron solucionarse, empleaban demasiados recursos computacionales y necesitaban mucho tiempo para lograr obtener una solución.

Posteriormente, efectuamos la evaluación algorítmica de los parámetros del método OA/ER/AP, directamente modificando opciones de DICOPT [15].

De los experimento descritos y realizados para determinar las mejores condiciones y parámetros algorítmicos para resolver nuestro problema MINLP de minimización de combustible en una red de gasoductos, podemos concluir que a pesar de las dificultades y retos que presenta este problema, debidos principalmente a la no convexidad y no linealidad de la función objetivo y algunas restricciones, además de las dificultades numéricas para resolver los tres tipos de topologías de red en que se presenta este problema, pudimos resolver la mayoría de los mismos (75% de las instancias de la base de datos propuesta) cuando realizamos la evaluación algorítmica del método OA/ER/AP. Aquí podemos señalar que el auto-preprocesamiento, el criterio de parada $s2$, la penalidad aumentada con valor de 1000 y un punto inicial para la variable g_{ij}^1 de entre el 1%, 5% y 50% del valor de la cota inferior de ella misma, nos ayudan a obtener los mejores resultados en un tiempo razonable. También subrayamos que los valores mencionados anteriormente, están avalados mediante pruebas de hipótesis con estadística no paramétrica.

Finalmente, de la comparación hecha de algunas instancias que fueron resueltas como problemas MINLP, contra las soluciones encontradas cuando estas instancias eran tratadas como programas no lineales (prefijando la variable discreta n_{ij}), podemos concluir que las ventajas que ofrece el resolver las instancias como MINLP es que puede proporcionar mejores soluciones en cuanto al valor de la evaluación de la función objetivo (minimizar el consumo de combustible en las estaciones compresoras a lo largo de toda la red), y que aunque el número de iteraciones y el tiempo requerido para solucionar estos problemas es mayor, no aumenta demasiado (todas las instancias fueron resueltas en menos de 3 minutos).

5.2 Aportaciones Científicas

Entre las aportaciones de este trabajo se encuentran:

- La ampliación de la base existente, pues en trabajos anteriores, solo se estudiaba 1 topología de cada tipo de red [39, 46], y aquí empleamos en total 11 instancias de red (3 del tipo A, 3 del tipo B y 5 del tipo C).
- La implementación computacional de códigos empleando Matlab[24], GAMS[5], con el objeto de:
 - Realizar las gráficas del dominio factible del compresor, fijando el valor de una variable de decisión (flujo e la estación compresora), en uno y tres niveles.
 - Implementación de las instancias para solucionar las mismas y para posteriormente realizar la evaluación algorítmica del método y la comparación con otras formas de solucionar el problema de nuestro estudio. Una muestra de los programas hechos en GAMS puede verse en al Apéndice A.
- El ofrecer una prometedora alternativa para resolver el problema en cuestión, sin las suposiciones de que en la estación compresora trabajan siempre un mismo número fijo de compresores, en todas las estaciones de la red.
- Mostrar empíricamente la valía de usar un modelo MINLP sobre un NLP. Al resolver el problema MINLP obtenemos buenos resultados (óptimos locales y soluciones factibles) en un tiempo razonable y tratando a todas las variables al mismo nivel, sin necesidad de resolver el problema en dos etapas. Este tipo de modelación, hasta donde tenemos conocimiento, no se había realizado.

Los resultados de este trabajo han sido publicados en:

1. D. Cobos-Zaleta y R. Z. Ríos-Mercado. A MINLP model for minimizing fuel consumption on natural gas pipeline networks. En *Proceedings of the XI Latin-Ibero-American Conference on Operations Research*, Concepción, Chile, Octubre 2002.

2. Y. Villalobos-Morales, D. Cobos-Zaleta, H. J. Flores-Villarreal, C. Borraz-Sánchez y R. Z. Ríos-Mercado. On NLP y MINLP formulations and preprocessing for fuel cost minimization of natural gas transmission networks, En *Proceedings of the 2003 NSF Conference*, Birmingham, Alabama, EUA, Enero 2003.

y presentados en:

1. D. Cobos-Zaleta. Aplicación de modelos mixtos no lineales en la industria del gas natural. Ciclo de seminarios de PISIS, FIME, UANL, Mayo 2002.
2. D. Cobos-Zaleta y R. Z. Ríos-Mercado. Un modelo MINLP para un problema de minimización del consumo de combustible en redes de gas natural. XXXV Congreso Nacional de la Sociedad Matemática Mexicana, Durango, Durango, Octubre 2002.
3. D. Cobos-Zaleta. Resolviendo un problema de minimización de consumo de combustible en redes de gas natural con variables continuas y discretas. Ciclo de seminarios de PISIS, FIME, UANL, Noviembre 2002.
4. D. Cobos-Zaleta y R. Z. Ríos-Mercado. Computational experience with an outer approximation with equality relaxation algorithm for solving a least cost natural gas transmission network problem. 8th INFORMS Computing Society Conference, Phoenix, EUA, Enero 2003.
5. D. Cobos-Zaleta y R. Z. Ríos-Mercado. Minimización de combustible en redes de gas natural en un modelo de programación no lineal entero mixto, Escuela Nacional de Optimización y Análisis Numérico, Saltillo, Coahuila, Marzo 2003.
6. D. Cobos-Zaleta y R. Z. Ríos-Mercado. Resolviendo un problema en la industria del gas natural usando investigación de operaciones. I Simposio de FIME de Investigación de Operaciones, FIME, UANL, Octubre 2003.

5.3 Recomendaciones para Trabajos Posteriores

Así como en este trabajo se incrementa el número de instancias para realizar las evaluaciones algorítmicas, una recomendación es seguir incrementando esta base de datos,

para que finalmente, los métodos con que se propone solucionar este importante problema, sea instancias de una diversidad mayor, las cuales por supuesto, serán mucho más grandes.

Como vimos en el desarrollo de esta tesis, al ser este problema no convexo, el algoritmo OA/ER/AP no puede garantizar que la solución obtenida sea una solución óptima global, sin embargo, se puede brincar esta barrera empleando otras técnicas de solución, como lo son las heurísticas y metaheurísticas que sirven para reportar buenas soluciones y pueden lidiar con la no convexidad. Ejemplos de estas son las técnicas de búsqueda local como el GRASP [7] y la Búsqueda Tabú [14].

Asimismo, este problema no ha sido tratado desde la perspectiva de optimización global [17], la cual es una rama que estudia técnicas enumerativas de solución para encontrar óptimos globales en problemas no convexos.

El tema tratado en este trabajo de investigación es de gran importancia. Se espera que los resultados obtenidos en esta investigación sean de gran utilidad para trabajos posteriores que aborden este trascendente problema en la

BIBLIOGRAFÍA

- [1] D. Bertsekas, G. Lauer, N. Sandell, and T. Posbergh. Optimal short term scheduling of large-scale power systems. *IEEE Transactions on Automatic Control*, 28(1):1–11, 1983.
- [2] D. B. Birewar and I. E. Grossman. Incorporating scheduling in the optimal design of multiproduct batch plants. *Computers and Chemical Engineering*, 1(2):141–161, 1989.
- [3] J. A. Bloom. Solving an electricity generating capacity expansion planning problem by generalized Bender's decomposition. *Operations Research*, 31(5):84–100, 1983.
- [4] B. Borchers and J. E. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programming. RPI Math Report No. 200, Rensselaer Polytechnic Institute, 1991.
- [5] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS: A User's Guide, Release 2.25*. The Scientific Press, San Francisco, EUA, 1992.
- [6] R. G. Carter. Pipeline optimization: Dynamic programming after 30 years. In *Proceedings of the 30th PSIG Annual Meeting*, Denver, EUA, October 1998.
- [7] A. Díaz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato, and F. T. Tseng. *Optimización Heurística y Redes Neuronales*. Paraninfo, Madrid, España, 1996.
- [8] M. A. Duran and I. E. Grossman. An outer approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
- [9] H. J. Flores-Villarreal and R. Z. Ríos-Mercado. Computational experience with a GRG method for minimizing fuel consumption on cyclic natural gas networks. In N. E. Mastorakis, I. A. Stathopoulos, C. Manikopoulos, G. E. Antoniou, V. M. Mladenov, and I. F. Gonos, editors, *Computational Methods in Circuits and Systems Applications*, pages 90–94. WSEAS Press, Athens, 2003.
- [10] H. J. Flores-Villarreal and R. Z. Ríos-Mercado. Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions. In *Proceedings of the International Applied Business Research Conference*, Acapulco, March 2003.

- [11] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization Fundamentals and Applications*. Oxford University Press, New York, EUA, 1995.
- [12] B. Gavish, D. Horsky, and K. Srikanth. An approach to optimal positioning of a new product. *Management Science*, 29(11):1277–1297, 1983.
- [13] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.
- [14] F. Glover and M. Laguna. *Tabu Search*. Kluwer, Boston, EUA, 1997.
- [15] I. E. Grossmann, J. Viswanathan, A. Vecchiotti, R. Raman, and E. Kalvelagen. *GAMS/DICOPT: A Discrete Continuous Optimization Peckage*. Carnegie Mellon University and GAMS Development Corporation, Washington, EUA, November 2001.
- [16] H. H. Hoang. Topological optimization of networks: A nonlinear mixed integer model employing generalizad Benders decomposition. *IEEE Transaction Automatic Control*, 27(1):164–169, 1982.
- [17] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.
- [18] G. E. Paules IV and C. A. Floudas. APROS: Algorithmic development methodology for discrete-continuous optimization problems. *Operations Research*, 37(6):902–915, 1989.
- [19] S. Kim, R. Z. Ríos-Mercado, and E. A. Boyd. A heuristic for minimum cost steady-state gas transmission networks. In *Proceedings of the 25th International Conference on Computers & Industrial Engineering*, New Orleans, EUA, March 1999.
- [20] S. Kim, R. Z. Ríos-Mercado, and E. A. Boyd. Heuristics for minimum cost steady-state gas transmission networks. In M. Laguna and J. L. González-Velarde, editors, *Computing Tools for Modeling, Optimization, and Simulation*, chapter 11, pages 203–213. Kluwer, Boston, EUA, 2000.
- [21] G. R. Kocis and I. E. Grossman. Relaxation strategy for the structural optimization of process flow sheets. *Industrial Engineering Chemical Research*, 26(9):1869–1880, 1987.

- [22] H. S. Lall and P. B. Percell. A dynamic programming based gas pipeline optimizer. In A. Bensoussan and J. L. Lions, editors, *Analysis and Optimization of Systems*, volume 144 of *Lecture Notes in Control and Information Sciences*, pages 123–132, Berlin, 1990. Springer-Verlag.
- [23] C. A. Luongo, B. J. Gilmour, and D. W. Schroeder. Optimization in natural gas transmission networks: A tool to improve operational efficiency. Presented at the 3rd SIAM Conference on Optimization, Boston, April 1989.
- [24] The MathWorks, Inc, Natick, Massachusetts, EUA. *MATLAB: Using Matlab Graphics*, 2000.
- [25] H. Mawengkang and B. A. Murtagh. Solving nonlinear integer programs with large scale optimization software. *Annals of Operation Research*, 5:425–437, 1986.
- [26] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.
- [27] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, EUA, 1988.
- [28] O. Odele and S. Macchietto. Computer aideed molecular design: A novel method for optimal solvent selection. *Fluid Phase Equilibria*, 82:39–46, 1993.
- [29] A. J. Osiadacz. *Simulation and Analysis of Gas Networks*. Gulf Publishing Company, Houston, EUA, 1987.
- [30] A. J. Osiadacz. Dynamic optimization of high pressure gas networks using hierarchical systems theory. In *Proceedings of the 26th PSIG Annual Meeting*, San Diego, EUA, October, 1994.
- [31] A. J. Osiadacz and M. Górecki. Optimization of pipe sizes for distribution gas network design. In *Proceedings of the 27th PSIG Annual Meeting*, Albuquerque, EUA, October 1995.
- [32] A. J. Osiadacz and S. Swierczewski. Optimal control of gas transportation systems. In *Proceedings of the 3rd IEEE Conference on Control Applications*, pages 795–796, August 1994.

- [33] P. M. Pardalos and G. Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, 7(1):33–35, 1988.
- [34] P. B. Percell and M. J. Ryan. Steady-state optimization of gas pipeline network operation. In *Proceedings of the 19th PSIG Annual Meeting*, Tulsa, EUA, October 1987.
- [35] G. V. Reklaitis. Perspectives on scheduling and planning process operations. In *Proceedings of the 4th International Symposium on Process Systems Engineering*, page III.2.1, Montreal, Canada, 1991.
- [36] R. Z. Ríos-Mercado. Natural gas pipeline optimization. In P. M. Pardalos and M. G. C. Resende, editors, *Handbook of Applied Optimization*. Oxford University Press, New York, EUA, 2001.
- [37] D. W. Schroeder. Hydraulic analysis in the natural gas industry. In J. J.-W. Chen and A. Mital, editors, *Advances in Industrial Engineering Applications and Practice I*, pages 960–965, Cincinnati, EUA, 1996. *International Journal of Industrial Engineering*.
- [38] J. Vaselenak, I. E. Grossman, and A. W. Westerberg. Optimal retrofit design of multipurpose batch plants. *Industrial Engineering Chemical Research*, 26(4):718–726, 1987.
- [39] Y. Villalobos. Pre-Procesamiento de un Problema de Optimización de Redes de Gas Natural. Master degree dissertation, FIME, UANL, San Nicolás de los Garza, NL, México, October 2002.
- [40] Y. Villalobos. Preprocesamiento efectivo de un problema de minimización de combustible en sistemas de transporte de gas natural. Technical Report PISIS-2002-01, FIME, UANL, San Nicolás de los Garza, NL, México, November 2002.
- [41] J. Viswanathan and I. E. Grossmann. A combined penalty function and outer approximation method for MINLP optimization. *Computers and Chemical Engineering*, 14(7):769–782, 1990.
- [42] R. E. Walpole and R. H. Myers. *Probabilidad y Estadística*. McGraw–Hill, México, México, 4ta edition, 2000.

- [43] P. J. Wong and R. E. Larson. Optimization of natural-gas pipeline systems via dynamic programming. *IEEE Transactions on Automatic Control*, AC-13(5):475–481, 1968.
- [44] S. Wu. *Steady-State Simulation and Fuel Cost Minimization of Gas Pipeline Networks*. Doctoral dissertation, University of Houston, Houston, EUA, August 1998.
- [45] S. Wu, E. A. Boyd, and L. R. Scott. Minimizing fuel consumption at gas compressor stations. In J. J.-W. Chen and A. Mital, editors, *Advances in Industrial Engineering Applications and Practice I*, pages 972–977, Cincinnati, EUA, 1996. *International Journal of Industrial Engineering*.
- [46] S. Wu, R. Z. Ríos-Mercado, E. A. Boyd, and L. R. Scott. Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modelling*, 31(2–3):197–220, 2000.

LISTA DE TABLAS

4.1	Relación del número de compresor y su nombre comercial.	30
4.2	Listado de todas la topologías de nuestra base de datos.	30
4.3	Evaluación del pre-pocesamiento. (+) $1 \times 10 + 03$	36
4.4	Instancias sin solución al emplear ó no el pre-procesamiento.	37
4.5	Resultados del status final, utilizando y sin usar el pre-pocesamiento.	38
4.6	Evaluación del criterio de parada ($s1=stop\ 0$ y $s2=stop\ 2$). (*) $1 \times 10 + 06$, (+) $1 \times 10 + 03$	39
4.7	Instancias donde el optimizador falló usando $s1$ y $s2$	40
4.8	Status final del modelo usando $s1$ y $s2$	40
4.9	Evaluación del parámetro p (penalidad aumentada). (+) $1 \times 10 + 03$	42
4.10	Instancias que no lograron ser resueltas por el optimizador, al variar el valor de p	42
4.11	Evaluación de p de acuerdo al status final del optimizador.	43
4.12	Evaluación del parámetro punto inicial de g_{ij}^1 . (+) $1 \times 10 + 03$	44
4.13	Instancias con soluciones infactibles al evaluar el punto inicial.	45
4.14	Resultados del status final cuando se modifica el valor del punto inicial de g_{ij}^1	46
4.15	Evaluación de los programas RMINLP, NLP-f y los MINLP. (*) $1 \times 10 + 06$	51
4.16	Comparación de IOR, T y NI de los programas RMINLP, NLP-f y los MINLP.	52

LISTA DE FIGURAS

1.1	Ejemplo de red con topología sencilla ó gunbarrel	4
1.2	Ejemplo de red con topología con ramas ó de árbol	4
1.3	Ejemplo de red con topología con ciclos	5
2.4	Estación Compresora configurada en paralelo	10
2.5	Gráfica del dominio de un compresor con la variable de flujo fija.	14
2.6	Gráfica del dominio de un compresor con variable flujo fija en tres niveles. .	15
2.7	Gráfica del dominio de un compresor con la variable flujo fija.	16

APÉNDICE A: GAMS/DICOPT

Muestra de Formulación usando DICOPT

La formulación del problema se guarda en un archivo texto con extensión *.gms (*nombre.gms*).

Este archivo contiene toda la información del problema, como tipo de problema, modelo matemático, información del optimizador, los parámetros y las variables del mismo. Asimismo, el programa optimizador puede interactuar con el usuario mediante el archivo dicop.opt, donde pueden ser cambiadas las opciones del optimizador.

GAMS ejecuta el archivo, crea el modelo, llama al optimizador y reporta los resultados en un archivo con extensión *.lst (*nombre.lst*).

A continuación presentamos un ejemplo del problema en una topología de red sencilla, con cinco nodos y un compresor centrífugo del tipo 1 (*net-a-5c1-1.gms*).

archivo: net-a-5c1-1.gms

```
$Title Optimizacion inteligente de redes de Transporte de gas natural
```

```
$Inlinecom{ }
```

```
Sets
```

```
i nodos en la red / 1, 2, 3, 4, 5 /
```

```
ir(i) nodos oferta /1/
```

```
id(i) nodos demanda /4, 5/
```

```
k unidad compresora /u1*u10/
```

```
prop propiedades del ducto /longitud, diametro, friccion, capacidad/
```

```
alias(i,j);
```

```
Sets
```

```
A(i,j) arcos en la red /1.2, 2.3, 3.4, 3.5/
```

```
Ac(i,j) arcos compresores en la red /2.3/
```

```
Ap(i,j) arcos ductos en la red /1.2, 3.4, 3.5/;
```

```
Scalar {Estos escalares son usados para calcular las ecuaciones que ligan a las variables  
originales (H,Q,S) con (x_ij, p_i, p_j)}
```

{Propiedades del gas}

gas_K	133050	constante del ducto
gas_e	1.287	exponente isoentropico
gas_R	85.2	constante del gas en (lbf-ft/lbm-°R)
gas_Z	0.95	factor de compresibilidad del gas
gas_Sg	0.6248	gravedad especifica
gas_T	519.67	temperatura promedio en la succion (°R)
fac	33.1944	factor de conversion de MMCS D a (lbm-min)

Table datducto(i,j,prop) {Propiedades del ducto: longitud(mi), diametro(ft), friccion}

	longitud	diametro	friccion	capacidad
1.2	50	3	0.0085	1200
3.4	50	3	0.0085	1200
3.5	50	3	0.0085	1200;

Parameter

qu(i,j)	limite superior del flujo volumetrico del compresor (i,j) unidades (ft ³ /min)
ql(i,j)	limite inferior del flujo volumetrico del compresor (i,j) unidades (ft ³ /min)
su(i,j)	limite superior de velocidad del compresor (i,j) unidades (ft/min)
sl(i,j)	limite inferior de velocidad del compresor (i,j) unidades (ft/min)
AH(i,j)	Coficiente para la ecuacion del Dominio de operacion de la E C (i,j)
BH(i,j)	Coficiente para la ecuacion del Dominio de operacion de la E C (i,j)
CH(i,j)	Coficiente para la ecuacion del Dominio de operacion de la E C (i,j)
DH(i,j)	Coficiente para la ecuacion del Dominio de operacion de la E C (i,j)
AE(i,j)	Coficiente para la ecuacion de eta del compresor (i,j)
BE(i,j)	Coficiente para la ecuacion de eta del compresor (i,j)
CE(i,j)	Coficiente para la ecuacion de eta del compresor (i,j)
DE(i,j)	Coficiente para la ecuacion de eta del compresor (i,j)
pu(i)	limite superior de presion en el nodo i (psia)
pl(i)	limite inferior de presion en el nodo i (psia)
rl(i,j)	valor minimo del cociente Q/S en la estacion (i,j)
ru(i,j)	valor maximo del cociente Q/S en la estacion (i,j)
diam(i,j)	conversion del diametro del ducto (i,j) de ft a pulgadas

$cf(i,j)$	conversion de la capacidad del flujo del gas en el ducto (i,j) en MMSCFD \rightarrow lbm/min
C	constante del gas
$c1(i,j)$	resistencia del ducto (i,j) con unidades de $psia^2/MMSCFD^2$
$r(i,j)$	convercion la resistencia del ducto (i,j) de $psia^2/MMSCFD^2 \rightarrow$ $psia^2/lbm/min^2$
gas_m	proporcion del radio especifico
gase	propiedades del gas
PHL_SURGE	ecuacion para calcular el hmin
PHL_STONE	ecuacion para calcular el hmax
$hu(i,j)$	limite superior de cabeza adiabatica del compresor (i,j) unidades (lbf-ft/lbm)
$hl(i,j)$	limite inferior de cabeza adiabatica del compresor (i,j) unidades (lbf-ft/lbm);
*compresor uno = cdbnk1	
$ql(i,j) = 4200;$	
$qu(i,j) = 11100;$	
$sl(i,j) = 4200;$	
$su(i,j) = 6300;$	
$AH(i,j) = 0.00022289;$	
$BH(i,j) = 0.00026112;$	
$CH(i,j) = -0.00013082;$	
$DH(i,j) = -0.00000504;$	
$AE(i,j) = 81.09393384;$	
$BE(i,j) = -70.28212702;$	
$CE(i,j) = 106.50361597;$	
$DE(i,j) = -39.93984952;$	
$pu(i) = 800;$	
$pl(i) = 200;$	
$rl(i,j) = ql(i,j)/sl(i,j);$	
$ru(i,j) = qu(i,j)/su(i,j);$	

```

diam(i,j) = datducto(i,j,'diametro')*12;
cff(i,j)$Ap(i,j) = datducto(i,j,'capacidad')*fac;
C = gas_K*gas_Z*gas_T*gas_Sg;
c1(i,j)$Ap(i,j) = C*((datducto(i,j,'longitud')*datducto(i,j,'friccion'))/(diam(i,j)**5));
r(i,j) = c1(i,j)/(fac**2);
gas_m = (gas_e - 1)/gas_e;
gase = (gas_Z*gas_R*gas_T)/gas_m;
PHL_SURGE(i,j)$Ac(i,j) = AH(i,j) + BH(i,j)*rl(i,j) + CH(i,j)*(rl(i,j)**2) + DH(i,j)
*(rl(i,j)**3);
PHL_STONE(i,j)$Ac(i,j) = AH(i,j) + BH(i,j)*ru(i,j) + CH(i,j)*(ru(i,j)**2) + DH(i,j)
*(ru(i,j)**3);
hl(i,j) = (sl(i,j)**2)*PHL_STONE(i,j);
hu(i,j) = (su(i,j)**2)*PHL_SURGE(i,j);

```

Parameter

```

o(ir) {Cantidad de flujo de los nodos oferta (MMSCFD)}
/1 1000/
d(id) {Cantidad de flujo de los nodos demanda (MMSCFD)}
/4 500
5 500/;

```

{Conversion de unidades: MMSCFD → lbm/min}

```

o(ir) = o(ir)*fac;
d(id) = d(id)*fac;

```

Parameter

```

b(i); {rhs de las ecuaciones de balance del flujo de masa (lbm/min)};
b(i) = 0; {Inicializamos b(i)}
b(ir) = o(ir); {Conjunto de b(i) para los nodos oferta}
b(id) = -d(id); {Conjunto de b(i) para los nodos demanda}

```


$\text{display } r_l, r_u, \text{gas_m, diam, C, gase, cf, c1, r, b, hl, hu};$

Variables

Y {Funcion Objetivo Min la sum de los costos del combustible en todas las estaciones}

$x(i,j)$ {Flujo de masa en cada arco (i,j) en la red unidades(lbm/min)}

$p(i)$ {Presion en el nodo i unidades (lbf/in²)}

$s(i,j)$ {Velocidad del flujo en la estacion (i,j) unidades (rpm)}

$q(i,j)$ {Flujo volumetrico en la estacion (i,j) unidades (ft³/min)}

$h(i,j)$ {Cabeza adiabatica en la estacion (i,j) unidades (lbf*ft/lbm)}

$g(i,j)$ {Costo de combustible en la estacion (i,j) unidades (lbf*ft/min)}

$\text{eta}(i,j)$ {Eta eficiencia en la estacion (i,j) }

$\text{ra}(i,j)$ {razon del cociente Q/S}

$\text{nx}(i,j,k)$ {indica que compresores estan encendidos al valer 1}

$\text{ncomp}(i,j)$ {numeros de compresores que funcionan en la estacion compresora};

Positive Variable $x,p,s,q,h,g,\text{eta},\text{ra},\text{ncomp};$

binary variable $\text{nx};$

Equations

costtotal {Function Objetivo}

$\text{cost}(i,j)$ {Costo en la estacion (i,j) }

$\text{balance}(i)$ {Balance de flujo en cada nodo i }

$\text{flujogas}(i,j)$ {Ecuacion de flujo de gas en cada ducto (i,j) }

$\text{cabeza}(i,j)$ {Cabeza adiabatica H en la estacion (i,j) }

$\text{auxiliar}(i,j)$

$\text{flvolum}(i,j)$ {Flujo volumetrico Q en cada estacion (i,j) }

$\text{dominio}(i,j)$ {Dominio de la estacion compresora (i,j) }

$\text{eficiencia}(i,j)$ {Eficiencia de la estacion compresora (i,j) }

$\text{razon}(i,j)$ {Razon del cociente Q/S}

$\text{presion}(i,j)$ {restriccion de la presion de succion y la presion de descarga};

```

costtotal.. Y =e= sum((i,j)$Ac(i,j), g(i,j));
cost(i,j)$Ac(i,j).. g(i,j)*eta(i,j) =e= 1 * (x(i,j)*h(i,j));
balance(i).. sum(j$A(i,j), x(i,j)) - sum(j$A(j,i), x(j,i)) =e= b(i);
auxiliar(i,j)$Ac(i,j).. ncomp(i,j) =e= sum(k, nx(i,j,k) );
flujogas(i,j)$Ap(i,j).. (p(i)**2 - p(j)**2) =e= r(i,j) * ((x(i,j))**2);
cabeza(i,j)$Ac(i,j).. p(j)**gas_m =e= p(i)**gas_m*((1/gase)*(h(i,j))+1);
flvolum(i,j)$Ac(i,j).. ncomp(i,j)*144*p(i)*q(i,j) =e= (gas_Z*gas_R*gas_T)*x(i,j);
dominio(i,j)$Ac(i,j).. h(i,j) =e= (AH(i,j) + BH(i,j)*ra(i,j) + CH(i,j)*(ra(i,j)**2) + DH(i,j)
*(ra(i,j)**3)) * (s(i,j)**2);
eficiencia(i,j)$Ac(i,j).. eta(i,j) =e= AE(i,j) + BE(i,j)*ra(i,j) + CE(i,j)*(ra(i,j)**2) + DE(i,j)
*(ra(i,j)**3);
razon(i,j)$Ac(i,j).. s(i,j)*ra(i,j) =e= q(i,j);
presion(i,j)$Ac(i,j).. p(i) =e= p(j);

```

Model mincosto /all /;

{ Los limites de presion dados en psia (lbf/in 2) }

p.up(i) = pu(i);

p.lo(i) = pl(i);

p.l(i) = (p.up(i) + p.lo(i)) / 2.0 ;

{ Limites de operacion en cada compresor (ft 3/min) }

q.fx(i,j) = 0;

q.up(i,j)\$Ac(i,j) = qu(i,j);

q.lo(i,j)\$Ac(i,j) = ql(i,j);

q.l(i,j)\$Ac(i,j) = (q.up(i,j) + q.lo(i,j)) / 2.0 ;

{ Limites de unidades compresoras (ft 3/min) }

ncomp.fx(i,j) = 0;

ncomp.up(i,j)\$Ac(i,j) = 10;

ncomp.lo(i,j)\$Ac(i,j) = 1;

ncomp.l(i,j)\$Ac(i,j) = 5 ;

{ Limites de operacion en cada compresor (ft 3/min) }

$$h.up(i,j) = hu(i,j);$$

$$h.lo(i,j) = hl(i,j);$$

$$h.l(i,j) = (h.up(i,j) + h.lo(i,j)) / 2.0 ;$$

{ Limites de la razon del flujo volumetrico y velocidad }

$$ra.fx(i,j) = 0;$$

$$ra.up(i,j) \$Ac(i,j) = ru(i,j);$$

$$ra.lo(i,j) \$Ac(i,j) = rl(i,j);$$

$$ra.l(i,j) \$Ac(i,j) = (ra.lo(i,j) + ra.up(i,j)) / 2.0 ;$$

{ Capacidad de flujo de gas en el ducto(i,j) y en la estacion (i,j) (lbm/min) }

$$x.up(i,j) \$Ap(i,j) = cf(i,j);$$

$$x.l(i,j) \$Ap(i,j) = (x.lo(i,j) + x.up(i,j)) / 2.0 ;$$

$$x.up(i,j) \$Ac(i,j) = (ncomp.up(i,j) * qu(i,j) * p.up(i) * 144) / (gas.Z * gas.R * gas.T);$$

$$x.lo(i,j) \$Ac(i,j) = (ql(i,j) * p.lo(i) * 144) / (gas.Z * gas.R * gas.T);$$

$$x.l(i,j) \$Ac(i,j) = (x.lo(i,j) + x.up(i,j)) / 2.0;$$

{ Limites de velocidad en cada compresor (rpm) }

$$s.fx(i,j) = 0;$$

$$s.up(i,j) \$Ac(i,j) = su(i,j);$$

$$s.lo(i,j) \$Ac(i,j) = sl(i,j);$$

$$s.l(i,j) \$Ac(i,j) = (s.lo(i,j) + s.up(i,j)) / 2.0 ;$$

{ Eficiencia del compresor (0-100 %) }

$$eta.fx(i,j) = 0;$$

$$eta.up(i,j) \$Ac(i,j) = 100;$$

$$eta.lo(i,j) \$Ac(i,j) = 50;$$

$$eta.l(i,j) \$Ac(i,j) = (eta.lo(i,j) + eta.up(i,j)) / 2.0 ;$$

{ Costo de combustible del compresor(i,j) }

$$g.fx(i,j) = 0;$$

$$g.up(i,j) \$Ac(i,j) = (x.up(i,j) * h.up(i,j)) / eta.lo(i,j);$$

$$g.lo(i,j) \$Ac(i,j) = (x.lo(i,j) * h.lo(i,j)) / eta.up(i,j);$$

$$g.l(i,j) \$Ac(i,j) = (g.lo(i,j) + g.up(i,j)) / 2.0;$$

```

Option nlp = CONOPT2;
Option mip = CPLEX;
Option rminlp = CONOPT2;
Option minlp = DICOPT;
Option limrow = 50;
mincosto.scaleopt = 1;
* eficiencia.scale(i,j)$Ac(i,j) = 1.0e+01;
* dominio.scale(i,j)$Ac(i,j) = 1.0e01;
flvolum.scale(i,j)$Ac(i,j) = 1.0e06;

```

Solve mincosto using rminlp minimizing Y;

Display x.l, p.l, q.l, ncomp.l;

```
mincosto.optfile = 1;
```

Solve mincosto using minlp minimizing Y;

Display x.l, x.m, p.l, s.l, q.l, h.l, g.l, eta.l, ra.l, ncomp.l;

Nota: Todas las líneas que comienzan con asterisco son comentario.

Cambiando los Parámetros en DICOPT

Dentro de un programa, como el descrito anteriormente, se invoca el empleo de las técnicas de pre-procesamiento, que se puede hacer de dos formas:

1. Dentro del cuerpo del programa *instancia.gms*. Para ello, hay que tomar en cuenta que cada técnica se invoca de manera diferente, las cuales describimos a continuación.

- (a) Para mandar a llamar al *escalamiento*. Primero hay que agregar la instrucción: *modelo.scaleopt = 1*; y a continuación se le indica al optimizador, en que ecuación

y por que factor se debe escalar. Ejemplo: $var.scale = fac$; lo cual quiere decir: escala la variable var en un factor fac .

(b) Para acotar variables. Se tienen que definir la cota superior, la cota inferior o ambas, para cada variable que se desea acotar. Esto se hace con las siguientes instrucciones:

- Cota superior: $var.up = val$.
- Cota inferior: $var.lo = val$.

Donde var es la variable que esta siendo acotada, la extensión up proviene de upper, vocablo de inglés que indica superior y la extensión lo proviene de lower, vocablo de inglés que indica inferior.

(c) Asignación del valor inicial. Este es un valor que tomará la variable var y es el punto de partida del algoritmo (el valor inicial de la variable), se define con la instrucción: $var.l = val$. La extensión l proviene de *level* que traducido del inglés significa nivel.

Nota 1: val es el valor que tomará la variable. Este valor numérico (num) puede ser designado allí ($var = num$) o puede quedar expresado con operaciones indicadas que deban efectuarse para obtener dicho valor ($var = A + B$, donde A y B deben haber sido definidos anteriormente y $A + B = num$).

2. Mediante el encendido de la opción de pre-procesamiento automático. Esto se logra agregando la instrucción $set lsscal true$ dentro del archivo $conopt2.opt$.

Nota 2: El archivo $conopt2.opt$ debe quedar en la misma ruta que los programas $instancia.gms$, para que el optimizador agregue esta opción y pueda encender el pre-procesamiento automático.

Para indicar al optimizador el criterio de parada, se debe especificar dentro del archivo $dicopt.opt$. En este trabajo probamos dos diferentes reglas para la terminación del algoritmo AO/ER/AP, que denominamos:

1. **s1.** Este criterio de parada se indica agregando la opción $stop 0$,
2. **s2.** De manera similar, para decirle al optimizador que utilice otro criterio de parada, agregamos la instrucción $stop 2$.

En ambos casos, el archivo *conopt.opt* ya contenía la instrucción para usar el pre-procesamiento, y en el archivo *dicopt.opt* se debe incluir la opción *stop 0* ó *stop 2*.

Al igual que cuando agregamos el archivo *conopt2.opt*, el archivo *dicopt.opt*, debe agregarse en el mismo directorio donde se encuentran los programas *instancias.gms* para que el optimizador pueda agregar las opciones deseadas cuando ejecute el algoritmo de solución y cambie el parámetro del default al que nosotros queremos asignarle.

Para modificar el valor del parámetro penalidad aumentada (p de la formulación (3.15)), se debe modificar el archivo ya creado *dicopt.opt* para incluirle la indicación *weight = num*. Donde *num* es el valor que tomará dicho parámetro. El valor default es *1000*. En este experimento, se comparan, tres diferentes valores para este parámetro: el valor default, contra otros dos: 100 y 100,000.

Para indicar al optimizador que modifique el valor del punto inicial (el valor default es 0) de alguna variable (en este trabajo solo probamos la variable g_{ij}^1), se indica directamente en el cuerpo del programa, probamos cuatro diferentes valores iniciales primeramente el punto medio (50%), después alejados un 10% de la cota inferior, luego alejados un 5% de la cota inferior y finalmente, alejados el 1% del valor de la cota inferior. Esto se logra con las siguientes instrucciones:

- El valor inicial fue fijado en el punto medio del rango de las cotas ($var.l = var.lo + [(var.up - var.lo)/2]$, es decir, $var.l = var.lo + 0.5(var.up - var.lo)$).
- Al 10% de la cota inferior, con la instrucción $var.l = var.lo + 0.1(var.up - var.lo)$.
- Al 5% de la cota inferior, mediante la sentencia $var.l = var.lo + 0.05(var.up - var.lo)$.
- Al 1% de la cota inferior, con la línea $var.l = var.lo + 0.01(var.up - var.lo)$.

Creación y Ubicación de los Programas

Experimento A

Para efectuar las pruebas que miden el impacto de usar o no el pre-procesamiento (experimento 1), se crearon dos clases de programas.

- Los primeros, ubicados en la ruta: */u/diana/miogams/tes/claio/prop/prop/allpar/sin*. Los cuales sólo presentan técnicas de pre-procesamiento en el cuerpo del programa, como ya se explicó anteriormente.
- Y los situados en el directorio: */u/diana/miogams/tes/claio/prop/prop*. Los cuales además de usar las técnicas antes mencionadas, presentan la característica de mantener el auto-pre-procesamiento encendido, con las instrucciones que ya se explicaron anteriormente. Tener el auto-pre-procesamiento encendido significa que las técnicas de pre-procesamiento se encienden automáticamente cada 20 iteraciones.

La comparación entre los archivos con pre-procesamiento encendido contra los de pre-procesamiento apagado, se realizó en una hoja de cálculo, situada en el directorio: */u/diana/miogams/tes/claio/prop/prop/allpar/pruebas estadísticas.xls*.

Para llevar a cabo la evaluación computacional de los dos criterios de parada (experimento 2), se crearon dos clases de programas.

- Los que contiene en criterio de parada *stop 0*, contenidos en el directorio: */u/diana/miogams/tes/claio/prop/prop/allpar/s/s1*.
- Y los situados en el directorio: */u/diana/miogams/tes/claio/prop/prop/allpar/s/s2*. Con la opción *stop 2* = criterio de parada número 2.

La comparación entre los archivos del directorio *s1* contra los del directorio *s2*, se efectuó en una hoja de cálculo, situada en el directorio: */u/diana/miogams/tes/claio/prop/prop/allpar/s/pruebas estadísticas.xls*.

Para efectuar la evaluación del impacto del parámetro de penalidad aumentada (experimento 3), se crearon tres directorios, cada uno de los cuales contiene diferentes clases de programas.

- El primer directorio, ubicado en: */u/diana/miogams/tes/claio/prop/prop/allpar/pa/100*. Con *weight = 100*, para indicarle a DICOPT que el valor del parámetro penalidad aumentada en esos programas estará fijo en 100.
- El siguiente, ubicado en: */u/diana/miogams/tes/claio/prop/prop/allpar/pa/mil*. El cual el valor default del parámetro penalidad aumentada.

- Y el directorio de la ruta: */u/diana/miogams/tes/claio/prop/prop/pa/100mil*. Los programas ahí ubicados, tiene fijo el valor de penalidad aumentada en 100,000.

La comparación entre los resultados obtenidos de los programas de los tres directorios anteriormente mencionados, se realizó en la hoja de cálculo, que se localiza en: */u/diana/miogams/tes/claio/prop/prop/allpar/pa/pruebas_estadisticas_pa.xls*.

Para probar el impacto de variar el valor del punto inicial de la variable consumo total de combustible g_{ij}^1 , se crearon cuatro directorios con diferentes clases de programas, en base a observaciones hechas de estos resultados, decidimos mover el punto inicial a una menor distancia de la cota inferior y probamos tres niveles más: 10%, 5% y 1%.

- Primero al 50% de la cota inferior, es decir, el punto medio entre las cotas. Los programas están ubicados en la dirección: */u/diana/miogams/tes/claio/prop/prop/allpar/pi/50*.
- Al 10% de la cota inferior, los programas ubicados en la dirección: */u/diana/miogams/tes/claio/prop/prop/allpar/pi/10*.
- Al 5% de la cota inferior. Estos archivos se encuentran en: */u/diana/miogams/tes/claio/prop/prop/allpar/pi/5*.
- Al 1% de la cota inferior, con los programas agrupados en la ruta: */u/diana/miogams/tes/claio/prop/prop/allpar/pi/1*.

Para realizar la comparación de los resultados obtenidos de los cuatro diferentes puntos iniciales para la variable g_{ij} de la ecuación 2.1 está ubicada en la dirección: */u/diana/miogams/tes/claio/prop/prop/allpar/pi/pruin.xls*.

Experimento B

Para este experimento se cambia el modelo de un MINLP a un RMINLP, en el cual la variable entera (n_{ij}) se trata como una variable continua, por lo cual el programa es básicamente un NLP. Para ello, se crean programas *instanciar.gms*. Estos programas están ubicados en: */u/diana/miogams/tes/claio/prop/prop/allpar/fijando/RMINLP*

Después se fija el valor de la variable n_{ij} y se convierte en un parámetro del problema. Para ello, se crean programas llamados *instanciaf.gms*, donde f es el valor del n_{ij} y puede

tomar los valores: 1, 2 o 3. Los programas localizados en:

- */u/diana/miogams/tes/claio/prop/prop/allpar/pi/pruin.xls*
- */u/diana/miogams/tes/claio/prop/prop/allpar/fijando/1,*
- */u/diana/miogams/tes/claio/prop/prop/allpar/fijando/2, y*
- */u/diana/miogams/tes/claio/prop/prop/allpar/fijando/3.*

Finalmente, resolveremos los problemas cuando el problema es un MINLP como ya describimos anteriormente. Están en la ruta */u/diana/miogams/tes/claio/prop/prop/allpar/fijando/MINLP*.

FICHA AUTOBIOGRÁFICA

Diana Cobos Zaleta

Candidato para el grado de Maestro en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis

Modelos de Optimización Entera Mixta No Lineal en Sistemas de Transporte de Gas Natural

Nacida en Poza Rica, de Hidalgo, Veracruz. Hija del Sr. Macario Cobos Ríos y la Sra. María Magdalena Zaleta Morgado, segunda de tres hermanos. Graduada en la Facultad de Ingeniería Mecánica y Eléctrica (FIME) en la Universidad Autónoma de Nuevo León (1995-2000) como Ingeniero Mecánico Electricista. Obtuvo Mención Honorífica por su excelente desempeño académico. Ingresó a estudiar la Maestría en Ciencias en Ingeniería en Sistemas en Febrero del 2001, después de haberse desempeñado como Ingeniero de Proyectos en 'Iluminación Total, S. A.'. Logró una beca de manutención por el proyecto J33187-A de CONACyT y se desempeñó como Asistente de Investigación durante sus estudios de maestría.

