

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE INGENIERIA MECANICA Y ELECTRICA

DIVISION DE ESTUDIOS DE POSGRADO



UNA METODOLOGIA DE SOLUCION BASADA EN
PROGRAMACION DINAMICA NO SECUENCIAL
Y BUSQUEDA TABU PARA LA OPERACION
EFICIENTE DE SISTEMAS DE TRANSPORTE DE
GAS NATURAL EN ESTADO ESTABLE

POR:

ING. CONRADO BORRAZ SANCHEZ

TESIS

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

CIUDAD UNIVERSITARIA

AGOSTO DE 2004

TM
Z5853
.M2
FIME
2004
.B6

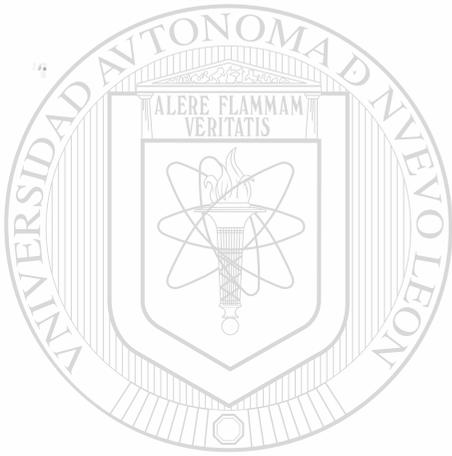
UNA METODOLOGIA DE SOLUCION BASADA EN PROGRAMACION
DINAMICA NO SECUENCIAL Y BUSQUEDA TABU PARA LA OPERACION
EFICIENTE DE SISTEMAS DE TRANSPORTES DE GAS NATURAL
EN ESTADO ESTABLE

C. B. S.

2004



1020150033



UANL

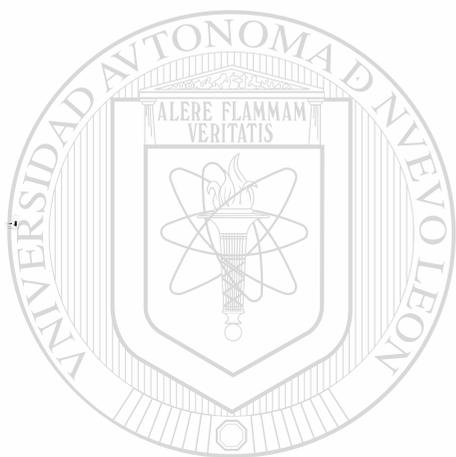
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

989638

TM
Z 5853
.M2
FINE
2004
.B6



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

®

DIRECCIÓN GENERAL DE BIBLIOTECAS



FONDO
TESIS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POSGRADO



**UNA METODOLOGÍA DE SOLUCIÓN BASADA EN
PROGRAMACIÓN DINÁMICA NO SECUENCIAL
Y BÚSQUEDA TABÚ PARA LA OPERACIÓN
EFICIENTE DE SISTEMAS DE TRANSPORTE DE
GAS NATURAL EN ESTADO ESTABLE**

DIRECCIÓN GENERAL DE BIBLIOTECAS
POR:

ING. CONRADO BORRAZ SÁNCHEZ

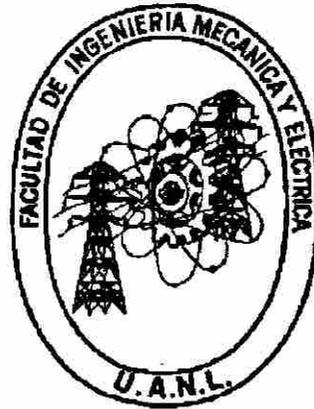
TESIS

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

CIUDAD UNIVERSITARIA

AGOSTO DE 2004

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POSGRADO**



**UNA METODOLOGÍA DE SOLUCIÓN BASADA EN
PROGRAMACIÓN DINÁMICA NO SECUENCIAL
Y BÚSQUEDA TABÚ PARA LA OPERACIÓN
EFICIENTE DE SISTEMAS DE TRANSPORTE DE
GAS NATURAL EN ESTADO ESTABLE**

POR:
DIRECCIÓN GENERAL DE BIBLIOTECAS
ING. CONRADO BORRAZ SÁNCHEZ

TESIS

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

CIUDAD UNIVERSITARIA

AGOSTO DE 2004

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 **“UNA METODOLOGÍA DE SOLUCIÓN BASADA EN PROGRAMACIÓN DINÁMICA NO-SECUENCIAL Y BÚSQUEDA TABÚ PARA LA OPERACIÓN EFICIENTE DE SISTEMAS DE TRANSPORTE DE GAS NATURAL EN ESTADO ESTABLE”**, realizada por el alumno Conrado Borraz Sánchez, matrícula 1224460, 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

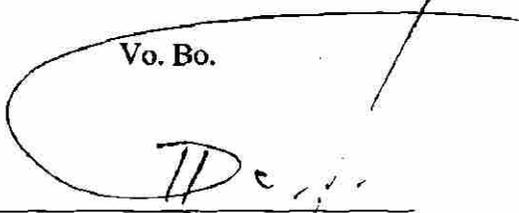
Revisor



Dr. José Luis González Velarde

Revisor

Vo. Bo.



Dr. G. Alan Castillo Rodríguez

Subdirector

División de Estudios de Posgrado

DEDICATORIAS

A mis padres, Conrado de Jesús Borraz León y Magnolia Sánchez de Borraz, quienes realmente me hacen ver el valor de este logro profesional; ya que sin ellos, el primer paso de este ideal nunca hubiera podido cristalizarse, puesto que nada sucede a menos que primero sea un sueño (Carl Sandburg); ya que sin ellos, no encontraría el sentido de la vida para procurar ser el mejor; ya que sin ellos, la simplicidad con que recibí las satisfacciones y pesadumbres no hubiera sido posible jamás.

Eternamente agradecido.

"Si estudio no es para aprender más, sino para ignorar menos, por lo que no quiero la sapiencia de un sabio, sino la certeza de un investigador".

A mis hermanos, Miroslava, José Antonio y Carolina, que desde lejos me brindaban su apoyo de la manera más humilde que un ser humano puede hacerlo... con amor.

"El hombre que no se alimenta de sueños, envejece pronto". William Shakespeare

DIRECCIÓN GENERAL DE BIBLIOTECAS

A toda mi familia en general (en especial a la Familia Ruiz Sánchez), ya que cuando el abatimiento y la nostalgia invadían mi ser, ellos me brindaban su calor desde la más franca razón del alma. Recordándome en todo momento que la capacidad del hombre es inaudita y placentera, la creatividad inimaginable y digna de alabanza, y el orgullo desmesurado e inquebrantable.

"Tan solo dejaré de aprender el día que alguien sea capaz de dibujar el sonido de una lagrima al caer".

AGRADECIMIENTOS

Quisiera agradecer primero a la vida, y a los sueños que ella promete convertir en realidad.

Quisiera expresar aquí mi gratitud a todas aquellas personas que de un modo u otro me facilitaron el camino para realizar este modesto trabajo. En primer lugar extendiendo profundamente mi agradecimiento a mi asesor, el Dr. Roger Z. Ríos Mercado, con quien de alguna manera inicié mi andadura en la filosofía de la investigación, ya que desde un verano científico hasta la culminación de mis estudios de postgrado, ha sido un gran apoyo y guía, y a quien considero realmente un amigo, dándome la oportunidad de ver realmente lo que un investigador es y debe ser.

A mis revisores, dos admirables y distinguidos Doctores, Ada M. Álvarez Socarrás y José Luis González Velarde, quienes se concedieron un tiempo muy valioso para examinar y fortalecer con apasionadas y críticas observaciones este trabajo de tesis, otorgándome la oportunidad de culminar este trabajo tan fascinado como lo fue estrechar su amistad.

Asimismo, a todos mis profesores y compañeros (actuales y antiguos) del Programa en Ingeniería de Sistemas (PISIS) de la Facultad de Ingeniería Mecánica y Eléctrica (FIME) en la Universidad Autónoma de Nuevo León (UANL), por todos aquellos agradables momentos que pasamos dentro y fuera del aula, a quienes he considerado mis más grandes amigos durante mis años de estudio, especialmente a mis profesores: Oscar Chacón, César Villarreal e Igor Litvinchev, y compañeros: Augusto, Daniel, Francisco y Humberto; por su abierta mentalidad en discusiones inimaginables, y hacer de mi estancia en el PISIS mucho más placentera de lo que hubiera podido imaginar.

Un profundo agradecimiento a la familia Aguirre Morales de San Nicolás de los Garza, Nuevo León, por ofrecerme su sincera amistad, haciendo de mi estancia en esta ciudad tan alegre y significativa como si nunca hubiese marchado de la mía; demostrándome en todo momento que la armonía no viene de la mente sino del corazón. Nunca olvidaré todos aquellos gratos momentos que me permitieron compartir con ellos, gracias por considerarme un amigo.

Quedo realmente agradecido con mi *alma mater*, la Universidad Autónoma de Nuevo León, a quién al igual que a la FIME pondré siempre muy en alto. También agradezco la oportunidad al PISIS por aceptarme como estudiante de tiempo completo para contribuir en el área de Investigación de Operaciones. Finalmente, quedo muy agradecido por el soporte de financiamiento para desarrollar esta investigación, el cual fue provisto por el Consejo Nacional para la Ciencia y Tecnología (CONACYT), sustentado por el proyecto J33187-A, el cual me permitió desarrollarme como Asistente de Investigación Graduado.

RESUMEN

Conrado Borraz Sánchez

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:

UNA METODOLOGÍA DE SOLUCIÓN BASADA EN PROGRAMACIÓN DINÁMICA NO SECUENCIAL Y BÚSQUEDA TABÚ PARA LA OPERACIÓN EFICIENTE DE SISTEMAS DE TRANSPORTE DE GAS NATURAL EN ESTADO ESTABLE

Número de Páginas: 141

Objetivos y Método de Estudio

El impulso de esta tesis se enfoca en el problema de minimizar el costo del consumo de combustible en sistemas de tuberías de transmisión de gas natural en estado estable.

El flujo de gas natural, manejado por presión, es transportado a través de tuberías en sistemas de redes. Durante este proceso, la energía y presión en el transporte del gas van disminuyendo debido tanto a la fricción existente entre el gas y las paredes internas de las tuberías, como por la transferencia de calor entre el gas y el medio ambiente. Para mantener el gas fluyendo a través del sistema, es necesario periódicamente incrementar las presiones; por lo que estaciones compresoras son instaladas en puntos específicos de la red. Estas estaciones típicamente consumen entre el 3% y 5% del gas transportado, lo que para las empresas constituye un costo económico relativamente alto. Por lo anterior el problema de encontrar cómo operar óptimamente a los compresores en el manejo del flujo de gas a través del sistema de tuberías, llega a ser significativamente importante.

El estado del arte de la investigación en este problema revela importantes hechos. Primero, hay dos tipos fundamentales de topologías de redes: no cíclicas y cíclicas. Las primeras han recibido la mayor atención durante los últimos 30 años; diferentes metodologías de solución han sido desarrolladas, muchas de ellas basadas en técnicas de Programación Dinámica (DP por sus siglas en inglés, *Dynamic Programming*). En contraste, las topologías cíclicas son mucho más difíciles de resolver, por lo que trabajos en esta área son prácticamente inexistentes.

En este trabajo nos enfocamos a un problema donde se consideran dos tipos de variables de decisión continuas: la tasa de flujo másico a través de cada arco y el valor de las presiones en cada nodo; presentando así un modelo de programación no lineal (NLP por sus siglas en inglés, *Non Linear Programming*) y no convexo, clasificándolo desde la perspectiva de la optimización, dentro de los tipos de problemas denominados *NP-completos* [13]. Este tipo de problemas presenta una característica principal, la cual indica que el esfuerzo de cómputo empleado por cualquier algoritmo para encontrar una solución óptima global crece exponencialmente en relación al incremento del tamaño del mismo.

Una dificultad inherente al problema de minimización del consumo de combustible (PMCC) es que la complejidad que presenta su estructura matemática influye a que los algoritmos tradicionales de programación no lineal, como en el caso del método del Gradiente Reducido Generalizado (GRG), presenten diversas dificultades numéricas en su proceso de búsqueda, causando inclusive en ocasiones que ni al menos una solución factible pueda ser encontrada.

El objetivo de esta tesis es proveer un método de búsqueda efectivo basado en técnicas no tradicionales de programación no lineal, como es el caso de la técnica de programación dinámica no secuencial -motivados por el trabajo de Carter [4]- y la técnica de reducción [41], para encontrar soluciones de buena calidad al PMCC. Además, proponemos la implementación de una heurística de Búsqueda Tabú, como promesa efectiva para obtener soluciones aproximadas de buena calidad sobre sistemas cíclicos de transmisión de gas natural, integrando todas estas técnicas dentro de una metodología de solución como parte de un esquema refinado de optimización, la cual se

divide básicamente en cuatro fases: a) *Fase de preprocesamiento*: Esta fase tiene dos objetivos, el primero es refinar el dominio factible de operación, y el segundo, ser aplicado en las siguientes fases para reducir el tamaño de la red (sin alterar su estructura matemática). b) *Fase de asignación de flujos*: Esta fase es ejecutada basándose en una técnica de asignación “clásica” y un algoritmo de Grafo Reducido que emplea una técnica de reducción [41]. c) *Fase de solución óptima para las variables de presión*: Esta fase se realiza al aplicar una técnica de DP no secuencial. Y d) *Fase de ejecución del procedimiento heurístico*: Esta fase está particularmente desarrollada para reflejar la optimización tanto en las variables del flujo másico como en las variables de presión, ya que se aplica una heurística de TS que emplea una técnica de DP no secuencial y un esquema de memoria corta.

El propósito de este trabajo se constituye como una propuesta matemática-computacional cuya operatividad se establece mediante una metodología apoyada en el empleo de técnicas no tradicionales para obtener soluciones orientadas hacia la optimización en redes de transmisión de gas natural.

Objetivos centrales de esta tesis:

1. Estudiar profundamente la estructura matemática del PMCC para el desarrollo e implementación de técnicas no tradicionales de programación no lineal que permitan encontrar la mejor forma de resolverlo.
2. Estudiar técnicas de preprocesamiento, tal como una técnica de reducción; asimismo, técnicas de programación dinámica y procedimientos heurísticos, como es el caso de una Búsqueda Tabú, como parte integral de la metodología de solución para resolver el PMCC sobre sistemas de redes de tuberías.
3. Implementar un método de solución para encontrar, de manera eficiente, soluciones óptimas sobre redes no cíclicas, y soluciones aproximadas de buena calidad para redes cíclicas.
4. Aplicar un procedimiento heurístico de TS para obtener soluciones aproximadas de buena calidad en sistemas cíclicos de gas natural.

5. Demostrar empíricamente que el método de solución puede tener un eficiente desempeño para obtener soluciones óptimas sobre una gran diversidad de topologías no cíclicas de gas natural.
6. Evidenciar que las soluciones obtenidas por el método de solución, sobre múltiples y diversas topologías cíclicas y no cíclicas, son de mejor calidad que las obtenidas por el método GRG.
7. Demostrar con una extensiva evaluación computacional la calidad de las soluciones obtenidas por el procedimiento heurístico de TS.

Para alcanzar estos objetivos se procedió con la siguiente metodología de investigación:

Como primer etapa, se realizó un profundo estudio de la estructura matemática del PMCC, ya que esto nos permitiría una proyección detallada de la dimensión real del problema; y de esta forma, ser capaces de aplicar eficientemente las técnicas más adecuadas. Como resultado de este análisis, se clasificaron dos tipos de variables de decisión: el flujo másico (x_{ij}) a través de cada arco, y las presiones (p_i) en cada nodo de la red.

En la siguiente etapa se desarrolló una técnica para obtener un conjunto de flujos factibles; para esto, mediante un análisis *a priori*, primero se procedió a realizar -exclusivamente para esta etapa- una reclasificación de las topologías de red; procediendo así, a la asignación de los flujos factibles sobre cada arco de la red con base en dos diferentes técnicas. Esta etapa de asignación de flujos factibles se resume con la siguiente analogía: *“Todas las topologías de grado 1 utilizarían una técnica de asignación ‘clásica’, y todas aquellas con grado mayor a 1 emplearían una técnica de reducción”*.

En una etapa consecuente, se introdujo una técnica de programación dinámica no secuencial para la búsqueda de presiones óptimas (para el conjunto de flujos factibles preespecificado) sobre cada nodo de la red. Es pertinente aclarar que hasta esta etapa, se obtuvieron soluciones óptimas sobre las topologías no cíclicas, ya que los flujos sobre éstas se determinan de manera única [41], a diferencia de las topologías cíclicas, donde

las soluciones encontradas son puntos factibles iniciales para proceder a una etapa final. En esta etapa final ejecutamos un procedimiento heurístico de Búsqueda Tabú para optimizar sobre las variables de flujo, este procedimiento heurístico emplea una técnica de DP no secuencial para obtener las presiones óptimas. Esta última etapa se introdujo como promesa de mejorar la calidad de las soluciones encontradas sobre las redes cíclicas, convirtiendo así a este trabajo de tesis en una nueva y especial contribución al PMCC en sistemas de transmisión de gas natural.

Como parte importante en este trabajo de tesis y motivados por la fase de la evaluación computacional, se diseñó una base de datos, que desde la perspectiva de la optimización en redes, se clasifica en dos principales grupos topológicos: no cíclicos y cíclicos. La base de datos cuenta actualmente con 20 topologías no cíclicas (10 del tipo lineal y 10 del tipo árbol) y 10 topologías cíclicas, todas con diversas estructuras y conteniendo múltiples tipos de compresores. Esta base de datos es completamente accesible desde la siguiente dirección URL: <http://www.valma.fime.uanl.mx/~pisis/>

Contribuciones y Conclusiones

Las contribuciones científicas de esta tesis fueron:

- Implementación de un método de búsqueda eficiente (NONDP_TS) para encontrar soluciones óptimas en redes no cíclicas y soluciones factibles a toda una gama de diversas topologías cíclicas.
- Elaboración de una amplia base de datos con diversas instancias de sistemas de transmisión de gas natural, incluyendo topologías cíclicas y no cíclicas, para la etapa de evaluación computacional de este trabajo, y por supuesto, para posteriores experimentaciones de cualquier colaborador en el problema de optimización en redes de transmisión de gas natural.
- Evidencia empírica de que las soluciones óptimas encontradas con nuestro método de solución son de mejor calidad que las obtenidas con métodos tradicionales como es el caso del GRG.

- Desarrollo e implementación de un procedimiento heurístico de Búsqueda Tabú, para encontrar soluciones aproximadas de buena calidad sobre redes cíclicas, las cuales remarcamos como las más difíciles de resolver; y trabajos en esta área son prácticamente inexistentes, provocando en los pocos métodos aplicados que algunos casos no sean capaces de entregar al menos un punto factible.
- Evidencia satisfactoria y determinante de la buena calidad que presentan las soluciones obtenidas por el procedimiento heurístico NONDP_TS, basándonos en una exhaustiva evaluación computacional y experimentación comparativa contra cotas inferiores. Presentando en muchos casos, que estas soluciones están a menos del 1% de desviación con respecto a las cotas inferiores.

Un punto relevante fue el hecho de que, con la mira hacia una implementación eficiente y una mayor validez de las evaluaciones computacionales, todos los datos relacionados con las estaciones compresoras fueron provistos de compresores reales entregados por una firma consultora en la industria de gasoductos.

Finalmente destacamos desde el marco científico, que se ha demostrado empíricamente que este problema de optimización en redes, al cual hemos definido particularmente como el PMCC en sistemas de redes de transporte de gas natural (*NP-completo*), pudo resolverse eficientemente al desarrollarse y aplicarse un algoritmo no exponencial, el cual entregó soluciones de muy buena calidad, sobrepasando en mucho a los métodos tradicionales de programación no lineal existentes.

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 Objetivo de la Tesis..... 3

1.3 Justificación..... 4

1.4 Estructura de la Tesis..... 5

CAPÍTULO 2. LITERATURA PREVIA: ANTECEDENTES 7

2.1 Métodos basados en Programación Dinámica..... 7

2.2 Métodos basados en Técnicas de Gradiente..... 10

2.3 Otras Técnicas..... 10

2.4 Resumen..... 12

CAPÍTULO 3. PLANTEAMIENTO DEL PROBLEMA 14

3.1 Suposiciones..... 14

3.2 Definición del Modelo Matemático..... 16

3.3 Dominio de Operación Factible de una Estación Compresora..... 18

3.3.1 Representación Algebraica del Dominio $D^{unit}_{(i,j)}$ 23

3.4 Función de Costo para un Compresor Centrífugo..... 26

CAPÍTULO 4. MÉTODO DE SOLUCIÓN: PREPROCESAMIENTO Y PROGRAMACIÓN DINÁMICA 29

4.1 Refinamiento del Modelo.....	31
4.1.1 Derivación de Cotas para x_{ij}	32
4.1.2 Derivación de Cotas para p_i	32
4.1.3 Técnica de Reducción.....	33
4.2 Descripción del Algoritmo NDP.....	35
4.2.1 Fase 1: Procedimiento de Generación de Flujos.....	36
4.2.2 Fase 2: Procedimiento de Búsqueda de Presiones Óptimas.....	43

CAPÍTULO 5. MÉTODO DE SOLUCIÓN: BÚSQUEDA TABÚ 55

5.1 Heurísticas.....	55
5.2 Metaheurísticas.....	57
5.3 Búsqueda Tabú.....	60
5.3.1 Metodología General.....	60
5.3.2 Antecedente Científicos de la Búsqueda Tabú.....	63
5.3.3 Implementación del Procedimiento de Búsqueda Tabú.....	65

CAPÍTULO 6. EVALUACIÓN COMPUTACIONAL 74

6.1 Descripción de la Base de Datos.....	74
6.1.1 Descripción del Nombre Mnemónico.....	74
6.1.2 Topologías No Cíclicas: Lineales y de Árbol.....	75
6.1.3 Topologías Cíclicas.....	76
6.2 Resultados Computacionales del Algoritmo NDP.....	78
6.2.1 Experimento 1: Evaluación sobre Redes No Cíclicas.....	79
6.2.2 Experimento 2: Evaluación sobre Redes Cíclicas.....	81
6.2.3 Experimento 3: Algoritmo NDP vs Método GRG sobre Redes Cíclicas.....	82

6.2.4 Conclusiones.....	83
6.3 Evaluación Computacional del Procedimiento de Búsqueda Tabú.....	84
6.3.1 Experimento 1: Evaluación de los Parámetros.....	85
6.3.2 Experimento 2: Evaluación de la Calidad de las Soluciones.....	87
6.3.3 Conclusiones.....	92

CAPÍTULO 7. CONCLUSIONES Y APORTACIONES 94

7.1 Conclusiones Generales.....	94
7.2 Aportaciones Científicas.....	97
7.3 Recomendaciones para Futuras Investigaciones.....	98

BIBLIOGRAFÍA	102
---------------------	------------

LISTA DE TABLAS	107
------------------------	------------

LISTA DE FIGURAS	108
-------------------------	------------

APÉNDICE A: Evaluación del Parámetro <i>Iter_max</i> de NONDP_TS	112
---	------------

APÉNDICE B: Evaluación de $\Delta p = \{1, 5\}$ en el Procedimiento NONDP_TS	114
--	------------

APÉNDICE C: Evaluación de la Convergencia del Procedimiento NONDP_TS sobre dos Parámetros de TS: <i>Ttenure</i> y <i>Nei_Size</i>	117
--	------------

APÉNDICE D: Teoría de Complejidad	127
--	------------

FICHA AUTOBIOGRÁFICA	141
-----------------------------	------------

CAPÍTULO 1

INTRODUCCIÓN

Siendo la energía parte fundamental del desarrollo económico de México y, por supuesto base de la vida moderna, desde el ámbito industrial, la producción y comercialización de energía son actividades económicas muy importantes del país.

El gas natural, también conocido como gas metano, es un combustible incoloro e inodoro al que se le agregan odorantes químicos, como el mercaptano, para que pueda detectarse en caso de fuga. Se trata de uno de los combustibles más utilizados en el mundo y al que se tiene acceso en nuestro país a través de una red de distribución que crece constantemente.

1.1 Descripción y Relevancia del Problema

El gas natural se distribuye a través de tuberías de acero o de polietileno, materiales de alta durabilidad y resistencia, que son adecuados para cualquier tipo de suelo y que permiten disponer del combustible las 24 horas del día, durante los 365 días del año. El flujo de gas natural, manejado por presión, es transportado a través de tuberías en sistemas de redes. Durante esta fase, la energía y presión en el transporte del gas van disminuyendo debido a la fricción existente entre el gas y las paredes internas de las tuberías, así como por la transferencia de calor entre el gas y el medio ambiente. Para mantener el gas fluyendo a través del sistema, es necesario periódicamente incrementar las presiones, por lo que estaciones compresoras son instaladas en la red. Estas estaciones normalmente consumen entre el 3% y 5% del gas transportado. Este costo de transportación es significativo, ya que

la cantidad de gas que se transporta en los sistemas reales en el mundo es enorme. Hoy en día, una red podría consistir en miles de tuberías, docenas de estaciones compresoras, y muchos otros dispositivos, como válvulas y reguladores. Así, una red podría transportar miles de MMCFD de gas ($1 \text{ MMCFD} = 10 \times 10^6$ pies cúbicos por día). Se estima [42] que la optimización global podría ahorrar, al menos, el 20% de combustible consumido por las estaciones. 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 [14]. Aquí, el problema de encontrar cómo operar óptimamente a los compresores para el manejo del flujo, en una red de tuberías, llega a ser significativamente importante.

El estado del arte de la investigación sobre este problema revela importantes hechos. Primero, hay dos tipos fundamentales de topologías de redes: no cíclicas y cíclicas. Las primeras han recibido la mayor atención durante los últimos 30 años; diferentes metodologías de solución han sido desarrolladas, muchas de ellas basadas en técnicas de Programación Dinámica (DP por sus siglas en inglés, *Dynamic Programming*). En contraste, las topologías cíclicas son mucho más difíciles de resolver, por lo que trabajos en esta área son prácticamente inexistentes.

Los elementos fundamentales en este tipo de redes son las estaciones compresoras, las cuales proporcionan el incremento de presión necesaria para llevar a cabo el transporte del gas dentro del sistema de tuberías.

Existen diversas variaciones de este problema, las cuales dependen de las suposiciones en el modelo. Una de estas variaciones hace referencia a los trabajos [10] y [11], donde se presenta un modelo de programación no lineal (NLP por sus siglas en inglés, *Non-Linear Programming*), y se reporta una evaluación computacional utilizando el *Algoritmo del Gradiente Reducido Generalizado* (GRG), encontrando que el algoritmo tuvo un éxito relativo al entregar soluciones óptimas locales sobre topologías no cíclicas, aunque el principal enfoque de estos trabajos estaba en tratar con estructuras cíclicas. Otra variación es el modelo de programación no lineal entero mixto (MINLP) presentado por Cobos-Zaleta y Ríos-Mercado [6], donde la introducción de una variable entera convertía al modelo en un problema totalmente diferente.

En este trabajo nos enfocamos a un problema donde se consideran dos tipos de variables de decisión continuas: la tasa de flujo másico a través de cada arco y el valor de las presiones en cada nodo [49]; presentando así un modelo de programación no lineal (NLP) y no convexo, clasificado desde la perspectiva de la optimización, dentro de los tipos de problemas denominados *NP-completos* [13] (véase Apéndice D).

1.2 Objetivo de la Tesis

El objetivo de este trabajo de investigación es proveer una metodología de solución que aplica una técnica de reducción [41] y una técnica de programación dinámica no secuencial (motivados por el trabajo de Carter [4]). Además, debido a que la complejidad de muchos problemas combinatorios -incluyendo al desarrollado en esta tesis- está descrita por una función exponencial, estos problemas se consideran intratables o *NP-completos*, es decir, no existen algoritmos exactos acotados de forma polinomial para resolver esta clase de problemas; por lo tanto, se emplea un procedimiento heurístico de Búsqueda Tabú (TS por sus siglas en inglés, *Tabu Search*) con una estrategia de memoria corta con la promesa de producir soluciones de buena calidad en un tiempo razonable al problema de minimización del consumo de combustible (PMCC) en redes cíclicas de gas natural. Para lograr esto, se propone la integración de todas estas técnicas no tradicionales dentro de una metodología de solución como parte de un esquema refinado de optimización.

La metodología de solución consiste básicamente de cuatro componentes:

- Preprocesamiento*: Esta fase es ejecutada tanto para refinar el dominio factible de operación dado por las variables de decisión, así como para reducir el tamaño de la red (sin alterar su estructura matemática).
- Asignación de flujos*: En esta fase, un conjunto de flujos factibles es obtenido por dos diferentes técnicas: una técnica de asignación “clásica” y un algoritmo de Grafo Reducido que emplea una técnica de reducción [41]. Esta fase de asignación se basa en un patrón que presentan las instancias de red a resolver.
- Solución óptima de las variables de presión*: En esta fase, un conjunto

de presiones (para los flujos determinados en la fase anterior) es encontrado al aplicar una técnica de DP no secuencial. Bajo los fundamentos teóricos encontrados en [41] (donde se demuestra que en topologías no cíclicas existe una única manera de asignar los flujos máxicos), en esta fase se obtienen soluciones óptimas globales en topologías no cíclicas para un dominio de operación factible discretizado. Pero, dado el hecho de que en topologías cíclicas no es posible encontrar una única forma de asignar los flujos, en esta fase la técnica de DP no secuencial entrega soluciones factibles para los sistemas cíclicos, por lo que se da seguimiento al cuarto componente.

d) *Procedimiento heurístico:* Aquí, nosotros proponemos la implementación de un procedimiento heurístico de TS para resolver topologías cíclicas, el cual empleando una técnica de DP no secuencial, refleja la optimización tanto en las variables del flujo máxico como las variables de presión, finalizando de esta manera con la metodología de solución.

Resumiendo, se comprende que la optimización en redes de transmisión de gas natural es el objetivo general de este trabajo, donde el propósito fundamental del PMCC es llevar el gas desde los centros de producción hasta los sitios donde se requiere para su consumo a través de sistemas de redes de distribución dentro de una región específica, sean ciudades, países o inclusive a nivel mundial. Por lo tanto, basándonos en el modelo matemático del PMCC y el empleo de diversas técnicas no clásicas de optimización no lineal, el objetivo específico es encontrar los niveles de operación óptimos para las estaciones compresoras que conforman el sistema de red, de tal forma que el combustible consumido por las mismas sea el mínimo, al tiempo que se satisfaca la demanda especificada y las demás restricciones tecnológicas del problema.

1.3 Justificación

Es preciso señalar que existen muchas ciudades en el mundo que están provistas de redes citadinas para satisfacer la demanda de los usuarios, por ejemplo: hay tuberías que parten de Siberia y llevan el gas hasta Europa; o desde Argentina,

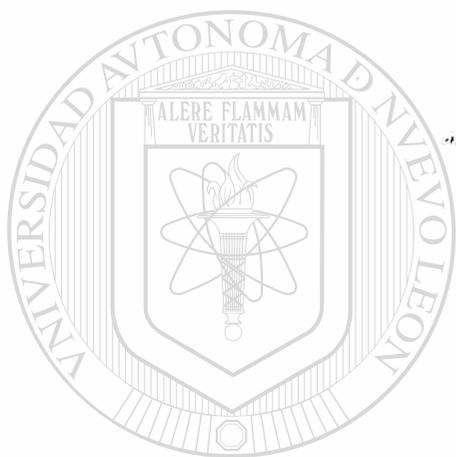
para satisfacer la demanda de Chile o de Brasil. En Colombia, desde 1976 se puso empeño en que los ingenieros y técnicos aprendieran a diseñar redes de gas y, en nuestros días, es una política gubernamental la “Masificación del gas”, con lo cual se pretende proveer de energía barata a la gran mayoría de los ciudadanos. En Venezuela, dado que es un país eminentemente petrolero, el gas natural ha sido utilizado desde comienzos del siglo XX; existen redes de tuberías que cubren toda la zona norte del país y ciudades, como Maracaibo, que poseen redes de gas desde el año 1938.

Este trabajo de tesis trata un tema que concierne tanto al sector privado como al sector público, de ahí su importancia y justificación pues tiene un tremendo impacto económico cuando se habla de miles de millones de pesos, aún con una mejora marginal del 1% al 2%. Se comprende entonces que la complejidad y las características de este problema, lo convierten en un excelente reto como tema de investigación. La satisfacción de presentar esta tesis, deja muy atrás las arduas tareas y preocupaciones personales por verla concluida como un reto que valió la pena realizar.

1.4 Estructura de la Tesis

La estructura de este trabajo está dada de la siguiente forma: Los trabajos más relevantes sobre esta área son presentados en el Capítulo 2. En el Capítulo 3 se introduce un estudio matemático del PMCC, presentando las suposiciones asumidas, la descripción detallada del modelo NLP y la estructura matemática del dominio factible no convexo de las estaciones compresoras. En el Capítulo 4 se presenta la primera fase del método de solución para el PMCC, describiendo las técnicas de reducción y de DP no secuencial en el proceso de búsqueda de soluciones óptimas. En el Capítulo 5 se presenta la fase final del método de solución, describiendo los fundamentos que permiten establecer, partiendo de la noción de heurística, un procedimiento TS para obtener soluciones aproximadas de buena calidad al PMCC sobre redes cíclicas. Así entonces, la efectividad y calidad de las soluciones obtenidas por las dos fases en la metodología de solución son evaluadas en una exhaustiva

experimentación computacional sobre una gran variedad de instancias topológicas en el Capítulo 6, el cual también incluye una extensa evaluación comparativa entre los algoritmos propuestos en esta tesis y los métodos tradicionales como el GRG. Finalmente, las conclusiones generales y contribuciones obtenidas al PMCC, así como las recomendaciones para futuras investigaciones son discutidas en el Capítulo 7.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

CAPÍTULO 2

LITERATURA PREVIA: ANTECEDENTES

En este capítulo, basándonos en la literatura previa, presentamos los antecedentes científicos del problema de minimización del consumo de combustible en sistemas de redes de transmisión de gas natural. Se exponen en la primera sección los métodos de solución al PMCC basados en Programación Dinámica, seguido de los métodos tradicionales de programación no lineal, tal como el Gradiente Reducido Generalizado, y otras técnicas empleadas a lo largo de las últimas décadas de investigación en optimización en redes de transmisión de gas natural.

2.1 Métodos basados en Programación Dinámica

En esta sección presentamos las principales contribuciones con técnicas de programación dinámica, las cuales se han aplicado para intentar resolver el problema de minimización en sistemas de transmisión de gas natural.

La Programación Dinámica (DP por sus siglas en inglés, *Dynamic Programming*) fue inventada por el matemático Richard Bellman [1] en 1957. Sus primeras aplicaciones fueron para la optimización de sistemas diferenciales dinámicos en el dominio del tiempo, o mejor dicho, el dominio espacial. A principios de 1950 se le dio el nombre de “programación”, por ser comúnmente utilizada como técnica matemática orientada por computadora. Los problemas orientados en el tiempo, para los cuales la Programación Dinámica fue primeramente aplicada, enfatizaron estrictamente la recursividad, derivación secuencial y formulación sobre este método. En los siguientes años fueron encontrados cientos de aplicaciones utilizando DP, incluyendo la optimización de tuberías de líquidos por Jefferson [20].

La DP en optimización de redes fue aplicada originalmente sobre sistemas lineales a finales de 1960. Fue una de las técnicas más útiles debido tanto a su rápido desempeño computacional como a su insensibilidad a la no linealidad sobre sistemas secuenciales.

Fueron Wong y Larson [46], quienes publicaron el primer trabajo de optimización de tuberías de gas basándose en la DP. Wong y Larson aplicaron el método para la minimización del costo de combustible únicamente a sistemas lineales usando una formulación recursiva.

En la industria de las tuberías, el primer intento de optimización de estructuras con ramas usando DP, fue hecha por Zimmer [50] en 1975. Una aproximación similar fue descrita por Lall y Percell [24] en 1990, quienes permitieron una rama divergente en sus sistemas; incorporando además al modelo una variable de decisión que representaba el número de compresores que serían operados en la estación compresora.

A finales de 1980 fueron producidos métodos reconocidos de enumeración con DP híbrida [4], con la finalidad de una optimización más general en redes con ramas y ciclos. Aun cuando estos métodos fueron muy exitosos en la optimización en redes de tuberías, en algunos casos, la naturaleza híbrida de los métodos era causa de un gran tiempo de ejecución o una reducida exactitud en la solución del problema discretizado.

En 1989, Gilmour, Luongo y Schroeder [14] publicaron una aproximación jerárquica que permitió tanto ciclos como ramas de complejidad arbitraria. Esto fue un gran avance, ya que pudieron determinarse las configuraciones de las tuberías en el mundo real; la desventaja era que su técnica no era puramente programación dinámica. Básicamente, la DP fue usada para configurar de manera óptima y secuencial los elementos de un sistema de red, reduciéndolo a un problema combinatorio mucho más pequeño, pero sin posibilidad de obtener su solución de manera recursiva. En la aplicación de esta técnica se verificaba

una condición importante: si el sistema reducido era suficientemente pequeño, entonces era resuelto de modo exacto por vía enumeración, si no era resuelto inexactamente usando la técnica de recocido simulado. Esta aproximación jerárquica trabajaba muy bien sobre muchas tuberías complejas, pero en algunos casos el costo computacional podía elevarse demasiado.

Fue a principios de 1990, cuando se aplicó la DP para optimizar sistemas lineales (gun-barrel), siendo de uso limitado para las compañías de tuberías, las cuales contaban con una gran diversidad de sistemas. Frecuentemente, cuando estas compañías requerían de la optimización, ellas consideraban solo pequeñas subredes de la red completa; de esta forma, del estudio realizado, en algunas ocasiones era posible ajustar los resultados obtenidos en cada subred en el sistema completo, aunque esto era un proceso fortuito.

La aproximación jerárquica depende de los valores del flujo conocidos en el sistema. Sin embargo, si los valores de los flujos -preferentemente los que fueron preespecificados por el usuario- permitieran una variación, entonces puede aplicarse otro algoritmo de optimización antes de la DP en la búsqueda jerárquica para obtener un óptimo sobre estos flujos, y así, proseguir con el proceso de aproximación jerárquica con una mayor exactitud.

Las características principales que la DP presentaba, para hacer pensar que era una técnica poderosa, eran que garantizaba encontrar el óptimo global y la facilidad con que manejaba la no convexidad. Y claro, una de las desventajas principales que presentaba era que su aplicación estaba estrictamente limitada a redes no cíclicas, como estructuras en línea recta y de árbol; además que el esfuerzo computacional se incrementaba exponencialmente según la dimensión del problema.

Fue Carter [4], usando DP no secuencial, quien desarrolló un algoritmo que trataba redes cíclicas cuando las variables del flujo másico estaban fijas. Sin embargo, la limitación es que considera flujos fijos. Es aquí donde se pretende contribuir en esta tesis, en desarrollar una metodología que pueda ir modificando los flujos usando la DP no secuencial iterativamente.

2.2 Métodos basados en Técnicas de Gradiente

En 1987, Percell y Ryan [34] emplean una metodología diferente, basándose en el método de optimización no lineal del gradiente reducido generalizado (GRG por sus siglas en inglés, *Generalized Reduced Gradient*). Las ventajas que el método GRG presenta son, en primer lugar, que evita el problema de dimensionalidad, y segundo, que puede ser aplicado a redes cíclicas. No obstante, dado que es un método basado en una búsqueda por gradiente, teóricamente no garantiza encontrar un óptimo global, especialmente en presencia de variables de decisión discretas, quedando así, atrapado en un óptimo local.

Flores-Villarreal y Ríos-Mercado en [10] y [11] amplían el estudio anterior mediante una extensa evaluación computacional del método GRG sobre una diversa base de datos del problema.

2.3 Otras Técnicas

Recientemente, Kim, Ríos-Mercado y Boyd [23], motivados por el trabajo de Carter, propusieron un algoritmo que iterativamente ajustaba la variable del flujo másico de forma heurística, resolviendo el problema con estructuras cíclicas, basándose en un procedimiento iterativo de dos etapas. En la primera etapa, las variables de flujo de gas están fijas y las variables de presión óptimas son obtenidas usando DP. En la segunda etapa, 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 así la estructura fundamental de la red.

Wu, Boyd y Scott [48] presentaron un modelo matemático para la minimización del costo de combustible en una estación compresora con una sola unidad. Wu et al. [49] hacen un análisis para este mismo problema, en donde ampliaron el estudio a múltiples unidades en las estaciones compresoras.

En estudios más recientes, Ríos-Mercado et al. [41] hacen uso de los resultados de teorías de grafos y análisis funcional, basándose en la investigación doctoral sobre la

minimización del consumo de combustible de Wu [47], para la demostración de existencia de la solución única para un sistema de ecuaciones algebraicas no lineales que surgen de los flujos de una red de transporte, concluyendo con un método de descomposición de la red, el cual presentaba como una característica relevante la disminución del tamaño de la red, sin alterar su estructura matemática.

Villalobos-Morales en [43] presenta técnicas de preprocesamiento para resolver el PMCC, en las cuales incluye el escalamiento, acotación de variables y asignación de un punto inicial, concluyendo su trabajo con una buena convergencia de la solución. Otro trabajo relacionado con estas técnicas de preprocesamiento puede encontrarse en [45].

Cobos-Zaleta y Ríos-Mercado en [5] y [6] presentaron recientemente una técnica de solución basada en un algoritmo de aproximación externa con relajamiento de igualdades y penalidad aumentada OA/ER/AP [18] (propuesto por Grossmann [12]), sobre un dominio discreto y continuo en un modelo de programación no lineal entero mixto (MINLP). La característica principal que estos trabajos presentaban era la introducción de una nueva variable discreta que representaba el número de unidades compresoras funcionando dentro de cada estación compresora; obteniendo resultados sobresalientes, ya que el método aplicado tenía la ventaja de tratar con problemas no convexos.

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 [28] es la mejor referencia en este tema.

Se sabe con certeza que los modelos transientes presentan una mayor complejidad, y que se han aplicado técnicas de optimización para resolverlos (véase las referencias [29] y [31]), obteniéndose un éxito moderado al igual que en diseño de redes (véase la referencia [30]). Para tener una idea más concisa de las técnicas de optimización aplicadas al problema de redes de transmisión de gas natural, véase la referencia [40]. Cabe aclarar que actualmente, las metodologías propuestas tienen un

buen desempeño bajo ciertos supuestos; sin embargo, al acercarnos más al mundo real, se ve la necesidad de realizar una mayor investigación para el desarrollo eficiente de nuevos procedimientos enfocados a la optimización.

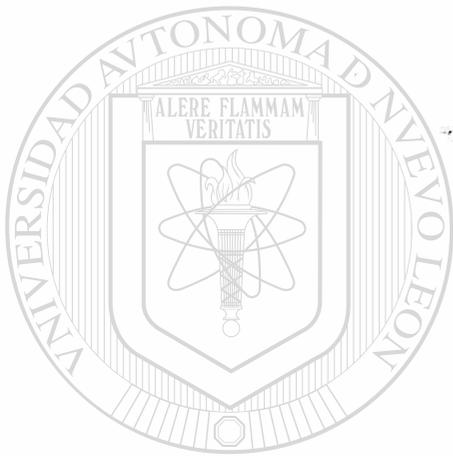
2.4 Resumen

Una extensa cantidad de literatura en aplicaciones tanto de simulaciones numéricas (ver [28]), de DP (ver [14], [20], [24], [46] y [50]), técnicas de gradiente (ver [10], [11] y [34]) y otras técnicas (ver [48]) han sido desarrolladas en las últimas décadas para el PMCC, prácticamente limitadas a redes con estructuras no cíclicas, tal como topologías lineales y de árbol; obteniendo un éxito considerable, especialmente con la ayuda de poderosas computadoras digitales; aunque en algunas de ellas, como es el caso de DP, los cálculos se incrementan exponencialmente con respecto a la dimensión del problema, comúnmente referida como la “*maldición de la dimensionalidad*”.

Diversos trabajos, algunos relacionados con DP no secuencial (ver [4] y [23]), teorías de grafos y análisis funcional (ver [41]), técnicas de preprocesamiento (ver [43] y otros (ver [5] y [6]) han sido desarrollados con la promesa de abordar topologías cíclicas. Sin embargo, dado que el problema es muy difícil de resolver debido a la presencia de una función objetivo y una región de factibilidad no convexas, la aplicación de estos algoritmos de optimización sobre redes cíclicas está relativamente en una fase de desarrollo; por lo que, algoritmos efectivos son prácticamente inexistentes. A saber, una topología cíclica es una red con al menos un ciclo conteniendo dos o más estaciones compresoras.

En este trabajo de tesis proponemos una metodología de solución eficiente para resolver el PMCC en sistemas de transmisión de gas natural, particularmente sobre topologías de redes cíclicas, dado que éstas presentan un excelente reto. Esencialmente, nuestra metodología de solución está basada tanto en una técnica

de DP no secuencial como en un procedimiento heurístico de búsqueda local; aplicando preliminarmente técnicas de preprocesamiento, que incluye una técnica de reducción de red con la promesa de explotar más eficientemente la estructura matemática del problema, y obtener por consiguiente, soluciones aproximadas de mayor calidad.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

CAPÍTULO 3

PLANTEAMIENTO DEL PROBLEMA

En este capítulo introducimos el planteamiento general del PMCC en redes de transmisión de gas natural, donde las suposiciones asumidas y el modelo matemático son presentados. Asimismo, tanto el dominio de operación factible como la función de costo de las estaciones compresoras son descritos detalladamente.

3.1 Suposiciones

En años recientes, el desarrollo de nuevas y mejoradas técnicas se ha hecho sentir en casi todas las facetas de la investigación científica. La solución no puede ser obtenida por la aplicación de simples técnicas, ya que se requiere de un mayor estudio en presencia de infinidad de restricciones. Esto cubre un terreno considerable puesto que nos encontramos en el mundo real.

Sería presuntuoso decir que el diseño matemático del problema, en su estado actual de desarrollo, puede manejar todas las situaciones que implican sus restricciones. Aún así, constantemente el punto principal de un diseño matemático es proporcionar un marco de referencia para examinar estas situaciones en forma lógica y sistemática. En otras palabras, el planteamiento de suposiciones bien definidas -partiendo de la vida real-, proporciona el modelo necesario para estudiar su estructura matemática y su dominio de factibilidad, para encontrar las mejores técnicas que aprovechen al máximo todas sus características. Por lo tanto, en nuestro trabajo de tesis hacemos las siguientes suposiciones para resolver el PMCC:

- El problema está en estado estable. Esto es, nuestro modelo proveerá soluciones a sistemas que han estado operando por una relativa cantidad grande de tiempo. Un análisis transiente requeriría incrementar el número de variables y la complejidad del problema, y esto es de hecho, uno de los más grandes desafíos en esta área.
- 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 del suministro es entregado completamente al flujo total demandante, sin pérdida alguna. Se tiene conocimiento de que las estaciones compresoras se alimentan con cierto porcentaje del combustible transportado dentro de las tuberías, por lo que para mantener el supuesto de estado estable consideramos el costo de este consumo como un costo (de oportunidad) extra en el modelo, lo que representaría la cantidad que nosotros deberíamos invertir si tuviésemos que comprar el combustible de terceras partes.
- Se considera un sistema isotérmico. Es decir, la temperatura ambiente se mantiene constante en cada ducto. Esto con la finalidad de hacer más tratable la restricción del flujo en los ductos.

-
- La red es dirigida. Esto significa que cada arco en la red tiene una dirección preespecificada.
 - En cada estación compresora se establece un número fijo de unidades compresoras. Es decir, se considera como parámetro conocido el número de compresores operando en cada estación. Sin embargo, en la práctica es posible considerar este dato como una variable de decisión, lo que convertiría este problema en un programa entero mixto no lineal, requiriendo una metodología de solución completamente diferente.
 - Las unidades compresoras que se manejan en cada estación compresora son centrífugas, idénticas e instaladas en paralelo. Se asumen centrífugas por ser los más utilizados en la industria del gas natural, ya que son de sencilla construcción,

están libres de mantenimiento y permiten un funcionamiento continuo durante largos períodos de tiempo. Además, se instalan en paralelo con la finalidad de suponer que el flujo que entra en la estación es repartida equitativamente entre el número de unidades compresoras dentro de ella, facilitando el análisis.

- El problema es determinístico, es decir, cada parámetro se supone conocido con certeza.

3.2 Definición del Modelo Matemático

El modelo matemático, definido tanto por el conjunto de restricciones como por la función objetivo, fue planteado como un modelo no lineal y no convexo. Para ello, la siguiente notación es utilizada.

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 de la tubería (i,j) ; $(i,j) \in A_p$
P_i^L, P_i^U	Límites inferior y superior de las presiones en cada nodo i ; $i \in V$
B_i	Tasa de flujo neto 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}	Flujo másico en el arco (i,j) ; $(i,j) \in A$
p_i	Presión en el nodo i ; $i \in V$

Formulación:

$$\text{Min} \quad \sum_{(i,j) \in A_c} g_{(i,j)}(x_{ij}, p_i, p_j) \quad (3.1)$$

Sujeto a:

$$\sum_{\{j|(i,j) \in A\}} x_{ij} - \sum_{\{j|(i,j) \in A\}} x_{ji} = B_i \quad i \in V \quad (3.2)$$

$$x_{ij} \leq U_{ij} \quad (i,j) \in A_p \quad (3.3)$$

$$p_i^2 - p_j^2 = R_{ij} x_{ij}^2 \quad (i,j) \in A_p \quad (3.4)$$

$$P_i^L \leq p_i \leq P_i^U \quad i \in V \quad (3.5)$$

$$(x_{ij}, p_i, p_j) \in D_{(i,j)}^{\text{unit}} \quad (i,j) \in A_c \quad (3.6)$$

$$x_{ij}, p_i \geq 0 \quad (i,j) \in A, i \in V \quad (3.7)$$

La expresión (3.1) representa la función objetivo, donde la función $g_{(i,j)}$ mide el costo del combustible consumido por una estación compresora. Lo que se indica en esta expresión es la minimización del costo de combustible consumido por el desempeño del

sistema en todas las estaciones compresoras. Esta función depende de las variables de decisión (x_{ij}, p_i, p_j) , $\forall (i,j) \in A_c$. Las ecuaciones y desigualdades restantes representan el conjunto de restricciones que definen el dominio de operación factible del PMCC.

Dos restricciones típicas en cualquier problema de flujo en redes son las presentadas por (3.2) y (3.3). La primera de ellas representa el balance de flujo en cada nodo de la red, donde un nodo fuente es definido por un valor positivo de B_i , un nodo demanda por un valor negativo, y un valor de B_i igual a cero representa un nodo de paso. La suposición de balance implica $\sum_{i \in V} B_i = 0$. La segunda ecuación es una restricción que define el límite superior de la capacidad del flujo en cada ducto de la red.

El enfoque principal, el cual hace realmente difícil este problema se presenta con las restricciones restantes. La ecuación (3.4) representa la dinámica del flujo de gas a través de cada ducto de la red, es decir, nos muestra la relación que existe entre la disminución de presión y el flujo de gas (válida para gases de alta presión, como es este el caso). Esta es conocida como la ecuación de Osiadacz [28]. Para un análisis más detallado véase [22]. La restricción (3.5) denota los límites inferior y superior de la presión en cada nodo de la red. Estos límites son definidos por las propiedades físicas de cada compresor. La expresión (3.6) representa el dominio $D^{unit}_{(i,j)}$ de operación factible para la estación compresora (i,j) , la cual se explicará detalladamente en la siguiente sección. Finalmente, el modelo matemático restringe por la expresión (3.7) a las variables de decisión $(x_{ij}$ y $p_i)$ como no negativas.

3.3 Dominio de Operación Factible de una Estación Compresora

En esta sección, basándonos en el trabajo de Wu et al. [47] y [49], presentamos un estudio matemático minucioso que define el dominio factible de operación de una estación compresora de gas natural; este dominio de operación está definido por un espacio (q_{ij}, h_{ij}, s_{ij}) basado en las variables de decisión que el operador de la red conoce

directamente, a saber:

q_{ij}	Tasa interna del flujo volumétrico en el 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) ; $(i,j) \in A_c$

Sin embargo, desde la perspectiva de una implementación computacional, es preferible trabajar en el espacio de (x_{ij}, p_i, p_j) ; por lo que, en la siguiente sección 3.3.1 -como contribución de esta tesis-, se desarrolla y describe una representación algebraica equivalente al dominio de operación factible tratado a continuación; esto, en la mira de

dos objetivos en particular: 1) Hacer más eficiente el desempeño de las técnicas implementadas, al explotar de una mejor manera la estructura matemática de un compresor; y 2) la optimización de los recursos computacionales en la aplicación de los métodos de solución para el desarrollo de este trabajo de investigación.

Pues bien, como se mencionó en la Sección 3.1 (suposiciones del modelo), las estaciones compresoras se componen de un número fijo de compresores centrífugos idénticos conectados en paralelo, y por simplicidad en las evaluaciones realizadas en esta tesis, se considera que cada estación compresora instalada en la red es operada por un solo compresor.

Como mencionamos al inicio de esta sección, el estudio de la estructura matemática de una estación compresora se basó en el trabajo de tesis doctoral realizado por Wu [47], en donde el dominio de operación factible $D^{unit}_{(i,j)}$ es dado por las siguientes desigualdades:

$$S_{ij}^L \leq s_{ij} \leq S_{ij}^U \quad (3.8)$$

$$R_{ij}^L \leq \frac{q_{ij}}{s_{ij}} \leq R_{ij}^U \quad (3.9)$$

donde a su vez h_{ij} , q_{ij} y s_{ij} se relacionan de la siguiente forma:

$$\frac{h_{ij}}{s_{ij}^2} = A_H + B_H \left(\frac{q_{ij}}{s_{ij}} \right) + C_H \left(\frac{q_{ij}}{s_{ij}} \right)^2 + D_H \left(\frac{q_{ij}}{s_{ij}} \right)^3 \quad (3.10)$$

Nótese que (3.8) y (3.9) implican:

$$Q_{ij}^L \leq q_{ij} \leq Q_{ij}^U \quad (3.11)$$

donde A_H , B_H , C_H , D_H son coeficientes que se asumen conocidos, los cuales dependen del tipo de compresor y se estiman típicamente por el método de mínimos cuadrados. S_{ij}^L y S_{ij}^U son parámetros conocidos que representan la velocidad mínima y máxima del compresor (i,j) , respectivamente. R_{ij}^L y R_{ij}^U representan la razón mínima y máxima de

q_{ij}^L / s_{ij}^L (*surge*) y q_{ij}^U / s_{ij}^U (*stonewall*) para el compresor (i,j), respectivamente. Donde $Q_{ij}^L = S_{ij}^L * R_{ij}^L$ y $Q_{ij}^U = S_{ij}^U * R_{ij}^U$, son parámetros que indican las cotas mínimas y máximas de la tasa de flujo volumétrico del compresor (i,j), respectivamente. Para cada Q_{ij} en este rango, h_{ij} es acotada inferiormente tanto por S_{ij}^L como por R_{ij}^U , y acotada superiormente por S_{ij}^U y R_{ij}^L .

Posteriormente, en un análisis realizado desde la perspectiva de la optimización en redes, se llegó a la determinación de que es preferible trabajar en un espacio de operación factible definido por las variables de decisión (x_{ij}, p_i, p_j) , ya que el flujo másico x_{ij} se conserva en cada nodo. Por lo tanto, la relación del dominio definido por (q_{ij}, h_{ij}, s_{ij}) y el dominio $D_{(ij)}^{unit}$ en el espacio (x_{ij}, p_i, p_j) se obtiene de un mapeo representado por las siguientes ecuaciones.

$$h_{ij} = \frac{ZRT_s}{m} \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right] \quad (3.12)$$

$$q_{ij} = ZRT_s \frac{x_{ij}}{p_i} \quad (3.13)$$

DIRECCIÓN GENERAL DE BIBLIOTECAS
donde los siguientes parámetros se asumen conocidos:

Parámetros

T_s	Temperatura del gas
Z	Factor de compresibilidad del gas
R	Constante del gas
M	Coefficiente específico de calor; $m = \frac{k-1}{k}$, siendo $k=1.297$

Finalmente, tenemos al dominio de operación factible $D_{(i,j)}^{unit}$ definido por:

$$D_{(i,j)}^{unit} = \{ (x_{ij}, p_i, p_j) : p_i^L \leq p_i \leq p_i^U, (3.10), (3.12) \text{ y } (3.13) \text{ se cumplen} \}$$

Gráfica de p_j vs x_{ij} y p_i

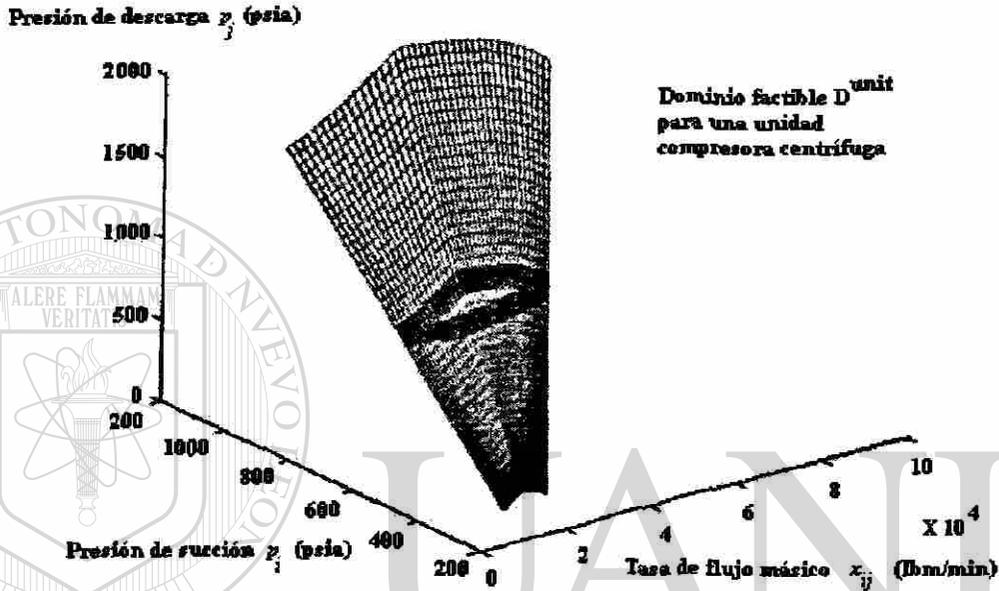


Figura 3.1: Dominio $D_{(i,j)}^{unit}$ de operación factible.

En la Figura 3.1 se muestra el dominio no convexo $D_{(i,j)}^{unit}$ de operación factible en función de las variables (x_{ij}, p_i, p_j) . Este comportamiento es común en los compresores centrífugos. Por ejemplo, al observar esta figura se identifica una franja oscura en el centro, la cual corresponde a la superficie del dominio cuando p_j es fijada en un valor dentro de los rangos factibles de operación. Esto puede verse más detalladamente en las Figuras 3.2 y 3.3, las cuales muestran el dominio de operación factible bajo el espacio (x_{ij}, p_i, p_j) para una x_{ij} y p_j fijos, respectivamente.

Para una explicación más detallada sobre la estructura matemática de las estaciones compresoras centrífugas ver Wu, Boyd y Scott [48] y Wu et al. [49].

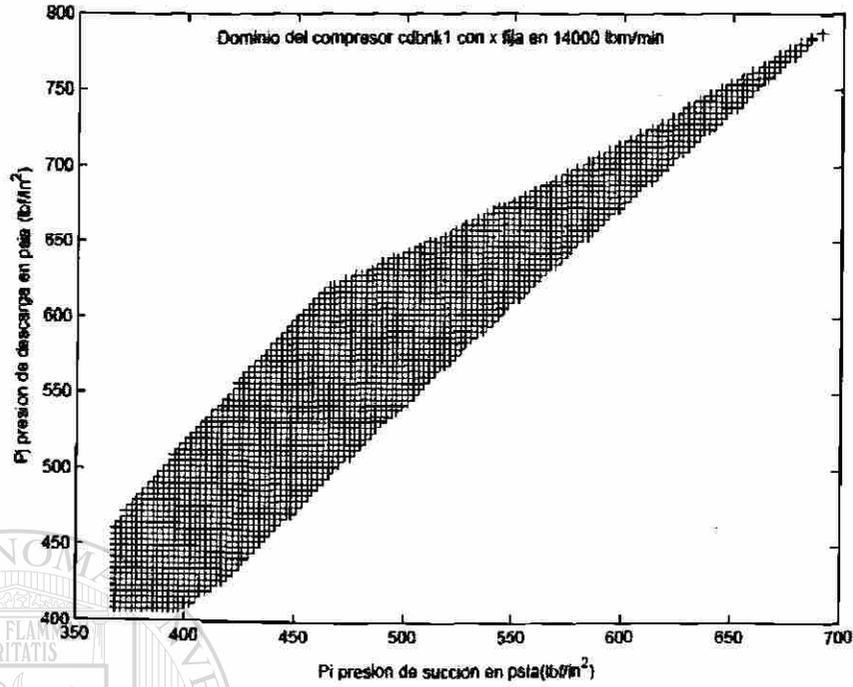


Figura 3.2: Gráfica del dominio $D^{unit}_{(i,j)}$ cuando x_{ij} está fija.

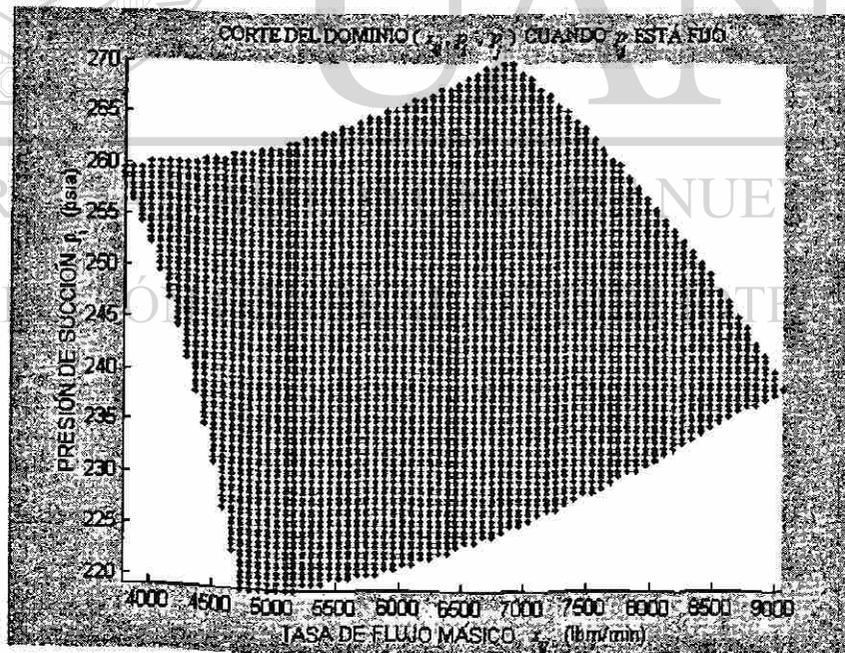


Figura 3.3: Superficie del dominio $D^{unit}_{(i,j)}$ cuando p_j está fijo.

La necesidad de definir este dominio factible de operación se respalda con la proyección de una optimización en la configuración de presión requerida en el proceso de transporte del gas natural a través del sistema de red de tuberías, ya que es inevitable para la compensación de la caída de presión debido al calor emergente entre la fricción del gas y las paredes internas de las tuberías.

3.3.1 Representación Algebraica del Dominio $D_{(i,j)}^{unit}$

Dado que desde la perspectiva de una implementación computacional, es preferible trabajar sobre un espacio factible de operación de una estación compresora definido por (x_{ij}, p_i, p_j) , y por supuesto, ya que en la sección anterior hemos descrito detalladamente el dominio factible en cuestión en función del dominio (q_{ij}, h_{ij}, s_{ij}) , en esta sección procedemos a desarrollar una representación algebraica equivalente al dominio $D_{(i,j)}^{unit}$ presentado por la expresión (3.6) del modelo matemático (NLP) exhibido en la Sección 3.2.

A continuación, se describe el desarrollo general de esta representación algebraica $D_{(i,j)}$ como un modelo basado en cuatro desigualdades, que corresponden a las cuatro desigualdades que definen el dominio de operación factible en el espacio (q_{ij}, h_{ij}, s_{ij}) dada por (3.8) y (3.9), las cuales son construidas esencialmente mediante operaciones matemáticas básicas. Este modelo define de manera equivalente el espacio factible de búsqueda (ver Figura 3.1) de una estación compresora representado por $D_{(i,j)}^{unit}$.

El modelo se desarrolla como sigue. Sea

$$\phi \left(\frac{Q}{S} \right) = A_H + B_H \left(\frac{Q}{S} \right) + C_H \left(\frac{Q}{S} \right)^2 + D_H \left(\frac{Q}{S} \right)^3,$$

$$\alpha = \beta / m, \text{ donde } \beta = ZRT_s$$

Por lo tanto, para acotar inferiormente el dominio de operación factible nos basamos en (3.8), (3.12) y (3.13), obteniendo de esta manera la **1ª Restricción** del modelo:

$$\alpha \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right] \geq (S_{ij}^L)^2 \phi \left(\frac{\beta \left(\frac{x_{ij}}{p_i} \right)}{S_{ij}^L} \right) \quad (3.14)$$

De manera similar, pero delimitando la parte superior del dominio factible tenemos a la **2ª Restricción** representada por:

$$\alpha \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right] \leq (S_{ij}^U)^2 \phi \left(\frac{\beta \left(\frac{x_{ij}}{p_i} \right)}{S_{ij}^U} \right) \quad (3.15)$$

Ahora bien, sea $\frac{Q_{ij}^L}{S_{ij}^L} = R_{ij}^L$, entonces $S_{ij}^L = \frac{Q_{ij}^L}{R_{ij}^L}$ (3.16a)

Basándonos en (3.8), (3.12), (3.13) y (3.16a) acotamos inferiormente, adyacente a (3.14), al dominio factible con la **3ª Restricción** representada por:

$$\alpha \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right] \geq (S_{ij}^L)^2 \phi (R_{ij}^L) \quad (3.16b)$$

Finalmente, sea $\frac{Q_{ij}^U}{S_{ij}^U} = R_{ij}^U$, entonces $S_{ij}^U = \frac{Q_{ij}^U}{R_{ij}^U}$ (3.17a)

Con un análisis similar a (3.16b), en base a (3.17a), la **4ª Restricción** es representada por:

$$\alpha \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right] \leq (S_{ij}^U)^2 \phi (R_{ij}^U) \quad (3.17b)$$

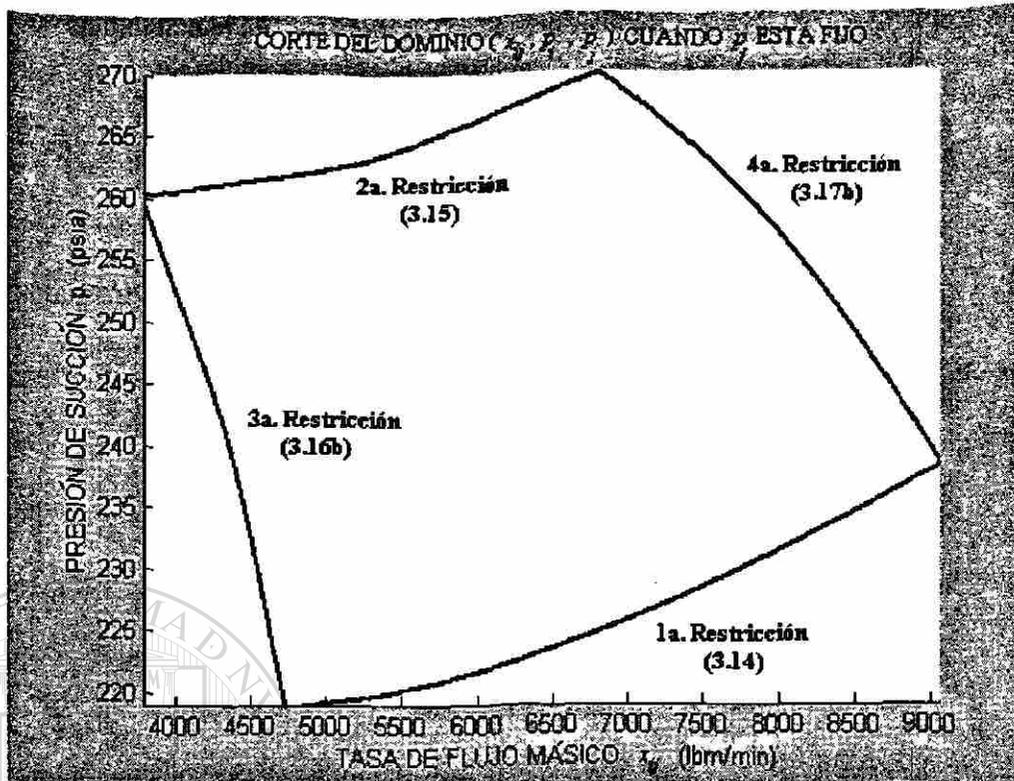


Figura 3.4: Frontera de operación factible D' definido por (3.14), (3.15), (3.16b) y (3.17b).

Por tanto, en el modelo matemático (3.1)-(3.7) de la Sección 3.2, la restricción (3.6), la cual representa el dominio D^{unit} de operación de una estación compresora, es reemplazada por el siguiente modelo algebraico D' :

$$D'_{(i,j)} = \left\{ (x_j, p_i, p_j) : p_i^L \leq p_i \leq p_i^U, (3.14), (3.15), (3.16b) \text{ y } (3.17b) \text{ se cumplen} \right\}$$

En la Figura 3.4 se muestra la frontera de operación factible definido por el modelo basado en las ecuaciones (3.14), (3.15), (3.16b) y (3.17b).

Con esta representación algebraica D' , equivalente al dominio factible D^{unit} , un punto (x_j, p_i, p_j) será factible siempre que satisfaga completamente con las ecuaciones establecidas en el modelo $D'_{(i,j)}$, $\forall (i,j) \in A_c$; por otro lado, un punto (x_j, p_i, p_j) será

considerado no factible, cuando éste se encuentre fuera de la frontera definida por el modelo D' .

3.4 Función de Costo para un Compresor Centrífugo

En esta sección describimos la función $g^{unit}_{(i,j)}$ presentada por el modelo matemático (NLP) de la Sección 3.2. Esta función representa el costo del consumo de combustible de una estación compresora centrífuga (i,j) , $\forall (i,j) \in A_c$.

Básicamente, la función de costo es el considerado por Wu [47], la cual puede definirse en base al espacio (x_{ij}, p_i, p_j) como sigue:

$$g^{unit}_{(i,j)}(x_{ij}, p_i, p_j) = \frac{c x_{ij} \left[\left(\frac{p_j}{p_i} \right)^m - 1 \right]}{\eta}, \quad \forall (x_{ij}, p_i, p_j) \in D^{unit} \quad (3.18)$$

con:

$$c = \frac{\alpha ZRT_s}{m}$$

donde α es una constante positiva, la cual por simplicidad en este trabajo se supone igual a 1, ya que el gas que se está manejando es de alta presión.

La eficiencia adiabática es representada por η , y está dada por la siguiente ecuación:

$$\eta = A_E + B_E \left(\frac{q_{ij}}{s_{ij}} \right) + C_E \left(\frac{q_{ij}}{s_{ij}} \right)^2 + D_E \left(\frac{q_{ij}}{s_{ij}} \right)^3$$

donde A_E , B_E , C_E y D_E son coeficientes que dependen del tipo de compresor y son típicamente estimados por el método de mínimos cuadrados.

La Ecuación (3.18) del consumo de combustible, formula la potencia necesaria para transportar cierta cantidad de flujo másico (x_{ij}) a través del arco compresor (i,j). El desempeño de esta función depende de las características de la unidad compresora. Sin embargo, es típico observar que la función de costo $g^{unit}_{(i,j)}$ del consumo de combustible se incrementa en relación a una razón de (p_j / p_i) y a la tasa del flujo volumétrico Q_{ij} o (x_{ij} / p_i) , y disminuye con respecto a la presión de succión p_i .

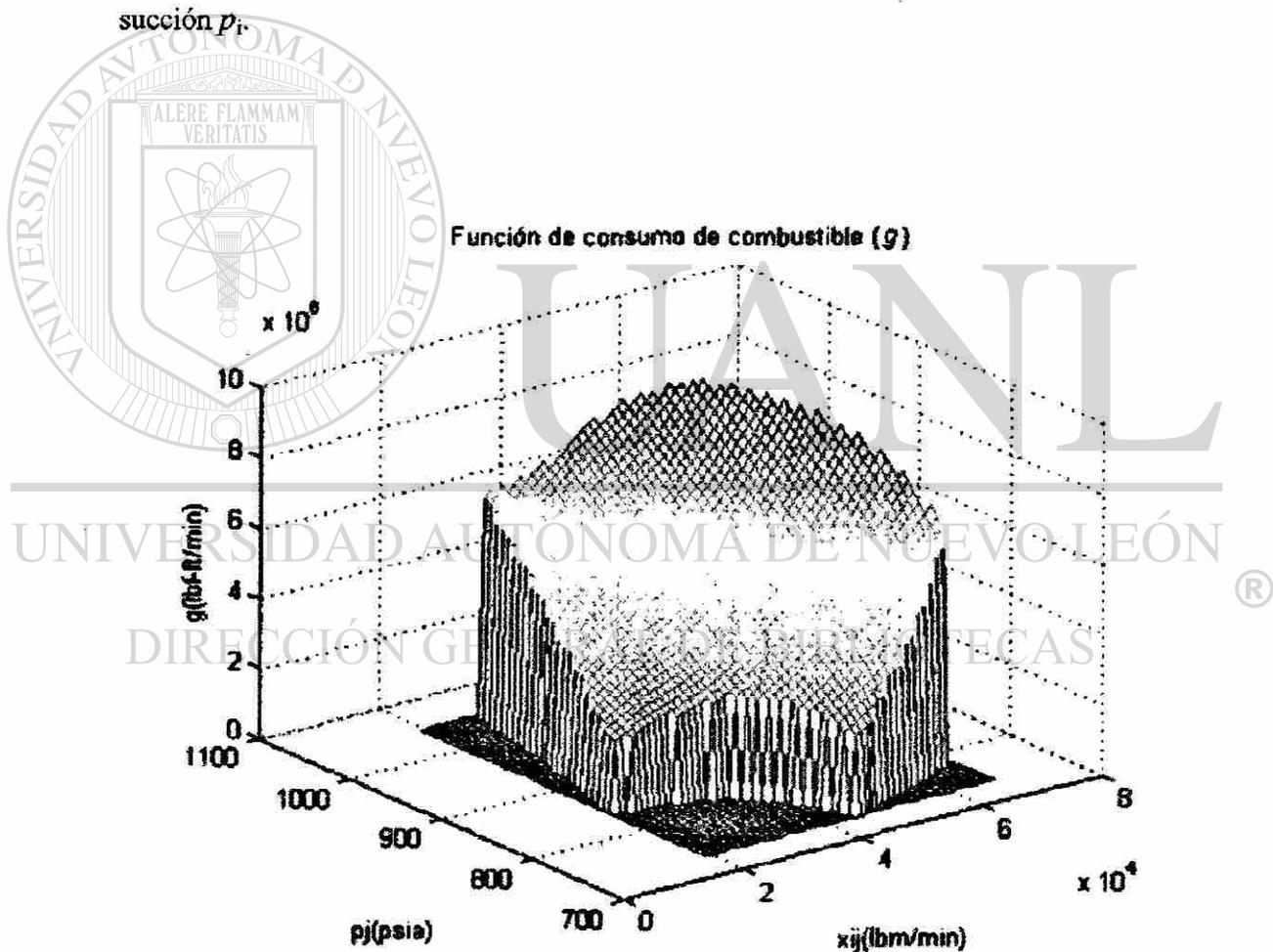


Figura 3.5: Gráfica de la función del consumo de combustible cuando p_i está fijo.

En la Figura 3.5 podemos observar la gráfica de la función del consumo de combustible para una estación compresora en el espacio (x_i, p_i, p_j) con la variable de presión de succión (p_i) fija.

Resumiendo, en este capítulo hemos presentado un modelo de equivalencia operacional del dominio de operación factible de una estación compresora, el cual proponemos como promesa de un mejor desempeño del esfuerzo computacional en la implementación de las técnicas a desarrollar. Además, la definición de una función de costo no convexa para medir el consumo de combustible de las estaciones compresoras ha sido presentada; por lo tanto, en el siguiente capítulo proponemos y describimos un método de solución para el PMCC, el cual aplica técnicas no tradicionales de programación no lineal para contrarrestar los problemas inherentes que el modelo matemático presenta, como son: la no linealidad y no convexidad.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

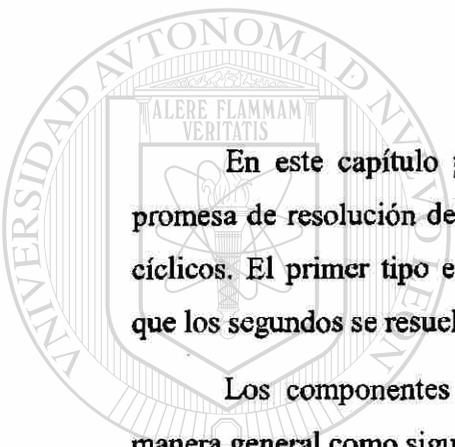
DIRECCIÓN GENERAL DE BIBLIOTECAS

CAPÍTULO 4

MÉTODO DE SOLUCIÓN:

PREPROCESAMIENTO Y PROGRAMACIÓN

DINÁMICA



En este capítulo presentamos un método de solución que proponemos como promesa de resolución del PMCC sobre dos tipos de topologías de redes: no cíclicos y cíclicos. El primer tipo es resuelto encontrando soluciones óptimas globales, mientras que los segundos se resuelven al obtener soluciones aproximadas de buena calidad.

Los componentes básicos de este método de solución pueden describirse de manera general como sigue:

(a) *Preprocesamiento*: Este componente tiene dos objetivos fundamentales:

1) Refinar las cotas del dominio de operación factible dado las variables de decisión (X_{ij} y P_i); y, 2) reducir el tamaño de la instancia de red [41] sin alterar su estructura matemática. Con esta fase, uno espera que las técnicas implementadas se desempeñen de una manera más eficaz sobre los límites de operación; explotando así, de una manera eficiente, la estructura matemática del problema.

(b) *Asignación de flujos*: En esta fase, un conjunto de flujos factibles iniciales es obtenido, de manera eficiente, basándonos en dos técnicas no tradicionales: una técnica de asignación “clásica” y un algoritmo de Grafo Reducido que emplea una técnica de reducción [41]. Esta asignación se basa en un patrón

que presentan las instancias de red.

(c) *Solución óptima de las variables de presión:* En esta fase, un conjunto de presiones óptimas (para el conjunto de flujos factibles obtenido en la fase anterior) es encontrado aplicando una técnica de DP no secuencial. Cabe hacer mención que, bajo los fundamentos teóricos encontrados en [41] (donde se demuestra que en topologías no cíclicas existe una única manera de asignar los flujos), en esta fase obtenemos soluciones óptimas globales para topologías no cíclicas en un dominio de operación factible discretizado. Pero, dado el hecho de que en topologías cíclicas no es posible encontrar una única forma de asignar los flujos, en esta fase, para los sistemas cíclicos encontramos puntos iniciales factibles (por cierto muy buenos) para dar seguimiento al cuarto componente que implementará un procedimiento heurístico, el cual utiliza este punto factible como fase inicial en su ejecución.

(d) *Procedimiento heurístico:* En esta fase, la cual desarrollamos y describimos con detalle en el siguiente capítulo, nosotros proponemos la implementación de un procedimiento heurístico de TS para modificar los flujos sobre las topologías cíclicas (recordemos que no pueden ser asignados de manera única). Esta heurística, como parte innovadora, emplea una técnica de DP no secuencial para obtener el conjunto de presiones óptimas. De esta manera, esta fase refleja la optimización tanto en las variables del flujo como en las variables de presión, finalizando así con la metodología de solución.

Los componentes (b) y (c) se vinculan en el algoritmo NDP descrito en la Sección 4.2, como método de búsqueda de soluciones que engloba la asignación de flujos y la búsqueda del conjunto óptimo de presiones en una relación de dependencia.

La técnica de reducción de red a la que hacemos alusión en la fase de preprocesamiento del componente (a), es utilizada como fase inicial por el componente (b) cuando describe el algoritmo de Grafo Reducido para asignar flujos factibles sobre redes cíclicas.

Ahora bien, en nuestro primer intento por asignar flujos factibles sobre la red, identificamos de que aparte de considerar los límites de operación impuestos por el modelo matemático (ver Sección 3.2), existía la posibilidad de refinar las cotas de las variables de decisión x_{ij} y p_i , refinando con esto el espacio factible de búsqueda. Esta consideración fue establecida debido a que, las restricciones físicas del desempeño operacional de una estación compresora no habían sido contempladas en conjunción con el desempeño general de las tuberías de red y las variables de decisión. Esto nos motivó para refinar el modelo matemático, y por supuesto, el dominio de operación factible; creando una fase con el objetivo de desarrollar y aplicar técnicas de preprocesamiento, como por ejemplo: derivación de cotas para x_{ij} y p_i , y reducción de la instancia de red sin alterar su estructura matemática. Esta fase de preprocesamiento es descrita a continuación.

4.1 Refinamiento del Modelo

En esta sección presentamos las técnicas de preprocesamiento empleadas con el fin de esperar encontrar las soluciones de una manera más eficiente.

Las dos primeras técnicas de derivación de cotas para x_{ij} y p_i , las cuales son presentadas en las Secciones 4.1.1 y 4.1.2, respectivamente, pueden ser definidas como operaciones matemáticas básicas que refinan las cotas de las variables de decisión (x_{ij} y p_i) en el dominio de operación factible D^{unit} descrito por la expresión (3.6), para conducir a mejores propiedades algorítmicas en el modelo matemático antes de intentar resolverlo.

La última etapa de la fase de preprocesamiento es presentada en la Sección 4.1.3, donde aplicamos una técnica de reducción propuesta por Ríos-Mercado et al. [41], la cual emplea el manejo de estructuras para conseguir disminuir el tamaño de una red sin alterar su estructura matemática.

4.1.1 Derivación de Cotas para x_{ij}

Las cotas se calculan en base a la relación de las ecuaciones (3.5)-(3.7) y (3.13), y el tipo de compresor instalado; esto es, para reducir la región factible o espacio de búsqueda, previniendo que el algoritmo examine dominios no factibles en cada arco compresor.

$$x_{ij}^L = \frac{p_i^L q_{ij}^L}{ZRT_s}, \quad x_{ij}^U = \frac{p_i^U q_{ij}^U}{ZRT_s} \quad (4.19)$$

Sean x_{ij}^L y x_{ij}^U las cotas inferior y superior del flujo másico del arco compresor (i,j), respectivamente. Así, por las ecuaciones (4.19) entonces la siguiente expresión limita al flujo másico x_{ij} de un arco que pertenece al compresor (i,j):

$$x_{ij}^L \leq x_{ij} \leq x_{ij}^U \quad \forall (i,j) \in A_c$$

4.1.2 Derivación de Cotas para p_i

Para la definición de los límites inferior y superior de las presiones en cada nodo, nosotros buscamos asignar los mejores valores iniciales seleccionándolos dentro de los límites de presiones en cada nodo. Calculamos las cotas de presión inferior y superior uniendo a las ecuaciones (3.8)-(3.10) y (3.12). Es posible reducir los límites de presiones como sigue.

Sea

$$\Phi\left(\frac{q_{ij}}{s_{ij}}\right) = A_H + B_H\left(\frac{q_{ij}}{s_{ij}}\right) + C_H\left(\frac{q_{ij}}{s_{ij}}\right)^2 + D_H\left(\frac{q_{ij}}{s_{ij}}\right)^3 \quad (4.20)$$

Por (3.9), (3.10) y (4.20), se sigue que la cabeza adiabática h_{ij} debe satisfacer a

las siguientes ecuaciones de igualdad:

$$h_{ij}^L = (s_{ij}^L)^2 \Phi(R_{ij}^U), \quad h_{ij}^U = (s_{ij}^U)^2 \Phi(R_{ij}^L) \quad (4.21)$$

Sean h_{ij}^L y h_{ij}^U las cotas inferior y superior de la cabeza adiabática, respectivamente. Basándonos en las ecuaciones (3.5), (3.12), (4.20) y (4.21) obtenemos:

$$p_{i2}^L = p_j^L \left[\frac{(m h_{ij}^U)}{ZRT_s} + 1 \right]^{-\left(\frac{1}{m}\right)} \quad (4.22)$$

y

$$p_{i2}^U = p_j^U \left[\frac{(m h_{ij}^L)}{ZRT_s} + 1 \right]^{-\left(\frac{1}{m}\right)} \quad (4.23)$$

Así, en base a las ecuaciones (4.22) y (4.23) tenemos:

$$\max \{ p_i^L, p_{i2}^L \} \leq p_i \leq \min \{ p_i^U, p_{i2}^U \}.$$

4.1.3 Técnica de Reducción

Como se ha mencionado anteriormente, el PMCC está clasificado dentro del tipo de problemas conocido como *NP-completo* (ver [13] y [32]). Uno de los principales obstáculos de este problema es la dimensionalidad de la red, ya que una red típicamente puede consistir en cientos de tuberías y docenas de compresores, por lo que el empleo de una técnica de reducción sobre la red, sin alterar su estructura matemática, es justificable.

La técnica de reducción está basada en el supuesto de que, si todos los recursos de flujos máscicos sobre cada nodo de la red son conocidos con anterioridad, es decir, si existe un balance de flujo máscico sobre cada nodo de la red, entonces todos los flujos en las tuberías pueden ser completamente determinados. En el caso de los sistemas no

cíclicos, los flujos se determinan de forma única; no así para las redes cíclicas, donde los flujos se establecen en base a los rangos de operación factibles, permitidos por las propiedades de las estaciones compresoras instaladas en la red. Estos rangos son refinados por las técnicas de preprocesamiento.

Descripción de la Técnica de Reducción

La técnica de reducción fue propuesta por Ríos-Mercado et al. [41]. Esta es una técnica que emplea el manejo de estructuras para conseguir disminuir el tamaño de una red sin alterar su estructura matemática.

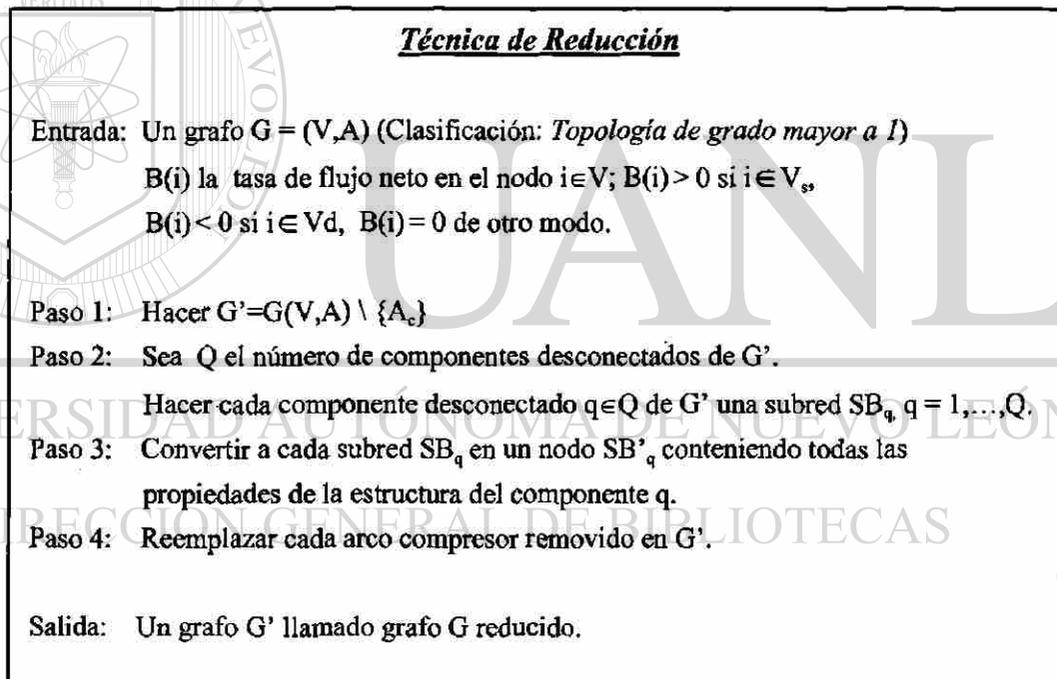


Figura 4.1: Algoritmo de la Técnica de Reducción aplicada a redes de gas natural.

En la Figura 4.1 se describe el algoritmo de la técnica de reducción, el cual como dato de entrada necesita un grafo G de una red de transporte de gas natural, clasificado como topología de grado mayor a 1 (ver Figura 4.4). Como primer paso se construye un

grafo no conexo G' al remover todos los arcos compresores; así, en el paso 2, se considera tener Q componentes desconectados llamados $SB_q, \forall q=1, \dots, Q$. En el paso 3, cada componente desconectado se convierte en un nodo SB'_q , comprimiendo todas las propiedades físicas de la estructura del componente $SB_q, \forall q=1, \dots, Q$. Ahora, como siguiente paso regresamos cada arco compresor removido uniendo cada nodo desconectado SB'_q . Así, finalmente tenemos un grafo G' llamado grafo reducido de G , el cual se construye mediante una técnica de reducción. Los detalles de esta técnica pueden ser encontrados en [41]. La Complejidad Computacional del procedimiento es $O(|A|)$. Véase Apéndice D.

4.2 Descripción del Algoritmo NDP

Partiendo del estudio de la estructura matemática del PMCC, desarrollamos un método de búsqueda que introduce técnicas no tradicionales que proporcionan excelentes propiedades para enfrentar las dificultades numéricas inherentes al problema.

En esta sección, nosotros proponemos un procedimiento de búsqueda, que hace referencia al algoritmo NDP (por su siglas en inglés, *Non-sequential Dynamic Programming*, ver Figura 4.2) para resolver el PMCC. El algoritmo propuesto obedece a dos objetivos primales. El primero de ellos es encontrar soluciones, sean estos óptimos (en el caso de redes no cíclicas) o factibles (para redes cíclicas); y el segundo, que sea de una manera rápida y eficiente.

El algoritmo de búsqueda NDP, presentado en la Figura 4.2, consiste básicamente de dos fases secuenciales. En la Fase 1, un conjunto de flujos factibles es obtenido por el empleo de dos técnicas: una técnica de asignación “clásica” y un algoritmo de Grafo Reducido que emplea una técnica de reducción. En la Fase 2, un conjunto de presiones óptimas es obtenido (para el conjunto de flujos factibles conseguido en la Fase 1) por una técnica de DP no secuencial, motivados por el trabajo

propuesto por Richard Carter. Estos procedimientos hacen uso *a priori* de las técnicas de preprocesamiento para refinar las cotas iniciales sobre las dos variables de decisión continuas, esperando de esta forma obtener soluciones de una manera más eficiente.

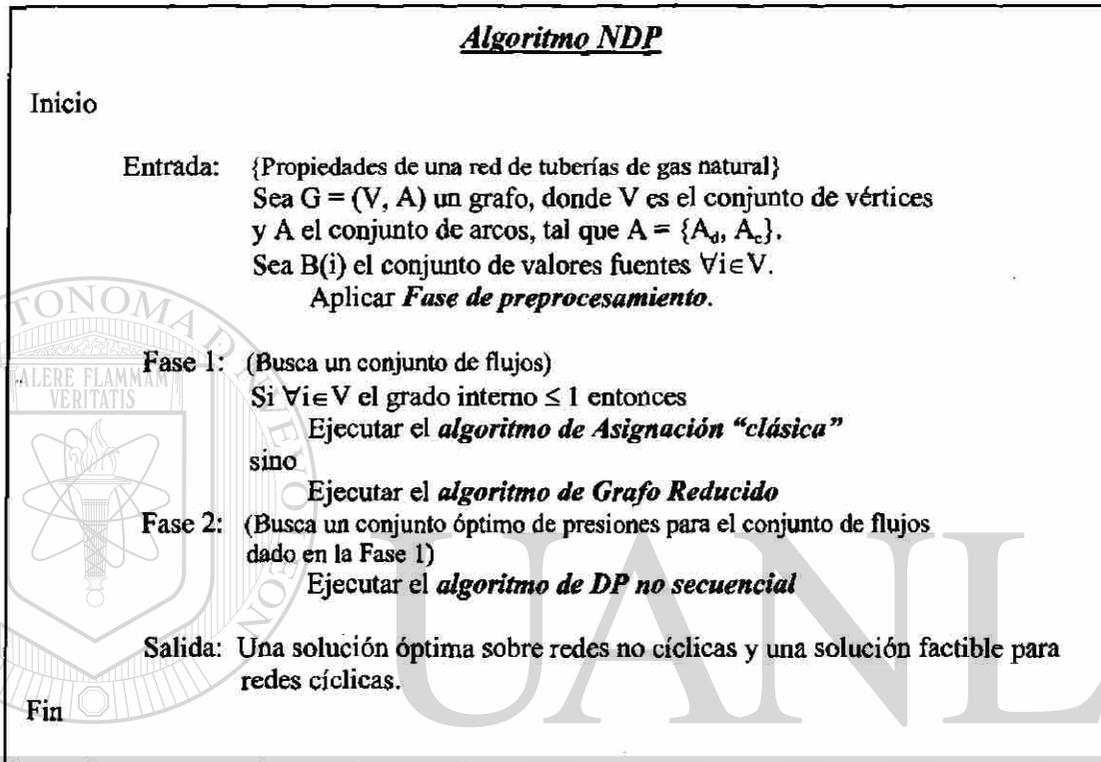


Figura 4.2: Algoritmo NDP para el PMCC.

DIRECCIÓN GENERAL DE BIBLIOTECAS

4.2.1 Fase 1: Procedimiento de Generación de Flujos

En un análisis preliminar sobre el estudio de las técnicas propuestas para la búsqueda de un conjunto factible de flujos, se identificó que en ciertos patrones de topologías de redes, el desempeño que presentaban dos de ellas era inmejorable; por lo que, en un estudio posterior se designaron a cada técnica ciertas topologías que presentaban el patrón identificado.

La reclasificación de los tipos de topologías a la que hacemos referencia, es

absolutamente exclusiva para la ejecución de las técnicas apropiadas en la búsqueda del conjunto de flujos factibles, siendo esta justificable cuando se vincula la eficiencia de los recursos computacionales con la calidad inmensurable de una solución esperada. La reclasificación puede realizarse como sigue:

- ❖ *Topologías de grado 1* (Figura 4.3): Son todas aquellas topologías no cíclicas que dentro de su estructura se componen exclusivamente de nodos con grado interno menor o igual a 1.

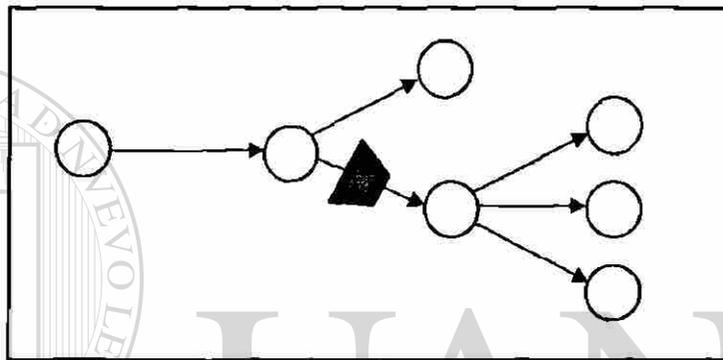


Figura 4.3: Reclasificación: Topología de grado 1.

- ❖ *Topologías de grado mayor a 1* (Figura 4.4): En esta reclasificación se incluyen todas las topologías cíclicas y no cíclicas en donde, al menos, uno de los nodos que la componen tiene un grado interno mayor a uno.

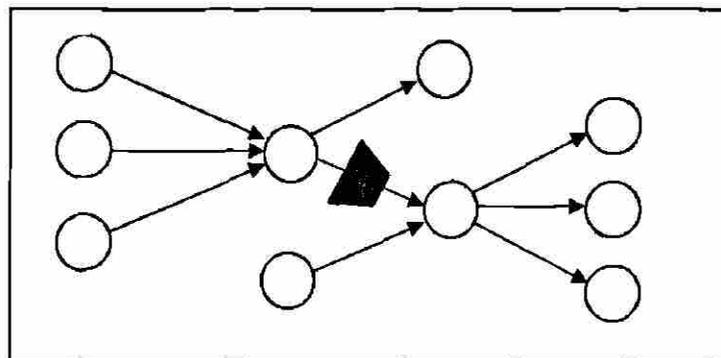


Figura 4.4: Reclasificación: Topología de grado mayor a 1.

El procedimiento de asignación de x_{ij} de la Fase 1 (ver Figura 4.2), está basado en dos algoritmos. El primero de ellos, un algoritmo de Asignación “clásica”, es aplicado únicamente sobre instancias topológicas de grado 1 (ver Figura 4.3) para asignar flujos factibles a cada arco de la red. En contraparte, para asignar flujos factibles sobre topologías de grado mayor a 1 (ver Figura 4.4), un algoritmo de Grafo Reducido es aplicado.

Técnica de Asignación “Clásica”

En esta sección describimos el algoritmo de asignación “clásica” (Figura 4.5) de la Fase 1 del algoritmo NDP (ver Figura 4.2) para la búsqueda de un conjunto de flujos óptimos sobre las instancias *topológicas de grado 1*, referidas en la reclasificación (ver Figura 4.3). La demarcación de “clásica” corresponde a la implementación de una técnica de asignación de flujo sobre cada arco dentro de rutas construidas en base a la topología. El algoritmo inicia asignando flujos sobre cada arco de una ruta, partiendo desde el último arco hacia el primero de la misma, es decir, hacia atrás.

Como fase inicial de este proceso de asignación observamos que una *topología de grado 1* (Figura 4.3) puede ser formada únicamente por topologías lineales o tipo árbol; por lo que, es posible definir al problema como asignación de flujos sobre un sistema de rutas finitas.

En este proceso de asignación, una iteración es ejecutada para cada ruta de la instancia de red, produciendo hasta antes de encontrar un rama divergente, una asignación óptima de flujos; esto, debido a que el flujo en cada arco puede ser determinado de forma única [41]. La iteración es establecida por una transición de asignación de flujos sobre la ruta hasta encontrar un nodo con grado interno menor a 1.

La matriz de adyacencia es quizás el concepto más importante en este modelo de asignación “clásica”, ya que representa la liga entre cada ruta del sistema, de tal manera que cuando cada asignación se realiza por separado, ésta resulta automáticamente

factible para el problema completo.

Conceptualmente, desde el punto de vista del cálculo, el método hacia atrás es más eficiente; esto se debe a la complejidad de hacer transformaciones sobre las ramas divergentes que se procesan en dos iteraciones consecutivas.

En la Figura 4.5 se presenta el algoritmo de Asignación “clásica” para asignar flujos óptimos sobre topologías de grado 1.

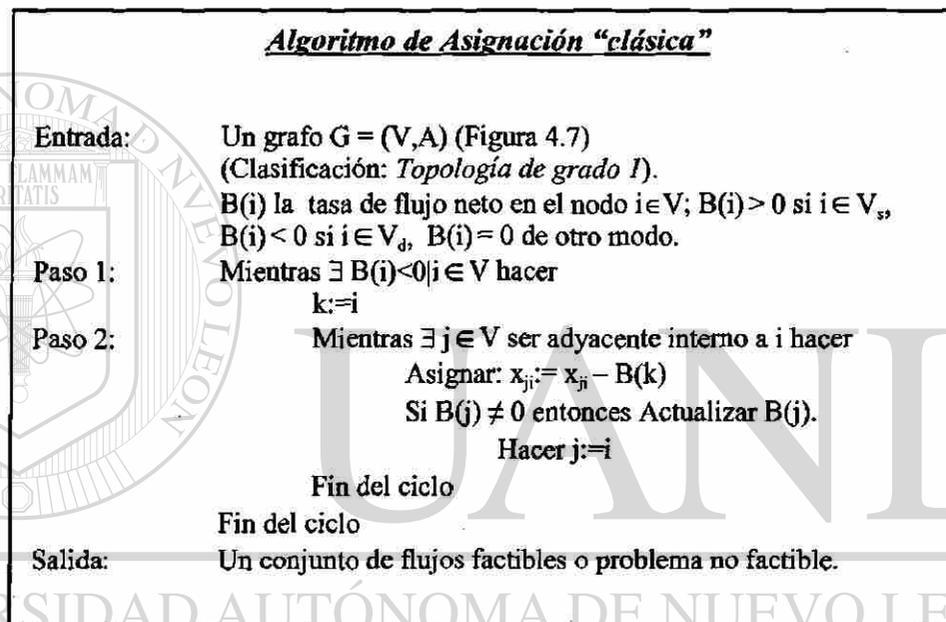


Figura 4.5: Algoritmo de Asignación “clásica”.

Este algoritmo de asignación “clásica” hace uso de las técnicas de preprocesamiento (ver Sección 4.1) para refinar el rango de factibilidad sobre cada arco compresor; esto con la finalidad de evitar la asignación de flujos máxicos no factibles.

A continuación presentamos el desarrollo de la ejecución del algoritmo sobre la instancia de red *net-b-12c3* (topología de grado 1, ver Figura 4.3) que pertenece a la base de datos, la cual es resuelta ejecutando tres simples iteraciones.

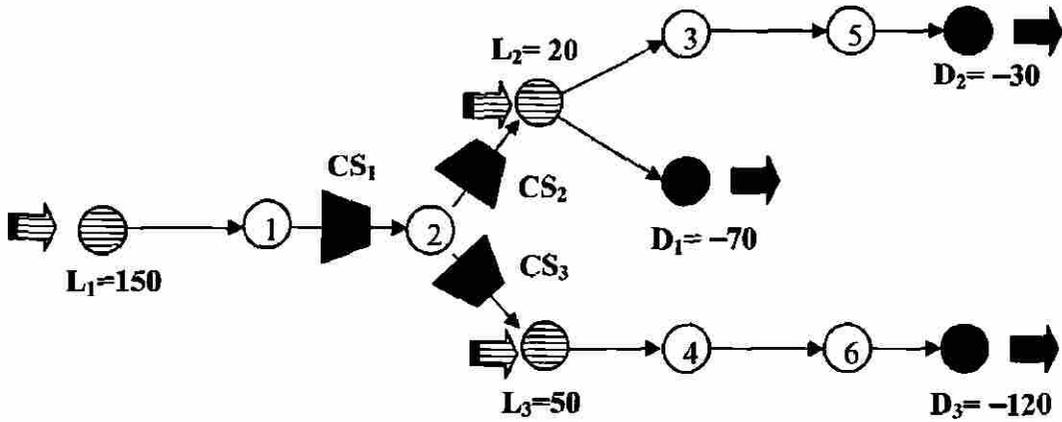


Figura 4.6: Grafo clasificado como topología de grado 1.

Primero observamos el grafo G mostrado en la Figura 4.6, y así procedemos a ejecutar la 1ª iteración tomando un nodo demanda al azar sobre la red. En este ejemplo, tomamos el nodo D_2 y proseguimos con la asignación del flujo másico demandado por D_2 sobre cada arco interno adyacente hasta encontrar el primer nodo sobre la red (en este ejemplo, el punto final es el nodo L_1 , ver Figura 4.7).

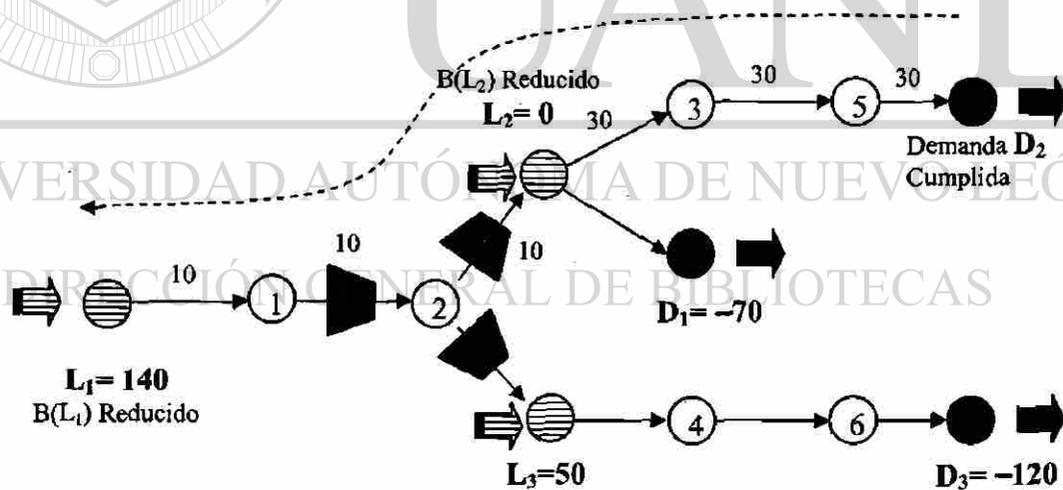


Figura 4.7: 1ª iteración del algoritmo de Asignación "clásica".

Procediendo de la misma manera que en la 1ª iteración, en las siguientes dos iteraciones se toman como puntos de referencia los nodos demanda D_1 y D_3 , respectivamente; y asignamos el flujo demandado sobre estos nodos como se muestra en

las Figuras 4.8 y 4.9, para cada nodo demanda respectivamente.

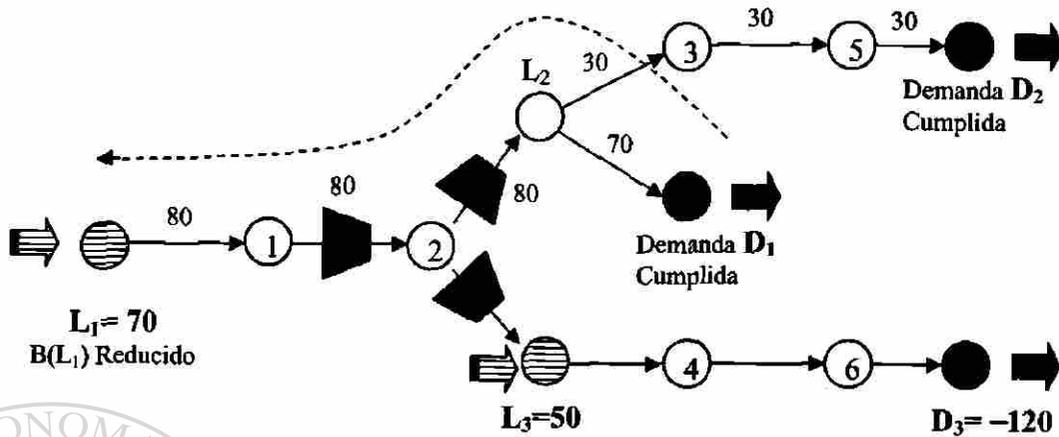


Figura 4.8: 2ª iteración del algoritmo de Asignación "clásica".

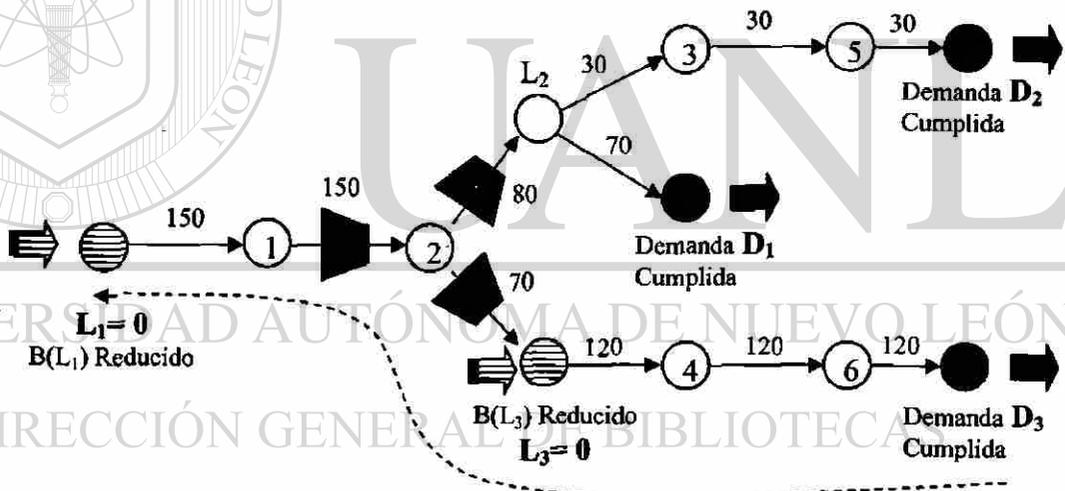


Figura 4.9: 3ª iteración del algoritmo de Asignación "clásica".

Obteniendo como salida para esta instancia de red, el conjunto de flujos factibles mostrado en la Tabla 4.1.

Es pertinente aclarar que el algoritmo de Asignación "clásica" encuentra siempre un conjunto de flujos factibles para las topologías de grado 1, que cumple con cada una

de las restricciones del modelo de programación matemática; o bien, si fuese el caso, entrega un mensaje de infactibilidad, terminando de esta forma con el procedimiento. El algoritmo tiene complejidad computacional $O(|B|)$, donde $B = \sum_{i \in V_d} B_i$, aunque en la práctica se comporta mucho mejor que esta cota. Véase Apéndice D.

Conjunto de flujos factibles sobre la topología net-b-12c3-C4	
$X(L_1,1) = 150$	$X(L_3,4) = 120$
$X(1,2) = 150$	$X(3,5) = 30$
$X(2,L_2) = 80$	$X(5,D_2) = 30$
$X(2,L_3) = 70$	$X(4,6) = 120$
$X(L_2,3) = 30$	$X(6,D_3) = 120$
$X(L_2,D_1) = 70$	

Tabla 4.1: Conjunto de flujos factibles obtenido por el algoritmo de Asignación "clásica" en 3 iteraciones.

Descripción del Algoritmo de Grafo Reducido

En esta sección, describimos detalladamente el algoritmo de Grafo Reducido aplicado en la Fase 1 del algoritmo NDP (ver Figura 4.2) sobre topologías de grado mayor a 1 (ver Figura 4.4). Al igual que la técnica de Asignación "clásica" anteriormente descrita (ejecutada sobre topologías de grado 1), este algoritmo de Grafo Reducido también hace uso de las técnicas de preprocesamiento que se describen en la Sección 4.1 para refinar las cotas de las variables de flujo; además, este algoritmo particularmente emplea una técnica de reducción para reducir el tamaño de la instancia de red, sin alterar su estructura matemática; procediendo así, de manera más eficiente, con el proceso de búsqueda de un conjunto de flujos factibles.

En la Figura 4.10 presentamos el pseudo-código del algoritmo de Grafo Reducido propuesto. En el Paso 1 se emplea una técnica de reducción para crear la red reducida G' del grafo original G ; se prosigue trabajando sobre la red reducida G' del Paso 2 al Paso 6, encontrado flujos factibles sobre cada arco compresor y arcos ductos de las subredes que constituyen la red reducida G' . En el paso 7 se reemplazan los flujos

factibles asignados en la red reducida G' al grafo original G ; obteniendo como salida un conjunto de flujos factibles o un mensaje de infactibilidad, terminando así la fase de asignación de flujos. El algoritmo tiene complejidad computacional $O(|A|)$. Véase Apéndice D.

Algoritmo de Grafo Reducido

Entrada: Un grafo $G = (V, A)$ (Clasificación: *Topología de grado mayor a 1*)
 $B(i)$ la tasa de flujo neto en el nodo $i \in V$; $B(i) > 0$ si $i \in V_s$,
 $B(i) < 0$ si $i \in V_d$, $B(i) = 0$ de otro modo.

Paso 1: Crear la red reducida G' .
Paso 2: Asignar flujo a cada arco que contenga un nodo con grado 1 en G' .
 Actualizar los valores fuente de los nodos que unen al arco.
 Eliminar el arco de G' .
Paso 3: Tomar un nodo r con grado externo 0 de G' .
Paso 4: Obtener las cotas para $x_{(i,r)} \forall i \in G'$.
Paso 5: Asignar un flujo a cada arco interno de r de la siguiente forma:
 Sea M el número de arcos internos de r , y

$$\alpha_m = \frac{(X_{(i,r)}^L + X_{(i,r)}^U)}{2}, \forall i \in G', m=1, \dots, M, \text{ y } \beta = (\alpha_1 + \alpha_2 + \dots + \alpha_M)$$

entonces, $x_{(i,r)} = \left(\frac{X_r^T}{\beta} \right) * B(r)$

Eliminar cada arco interno de r en G' .
Paso 6: Regresar al Paso 2 mientras $\exists x_{(i,j)}, \forall (i,j) \in G'$ sin flujo asignado.
Paso 7: Asignar flujos a cada Subred de G' .
Paso 8: Reemplazar los flujos de las subredes y de G' en la red original G .

Salida: Un conjunto de flujos factibles para el grafo G , o mensaje de infactibilidad.

Figura 4.10: Pseudo-código del Algoritmo de Grafo Reducido.

4.2.2 Fase 2: Procedimiento de Búsqueda de Presiones Óptimas

En esta sección, de manera similar a la sección anterior, describimos la implementación de la Fase 2 del algoritmo NDP (ver Figura 4.2), que introduce el

empleo de una técnica de DP no secuencial (ver Figura 4.14) para encontrar un conjunto de presiones óptimas (para el conjunto de flujos factibles obtenido en la Fase 1). En este procedimiento de asignación de p_i , al igual que en la Fase 1, también se aplican las técnicas de preprocesamiento para refinar las cotas de operación factible de las estaciones compresoras, y esperar con esto, un proceso de búsqueda más eficiente.

Técnica de Programación Dinámica No Secuencial

Para ejecutar la Fase 2 del algoritmo NDP se aplica una técnica de DP no secuencial, la cual fue originalmente propuesta por Richard Carter [4] en 1998. Esta técnica se implementa como una propuesta de encontrar presiones óptimas en cada nodo de la red (para el conjunto de flujos factibles obtenidos en la Fase 1).

Básicamente el proceso de esta técnica de DP no secuencial (ver Figura 4.14) consiste en observar el sistema de redes y centrar nuestro análisis en dos compresores conectados, reemplazándolos por un solo elemento “virtual” que representa la configuración de operación óptima de ambos compresores. Es pertinente aclarar que estos dos compresores conectados a combinarse pueden ser seleccionados de cualquier forma en el sistema, por lo que la filosofía de *recursividad* de la DP clásica adquiere un matiz no secuencial (para una referencia más detallada de los fundamentos de la DP no secuencial ver [2]). Este proceso de combinación continúa ejecutándose, reduciendo el número de elementos a combinar, uniendo dos a la vez hasta que el sistema no puede reducirse más; esto sucede cuando ha quedado exactamente un único elemento virtual, el cual caracteriza íntegramente el desempeño óptimo del sistema completo de red. Concluyendo así, que el costo óptimo incurrido en la configuración de operación sobre todas las estaciones compresoras de la red, es el mínimo valor de la última tabla “virtual” creada con esta técnica DP no secuencial.

Motivación

Partiendo del objetivo general de la optimización, el cual es encontrar soluciones de calidad a un problema dado bajo cierto conjunto de restricciones, la DP hace uso del mismo objetivo, como un método que propone soluciones a estos problemas que no

admiten una única solución.

De todas las técnicas de IO, la programación dinámica es una técnica que emplea conceptos simples; sin embargo, desde una perspectiva computacional, es considerada muy difícil de aplicar, ya que la implementación de apropiadas estructuras de datos debe ser realizada eficientemente para permitir la reconstrucción de las configuraciones de presiones óptimas actuales al sistema original; esto, una vez que el objetivo óptimo ha sido alcanzado.

Una de las ventajas de la aplicación de técnicas de DP encontramos que se propone como la más apropiada para resolver problemas que requieren decisiones interrelacionadas, es decir, decisiones que se deben tomar en forma secuencial, y las cuales determinan las decisiones futuras. Otra ventaja presente en la DP es que se considera una técnica matemática utilizada principalmente para mejorar la eficiencia de cómputo en ciertos problemas de optimización. La idea básica de la técnica es descomponer el problema en subproblemas, los cuales son computacionalmente más manejables.

Esta es una técnica que puede ser aplicada a muchos problemas de optimización, donde normalmente las soluciones son obtenidas en un proceso en reversa, es decir, desde la última etapa de un problema hacia la primera; por lo que un problema grande y complejo se convierte en una serie de problemas más pequeños y tratables.

Particularmente en el PMCC, observamos que la aplicación de una técnica de DP no secuencial tiene como ventaja principal contrarrestar la no convexidad que este problema presenta, la cual se muestra como la mayor dificultad en la aplicación de técnicas tradicionales de optimización no lineal.

Por supuesto, existen algunas dificultades de cálculo asociadas a esta técnica, las cuales son:

- A cada etapa combinada del proceso corresponde una política de decisión para el problema completo, y por tanto, el procedimiento típicamente requiere enumerar todas las combinaciones posibles en una etapa.
- Así también, las combinaciones infactibles en cada etapa no se eliminan por adelantado.

Por consiguiente, para evitar estas dificultades, se hacen las siguientes observaciones fundamentales:

- El problema se descompone en subproblemas llamados etapas, y cada etapa se optimiza solo sobre las alternativas posibles, de manera que nunca sea necesario enumerar todas las combinaciones anticipadamente.
- Así también, debido a que la optimización es aplicada en cada subproblema (etapa), todas las combinaciones no factibles se descartan sistemáticamente.
- Los subproblemas están ligados de forma especial de manera que nunca es posible optimizar sobre combinaciones infactibles.

Así, la formulación del problema de DP puede traducirse en tres elementos básicos.

1. Etapa.
2. Alternativas, variable de decisión en cada etapa y su función de rendimiento asociada.
3. Estado del sistema. Cada etapa representa una porción del problema para el cual se debe tomar una decisión.

El estado del sistema es quizás el concepto más importante en un modelo de DP, ya que representa la liga entre dos etapas subsecuentes, de tal manera que cuando cada etapa se optimiza por separado, la decisión resultante es automáticamente factible para el problema completo.

Implementación

Consideramos una red de transmisión de gas natural en estado estable con N estaciones compresoras y un conjunto de flujos máxicos factibles. Se sabe con certeza que cada estación compresora puede tener una configuración de presión de succión p_i y

presión de descarga p_j dentro de un rango de operación factible infinito. Al contrario de considerar cualquier presión entre los límites inferior y superior, consideraremos solo un conjunto discretizado en Δp unidades de presión. Por ejemplo, si el rango factible de operación está entre 600 y 800 unidades PSIG, y tenemos una discretización de $\Delta p=10$ unidades, entonces nosotros consideramos sólo presiones en un incremento de 10 unidades: 600, 610, ..., 790, 800.

El rango de operación factible que describe cada estación compresora se basa en las restricciones físicas de cada compresor individual. Por tanto, si una estación compresora puede operar bajo presiones de entrada y salida específicas, entonces nosotros suponemos que pueden calcularse los costos de operación de una configuración predeterminada; y por supuesto, el costo total de operación del sistema completo es la suma de los costos de operación sobre cada estación compresora m ; donde i_m y j_m son las variables de las presiones de succión y descarga en la estación compresora m , respectivamente. Recordemos de la Sección 3.4, que la función de costo $g_{(i,j)}^{\text{unit}}, \forall (i,j) \in A_c$ del consumo de combustible está basada en un espacio (x_{ij}, p_i, p_j) . Por tanto, para un flujo x_{ij} predeterminado, el costo simplemente es función de p_i y p_j .

Aasombrosamente, solo tres tipos de operaciones de composición son necesarias para reducir un sistema de redes. Dependiendo de las características de la red es posible necesitar alguna de las siguientes combinaciones: lineal, ramificada (término en inglés "*dangling*") y paralela. En esencia, lo que se persigue es reemplazar dos elementos (compresores) conectados A y B, cuyas matrices de costos están dadas por C_A y C_B , respectivamente, por un compresor "virtual" AB que refleja el comportamiento *óptimo* de ambos. En este sentido, cada elemento $C_{AB}(i,j)$ de la matriz de costos de este compresor virtual refleja la mejor manera de operarlo para una presión de entrada i y presión de salida j . Es aquí donde se hace uso del concepto de la DP no secuencial. A continuación describimos cómo calcular este costo C_{AB} para cada uno de los tres casos posibles.

Caso A: Combinando dos compresores en línea recta

En la Figura 4.11 se muestra cómo combinar dos elementos A y B en línea recta, es decir, estructuras lineales o gun-barrel, en un solo compuesto óptimo. Aquí el elemento A va del nodo I al nodo K, y el elemento B, va del nodo K al nodo J; por lo que, el elemento compuesto va del nodo I al nodo J. Entonces el elemento compuesto tiene su propia tabla de costos C_{AB} . Si solo esta transformación es permitida, el método resultante es esencialmente igual al método jerárquico de Gilmour, Luongo y Schroeder [14].

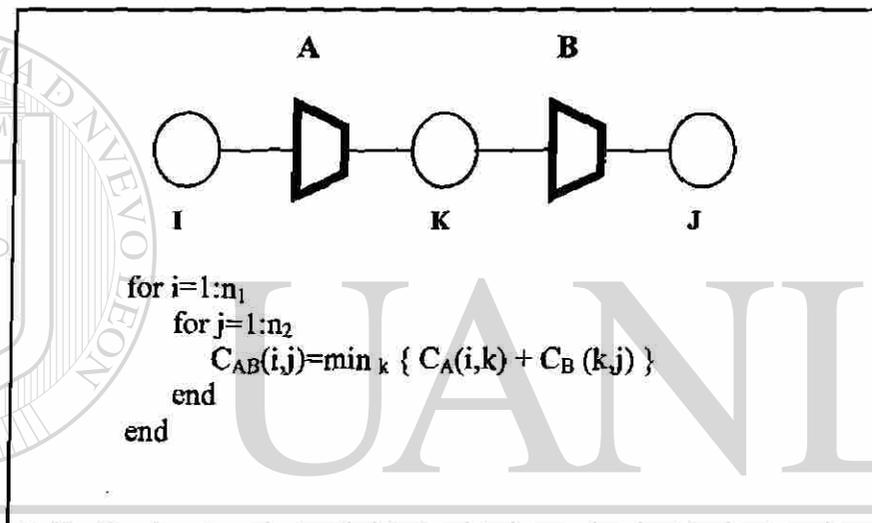


Figura 4.11: Combinación de dos estaciones en serie.

Caso B: Combinando dos compresores tipo árbol, como elementos “colgantes”

En la Figura 4.12 se muestra cómo combinar un elemento B “colgante” dentro de un elemento adyacente A. Aquí, el elemento A va del nodo I al nodo J, y el elemento B va del nodo J al nodo K; por lo que, el elemento compuesto va del nodo I al nodo J. Concluyendo así, que el nodo K no puede ser anexado a ningún otro elemento que B; he aquí la terminología de “elemento colgante” (basado en la definición del término en inglés “*dangling element*”).

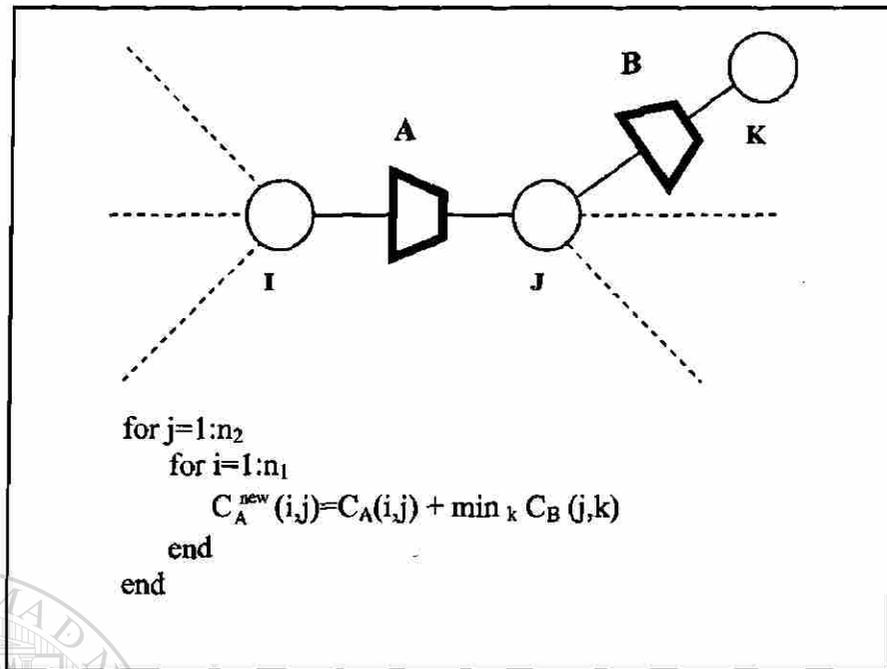


Figura 4.12: Combinación de dos compresores con un elemento colgante.

Caso C: Combinando dos estaciones compresoras en paralelo

En la Figura 4.13 se muestra cómo combinar dos estaciones compresoras en paralelo A y B, dentro de un solo elemento compuesto óptimo. Aquí, los elementos A y B, van del nodo I al nodo J, por lo que el elemento compuesto óptimo va del nodo I al nodo J.

Estas operaciones pueden aplicarse a redes complejas y eventualmente reducirlas a un único elemento compuesto equivalente a todo el sistema de configuración de presiones. Debemos hacer notar que algunas de las operaciones de combinación se inclinan, como debe ser, a aplicarse de forma recursiva hacia los elementos compuestos. Uno de los factores primordiales para que la técnica de DP no secuencial tenga éxito, es tener una estructura de datos apropiada para permitir la reconstrucción de la configuración de presiones óptimas actuales del sistema original, una vez que el objetivo óptimo ha sido leído de la última tabla compuesta.

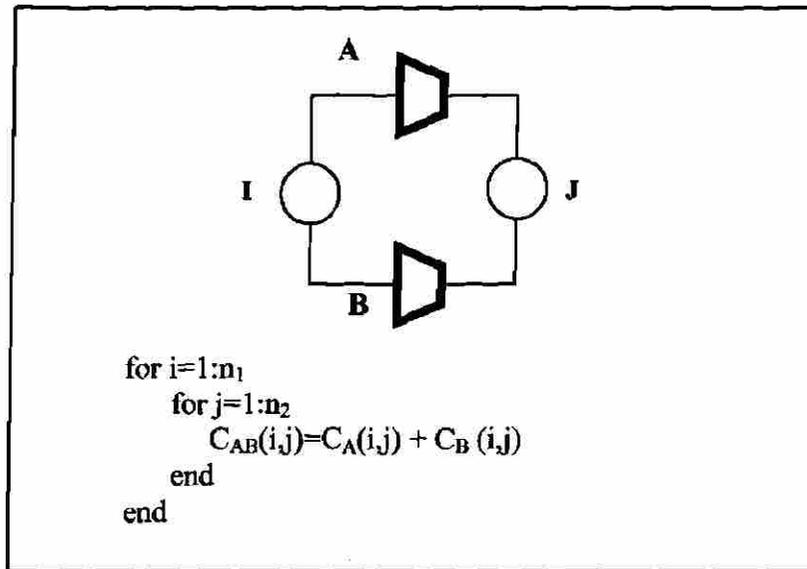


Figura 4.13: Combinación dos estaciones en paralelo.

Técnica de DP No Secuencial

Entrada: Un grafo $G = (V,A)$, donde $A = \{A_c\}$. Un flujo factible x .
M: Número de estaciones compresoras.
 $[P_m^L, P_m^U]$: Límites inferior y superior sobre las presiones, respectivamente.
 Donde $m=1, \dots, M$

Paso 1: Discretizar los rangos factibles de operación de las variables de presión en Δp unidades.

Introducción de variables auxiliares:
 i_m : Variable de decisión óptima de la presión de succión en la estación m .
 j_m : Variable de decisión óptima de la presión de descarga en la estación m .
 $C_m(i_m, j_m)$: Costo óptimo de la configuración operacional de la estación m .

Paso 2: Hacer
 Tomar dos compresores conectados.
 Reemplazarlos por un elemento compuesto "virtual" en base a los casos A, B o C.
 repetir (hasta que el sistema no puede reducirse más)

Paso 3: Buscar el mínimo valor incurrido en C_M , donde C_M representa la última tabla de desempeño óptimo del sistema completo de red.

Salida: Un conjunto de presiones óptimas. (Para el flujo especificado x)

Figura 4.14: Descripción de la Técnica de DP No Secuencial.

Finalmente, tomando como base los tres tipos de combinación (casos A, B y C) descritos anteriormente, la Fase 2 (ver Figura 4.2) para obtener las presiones óptimas es ejecutada utilizando una técnica de DP no secuencial, presentada en la Figura 4.14. Para ilustrar mejor el proceso de esta técnica veamos su desarrollo con el siguiente ejemplo:

Técnica de DP no secuencial aplicada sobre una topología cíclica

Tenemos la siguiente topología cíclica *net-c-19c7-C1*:

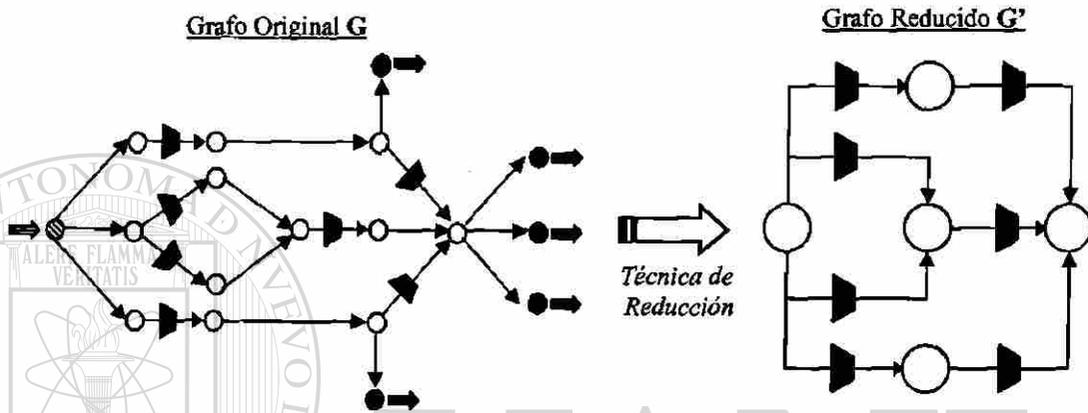


Figura 4.15: Grafo original G y su grafo reducido G' correspondiente.

Observamos el grafo original G y le aplicamos una fase de preprocesamiento empleando una técnica de reducción [41], obteniendo su grafo reducido G' (Figura 4.15) sin alterar su estructura matemática; y tomando como base los tres tipos de operaciones de combinación descritos anteriormente, procedemos a ejecutar el Paso 2 del algoritmo de DP no secuencial sobre la red reducida G' (Figura 4.16).

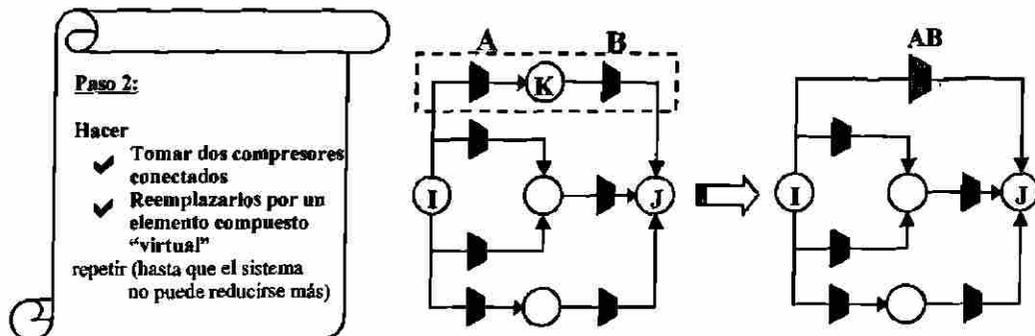


Figura 4.16: Ejecución del Paso 2 del algoritmo de DP no secuencial.

Paso 2:
Hacer
 Tomar dos compresores conectados.
 Reemplazarlos por un elemento compuesto "virtual"
 repetir (hasta que el sistema no puede reducirse más)

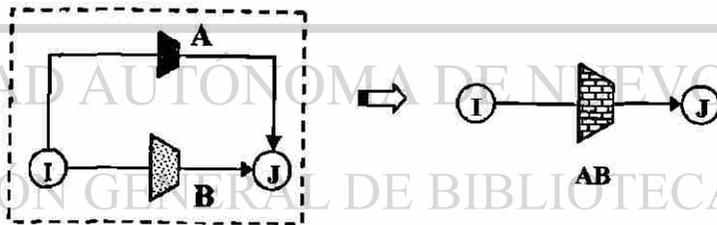
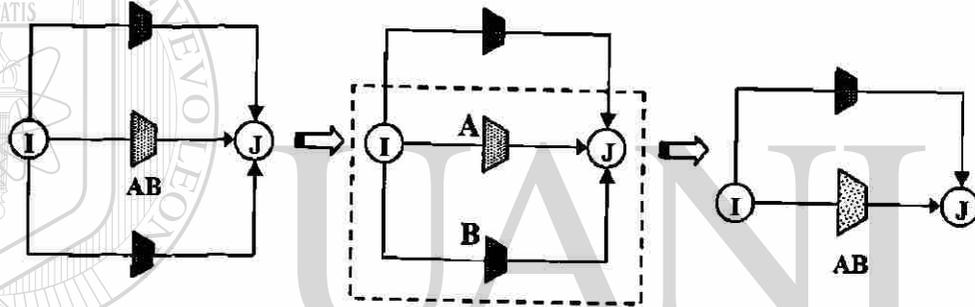
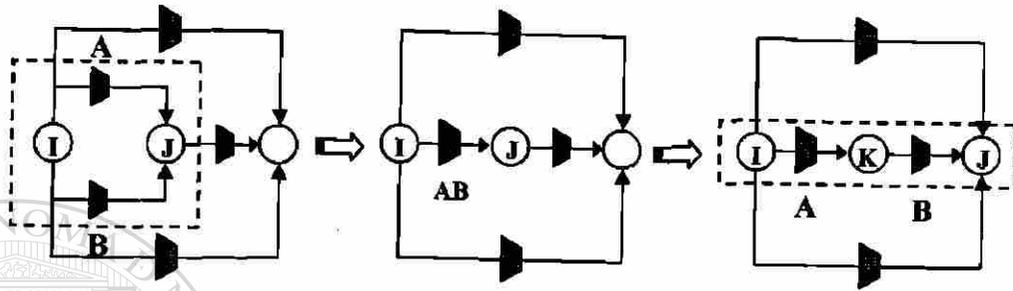
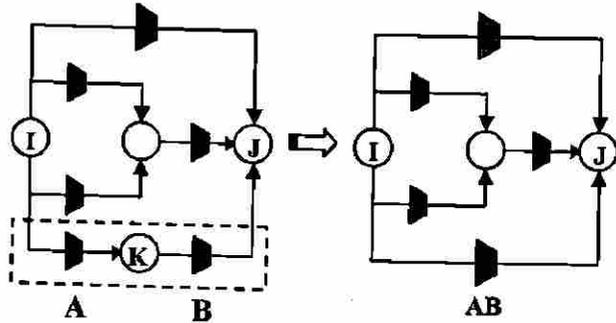


Figura 4.17: Reducción óptima de un sistema de red por la técnica de DP no secuencial.

En el Paso 2 tomamos al par de compresores conectados para reemplazarlos por un solo elemento compuesto "virtual", el cual se desempeña como sus elementos combinados de manera óptima.

En la Figura 4.17 se muestra cómo hemos continuado reduciendo el número de elementos de la red, un par cada vez, hasta que el sistema no ha podido reducirse más. Es pertinente aclarar que algunas de estas operaciones de combinación pueden ser

aplicadas de manera recursiva para obtener los elementos compuestos. Así, obtenemos finalmente un único elemento compuesto “virtual”, el cual desde la perspectiva de la implementación computacional corresponde a una tabla de costos de las configuraciones factibles dentro de un dominio discretizado. Procedemos entonces a ejecutar el Paso 3 buscando sobre la tabla de costos que corresponde al último elemento compuesto “virtual”, para obtener el mínimo valor, el cual describe completamente el costo incurrido por todas las estaciones compresoras en el sistema completo de red.

Supongamos que el rango de factibilidad de las presiones es de [200, 800] unidades PSIG y tenemos una discretización de $\Delta p=50$ unidades. Por tanto, la última tabla de costos C_{AB} podría suponerse como sigue.

P_j

P_i	200	250	300	350	400	450	500	550	600	650	700	750	800
200	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
250	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
300	-1	-1	-1	22312	2543	-1	-1	-1	-1	-1	-1	-1	-1
350	-1	-1	-1	4565	4789	987	765	-1	-1	-1	-1	-1	-1
400	-1	-1	-1	345	4567	6783	444	943	-1	-1	-1	-1	-1
450	-1	-1	-1	2343	1546	987	453	312	993	-1	-1	-1	-1
500	-1	-1	-1	1233	1232	1111	955	754	822	912	-1	-1	-1
550	-1	-1	-1	-1	-1	5643	453	1344	583	663	3201	-1	-1
600	-1	-1	-1	-1	-1	3455	999	923	765	653	2345	-1	-1
650	-1	-1	-1	-1	-1	-1	-1	466	477	913	5435	-1	-1
700	-1	-1	-1	-1	-1	-1	-1	-1	-1	841	9183	421	-1
750	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	991	-1
800	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Tabla 4.2: Tabla de costos incurridos por configuraciones óptimas de operación.

Por lo que de la Tabla 4.2, podemos observar que el mínimo valor incurrido por las configuraciones óptimas de operación es de 312, el cual corresponde a las presiones de succión $p_i = 450$ y de descarga $p_j = 550$, de los nodos i y j de la red original.

Así, finalmente en base a una estructura de datos apropiada procedemos a la reconstrucción de las configuraciones de presiones óptimas actuales al sistema original; esto, una vez que el valor objetivo óptimo ha sido leído de la última tabla de costo.

En resumen, de acuerdo con la situación actual, hemos aplicado el esfuerzo de optimización para resolver las variables de presión (p_i) en cada nodo. El método de

solución previamente descrito en este capítulo obtiene un conjunto de flujos factibles, para después proseguir con un proceso de búsqueda de presiones óptimas (para los flujos preespecificados), utilizando en ambos procesos técnicas de preprocesamiento para la refinación del modelo y reducción de la red, así como una técnica de DP no secuencial para la búsqueda de presiones óptimas. El procedimiento de la Fig. 4.14 tiene complejidad $O(|A_c| \cdot N_p^2)$, donde $N_p = pmax/\Delta P$ es el número de puntos en la discretización del rango de presiones (asumiendo un rango ΔP y un valor máximo de presión $pmax$). Esto implica que el procedimiento global NDP (Fig. 4.2) es $O(|A| \cdot N_p^2)$.

Ahora bien, continuando con el objetivo de enriquecer más la calidad de la solución al PMCC, en el siguiente capítulo presentamos el último componente (d) propuesto en la metodología de solución, el cual introduce un procedimiento heurístico que refleja el esfuerzo de optimización tanto en las variables del flujo másico (x_{ij}) sobre cada arco en el sistema de red, como en las variables de presión para cada nodo del mismo. Este procedimiento heurístico aplica simultáneamente a su estrategia de búsqueda local, una técnica de DP no secuencial (ver Sección 4.2) para obtener presiones óptimas, y por tanto, proveer soluciones aproximadas de mayor calidad a las topologías cíclicas, las cuales son el reto en este trabajo de investigación.

CAPÍTULO 5

MÉTODO DE SOLUCIÓN: BÚSQUEDA TABÚ

En este capítulo, haciendo referencia al método de solución descrito en el capítulo anterior, introducimos el procedimiento heurístico que se propone como promesa determinante y esencial en la metodología de solución, para proveer soluciones aproximadas de buena calidad en topologías cíclicas.

En la primera sección de este capítulo, sustentándonos en la idea de justificar este nuevo procedimiento, presentamos un análisis detallado de la noción de heurística para respondernos la siguiente importante cuestión:

¿Por qué una heurística puede verse como una buena (o inclusive la mejor) opción en la búsqueda de soluciones de calidad al PMCC sobre redes cíclicas de gas natural, el cual típicamente se conoce como NP-completo?

En las secciones restantes, tanto una descripción puntual de la metodología general como la implementación del procedimiento heurístico propuesto, son presentadas.

5.1 Heurísticas

La idea más genérica del término *heurístico* está relacionada con la tarea de resolver inteligentemente problemas reales usando el conocimiento disponible. El término *heurístico* proviene de una palabra griega con un significado relacionado con el concepto de encontrar, y se vincula a la supuesta exclamación *eureka* de Arquímedes al descubrir su famoso principio [27].

La concepción más común en Inteligencia Artificial (IA) es interpretar que heurístico es el *calificativo* apropiado para los procedimientos que, empleando conocimientos acerca de un problema y de las técnicas aplicables, tratan de aportar soluciones (o acercarse a ellas) usando una cantidad razonable de recursos (generalmente tiempo). En un problema de optimización, aparte de las condiciones que deben cumplir las soluciones factibles del problema, se busca la que es óptima según algún criterio de comparación entre ellas. En Investigación de Operaciones, el término heurístico se aplica a un procedimiento de solución de problemas de optimización con una concepción diferente. Se califica de heurístico a un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un costo computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación. Se usa el calificativo heurístico en contraposición a exacto, que se aplica a los procedimientos que se les exige que la solución aportada sea óptima o factible. Una *solución heurística* de un problema es la proporcionada por un *método heurístico*, es decir, aquella solución sobre la que se tiene cierta confianza por considerarse de buena calidad. También es usual aplicar el término heurístico cuando, utilizando el conocimiento que se tiene del problema, se realizan modificaciones al procedimiento de solución del problema que, aunque no contrarrestan la complejidad del mismo, si mejoran el rendimiento en su comportamiento práctico.

Ciertas *heurísticas* para resolver un problema de optimización pueden ser más generales o específicas que otras, tales como GRASP [9], búsqueda tabú [16], etc. Los métodos heurísticos específicos deben ser diseñados a propósito para cada problema, utilizando toda la información disponible y el análisis teórico del modelo. Los procedimientos específicos bien diseñados suelen tener un rendimiento significativamente más alto que las heurísticas generales.

Las *heurísticas* más generales presentan otro tipo de ventajas, como la sencillez, adaptabilidad y robustez de los procedimientos. Sin embargo, las *heurísticas* generales emanadas de las *metaheurísticas* pueden mejorar su rendimiento utilizando recursos computacionales y estrategias inteligentes.

El término metaheurística se obtiene de anteponer a *heurística* el sufijo *meta* que significa “más allá” o “a un nivel superior”. Los conceptos actuales de lo que es una metaheurística están basados en las diferentes interpretaciones de lo que es una forma inteligente de resolver un problema. Las metaheurísticas son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. El término metaheurística apareció por primera vez en el artículo seminal sobre búsqueda tabú de Fred Glover en 1986 [15]. A partir de entonces han surgido muchos procedimientos que aumentan el campo de aplicación de las heurísticas, adoptando la denominación de metaheurísticas.

5.2 Metaheurísticas

Las metaheurísticas son estrategias para diseñar procedimientos heurísticos. Por tanto, los tipos de metaheurísticas se establecen, en primer lugar, en función del tipo de procedimientos a los que se refiere. Algunos de los tipos fundamentales son las metaheurísticas para los métodos de relajación, las metaheurísticas para los procesos constructivos, las metaheurísticas para las búsquedas por entornos y las metaheurísticas para los procedimientos evolutivos.

- Las metaheurísticas de relajación se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original (es decir, modificaciones del modelo que hacen al problema más fácil de resolver), cuya solución facilita la solución del problema original.
- Las metaheurísticas constructivas se orientan a los procedimientos que tratan de la obtención de una solución a partir del análisis y selección paulatina de las componentes que la forman.
- Las metaheurísticas de búsqueda guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explotar las estructuras de entornos asociadas.
- Las metaheurísticas evolutivas están enfocadas a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de

soluciones.

Algunas metaheurísticas surgen combinando metaheurísticas de distinto tipo, como la metaheurística GRASP (por sus siglas en inglés, *Greedy Randomized Adaptive Search Procedure*) [38] y [39], que combina una fase constructiva con una fase de búsqueda de mejora. Otras metaheurísticas se centran en el uso de algún tipo de recurso computacional o formal especial como las redes neuronales, los sistemas de hormigas o la programación por restricciones y no se incluyen claramente en ninguno de los cuatro tipos anteriores.

Por otro lado, de una u otra forma, todas las metaheurísticas se pueden concebir como estrategias aplicadas a procesos de búsqueda, donde todas las situaciones intermedias en el proceso de resolución del problema se interpretan como elementos de un espacio de búsqueda, que se van modificando a medida que se aplican las distintas operaciones diseñadas para llegar a la resolución definitiva. Por ello, y porque los procesos de búsqueda heurística constituyen el paradigma central de las metaheurísticas, es frecuente interpretar que el término metaheurística es aplicable esencialmente a los procedimientos de búsqueda sobre un espacio de soluciones alternativas.

¿Por qué utilizar una Metaheurística?

Para respondernos esta pregunta es conveniente, primero, definir las propiedades deseables de las metaheurísticas -propiedades que favorezcan el interés práctico y teórico - dando pauta a dirigir los esfuerzos para contribuir al desarrollo científico; pero no será posible mejorar todas las propiedades a la vez, dado que algunas son parcialmente contrapuestas. Una relación de tales propiedades debe incluir las siguientes:

- ✓ *Simple*. La metaheurística debe estar basada en un principio sencillo y claro; fácil de comprender.
- ✓ *Precisa*. Los pasos y fases de la metaheurística deben estar formulados en términos concretos.

- ✓ *Cohérente.* Los elementos de la metaheurística debe deducirse naturalmente de sus principios.
- ✓ *Efectiva.* Los algoritmos derivados de la metaheurística deben proporcionar soluciones de muy alta calidad; óptimas o muy cercanas a las óptimas.
- ✓ *Eficaz.* Por la capacidad de búsqueda para alcanzar soluciones de buena calidad, a la vez que se considera como una buena opción de resolución a este tipo de problemas.
- ✓ *Eficiente.* La metaheurística debe realizar un buen aprovechamiento de recursos computacionales; tiempo de ejecución y espacio de memoria.
- ✓ *General.* La metaheurística debe ser utilizable con buen rendimiento en una amplia variedad de problemas.
- ✓ *Adaptable.* La metaheurística debe ser capaz de adaptarse a diferentes contextos de aplicación o modificaciones importantes del modelo.
- ✓ *Robusta.* El comportamiento de la metaheurística debe ser poco sensible a pequeñas alteraciones del modelo o contexto de aplicación.
- ✓ *Interactiva.* La metaheurística debe permitir que el usuario pueda aplicar sus conocimientos para mejorar el rendimiento del procedimiento.
- ✓ *Múltiple.* La metaheurística debe suministrar diferentes soluciones alternativas de alta calidad entre las que el usuario pueda elegir.
- ✓ *Autónoma.* La metaheurística debe permitir un funcionamiento autónomo, libre [®] de parámetros o que se puedan establecer automáticamente.

Un punto importante es que, en el campo de investigación sobre las metaheurísticas, se ofrecen más oportunidades para aplicar la intuición que la deducción. En contraste con el éxito práctico de muchas metaheurísticas, el estudio teórico está más retrasado. Frecuentemente se obtienen buenas nuevas heurísticas, con algo de inventiva y gran esfuerzo en el ajuste de numerosos parámetros, pero las razones del por qué funcionan tan bien permanecen desconocidas. La situación es incluso peor para los híbridos, donde las aportaciones de las metaheurísticas implicadas y el beneficio de la interacción raramente son objetos de un estudio experimental bien diseñado.

Debido a que las metaheurísticas se definen básicamente como métodos de solución que unifican los procedimientos de mejoramiento local y las estrategias más eficaces, creando así, un proceso suficientemente eficiente para salir de un óptimo local y ejecutar una búsqueda robusta en un espacio de solución complejo, hacen de este tema de estudio un punto fundamental en el tratamiento de problemas complejos en el mundo real. Con el tiempo, estos métodos han sido incluidos en los procedimientos que emplean una o más estructuras de vecindad para definir los posibles movimientos de transición de una solución a otra.

Con el paso de los años, se ha visto la necesidad de crear una nueva línea de investigación con el objetivo de aplicar procedimientos metaheurísticos, los cuales se ven como promesa de obtener buenas soluciones aproximadas en muchos tipos de problemas complejos, particularmente aquellos de naturaleza combinatoria; esto, ya que no han podido ser resueltos eficientemente por técnicas tradicionales debido a las dificultades matemáticas inherentes que conllevan. Incluso, podemos remarcar que en algunas de las aplicaciones más notables de estos métodos tradicionales, se ha visto la incorporación de estrategias metaheurísticas como parte de su eficiencia.

5.3 Búsqueda Tabú

En esta sección proponemos y describimos un procedimiento heurístico de TS para alcanzar soluciones aproximadas de buena calidad en redes cíclicas; esto, como parte secuencial del método de solución propuesto en el Capítulo 4. Aquí, la metodología general e implementación de un TS con estrategia de memoria corta para resolver el PMCC, es presentada.

5.3.1 Metodología General

La palabra *Tabú* proviene de “Tongan” una lengua hablada en la Polinesia, la cual es usada por los aborígenes de la isla de Tonga para describir aquello que no puede ser tocado por ser sagrado [16]. Por otro lado, tomando como referencia el diccionario Webster’s, la palabra *Tabú* también significa “una prohibición impuesta por acuerdo

social como una medida preventiva”.

La Búsqueda Tabú (TS por sus siglas en inglés, *Tabu Search*) [16] es un procedimiento metaheurístico cuya característica distintiva es el uso de memoria adaptativa y de estrategias especiales de resolución de problemas. Esta técnica fue desarrollada inicialmente por Fred Glover [16] en 1986, siendo una década más tarde cuando junto con M. Laguna [36] (1997), hacen la primera publicación oficial describiendo la metodología general.

La filosofía de TS se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje. El marco de memoria adaptativa de TS explota la historia del proceso de resolución del problema haciendo referencia a cuatro dimensiones principales: Ser reciente, frecuente, influyente y de calidad. En esencia, TS es una metaheurística que puede ser utilizada para guiar cualquier procedimiento en la búsqueda agresiva del óptimo local del problema. Por agresiva nos referimos a una estrategia en la que cada paso tomará la mejor -o menos mala- solución del entorno dado. A tal efecto, TS toma de la Inteligencia Artificial (IA) el concepto de memoria y lo implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta la historia de ésta. Es decir, el procedimiento trata de extraer información de lo sucedido y actuar en consecuencia. En este sentido puede decirse que hay un cierto aprendizaje y que la búsqueda es inteligente. El principio de TS podría resumirse como:

Es mejor una mala decisión basada en información que una buena decisión al azar, ya que, en un sistema que emplea memoria, una mala elección basada en una estrategia proporcionará claves útiles para continuar la búsqueda. Una buena elección fruto del azar no proporcionará ninguna información para posteriores acciones.

Así podemos concluir que las dos características principales que un algoritmo de búsqueda tabú posee –y que lo distinguen de otras estrategias de búsqueda- son:

- 1) La utilización de una estrategia basada en el uso de estructuras de memoria para escapar de los óptimos locales, en los que se puede caer al “moverse” de una solución a otra en el espacio de soluciones factibles; y

- 2) La intención de usar una *lista tabú* (*Tabu List*) es evitar los ciclos donde la búsqueda oscila entre los estados previamente visitados.

El método TS opera bajo el supuesto de que se puede construir un entorno para identificar soluciones adyacentes (llamadas típicamente “soluciones vecinas”) que puedan ser alcanzadas desde la solución actual.

TS comienza de la misma forma que cualquier procedimiento de búsqueda local, procediendo iterativamente de una solución x a otra, basándose en un espacio de solución. Sin embargo, en lugar de considerar todo el entorno de una solución, TS define el espacio reducido $V(x)$ como aquellas soluciones disponibles del entorno de x . Así, se considera que a partir de x , sólo las soluciones del entorno reducido son alcanzables.

Existen muchas maneras de definir el entorno reducido de una solución. La más sencilla es etiquetar como tabú las soluciones previamente visitadas en un pasado cercano. Esta forma se conoce como **memoria a corto plazo** (*short term memory*) y está basada en guardar en una *lista tabú* las soluciones visitadas recientemente (*Recency*). Así en una iteración determinada, el entorno reducido de una solución se obtendría como el entorno usual eliminando las soluciones etiquetadas como tabú.

Como se ha mencionado anteriormente, el objetivo principal de etiquetar como tabú a las soluciones previamente visitadas, es para evitar que la búsqueda se cicle en un óptimo local. Por ello, se considera que tras un cierto número de iteraciones, la búsqueda está en una región distinta y puede librarse del estatus tabú (perteneciente a *Tabu List*) a las soluciones antiguas. De esta forma se reduce el esfuerzo computacional de calcular el entorno reducido en cada iteración. En los orígenes de TS se sugerían listas de tamaño pequeño, actualmente se considera que las listas pueden ajustarse dinámicamente según la estrategia que se esté utilizando.

Se define un nivel de aspiración como aquellas condiciones que, de satisfacerse, permitirían alcanzar una solución aunque tenga status tabú. Una implementación sencilla consiste en permitir alcanzar una solución siempre que mejore a la mejor almacenada, aunque esté etiquetada tabú. De esta forma se introduce cierta flexibilidad en la búsqueda y se mantiene su carácter agresivo.

Es importante considerar que los métodos basados en búsqueda local requieren de la exploración de un gran número de soluciones en poco tiempo, por ello es crítico el reducir al mínimo el esfuerzo computacional de las operaciones que se realizan a menudo. En ese sentido, la memoria a corto plazo de TS está basada en atributos en lugar de ser explícita; esto es, en lugar de almacenar las soluciones completas (como ocurre en los procedimientos de búsqueda exhaustiva) se almacenan únicamente algunas características de éstas.

La memoria mediante atributos produce un efecto más útil y beneficioso en la búsqueda, ya que un atributo o grupo de atributos identifica a un conjunto de soluciones del mismo modo que los hiperplanos utilizados en los algoritmos genéticos. A largo plazo, el identificar y mantener atributos que inducen a una cierta estructura beneficiosa en las soluciones visitadas, hace posible mantenerlos en consideración, aún perteneciendo a una solución tabú.

Los puntos esenciales en un algoritmo de búsqueda local son los siguientes:

- Selección de una solución inicial x_0
 - Elección del entorno $V(x_0)$ y generación de una solución candidata x'
 - Evaluación de la función objetivo $f(x')$
 - Actualización de la mejor solución y de la mejor de las mejores soluciones
-
- Criterio de finalización

5.3.2 Antecedentes Científicos de la Búsqueda Tabú

La Búsqueda Tabú es un método metaheurístico propuesto originalmente por Glover en 1986, y ha sido aplicado en múltiples y diversos problemas combinatorios que han aparecido durante el transcurso de las últimas décadas en el área de la investigación de operaciones. En muchos de estos problemas, este método ha provisto soluciones de muy buena calidad, muchas de ellas cercanas a la optimalidad; por lo que ha llegado a considerarse como una propuesta muy efectiva. Inclusive cuando en ocasiones no ha sido el mejor, sí ha sido lo suficientemente bueno para resolver los problemas difíciles, en los cuales los métodos tradicionales han llegado a tener un éxito relativo.

El hecho de haber aplicado el método sobre una amplia gama de problemas complejos durante las últimas décadas, obteniendo muy buenos resultados, ha permitido que muchos investigadores se vean interesados en su implementación para obtener buenas soluciones sobre los diversos problemas (combinatorios) que aquejan al mundo real, siendo estos muy difíciles de resolver de manera exacta.

TS tiene sus antecedentes en métodos diseñados para sobrellevar los límites de factibilidad u optimalidad local, límites comunes que son tratados como barreras en los procedimientos clásicos. Entonces este método, refina o elimina las cotas sistemáticamente, para permitir la exploración de regiones no consideradas en los métodos tradicionales. Una característica distintiva de este procedimiento es el uso de memoria adaptativa y de estrategias especiales de resolución de problemas. TS es el origen del enfoque basado en memoria y estrategia intensiva en la literatura de las metaheurísticas, en contraposición con los métodos que no tienen memoria o que sólo usan una débil memoria basada en herencia. De forma opuesta a los procesos que confían casi exclusivamente en la aleatorización, TS hace uso del espacio de solución basándose en la explotación de los patrones históricos que han venido registrándose durante la búsqueda de soluciones.

Los principios fundamentales de TS fueron establecidos en una serie de artículos a finales de los años 80. Por lo que, el destacable éxito de TS en la resolución de problemas de optimización complejos (particularmente sobre aquellos que surgen en aplicaciones del mundo real) ha causado una explosión de nuevas aplicaciones de TS durante los últimos años.

La filosofía de la búsqueda tabú es derivar y explotar una colección de estrategias inteligentes para la resolución de problemas, basados en procedimientos implícitos y explícitos de aprendizaje. El marco de memoria adaptativa de TS no sólo explota la historia del proceso de resolución del problema, sino que también exige la creación de estructuras para hacer posible tal explotación.

5.3.3 Implementación del Procedimiento de Búsqueda Tabú

Con el objetivo de realizar la implementación del algoritmo de Búsqueda Tabú, denominado NONDP_TS (por sus siglas en inglés, *NON-sequential Dynamic Programming aplicado en Tabu Search*, ver Figura 5.2), en esta sección presentamos como primera etapa una descripción del tipo de soluciones empleadas en el desarrollo de la heurística. Por tanto, en la Tabla 5.1, se muestra la representación natural de una solución factible con base en tres tipos de variables o componentes fundamentales, los cuales están directamente relacionados con un ciclo -formado por los compresores-tomado al azar sobre la topología de red en cuestión (ver Figura 5.1).

COMPONENTE	DESCRIPCIÓN
ESTÁTICO	Todos aquellos valores de flujo másico que no pertenecen a un ciclo de la topología de red.
VARIABLE	Todos aquellos valores de flujo másico que pertenecen al ciclo de la topología de red.
BÚSQUEDA	Todos aquellos valores de la presión en cada nodo de la topología de red.

Tabla 5.1. Descripción de los tipos de componentes que describen a una solución del procedimiento NONDP_TS.

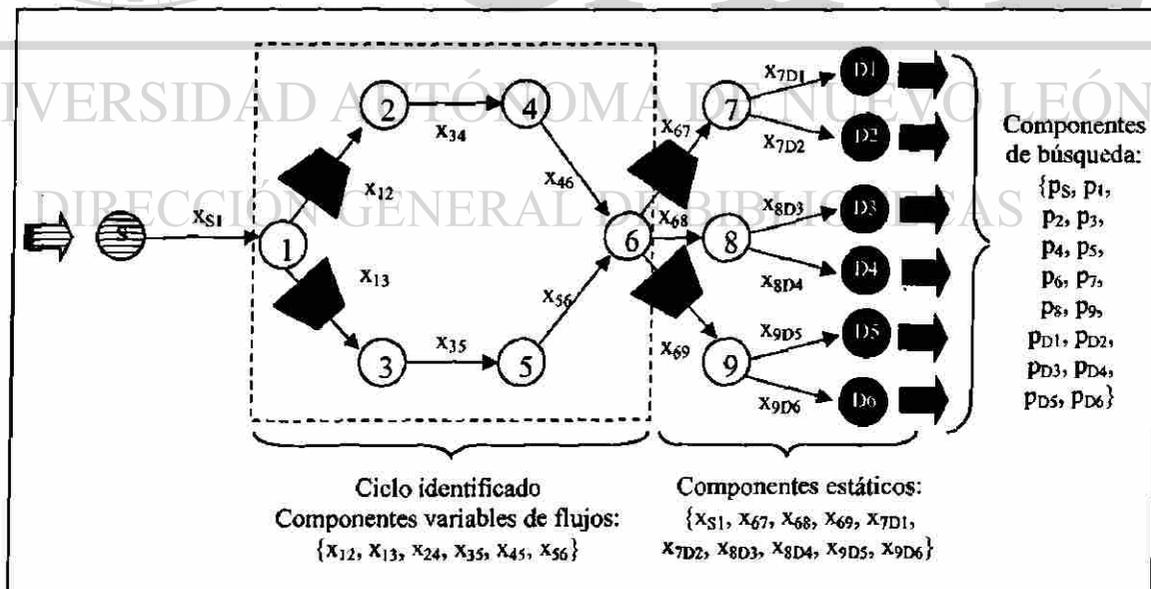


Figura 5.1: Topología cíclica en la cual se definen los tres componentes básicos de una solución factible en el procedimiento NONDP_TS.

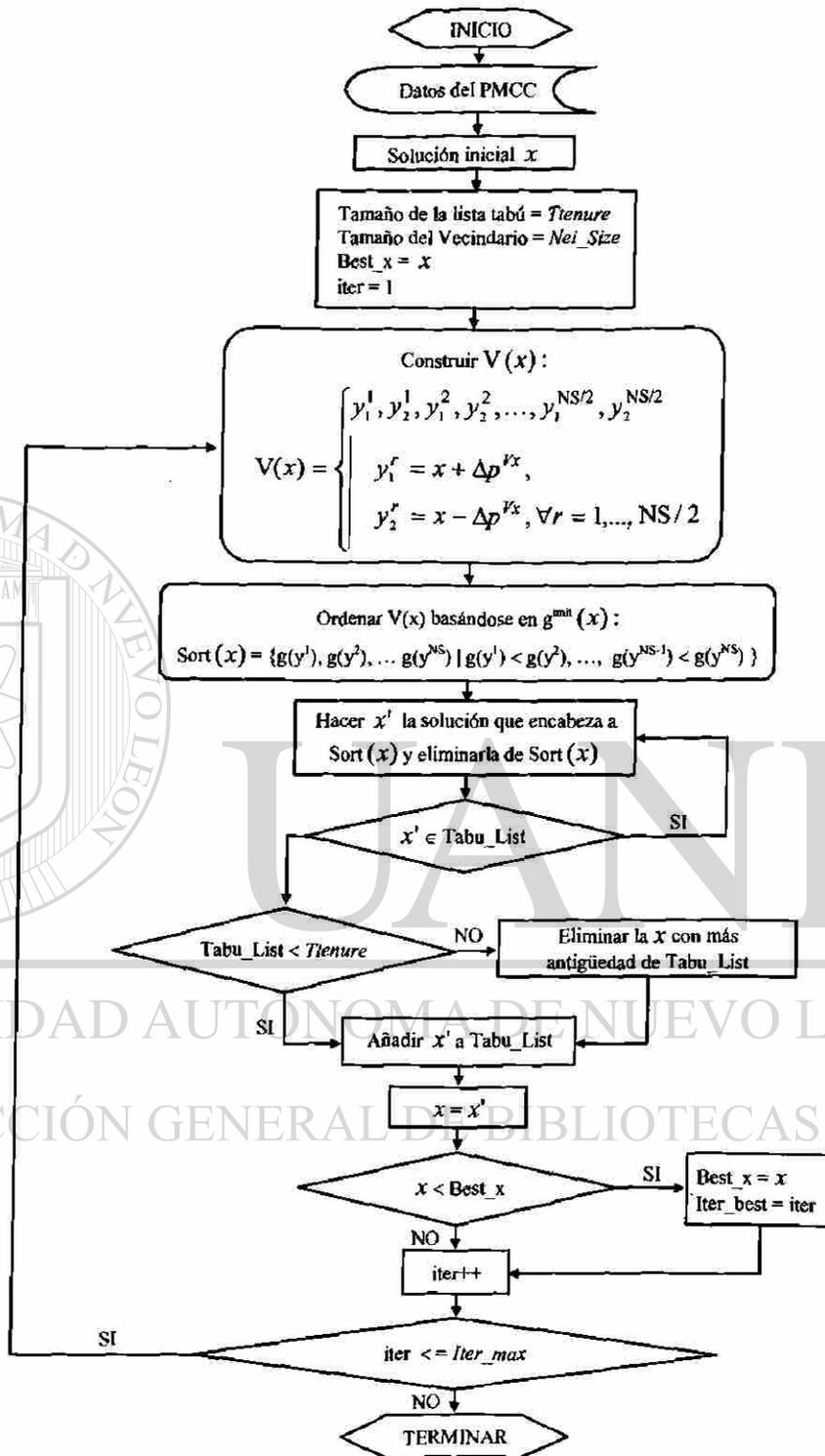


Figura 5.2: Diagrama de flujo del Algoritmo NONDP_TS.

En la Figura 5.1, mediante una topología cíclica, se muestran gráficamente los componentes básicos que conforman a una solución factible usada en el algoritmo NONDP_TS.

Ahora introducimos la implementación práctica del procedimiento NONDP_TS, el cual consiste en los diversos pasos descritos a continuación y presentados por el diagrama de flujo de la Figura 5.2.

Esto es, el espacio de búsqueda de la parte combinatoria que este procedimiento heurístico aborda es la selección del flujo másico (x_{ij}) en cada arco, ya que las variables de presión (p_i) son obtenidas de manera óptima por una técnica de DP no secuencial, la cual es utilizada de forma iterativa dentro del procedimiento heurístico; por lo que, es natural representar un estado en la búsqueda local por un vector $x = \langle x_1, \dots, x_m \rangle$, donde x_m es un arco que pertenece a un ciclo seleccionado al azar dentro de la instancia de red. Por lo tanto, el atributo de una solución de TS está definido por el valor de una variable de flujo del ciclo seleccionado al azar.

Como primera fase, una solución factible inicial es generada por el algoritmo NDP, el cual usa una técnica de DP no secuencial; a la vez que los valores de los parámetros apropiados del TS son determinados.

Después, todos los posibles vecinos de la solución x son generados y evaluados con base a un cambio del atributo. Un vecino es un diseño que puede ser alcanzado desde el diseño actual por una simple y básica transformación de este actual diseño. Cada vecino en turno puede ser ajustado añadiendo o sustrayendo Δp^{v*} unidades de

flujo másico sobre un arco del ciclo seleccionado al azar, y actualizando los demás. Así, durante una iteración, un número Nei_Size de vecinos han sido desarrollados y evaluados. Paralelamente a este proceso de construcción de la vecindad $V(x)$, el procedimiento ordena a los vecinos $x' \in V(x)$ de manera ascendente con respecto a su valor objetivo. De estas soluciones vecinas, la mejor x' es seleccionada; notar que esta solución x' no necesita ser mejor que la solución actual x . El valor del atributo de esta solución x' seleccionada, dentro del movimiento de la solución actual a la mejor solución del vecindario, es añadido a la *lista tabú*. Haciendo $x = x'$, una nueva búsqueda local es ejecutada, manteniendo en memoria el mejor valor del atributo de la solución al

momento. Así, nuevamente la mejor solución vecina es seleccionada como un candidato para el próximo movimiento. Sin embargo, el movimiento no debe regresar la búsqueda a la solución inicial previa. Para prevenir tal ciclamiento, el movimiento es verificado sobre la *lista tabú* de los movimientos anteriores. Si el movimiento no está sobre la lista, entonces éste es permitido; en caso contrario, la mejor siguiente solución sobre el vecindario $V(x)$ es verificado para su aceptabilidad. Este chequeo es repetido hasta que un vecino aceptable es encontrado, el cual no pertenece a la categoría de “tabú”. Este proceso de búsqueda local y selección del mejor vecino no tabú es entonces repetido hasta que un criterio de terminación del número máximo de iteraciones es satisfecho.

Las dos características esenciales de una TS, la cual la distingue de cualquier otra estrategia son:

- 1) El uso de una lista tabú, la cual previene un posible ciclamiento donde la búsqueda oscilaría entre los estados visitados previamente con las nuevas áreas no exploradas;
- 2) La aceptación de una nueva solución vecina, la cual no depende del valor actual de la función objetivo, así permite que TS escape de un mínimo local.

○ Como analogía esencial en este último componente de la metodología de solución, se menciona la siguiente afirmación:

Aun cuando las estrategias de diversificación e intensificación son empleadas típicamente en los modelos de Búsqueda Tabú, en este trabajo, para propósitos de alcanzar soluciones aproximadas de buena calidad, una estrategia de memoria corta es observada e implementada como promesa de conseguir este objetivo, siendo considerada suficiente y de vital importancia.

A continuación, los componentes básicos del procedimiento heurístico NONDP_TS propuesto (ver Figura 5.2) son discutidos detalladamente: Generación de una solución inicial, selección de los parámetros adecuados del TS, construcción de una vecindad $V(x)$, definición de movimiento y lista tabú.

Generación de una solución inicial

Para generar la solución inicial del procedimiento heurístico de TS, se aplica el algoritmo NDP discutido en el capítulo anterior, el cual se basa en una técnica de DP no secuencial.

Selección de los parámetros adecuados

Los parámetros del algoritmo NONDP_TS y las consideraciones para su elección son las siguientes:

Iter_max

Número máximo de iteraciones. Se realizaron pruebas que sustentan la idea de que cantidad no significa calidad. Se mostró que 100 iteraciones arrojaron soluciones igualmente buenas a las encontradas con 500, 1000 ó 2000 iteraciones. (Ver Apéndice A para observar los resultados de las gráficas de desempeño del algoritmo NONDP_TS sobre las instancias probadas con 100 y 500 iteraciones).

Ttenure

Tamaño de la lista tabú (*Tabu List*). Este parámetro parece ser el de mayor importancia en esta aplicación. Previamente se consideró un valor de 8, para contrarrestar con la necesidad de una estrategia enfocada a la diversificación.

Nei_Size

Tamaño de la vecindad $V(x)$. Este constituye el número de vecinos a construir para $V(x)$. Previamente el valor considerado fue de 20, para suplir preliminarmente una estrategia de intensificación.

Δp^{Vx}

Cantidad de unidades de flujo a ser modificadas en la solución actual para la construcción del vecindario. En este parámetro se consideraron tres diferentes valores de unidades de flujo (2, 5 y 10) sobre un rango factible de operación de las variables de flujo, típicamente encontradas entre 150 y 3000 unidades de flujo.

Criterio de parada: Definido cuando se alcanza un cierto número de iteraciones sin mejoramiento del valor objetivo o por un número de iteraciones fijas; para nuestra evaluación computacional y análisis paramétrico del procedimiento heurístico NONDP_TS, se fijó un número máximo de 100 iteraciones.

Las evaluaciones computacionales para determinar los valores en los parámetros anteriores, se presentan en el siguiente capítulo.

Construcción de una vecindad $V(x)$

En este proceso de construcción se generan nuevas soluciones factibles, las cuales son “vecinas” de la solución actual. En la implementación del procedimiento para la construcción de la vecindad $V(x)$ se utilizaron dos estrategias, las cuales se emplean para modificar el segundo componente que forma parte de la estructura de la solución mostrada por la Tabla 5.1: *Componente variable de los flujos sobre el ciclo*. Las estrategias para crear los vecinos pueden describirse como sigue:

- Estrategia 1: Incrementar en Δp^{Vx} unidades de flujo másico a un arco que pertenece al ciclo de la topología a resolver.

- Estrategia 2: Decrementar en Δp^{Vx} unidades de flujo másico a un arco que pertenece al ciclo de la topología a resolver.

El método puede ser visto como una técnica iterativa, la cual explora un conjunto de soluciones del problema, denotada por \mathfrak{R} , ya que repetidamente se hacen movimientos de una solución x_0 a otra solución x' localizada en el vecindario $V(x_0)$ de x_0 . Estos movimientos son ejecutados en pro de una eficiente búsqueda de la solución óptima, mediante la minimización de la función objetivo $g(x)$.

Definamos ahora la noción de vecindario $V(x)$ para cada solución $x \in \mathfrak{R}$. Sea $V(x)$

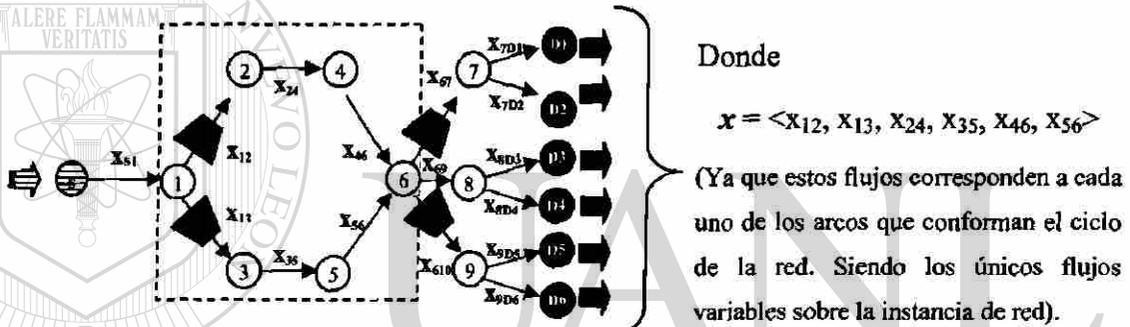
un conjunto de soluciones en \mathfrak{R} , alcanzable desde x por una ligera modificación de Δp^{vx} unidades, por lo que:

$$V(x) = \{ x' \in \mathfrak{R} \mid x' = x \oplus \Delta p^{vx}, \Delta p^{vx} \in \Psi \}$$

donde Ψ contiene todas las posibles modificaciones y $x' = x \oplus \Delta p^{vx}$ significa que x' es obtenida al aplicar una modificación de Δp^{vx} unidades a x .

Nótese que para una $x' \in \mathfrak{R}$ factible dada, los únicos flujos que pueden modificarse son aquéllos que pertenecen a un ciclo formado por los compresores.

Por ejemplo, tomemos como referencia al grafo de la Figura 5.1.



Ahora bien, sea $x = \langle 130, 70, 130, 70, 130, 70 \rangle$ y $\Delta p^{vx} = 5$ unidades de flujo, basándonos en las estrategias para la construcción de la vecindad $V(x)$, tenemos entonces:

Estrategia 1: Incrementando en Δp^{vx} unidades a un arco de x , por lo tanto:

$$x_1' = \langle (130 + 5), 70, 130, 70, 130, 70 \rangle$$

Actualizando todos los arcos restantes, tenemos:

$$x_1' = \langle 135, 65, 135, 65, 135, 65 \rangle$$

Estrategia 2: Disminuyendo en Δp^{vx} unidades a un arco de x , por lo tanto:

$$x_2' = \langle (130 - 5), 70, 130, 70, 130, 70 \rangle$$

Actualizando todos los arcos restantes, tenemos:

$$x_2' = \langle 125, 75, 125, 75, 125, 75 \rangle$$

Por lo tanto, hemos obtenido dos vecinos (x_1' y x_2') de x al modificar en Δp^{vx} unidades de flujo másico a un solo arco del ciclo. Esto puede repetirse para cada arco contenido en el ciclo, y para diversos valores de Δp^{vx} . Es así, como un vecindario $V(x)$ de x es construido.

Definición de Movimiento

A cada paso del procedimiento, un conjunto $V^* \in \mathfrak{R}$ es generado y el problema de optimización local $\min \{ g(x') | x' \in V^* \subseteq \mathfrak{R} \}$ es resuelto. Además, para escapar del mínimo local, la idea es moverse al mejor vecino x' en V^* aún si $g(x') > g(x_0)$. Forzando de esta forma salir de un mínimo local hacia una mejor aproximación del óptimo; así, cada movimiento da como resultado una mejor aproximación (o al menos, en el menor deterioro posible) del valor de la función objetivo.

A cada movimiento de este procedimiento iterativo heurístico, muchas soluciones tienen que ser evaluadas, por lo que es importante ejecutar este cálculo de una manera eficiente. Así, en algunos casos es posible almacenar los valores de las soluciones que fueron empleadas en el vecindario previo. En tales casos, si la creación de vecinos puede ser aplicada en dos vecindarios consecutivos, entonces sucede frecuentemente que los vecinos pueden añadirse sin ninguna dificultad al vecindario actual, con un pequeño esfuerzo computacional. Sin embargo, para prevenir que la búsqueda irremediablemente se cicle entre las mismas soluciones, una *lista tabú* es introducida, la cual se describe a continuación.

Lista Tabú (Tabu List)

Esta lista mantiene un registro de las últimas *Ttenure* mejores modificaciones del valor del atributo de una solución x , las cuales han sido aprobadas durante el proceso de búsqueda. Un movimiento de x a x' será considerado *tabú* si éste ha sido ejecutado anteriormente y el valor de su atributo está contenido en la *lista tabú*.

Se ha visto que podría ser conveniente usar al mismo tiempo, diversos tamaños de la *lista tabú*. En este trabajo nos limitamos a la discusión del caso de un único tamaño de *lista tabú*, pero lo siguiente podría ser extendido al caso más general.

Como sabemos, el objetivo primordial de la *lista tabú* es prevenir que el procedimiento se cicle. Si la longitud de la lista es demasiado pequeña, este objetivo podría no ser conseguido; inversamente, un tamaño demasiado grande crea demasiadas restricciones, y se ha observado que las soluciones visitadas crecen con el incremento del tamaño de *lista tabú*. Usualmente, un tamaño adecuado de la lista podría ser fácilmente determinado; sin embargo, en algunos problemas de optimización se dificulta o inclusive se hace imposible encontrar un valor que prevenga el ciclamiento y al mismo tiempo no limite la búsqueda sobre todas las instancias del problema dado.

Una manera efectiva para sobrellevar esta dificultad, es usar una lista tabú con un tamaño variable. Así, para cada elemento de la lista corresponde un número de iteraciones que es acotado por valores mínimos y máximos dados. Sin embargo, el concepto de *lista tabú* puede definirse como restrictivo.

Complejidad Computacional: Como puede verse, el tiempo de cómputo está dominado por el trabajo de generación de los vecinos y cálculo de su correspondiente valor de la función objetivo. El número de vecinos está acotado por *Nei_Size* (parámetro definido por el usuario) y para calcular su valor de la función objetivo se aplica el procedimiento NDP, que toma, como ya se vio, $O(|A| \cdot N_p^2)$ operaciones. Si se implementa inteligentemente, no es necesario recalcular el valor del objetivo para una misma solución más de una vez. Como el ciclo del algoritmo está delimitado por *Iter_max*, NONDP_TS tiene una complejidad de $O(Iter_max \cdot Nei_Size \cdot |A| \cdot N_p^2)$.

Una vez que hemos descrito detalladamente cada una de las características principales del procedimiento heurístico NONDP_TS propuesto, damos paso al siguiente capítulo, en el cual presentamos una exhaustiva evaluación computacional, con la finalidad de probar la eficiencia de las técnicas propuestas.

CAPÍTULO 6

EVALUACIÓN COMPUTACIONAL

En este capítulo se desarrolla y describe una exhaustiva evaluación computacional, para estimar la efectividad en términos de calidad de las soluciones obtenidas y eficiencia de los recursos computacionales requeridos de los procedimientos propuestos en los Capítulos 4 y 5; por lo cual, una base de datos con múltiples y diversas instancias sobre diferentes tipos de topologías se introduce y describe en la siguiente sección; esto, con la finalidad de sustentar la evaluación computacional como contribución de este trabajo de investigación.

6.1 Descripción de la Base de Datos

El diseño y la implementación de esta base de datos tienen como objetivo fundamental la evaluación de las diferentes técnicas implementadas en el PMCC, previendo de esta forma una más eficiente y confiable medida de rendimiento. Y es que desde nuestro conocimiento, actualmente no hay base de datos con diversas topologías de redes sobre el PMCC, por lo que ésta resulta ser una contribución especial en este trabajo de tesis.

Aún cuando en el ámbito de la industria del gas natural existen infinidad de tipos de topologías, desde la perspectiva de la optimización en redes, se han clasificado dos grupos topológicos: no cíclicos (conteniendo a las topologías lineales y de árbol), y cíclicos.

6.1.1 Descripción del Nombre Mnemónico

Una característica importante en el diseño de la base de datos es determinar un nombre mnemónico para definir a cada topología por sus características principales,

como número de nodos, de estaciones compresoras, etc. En la Figura 6.1 se describe la nomenclatura que es utilizada para integrar el nombre de una topología que pertenece a nuestra base de datos.

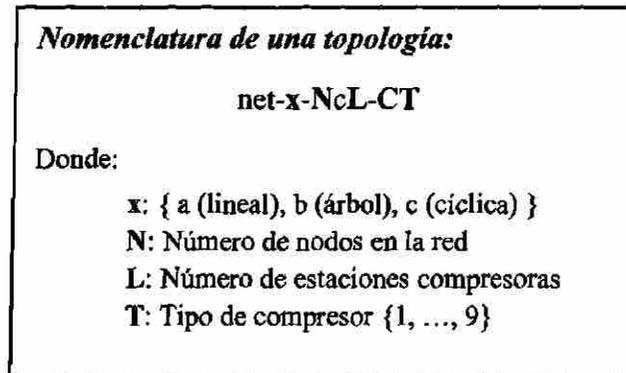


Figura 6.1: Descripción de nomenclatura de instancias.

Por ejemplo, la topología *net-c-15c5-C6*, indica que la instancia de red en cuestión es un sistema de red con estructura cíclica, conteniendo 15 nodos y 5 estaciones compresoras del tipo 6.

6.1.2 Topologías No Cíclicas: Lineales y de Árbol

En este trabajo, cuando nos referimos a topologías no cíclicas, describimos a dos tipos de topologías con estructuras definidas, que a diferencia de la perspectiva de la tipificación de las redes en general, nos adentramos a un punto de vista más específico de la optimización en redes de transmisión de gas, basándonos exclusivamente en la localización de las estaciones compresoras que comprenden los sistemas, para identificar a qué clase pertenece cada topología, a saber:

- Topología lineal o *gun-barrel* (Figura 6.2): Cuando en su estructura solamente se encuentran estaciones compresoras en forma secuencial, sin existir ninguna rama divergente al sistema que contenga un compresor.

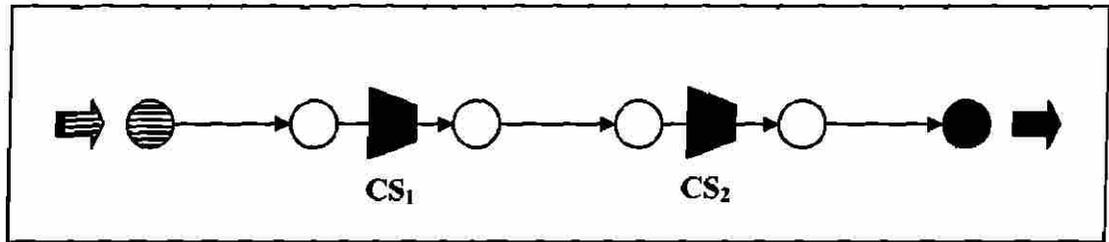


Figura 6.2: Topología no cíclica: Estructura lineal.

- Topología tipo árbol (Figura 6.3): Cuando en la distribución de las estaciones compresoras existen ramas disidentes que contienen al menos una estación compresora.

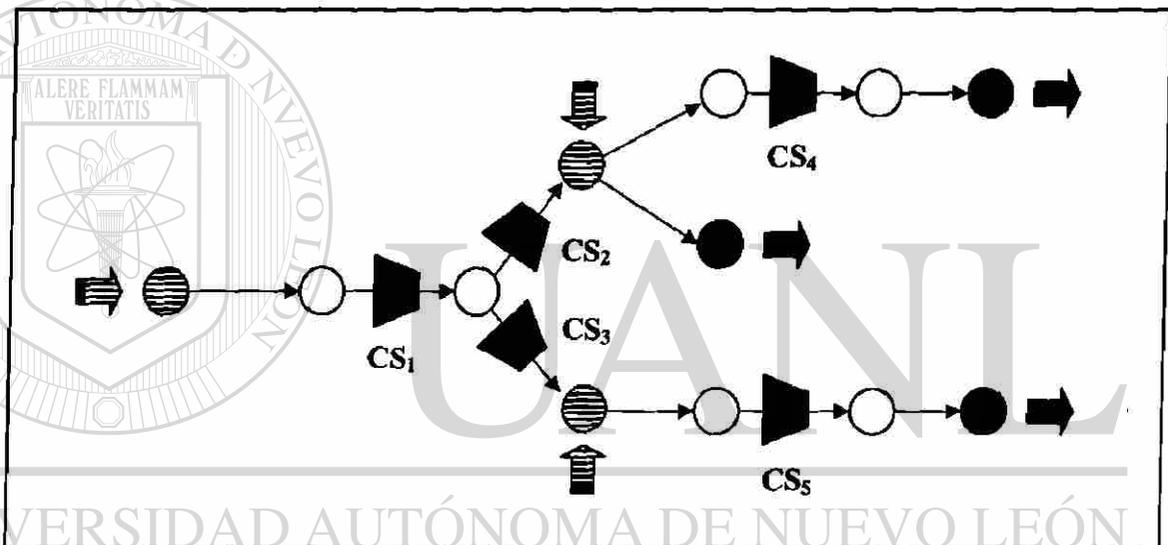


Figura 6.3: Topología no cíclica: Estructura tipo árbol.

6.1.3 Topologías Cíclicas

El término cíclico para las redes de transportación de gas natural, a diferencia de la definición clásica del término cíclico en redes generales, lo definimos con respecto a la localización de las estaciones compresoras sobre la red; para esto, clasificamos a una topología como cíclica (ver Figura 6.4), cuando existen dos o más rutas del nodo i al nodo j que contienen en cada una de ellas al menos una estación compresora.

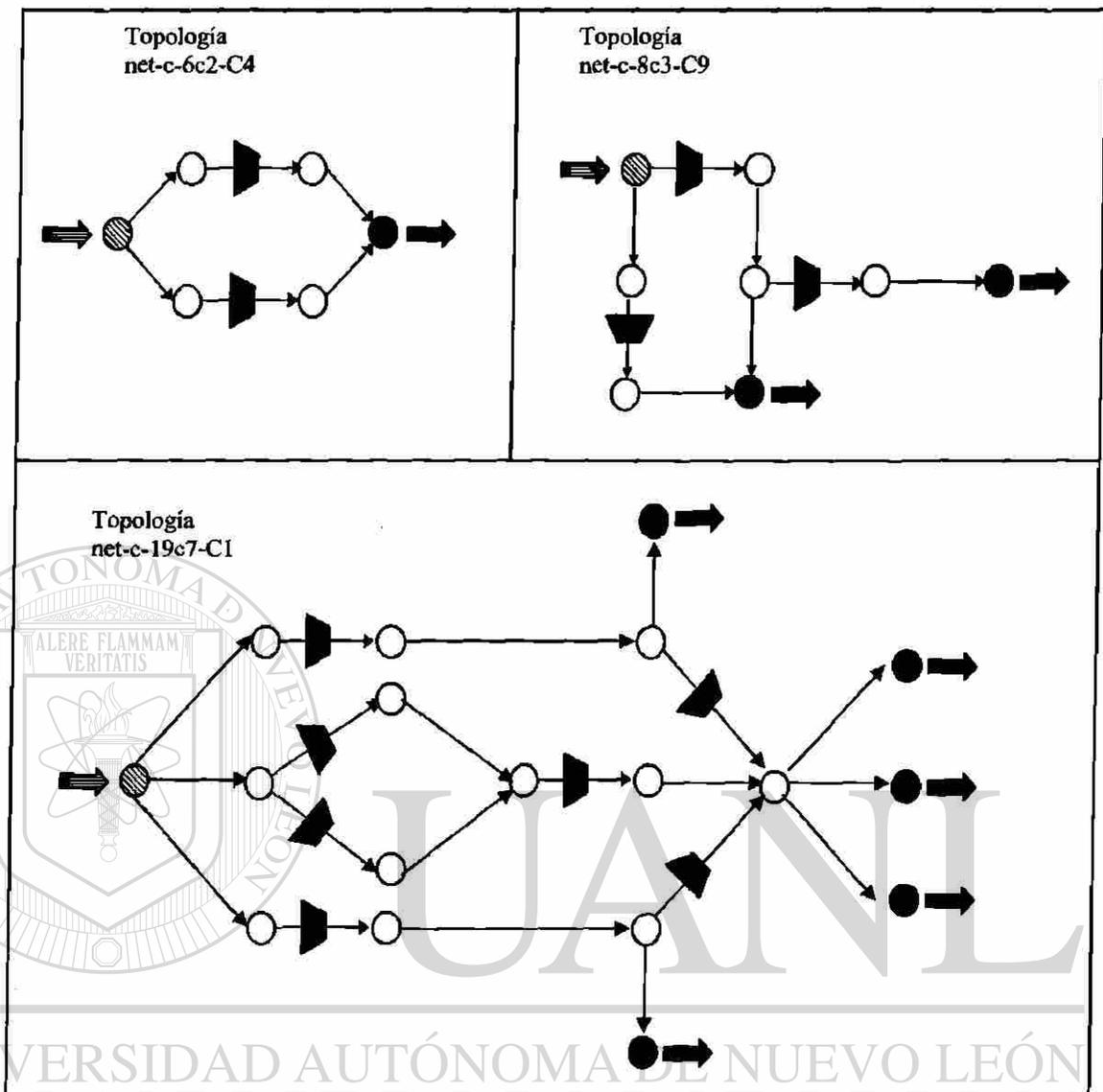


Figura 6.4: Ejemplos de instancias con estructuras cíclicas.

Por ejemplo, en las Figuras 6.2-6.4, un nodo rayado (mostrado con una flecha entrante a él) representa un punto suministro; un nodo negro (mostrado con una flecha saliente a él) representa un punto demanda, y un nodo blanco es un nodo de paso. Así, un arco dirigido que une a dos nodos representa a un arco ducto (una tubería), y un arco dirigido que contiene un trapecioide negro uniendo a dos nodos, representa un arco compresor (una estación compresora).

Nosotros hemos tratado de colocar un registro en la base de datos de todas las características que pueden ser importantes en cada sistema, pero siempre manteniendo la

posibilidad de integrar alguna otra información que pudiera ser relevante en un nivel más profundo en cada caso. Se comprende que los detalles de la información registrada en la base de datos, serán pertinentes sólo para los estudios de cierto fenómeno en cuestión.

La base de datos cuenta actualmente con 20 topologías no cíclicas (10 del tipo lineal y 10 del tipo árbol) y 10 topologías cíclicas, todas con diversas estructuras y conteniendo múltiples tipos de compresores. Siendo esta base de datos completamente accesible desde la siguiente dirección URL: <http://www.yalma.fime.uanl.mx/~pisis/>

6.2 Resultados Computacionales del Algoritmo NDP

En esta sección nos enfocamos concretamente en la evaluación de la eficiencia del algoritmo de búsqueda NDP, al cual hacemos referencia como parte esencial del método de solución presentado en el Capítulo 4. Esta evaluación está considerada sobre los dos tipos fundamentales de topologías de red. Para esto, consideramos tres experimentos computacionales específicos. El experimento 1, el cual es presentado en la Sección 6.2.1 tiene como objetivo principal evaluar el desempeño del algoritmo NDP únicamente sobre topologías no cíclicas, ya que a nuestra consideración, son en estas topologías donde las soluciones obtenidas son óptimas bajo cierta discretización del dominio factible. En contraparte, un segundo experimento presentado en la Sección 6.2.2 es realizado únicamente sobre topologías con estructuras cíclicas, ya que éstas demuestran ser las más complicadas; por lo cual, las soluciones obtenidas son típicamente clasificadas como factibles. Finalmente, un experimento 3 es presentado en la Sección 6.2.3 con el objetivo de verificar la calidad de las soluciones obtenidas sobre las topologías cíclicas del experimento anterior; al compararlas con las obtenidas por el Método de Gradiente Reducido Generalizado (GRG).

Nuestra evaluación está sustentada en una amplia base de datos con diversas configuraciones de flujos y 9 diferentes tipos de compresores centrífugos. Los detalles de esta base de datos pueden ser encontrados en [44]. Los procedimientos fueron codificados en lenguaje C++ y compilados con el compilador CC de Sun versión 2.0.1,

usando la opción de compilación `-o` en una estación de trabajo Sun Ultra 10, bajo el sistema operativo Solaris 7. Los tiempos de ejecución (CPU) se obtuvieron con el comando `time`.

6.2.1 Experimento 1: Evaluación sobre Redes No Cíclicas

En este primer experimento se decidió evaluar únicamente topologías no cíclicas: lineales o gun-barrel y de árbol, con el objetivo de obtener soluciones óptimas. La evaluación computacional estuvo basada para diferentes tamaños de malla con $\Delta p = \{20, 5, 1\}$; presentando tanto el valor objetivo óptimo como el tiempo de cómputo. Los resultados computacionales sobre redes lineales son mostrados en la Tabla 6.1. Las instancias probadas aparecen en la primera columna; las columnas siguientes muestran el tiempo CPU (seg.) y el valor de la función objetivo para cada uno de los tamaños de mallas probados. Similarmente, los resultados computacionales de las topologías de árbol, son mostrados en la Tabla 6.2.

Topología	$\Delta p = 20$ PSIG		$\Delta p = 5$ PSIG		$\Delta p = 1$ PSIG	
	t	Objetivo	t	Objetivo	t	Objetivo
net-a-5c1-C4	0.04	280,225.42	0.05	164,914.28	0.77	162,561.73
net-a-6c2-C1	0.07	2,305,101.07	0.17	1,925,665.87	6.48	1,802,876.73
net-a-6c2-C2	0.07	2,305,101.07	0.19	1,925,665.87	6.43	1,802,876.73
net-a-6c2-C3	0.04	1,065,709.48	0.29	399,126.31	15.03	343,249.56
net-a-6c2-C4	0.04	1,497,610.99	0.24	1,342,521.56	9.99	944,518.83
net-a-6c2-C5	0.05	3,650,144.05	0.51	3,259,430.33	21.77	3,044,941.55
net-a-6c2-C6	0.06	5,012,502.38	0.23	4,210,341.74	9.04	3,822,415.39
net-a-6c2-C7	0.03	1,269,300.15	0.16	694,097.81	3.39	559,114.25
net-a-6c2-C8	0.07	2,702,943.75	0.22	1,674,900.94	6.94	1,546,357.70
net-a-6c2-C9	0.01	5,043,312.43	0.07	4,066,445.48	1.77	4,012,593.03
net-a-8c3-C4	0.11	2,998,721.59	0.79	1,701,982.12	20.07	1,322,802.05
Topologías con estructuras especiales						
net-a-19c7-C4	0.07	4,652,595.76	0.91	3,340,853.26	39.31	3,046,053.04
net-a-12c1-C4	0.02	1,201,292.38	0.08	1,165,912.02	1.14	1,164,170.98
net-a-12c3-C4	0.05	4,085,543.14	0.49	3,334,569.62	24.31	3,213,108.46
net-a-18c1-C4	0.06	1,009,457.57	0.27	764,997.59	5.46	710,859.90
net-a-20c2-C4	0.06	3,018,255.30	0.70	2,261,313.45	40.64	2,099,562.45

Tabla 6.1: Resultados computacionales sobre redes lineales.

Observando los resultados presentados en la Tabla 6.1, podemos encontrar tiempos de ejecución del algoritmo NDP menores a un segundo, esto realmente demuestra la eficiencia de nuestro procedimiento al aplicarlo sobre topologías simples, aún cuando éstas son consideradas dentro de los tipos de redes más fáciles de resolver. Así, también puede advertirse cómo en base a una discretización más refinada de Δp , el valor objetivo disminuye considerablemente.

Topología	$\Delta p = 20$ PSIG		$\Delta p = 5$ PSIG		$\Delta p = 1$ PSIG	
	t	Objetivo	t	Objetivo	T	Objetivo
net-b-10c3-C1	0.08	5,151,874.09	1.20	4,926,640.97	60.07	4,294,872.70
net-b-10c3-C2	0.10	5,151,874.09	1.13	4,926,640.97	57.12	4,294,872.70
net-b-10c3-C4	0.11	9,558,262.12	0.50	7,861,105.38	20.54	7,216,523.11
net-b-10c3-C5	0.08	6,323,834.73	1.00	5,651,798.53	42.58	5,191,111.05
net-b-11c4-C4	0.22	10,803,366.07	1.99	9,201,821.43	91.11	8,972,057.94
net-b-12c4-C2	0.04	3,132,246.81	0.37	2,917,123.65	2.43	2,854,285.65
net-b-12c4-C3	0.05	11,516,568.35	0.14	10,832,946.19	3.38	10,597,128.61
net-b-12c4-C4	0.05	2,984,542.04	0.30	2,684,886.41	9.77	2,258,057.33
net-b-15c6-C4	0.21	8,234,564.12	1.74	6,342,045.90	70.69	5,282,939.80
net-b-41c14-C1	0.30	28,952,769.30	2.94	24,198,619.10	161.3	22,223,410.34
net-b-41c14-C2	0.23	28,952,769.30	2.98	24,198,619.10	161.12	22,223,410.34
net-b-41c14-C3	0.20	37,244,143.96	2.10	34,386,871.29	114.71	32,475,771.12
net-b-41c14-C4	0.67	36,476,126.38	2.65	31,298,741.09	129.24	30,809,127.04

Tabla 6.2: Resultados computacionales sobre redes tipo árbol.

Asimismo, pueden concluirse que el desempeño del algoritmo NDP puede considerarse eficiente para resolver topologías tipo árbol, dado que los resultados presentados en la Tabla 6.2 muestran tiempos (t) de ejecución menores a 1 segundo, y con la discretización más refinada ($\Delta p = 1$) el tiempo no excede a los 3 minutos. También puede observarse al igual que los resultados presentados en la Tabla 6.1, que el valor objetivo disminuye en base a una discretización más refinada de Δp . Por tanto, es pertinente aclarar que mientras más refinada sea la discretización, más exacta será la solución encontrada.

6.2.2 Experimento 2: Evaluación sobre Redes Cíclicas

En este experimento -el de mayor satisfacción-, al igual que en los experimentos anteriores, también se incluyen evaluaciones con 3 diferentes tamaños de mallas, con una $\Delta p = \{20, 5, 1\}$ sobre topologías cíclicas con diversas configuraciones de flujos. En la Tabla 6.3 se ilustran los resultados computacionales obtenidos en esta evaluación sobre diversos sistemas cíclicos con múltiples tipos de compresores. En la primera columna se presentan las topologías probadas, seguidas de las columnas que muestran el tiempo CPU (seg) y el valor de la función objetivo para cada uno de los diferentes tamaños de mallas.

Topología	$\Delta p = 20 \text{ PSIG}$		$\Delta p = 5 \text{ PSIG}$		$\Delta p = 1 \text{ PSIG}$	
	t	Objetivo	t	Objetivo	t	Objetivo
net-c-6c2-C1	0.03	2,279,711.77	0.16	1,884,060.48	3.24	1,852,252.53
net-c-6c2-C2	0.01	2,279,711.77	0.18	1,884,060.48	3.29	1,852,252.53
net-c-6c2-C3	0.02	1,200,668.26	0.20	1,041,265.76	4.07	1,020,842.29
net-c-6c2-C4	0.04	972,369.09	0.19	744,261.08	3.95	653,675.58
net-c-6c2-C5	0.02	2,999,798.61	0.19	2,803,372.66	3.36	2,716,611.07
net-c-6c2-C6	0.02	4,342,711.79	0.19	3,561,310.69	3.55	3,442,838.76
net-c-6c2-C7	0.001	1,082,391.42	0.18	850,674.93	2.95	788,218.02
net-c-6c2-C8	0.04	1,853,866.09	0.15	1,468,605.58	3.33	1,405,443.48
net-c-6c2-C9	0.03	1,499,524.36	0.17	1,294,408.69	2.95	1,243,179.03
net-c-10c3-C1	0.06	6,022,631.39	0.41	4,629,404.41	16.01	4,267,949.79
net-c-10c3-C2	0.04	6,022,631.39	0.38	4,629,404.41	15.32	4,267,949.79
net-c-10c3-C3	0.09	5,123,605.66	0.72	4,081,328.16	30.25	3,978,465.67
net-c-10c3-C4	0.05	6,076,776.75	0.31	4,652,223.35	7.31	4,360,146.30
net-c-10c3-C5	0.07	7,510,579.28	0.62	7,145,875.03	19.59	6,624,644.75
net-c-10c3-C6	0.11	11,908,871.1	0.61	10,996,571.93	15.51	10,766,001.9
net-c-10c3-C8	0.08	6,257,129.08	0.29	5,265,444.54	6.04	5,181,127.14
net-c-10c3-C9	0.12	4,223,456.43	0.48	3,315,770.23	11.17	3,281,774.65
net-c-15c5-C2	0.16	7,456,824.23	0.37	6,093,101.33	10.22	5,789,941.01
net-c-15c5-C4	0.18	3,143,098.87	0.45	2,612,943.12	18.43	2,154,914.02
net-c-19c7-C4	0.27	21,983,920.4	1.17	20,659,013.54	35.4	20,119,966.2

Tabla 6.3: Resultados computacionales sobre redes cíclicas.

En base a los resultados presentados en la Tabla 6.3 sobre topologías cíclicas, podemos observar cómo el comportamiento típico de la disminución del valor objetivo con respecto a una más refinada discretización, se ha conservado. En esta tabla podemos

observar que, aún cuando este tipo de topologías es considerado como el más difícil de resolver, los tiempos de CPU son muy razonables; por ejemplo, la prueba que requirió más esfuerzo ($\Delta p = 1$) nunca excedió a los 3 minutos. Demostrando con esto que la eficiencia del algoritmo NDP sobre redes cíclicas puede ser considerada alta.

6.2.3 Experimento 3. Algoritmo NDP vs Método GRG sobre Redes Cíclicas

Este experimento tiene como objetivo un análisis comparativo para evaluar el desempeño del algoritmo NDP contra una técnica tradicional de programación no lineal como el GRG.

Se sabe que la manera más eficiente de evaluar la calidad de una solución es determinando qué tan lejos o cerca se encuentra de la solución óptima global, como esto es imposible de realizar en nuestra experimentación, ya que para este problema en particular no es posible encontrar soluciones óptimas globales sobre redes cíclicas; por tanto, en su lugar realizamos una comparación de los resultados obtenidos por nuestro algoritmo NDP contra un método tradicional de optimización no lineal, como el método del GRG. Así, con esto proveemos una estimación del desempeño de nuestro algoritmo NDP con respecto a la calidad y tiempo de su solución.

Para realizar esta experimentación se utilizó el modelador algebraico GAMS [3] para implementar el modelo matemático del problema [10], y de esta forma, obtener el conjunto de soluciones a comparar con nuestro algoritmo sobre la base de datos completa.

En la Tabla 6.4 se muestran los resultados de la comparación entre los dos métodos. Las instancias probadas son mostradas en la primera columna; inmediatamente, el estatus, tiempo CPU (seg.) y el mejor valor objetivo encontrado son presentados para cada método en el análisis. En la columna del estatus una solución factible es representada por F, una solución óptima local es representada por L, y una solución no factible es representada por N.

Topología	Algoritmo NDP			Método GRG		
	Estatus	t	Objetivo	Estatus	t	Objetivo
net-c-6c2-C1	F	3.24	1,852,252.53	L	0.82	2,312,548.24
net-c-6c2-C2	F	3.29	1,852,252.53	L	0.82	2,312,548.24
net-c-6c2-C3	F	4.07	1,020,842.29	L	0.67	1,751,520.99
net-c-6c2-C4	F	3.95	653,675.58	L	1.01	1,393,061.12
net-c-6c2-C5	F	3.36	2,716,611.07	L	1.12	3,099,415.46
net-c-6c2-C6	F	3.55	3,442,838.76	N		
net-c-6c2-C7	F	2.95	788,218.02	L	0.57	988,998.79
net-c-6c2-C8	F	3.33	1,405,443.48	L	0.35	1,706,693.05
net-c-6c2-C9	F	2.95	1,243,179.03	N		
net-c-10c3-C1	F	16.01	4,267,949.79	N		
net-c-10c3-C2	F	15.32	4,267,949.79	N		
net-c-10c3-C3	F	30.25	3,978,465.67	N		
net-c-10c3-C4	F	7.31	4,360,146.30	L	1.03	5,610,932.12
net-c-10c3-C5	F	19.59	6,624,644.75	N		
net-c-10c3-C6	F	15.51	10,766,001.9	N		
net-c-10c3-C8	F	6.04	5,181,127.14	N		
net-c-10c3-C9	F	10.72	3,281,774.65	N		
net-c-15c5-C2	F	10.22	5,789,941.01	L	0.4	6,313,810.78
net-c-15c5-C4	F	18.43	2,154,914.02	L	0.18	3,555,353.60
net-c-19c7-C4	F	35.4	20,119,966.2	N		

Tabla 6.4: Desempeño de Algoritmo NDP contra el Método GRG.

En la Tabla 6.4 podemos observar cómo nuestro algoritmo NDP, aún cuando hasta esta etapa solamente entrega soluciones factibles en topologías cíclicas, estas soluciones se muestran de mayor calidad que las obtenidas por el método GRG, las cuales son consideradas óptimas locales. Quizás un factor importante que afecta el desempeño del método GRG es el punto inicial, ya que es en dependencia de éste, que el algoritmo ejecuta su proceso de búsqueda; por consiguiente, se observa inclusive que en 10 de las 20 topologías probadas este algoritmo falló en la búsqueda de soluciones, mientras que nuestro algoritmo NDP fue capaz de entregar soluciones a todas las instancias probadas. Resumiendo con esto, que el algoritmo NDP sobresale del método GRG en términos de la calidad de las soluciones.

6.2.4 Conclusiones

Una de las primeras conclusiones a la que podemos llegar es que el tamaño de la malla llega a ser un factor importante en términos de exactitud de la solución.

Debemos puntualizar que los resultados obtenidos sobre las instancias no cíclicas (ver Tablas 6.1 y 6.2) son considerados óptimos globales (bajo cierto dominio discretizado), mientras que las soluciones encontradas en las topologías cíclicas (ver Tabla 6.3) son clasificadas como soluciones factibles; sin embargo, éstas demostraron ser de mejor calidad que las obtenidas por el método GRG. Un punto importante es que los tiempos de CPU son muy razonables; por ejemplo, la prueba que requirió más esfuerzo de cómputo es ($\Delta p = 1$), y nunca excede de tres minutos.

Finalmente, en la Tabla 6.4 se muestra el excelente desempeño del algoritmo NDP contra el método GRG sobre redes cíclicas. Primero, el algoritmo NDP fue capaz de entregar soluciones para todas las instancias probadas, mientras que el método GRG falló para 10 de esas; segundo, pudo observarse que el algoritmo NDP sobresale al método GRG en términos de calidad en las soluciones obtenidas. Esto nos garantiza que la tolerancia de la calidad en la solución obtenida por el algoritmo NDP con respecto a la solución óptima global (por cierto, desconocida) es aún mejor que el método exacto probado.

6.3 Evaluación Computacional del Procedimiento de Búsqueda Tabú

En esta sección presentamos una evaluación computacional exhaustiva del procedimiento heurístico NONDP_TS descrito en la Sección 5.3.3, para determinar la calidad de las soluciones aproximadas obtenidas al PMCC, exclusivamente sobre topologías cíclicas.

Previamente, múltiples pruebas fueron realizadas sobre una amplia gama de instancias cíclicas para determinar los mejores valores sobre los parámetros externos de la heurística; esto, con el objetivo de minimizar el tiempo de cómputo en las evaluaciones computacionales del procedimiento heurístico NONDP_TS. Por tanto, en base a los resultados presentados por estas pruebas se fijó una discretización de $\Delta p = 5$ unidades PSIG sobre la técnica de DP no secuencial, la cual es asociada a una estrategia de memoria corta para encontrar soluciones aproximadas en topologías cíclicas. Este

valor se determinó dado que proporcionaba soluciones tan buenas como una discretización más refinada, pero con la ventaja de utilizar un esfuerzo computacional menor (tiempo CPU). (Para ver los resultados completos obtenidos en esta evaluación previa, ver Apéndice B). El procedimiento heurístico NONDP_TS propuesto fue implementado en lenguaje de programación C++ y diseñado sobre la base de datos obtenida de [44].

6.3.1 Experimento 1: Evaluación de los Parámetros

El primer experimento se realizó con el objetivo de encontrar, en base a una estimación conjunta del beneficio computacional y la calidad del espacio de búsqueda, los mejores valores de los parámetros básicos de TS. Esto, para demostrar cuán ventajoso resulta determinar previamente los parámetros que afectan directamente a la aplicación de un procedimiento en sus futuras evaluaciones.

Con la finalidad de una eficiente evaluación computacional que estimara la calidad de las soluciones del procedimiento heurístico NONDP_TS, se fijaron previamente los siguientes parámetros:

$$Iter_max = 100$$

Número máximo de iteraciones

$$\Delta p^{Vx} = 5$$

Dispersión de las soluciones en $V(x)$

Con estos parámetros fijos se procedió a efectuar la evaluación computacional de los siguientes parámetros de TS con tres diferentes valores:

$$Ttenure = \{5, 8, 10\}$$

Tamaño de la lista tabú

$$Nei_Size = \{20, 30, 40\}$$

Tamaño del vecindario

A continuación, presentamos los resultados obtenidos sobre cuatro topologías cíclicas tomadas al azar dentro de la base de datos. Sin embargo, esta experimentación se realizó sobre la base de datos completa obtenida de [44], presentado resultados similares a los mostrados en la Tabla 6.5.

Tenure	Nei_Size = 20					Nei_Size = 30					Nei_Size = 40				
	Tiempo (seg)	Iter	Valor Inicial	Mejor valor encontrado	%	Tiempo (seg)	Iter	Valor Inicial	Mejor valor encontrado	%	Tiempo (seg)	Iter	Valor Inicial	Mejor valor encontrado	%
Topología: <i>tree</i> <i>tree</i> <i>tree</i>															
5	326.35	7	744261.09	549990.76	26.1	485.69	7	744261.09	549990.76	26.1	642.75	7	744261.09	549990.76	26.1
8	323.65	37	744261.09	549990.81	26.1	480.01	37	744261.09	549990.81	26.1	671.89	37	744261.09	549990.81	26.1
10	318.37	67	744261.09	549990.76	26.1	482.15	56	744261.09	549991.97	26.1	627.18	20	744261.09	549995.36	26.1
Topología: <i>tree</i> <i>tree</i> <i>tree</i>															
5	275.10	98	2803372.66	2717572.36	3.15	411.71	98	2803372.66	2717572.36	3.15	544.58	98	2803372.66	2717572.36	3.15
8	271.13	10	2803372.66	2717572.36	3.15	408.51	10	2803372.66	2717572.36	3.15	537.71	10	2803372.66	2717572.36	3.15
10	269.24	76	2803372.66	2717572.54	3.15	404.83	76	2803372.66	2717572.54	3.15	535.70	76	2803372.66	2717572.54	3.15
Topología: <i>tree</i> <i>tree</i> <i>tree</i>															
5	270.67	7	1468605.588	1405226.98	4.51	411.13	7	1468605.588	1405226.98	4.51	545.14	7	1468605.58	1405226.98	4.51
8	271.14	11	1468605.588	1405226.98	4.51	405.79	10	1468605.58	1405226.98	4.51	526.25	10	1468605.58	1405226.98	4.51
10	268.07	83	1468605.588	1405226.99	4.51	395.07	94	1468605.58	1405227.05	4.51	557.71	92	1468605.58	1405228.03	4.51
Topología: <i>tree</i> <i>tree</i> <i>tree</i>															
5	395.63	92	5483509.96	4831584.06	13.4	598.06	56	5483509.96	4831575.46	13.4	804.04	77	5483509.96	4831565.70	13.4
8	401.04	93	5483509.96	4831796.92	13.4	604.88	79	5483509.96	4831562.25	13.4	825.10	79	5483509.96	4831562.25	13.4
10	413.24	85	5483509.96	4831633.05	13.4	622.41	92	5483509.96	4833344.31	13.4	823.29	76	5483509.96	4832108.44	13.4

Tabla 6.5: Resultados computacionales del análisis comparativo de los parámetros del procedimiento NONDP_TS.

La Tabla 6.5 muestra los resultados computacionales del análisis paramétrico realizado sobre la heurística de TS, en la cual se observan cuatro columnas principales. En la primera columna se especifica el tamaño de la *lista tabú* para los tres diferentes valores $Tenure = \{5, 8 \text{ y } 10\}$ probados. En las columnas restantes tres diferentes tamaños del vecindario $Nei_Size = \{20, 30 \text{ y } 40\}$ son evaluados; observando en cada uno de ellos 5 columnas que nos muestran el tiempo de CPU invertido en la instancia de red, número de iteración de la mejor solución encontrada, valor objetivo inicial, mejor valor objetivo encontrado y porcentaje de mejoramiento desde la solución inicial.

De los resultados presentados en la Tabla 6.5, podemos concluir que el mejor valor encontrado para el tamaño apropiado de la *lista tabú* fue de 8, ya que observamos en todas las topologías probadas, que tanto el tiempo de CPU como el número de la mejor iteración fueron menores con este valor; al mismo tiempo que las gráficas de desempeño de la Búsqueda Tabú presentaron una mayor variación en las soluciones encontradas (ver Apéndice C). Asimismo, el mejor valor observado en la tabla que determinaba el tamaño apropiado del vecindario fue de 20, pues aún cuando el tamaño

del vecindario era aumentado (30 ó 40) no incurría en un mejor valor objetivo, pero sí en un incremento considerable del esfuerzo computacional invertido (tiempo CPU).

Una vez determinados los mejores valores para cada parámetro de TS, debido a la obvia necesidad de evaluar la calidad de las soluciones encontradas por nuestro procedimiento heurístico NONDP_TS, se introduce en la siguiente sección un procedimiento de relajación del modelo matemático para obtener cotas inferiores.

6.3.2 Experimento 2: Evaluación de la Calidad de las Soluciones

Cota Inferior

Con la mira hacia una experimentación comparativa computacional que demuestre la calidad de las soluciones obtenidas por nuestro procedimiento heurístico NONDP_TS, introducimos en esta sección un procedimiento para obtener cotas inferiores.

Estas cotas inferiores se basan en una evaluación exhaustiva de las variables de decisión sobre un modelo relajado. Basándonos en un procedimiento de búsqueda sobre todo el espacio factible, obtenemos las presiones óptimas para cada flujo especificado dentro del rango de factibilidad de la variable de decisión x_{ij} para cada arco compresor.

Existen tres elementos dentro del modelo matemático (NLP) presentado en la Sección 3.2, que hacen al PMCC realmente difícil de resolver, a saber: (a) La no convexidad en la función objetivo g^{unit} ; (b) la no convexidad en el dominio de operación factible D^{unit} para cada compresor; y, (c) la presencia de restricciones en las tuberías.

Citando a la Sección 3.2, en la que el modelo matemático está dado por las ecuaciones (3.1)-(3.7), básicamente desarrollamos el procedimiento de evaluación exhaustiva para obtener cotas inferiores mediante una relajación del modelo matemático de la sección previamente citada. Así, implementamos el procedimiento de cotas inferiores relajando el modelo NLP al remover las ecuaciones (3.2), (3.3) y (3.4). Las dos primeras corresponden a la restricción del balance de flujo en cada nodo y la

capacidad en cada arco ducto. La última es una restricción de igualdad no lineal conocida como la ecuación de Osiadacz [28], que representa la dinámica del flujo de gas a través de cada ducto de la red, es decir, eliminamos la restricción en las tuberías. Una vez eliminadas estas restricciones, el problema relajado se vuelve separable en los compresores, es decir, basta con optimizar cada compresor por separado. Como el consumo de combustible en cada compresor es función de solo tres variables, es posible definir una discretización que permita hacer una evaluación por fuerza bruta y encontrar la solución óptima para cada compresor. En casos en que, por preprocesamiento es posible fijar el valor de flujo de antemano, la enumeración exhaustiva se hace solo sobre las otras dos variables.

Finalmente, con este modelo relajado procedemos a buscar el desempeño óptimo de las estaciones compresoras sobre ambos tipos de variables de decisión, aplicando un proceso exhaustivo de evaluación que utiliza una técnica de DP no secuencial (ver Sección 4.2.2) para encontrar las presiones óptimas sobre un rango discretizado de factibilidad de la variable del flujo másico X_{ij} para cada arco compresor en la red. Nosotros aplicamos este método exhaustivo sobre la base de datos completa. Es de esta manera como hemos desarrollado cotas inferiores que nos permiten evaluar la calidad de las soluciones obtenidas por nuestro procedimiento NONDP_TS.

No obstante la obvia exigencia de evaluar la calidad de la aplicación del algoritmo NONDP_TS, a continuación presentamos dos diferentes análisis comparativos dentro de una experimentación computacional.

DIRECCIÓN GENERAL DE BIBLIOTECAS

Análisis Comparativo 1

El primer análisis comparativo, cuyo objetivo es determinar el porcentaje de mejoramiento de la heurística NONDP_TS desde una solución de NDP simple, es ilustrado en la Tabla 6.6, en la cual en la primera columna se muestran las instancias probadas; la segunda columna muestra el valor objetivo de la solución NDP simple; la tercera columna muestra el mejor valor encontrado por el algoritmo NONDP_TS; y finalmente, en la última columna se presenta el porcentaje de mejoramiento desde la solución NDP simple.

<i>Topología</i>	<i>Solución NDP simple</i>	<i>Heurística NONDP_TS</i>	<i>(%) Mejoramiento desde la solución NDP simple</i>
net-c-6c2-C1	2,317,794.61	2,288,252.53	1.27
net-c-6c2-C4	1,394,001.99	1,393,001.99	0.07
net-c-6c2-C7	1,198,415.69	1,140,097.39	4.86
net-c-10c3-C2	6,000,240.25	4,969,352.82	17.18
net-c-10c3-C4	2,533,470.72	2,237,507.93	11.68
net-c-15c5-C2	6,006,930.42	4,991,453.59	16.90
net-c-15c5-C4	3,669,976.44	3,371,985.41	8.11
net-c-15c5-C5	8,060,452.17	7,962,687.43	1.21
net-c-17c6-C1	9,774,345.45	8,659,890.72	11.40
net-c-19c7-C4	12,019,962.22	8,693,003.78	27.67
net-c-19c7-C8	8,693,003.78	7,030,280.45	19.12

Tabla 6.6: Porcentaje de mejoramiento del procedimiento NONDP_TS desde la solución NDP simple.

Basándonos en los valores objetivos presentados en la Tabla 6.6, el porcentaje de mejoramiento del procedimiento heurístico sobre la solución NDP simple está definido como:

$$\frac{NDP \text{ simple} - NONDP_TS}{NDP \text{ simple}} \times 100,$$

Donde podemos observar que el porcentaje de mejoramiento de las soluciones obtenidas por el procedimiento NONDP_TS es mayor del 10% para 6 de 11 topologías probadas, e inclusive solo en una de ellas pudo encontrarse un porcentaje de mejoramiento menor al 1%.

Análisis Comparativo 2

La intención del segundo análisis comparativo es mostrar la exactitud de la solución; por lo cual, en esta experimentación hacemos uso de los resultados obtenidos mediante el procedimiento externo de evaluación exhaustiva donde se obtuvieron cotas inferiores.

Este experimento es justificado dado que obviamente, por las observaciones realizadas de los resultados obtenidos en la evaluación computacional de la Sección 6.2.3, los métodos de gradiente que han sido implementados recientemente para resolver redes cíclicas -como en el caso del método GRG-, presentan soluciones de menor calidad que las obtenidas por el algoritmo NDP, y dado que estas son las soluciones iniciales de nuestro procedimiento NONDP_TS, es lógico suponer que en el peor de los casos, se seguirá conservando la ventaja sobre los mismos.

Así finalmente, en la Tabla 6.7 presentamos los resultados obtenidos de esta evaluación computacional para verificar la calidad de las soluciones encontradas por nuestro procedimiento heurístico NONDP_TS sobre una amplia gama de diversas topologías cíclicas con diferentes tipos de compresores.

<i>Topología</i>	g^{TS}	g^{CI}	<i>dp (%)</i>
net-c-6c2-C1	2,288,252.53	2,287,470.58	0.03
net-c-6c2-C4	1,393,001.99	1,392,354.29	0.05
net-c-6c2-C7	1,140,097.39	949,909.48	8.67
net-c-8c3-C1	2,025,752.91	1,866,761.46	7.85
net-c-8c3-C5	2,675,004.32	2,537,154.37	5.15
net-c-10c3-C2	4,969,352.82	4,303,483.50	13.40
net-c-10c3-C4	2,237,507.93	1,737,775.98	22.33
net-c-13c5-C3	3,373,601.19	2,231,750.39	5.98
net-c-13c5-C4	5,812,350.98	3,138,412.32	17.68
net-c-15c5-C2	4,991,453.59	4,955,752.90	0.72
net-c-15c5-C4	3,371,985.41	3,103,697.48	7.96
net-c-15c5-C5	7,962,687.43	6,792,248.08	5.17
net-c-17c6-C4	22,483,581.84	22,416,202.47	0.30
net-c-17c6-C6	27,100,482.32	27,069,950.03	0.11
net-c-18c6-C2	4,372,791.28	4,297,196.46	1.73
net-c-18c6-C4	3,045,981.85	2,919,574.33	4.15
net-c-19c7-C2	11,943,865.02	11,210,746.13	6.14
net-c-19c7-C4	8,693,003.78	7,491,897.18	2.61
net-c-19c7-C6	6,045,434.86	5,627,506.51	6.91
net-c-19c7-C8	7,030,280.45	5,897,768.92	16.1

Tabla 6.7: Porcentaje relativo de error de las soluciones NONDP_TS vs cotas inferiores.

En la primera columna de la Tabla 6.7 se muestran las instancias con estructuras cíclicas probadas. La segunda y tercera columnas muestran el mejor valor g^{TS} encontrado por el procedimiento NONDP_TS y la cota inferior g^{CI} , respectivamente. La última columna muestra el porcentaje de la exactitud de la solución obtenida por el procedimiento NONDP_TS con respecto a su cota inferior.

Basándonos en los valores objetivos presentados en la Tabla 6.7, el porcentaje de desviación (dp) fue definido de la siguiente manera:

$$dp = \frac{g^{TS} - g^{CI}}{g^{CI}} \times 100$$

De acuerdo a la Tabla 6.7, es posible observar que en 17 de 20 topologías presentadas, la desviación del error de las soluciones obtenidas por el procedimiento NONDP_TS se encuentra a menos del 10% de la cota inferior; e inclusive, en 5 de ellas podemos observar que los porcentajes de desviación están a menos del 1%, lo cual es realmente significativo.

Las Figuras 6.5 y 6.6 muestran la convergencia del procedimiento NONDP_TS sobre las topologías net-c-6c2-C7 y net-c-10c3-C1, respectivamente, las cuales se obtuvieron de la base de datos en [44]. Estas figuras también muestran que el procedimiento heurístico propuesto típicamente converge en las primeras 50 iteraciones.

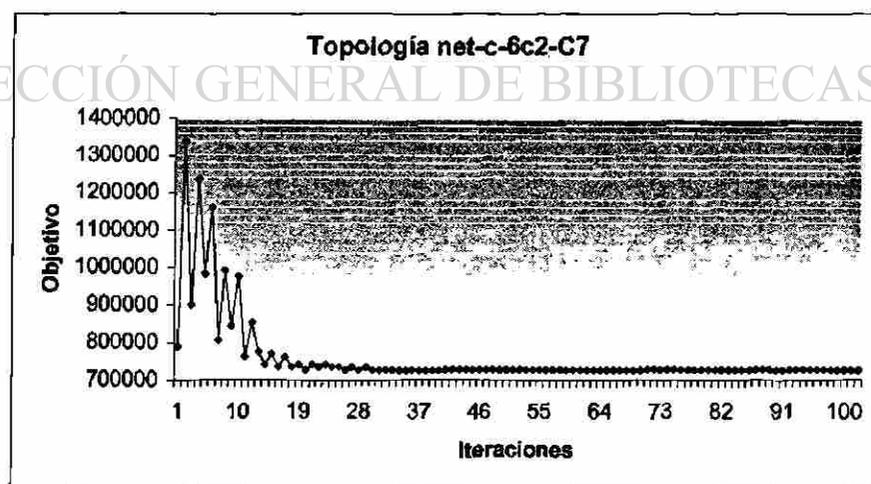


Figura 6.5: Convergencia de NONDP_TS en instancia net-c-6c2-C7.

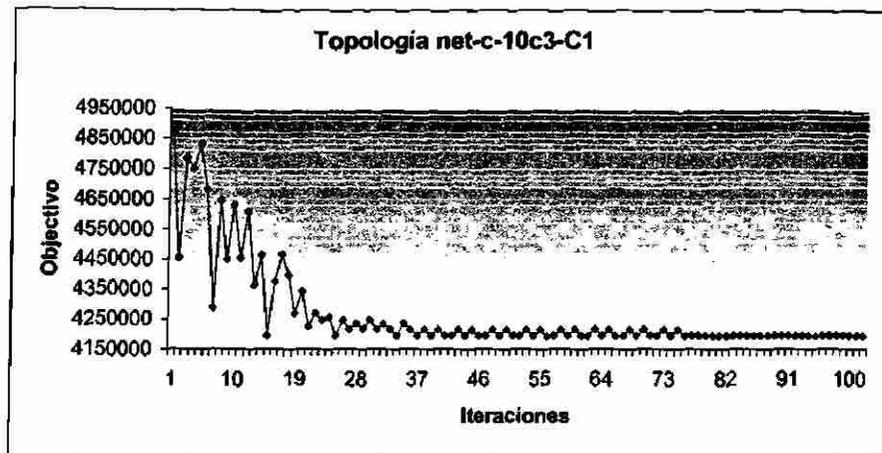


Figura 6.6: Convergencia de NONDP_TS en instancia net-c-10c3-C1.

6.3.3 Conclusiones

Concluyendo de las evaluaciones computacionales anteriores, podemos hacer mención que los mejores resultados se obtuvieron estableciendo los siguientes puntos en la ejecución del procedimiento heurístico NONDP_TS:

- ✓ Un número máximo de 100 iteraciones.
- ✓ Una dispersión de las soluciones del vecindario $V(x)$ de $\Delta p^{Vx} = 5$ unidades.
- ✓ Un tamaño fijo de la lista tabú de 8 iteraciones.
- ✓ Un tamaño fijo del vecindario $V(x)$ de 20 vecinos.

Estos valores estiman ser los mejores encontrados, puesto que con ellos el procedimiento heurístico pudo acceder a un mayor número de soluciones desde una solución NDP simple dada.

Una punto importante fue la relación entre el tamaño del vecindario $V(x)$ y la lista tabú, ya que esto permitió aceptar un mayor número de soluciones de peor valor, obligando de esta manera a la Búsqueda Tabú salir de un mínimo local.

En base a las observaciones similares obtenidas sobre la convergencia de nuestro

procedimiento NONDP_TS, con relación a la base de datos completa podemos concluir al tomar como parámetro el número de la mejor iteración, que *cantidad no es sinónimo de calidad*. (Para un análisis más detallado de las pruebas realizadas, ver Apéndice A).

Vale la pena hacer notar que una variante de la heurística de búsqueda tabú, como una diversificación continua no proporciona ningún mejoramiento adicional. La explotación de una memoria adaptativa dentro de nuestra heurística, la cual es dirigida a favorecer la exploración de nuevas áreas de interés en el espacio de búsqueda, explica la redundancia de la aproximación de diversificación continua.

La eficiencia de las soluciones obtenidas en la ejecución del procedimiento heurístico NONDP_TS se encuentra alrededor del 15% de mejora sobre las instancias cíclicas probadas, lo cual es aceptable considerando la complejidad del problema y el hecho de que aún con un mejoramiento marginal del 1% llega a ser realmente significativo en términos económicos.

La implementación de técnicas basadas en búsqueda tabú depende de la forma en que éstas sean dirigidas, esperando siempre que la heurística esté dotada de la suficiente inteligencia para dirigirse y escapar de óptimos locales que se presenten.

Finalmente, como la complejidad asociada a este problema de estudio es considerada muy alta, las aproximaciones heurísticas pueden verse como las mejores opciones de solución al PMCC. Aún en el caso de que nuestro procedimiento heurístico pudiera resultar en un esfuerzo computacional mayor que los métodos tradicionales, si este produce un pequeño mejoramiento en la solución, puede ser fácilmente justificado dado que el costo involucrado en la transportación de gas natural está en el orden de miles de millones de dólares. Por lo que, aún con un porcentaje marginal en la calidad de la solución de nuestro procedimiento, esto resulta muy conveniente para la industria del gas natural.

Ahora bien, como parte final de este trabajo de tesis, en el siguiente capítulo presentamos las conclusiones y aportaciones generales concernientes al desarrollo y aplicación de nuestra investigación, la cual hace referencia a la optimización en redes de transporte del gas natural.

CAPÍTULO 7

CONCLUSIONES Y APORTACIONES

En esta tesis hemos presentado una metodología de solución para resolver el PMCC en sistemas de tuberías de gas natural, la cual consiste primeramente de una fase de preprocesamiento que incluye una técnica de reducción, seguida de fases de búsqueda basadas en un algoritmo de Grafo Reducido, una técnica de DP no secuencial y un procedimiento heurístico de Búsqueda tabú. Las heurísticas son particularmente indicadas en este caso dado que los métodos tradicionales no pueden actualmente ser eficientemente aplicados en instancias con estructuras cíclicas.

7.1 Conclusiones Generales

Fue posible observar con mucha claridad el impacto positivo que se logró con la aplicación de las técnicas de reducción y de DP no secuencial, así como la implementación de una Búsqueda Tabú para la solución de este problema. La eficacia de la metodología no solo se encuentra en otorgar soluciones de mayor calidad para la base de datos completa, sino también en reducir el esfuerzo computacional empleado por el procedimiento de búsqueda.

En todas las evaluaciones computacionales realizadas sobre los dos tipos fundamentales de topologías de red se pudo constatar el excelente desempeño de la metodología de solución, obteniendo soluciones óptimas globales en todas las instancias no cíclicas y soluciones aproximadas de muy buena calidad en topologías cíclicas. Nuestros resultados computacionales demostraron empíricamente cómo la estructura del

problema pudo ser eficientemente explotada por nuestro algoritmo de búsqueda NDP, el cual emplea una técnica de DP no secuencial.

Concluimos también, de la observación realizada en el experimento 2 del algoritmo NDP (ver Secciones 6.2.1 y 6.2.2), que mientras más refinada sea la discretización del dominio de operación factible utilizado, ésta corresponde proporcionalmente a soluciones más exactas, sin notar un incremento relevante en el esfuerzo computacional del algoritmo (tiempo de CPU). Asimismo, se realizó un estudio computacional de estos resultados en un tercer experimento (ver Sección 6.2.3), para comparar su exactitud contra un método tradicional de optimización no lineal, como el método GRG; esto, con el propósito de demostrar, de la manera más eficiente posible, la calidad de los resultados obtenidos por la metodología de solución. Se concluyó que evidentemente el algoritmo NDP sobresalió del método GRG, demostrando un excelente desempeño en los dos tipos fundamentales de topologías de redes: cíclicas y no cíclicas.

Debemos puntualizar que el algoritmo NDP ha sido aplicado a una amplia gama de diversas instancias de redes, representando decenas de diferentes sistemas de tuberías sobre una gran variedad de condiciones de flujo, obteniendo uniformemente buenos resultados. Concluimos entonces que este algoritmo de búsqueda no solamente encuentra soluciones realmente buenas, ya que un mejoramiento marginal del 1% implicaría millones de dólares ahorrados, sino que además reduce los recursos usados por la computadora (tiempo CPU). Esto representa una contribución significativa al PMCC incurrido por las estaciones compresoras en el curso de la transmisión del gas natural, especialmente cuando se obtuvieron excelentes resultados sobre redes cíclicas, donde las aproximaciones anteriores han tenido poco éxito.

Un punto importante que debemos reiterar es que las soluciones obtenidas sobre todas las instancias de redes no cíclicas son consideradas óptimas globales, ya que el conjunto de flujos asignado sobre éstas puede ser determinado de forma única [41]. En contraste, sobre las instancias de redes cíclicas esto no es posible, por lo que las soluciones obtenidas sobre éstas se declaran como soluciones aproximadas, por cierto, de buena calidad.

En esta propuesta metodológica podemos hacer mención a dos períodos de optimización consecutivos muy marcados. En el primer período observamos que la optimización puede verse duramente involucrada con las variables de decisión de las presiones (p_i) en cada nodo de la red. Consecutivamente, un segundo período refleja el esfuerzo de optimización en los dos tipos de variables de decisión (x_{ij} y p_i) considerados en el modelo, ya que en base a una *estrategia de memoria corta* del procedimiento heurístico de TS, se busca el mejor conjunto de flujos sobre el entorno factible de asignación en cada arco, al tiempo que es asociada con una *técnica de DP no secuencial* para obtener las presiones óptimas en cada conjunto de flujos muestreado.

Un punto culminante y muy satisfactorio en este trabajo de tesis ha sido la implementación del procedimiento heurístico NONDP_TS de Búsqueda Tabú, ya que además de mejorar la eficiencia en el proceso de exploración entregando soluciones aproximadas de alta calidad, también mantuvo una información histórica relacionada a la evolución de la búsqueda. Así, esta información fue usada para guiar el movimiento de una solución a otra, evitando caer en un óptimo local; siendo ésta una de las características más importantes de nuestra heurística NONDP_TS propuesta. Asimismo, las soluciones encontradas con este procedimiento heurístico fueron comparadas contra cotas inferiores para demostrar su calidad, observando resultados satisfactorios de excelente calidad, muy por encima de los métodos tradicionales de optimización no lineal existentes.

Finalmente, es pertinente aclarar que todos los resultados obtenidos han sido avalados sobre una amplia gama de diversas instancias topológicas de sistemas de transmisión de gas natural, con diferentes combinaciones de flujos y diferentes rangos de factibilidad. Asimismo, un hecho relevante que demuestra aún más la validez de esta investigación, es que todos los datos relacionados a los compresores fueron provistos de compresores reales entregados por una firma consultora en la industria de gasoductos.

7.2 Aportaciones Científicas

Las contribuciones científicas de esta tesis fueron:

- Implementación de un **método de solución** que integra una fase de **preprocesamiento** que incluye una *técnica de reducción*, una fase de **asignación de flujos** basada en una *técnica de asignación "clásica"* y un *algoritmo de grafo reducido*, y una **fase de búsqueda de presiones óptimas** basada en una *técnica de DP no secuencial*; esto, con la finalidad de obtener soluciones óptimas globales sobre redes no cíclicas y soluciones factibles iniciales de alta calidad sobre redes cíclicas.
- Desarrollo e implementación de un **procedimiento heurístico de Búsqueda Tabú** enfocado a encontrar soluciones aproximadas de alta calidad sobre redes cíclicas, basándose en puntos factibles iniciales entregados por el *algoritmo NDP* (contribución de esta tesis). Se demostró con extensas evaluaciones computacionales que este procedimiento heurístico produce mejores resultados que los métodos tradicionales como el *gradiente reducido generalizado*, por mencionar alguno. Fundamentándonos en lo mejor de nuestro conocimiento, hacemos mención que esta es la primera aplicación de una *meta-heurística de Búsqueda Tabú* a este problema en particular.
- Elaboración de una amplia base de datos con diversas instancias de sistemas de transmisión de gas natural, incluyendo topologías de redes cíclicas y no cíclicas.

Los resultados de este trabajo han sido publicados en las siguientes revistas.

- C. Borraz-Sánchez* y R. Z. Ríos-Mercado. A non-sequential dynamic programming approach for natural gas network optimization. *WSEAS Transactions on Systems*. 3(4):1384-1389, 2004.
- C. Borraz-Sánchez* y R. Z. Ríos-Mercado. A procedure for finding initial feasible solutions on cyclic natural gas networks. En *Proceedings of the 2004 NSF Design, Service and Manufacturing Grantees and Research Conference*. Dallas, EUA, Enero 2004.

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

Además, han sido presentados en los siguientes foros.

- “A Non-Sequential Dynamic Programming Approach for Natural Gas Network Optimization”. 6th WSEAS International Conferences on Algorithms, Scientific Computing, Modelling and Simulation. Cancún, México, Mayo 2004.
- “Una Técnica de Programación Dinámica No-Secuencial para el PMCC en Redes Cíclicas de Transmisión de Gas Natural: Un Estudio Computacional”. Seminario de Investigación. Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León (UANL), San Nicolás de los Garza, N.L., Marzo 2004.
- “A Non-Sequential Dynamic Programming Technique for Minimizing Fuel Consumption on Cyclic Pipeline System: A computational study”. X Escuela Latino-Americana de Verano de Investigación de Operaciones, ELAVIO. Montevideo, Uruguay, Febrero 2004.
- “Procedimiento para Encontrar Soluciones Factibles Iniciales en Redes Cíclicas de Transmisión de Gas Natural”. Seminario de Investigación. Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León (UANL), San Nicolás de los Garza, N.L., Noviembre 2003.
- “Diseño de una Metaheurística para la Minimización del Consumo de Combustible en una Red de Gas Natural”. Seminario de Investigación. Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León (UANL), San Nicolás de los Garza, N.L., Mayo 2003.

7.3 Recomendaciones para Futuras Investigaciones

Con la expectativa de continuar la búsqueda de alternativas que permitan acercarse aún más a la solución óptima de este problema, se establecen los siguientes planteamientos para trabajos futuros:

Con respecto al PMCC sobre sistemas de tuberías de redes de gas natural, se sabe de antemano que pueden existir diversas variaciones del problema. Una en particular es la que se describió en esta investigación, tratando a un modelo bajo cierto grupo de

suposiciones específicas. Una de estas suposiciones, quizás considerada en el modelo y desarrollo de nuestra investigación como la más significativa, es la de tratar con un modelo en estado estable, permitiéndonos de esta manera aplicar técnicas de solución no tradicionales a sistemas de redes de transmisión de gas natural que han estado operando por una relativa cantidad grande de tiempo. En contraste, un reto formidable sería resolver este problema asumiendo un modelo transiente, es decir, en un proceso cuya evolución en el tiempo sea aleatoria (dependiente del tiempo), implicando con ello aumentar la dimensionalidad y complejidad del mismo. Esto es actualmente uno de los mayores retos en la industria del gas natural.

Otra suposición importante es que este modelo es determinístico, siendo un punto muy interesante el plantearse el problema como un modelo estocástico, basándose en los posibles recursos y demandas.

Por supuesto, cabe la posibilidad de proponer diferentes métodos de solución que impliquen la adquisición de nuevas técnicas para explotar de una manera muy distinta la estructura matemática del problema, con la esperanza de explorar mejores espacios de solución. Quizás un método que debería proponerse, sería el tratar al problema desde la perspectiva de una optimización global [20], en el cual se recurriría a la rama en la que se estudian técnicas enumerativas de solución, para encontrar óptimos globales en problemas no convexos.

Con respecto al procedimiento heurístico de Búsqueda Tabú:

Principalmente, una propuesta que muestra ser una buena alternativa para un trabajo futuro, sería la implementación de este procedimiento heurístico bajo diferentes estrategias de construcción de la vecindad $V(x)$. Una nueva estrategia sería modificar otros componentes de la topología de red cíclica, como las presiones en cada nodo, y así realizar una búsqueda más extensa (aunque complicada) en el espacio de soluciones.

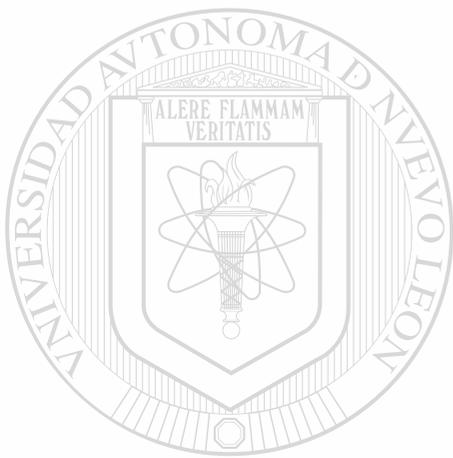
También se propone como trabajo futuro, la implementación de dos estrategias básicas en los modelos de TS, a saber: las estrategias de intensificación y diversificación. La primera podría basarse en la modificación del procedimiento de

búsqueda mediante la selección de nuevas reglas que permitan “concentrarse” en regiones que muestren soluciones atractivas. Por lo que, el uso de heurísticas más elaboradas, ampliación del vecindario o inclusive la utilización de métodos exactos, pueden verse como estrategias clásicas para intensificar la búsqueda. Es también posible ejecutar una intensificación basada en memoria de largo plazo. Cada solución o movimiento puede ser caracterizado por un conjunto de componentes. Los componentes de *buenos movimientos* o *buenas soluciones*, son mantenidas en memoria. Durante la fase de intensificación, las soluciones o los movimientos son evaluados tomando en cuenta la cantidad de sus *buenos componentes*. Esta memoria de largo plazo puede ser vista como un proceso de aprendizaje.

En cuanto a la estrategia de diversificación podríamos referirla a reglas que incrementen la capacidad de “revisar” a nivel macro aquellas zonas que no han sido visitadas por medio de la búsqueda de soluciones, que difieran lo más posible de aquellas que han sido previamente revisadas. Dado que la importancia de la diversificación está en evitar que una región grande del espacio de solución permanezca sin explorarse, la manera más simple para contrarrestar esto, podría ser la ejecución de diferentes puntos iniciales. Otra manera para garantizar la exploración de regiones no visitadas, es penalizar los movimientos ejecutados o soluciones frecuentemente visitadas. Cuando las soluciones que se consideran factibles tienen que satisfacer con exactitud un conjunto de restricciones, éstas pueden verse como montañas inquebrantables de altura infinita; por lo que, relajando estas restricciones y penalizando la trasgresión correspondiente, estas “*montañas*” pueden reducir su altura considerablemente. Es entonces posible librar esta barrera y alcanzar “*valles*” menos abruptos en pocos pasos. Durante esta etapa de diversificación, las soluciones visitadas no son necesariamente factibles (dado que las restricciones han sido relajadas). Para obtener nuevamente una solución factible, la penalidad de la violación de las restricciones relajadas debe ser gradualmente incrementada.

Es así, como finalmente hemos llegado al punto concluyente en este trabajo de tesis, donde destacamos desde el marco científico, que se ha demostrado empíricamente

que este problema de optimización en redes, al cual hemos definido particularmente como el PMCC en sistemas de redes de transporte de gas natural (*NP-completo*), pudo resolverse eficientemente al desarrollarse y aplicarse un algoritmo no exponencial (véase Apéndice D), el cual entregó soluciones de muy buena calidad, sobrepasando en mucho a los métodos tradicionales de programación no lineal existentes.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

BIBLIOGRAFÍA

- [1] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, EUA, 1957.
- [2] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, N. J., EUA, 1987.
- [3] A. Brooke, D. Kendrick y A. Meeraus. *GAMS: A User's Guide, Release 2.25*. The Scientific Press, San Francisco, EUA, 1992.
- [4] R. G. Carter. Pipeline optimization: Dynamic programming after 30 years. En *Proceedings of the PSIG Meeting*, pp. 1-19, Denver, EUA, Octubre 1998.
- [5] D. Cobos Zaleta. *Modelos de Optimización Entera Mixta No Lineal en Sistemas de Transporte de Gas Natural*. Tesis de Maestría, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, México, Noviembre 2003.
- [6] 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*, artículo A48-01, pp. 1-9. Concepción, Chile, Octubre 2002.
- [7] S. A. Cook. The Complexity of Theorem-proving Procedures. *Proceedings 3rd Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, pp. 151-158, New York, 1971.
- [8] N. Creignou, S. Khanna y M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. SIMM Monografías sobre Matemáticas Discretas y Aplicaciones. New Jersey, EUA, 2001.
- [9] A. Díaz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato y F. T. Tseng. *Optimización Heurística y Redes Neuronales*. Editorial Paraninfo, Madrid, España, 1996.
- [10] H. J. Flores-Villarreal y R. Z. Ríos-Mercado. Computational experience with a GRG method for minimizing fuel consumption on cyclic natural gas networks. En N. E. Mastorakis, I. A. Stathopoulos, C. Manikopoulos, G. E. Antoniou, V. M.

- Mladenov y I. F. Gonos, editores, *Computational Methods in Circuits and Systems Applications*, pp. 90-94. WSEAS Press, Atenas, Grecia, 2003.
- [11] H. J. Flores-Villarreal y R. Z. Ríos-Mercado. Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions. En *Proceedings of the International Applied Business Research Conference*, artículo 352, pp.1-7. Acapulco, México, Marzo 2003.
- [12] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization Fundamentals and Applications*. Oxford University Press, New York, EUA, 1995.
- [13] M. R. Garey y D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, EUA, 1979.
- [14] B.J. Gilmour, C.A. Luongo y D.W. Schroeder. Optimization in natural gas transmission networks: A tool to improve operational efficiency. Reporte Técnico, Stoner Associates Inc., presentado en la Tercera Conferencia sobre optimización SIAM, Boston, EUA, Abril 1989.
- [15] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5(3):533-549, 1986.
- [16] F. Glover y M. Laguna. *Tabu Search*. Kluwer, Boston, EUA, 1997.
- [17] F. Glover, M. Laguna. OpenTS – Java Tabu Search, sitio web <http://faculty.bus.olemiss.edu/fglover/>
<http://www.worldscinet.com/ijitdm/01/sample/S0219622002000026.html>
- [18] I. E. Grossmann, J. Viswanathan, A. Vecchiotti, R. Raman y E. Kalvelagen. *GAMS/DICOPT: A Discrete Continuous Optimization Peckage*. Carnegie Mellon University and GAMS Development Corporation, Washington, EUA, Noviembre 2001.
- [19] R. Harder. OpenTS - Java Tabu Search, sitio web <http://www.iHarder.net>
- [20] R. Horst, P. M. Pardalos y N. V. Thoai. *Introduction to Global Optimization*. Kluwer, Dordrecht, Holanda, 1995.
- [21] J. T. Jefferson. Dynamic programming. *Oil and Gas Journal*, pp. 102-107, Mayo 1961.
- [22] S. Kim, R. Z. Ríos-Mercado y E. A. Boyd. A heuristic for minimum cost steady-state gas transmission networks. En *Proceedings of the 25th International*

- Conference on Computers & Industrial Engineering*, New Orleans, EUA, Marzo 1999.
- [23] S. Kim, R. Z. Ríos-Mercado y E. A. Boyd. Heuristics for minimum cost steady-state gas transmission networks. En M. Laguna y J. L. González-Velarde, editores, *Computing Tools for Modeling, Optimization, and Simulation*, capítulo 11, pp. 203–213. Kluwer, Boston, EUA, 2000.
- [24] H. S. Lall y P. B. Percell. A dynamic programming based gas pipeline optimizer. En A. Bensoussan y J. L. Lions, editores, *Analysis and Optimization of Systems*, pp. 123-132, Springer-Verlag, Berlín, Alemania, 1990.
- [25] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan y D. Shmoys. Sequencing and scheduling: Algorithms and complexity. En S. S. Graves, A. H. G. Rinnooy Kan y P. Zipkin, editores, *Handbook in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory*, 445-522. North-Holland, New York, 1993.
- [26] A. V. Levitin. *Introduction to the Design and Analysis of Algorithms*. Addison-Wesley, N. Y., EUA, 2002.
- [27] B. Melián, J. A. Moreno y J. M. Moreno. Metaheuristics: A global view. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 6(19):7-28, 2003.
-
- [28] A. J. Osiadacz. *Simulation and Analysis of Gas Networks*. Gulf Publishing Company, Houston, EUA, 1987.
- [29] A. J. Osiadacz. Dynamic optimization of high pressure gas networks using hierarchical systems theory. En *Proceedings of the 26th PSIG Annual Meeting*, San Diego, EUA, Octubre 1994.
- [30] A. J. Osiadacz y M. Górecki. Optimization of pipe sizes for distribution gas network design. En *Proceedings of the 27th PSIG Annual Meeting*, Albuquerque, EUA, Octubre 1995.
- [31] A. J. Osiadacz y S. Swierczewski. Optimal control of gas transportation systems. En *Proceedings of the 3rd IEEE Conference on Control Applications*, pp. 795–796, Agosto 1994.

- [32] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, N. Y., EUA, 1995.
- [33] C. H. Papadimitriou y K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Publicaciones Dover, EUA, 1999.
- [34] P. B. Percell y M. J. Ryan. Steady-state optimization of gas pipeline network operation. En *Proceedings of the 19th PSIG Annual Meeting*, Tulsa, EUA, Octubre 1987.
- [35] D.T. Pham y D. Karboga. *Intelligent Optimisation Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer, New York, EUA, 2000.
- [36] M. Pinedo. *Scheduling, Theory, Algorithms, and Systems*. Segunda Edición. Prentice-Hall, Englewood Cliffs, EUA, 2002.
- [37] V. J. Rayward-Smith. *A First Course in Computability*. Blackwell, 1986.
- [38] M. Resende y J. L. González-Velarde. GRASP: Procedimientos de búsqueda miopes aleatorizados y adaptativos. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 6(22):42-61, 2003.
- [39] M. G. C. Resende y C. C. Ribeiro. Greedy randomized adaptive search procedures. En F. Glover y G. G. Kochenberger, editores, *Handbook of Metaheuristics*, Kluwer, Boston, EUA, 2003.
-
- [40] R. Z. Ríos-Mercado. Natural gas pipeline optimization. En P. M. Pardalos y M. G. C. Resende, editores, *Handbook of Applied Optimization*, capítulo 18.8.3, pp. 813-825. Oxford University Press, New York, EUA, 2002.
- [41] R. Z. Ríos-Mercado, S. Wu, L. R. Scott y E. A. Boyd. A reduction technique for natural gas transmission network optimization problems. *Annals of Operations Research*, 117(1-4):217-234, 2002.
- [42] D.W. Schroeder. Hydraulic analysis in the natural gas industry. En J. J.-W. Chen y A. Mital, editores, *Advances in Industrial Engineering Applications and Practice I*, pp. 960-965, Cincinnati, EUA, 1996.
- [43] Y. Villalobos Morales. *Pre-procesamiento de un Problema de Optimización de Redes de Gas Natural*. Tesis de Maestría, Universidad Autónoma de Nuevo

León, San Nicolás de los Garza, México, Octubre 2002.

- [44] Y. Villalobos-Morales, D. Cobos-Zaleta, H. J. Flores-Villarreal, C. Borraz-Sánchez y R.Z. Ríos-Mercado. On NLP and MINLP formulations and preprocessing for fuel cost minimization of natural gas transmission networks. En *Proceedings of the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference*. Birmingham, Alabama, EUA, Enero 2003.
- [45] Y. Villalobos Morales y R. Z. Ríos Mercado. Pre-procesamiento efectivo de un problema de minimización de combustible en sistemas de transporte de gas natural. Reporte Técnico PISIS-2002-01, FIME, UANL, San Nicolás de los Garza, NL, Noviembre 2002.
- [46] P. J. Wong y R. E. Larson. Optimization of natural gas pipeline systems via dynamic programming. *IEEE Transactions on Automatic Control*, AC-13(5):475-481, 1968.
- [47] S. Wu. *Steady-State Simulation and Fuel Cost Minimization of Gas Pipeline Networks*. Tesis doctoral, University of Houston, Houston, EUA, Agosto 1998.
- [48] S. Wu, E. A. Boyd y L. R. Scott. Minimizing fuel consumption at gas compressor stations. En J. J.-W. Chen y A. Mital, editores, *Advances in Industrial Engineering Applications and Practice I*, pp. 972-977, Cincinnati, EUA, 1996.
-
- [49] S. Wu, R. Z. Ríos-Mercado, E. A. Boyd y L. R. Scott. Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modeling*, 31(2-3): 197-220, 2000.
- [50] H. I. Zimmer. Calculating optimum pipeline operations. Reporte Técnico, Compañía de Gas Natural El Paso, 1975. Presentado en la Conferencia de Transmisión AGA, 1975.

LISTA DE TABLAS

4.1	Conjunto de flujos factibles obtenido por el algoritmo de Asignación “clásica” en 3 iteraciones.....	42
4.2	Tabla de costos incurridos por configuraciones óptimas de operación..	53
5.1	Descripción de los tipos de componentes que describen a una solución del procedimiento NONDP_TS.....	65
6.1	Resultados computacionales sobre redes lineales.....	79
6.2	Resultados computacionales sobre redes tipo árbol.....	80
6.3	Resultados computacionales sobre redes cíclicas.....	81
6.4	Desempeño del Algoritmo NDP contra el Método GRG.....	83
6.5	Resultados computacionales del análisis comparativo de los parámetros del procedimiento NONDP_TS.....	86
6.6	Porcentaje de mejoramiento del procedimiento NONDP_TS desde la solución NDP simple.....	89
6.7	Porcentaje relativo de error de las soluciones NONDP_TS vs cotas inferiores.....	90

APÉNDICE

A1	Observaciones de las gráficas de desempeño de TS.....	113
B1	Resultados computacionales de las topologías probadas con la heurística NONDP_TS con una $\Delta p = 5$	115
B2	Resultados computacionales de las topologías probadas con la heurística NONDP_TS con una $\Delta p = 1$	115
B3	Comparación de los resultados obtenidos por dos tamaños de mallas: $\Delta p = \{5, 1\}$	116
C1	Gráficas de convergencia del procedimiento NONDP_TS.....	117

LISTA DE FIGURAS

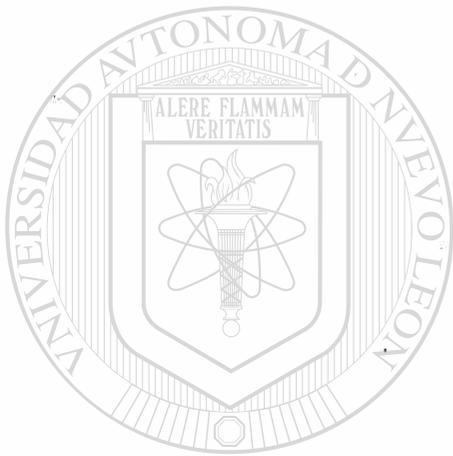
3.1	Dominio $D^{unit}_{(i,j)}$ de operación factible.....	21
3.2	Gráfica del dominio $D^{unit}_{(i,j)}$ cuando x_{ij} está fija.....	22
3.3	Superficie del dominio $D^{unit}_{(i,j)}$ cuando p_i está fijo.....	22
3.4	Frontera de operación factible D^* definido por (3.14), (3.15), (3.16b) y (3.17b).....	25
3.5	Gráfica de la función del consumo de combustible cuando p_i está fijo.	27
4.1	Algoritmo de la Técnica de Reducción aplicada a redes de gas natural.....	34
4.2	Algoritmo NDP para el PMCC.....	36
4.3	Reclasificación: Topología de grado 1.....	37
4.4	Reclasificación: Topología de grado mayor a 1.....	37
4.5	Algoritmo de Asignación “clásica”.....	39
4.6	Grafo clasificado como topología de grado 1.....	40
4.7	1ª iteración del algoritmo de Asignación “clásica”.....	40
4.8	2ª iteración del algoritmo de Asignación “clásica”.....	41
4.9	3ª iteración del algoritmo de Asignación “clásica”.....	41
4.10	Pseudo-código del Algoritmo de Grafo Reducido.....	43
4.11	Combinación de dos estaciones en serie.....	48
4.12	Combinación de dos compresores con un elemento colgante.....	49
4.13	Combinación de dos estaciones en paralelo.....	50
4.14	Descripción de la Técnica de DP No Secuencial.....	50
4.15	Grafo original G y su grafo reducido G^* correspondiente.....	51
4.16	Ejecución del Paso 2 del algoritmo de DP no secuencial.....	51
4.17	Reducción óptima de un sistema de red por la técnica de DP no secuencial.....	52
5.1	Topología cíclica en la cual se definen los tres componentes básicos de una solución factible en el procedimiento NONDP_TS.....	65
5.2	Diagrama de flujo del Algoritmo NONDP_TS.....	66
6.1	Descripción de nomenclatura de instancias.....	75
6.2	Topología no cíclica: Estructura lineal.....	76
6.3	Topología no cíclica: Estructura tipo árbol.....	76

6.4	Ejemplos de instancias con estructuras cíclicas.....	77
6.5	Convergencia de NONDP_TS en instancia net-c-6c2-C7.....	91
6.6	Convergencia de NONDP_TS en instancia net-c-10c3-C1.....	92

APÉNDICE

A1	Convergencia del procedimiento NONDP_TS en instancia net-c-6c2-C1 con 100 iteraciones.....	112
A1	Convergencia del procedimiento NONDP_TS en instancia net-c-6c2-C1 con 500 iteraciones.....	112
C-1a	Convergencia de NONDP_TS en instancia net-c-6c2-C4 para <i>Nei_Size</i> = 20 y <i>Ttenure</i> = 5.....	119
C-1b	Convergencia de NONDP_TS en instancia net-c-6c2-C4 para <i>Nei_Size</i> = 20 y <i>Ttenure</i> = 8.....	119
C-1c	Convergencia de NONDP_TS en instancia net-c-6c2-C4 para <i>Nei_Size</i> = 20 y <i>Ttenure</i> = 10.....	119
C-2a	Convergencia de NONDP_TS en instancia net-c-6c2-C4 para <i>Nei_Size</i> = 30 y <i>Ttenure</i> = 5.....	120
C-2b	Convergencia de NONDP_TS en instancia net-c-6c2-C4 para <i>Nei_Size</i> = 30 y <i>Ttenure</i> = 8.....	120
C-2c	Convergencia de NONDP_TS en instancia net-c-6c2-C4 para <i>Nei_Size</i> = 30 y <i>Ttenure</i> = 10.....	120
C-3a	Convergencia de NONDP_TS en instancia net-c-6c2-C5 para <i>Nei_Size</i> = 20 y <i>Ttenure</i> = 5.....	121
C-3b	Convergencia de NONDP_TS en instancia net-c-6c2-C5 para <i>Nei_Size</i> = 20 y <i>Ttenure</i> = 8.....	121
C-3c	Convergencia de NONDP_TS en instancia net-c-6c2-C5 para <i>Nei_Size</i> = 20 y <i>Ttenure</i> = 10.....	121
C-4a	Convergencia de NONDP_TS en instancia net-c-6c2-C5 para <i>Nei_Size</i> = 30 y <i>Ttenure</i> = 5.....	122
C-4b	Convergencia de NONDP_TS en instancia net-c-6c2-C5 para <i>Nei_Size</i> = 30 y <i>Ttenure</i> = 8.....	122
C-4c	Convergencia de NONDP_TS en instancia net-c-6c2-C5 para <i>Nei_Size</i> = 30 y <i>Ttenure</i> = 10.....	122

C-5a	Convergencia de NONDP_TS en instancia net-c-6c2-C8 para <i>Nei_Size = 20 y Ttenure = 5</i>	123
C-5b	Convergencia de NONDP_TS en instancia net-c-6c2-C8 para <i>Nei_Size = 20 y Ttenure = 8</i>	123
C-5c	Convergencia de NONDP_TS en instancia net-c-6c2-C8 para <i>Nei_Size = 20 y Ttenure = 10</i>	123
C-6a	Convergencia de NONDP_TS en instancia net-c-6c2-C8 para <i>Nei_Size = 30 y Ttenure = 5</i>	124
C-6b	Convergencia de NONDP_TS en instancia net-c-6c2-C8 para <i>Nei_Size = 30 y Ttenure = 8</i>	124
C-6c	Convergencia de NONDP_TS en instancia net-c-6c2-C8 para <i>Nei_Size = 30 y Ttenure = 10</i>	124
C-7a	Convergencia de NONDP_TS en instancia net-c-10c3-C2 para <i>Nei_Size = 20 y Ttenure = 5</i>	125
C-7b	Convergencia de NONDP_TS en instancia net-c-10c3-C2 para <i>Nei_Size = 20 y Ttenure = 8</i>	125
C-7c	Convergencia de NONDP_TS en instancia net-c-10c3-C2 para <i>Nei_Size = 20 y Ttenure = 10</i>	125
C-8a	Convergencia de NONDP_TS en instancia net-c-10c3-C2 para <i>Nei_Size = 30 y Ttenure = 5</i>	126
C-8b	Convergencia de NONDP_TS en instancia net-c-10c3-C2 para <i>Nei_Size = 30 y Ttenure = 8</i>	126
C-8c	Convergencia de NONDP_TS en instancia net-c-10c3-C2 para <i>Nei_Size = 30 y Ttenure = 10</i>	126
<hr/>		
D1	Comparación de complejidad: polinomial ($f(t)$) y exponencial ($g(t)$).....	131
D2	Secuencia de los algoritmos que conforman la metodología de solución..	138



APÉNDICE

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

®

DIRECCIÓN GENERAL DE BIBLIOTECAS

APÉNDICE A: Evaluación del Parámetro *Iter_max* de NONDP_TS

En las Figuras A1 y A2 se muestran las gráficas de desempeño del procedimiento NONDP_TS sobre la topología net-c-6c2-C1, con 100 y 500 iteraciones máximas, respectivamente.

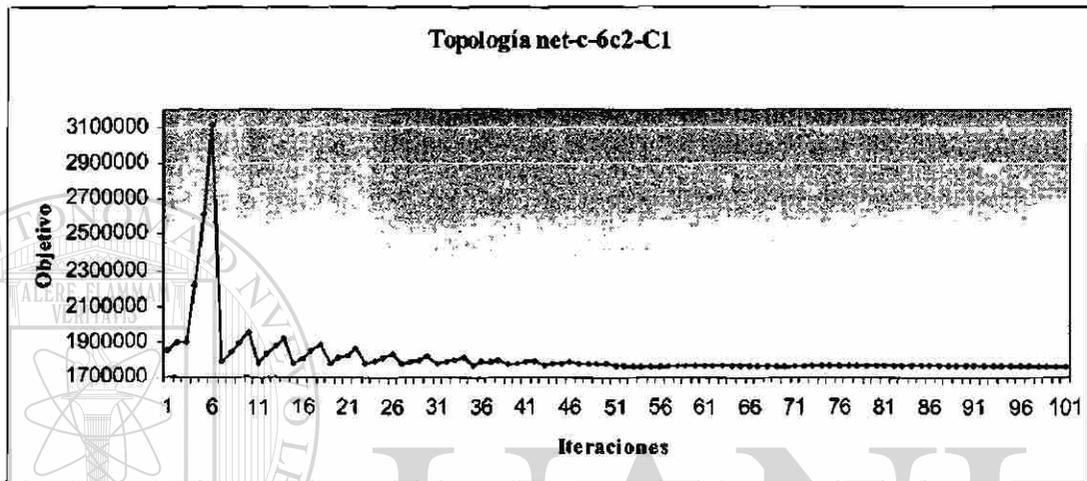


Figura A1. Convergencia del procedimiento NONDP_TS en instancia net-c-6c2-C1 con 100 iteraciones.

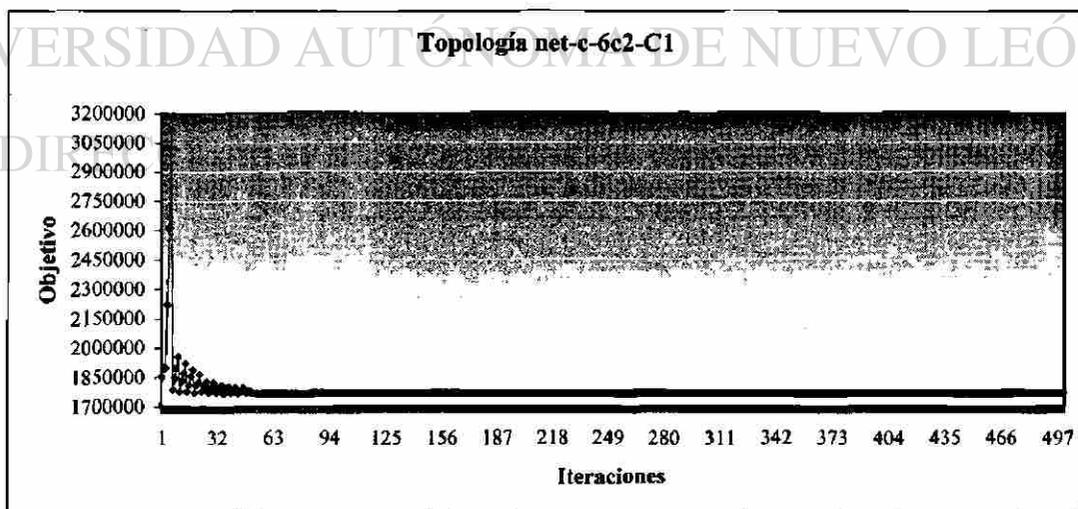


Figura A2. Convergencia del procedimiento NONDP_TS en instancia net-c-6c2-C1 con 500 iteraciones.

Observaciones	100 Iter (Fig. A1)	500 Iter (Fig. A2)
Valor inicial	1852252.54	1852252.54
Mejor valor encontrado	1765178.63	1765178.63
Mejor iteración	51	51
Tiempo de ejecución	3 hrs 06 min 35 seg (11195 seg)	12 hrs 22 min 11 seg (44531 seg)

Tabla A1. Observaciones de las gráficas de desempeño de TS.

En la Tabla A1 se presentan los resultados observados en las gráficas de desempeño de las Figuras A1 y A2 con 100 y 500 iteraciones, respectivamente. En la primera columna se muestran las medidas observadas: valor inicial, mejor valor encontrado, mejor iteración y tiempo de ejecución. En la segunda y tercera columnas se muestran los valores resultantes en la observación con 100 y 500 iteraciones, respectivamente.

Como podemos ver en la Tabla A1, el único valor modificado en los experimentos de evaluación es el tiempo de cómputo invertido, con una gran diferencia en la ejecución con 500 iteraciones. Resultados similares se presentaron en las evaluaciones de toda la amplia gama de la base de datos. Por lo tanto, se concluye que cantidad no significa calidad, tomando de esta manera para las evaluaciones posteriores del procedimiento NONDP_TS un número máximo de 100 iteraciones.

DIRECCIÓN GENERAL DE BIBLIOTECAS

APÉNDICE B: Evaluación de $\Delta p = \{1, 5\}$ en el Procedimiento NONDP_TS

Este experimento consistió en determinar en el procedimiento heurístico NONDP_TS de búsqueda tabú el uso del parámetro de discretización del algoritmo NDP con una de $\Delta p = \{1, 5\}$ unidades PSIG; esto con la finalidad de medir el tiempo y calidad de las soluciones óptimas encontradas. Asimismo, en esta evaluación se estableció para el procedimiento heurístico NONDP_TS los siguientes parámetros estáticos:

- | | |
|------------------------|----------------------------------|
| • $Nei_Size = 50$ | Tamaño del Vecindario $V(x)$ |
| • $Ttenure = 10$ | Tamaño de <i>tabu_list</i> |
| • $Iter_max = 100$ | Número máximo de iteraciones |
| • $\Delta p^{V*} = 10$ | Dispersión de la Vecindad $V(x)$ |

En la Tabla B1 se muestran los resultados obtenidos con una $\Delta p=5$ unidades de discretización, presentando en la primera columna las instancias probadas, seguidas del tiempo de cómputo invertido, la iteración de la mejor solución encontrada, el valor objetivo de la solución NDP simple y el mejor valor encontrado, así como el porcentaje de mejora con respecto a la solución NDP simple. De manera similar se presentan en la Tabla B2 los resultados obtenidos con una $\Delta p=1$ unidades de discretización.

Finalmente, en la Tabla B3 se presentan los resultados obtenidos de la comparación realizada con las dos discretización en cuestión, presentado en la primera columna las instancias probadas, seguida por dos columnas donde se muestran el tiempo de cómputo y el mejor valor objetivo encontrado para cada discretización, y así mostrar en la última columna el porcentaje de mejoramiento de las soluciones obtenidas con una discretización de $\Delta p=1$ en comparación con una mayor discretización de $\Delta p=5$. Observando cómo el esfuerzo computacional se incrementa considerablemente con una $\Delta p=1$, ya que se habla de horas de diferencia con respecto a $\Delta p=5$. Por lo tanto, para efecto de las evaluaciones computacionales del procedimiento heurístico NONDP_TS presentado en la Sección 5.3.3, se propone utilizar una discretización de $\Delta p=5$ unidades PSIG, con el objetivo de minimizar los recursos computacionales de las mismas.

Topología	$\Delta p = 5$ PSIG				
	Tiempo (min:seg)	Mejor Iter	Valor objetivo inicial	Mejor valor encontrado	% Óptimo
net-c-6c2-C1	15:32	12	1,884,060.48	1,786,399.22	5.47
net-c-6c2-C3	35:38	11	1,041,265.76	972,892.87	7.03
net-c-6c2-C4	28:41	35	744,261.08	636,148.21	16.99
net-c-6c2-C5	28:55	87	2,803,372.66	2,717,573.08	3.16
net-c-6c2-C6	26:43	12	3,561,310.69	3,561,310.69	0.00
net-c-6c2-C7	11:44	1	850,674.93	850,674.93	0.00
net-c-6c2-C8	14:46	47	1,468,605.58	1,405,226.98	4.51
net-c-6c2-C9	11:20	1	1,294,408.69	1,294,408.69	0.00
net-c-10c3-C2	22:55	57	5,483,509.96	4,831,979.07	13.48
net-c-10c3-C3	47:10	35	4,081,328.16	3,767,167.75	8.33
net-c-10c3-C4	68:39	62	4,250,905.33	4,094,274.35	3.82
net-c-10c3-C5	63:54	38	7,146,123.19	6,153,153.87	16.13
net-c-10c3-C6	59:24	92	10,996,571.93	10,862,847.37	1.23
net-c-10c3-C8	48:17	96	5,265,444.54	5,195,376.60	1.34
net-c-10c3-C9	22:55	1	3,258,260.47	3,258,260.47	0.00

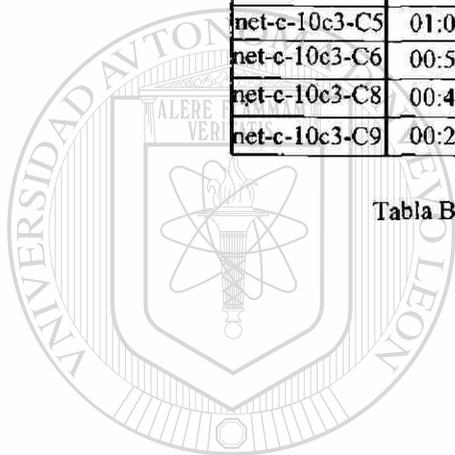
Tabla B1. Resultados computacionales de las topologías probadas con la heurística NONDP_TS con una $\Delta p = 5$.

Topología	$\Delta p = 1$ PSIG				
	Tiempo (hrs:min:seg)	Mejor Iter	Valor objetivo Inicial	Mejor valor encontrado	% Óptimo
net-c-6c2-C1	3:12.48	27	1,852,252.53	1,786,399.22	3.68
net-c-6c2-C3	4:04.90	34	1,020,842.29	907,728.31	12.46
net-c-6c2-C4	6:32:09	30	653,675.58	527,513.93	19.3
net-c-6c2-C5	3:53.25	27	2,716,611.07	2,646,948.81	2.63
net-c-6c2-C6	4:18.05	28	3,442,838.76	3,407,748.11	1.03
net-c-6c2-C7	4:42.33	62	788,218.02	726,355.04	8.52
net-c-6c2-C8	4:31.55	51	1,405,443.48	1,356,798.94	3.59
net-c-6c2-C9	4:28.44	47	1,243,179.03	1,209,864.45	2.75
net-c-10c3-C2	3:58.13	25	4,869,023.74	4,714,008.54	3.28
net-c-10c3-C3	3:55.56	2	3,978,465.67	3,785,841.38	5.09
net-c-10c3-C4	4:44.03	26	1,082,614.93	888,915.07	21.79
net-c-10c3-C5	6:52.15	30	6,624,644.75	5,890,916.41	12.46
net-c-10c3-C6	6:42.13	41	10,766,001.9	10,637,203.60	1.21
net-c-10c3-C8	3:46.55	1	3,316,383.49	3,316,383.49	0.00
net-c-10c3-C9	5:17:22	56	3,039,766.16	2,921,299.37	4.06

Tabla B2. Resultados computacionales de las topologías probadas con la heurística NONDP_TS con una $\Delta p = 1$.

Topología	$\Delta p = 5$ PSIG		$\Delta p = 1$ PSIG		% Opt. $\Delta p = 1$
	Tiempo hh:mm:ss	Mejor valor encontrado	Tiempo hh:mm:ss	Mejor valor encontrado	
net-c-6c2-C1	00:15:48	1,786,399.22	03:12.48	1,786,399.22	0.0
net-c-6c2-C2	00:15:32	1,786,399.22	03:21.52	1,786,399.22	0.0
net-c-6c2-C3	00:35:38	972,892.87	03:04.90	907,728.31	7.17
net-c-6c2-C4	00:28:41	636,148.21	06:32:09	527,513.93	17.07
net-c-6c2-C5	00:28:55	2,717,573.08	03:53.25	2,646,948.81	2.66
net-c-6c2-C6	00:26:43	3,561,310.69	04:18.05	3,407,748.11	4.5
net-c-6c2-C7	00:11:44	850,674.93	04:42.33	726,355.04	17.11
net-c-6c2-C8	00:14:46	1,405,226.98	04:31.55	1,356,798.94	3.56
net-c-6c2-C9	00:11:20	1,294,408.69	04:28.44	1,209,864.45	6.98
net-c-10c3-C2	00:22:55	4,831,979.07	03:58.13	4,714,008.54	2.5
net-c-10c3-C3	00:47:10	3,785,841.38	03:55.56	3,767,167.75	0.49
net-c-10c3-C5	01:03:54	6,153,153.87	06:52.15	5,890,916.41	4.45
net-c-10c3-C6	00:59:24	10,862,847.37	06:42.13	10,637,203.60	2.12
net-c-10c3-C8	00:48:17	5,195,376.60	03:46.55	3,316,383.49	36.16
net-c-10c3-C9	00:22:55	3,258,260.47	05:17:22	2,921,299.37	11.53

Tabla B3. Comparación de los resultados obtenidos por dos tamaños de mallas: $\Delta p = \{5, 1\}$.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



APÉNDICE C: Evaluación de la Convergencia del Procedimiento NONDP_TS sobre dos Parámetros de TS: *Ttenure* y *Nei_Size*

Este experimento se realizó con el objetivo de mostrar las gráficas de desempeño del procedimiento NONDP_TS (ver Sección 5.3.3), presentando así una evaluación computacional sobre dos parámetros esenciales de la Búsqueda Tabú, a saber: Tamaño de la *lista tabú* para un *Ttenure* = {5, 8, 10} y tamaño del vecindario $V(x)$ para un *Nei_Size* = {20, 30}. Estas gráficas muestran el mejor valor objetivo encontrado en cada iteración.

Las gráficas de desempeño de las evaluaciones realizadas sobre 4 diferentes topologías cíclicas tomadas de la base de datos pueden verse a continuación en el orden que nos muestra la Tabla C1.

Topología	Tamaño de la lista tabú <i>Ttenure</i>	Tamaño del Vecindario	
		<i>Nei_Size</i> = 20	<i>Nei_Size</i> = 30
net-c-6c2-C4	5	Figura C-1a	Figura C-2a
	8	Figura C-1b	Figura C-2b
	10	Figura C-1c	Figura C-2c
net-c-6c2-C5	5	Figura C-3a	Figura C-4a
	8	Figura C-3b	Figura C-4b
	10	Figura C-3c	Figura C-6c
net-c-6c2-C8	5	Figura C-5a	Figura C-6a
	8	Figura C-5b	Figura C-6b
	10	Figura C-5c	Figura C-6c
net-c-10c3-C2	5	Figura C-7a	Figura C-8a
	8	Figura C-7b	Figura C-8b
	10	Figura C-7c	Figura C-8c

Tabla C1. Gráficas de convergencia del procedimiento NONDP_TS.

En las gráficas presentadas por la Tabla C1 se pudieron observar los siguientes puntos:

- En todas las gráficas de desempeño puede verse la convergencia de una Búsqueda Tabú tradicional hacia la mejor solución encontrada.
- En todas las topologías evaluadas con un tamaño *Ttenure* = 10 de la lista

tabú, las gráficas respectivas demostraron tener una mayor diversificación de las mejores soluciones obtenidas en cada iteración; aunque éstas convergieron en un número mayor de iteraciones que con un tamaño de lista menor.

- En todas las topologías evaluadas sobre los dos tamaños del vecindario $Nei_Size = \{20, 30\}$, para un mismo tamaño de *lista tabú*, las gráficas de desempeño se comportaron de manera similar en cuanto a la mejor iteración presentada por el procedimiento heurístico NONDP_TS; por lo que, la implementación de cualquiera de estos dos diferentes valores no se considera un factor predominante en la exploración del espacio de búsqueda, pero sí en un factor determinante en el incremento de los recursos computacionales requeridos por un tamaño del vecindario mayor. Cabe aclarar, que aún cuando solo se presentan tres tamaños diferentes del vecindario en este trabajo (20, 30 y 40), para la conclusión de este experimento se agregó la evaluación de dos diferentes tamaños más {10 y 50}, presentando resultados similares a los obtenidos por esta evaluación.
- Todas aquellas topologías evaluadas sobre los parámetros combinados de $Ttenure = 8$ y $Nei_Size = 20$, las gráficas respectivas mostraron que el procedimiento heurístico NONDP_TS convergió mucho antes que las otras combinaciones probadas, resultando en la misma solución en un menor tiempo de ejecución; encontrando además, una mayor exploración del espacio de búsqueda.

Finalmente, en base a las observaciones realizadas en este análisis, se concluyó que los valores de $Ttenure = 8$ y $Nei_Size = 20$ representaron un mayor beneficio tanto en la exploración del espacio de búsqueda como en la minimización de los recursos computacionales requeridos; por lo que, estos valores son fijados en todas las experimentaciones para evaluar la calidad de las soluciones obtenidas por el procedimiento NONDP_TS.

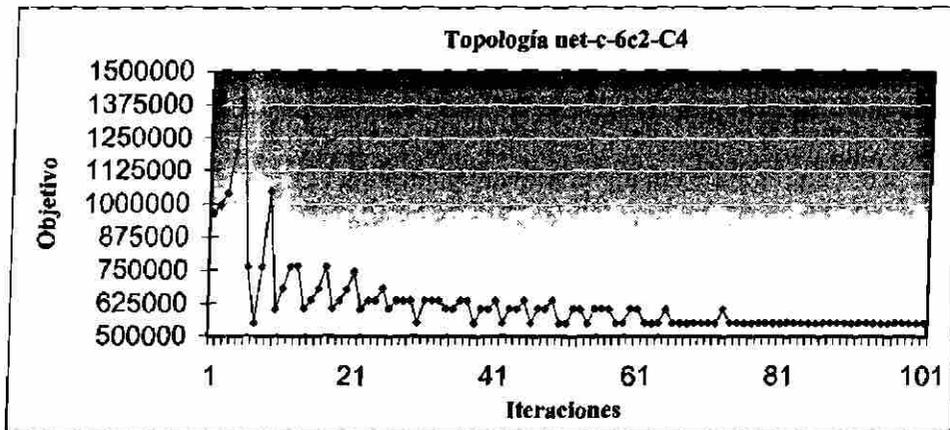


Figura C-1a: Convergencia de NONDP_TS en instancia net-c-6c2-C4 para $Nei_Size = 20$ y $Tienure = 5$.

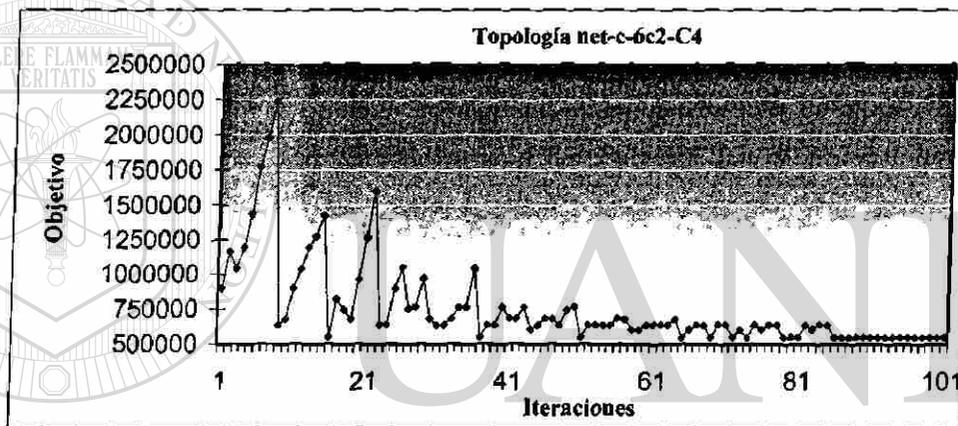


Figura C-1b: Convergencia de NONDP_TS en instancia net-c-6c2-C4 para $Nei_Size = 20$ y $Tienure = 8$.

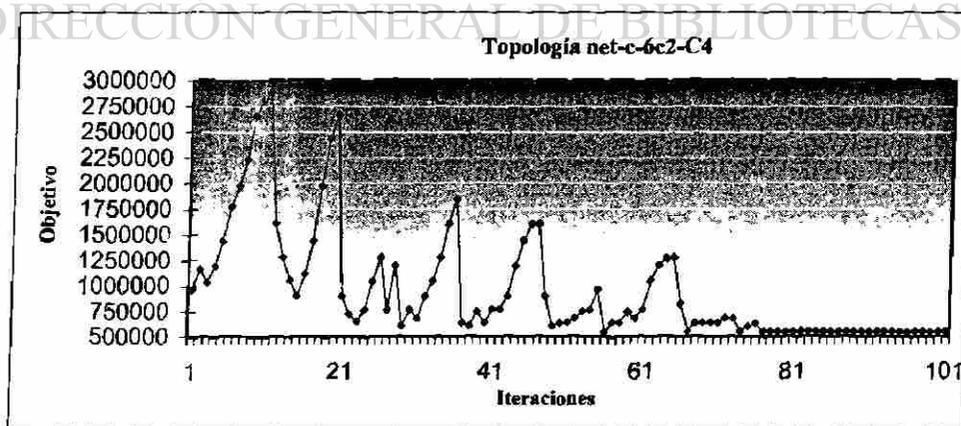


Figura C-1c: Convergencia de NONDP_TS en instancia net-c-6c2-C4 para $Nei_Size = 20$ y $Tienure = 10$.

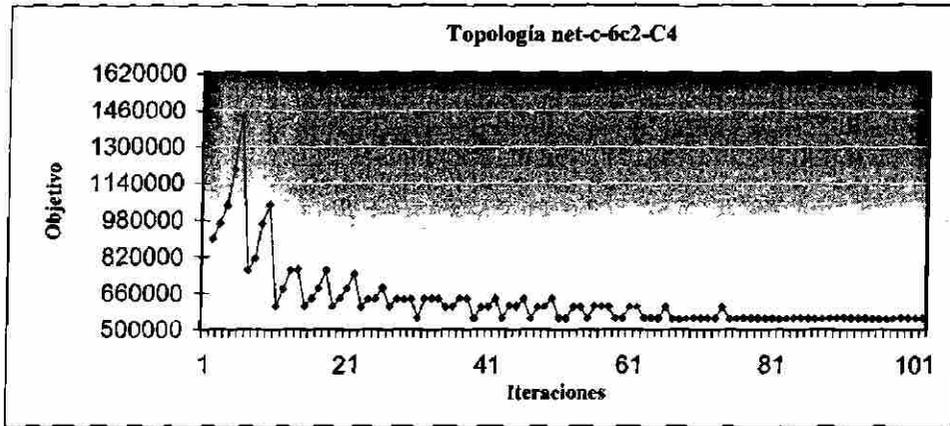


Figura C-2a: Convergencia de NONDP TS en instancia net-c-6c2-C4 para $Nei_Size = 30$ y $Ttenure = 5$.

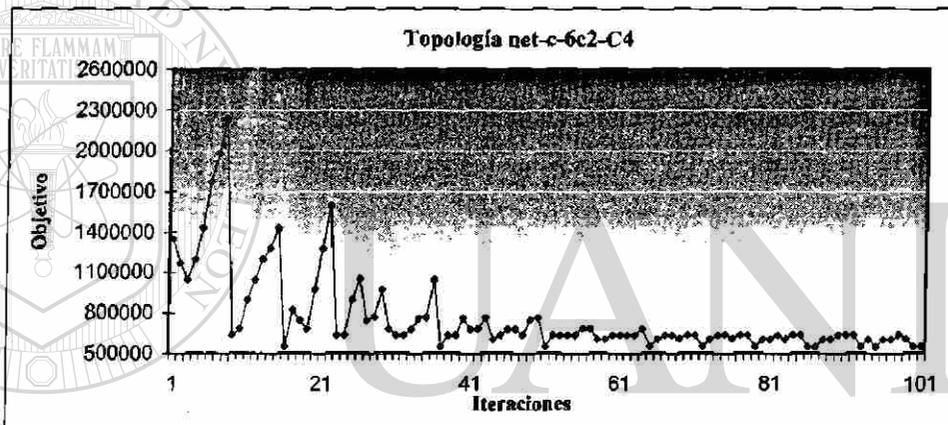


Figura C-2b: Convergencia de NONDP TS en instancia net-c-6c2-C4 para $Nei_Size = 30$ y $Ttenure = 8$.

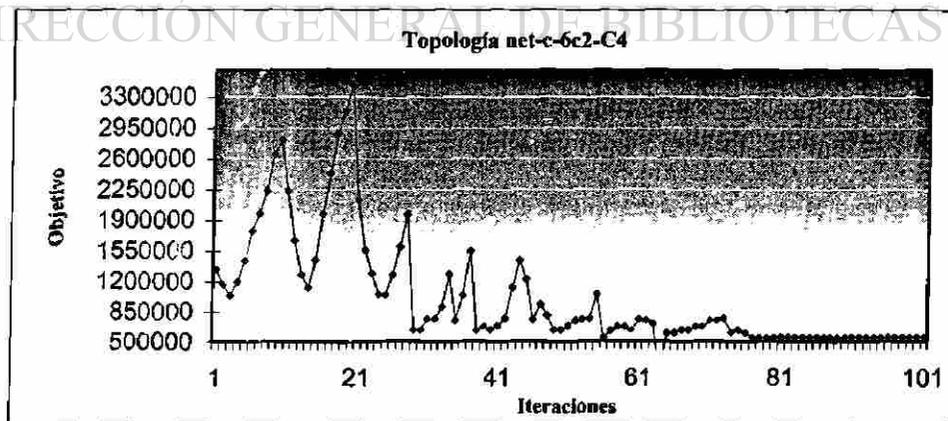


Figura C-2c: Convergencia de NONDP TS en instancia net-c-6c2-C4 para $Nei_Size = 30$ y $Ttenure = 10$.

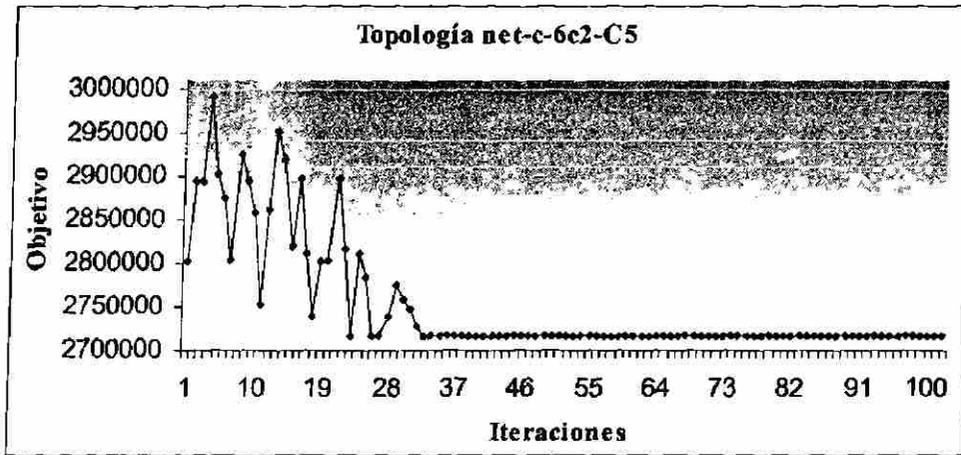


Figura C-3a: Convergencia de NONDP_TS en instancia net-c-6c2-C5 para $Nei_Size = 20$ y $Ttenure = 5$.

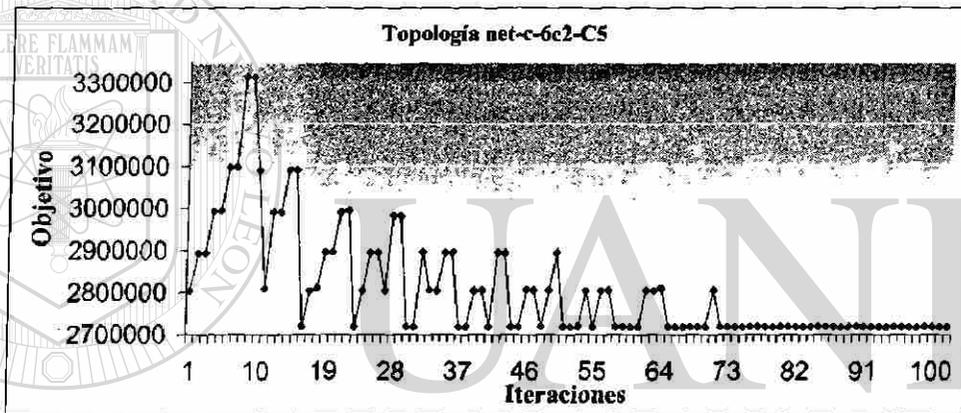


Figura C-3b: Convergencia de NONDP_TS en instancia net-c-6c2-C5 para $Nei_Size = 20$ y $Ttenure = 8$.

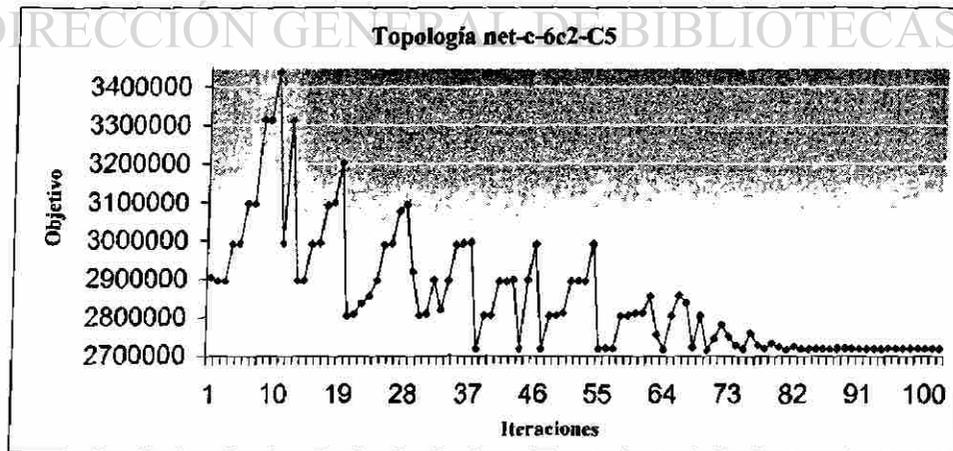


Figura C-3c: Convergencia de NONDP_TS en instancia net-c-6c2-C5 para $Nei_Size = 20$ y $Ttenure = 10$.

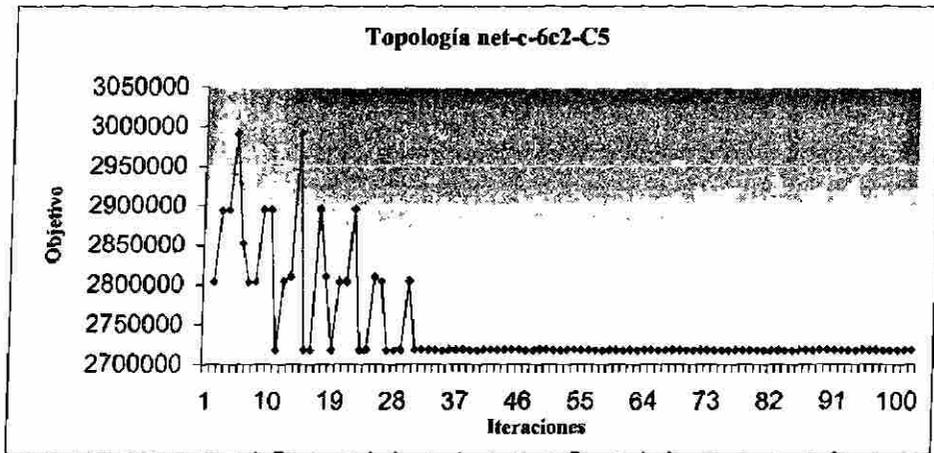


Figura C-4a: Convergencia de NONDP_TS en instancia net-c-6c2-C5 para $Nei_Size = 30$ y $Ttenure = 5$.

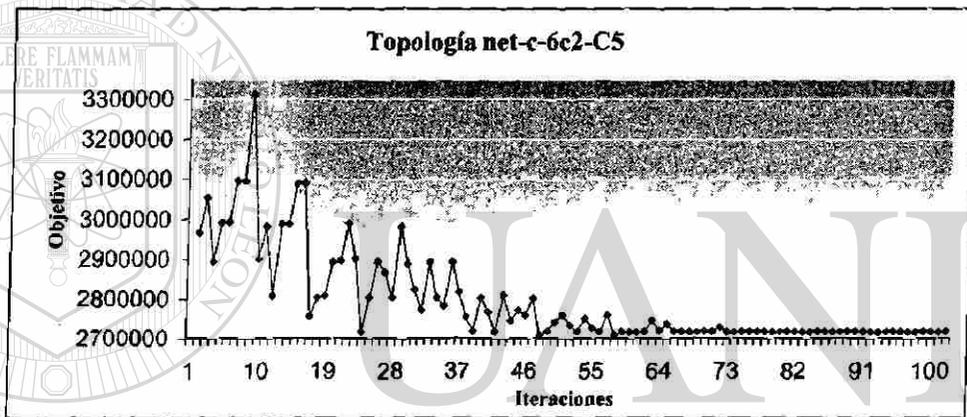


Figura C-4b: Convergencia de NONDP_TS en instancia net-c-6c2-C5 para $Nei_Size = 30$ y $Ttenure = 8$.

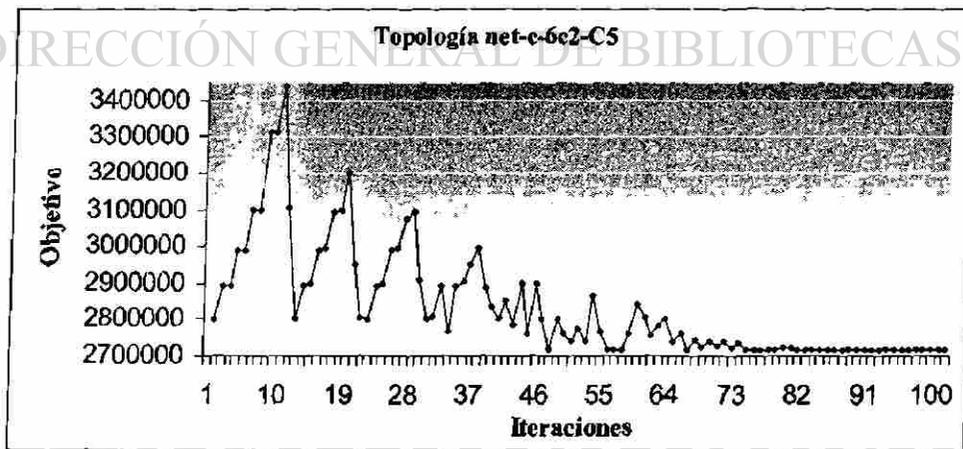


Figura C-4c: Convergencia de NONDP_TS en instancia net-c-6c2-C5 para $Nei_Size = 30$ y $Ttenure = 10$.

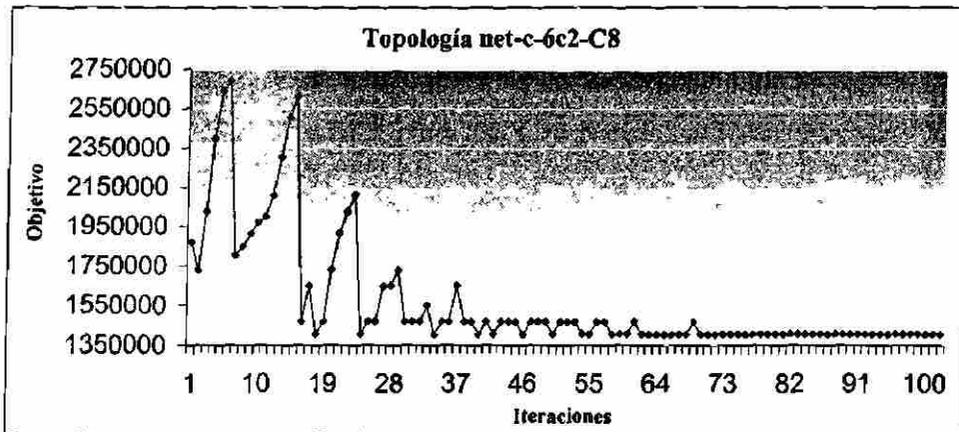


Figura C-5a: Convergencia de NONDP_TS en instancia net-c-6c2-C8 para $Nei_Size = 20$ y $Ttenure = 5$.

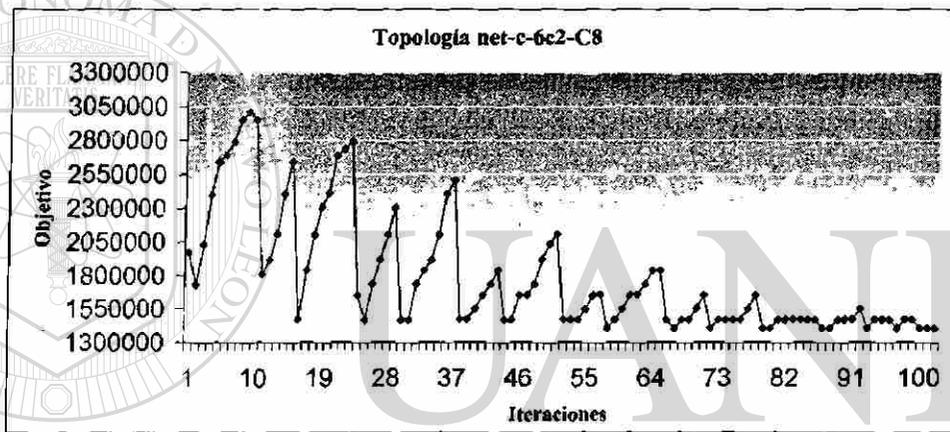


Figura C-5b: Convergencia de NONDP_TS en instancia net-c-6c2-C8 para $Nei_Size = 20$ y $Ttenure = 8$.

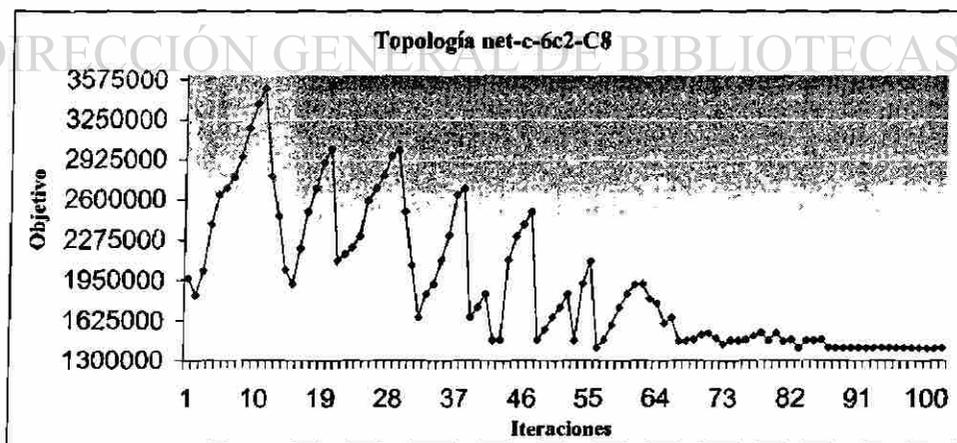


Figura C-5c: Convergencia de NONDP_TS en instancia net-c-6c2-C8 para $Nei_Size = 20$ y $Ttenure = 10$.

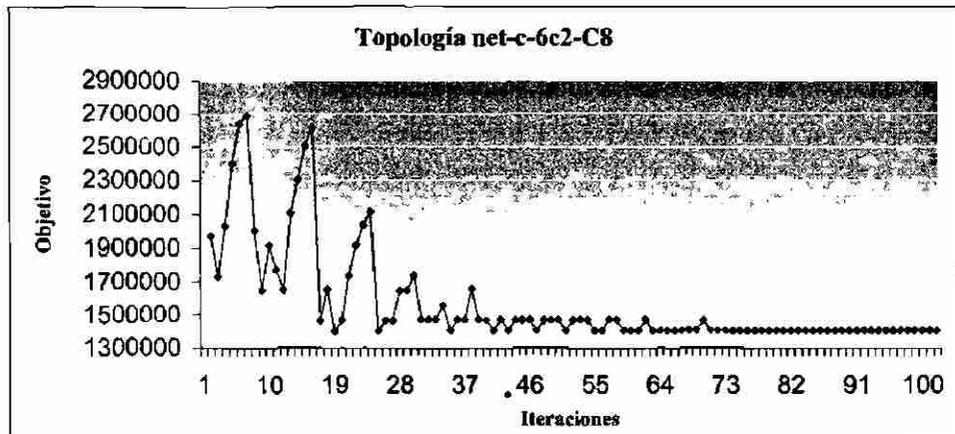


Figura C-6a: Convergencia de NONDP_TS en instancia net-c-6c2-C8 para $Nei_Size = 30$ y $Tenure = 5$.

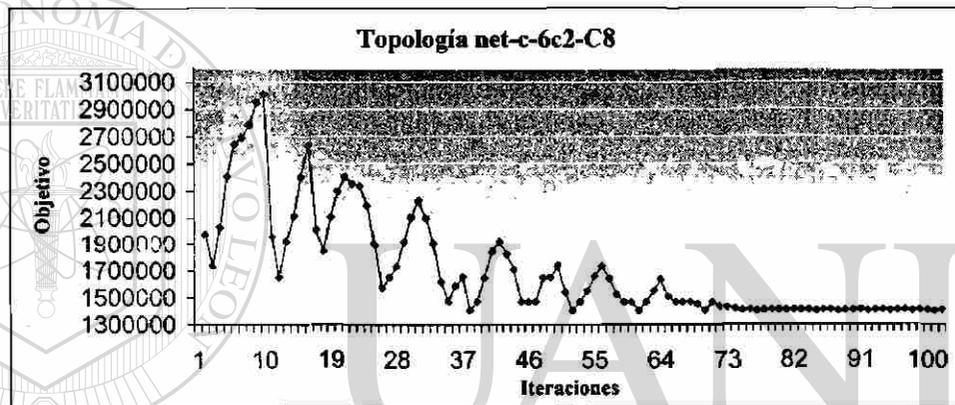


Figura C-6b: Convergencia de NONDP_TS en instancia net-c-6c2-C8 para $Nei_Size = 30$ y $Tenure = 8$.

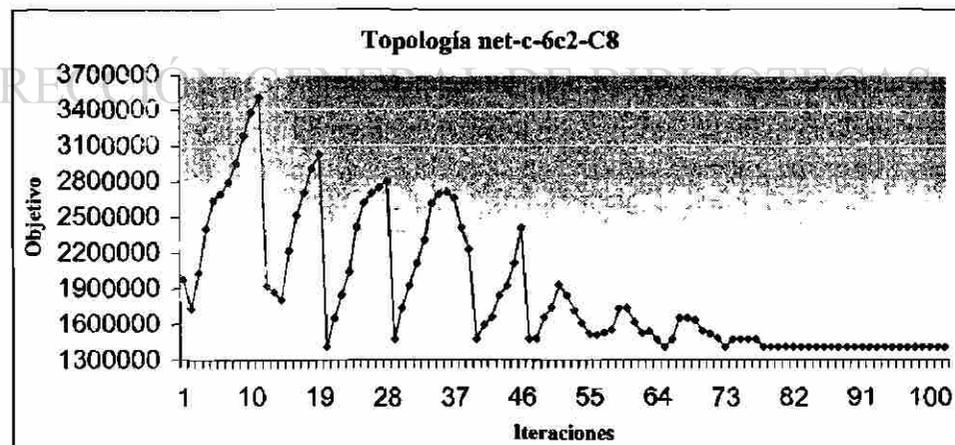


Figura C-6c: Convergencia de NONDP_TS en instancia net-c-6c2-C8 para $Nei_Size = 30$ y $Tenure = 10$.

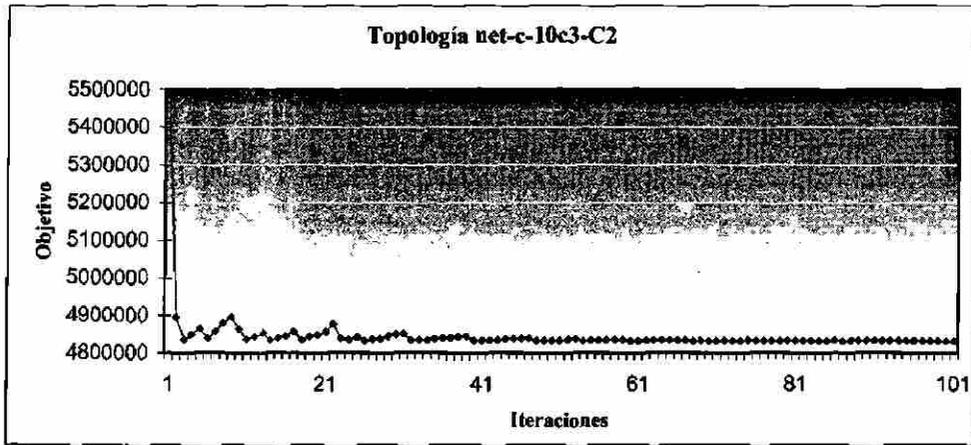


Figura C-7a: Convergencia de NONDP_TS en instancia net-c-10c3-C2 para $Nei_Size = 20$ y $Ttenure = 5$.

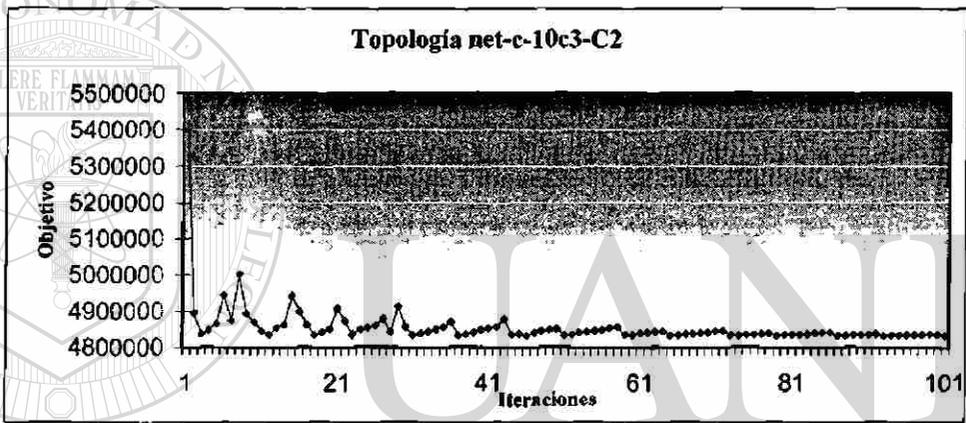


Figura C-7b: Convergencia de NONDP_TS en instancia net-c-10c3-C2 para $Nei_Size = 20$ y $Ttenure = 8$.

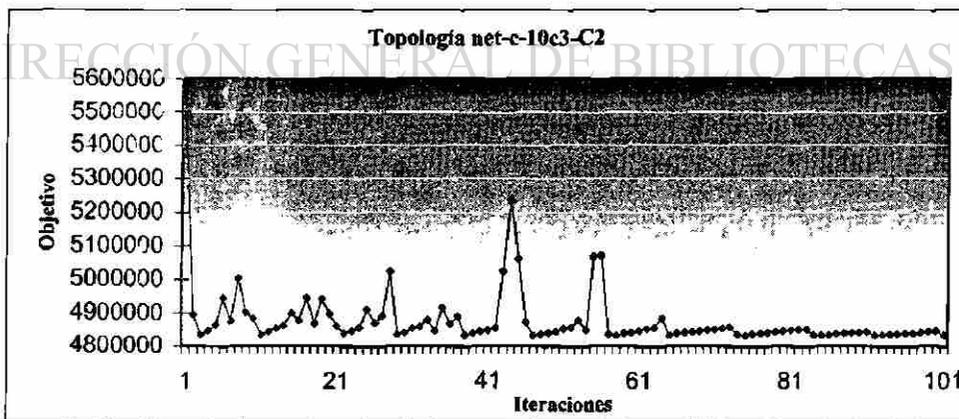


Figura C-7c: Convergencia de NONDP_TS en instancia net-c-10c3-C2 para $Nei_Size = 20$ y $Ttenure = 10$.

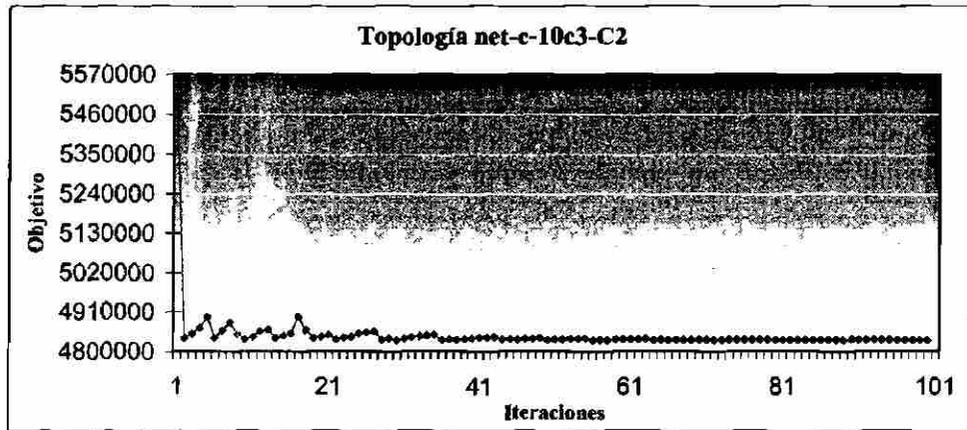


Figura C-8a: Convergencia de NONDP_TS en instancia net-c-10c3-C2 para $Nei_Size = 30$ y $Tienure = 5$.

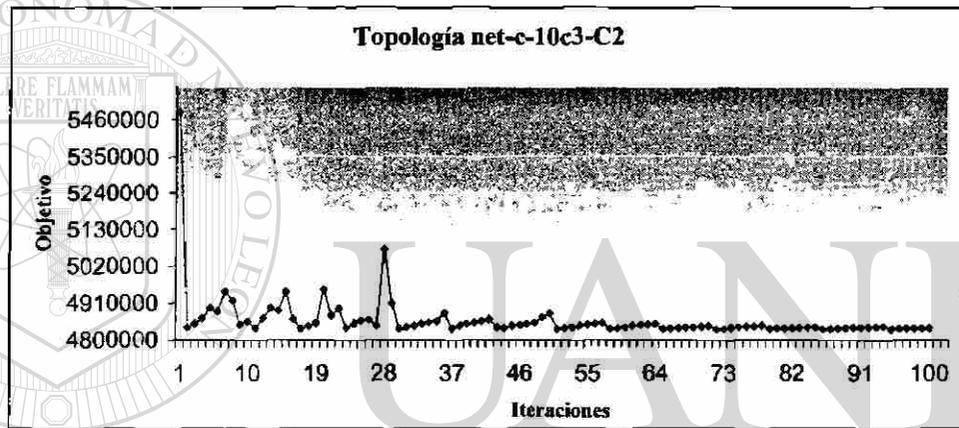


Figura C-8b: Convergencia de NONDP_TS en instancia net-c-10c3-C2 para $Nei_Size = 30$ y $Tienure = 8$.

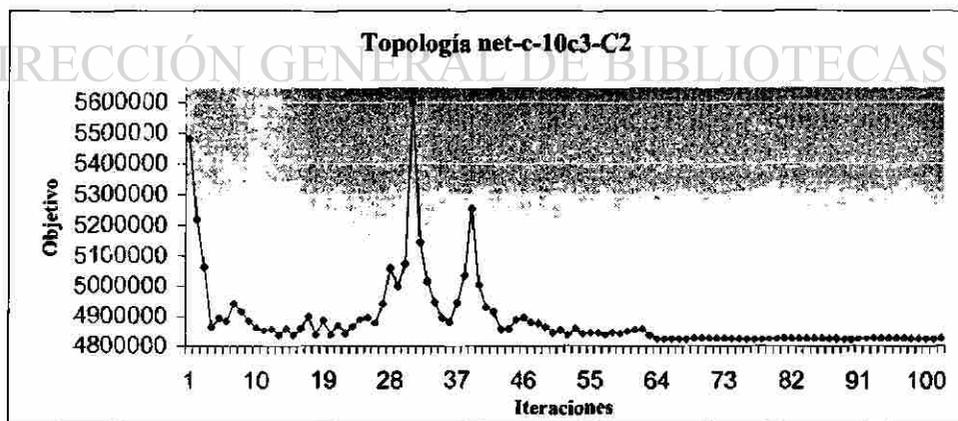


Figura C-8c: Convergencia de NONDP_TS en instancia net-c-10c3-C2 para $Nei_Size = 30$ y $Tienure = 10$.

APÉNDICE D: Teoría de Complejidad

En este apéndice se presentan los conceptos relativos al tema de la Teoría de Complejidad, a saber: complejidad y orden de los algoritmos. También se describen las diferentes clases de problemas de decisión (P y NP) y su correspondencia con los problemas de optimización.

D.1 Complejidad de los Algoritmos

La Teoría de Complejidad de los algoritmos concierne a la identificación de los problemas que, desde una perspectiva computacional, son fáciles de resolver y aquéllos que son difíciles de resolver (ver [13] y [37]).

En la Ciencia de la Computación, al hablar de la teoría de complejidad, no se está refiriendo a la dificultad que se tendría para diseñar un programa. Esta teoría está vinculada con la identificación de los algoritmos que son eficientes y aquéllos que son ineficientes desde un punto de vista computacional. Desde una perspectiva general, cuando se analiza y compara el desempeño de un algoritmo, su eficiencia se fija en términos de los recursos de cómputo que son necesarios para ejecutarlo, a saber: tiempo de ejecución y espacio. Al analizar la complejidad de un algoritmo, el tiempo se expresa en término de los pasos de computación elementales (asignaciones, comparaciones, multiplicaciones, etc.) que le toma al algoritmo calcular el resultado a partir de los datos de entrada; por ejemplo, una operación de asignación ocupa una unidad de tiempo para ejecutarse, un ciclo ocupa el número de iteraciones en que está definido, etc. El espacio es un indicador de la cantidad de memoria que se necesita para ejecutar el algoritmo. Sin embargo, en la teoría de la complejidad de los algoritmos, la eficiencia está usualmente expresada en términos de la complejidad del tiempo. La *complejidad temporal* tiene que ver con el tiempo que tarda un algoritmo en ejecutarse. La *complejidad espacial* estudia la cantidad de espacio de almacenamiento que es necesario para una operación.

Es evidente que al medir el tiempo que tarda un algoritmo en ejecutarse, no es conveniente especificar que tarda cinco segundos en una máquina de 166 MHz y que tarda menos en una máquina más rápida. Para que sea relevante el definir la complejidad temporal de un algoritmo, ésta debe ser expresada no en términos de la máquina, sino en

cuanto al *tamaño de la entrada*. Al referirse la complejidad con respecto al tamaño de entrada, esta medida resulta ser más representativa y útil.

Cuando se define la complejidad en función del tamaño de entrada, no se está considerando a los datos que forman esa entrada, sino al conjunto en su tamaño. En la práctica se observa que un algoritmo ocupa diferente tiempo de ejecución para entradas del mismo tamaño, pero con diferentes datos. Por ejemplo, un método de ordenación de valores puede tardar menos tiempo si su entrada está ordenada, o tardar mucho mayor tiempo incluso con los mismos valores, pero presentados en desorden; es decir, el mismo tamaño de entrada tiene diferente comportamiento. Por lo tanto, se ha adoptado el criterio de tomar siempre como base de análisis de la complejidad de un algoritmo, el caso en el cual consuma el mayor tiempo de ejecución, es decir, *el peor caso* (ver [32] y [33]), el cual es el número máximo de operaciones básicas que el algoritmo espera ejecutar para una entrada de tamaño n .

Orden de un Algoritmo

La función que define el tiempo de ejecución de un algoritmo proporciona información interesante para clasificar los diferentes algoritmos que existen para resolver problemas. Con esta función es posible comparar el desempeño de diferentes algoritmos desarrollados para un problema en particular.

Para simplificar el estudio de la complejidad, se han adoptado ciertas convenciones en su notación, una de ellas es la del *concepto de orden* [32], el cual se expresa usando la notación $O(T(n))$, donde n simboliza el tamaño de entrada, la cual es definida como sigue:

Una función $T(n)$ se dice ser $O(T(n))$, si hay una constante k , tal que $|T(n)| \leq k \cdot |T(n)|, \forall n \geq 0$. En otras palabras, $O(T(n))$ se refiere a las funciones que no crecen más rápido que $T(n)$ y la notación $O(T(n))$ indica que la complejidad del tiempo del *peor caso* del algoritmo está acotada por $T(n)$. Esta notación indica, de forma simple, el grado de complejidad de un algoritmo sin considerar por completo la función de tiempo. Por ejemplo, tomando el método de la burbuja para ordenamiento de datos, éste tiene una función de tiempo $n(n-1)/2 = (n^2-n)/2$, y se dice que es de orden n^2 , o simplemente $O(n^2)$.

Ejemplo D1: La función que define el tiempo de ejecución del siguiente algoritmo es $T(n) = 4n + 3$.

Paso 1: suma = 0;
Paso 2: for (i = 1; i ≤ n; i++)
Paso 3: suma+ = i;

- El Paso 1 realiza una asignación y ocupa una unidad de tiempo $T(1)$.
- El Paso 2 tiene involucrada una asignación que utiliza una unidad de tiempo, $n+1$ comparaciones y n incrementos, por lo que ocupa $2n+2$ unidades de tiempo.
- El Paso 3 ocupa dos unidades de tiempo, una para la suma y otra para la asignación, y es ejecutada n veces, por lo que ocupa $2n$ unidades de tiempo.

Por tanto, el tiempo total del algoritmo es $T(n) = 1 + 2n + 2 + 2n$, esto es: $T(n) = 4n + 3$.

Para determinar el orden un algoritmo, a partir de su función de tiempo, se eliminan todos los términos excepto el de mayor grado y se eliminan todos los coeficientes del término mayor. Esto debido a que al aumentar el tamaño de entrada, es más significativo el incremento en el término de mayor orden que el de los demás.

Ejemplo D2: El orden de un algoritmo cuyo tiempo de ejecución está dado por $T(n) = 3n^3 + 2n + 6$ es:
 $O(T(n)) = O(n^3)$.

Los algoritmos que tienen una complejidad del tiempo descrita por una función polinomial (por ejemplo, $O(4n)$, $O(n^3)$, etc.) son considerados eficientes porque pueden ser ejecutados en una cantidad de tiempo razonable para entradas de tamaño considerable. Sin embargo, si la complejidad del tiempo del algoritmo está descrita por una función exponencial (por ejemplo, $O(3^n)$, $O(n^{\log n})$, etc.) entonces el algoritmo es considerado ineficiente porque puede ser ejecutado en una cantidad de tiempo razonable solo para entradas de longitudes pequeñas, pero para entradas más grandes ejecutar el algoritmo llegar a no resultar práctico.

La diferencia entre algoritmos de tiempo polinomial y algoritmos de tiempo exponencial es la razón en la cual su complejidad de tiempo computacional crece dado un incremento en el tamaño de la entrada (n). Hay algunos algoritmos de tiempo polinomial que no son muy útiles en la práctica, ya que n es típicamente grande para instancias prácticas. Por el contrario, hay algoritmos de tiempo exponencial que se consideran útiles, ya que el tiempo de su ejecución en las instancias prácticas se considera relativamente bueno, debido a que los valores de n son pequeños.

Existen algunas reglas que son útiles para determinar el orden de un algoritmo:

- **Regla 1 - Ciclos FOR:** El tiempo de ejecución de un ciclo *FOR* es al menos el tiempo de ejecución de las instrucciones dentro de él multiplicado por el número de iteraciones.

Ejemplo D3: El orden del siguiente algoritmo es $O(n)$.

```
Paso 1: for (i = 1; i ≤ n; i++)
Paso 2:   suma + = i;
```

La función de tiempo para el algoritmo es $T(n) = 4n + 2$, ya que el ciclo ocupa $2n + 2$ y la suma $2n$, por lo tanto tiene un orden $O(n)$.

- **Regla 2 - Ciclos FOR anidados:** Se analiza desde dentro hacia afuera. El tiempo total de una instrucción dentro de un conjunto de ciclos anidados es igual al tiempo de ejecución de las instrucciones internas multiplicado por el producto del tamaño de los ciclos.

Ejemplo D4: El orden del siguiente algoritmo es $O(n^2)$.

```
Paso 1: for (i = 1; i ≤ n; i++)
Paso 2:   for (j = 1; j ≤ n; j++)
Paso 3:     suma + = ij;
```

- **Regla 3 - Condicional:** El tiempo de ejecución nunca es mayor que el tiempo de ejecución de la condicional más el mayor de los tiempos de ejecución de las alternativas.

Ejemplo D5: El orden del siguiente algoritmo es $O(n^2)$:

```
Paso 1: if (i == j)
Paso 2:   for (i = 1; i ≤ n; i++)
Paso 3:     suma + = i;
Paso 4: else
Paso 5:   for (i = 1; i ≤ n; i++)
Paso 6:     for (j = 1; j ≤ n; j++)
Paso 7:       suma + = ij;
```

Debido a que el tiempo de ejecución de la condición (Paso 1) es de orden $O(1)$ más el mayor tiempo de ejecución de las alternativas, es decir, cuando se cumple la condición es $O(n)$ y cuando no se cumple es $O(n^2)$, por lo tanto el orden es $O(1) + O(n^2)$. Como ya se ha dicho, solo se toma el mayor, quedando el orden del algoritmo en $O(n^2)$.

Si se gráfica el orden de un algoritmo para varios tamaños de entrada, se puede observar su comportamiento. Como se muestra en la Figura D1, un algoritmo de orden $O(n^2)$ (línea gruesa) tiene una velocidad de crecimiento menor que uno con orden $O(2^n)$, representado por la línea delgada. El eje vertical representa $T(n)$ y el eje horizontal a n .

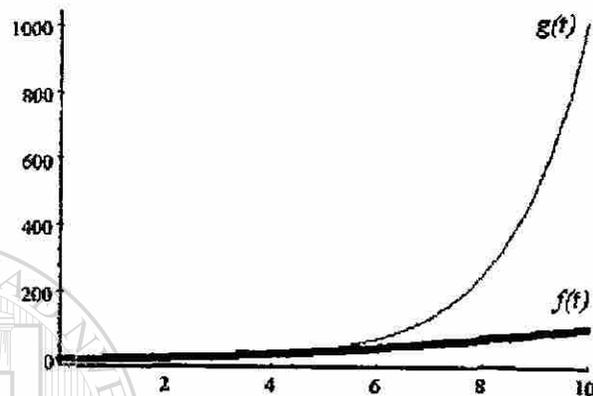


Figura D1: Comparación de complejidad: polinomial ($f(t)$) y exponencial ($g(t)$).

Se puede decir que un algoritmo tiene complejidad polinomial, o se ejecuta en *tiempo polinomial* si tiene un orden $O(n^x)$, donde x es un número fijo. Estos algoritmos se dice que son algoritmos eficientes y los problemas que se resuelven con estos algoritmos se dice que son problemas tratables.

Un algoritmo tiene complejidad exponencial si la función de tiempo $T(n)$ tiene un orden $O(x^n)$, donde x es un número fijo. Los problemas para los cuales no existe un algoritmo polinomial que los resuelva, es decir, aquellos que la única forma de resolverse es mediante algoritmos de *tiempo exponencial* se conocen como problemas intratables y a los algoritmos como algoritmos ineficientes.

¿Por qué es importante el tiempo de ejecución de un algoritmo?

Es importante porque si conocemos, o al menos tenemos una idea del tiempo de ejecución del algoritmo, podemos saber qué tanto va a tardar en entregarnos una respuesta, y por tanto, podríamos decidir si la esperamos y nos vamos a tomar un café, o si mejor regresamos después de una semana.

Aún cuando las computadoras de hoy en día son muy rápidas, comparadas con sus similares de hace algunos años, y son capaces de llevar a cabo millones de operaciones en un segundo, resulta fácil encontrar ejemplos de problemas y soluciones que tardarían años en solucionarse.

Cuando se analiza el tiempo de ejecución de un algoritmo, más que el tiempo exacto que tardará el algoritmo en milisegundos, lo que importa es la función de crecimiento del algoritmo. Esta función nos da una idea de qué tanto aumentará el tiempo de ejecución con respecto al incremento del tamaño de entrada.

Veamos como ejemplo el problema de ordenar una lista de números. Se tiene un conjunto $A = \{x_1, x_2, \dots, x_n\}$, y un algoritmo de ordenación que debe entregar como salida una permutación del conjunto A , tal que $x_i \leq x_j$, para cualquier $i < j$. Es decir, debe ordenar de manera ascendente al conjunto A .

Ordenar una lista de números es quizá la operación más común en las computadoras, cientos de programas ordenan datos para poder trabajar con ellos. Debido a esto, la cantidad de algoritmos de ordenación que se encuentra en la literatura es extensa, así como el tiempo que se ha dedicado a estudiarlos. Observemos como ejemplo el comportamiento de dos algoritmos clásicos de ordenación: *Por inserción (Insertion Sort)* y *por mezcla (Merge Sort)* [26].

La ordenación *por inserción* tiene una función de crecimiento $T(n) = n^2$, mientras que la ordenación *por mezcla* tiene una función de crecimiento $T(n) = n \log n$ (donde $\log n$ representa el logaritmo base 2 de n). Para demostrar el por qué es necesario realizar un análisis de complejidad, comparemos estos dos algoritmos. Supongamos que el mejor programador del mundo implementó la ordenación *por inserción* en lenguaje máquina logrando una constante de 2, por lo que el tiempo total de corrida será $t = 2n^2$, donde n representa la cantidad de números que deseamos ordenar. Ahora supongamos que un programador promedio implementó la ordenación *por mezcla* utilizando un lenguaje de alto nivel obteniendo una constante de 50, por lo tanto, el tiempo total de ejecución será de $t = 50 n \log n$. Supongamos además que tenemos una computadora capaz de ejecutar 10,000,000 de operaciones por segundo.

Comparemos ahora el desempeño de ambas soluciones, supongamos primero que queremos ordenar 1000 números.

$$\text{inserción } (t) = 2 (1000)^2 = 2,000,000 = 0.2 \text{ seg.}$$

$$\text{mezcla } (t) = 50 (1000) (11) = 550,000 = 0.055 \text{ seg.}$$

Como podemos observar, el desempeño de la ordenación por mezcla es más eficiente. La diferencia se va haciendo más radical conforme el tamaño de la entrada va incrementándose; por ejemplo, si quisiéramos ordenar 100,000,000 de números, los tiempos serían:

$$\text{inserción } (t) = 2 (100000000)^2 = 20,000,000,000,000,000 \text{ (Un poco más de 63 años!)}$$

$$\text{mezcla } (t) = 50 (100000000) (27) = 135,000,000,000 \text{ (3 horas con 45 minutos)}$$

Mientras que la ordenación *por inserción* tomaría todo el resto de sus vidas, la ordenación *por mezcla* apenas y necesitaría menos de 4 horas. De ahí que, realmente vale la pena realizar un análisis de complejidad de un algoritmo antes de implementarlo.

El estudio de complejidad está relacionado con el tipo de cómputo donde se aplica el algoritmo. Actualmente, los modelos de cómputo se pueden agrupar en dos grandes conjuntos (ver [32]): determinísticos y no determinísticos.

El modelo determinístico está asociado históricamente a la máquina de Turing (ver [33]), ya que esta máquina es un modelo matemático estándar que formaliza la noción de computadora.

Una máquina de Turing determinista está formada por un control basado en estados finitos, una cinta infinita en ambos sentidos dividida en celdas, y una cabeza de lectura y escritura. El determinismo estriba en que para una misma entrada, el resultado de aplicar repetidas veces un algoritmo, el resultado siempre es el mismo. El modelo no determinístico es equivalente a decir que pueden obtenerse resultados diversos para una misma entrada. Este modelo contiene dos fases: una fase de adivinación y una de verificación. La fase de adivinación (no determinista) escoge al azar una estructura que puede ser la solución, y la fase de verificación (determinista) consiste en determinar si la estructura escogida es una solución.

Clases P y NP

La complejidad computacional de un problema es determinado por el mejor algoritmo que puede encontrarse para resolver el problema (ver [13]). En un nivel de abstracción alto, si puede encontrarse un algoritmo de tiempo polinomial para un problema dado, entonces el problema es considerado tratable o no tan duro. Pero si tal algoritmo no puede encontrarse, es decir, solo algoritmos en tiempo exponencial pueden ser construidos, el problema se considera intratable o muy duro, aún si el problema es soluble. La teoría de la complejidad computacional ha sido desarrollada considerando principalmente a los problemas de decisión. Muchos de los problemas de optimización pueden ser expresados como un problema de decisión. Un problema de decisión es un problema para el cual la respuesta es "sí" o "no" de acuerdo a si las entradas satisfacen o no las condiciones dadas en el problema.

Algunos ejemplos de problemas de decisión son dados a continuación.

- *Par*. Dado un número natural n , ¿Es n un número par? La respuesta es “sí”, si n es par, o “no”, si n es impar.
- *Primo*. Dado un número natural n , ¿Es n un número primo? La respuesta es “sí”, si n es primo, o “no”, si n es compuesto.
- *Satisfactibilidad*. Dada una expresión booleana $f(x_1, x_2, \dots, x_n)$, ¿pueden las variables x_1, x_2, \dots, x_n ser fijadas a valores que hagan el valor de f “verdadero”? La respuesta es “sí”, si hay una configuración de las variables que hagan a f “verdadero”, y “no” de otro modo.
- *Ciclo Hamiltoniano*. Dado un grafo $G(V, E)$ con N nodos, ¿hay un ciclo de aristas en G que incluya cada uno de los N nodos? La respuesta es “sí”, si tal ciclo existe, y “no” de otro modo.
- *Asignación de Espacio*. Dada n entidades y m cuartos disponibles, ¿es posible construir una asignación de las n entidades a los m cuartos, de tal manera que todas las restricciones existentes (duras y suaves) sean satisfechas y el espacio desperdiciado sea a lo mucho W ? La respuesta es “sí”, si tal asignación existe, y “no” de otro modo.

Hay dos clases en las cuales los problemas son clasificados (ver [13] y [37]): clases **P** y **NP**. La clase **P** está constituida por todos aquellos problemas para los cuales un algoritmo determinístico en tiempo polinomial ha sido encontrado. La clase **NP** (por sus siglas en inglés, *Non-deterministic Polynomial*) incluye a todos aquellos problemas para los cuales un algoritmo de tiempo polinomial no determinístico es conocido para resolver el problema. Un algoritmo no determinístico puede ser descrito como consistencia de dos fases. La primera fase supone una estructura para el problema y la segunda fase verifica si la estructura dada es o no una solución del problema. Entonces, el algoritmo se dice ser un algoritmo de tiempo polinomial no determinístico si para cada instancia del problema existe la conjetura de que puede ser verificado por una fase determinística para una respuesta “sí” en tiempo polinomial.

Entonces, si la clase **P** son los problemas resueltos en tiempo polinomial por un algoritmo determinístico y la clase **NP** son los problemas resueltos en tiempo polinomial por un algoritmo no determinístico, la pregunta es si $\mathbf{P} = \mathbf{NP}$ o $\mathbf{P} \neq \mathbf{NP}$. De hecho, esta es la pregunta abierta más importante en la teoría de la complejidad computacional. Es claro que $\mathbf{P} \subseteq \mathbf{NP}$, lo cual significa que los algoritmos no determinísticos son más poderosos que los algoritmos determinísticos. Si hay un algoritmo determinístico para

un problema, uno no determinístico puede ser construido simplemente al no usar una fase suposición.

Hay muchos problemas conocidos con certeza que pertenecen a la clase **NP**, para los cuales un algoritmo eficiente no ha sido encontrado, y estos problemas son considerados *NP-duros* en sentido fuerte, es decir, estos son intratables. Esto confirma la creencia de que $P \neq NP$, pero ésta es simplemente una conjetura que aún no ha sido demostrada.

Si es verdad que $P \neq NP$, entonces los problemas en el conjunto $NP \setminus P$ son intratables. Por tanto, cuando un problema en particular es abordado, es importante saber si el problema pertenece a la clase de los problemas tratables o intratables. Una manera de hacer esto es determinar si el problema de interés, está o no relacionado a otro problema que ha sido anteriormente demostrado ser tratable o intratable. Reduciendo un problema a otro, es la técnica usada para demostrar si los dos problemas se relacionan o no. La reducción es provista por una transformación que permite el mapeo de una instancia del primer problema dentro de una instancia del segundo problema. Esta transformación permite convertir un algoritmo que resuelve uno de los problemas a un algoritmo que resuelva al otro problema.

Transformación Polinomial

Una transformación polinomial es una función que permite cambiar la representación de problema D_1 a otro problema D_2 aplicando un algoritmo determinista de tiempo polinomial. Lo anterior se puede representar como " D_1 se transforma a D_2 " o $D_1 \propto D_2$. Cualquier elemento del problema D_1 tiene un elemento equivalente en D_2 .

Las transformaciones polinomiales son importantes porque sirven para determinar la pertenencia de los problemas a las clases **P** y **NP**, y permiten definir la clase *NP-completo*.

Si D_1 se transforma a D_2 y D_2 pertenece a la clase **P**, entonces D_1 también pertenece a la clase **P**, porque si para cambiar de D_1 a D_2 se utiliza un algoritmo de tiempo polinomial y si D_2 (conocido con certeza dentro de la clase **P**) tiene un algoritmo de solución que se ejecuta en tiempo polinomial, D_1 debe tener asociado un algoritmo que lo resuelva también en tiempo polinomial, y D_1 debe pertenecer a la clase **P**. Se puede decir en general, que un problema D_1 está en **P** si cualquier problema que se sabe está en **P**, se puede transformar a D_1 . También se puede decir que un problema D_1 está en **NP**, si cualquier problema que se sabe pertenece a la clase **NP**, se puede transformar a D_1 .

Un problema D_1 es *NP-completo* si pertenece a la clase **NP**, y otro problema D_2 que también pertenece a **NP**, se puede transformar a D_1 . Los problemas *NP-completos* son los problemas más difíciles en la clase **NP**. La palabra *completo* significa que la solución de un problema de decisión **NP**, contiene de alguna forma, la solución a todos los problemas de decisión de la clase **NP**.

Si D_1 se transforma a D_2 y D_2 es un problema *NP-completo*, entonces D_1 también es un problema *NP-completo*. Esto es importante, ya que establece que se puede probar que un problema de decisión D_1 es *NP-completo*, si algún problema que se sabe es *NP-completo* puede transformarse al primero.

Problemas NP-duros

En ocasiones puede demostrarse que los problemas en **NP** pueden transformarse a algún problema A ($D_{NP} \propto A$), pero no puede decirse que A sea **NP** o *NP-completo*. Sin embargo, es indudable que A es tan difícil (duro) como cualquier problema en **NP**, y sea entonces intratable. Para este tipo de problemas se asocia el término de *NP-duros*.

Dentro de los problemas considerados *NP-completos*, el primer trabajo reportado en la literatura fue el problema de Satisfactibilidad (SAT) demostrado por Cook en 1971 (ver [7]).

A saber, el problema de Satisfactibilidad es un problema central en la lógica matemática y la teoría de la computación. SAT es fundamental para la solución de varios problemas en razonamiento automático, diseño y manufactura asistida por computadoras, planificación, visión computacional, bases de datos, robótica, diseño de circuitos integrados, de arquitectura de computadoras y de redes de computadoras, entre otros.

La satisfactibilidad proposicional es el problema de decidir si existe una asignación de valores de verdad a los átomos de una fórmula proposicional que la hacen verdadera.

Entre otros resultados, Cook probó que cualquier problema en **NP** puede ser reducido al problema de Satisfactibilidad. Esto significa que si hay un algoritmo eficiente para resolver el problema de Satisfactibilidad, entonces cualquier problema en **NP** puede también ser resuelto por un algoritmo eficiente. Estos problemas se dicen ser *NP-completos* y son considerados los más duros dentro de la clase **NP**. Esto es, porque si un problema *NP-completo* no tiene un algoritmo eficiente para resolverlo, entonces ninguno de ellos tiene un algoritmo eficiente y son todos intratables. Muchos problemas han sido demostrados ser *NP-completos* (o reducidos al problema de Satisfactibilidad),

pero esto aún no prueba que estos problemas sean intratables. Sin embargo, generalmente se asume que encontrar un algoritmo eficiente para cualquier problema dentro de la clase *NP-completo* es improbable.

Por tanto, si un problema es *NP-completo* y $P \neq NP$, entonces el problema pertenece al conjunto $NP \setminus P$. En otras palabras, el problema (y todos en *NP-completo*) pueden pertenecer a P , sí y solo sí $P = NP$. Así, si se asume que los problemas *NP-completos* son intratables, es decir, $P \neq NP$, y un problema ha sido demostrado ser *NP-completo*, entonces el enfoque no debe estar en encontrar un algoritmo eficiente, sino en diseñar algoritmos que produzcan soluciones de alta calidad, sin garantizar la optimalidad; es decir, diseñar algoritmos útiles que aborden al problema en la práctica.

Conclusiones

Es muy importante que siempre que se piense en un algoritmo, se realice un análisis de complejidad del algoritmo, aunque no todos los algoritmos son tan sencillos de analizar, para la mayoría es fácil determinar la función de crecimiento del mismo.

Determinar la complejidad del algoritmo puede dar una idea muy clara de cuánto tiempo va a tardar el mismo, y en base a los tamaños de entrada poder determinar si el algoritmo se ejecutará en un tiempo práctico o no.

D.2 Análisis de Complejidad de los Procedimientos Desarrollados

La metodología de solución del PMCC puede ser descrita por la secuencia de los algoritmos presentada por la Figura D2.

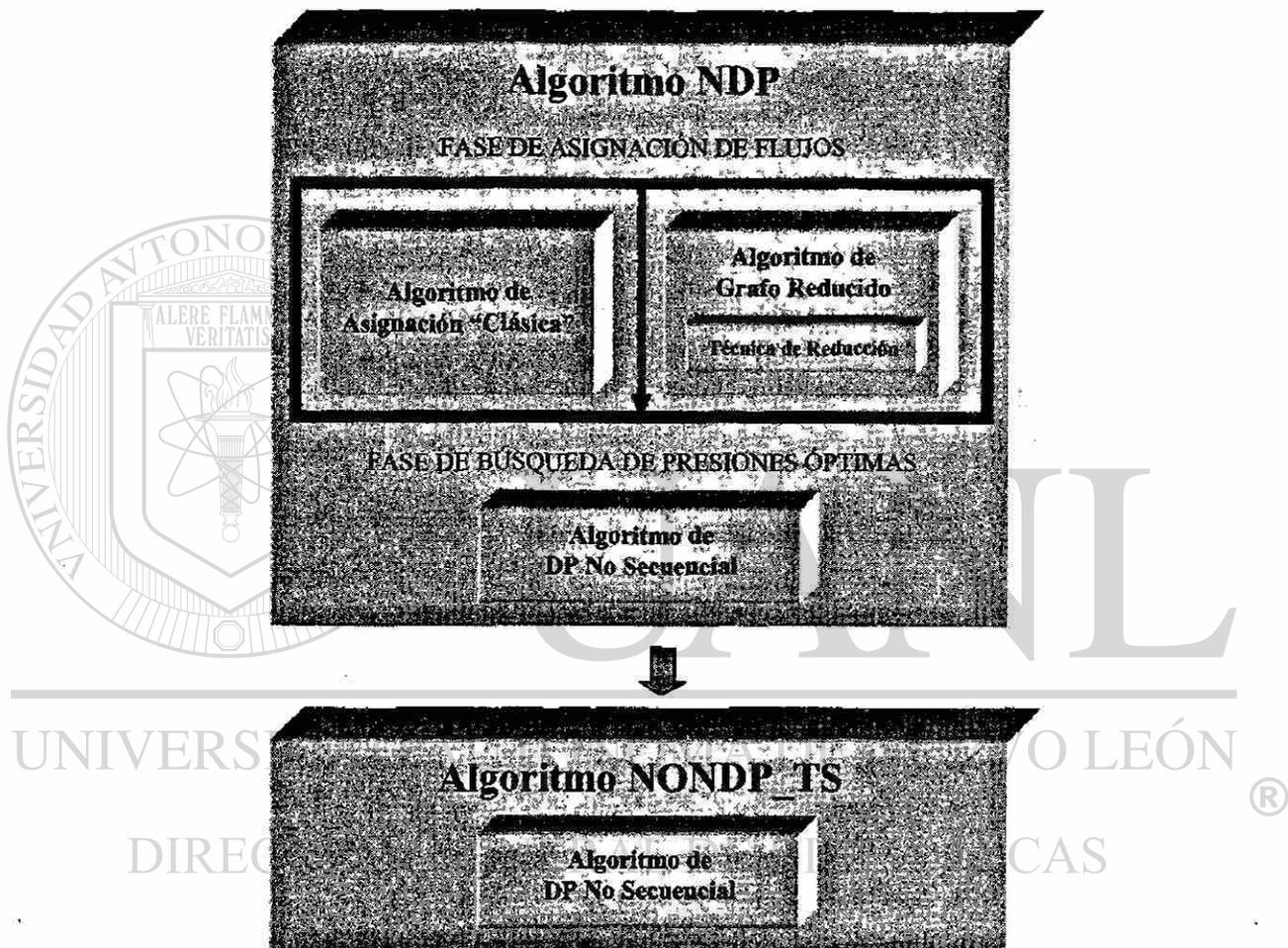


Figura D2: Secuencia de los algoritmos que conforman la metodología de solución.

Revisemos ahora el costo computacional de los algoritmos partiendo desde el centro hacia fuera de la secuencia presentada por la Figura D2.

Técnica de Reducción

Complejidad Computacional: Para referencia, véase Sección 4.1.3 y Fig. 4.1. El Paso 1 del algoritmo toma $O(|A_c|)$ operaciones. Para los Pasos 2 y 3, hay que visitar todos los arcos y nodos restantes por lo que toma un tiempo $O(|V|+|A_p|)$. El Paso 4, al igual que el Paso 1, toma un tiempo $O(|A_c|)$. Por consiguiente, asumiendo que $|V|$ esta acotado por $|A|$, el procedimiento es $O(|A|)$.

Algoritmo de Grafo Reducido

Complejidad Computacional: Para referencia, véase pág. 42 y Fig. 4.10. El Paso 1 toma $O(|A|)$ operaciones como ya se vio en la Sección 4.1.3. En el resto del algoritmo, nótese que cada arco se visita una sola vez. En esta visita, se hacen cálculos y asignaciones de $O(1)$ (Pasos 3-5), y una vez asignado el flujo al arco, este se elimina y no vuelve a tomarse en cuenta. Por tanto, el algoritmo es $O(|A|)$.

Algoritmo de Asignación Clásica

Complejidad Computacional: Para referencia, véase pág. 38 y Fig. 4.5. El algoritmo consiste básicamente en encontrar una ruta de un nodo oferta a un nodo demanda y enviar el flujo requerido. Como se asume que siempre hay un flujo factible, y en el peor de los casos puede enviarse una unidad de flujo en cada ocasión (asumiendo flujos enteros), el algoritmo tiene complejidad $O(|B|)$, donde $B = \sum_{i \in V_d} B_i$. Sin embargo, en la práctica nunca se ha observado tal comportamiento, ya que casi siempre es posible enviar la cantidad completa disponible de B_k (en el Paso 2) en una sola iteración.

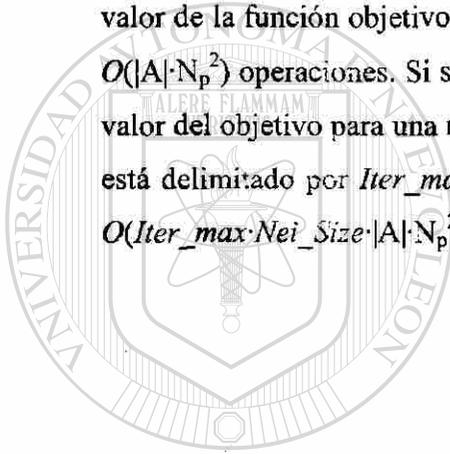
Algoritmo de DP No Secuencial

Complejidad Computacional: Para referencia, véase Sección 4.2.2 y Fig. 4.14. Asumiendo una rango de presiones ΔP , el número máximo de puntos está acotado por $p_{max}/\Delta P = N_p$, donde p_{max} es el máximo valor de las cotas superiores de las presiones.

Así las cosas, cada “optimización” (en el Paso 2) toma en el peor de los casos $O(N_p^2)$ operaciones. Para $|A_c|$ compresores, esto se repite $|A_c|-1$ veces, por tanto la complejidad es $O(|A_c| \cdot N_p^2)$. Esto implica que el procedimiento global NDP (Fig. 4.2) es $O(|A| \cdot N_p^2)$.

Algoritmo NONDP_TS

Complejidad Computacional: Para referencia, véase Sección 5.3.3 y Fig. 5.2. Como puede verse, el tiempo de cómputo está dominado por el trabajo de generación de los vecinos y cálculo de su correspondiente valor de la función objetivo. El número de vecinos está acotado por *Nei_Size* (parámetro definido por el usuario) y para calcular su valor de la función objetivo se aplica el procedimiento NDP, que toma, como ya se vio, $O(|A| \cdot N_p^2)$ operaciones. Si se implementa inteligentemente, no es necesario recalcular el valor del objetivo para una misma solución más de una vez. Como el ciclo del algoritmo está delimitado por *Iter_max*, el procedimiento NONDP_TS tiene una complejidad de $O(Iter_max \cdot Nei_Size \cdot |A| \cdot N_p^2)$.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Ficha Autobiográfica

Conrado Borraz Sánchez

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

Universidad Autónoma de Nuevo León

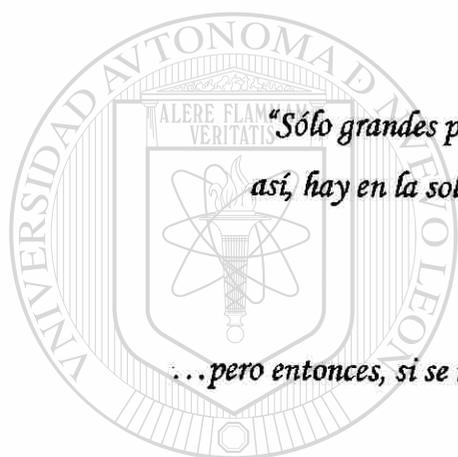
Facultad de Ingeniería Mecánica y Eléctrica

Tesis

***Una Metodología de Solución basada en Programación Dinámica
No Secuencial y Búsqueda Tabú para la Operación Eficiente de
Sistemas de Transporte de Gas Natural en Estado Estable***

Nacido en Tuxtla Gutiérrez, Chiapas. Hijo del Sr. Conrado de Jesús Borraz León y Sra. Magnolia Sánchez de Borraz, segundo de cuatro hermanos. Graduado en el Instituto Tecnológico de Tuxtla Gutiérrez (1997-2001) como Ingeniero en Sistemas Computacionales, y en el Centro de Estudios de Computación del Sureste (1993-1995) como Técnico Programador y Analista de Sistemas. Inició sus estudios de Maestría en Ciencias en Ingeniería de Sistemas en Febrero del 2002, después de haber realizado un verano de investigación científico en la Universidad Autónoma de Nuevo León. Logrando así, una beca de manutención por el proyecto J33187-A de CONACYT; desempeñándose entonces como Asistente de Investigación durante sus estudios de Maestría.

Pensemos que:



*“Sólo grandes pensamientos permiten resolver grandes problemas,
así, hay en la solución de todo problema, un poco de magnificencia”*

...pero entonces, si se resuelve un problema y llega a excitar nuestra curiosidad,

UANL

*“este género de experiencia, a una determinada edad, puede determinar el gusto del
trabajo intelectual y dejar, tanto en el espíritu como en el carácter
...una huella que durará toda una vida”.*

DIRECCIÓN GENERAL DE BIBLIOTECAS



