



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERIA ELÉCTRICA

ESTUDIO COMPARATIVO DE LA RELAJACIÓN LAGRANGEANA Y LA
PROGRAMACIÓN ENTERA-MIXTA EN EL PROBLEMA DEL PRE-DESPACHO DE
SISTEMAS MEDIANOS

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CS. DE LA INGENIERÍA,
MENCIÓN INGENIERÍA ELÉCTRICA.

FRANK JOSE LEAÑEZ GRAU

PROFESOR GUÍA:
RODRIGO PALMA-BEHNKE

MIEMBROS DE LA COMISIÓN:
LUIS VARGAS DÍAZ
HUGH RUDNICK

SANTIAGO DE CHILE
2013

ESTUDIO COMPARATIVO DE LA RELAJACIÓN LAGRANGEANA Y LA PROGRAMACIÓN ENTERA-MIXTA EN EL PROBLEMA DEL PRE-DESPACHO DE SISTEMAS MEDIANOS

El trabajo de tesis compara en forma sistemática las diferencias de rendimiento entre las principales metodologías de solución del problema de Predespacho (UC) y muestra la aplicación de la relajación lagrangeana a casos reales del Sistema Interconectado del Norte Grande (SING). Se propone una metodología para la creación de casos sintéticos de UC que permitan obtener diversidad a distintas escalas sin que resulten alejados de la realidad, siendo ésta la principal contribución del presente trabajo junto con los resultados estadísticos que favorecen MIP por sobre LR para sistemas de medianas dimensiones.

La relajación lagrangeana (LR) y la programación entero mixta (MIP) son los métodos de solución que han encontrado el mayor número de aplicaciones prácticas al UC. Sin embargo, las comparaciones de rendimiento y calidad entre ambas, aunque propuestas conceptualmente y cuyas nociones intuitivas se encuentran dispersas en la bibliografía especializada, adolecen de falta de generalidad, escasa representación de la realidad, ausencia de especificaciones computacionales o, inclusive, éstas pueden haber quedado obsoletas. Ante la escasez de librerías de modelos de UC o, al menos, de métodos de creación de casos de aplicación, en el presente trabajo se propone y aplica un generador de casos sintéticos de prueba de UC. Este método formula casos basados en la combinación de datos estandarizados con la generación aleatoria de parámetros. De esta forma, los problemas de UC creados adquieren diversidad (universalidad) sin que las instancias se alejen demasiado de la realidad. La diversidad de las instancias generadas es controlada mediante los parámetros de las funciones de distribución de probabilidades, tanto para la selección de unidades candidatas como para la variedad en los parámetros técnicos característicos. Estas instancias son resueltas por los métodos para resolver el UC en una misma plataforma computacional.

El método de LR encuentra soluciones factibles para 429 instancias de las 480 (89,4%), cumpliendo con el *gap* objetivo de 0,01% en tan sólo 90 casos (18,8%), mientras que el MIP encuentra la solución óptima para el 98,3% de los casos. Los resultados obtenidos verifican que para el rango de 10 a 100 unidades, la mejor solución entera factible alcanzada por LR tiene una menor calidad (valor mayor de la función objetivo en problemas de minimización) que la encontrada por el optimizador MIP. Los resultados permiten determinar que el sobrecosto (respecto al MIP) esperado de las soluciones mediante LR fue de 6,03%. El aumento de la dispersión en los parámetros técnicos beneficia ambos métodos al obtener la mejor solución factible en menores tiempos de ejecución. De esta forma se muestra que no sólo la calidad de la solución por LR se ve afectada por unidades similares, sino que también afecta al método MIP. Se comprueba que a medida que los problemas son más difíciles (juzgando por la adaptabilidad y por el número de unidades), la cota inferior de LR resulta en promedio mayor (mejor) a la cota inferior promedio por MIP.

Respecto al caso práctico inspirado en el SING, la calidad de la solución por LR se ve especialmente deteriorada cuando las restricciones de mínimos técnicos de las unidades obligan que la solución óptima entera se aleje de la relajación lineal. Si bien para este caso la resolución mediante MIP también se dificulta requiriendo de hasta un 60% más tiempo de ejecución, este efecto es comparativamente menor al deterioro de la calidad de la solución observada mediante LR (que resulta hasta 4.5% mayor que MIP). Conceptualmente, este hallazgo se explica por los valores de actualización de los multiplicadores en LR, lo que se traduce en imposibilidad para explorar zonas del espacio de soluciones.

Como conclusión general, el estudio entrega evidencia práctica a favor de la tendencia observada en la revisión bibliográfica en relación a privilegiar desarrollos de tipo MIP, la cual es aplicable a sistemas medianos como el caso real en estudio basado en el SING chileno. Esta conclusión no se extiende para otros sistemas “reales” de grandes dimensiones (eg. CAISO, MISO, UCTE). Como futuros desarrollos se sugiere explorar esquemas integrados LR-MIP explotando las mejoras de la cota inferior que proporciona LR en comparación con la relajación lineal del MIP y la creación de soluciones factibles por el método LR para inicializar el MIP.

Tabla de Contenido

Capítulo 1 Introducción.....	6
1.1 Objetivos	6
1.2 Motivación	6
1.2.1 Reconocimiento bibliográfico como las más efectivas	8
1.2.2 Coexistencia de ambos métodos como motores de solución.....	9
1.2.3 Dudas acerca de la calidad de la solución de la LR.....	10
1.2.4 Dudas acerca de los tiempos de ejecución del MIP	11
1.2.5 Dudas acerca de la factibilidad de implementación de LR.....	13
1.3 Metodología	13
1.4 Organización.....	14
Capítulo 2 Definición del problema de Pre-despacho	16
2.1 Planteamiento del problema	16
2.2 Métodos de Solución.....	19
2.2.1 Revisión bibliográfica métodos de solución.....	20
2.2.2 Relajación Lagrangeana.....	23
2.2.3 Programación entero-mixta	26
2.3 Resumen.....	28
Capítulo 3 Algoritmo de UC por relajación Lagrangeana	29
3.1 Introducción	29
3.2 Generalidades del Método.....	29

3.2.1	Inicialización o relajación de los multiplicadores de Lagrange	30
3.2.2	Evaluación de la Función Dual.....	32
3.2.3	Despacho Económico sin Restricciones de Transmisión	32
3.2.4	Verificación de Convergencia	33
3.2.5	Actualización de los Multiplicadores.....	33
3.3	Resumen.....	35
Capítulo 4 Comparaciones MIP vs. LR.....		36
4.1	Introducción	36
4.2	Comparación bibliográfica	37
4.2.1	Factores de Ponderación.....	38
4.2.2	Tiempos de Ejecución.....	41
4.2.3	Calidad de las Soluciones	43
4.2.4	Tiempo vs. Optimalidad	44
4.2.5	Análisis de Resultados	45
4.3	Comparaciones Experimentales.....	45
4.3.1	Ejemplo sintético.....	45
4.3.2	Comparación estadística	55
4.3.3	Aplicación al SING.....	75
4.4	Resumen.....	88
Capítulo 5 Discusión y Conclusiones		89
Capítulo 6 Bibliografía		93

Capítulo 7 ANEXOS	100
7.1 ANEXO A - Relajación Lagrangeana: Teoremas de Dualidad.....	100
7.1.1 Introducción	100
7.1.2 Optimización Dual [1].....	100
7.1.3 Teoremas de Dualidad de la Función Lagrangeana.....	101
7.1.4 Dual en Programación Lineal (LP) [2]	102
7.1.5 Referencias.....	103
7.2 ANEXO B – Programación Dinámica Individual de Unidades	104
7.2.1 Introducción	104
7.2.2 Algoritmo de Solución	106
7.2.3 Referencias.....	107
7.3 ANEXO C - Método Subgradiente.....	108
7.3.1 Introducción	108
7.3.2 Reglas de control de los pasos	109
7.3.3 Convergencia.....	110
7.3.4 Referencias.....	111
7.4 ANEXO D - Relajación Lagrangeana Aumentada.....	112
7.4.1 Introducción	112
7.4.2 Técnicas de Descomposición.....	113
7.4.3 Referencias.....	115
7.5 ANEXO E – DATOS DEL SISTEMA RTS-96	116

7.5.1	Datos del RTS-96 necesarios para el pre-despacho	116
7.5.2	Construcción de formulaciones de pre-despacho	117
7.5.3	Referencias	118

Capítulo 1

Introducción

1.1 Objetivos

El objetivo central de esta tesis es realizar comparaciones prácticas entre los métodos de programación entero-mixta (MIP), basados en el método *branch-and-bound* en optimizadores de uso general y el método por relajación lagrangeana (LR) al resolver el problema del Predespacho. La forma de realizar las comparaciones se basan en 3 documentados métodos de comparación de algoritmos de optimización: 1) Histórico (bibliográfico) 2) Sintético-Aleatorio y 3) Práctico.

El presente trabajo reúne en un contexto uniforme las ideas intuitivas acerca de los desaciertos de la relajación lagrangeana como método general de solución, tales como problemas para encontrar soluciones factibles cuando coexisten unidades idénticas (o muy similares) o la necesidad de sintonización de los parámetros del subgradiente y las expone mediante comparaciones estadísticas de los resultados de optimalidad versus requerimiento computacional, lo cual es otro de los aportes de la presente tesis.

Finalmente, se desea verificar la aplicabilidad de ambos métodos a casos reales de Predespacho de mediana escala basados en las características de las unidades y demanda del sistema eléctrico chileno SING.

1.2 Motivación

El problema de asignación de unidades o Predespacho, en adelante abreviado por sus siglas en inglés *Unit Commitment* (UC), se plantea en forma generalizada como un problema de optimización no-lineal, entero-mixto, de gran escala, del tipo combinatorial np-complete [1]. La dimensionalidad del espacio combinatorial hace que sea imposible en la práctica resolverlo mediante una enumeración exhaustiva de

todas las combinaciones posibles. Por lo tanto, se han planteado en la bibliografía una diversidad de algoritmos de solución, basados en métodos matemáticos o heurísticos, que prometen encontrar soluciones cercanas al óptimo en tiempos de cómputo razonables. En [2] es posible encontrar una amplia revisión bibliográfica conceptual de una diversidad de métodos propuestos en los últimos 20 años de investigación especializada en UC. Aunque la revisión comprende 151 artículos publicados en revistas especializadas, no están orientados a comparar métodos en el sentido práctico. En la referencia [3] en cambio, se hace un esfuerzo por concentrar resultados de rendimiento de las pruebas realizadas y reportadas en la bibliografía, “normalizando” las comparaciones al multiplicar o penalizar los tiempos de ejecución obtenidos por cada algoritmo por un factor de ajuste basado en la potencia de la plataforma computacional donde fueron ejecutados respectivamente. Este factor fue obtenido basado en la estadística de rendimiento de los computadores en distintos benchmarks internacionales, agregando una banda de error proporcionada por las máximas diferencias de los distintos resultados consultados.

En resumen, las principales dificultades para llevar a cabo una comparación de rendimiento son:

- No existe en la bibliografía de UC un problema genérico tipo benchmark sobre el cual se puedan realizar los experimentos y, por lo tanto, establecer comparaciones de rendimiento entre dos o más algoritmos probablemente se refieren a comparaciones de casos numéricos distintos.
- No existen datos de las instancias utilizadas en los 151 reportes con experimentos relacionados al UC en [2], más otros tantos referidos en [3]. En tan sólo diez de ellos es posible extraer especificaciones mínimas, tanto computacionales como de características de los casos de prueba.
- Otras particularidades tales como la escalabilidad del problema sólo ha sido explorada en función del número de unidades, la modelación no siempre se refiere a una estructura idéntica del problema en cuanto a las restricciones y la función objetivo, dependencia con parámetros de sintonización de los algoritmos evaluados.

Continuando con [3], se concluyó que históricamente y como respuesta a la ausencia de librerías benchmark de UC, un conjunto de autores han empleado un ejemplo sintético, cuyos datos se

encuentran sintetizados en la misma referencia, que entre otras características presenta las siguientes limitaciones:

- Se especifican curvas de costo cuadráticas para las unidades generadoras, lo cual dificulta la comparación con métodos lineales.
- Los problemas de mayores dimensiones se obtienen repitiendo el conjunto base compuesto por 10 unidades generadoras.
- Se mantiene la curva de demanda, escalando la demanda máxima proporcional al número de unidades totales.

Por otra parte, a partir de su versión 1996, el “IEEE Reliability Test System” [4], presenta un conjunto muy reducido, de apenas nueve unidades termoeléctricas pertenecientes a tecnologías diferentes, con datos suficientes para preparar ejemplos de UC sintéticos. A pesar que [4] pudo haber sido empleado para crear instancias mínimas de UC, a penas en una de las referencias revisadas en [3], se tomaron sus datos sistémicos para la elaboración de los casos numéricos.

1.2.1 Reconocimiento bibliográfico como las más efectivas

Entonces, la dificultad de comparaciones extiende las especificaciones computacionales de las pruebas, habiéndose detectado que además existe una carencia de librería(s) de casos de prueba en la bibliografía sobre la cual construir instancias de prueba de UC. Por lo anterior, el esfuerzo de normalizar los resultados en [3], de acuerdo a lo expuesto anteriormente, tuvieron que limitarse a apenas 10 reportes prácticos con UC con las siguientes características explícitas:

- Datos de velocidad del procesador.
- Ejemplos numéricos similares (no idénticos) del orden de 10-100 unidades.
- Resultados de tiempo computacional de ejecución y valor de la mejor solución entero-factible.

Los resultados normalizados corresponden a algunas nociones intuitivamente aceptadas, en donde un método de programación MIP (B&B en combinación con CLP) y la relajación lagrangeana LR, requirieron, para una calidad de solución comparable, un tiempo de ejecución órdenes de magnitud menores

respecto a las metaheurísticas. Quiere decir que, inclusive considerando amplios márgenes de error, los casos puntuales comparables que se pueden rescatar de la bibliografía referente al UC, penalizando aquellos realizados en computadores “más modernos”, entregan respuestas similares a aquellas que se pueden conseguir en forma dispersa en los distintos software y papers de discusión referente a los métodos de solución del UC. De hecho en [5], se describe un proceso de evolución de un sistema comercial de UC, donde se abandona la resolución del problema mediante el método de LR por el MIP resuelto mediante una rutina de uso general CPLEX. Aunque es de relevancia y parece anotar una victoria del MIP sobre la LR, esta comparación sin embargo, fue diseñada y aplicada a un sistema específico de potencia de grandes dimensiones (PJM). Una ventaja que se apunta el uso de rutinas comerciales MIP es la facilidad de modelación, dado que no dependen (como LR) de una estructura específica del problema que permita la descomposición.

1.2.2 Coexistencia de ambos métodos como motores de solución

De acuerdo con [6], LR y MIP son los dos principales contendientes para resolver en problema de UC. Existe una diferencia de esfuerzos de desarrollo computacional notable entre ambos. Mientras que la relajación debe estar profundamente integrada con el proceso completo de solución del problema, las soluciones MIP tienden entregar a un optimizador, tipo “caja negra”, en ocasiones en forma iterativa. El agregar nuevas restricciones para ser mucho más fácil en MIP, dado que no requiere la integración de un nuevo multiplicador que actualizar [5]. Se dice que MIP tiende a buscar en un espacio mayor en problemas con numerosas restricciones, pero que su convergencia no está asegurada. También en [6] se asegura que para problemas con algunas decenas de periodos ambos métodos tienen un desempeño similar, sin embargo, de igual forma se señala que una mayor fuente de comparaciones entre ambos métodos se hace necesaria.

1.2.3 Dudas acerca de la calidad de la solución de la LR

La relajación lagrangeana logró posicionarse como uno de los algoritmos más efectivos para resolver problemas de UC de grandes dimensiones. El desacople de las restricciones “difíciles”, fue una de las ideas que mejores avances introdujo al desarrollo de algoritmos de optimización en décadas pasadas [7]. Conceptualmente, la relajación resultó prodigiosamente atractiva para resolver problemas de UC por dos motivos principales:

- La estructura intrínseca del UC permite clasificar las restricciones en dos grandes bloques: las propias de las unidades (eg. potencias máximas/mínimas, tiempos mínimos de operación/detención) y las sistémicas (eg. balance de energía, requerimiento de reserva en giro).
- Porque los subproblemas luego del desacople, parecen fáciles de resolver aun cuando cumplen con un conjunto extenso de restricciones (el problema de asignación individual de unidades).

Sin embargo, la utilización de varios multiplicadores para la modelación de las restricciones complicadas y la aplicación de heurísticas para actualizar los valores de los multiplicadores, puede llegar a ser poco práctico para la modelación de sistemas eléctricos reales. En general, no es posible garantizar un conjunto de multiplicadores para el cual la solución a la relajación será igual al óptimo del problema completo, aunque esto puede suceder por razones particulares, siendo el mejor conjunto aquél que maximiza el dual del problema. Sin embargo, el óptimo del dual no necesariamente tiene que ser factible por lo que se han propuesto una diversidad de alternativas de perturbación de la vecindad al máximo del dual, para obtener soluciones factibles de calidad [7]. Lo que puede ser inclusive peor, puede que durante todo el proceso de maximización no se encuentre ninguna solución factible. Por esta razón, la relajación lagrangeana ha sido cuestionada como método de solución del UC porque la pura maximización no generaliza su aplicación a cualquier caso, por lo que rutinas basadas en heurísticas ad-hoc son requeridas. Para darle generalidad a la LR, post-procesos de búsqueda y mejoras de soluciones factibles son requeridos como parte integral del algoritmo de solución del UC [8].

1.2.4 Dudas acerca de los tiempos de ejecución del MIP

La garantía de optimalidad del branch-and-bound, usualmente medida por el gap entre la mejor solución entera y la cota inferior (solución a la relajación LP), suele ser la señal de optimalidad que entregan los optimizadores comerciales de uso general. El valor del gap que mejores resultados puede traer a un sistema determinado, debe ser obtenido de forma empírica y su efecto ha sido estudiado escasamente en la bibliografía especializada del UC.

Los problemas de UC son considerados de la clase de problemas difíciles tipo np-completos [9]. Aunque es cierto que se han reportado aplicaciones a problemas de dimensiones reales, por ejemplo en [5], las características intrínsecas del problema imposibilitan garantizar la generalidad de los de los tiempos de ejecución para todas las instancias. En efecto en [10] se muestran ejemplos numéricos que, basándose en un sistema de pocas unidades cuyo horizonte de evaluación fue discretizado en 252 periodos, los tiempos de ejecución en un computador moderno, para 2 de los 10 casos analizados en detalle, superó los 1800 segundos (30 minutos) de cómputo sin satisfacer un valor de gap objetivo del 0.4%. Quiere decir que el problema de UC resuelto por la misma rutina comercial que en [5] -aunque mediante una versión más avanzada- para un problema mucho más sencillo en cuanto al número de unidades (apenas 32 unidades aunque un horizonte mayor de 2 semanas), puede requerir 72 veces más tiempo de cómputo para intentar satisfacer la misma calidad de solución. Siguiendo con [10], la dispersión de los tiempos de ejecución para los 10 casos también fue elevada. Alrededor del promedio, la desviación estándar alcanzó el 85%.

Conceptualmente, la gran mayoría de las veces que el UC de dimensiones reales es ejecutado mediante una rutina MIP, a menos que el proceso logre obtener el óptimo global del problema completo, lo que ocurrirá en un número ínfimo de casos reales, la mejor solución entera que cumple con el gap deseado o alcanza el límite de tiempo, por lo general ha sido obtenida mediante las heurísticas generalizadas del optimizador. Entonces, surge la pregunta si las rutinas MIP pueden ser sistemáticamente aplicadas como motores de solución para el “market clearing”, la solución entera final delegada a las heurísticas primales

cumplen con una garantía de calidad suficiente, especialmente cuando es detenido el algoritmo en tiempos razonables, como para considerarse implementables en mercados reales. La dispersión de los tiempos de ejecución de los paquetes de optimización MIP permite realmente implementar un tomador de decisiones basado en resultados del UC. Podría realmente establecerse una relación confiable entre la calidad de la solución y el valor del gap objetivo/tiempo máximo de ejecución. Es posible implementar en un sistema como el SING una rutina branch-and-bound a sabiendas que casos similares pueden requerir más de 72 veces mayor tiempo computacional.

Estas preguntas son consistentes con los más recientes esfuerzos en por lograr formulaciones entero-mixta del UC usando MIP más “ajustadas” [11], [12], [13], [14] (Ajuste se refiere, en el contexto de problemas de minimización MIP, obtener en el nodo raíz del “branch-and-bound” una cota inferior mayor). Por ejemplo en [11] se propone una reformulación de las restricciones de tiempo mínimos de operación, condiciones iniciales, potencia máxima/mínima que proveen, para los casos numéricos resueltos, una cota inferior más ajustada. En [12] se propone la reformulación de las rampas de partida y detención de tal forma de favorecer la búsqueda de soluciones factibles con MIP. Los resultados de los casos experimentales muestran mejoras de la cota inferior en comparación con formulaciones clásicas, además de resultar en un formulación del UC más “compacta” (menos restricciones, variables y no-ceros). En [13] se formulan desigualdades (cortes) agregadas al problema UC que mejoran la envolvente convexa (“*convex hull*”), enfocándose en las restricciones de rampa fundamentalmente. De esta forma, se mejora (aumenta) la cota inferior en el nodo raíz en la totalidad de los experimentos. Sin embargo, pueden hacer que la relajación lineal en el nodo raíz resulte considerablemente más difícil de resolver. La referencia [14] se concentra en la reformulación de la función objetivo mediante una aproximación lineal por tramos que busca encontrar la división óptima en tramos de la función cuadrática de costos de operación. Los resultados prácticos permitieron concluir que “facilita” el proceso de solución mediante optimizadores MILP (“*Mixed-Integer Linear Programming*”).

El presente trabajo se concentrará en encontrarles respuesta a algunas de las grandes interrogantes planteadas anteriormente, especialmente orientadas a dar ideas al caso real del SING (sistemas medianos) que se usa como objeto central de la presente evaluación.

1.2.5 Dudas acerca de la factibilidad de implementación de LR

La relajación lagrangeana tiene serias dificultades para lidiar con unidades iguales [15]. Dado que en la etapa de la programación dinámica individual de las unidades, las señales de precio, en todas las iteraciones del algoritmo, son iguales para todas las unidades, el grupo de unidades idénticas (o muy similares) reacciona de igual forma a estas señales. De esta forma, serán encendidas/apagadas en forma simultánea. La manipulación de las curvas de eficiencia de las unidades para crear diversidad, puede no ser la forma más justa de resolver el problema. En [15] se señala al problema de unidades iguales como la razón central por la cual LR no pueda encontrar soluciones óptimas e incluso podría no encontrar soluciones factibles. En la misma referencia, se hace una sensibilidad para determinar “el porcentaje de similitud”, para un caso numérico de 4 unidades, que hace que la solución óptima no pueda ser alcanzada en ninguna de las fases del algoritmo. Sin embargo, la medición del deterioro de la calidad de las soluciones y la dificultad de obtención de soluciones factibles, las cuales no sólo corresponden a los casos de unidades idénticas, han estado lejos de poder haber sido identificados y mucho menos cuantificados.

1.3 Metodología

El estudio comprende abordar las comparaciones de rendimiento entre las metodologías MIP y relajación lagrangeana mediante una revisión bibliográfica y tres enfoques prácticos diferentes:

1. El primer enfoque del tipo empírico consistirá en resolver un caso “pequeño” de la bibliografía mediante ambos métodos, escalando el problema en función del número de unidades. Este es el enfoque clásico en la literatura de pre-despacho para la comparación de métodos de solución, pero con algunas variantes innovadoras, tales como análisis del efecto de inclusión de algunas restricciones,

ampliación del horizonte de evaluación, etc. Permitirá exponer la necesidad sintonización del algoritmo de relajación lagrangeana.

2. Sin dudas el enfoque central de la presente tesis será la comparación estadística del desempeño de las principales metodologías ante un conjunto de instancias amplio pero representativo. Los casos son creados mediante un generador aleatorio de instancias, el cual deberá cumplir con las principales sugerencias especializadas en el diseño de experimentos con métodos computacionales disponibles en la bibliografía. Las principales variables de rendimiento y calidad, serán analizadas en forma gráfica y estadística en funciones de los parámetros controlados de los experimentos.

3. Finalmente y considerando el interés que los países latinoamericanos deben tener en estudios de sistemas de dimensiones medianas, serán resueltas instancias puntuales interesantes de Predespacho, basadas en las características de las unidades y demanda del sistema chileno SING.

Sería ideal poder demostrar sistémicamente que para un universo suficientemente representativo del UC los tiempos de ejecución del MIP son estables y que efectivamente logra sobreponerse por sobre la mejor alternativa clásica, la relajación lagrangeana. Las instancias deberán ser acotadas para evitar caer en la trampa del no free lunch, pero con la suficiente generalidad para abordar las que caracterizan al UC.

1.4 Organización

La organización del presente trabajo es como sigue: En el siguiente capítulo se mostrará el planteamiento matemático del UC, la revisión bibliográfica de las metodologías, así como las generalidades de los dos métodos analizados en profundidad. El tercer capítulo describirá los fundamentos teóricos del algoritmo de solución por relajación lagrangeana. Se darán detalles de los pasos de ejecución de la rutina, la cual fue desarrollada y programada en JAVA en el marco de la presente tesis. El Capítulo 4 muestra un completo análisis empírico de las comparaciones prácticas entre ambos métodos de solución. El análisis comparativo se basa en tres enfoques distintos: la comparación de resultados existentes en la bibliografía haciendo uso de penalizadores a computadores antiguos (lo

cual es el primer aporte del presente trabajo) con lo cual se pretende responder a la pregunta de si son realmente las dos mejores alternativas según la bibliografía; la comparación estadística de soluciones sistemáticas a instancias sintéticas aleatorias del UC elaboradas en función de distintos factores de escalamiento y diversidad (lo cual es el segundo aporte de la presente tesis) con lo cual se intentará responder la difícil pregunta acerca de la generalidad y aplicabilidad del MIP versus LR; y, finalmente, se aplicará a algunas instancias del problema basado en las características de generación y demanda de un sistema real (sistema chileno SING) con el fin de resolver incógnitas relacionadas a las características de sistemas eléctricos medianos, característicos de Latino América. En el último capítulo se dará a conocer en un contexto uniforme los resultados más consecuentes de los análisis anteriores, así como puntualizar aquellos aspectos que claramente puedan señalar advertencias para la implementación de uno u otro método. También serán incluidas las futuras líneas de investigación que se prevén de interés relacionadas al tema.

Capítulo 2

Definición del problema de Pre-despacho

2.1 Planteamiento del problema

La primera dificultad que surge para poder establecer comparaciones de rendimiento entre distintos reportes del UC es la necesidad de resolver una simplificación del problema completo. Es decir, la bibliografía presenta una variedad de alternativas de solución para una versión simplificada del Unit Commitment Generalizado [1].

Esto da origen a la primera diferencia, las simplificaciones al UC no suelen ser homogéneas. Por ejemplo, en la referencia [16] se evalúa un modelo de UC con curvas de costos lineales por tramos, mientras que el popular ejemplo sintético en [17], empleado como benchmark en una variedad de experimentos según se muestra en [3], se entregan datos de curvas de costos totales cuadráticas. De hecho, alguno de los motivos de la rica variedad de modelos de UC es justamente la respuesta a hasta qué punto modelar cada vez más complejidades, de tal forma de acercarse a una propuesta para resolver el problema en [1]. Para todos los efectos de las comparaciones presentadas en este trabajo, el modelo de UC corresponde a una modelación MILP simplificada de [1], como se muestra a continuación:

Minimizar: Costos Operativos + Costos de Partida + Costos de Parada

$$\text{Min: } \sum_t^T \sum_i^I U_{i,t} \cdot FC(P_{i,t}) + S_{i,t}^{start} + S_{i,t}^{stop} \quad (2.1)$$

Sujeto a las siguientes restricciones:

- Restricciones de balance energético (horario):

$$\sum_i^I U_{i,t} \cdot P_{i,t} = \sum_c^C \hat{P}_{c,t}^{carga} \quad \forall t = 1, \dots, T \quad (2.2)$$

- Requerimientos de reserva en giro:

$$\sum_i^I U_{i,t} \cdot (P_{i,t}^{\max} - P_{i,t}) \geq R^t \quad \forall t = 1, \dots, T \quad (2.3)$$

- Límites de capacidad de las unidades:

$$P_{i,t}^{\min} \leq P_{i,t} \leq P_{i,t}^{\max} \quad \forall i = 1, \dots, I; \quad \forall t = 1, \dots, T \quad (2.4)$$

- Costos de Partida:

$$S_{i,t}^{\text{start}} = U_{i,t} \cdot (1 - U_{i,t-1}) \cdot \hat{S}_{i,t} \quad (2.5)$$

Donde:

$$\hat{S}_{i,t} \begin{cases} \hat{S}_i^{\text{hot}} & \text{si } t_{i,t}^{\text{off}} \leq \hat{t}_i^{\text{hot}} \\ \hat{S}_i^{\text{warm}} & \text{si } \hat{t}_i^{\text{hot}} < t_{i,t}^{\text{off}} < \hat{t}_i^{\text{cold}} \\ \hat{S}_i^{\text{cold}} & \text{si } \hat{t}_i^{\text{cold}} \leq t_{i,t}^{\text{off}} \end{cases} \quad (2.6)$$

- Límites de rampa (tasas de toma de carga):

$$P_{i,t} - P_{i,t-1} \leq \hat{G}_i^{\text{Up}} \cdot 60 \quad \forall i = 1, \dots, I; \quad \forall t = 1, \dots, T \quad (2.7)$$

- Tiempos mínimos de parada/arranque:

Se componen fundamentalmente de dos restricciones operativas:

- Tiempo mínimo en servicio: tiempo mínimo que la unidad deber permanecer en servicio sincronizada a la red.
- Tiempo mínimo de detención: tiempo mínimo que la unidad debe estar detenida inmediatamente después de salir de servicio.

$$U_{i,t} \begin{cases} 1 & \text{si } t_{i,t}^{\text{on}} \leq \hat{t}_i^{\text{on}} \\ 0 & \text{si } t_{i,t}^{\text{off}} \leq \hat{t}_i^{\text{off}} \end{cases} \quad (2.8)$$

Donde las variables de decisión son:

$U_{i,t}$ Variable binaria on/off de operación (1 si la unidad está en operación; 0 si está detenida) del generador “i” en el tiempo “t”.

$P_{i,t}$ Potencia activa neta de generación del generador “i” en el tiempo “t”.

$S_{i,t}^{start}$ Variable binaria on/off de encendido (1 si la unidad “i” tiene una partida en el tiempo “t”; 0 en caso contrario).

$S_{i,t}^{stop}$ Variable binaria on/off de encendido (1 si la unidad “i” se detiene en el tiempo “t”; 0 en caso contrario).

Con los siguientes parámetros técnicos (datos de entrada):

$P_{i,t}^{min}$ Potencia mínima de operación del generador “i” en el tiempo “t”.

$P_{i,t}^{max}$ Potencia máxima de operación del generador “i” en el tiempo “t”.

$\hat{S}_{i,t}$ Costo de partida del generador “i” en el tiempo “t”.

\hat{S}_i^{hot} Costo de partida desde estado “caliente” del generador “i” en el tiempo “t”.

\hat{S}_i^{warm} Costo de partida desde estado “tibio” del generador “i” en el tiempo “t”.

\hat{S}_i^{cold} Costo de partida desde estado “frio” del generador “i” en el tiempo “t”.

$\hat{P}_{c,t}^{carga}$ Demanda de potencia activa de consumo “c” en el tiempo “t”.

R^t Requerimiento de reserva en giro en el tiempo “t”.

\hat{G}_i^{Up} Tasa de toma de carga máxima del generador “i” en [MW/min]

\hat{t}_i^{on} Tiempo mínimo de operación de la unidad “i”.

\hat{t}_i^{off} Tiempo mínimo de detención de la unidad “i”.

$FC(P_{i,t})$ Función de costos. Típicamente como un polinomio lineal (por tramos), cuadrático o cúbico en función de la potencia activa de generación $P_{i,t}$.

I Número total de unidades.

T Número total de periodos en el horizonte de evaluación.

Adicional a las restricciones del modelo referenciado, existe una variedad de restricciones no mencionadas enmarcadas dentro del pre-despacho. La “generalización” de este problema puede encontrarse en [1].

Para los propósitos del presente trabajo, se dividirán las restricciones presentadas anteriormente en dos clases: individual y de acoplamiento. Se define como **restricción de acoplamiento** aquella que involucre en la misma ecuación, bien sea de igual o de desigualdad, variables de estado pertenecientes a más de una unidad generadora. Por ejemplo, en el planteamiento de pre-despacho anterior, (2.2) y (2.3) son las restricciones de acoplamiento.

Se define como **restricción individual** aquella que puede involucrar variables pertenecientes a distintos períodos de tiempo pero correspondientes a una misma unidad generadora. Por ejemplo, (2.4)-(2.7) son restricciones individuales para cada una de las unidades. Nótese que algunas, como por ejemplo (2.5) y (2.6), involucran a más de una variable de estado perteneciente a distintos tiempos.

2.2 Métodos de Solución

El problema de optimización combinatorial entero-mixto presentado anteriormente, no puede ser resuelto por simple enumeración exhaustiva de todas las posibles combinaciones de despacho. Entonces, una metodología es necesaria para obtener soluciones factibles y, afortunadamente, una prueba de optimalidad. Existe una diversidad amplia de métodos propuestos en la bibliografía, tanto heurísticos como matemáticos, de los cuales se escogieron las metodologías MIP y LR por ser reconocidas como las más efectivas para resolver el UC y coexistir en la actualidad en motores de UC comerciales (detalles se encuentran en la sección 1.2 “Motivación”). Por lo tanto, la presente sección muestra una extensa revisión de los métodos de solución en la bibliografía especializada y, a continuación, se explican los principales fundamentos teóricos de los 2 métodos seleccionados.

2.2.1 Revisión bibliográfica métodos de solución

La literatura presenta extensas revisiones bibliográficas de las metodologías para resolver el problema del pre-despacho. La primera resume las metodologías clásicas disponibles hasta el año de su publicación (1994) [18]. La segunda es una exhaustiva compilación de los puntos principales en 151 artículos internacionales publicados en los últimos 35 años [2]. En estas dos referencias se pueden encontrar citas para casi cualquier metodología de solución del problema de UC.

Las listas de mérito, programación dinámica y relajación lagrangeana son, de acuerdo con la clásica referencia [19], los métodos “más revisados” al año de su publicación. En las revisiones se presentan explicaciones sencillas de los principales aspectos que caracterizan estas tres técnicas acompañadas con ejemplos numéricos. Todas ellas tienen versiones mejoradas que prometen menores tiempos de ejecución, mejor calidad de las soluciones, mejor convergencia o combinaciones de varios de estos atributos. Una solución rápida al UC usando listas de prioridades mejoradas o de mérito mejoradas se puede encontrar en [20]. En la referencia [18] se listan quince formas distintas de realizar la programación dinámica de las unidades. Son básicamente quince propuestas distintas para limitar la búsqueda en el enorme espacio de estado producido por la “explosión” combinatorial. La relajación lagrangeana ha sido objeto de intensas investigaciones en los últimos quince años. Ciertamente esta metodología requiere de una delicada afinación de parámetros, evidencia una convergencia irregular (especialmente cuando existen pocas unidades) y presenta dificultad para encontrar soluciones factibles (en especial cuando coexisten unidades con costos muy similares). Sin embargo, los autores coinciden en que es una técnica rápida, de calidad aceptable y aplicable a grandes sistemas con muchas restricciones. La referencia [8] explica el método más usado para actualizar los multiplicadores de Lagrange en cada iteración. En [21] los autores probaron planos cortantes (solos) así como una combinación de punto interior con planos cortantes para realizar la actualización de estos multiplicadores. A diferencia de [8], no necesitan afinar parámetros pero muestran una convergencia irregular, incrementándose adicionalmente los tiempos de ejecución. En [22] se propone una alternativa para este efecto haciendo

uso de un algoritmo genético. En este caso se presentaron similares inconvenientes a los reportados en [8].

La relajación lagrangeana aumentada [23], [24] consiste en agregar un término cuadrático a la función objetivo para manejar de forma más eficiente la brecha entre las soluciones del dual y las del primal. Sin embargo, este término involucra más de una variable de decisión eliminando la posibilidad de separar el problema en asignación individual de unidades. Para restablecer la separabilidad, el término cuadrático es linealizado alrededor de la solución de la iteración anterior usando técnicas de descomposición y coordinación. Dos métodos distintos de descomposición, aunque aplicables al UC de acuerdo con [24], hacen que el problema continúe evidenciando convergencia irregular.

La “desasignación de unidades” [25], [26], en inglés “Unit Decommitment”, parece ser la solución matemática al problema de sobreasignación (soluciones factibles) normalmente causado por la relajación lagrangeana, pero no necesariamente la más justa [15]. Un método adaptativo basado en relajación lagrangeana para resolver el UC que básicamente resume varias de las mejoras antes mencionadas puede encontrarse en [27]. Los autores además propusieron un método heurístico para realizar la programación dinámica de cada unidad por separado de forma rápida. Los resultados mostrados parecen prometedores. Además de la diversidad de mejoras propuestas, la relajación lagrangeana es sin duda la técnica más estudiada a la hora de considerar el efecto de otras restricciones operativas en el UC, tales como restricciones de rampa [23], [28] (o de toma de carga), restricciones de transmisión [29], [30], de seguridad y voltaje [28], [30] e incluso, las restricciones del flujo de carga óptimo [31] [32].

Métodos primales, como las técnicas Branch & Bound y la programación lineal, son pioneros en la resolución de problemas de optimización como el UC. Empleando algoritmos y software (CPLEX) considerados como clásicos, de acuerdo con [33], los métodos duales como la relajación lagrangeana son más rápidos, pero con menor precisión que los primarios Branch & Bound y programación lineal. La programación lógica puede ser usada para obtener un eficiente chequeo de restricciones [34]. Esto,

combinado con técnicas Branch & Bound para la búsqueda en el espacio puede llevar a un eficiente método para resolver el UC llamado programación lógica de restricciones. Los resultados presentados usando el lenguaje CHIP para casos pequeños son ciertamente impresionantes y serán presentados y analizados posteriormente. Algunas propuestas clásicas de aplicación de programación entero-mixta, programación lineal y técnicas Branch & Bound al UC pueden ser encontradas en diversas referencias citadas en [18].

Las técnicas meta-heurísticas también tienen diversas aplicaciones en la solución del UC. El complejo problema de optimización que representa el UC parece ser sumamente atractivo para modelos matemáticos no exactos. Sin embargo, todos ellos parecen aumentar considerablemente los esfuerzos computacionales. Los algoritmos genéticos [35]- [36], la programación evolutiva [37] y lógica difusa [38], [39], representan algunas de las rutinas exploradas. La calidad de la solución ha sido reportada como superior a la de métodos clásicos, lo que se ha manifestado en mayores tiempos de proceso.

La referencia [40] utiliza un algoritmo genético que emplea una clasificación de unidades de acuerdo con las características de costos de operación y partida, mostrando que es factible reducir los tiempos computacionales. Sin embargo, los resultados siguen siendo muy inferiores a la relajación lagrangeana. Las redes neuronales artificiales [9], [41] y otros sistemas expertos han sido también investigados. La principal queja de los autores ha sido la falta de resultados con óptimos globales para sistemas de gran escala. Por ende, las aplicaciones reportadas de redes neuronales sólo pueden ser entrenadas adecuadamente para resolver problemas de dimensiones reducidas.

Las simulaciones de temperatura [42], [43] basadas en el mínimo de energía de un metal enfriado lentamente, en inglés "Simulated Annealing", es otra de las formas planteadas en la bibliografía de resolver el problema del UC. Prometen mayores probabilidades de obtener soluciones cercanas al óptimo. Sin embargo, en aplicaciones prácticas, los tiempos requeridos para obtener esta calidad de soluciones son elevados. La combinación de algoritmos genéticos, simulaciones de temperatura, y búsquedas tabú [44] ha reportado mejores resultados, al menos en problemas de pocas unidades.

Una variante de la búsqueda loca ("*local search*") denominada "*External Optimization*" es empleada en [45] para encontrar soluciones factibles que, de acuerdo a la misma referencia, resultan mejores (menor costo total de operación) para los mismos casos en el rango 10-100 unidades resueltos por otras metaheurísticas en [35] y [37]. Sin embargo, no se presentan resultados comparativos de tiempos computacionales.

La búsqueda tabú ha sido también combinada con la programación evolutiva para solucionar el UC [46], pero incluso un pequeño ejemplo numérico de 34 unidades excedió la hora de cálculo. En [2] se puede encontrar otra serie de propuestas relacionadas con metaheurísticas e inteligencia artificial. De acuerdo con [18] y [2] existen otras metodologías como el Análisis de Riesgos, Análisis de Decisión, Sistemas Expertos, Programación Separable, Programación por Flujo de Redes, Algoritmos de búsqueda por colonias de hormigas y otros modelos híbridos que combinan uno o varios de los antes mencionados.

Esta variedad de métodos y publicaciones vienen por lo general acompañados con ejemplos numéricos y tiempos de ejecución. Sin embargo, por haber sido ejecutadas en plataformas diferentes, en la literatura no se presentan comparaciones objetivas de los rendimientos, dificultándose el desarrollo y la selección de los mismos en aplicaciones reales. Por lo tanto, el presente trabajo presenta resultados de rendimientos con full detalle de las instancias resultas y especificaciones técnicas del hardware y algoritmos utilizados para resolver los problemas de UC. Esto facilitaría a distintos autores realizar comparaciones de performance respecto a las metodologías en el presente estudio (MIP y LR) en el futuro con mayor precisión.

2.2.2 Relajación Lagrangeana

Es uno de los más conocidos métodos para resolver problemas de optimización de gran escala [47]. Se suele aplicar a problemas que por sus condiciones permite clasificar las restricciones en dos tipos: sencillas y complejas. La idea principal consiste en resolver problemas de optimización sujeto a las restricciones sencillas (definidas en el ámbito específico de UC como "individuales"), "olvidando" o como

se suele llamar “relajando”, las restricciones más complejas o complicadas (definidas anteriormente en el ámbito del UC como de “acoplamiento”).

Llamemos primal al siguiente problema de optimización:

$$\begin{aligned} \min: & f(x) \\ \text{s. a.:} & g(x) \leq 0 \end{aligned} \tag{2.9}$$

Donde: $x = [y \quad z]$

y son variables enteras y z pertenece a \mathfrak{R}^n .

Supongamos que existen condiciones que permiten dividir (o clasificar) las restricciones en dos grupos.

Denominemos $g_1(x)$ las restricciones de igualdad o desigualdad que son consideradas complejas y

denominemos $g_2(x)$ al resto de restricciones, las cuales son consideradas sencillas tal que:

$g(x) := \{g_1(x), g_2(x)\}$. Re-escribiendo el problema primal con estas definiciones se tiene:

$$\begin{aligned} \min: & f(x) \\ \text{s. a.:} & g_1(x) \leq 0 \\ & g_2(x) \leq 0 \end{aligned} \tag{2.10}$$

Al incluir a la funcional $f(x)$ las restricciones $g_1(x) \leq 0$ con factores de ponderación no negativos (multiplicadores de Lagrange), se obtiene:

$$\begin{aligned} \min: & f(x) + \hat{\lambda} \cdot (g_1(x)) \\ \text{s. a.:} & g_2(x) \leq 0 \end{aligned} \tag{2.11}$$

donde:

$\hat{\lambda}$: Multiplicadores de Lagrange asociados a las restricciones “complejas” o “restricciones de acoplamiento”, como fuese definido en la sección 2.1. Se dice que (2.11) es la relajación Lagrangeana del problema (2.9).

El uso de la relajación lagrangeana es efectivo cuando las características del problema hacen que resolver (2.11) sea mucho más sencillo. Por ejemplo, si las restricciones $g_2(x) \leq 0$ consisten de bloques no-conexos, es decir, la “forma” de la matriz de coeficientes es agrupable en bloques, entonces el problema puede además ser descompuesto en subproblemas de dimensiones reducidas. Es exactamente éste el caso del Predespacho.

Para cualquier $\lambda \in \mathfrak{R}_+^m$ se cumple la siguiente desigualdad [47]:

$$\psi(\lambda) \leq \varphi \quad (2.12)$$

Donde:

$\psi(\lambda)$: Es la solución de (2.11) para un valor $\lambda \in \mathfrak{R}_+^m$

φ : Es la solución óptima de (2.10).

Esto quiere decir que resolviendo el problema (2.11) se obtiene una cota inferior al problema original. Quizás la siguiente sea la pregunta más pertinente: ¿Cuál vector de multiplicadores de Lagrange permiten realizar la mejor estimación de la cota inferior? La respuesta es aquel $\lambda \in \mathfrak{R}_+^m$ que entrega el mayor valor posible de $\psi(\lambda)$.

El siguiente, es el llamado problema dual de Lagrange:

$$\begin{aligned} z = \max \psi(\lambda) \\ \text{s.a.: } \lambda \geq 0 \end{aligned} \quad (2.13)$$

La función $\psi(\lambda)$ es por definición convexa (lineal a trozos cuando se trata de programación lineal) [47]. Por lo tanto, puede resolverse mediante cualquiera de los abundantes métodos de maximización de problemas convexos no-diferenciables que existen en la bibliografía. La diferencia $\varphi - z$ es conocida en la literatura como la brecha dual, en adelante (DG) o simplemente “gap” “duality gap” por su significado en inglés. El DG aparece por la presencia de variables enteras y por la admisibilidad del dominio no-convexo de (2.9).

A veces la solución óptima del dual (2.13) puede obtenerse sin maximizar la función $\psi(\lambda)$. Se dice que el valor óptimo se ha obtenido si el valor de la funcional $\psi(\lambda)$ no varía ante la ausencia de las condiciones de integralidad de las variables enteras.

Entonces, si estas condiciones se cumplen, es suficiente con resolver el siguiente problema de programación lineal:

$$\begin{aligned} \zeta = \min: & f(x) \\ \text{s. a.: } & g_1(x) \leq 0 \end{aligned} \tag{2.14}$$

$$g_2(x) \leq 0 \text{ tal que: } g(x) := \{g_1(x), g_2(x)\}; x \in \mathfrak{R}^n$$

Nótese que todas las variables x son continuas (relajación de las condiciones de integralidad), por lo tanto, se cumple que: $\zeta = z$.

Tradicionalmente se ha utilizado al gap dual (DG) como una medida de convergencia. Según [19], mientras mayor sea la dimensión del problema, menor será la brecha relativa, denominada en adelante (RDG) por sus siglas en inglés “Relative Duality Gap”:

$$RDG = \frac{DG}{\varphi} \tag{2.15}$$

Algunos métodos numéricos iterativos se detienen cuando encuentran un $RDG \leq \varepsilon$, para un valor pre-establecido de ε , señalando el resultado actual como la solución al problema (2.9). No existen entonces garantías de que la solución sea factible, haciendo que la relajación lagrangeana sea dependiente de un post-proceso, típicamente heurístico, para encontrar soluciones factibles en la cercanía de la solución encontrada. Existe una abundante bibliografía con tales métodos adaptados al problema de asignación de unidades [26].

2.2.3 Programación entero-mixta

Existe una rica bibliografía especializada en programación lineal-entera. La referencia [48] es un completo manual de referencia que cubre desde los fundamentos matemáticos básicos sobre la cual se sustenta la programación lineal, la teoría de grafos y la teoría poliedral, hasta los detalles más específicos

de los principales algoritmos para la programación lineal-entera-mixta, incluso detallando adaptaciones a las principales aplicaciones a problemas de programación entera conocidos (eg. vendedor viajero, problema de la mochila, etc). Los detalles de los fundamentos teóricos como de las implementaciones prácticas del motor de optimización MIP empleado en el contexto de la presente investigación [49] están fuera del alcance los objetivos del presente trabajo, por lo que se sugiere al lector revisar las referencias [48] y [49] para más detalles.

El problema de UC en las ecuaciones (2.1) – (2.8) corresponde al tipo lineal entero-mixto, cuyas variables de estado de las unidades requerirán cumplir con la condición de integralidad (ie. variables binarias). Aunque conceptualmente existen otras modalidades, todas las rutinas comerciales se basan en el método branch-and-bound mediante relajaciones lineales.

Supongamos el siguiente problema generalizado de optimización entera:

$$\begin{aligned} Z_{ip} &= \max\{cx: x \in S\} \\ S &= \{x \in Z_+^n: Ax \leq b\} \end{aligned} \tag{2.16}$$

Un algoritmo de enumeración implícita, es decir un algoritmo branch-and-bound para (2.16), según [48], posee los siguientes pasos:

- **Paso 1- Inicialización:** Se plantea el problema como una colección, denominado L, el cual es una colección de problemas enteros IP_i , cada uno de los cuales consiste en la forma $z_{ip}^i = \max\{cx: x \in S^i\}$ donde $S^i \subseteq S$.
- **Paso 2- Prueba de finalización:** Si $L = \emptyset$, entonces la solución a z_{ip} es óptima.
- **Paso 3- Selección de problema y relajación:** Seleccionar y extraer de L un problema IP_i . Resolver su relajación RP^i . Se denominará z_R^i al valor óptimo de la relajación LP y x_R^i a la solución óptima si existe.
- **Paso 4- Podado:**
 - Si $z_R^i \leq z_{JP}$ ir al paso 2.

- Si $x_R^i \notin S^i$ ir al paso 5.
- Si $x_R^i \in S^i$ y $cx_R^i > \underline{z}_{IP}$, asignar $\underline{z}_{IP} = cx_R^i$. Eliminar de L todos los problemas con cota superior menor a \underline{z}_{IP} . Si $cx_R^i = z_R^i$ ir al paso 2.
- **Paso 5- División:** Plantear y agregar divisiones de S^i a L, donde la cota inferior de cada uno de las divisiones será z_R^i . Ir al paso 2.

Las principales propiedades del método branch-and-bound son las divisiones, las estrategias de desarrollo del árbol, reducción del árbol, mínimo árbol [48].

2.3 Resumen

El presente capítulo definió las bases tanto teóricas como metodológicas del presente estudio. La definición (y alcance) del UC fue presentado con su formulación matemática entera-mixta. Tanto la función objetivo como sus restricciones básicas fueron expuestas. Una vez definido el problema en estudio, se presentaron los posibles métodos de solución (revisión bibliográfica), incluyendo detalles técnicos de los 2 métodos en estudio. Sin embargo, la relajación Lagrangeana no es una rutina genérica de optimización, sino requiere de un desarrollo específico para el UC. Dada la diversidad de variantes propuestas en la literatura, como se describió en la presente sección, el capítulo a continuación presenta todos los detalles del algoritmo desarrollado en el marco de la presente tesis.

Capítulo 3

Algoritmo de UC por relajación Lagrangeana

3.1 Introducción

Considerando la diversidad de propuestas en la literatura para la elaboración de un algoritmo de Relajación Lagrangeana para resolver el UC, el capítulo a continuación presenta los detalles técnicos del algoritmo desarrollado en el marco de la presente tesis.

El algoritmo de programación elaborado en el contexto del presente trabajo está basado en la relajación lagrangeana aumentada [28]. A continuación se describen las etapas fundamentales de dicho algoritmo. Algunos detalles de los métodos matemáticos subordinados que se requieren han sido ubicados en los anexos A, B y C del trabajo. Para la presente explicación además, se ha preferido dejar la variante que incluye el término de “aumentación” para el Anexo D. Los fundamentos básicos del método y la validez de la descomposición son presentados en la sección 2.2.2, sin embargo, se sugiere referirse a [47] para detalles de los fundamentos teóricos que soportan el método como tal, así como ciertos aspectos formales y demostrativos. Considerando además que el método branch-and-bound en las rutinas comerciales MIP es de conocida documentación y que, adicionalmente, escapa del presente trabajo una sintonización de parámetros del MIP que involucren un profundo conocimiento de las diversas etapas de solución, se sugiere al lector revisar detalles conceptuales del método en la referencia [49].

3.2 Generalidades del Método

La relajación lagrangeana es uno de los métodos de mayor aplicación al problema del pre-despacho, en especial a problemas de gran-escala [19]. Se basa en la descomposición del dual al “relajar” las restricciones de acoplamiento (ver sección 2.2.2). En la ilustración 3.1 se presenta un esquema de ejecución clásico de relajación lagrangeana basado en [19].

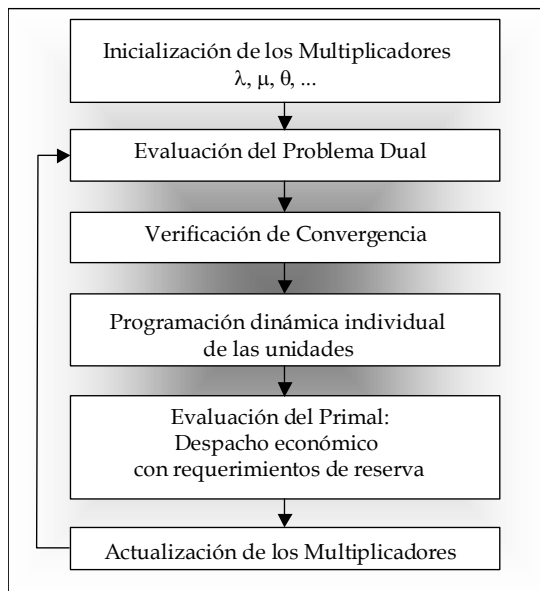


Ilustración 3.1. Esquema general de ejecución del algoritmo de relajación lagrangeana.

Cada una de las etapas del algoritmo será brevemente descrita en las secciones a continuación.

3.2.1 Inicialización o relajación de los multiplicadores de Lagrange

Al relajar las restricciones de balance de potencia y de requerimientos de reserva, el problema del pre-despacho básico se desacopla en problemas de optimización en el tiempo individuales para cada una de las unidades generadoras en consideración. Para ello, las variables duales asociadas a esas restricciones deberán ser constantes. Los valores que se sugieren en la bibliografía pueden ser muy variados y puede que el algoritmo converja a soluciones diferentes [15]. El siguiente método es una combinación de la sugerencia presentada en [27] y un aporte propio. Consiste fundamentalmente en resolver el problema del pre-despacho por listas de prioridades, sin considerar las restricciones que acoplan al problema en el tiempo. Para ello se realiza el siguiente procedimiento heurístico:

- **Restricciones de balance de Potencia:** Los multiplicadores de Lagrange o variables duales asociadas a la restricción (2.2) serán iguales al valor del costo marginal (o linealización) de la unidad “más cara” que cumpla con el balance de potencia, siendo asignadas de acuerdo al orden de mérito establecido en la lista, *sin verificar* las restricciones de tiempos mínimos de operación ni los costos de partida asociados.

- **Restricciones de requerimientos de reserva:** Las variables duales asociadas a (2.3) son inicializadas de acuerdo al siguiente criterio:

$$\mu_t = 0; \Rightarrow \sum_i^M p_i^{\max} - (\mathbf{R}^t + P_{\text{demanda}}^t) \geq 0 \quad (3.1)$$

$$\mu_t = \frac{(\beta_s - \lambda^t) \cdot p_{\min}}{p_{\max}}; \Rightarrow \sum_i^M p_i^{\max} - (\mathbf{R}^t + P_{\text{demanda}}^t) < 0 \quad (3.2)$$

Donde μ es la variable dual a relajar, β_s es el costo marginal de la unidad "S" necesaria para cumplir con la reserva en giro mínima exigida, es decir, aquella que cumpla: $\sum_i^M P_i \geq \mathbf{R}^t + P_{\text{demanda}}$ (con M = número de unidades necesarias para satisfacer la demanda).

- **Programación Dinámica de las Unidades:** Se realiza la programación dinámica individual de las unidades en el sentido del tiempo (en inglés forward DP). En esta etapa deben cumplirse las siguientes restricciones:

- Límites Operativos de las unidades.
- Tiempos mínimos de encendido/apagado.
- Límites de potencia máxima por: reserva en giro impuesta individualmente para cada unidad, garantías de entrega de reserva primaria, límite sistémico por estabilidad dinámica/transiente, consumos propios, etc.

Se realiza un proceso de optimización aplicando el método de programación dinámica, mediante el cual, se calculan los valores de las variables binarias (U_i^t) y de las potencias ($P_{i,t}$) que minimizan la función lagrangeana para multiplicadores $\hat{\lambda}_t$ y $\hat{\mu}_t$ constantes. Siguiendo con la notación en 2.2.2, esta etapa de programación dual corresponde con el cálculo de:

$$\min_x := L(x, \hat{\lambda}) = f(x) + \hat{\lambda} \cdot (g_1(x)) \quad (3.3)$$

$$\text{sa: } g_2(\mathbf{x}) \leq 0$$

El algoritmo de programación dinámica empleado, por su parte, se encuentra detallado en el Anexo B. La programación dinámica aplicada al pre-despacho como método único de solución, es decir, no como un subproceso dentro de la relajación lagrangeana, puede encontrarse explicada y ejemplificada en [19].

3.2.2 Evaluación de la Función Dual

El proceso iterativo aplicado a la programación dual exige que en cada iteración se deba verificar la diferencia entre las soluciones del dual y el primal. Entonces, la evaluación del dual consiste en sustituir en la función lagrangeana el valor constante de las variables duales junto con el valor del vector binario de decisión y la potencia obtenida por programación dinámica. Es decir, se evalúa el valor del llamado “lagrangeano” (L) en el resultado obtenido por la minimización en U_i^t y $P_{i,t}$ con programación dinámica y con las variables duales obtenidas de la maximización del dual.

$$L(U, P, \hat{\lambda}) = f(U, P) + \hat{\lambda} \cdot (g_1(U, P)) \quad (3.4)$$

Nótese que corresponden en forma estricta con la definición de la optimización dual definida en el Anexo A.

3.2.3 Despacho Económico sin Restricciones de Transmisión

Esta etapa es también conocida como *evaluación del primal*. Consiste en encontrar una solución factible al problema del pre-despacho completo. Existe una variedad de métodos disponibles en la bibliografía, las cuales pueden agruparse en dos grupos:

- Heurísticos y Meta-heurísticos: Consiste en encontrar una solución factible al problema haciendo uso de técnicas no-precisas basadas en heurísticas.
- Basados en el vector binario del dual: Consiste en sustituir las variables binarias (enteras), obtenidas de la programación dinámica individual de las unidades, en el problema original (2.1)-(2.7). Al asumir conocidas las variables enteras, el problema resultante se resuelve por programación lineal o

cuadrática. Es conocido en la literatura clásica como el despacho económico para cada una de las etapas.

La alternativa seleccionada a ser aplicada por la rutina de programación fue la opción basada en el vector binario del dual porque es la que resulta en los mejores tiempos de ejecución. Presenta como principal desventaja la dificultad para encontrar soluciones factibles en un número finito de iteraciones. Sin embargo, un debido control de los parámetros de sintonización puede resultar en la sucesiva evaluación de soluciones factibles.

3.2.4 Verificación de Convergencia

Se realiza haciendo una verificación de la “brecha dual relativa” (RDG). La RDG está definida por:

$$RDG = \frac{L^* - L(x, \hat{\lambda})}{L(x, \hat{\lambda})} \quad (3.5)$$

La mayoría de los métodos numéricos sugiere que el algoritmo debe detenerse cuando se encuentre un $RDG \leq \varepsilon$, siendo ε un valor pre-establecido. Existen alternativas referidas a las soluciones ε -óptimas obtenidas cuando se resuelve un problema de optimización que no cumplen algunas de las restricciones relajadas. En el Anexo A se analiza esta alternativa.

3.2.5 Actualización de los Multiplicadores

La actualización de los multiplicadores de Lagrange se corresponde con la maximización de la función lagrangeana respecto a las variables duales. Siguiendo la notación en 2.2.2, corresponde a la solución de la siguiente etapa:

$$\begin{aligned} z = \max: & \psi(\lambda) \\ \text{s.a: } & \lambda \geq 0 \end{aligned} \quad (3.6)$$

El problema de optimización lineal que plantea (3.6) suele ser resuelto por el método del subgradiente por la simplicidad de su planteamiento e implementación [19]. Sin embargo, es también reconocido por ser un método inestable y, en ocasiones, lento para resolver problemas de descomposición entero-mixtos [27]. Más aún, se le suele asociar con la dificultad que caracteriza a la relajación lagrangeana para

encontrar soluciones factibles. En todo caso, lo que es evidente es que es una etapa crítica del método y, como tal, merece especial cuidado.

Existen diversas maneras de plantear el subgradiente. Las más conocidas se muestran en el Anexo C. El mismo anexo muestra la validez de los métodos y las demostraciones de la convergencia al valor óptimo (o a una tolerancia pre-establecida). El método de actualización del algoritmo tiene dos variantes:

- **Prioridad → búsqueda de soluciones factibles:**

Emplea la variante Longitud del paso constante (ver Anexo C y las referencias citadas). De tal manera, los multiplicadores de Lagrange se actualizan según la siguiente relación:

$$\lambda^{k+1} = \lambda^k + \alpha_k \cdot \frac{\partial L(\lambda)}{\partial \lambda}$$

$$\text{con: } \alpha_k = \frac{h}{\left\| \frac{\partial L(\lambda)}{\partial \lambda} \right\|_{\lambda^k} \|_2} \quad (3.7)$$

El valor de la longitud del subgradiente h es escogida por el usuario o programador. Su valor suele determinarse experimentalmente [19], [27]. El error se encuentra acotado por:

$$Error \leq \frac{R^2 + G^2 h^2 k}{2hk} \quad (3.8)$$

Donde R es la diferencia entre el valor inicial y la solución óptima, G es la cota superior de la norma del subgradiente y k es el número de iteraciones. Entonces, para un número de iteraciones muy elevado el error $Error \rightarrow \frac{G^2 h^2}{2}$. Evidentemente, la búsqueda abandona la garantía del óptimo pero puede generar

con mayor probabilidad soluciones factibles para ciertos valores de h si se le combina con la siguiente

heurística sencilla de búsqueda de soluciones factibles:

- Si la solución no es factible, hacer:

$$\alpha_k = \frac{h \cdot 0,2}{\left\| \frac{\partial L(\lambda)}{\partial \lambda} \right\|_{\lambda^k} \|_2} \quad (3.9)$$

- Si es factible, usar el h por usuario o programador.

- **Prioridad → búsqueda del óptimo:**

Emplea la variante series sumables pero cuadrados no sumables (ver Anexo C y las referencias citadas en el mismo). La aplicación de esta regla demuestra que se puede obtener el óptimo cuando el número de iteraciones $k \rightarrow \infty$. Es el más inestable y puede que no genere ninguna solución factible para un número finito de iteraciones. Sin embargo, se mantiene la posibilidad de encontrar los valores óptimos de los multiplicadores.

$$\alpha_k = \frac{\left. \frac{\partial L(\lambda)}{\partial \lambda} \right|_{\lambda}}{\alpha_{sub} + \beta_{sub} * \left\| \frac{\partial L(\lambda)}{\partial \lambda} \right\|_2} \quad (3.10)$$

Donde $\alpha_{sub} \geq 0, \beta_{sub} > 0$, nuevamente seleccionados por el programador o usuario del programa por métodos experimentales.

3.3 Resumen

Considerando la necesidad de personalizar un algoritmo basado en LR al tipo de problema que se desee resolver, combinado con el requerimiento de sintonización de ciertos parámetros, características intrínsecas de LR, se presentaron en detalle los fundamentos matemáticos y características principales del método de solución LR, aplicados al problema del UC. La rutina de programación desarrollada sigue en forma estricta la definición de relajación, método de subgradiente, programación dinámica y criterio de convergencia definido en el presente capítulo. Dado que para efectos comparativos se empleó un optimizador comercial MIP tipo “black-box”, no fue requerida la elección de alternativas de desarrollo o sintonización de parámetros, por lo que se sugiere al lector referirse a la documentación en [49].

Capítulo 4

Comparaciones MIP vs. LR

4.1 Introducción

El presente capítulo entrega los resultados de un análisis extensivo de rendimiento entre las metodologías para resolver el UC. El capítulo comienza con un análisis de los resultados de rendimiento publicados en la bibliografía. Además de resumir por primera vez en un único documento los resultados de una diversidad de trabajos de investigación acerca del UC, se plantea y aplica una metodología para “estandarizar” los tiempos de ejecución, haciendo uso de factores de rendimiento tomados desde reconocidos benchmark computacionales, de tal forma de hacerlos comparables. Posteriormente, se presentan los resultados de todas las simulaciones realizadas en el contexto del presente trabajo. Se muestran los resultados, tanto de performance como de calidad de solución, de las simulaciones por ambos métodos (LR y MIP), comparados mediante 3 enfoques diferentes:

- 1) Comparación clásica haciendo uso de casos sintéticos: Se crearon y resolvieron casos sintéticos sencillos que permitieron sintonizar y validar la rutina de LR desarrollada.
- 2) Comparación de casos aleatorios. Se generaron y resolvieron, mediante ambos métodos, múltiples casos de UC creados mediante la combinación de las técnicas de creación de casos aleatorios con correlación explícita y el uso de librerías estándar. Esta sección incluye además todos los detalles del método de generación de instancias aleatorias y método de análisis.
- 3) Comparación aplicada a casos reales: Ejemplos de programación de la operación del Sistema eléctrico chileno SING.

4.2 Comparación bibliográfica

Plantear una comparación de rendimiento entre distintos computadores puede ser una tarea difícil. El proceso de obtener soluciones computacionales pasa por una variedad de “cuellos de botella” como pueden ser la memoria RAM, el bus del controlador, el bus de video y otros. Comparar la eficiencia de distintos algoritmos en computadores diferentes es una tarea aún más difícil. Entre otras, éstas son las principales limitantes: No siempre los autores publican las especificaciones del computador o del sistema operativo, o del lenguaje de programación en que fueron desarrollados los casos numéricos; no existe un caso base estandarizado IEEE para el UC que pueda ser utilizado con fines comparativos; no se puede evaluar la eficiencia del algoritmo en sí (métodos de inversión de matrices, eficiencia en la adquisición de datos, ciclos internos, etc.) Lamentablemente, tuvieron que ser descartados algunos resultados interesantes de la presente comparación debido a la ausencia de información del hardware evaluador.

En general, el objetivo de la comparación bibliográfica es encontrar rangos aceptables donde ubicar los tiempos de ejecución de las diversas metodologías, sobre una misma base. Es decir, la idea es emplear resultados públicos de “benchmarks” oficiales para poder “trasladar” artificialmente todos los resultados a un mismo computador. Entre las principales organizaciones para la medición del performance se encuentran: SPEC - Standard Performance Evaluation Corporation, BAPCo - Business Applications Performance Council, EEMBC - EDN Embedded Microprocessor Benchmark Consortium, I/Ozone - I/O File performance Test benchmark y TPC: Transaction Processing Performance Council.

La información más completa acerca de los diversos rendimientos se encuentra en la página oficial de SPEC [50] y serán usados posteriormente en el presente trabajo. La referencia [51] por su parte, muestra una excelente comparación de los rendimientos de una variedad amplia de computadores resolviendo problemas de ecuaciones lineales. Los tres benchmarks empleados en [51] fueron: Linpack® [52], TPP (“Towards Pick Performance”) y TP “Theoretical Pick Performance”. La comparación se fundamenta en la medición en MFlops (Millones de operaciones completas de punto flotante) al resolver sistemas lineales de n ecuaciones. Los detalles de las mediciones y su validez se encuentran en [52]. Por su parte,

Dhrystone [53] y Whetstone [54] son dos conocidos benchmarks sintéticos empleados desde 1988 y 1976 respectivamente para la medición y comparación de los rendimientos de diversos computadores. Dhrystone fue diseñado para medir en MIPS (Millones de operaciones con enteros por segundo) operaciones aritméticas típicas de varias aplicaciones. Más antiguo inclusive, Whetstone sigue siendo utilizado en la actualidad para medir rendimientos de FPU y Coprocesadores, especialmente en el área de ingeniería. Mide el tiempo que tarda en realizarse una serie de operaciones de punto flotante (convertidos a MFlops). Los resultados de ambos “benchmarks” usando el software comercial ©Syssoftware Sandra complementarían la data requerida para la comparación de rendimientos empleada en este trabajo.

4.2.1 Factores de Ponderación

Los tiempos serán ponderados al multiplicarlos por un factor escalar comprendido entre cero y uno. Este factor será determinado como el promedio del puntaje obtenido por el computador en cada uno de los siete “benchmarks” referidos, divididos entre el máximo puntaje. De esta forma, trasladamos al computador más veloz, de acuerdo con los “benchmarks”, los tiempos obtenidos en otros computadores. El error inicial será determinado por la desviación estándar del factor según los distintos “benchmarks”. Los resultados son presentados a continuación.

Nombre	SPEC®				rank	
	int95	fp95	int2000	fp2000	int	fp
486 Dx2 /66	-	-	-	-	-	-
HP 9000/720	1.57	2.02	-	-	0.0280	0.0294
HP 9000/735	4.04	4.55	-	-	0.0720	0.0662
Sun Ultra 2 2200	6.85	12.9	-	-	0.1220	0.1877
P- 200 Mhz	6.40	4.68	-	-	0.1139	0.0680
HP C160	10.4	16.3	-	-	0.1853	0.2371
P4 - 1.5Ghz	-	-	562	615	0.9558	0.9565
P4 - 1.6Ghz	-	-	588	643	1	1
Dell Precision Workstation 420 (733Mhz)	35.7	31	374	290		

Tabla 4.1. Ranking usando los Benchmarks SPECint® y SPECfp® [50].

La primera columna en la tabla 4.1 muestra los nombres comerciales de los computadores empleados por los autores en las distintas referencias para obtener los resultados numéricos en sus publicaciones originales. Las columnas segunda a la quinta de la misma tabla muestran la puntuación asignada por los “benchmarks” SPECint®95, SPECfp®95, SPECint®2000 y SPECfp®2000 respectivamente.

De acuerdo con la política de “uso justo” de SPEC® [35], no existe ninguna forma lineal de comparar los resultados de “benchmark” distintos, lo cual representa un grave problema porque no hay datos disponibles sino los mostrados en la tabla 1. Para poder establecer el ranking lineal deseado, se empleó el computador “Dell Precision Workstation 420 (733Mhz)” como referencia. Los números presentados en las dos últimas columnas fueron calculados de acuerdo a la siguiente relación:

$$\text{int}(i) = \text{int } 95(i) \frac{\text{int } 2000(9)}{\text{int } 95(9) \times \max\{\text{int } 2000(i)\}} \quad (4.1)$$

Donde:

i: representa la columna “i” en la tabla 4.1.

(9): representa los datos del computador de referencia.

La misma relación (4.1) fue empleada para calcular los valores de la columna 7 (cambiando int por fp).

Los resultados extraídos de [51] son los mostrados en la tabla a continuación:

Nombre	LinP N=100	TPP N=1000	pick	rank		
				N=100	N=1000	pick
486 Dx2 /66	2.4	-	-	0.00356	-	-
HP 9000/720	18	36	50	0.02667	0.02584	0.01471
HP 9000/735	41	120	198	0.08192	0.08615	0.05824
Sun Ultra 2 2200	114	117	500	0.16889	0.08399	0.14706
P- 200 Mhz	38	-	200	0.07592	-	0.05882
HP C160	140	421	640	0.20741	0.30223	0.18824
P4 - 1.5Ghz	500.5	1133	2250	0.74148	0.81335	0.66176
P4 - 1.6Ghz	363	1393	3400	0.53778	1	1

Tabla 4.2. Ranking usando el Ecuaciones Lineales [51].

Al igual que en la tabla 4.1, la primera columna de la tabla 4.2 muestra los nombres comerciales de los computadores a evaluar. En los casos en que sólo se disponía de la información del procesador, se tomó el peor rendimiento (mín) y el mejor (máx) de sistemas que emplean procesadores de la misma

característica y velocidad. Los márgenes de error tomarán en cuenta estas diferencias. Las columnas 2, 3 y 4 muestran los MFlops promedio registrados para los tres “benchmarks” antes mencionados. El ranking lineal en este caso fue calculado sencillamente como el cociente entre la puntuación del computador y la máxima entre los computadores para la misma operación. Los resultados pueden verse en las columnas 5, 6 y 7 de la tabla 4.2.

Finalmente, los dos benchmark sintéticos Dhrystone y Whetstone, sirven como una tercera referencia para ampliar los márgenes de error y tratar así de asegurar la certeza de los resultados y la validez de las comparaciones. Los siguientes resultados fueron obtenidos mediante el uso del programa ©SYSSoftware Sandra 2004.2.9.104:

Nombre	Dhrystone ALU (MIPS)	Whetstone (MFLOPS)	rank MIPS	rank MFLOPS
486 Dx2 /66	83	68	0.020	0.057
P- 200 Mhz	450	268	0.109	0.226
P4 - 1.5Ghz (min)	3768	1019	0.914	0.861
P4 - 1.6Ghz	4121	1184	1	1

Tabla 4.3. Ranking usando Benchmark Sintéticos [53], [54].

Finalmente, de las tablas 4.1 – 4.3 se puede obtener la siguiente clasificación de los rendimientos promedio y la desviación estándar para los diversos computadores empleados en la bibliografía referida al UC. Los detalles del cálculo se encuentran en [3]. La tabla 4.4 a continuación muestra estos resultados:

Nombre	Proc #	Clock (Mhz)	Relac Vel.	AVG (pu)	Error	
					(pu)	%
486 Dx2 /66	1	66	0.04125	0.027	0.028	102.0
HP 9000/720	1	50	0.03125	0.021	0.009	44.8
HP 9000/735	1	99	0.06188	0.069	0.004	5.9
P-200Mhz	1	200	0.125	0.10182	0.05799	57.0
Sun Ultra 2 2200	2	200	0.125	0.152	0.047	30.6
HP C160	1	160	0.1	0.224	0.048	21.6
P4 - 1.5Ghz	1	1500	0.9375	0.849	0.130	15.4
P4 - 1.6Ghz	1	1600	1	0.934	0.175	18.7

Tabla 4.4. Factores de ponderación.

La columna 2 muestra el número de procesadores del PC o Workstation. La columna 3 la velocidad del reloj de cada procesador. “Relac Vel.” es el cociente entre la velocidad del procesador actual y la del más

rápido (en este caso el reloj más veloz es de 1600Mhz). La columna 4 es el factor de ponderación lineal promedio junto a su desviación estándar (en por unidad y en porcentaje). Como puede observarse, los márgenes de error correspondientes a computadores cuyo único dato disponible era el procesador son muy amplios.

Este ranking será empleado como único factor de ponderación de los tiempos obtenidos en los ejemplos numéricos. Quiere decir que serán totalmente ignorados los siguientes aspectos también influyentes en los tiempos de ejecución: eficiencia del algoritmo, lenguaje de programación, expansiones y/o mejoras de hardware realizadas postventa.

4.2.2 Tiempos de Ejecución

En la Tabla 4.5 y Tabla 4.6, presentadas a continuación, se muestran los tiempos de ejecución que fueron requeridos por cada metodología para solucionar el UC de distintas dimensiones, multiplicados por los factores de ponderación mostrados en la tabla 4.4 del presente trabajo.

Unidades	ALR-APP	ALR-BCD	EPL[4]	DP [20]	LR [20]	CLP [20]	LR [6]
	Sun Ultra 2 2200 [9]	P4 - 1.6Ghz	486 Dx2 /66			P-200Mhz	
10	0.412	0.515	0.672	0.162	0.108	0.081	
20	1.037	1.770	2.774	0.189	0.379	0.135	
30	1.463	1.210		0.460	0.649	0.189	1.094
40	2.744	2.487	11.114	0.730	0.433	0.270	2.089
50	4.466	4.369			1.217	0.514	
60	5.793	8.156	21.481		1.298	1.163	
70	7.469	15.236			1.541	1.704	3.979
80	10.152	22.025	41.468		1.271	2.218	
90	13.917	11.181					
100	22.759	16.693	60.241				5.669

Tabla 4.5. Tiempos de Ejecución Ponderado (en segundos) Metodologías “Rápidas”.

Unidades	LRGA [7]	BCGA [21]	ICGA [23]	EP [24]	ELR [13]	DPLR [13]
	486 Dx2 /66	HP 9000/720	P4 - 1.6Ghz	HP C160	P4 - 1.5Ghz	
10	14.0	4.6	6.9	22.4	3.4	90.5
20	31.0	15.2	20.9	76.2	13.4	250.6
40	58.5	55.8	54.5	263.5	43.6	1005.8
60	65.3	120.9	109.6	507.9	94.7	2681.4
80	91.5	207.7	164.4	803.0	175.2	7080.3
100	109.4	346.3	226.5	1371.3	289.2	10424.7

Tabla 4.6. Tiempos de Ejecución Ponderados (en segundos) Metodologías “Lentas”.

Las figuras en Ilustración 4.1 e Ilustración 4.2, presentadas a continuación, muestran gráficamente los resultados presentados en las tablas anteriores. Márgenes inferiores y superiores representan el error según lo expuesto en la Tabla 4.4.

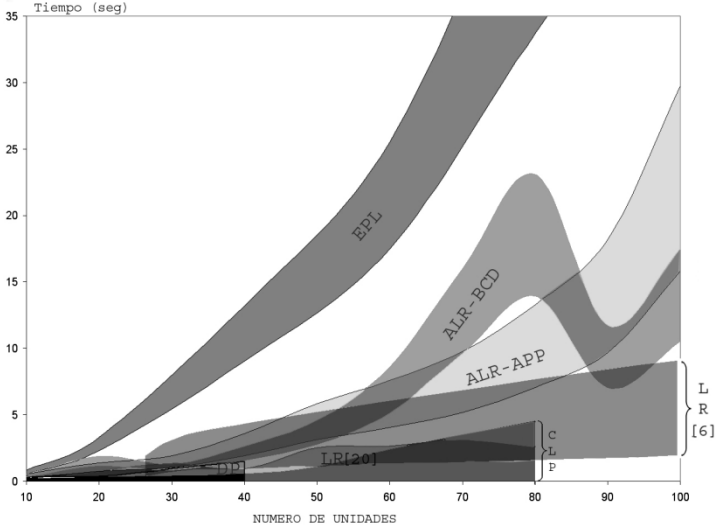


Ilustración 4.1. Márgenes de Ejecución. Metodologías Rápidas.

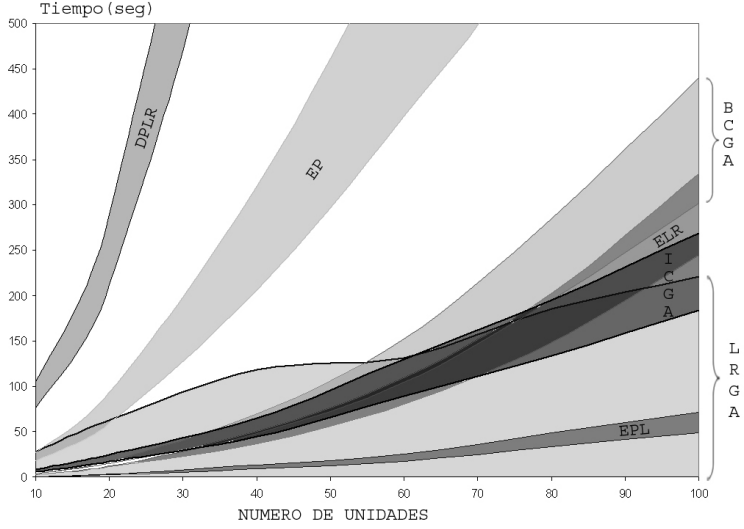


Ilustración 4.2. Márgenes de Ejecución. Metodologías Lentas.

Nótese la diferencia de escala de los ejes de las coordenadas (eje “y”) en cada gráfico. Haciendo una simple comparación de la Ilustración 4.1, comparada a la Ilustración 4.2, es posible observar que los tiempos de ejecución “normalizados”, incluyendo los amplios márgenes de error, de los métodos matemáticos como la Relajación Lagrangeana, Branch-and-Bound o Programación dinámica, resultan en

el orden de 0-30 segundos, mientras que la mayoría de las meta-heurísticas se encuentran en el rango 0-500 segundos.

Esta comparación estandarizada de rendimientos publicados en la bibliografía del UC es realizada por primera vez en el presente trabajo y reafirma por qué LR y MIP (métodos matemáticos) son los de mayores aplicaciones prácticas.

4.2.3 Calidad de las Soluciones

Medir la calidad de las soluciones es también una tarea difícil y los márgenes de error asociados son también amplios. Las dificultades principales son las siguientes: no todos los resultados se refieren al mismo ejemplo numérico; no siempre se dispone del criterio de convergencia. Sin embargo, existe un número importante de publicaciones citadas a continuación en las que se dispone del ahorro en (%) con respecto a otra metodología de referencia. La muestran algunos resultados del ahorro empleando otra metodología distinta a la relajación lagrangeana.

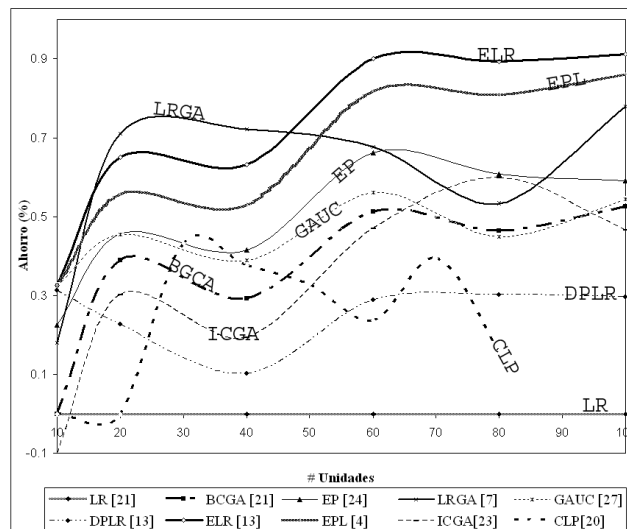


Ilustración 4.3. Ahorro relativo en (%) de las diversas metodologías usando la Relajación Lagrangeana LR como referencia.

Con la excepción de CLP y DP tomados ambos de [34], todos los ahorros presentados en esta sección [20], [22], [27], [38], [40] pertenecen al mismo ejemplo numérico. De manera que existe un margen de

error para DP y CLP en la Ilustración 4.3, pero para el resto la comparación es directa. Todos los datos para resolver este ejemplo común se encuentran en [55].

4.2.4 Tiempo vs. Optimalidad

Los datos requeridos para realizar el gráfico de la Ilustración 4.4 provienen de los resultados presentados en las secciones 4.2.2 y 4.2.3, extrayendo de ellos los tiempos ponderados y el ahorro respecto a LR, para una dimensión del problema equivalente a 38-40 unidades. Estas dimensiones son aproximadas a las unidades candidatas para el despacho del Sistema Interconectado del Norte Grande [56] (excluyendo las combinaciones producidas por los modos de operación de los ciclo combinados y combustibles alternativos), por lo que la presente comparación, mostrada en resumen en la Ilustración 4.4 *es la primera comparación bibliográfica “normalizada” realizada para el SING.*

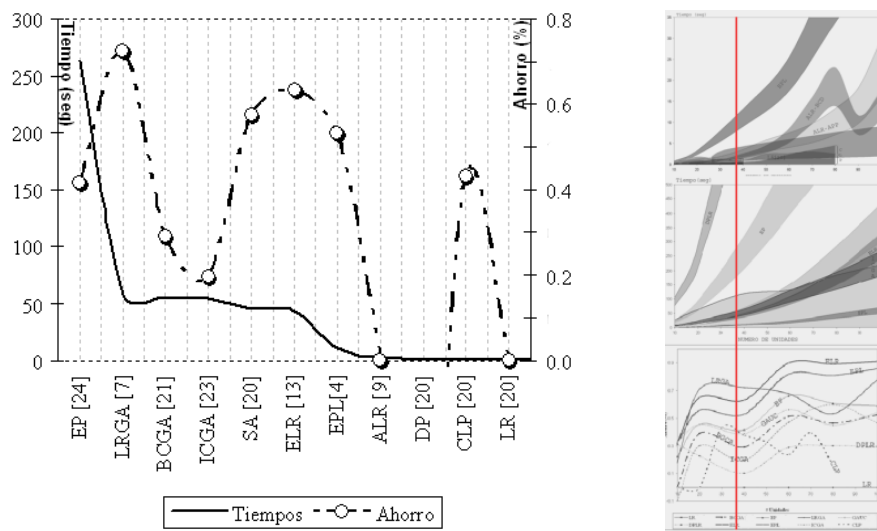


Ilustración 4.4. Tiempo y Ahorro de las Distintas Metodologías para un problema de 38 unidades.

Lamentablemente no se disponían de especificaciones ni del computador, ni del ahorro respecto a alguna otra publicación en metodologías que suponen un ahorro importante como [43], [44], [46]. La información “Power PC” en [43], [44] resultaba muy escasa para darle un valor en el ranking. Los resultados de SA en [42] fueron obtenidos en un computador IBM de 8Mhz. La comparación está fuera de orden para este caso.

4.2.5 Análisis de Resultados

A pesar de la amplitud de los márgenes de error utilizados, los gráficos que se construyeron en la sección 4.2.2 muestran el aumento los tiempos de ejecución para los métodos meta-heurísticos y las mejoras a la relajación lagrangeana. Nótese que, inclusive, tuvieron que dividirse en dos gráficos apartes el set de resultados, de tal forma que el ajuste de escala permitiera visualizar todos los resultados. Evidentemente, son varios órdenes de magnitud mayor el performance de los algoritmos en Ilustración 4.2 respecto a los algoritmos en la Ilustración 4.1. De igual forma, se observa la dificultad generalizada de los métodos para mantener la promesa de crecimiento de los tiempos de ejecución en forma lineal respecto a las dimensiones del problema.

En la sección 4.2.3 se mostró también que para estos ejemplos pequeños ninguna de las metodologías pudo superar el 1% de ahorro. Lamentablemente, la falta de información no permitió la inclusión de [43] ni [44] los cuales prometen ahorros entre 0.79%-2.1% y 1.05-2.15%, respectivamente. En esta revisión la relajación lagrangeana probó ser el método más eficiente para resolver el UC. Mejorar tan solo un 0.7%, la solución que ésta entrega puede costar, de acuerdo al resultado para las dimensiones 38 unidades, hasta 3000% del tiempo de ejecución si se intenta con otra metodología. Por supuesto que ese 0.7% puede representar varios millones de dólares en ahorro para las empresas del sector.

4.3 Comparaciones Experimentales

4.3.1 Ejemplo sintético

Para las simulaciones presentadas a continuación se elaboró un ejemplo sintético, construido a partir del modelo de confiabilidad RTS-96 del IEEE, abreviaturas del inglés “Reliability Test System - 96” presentado en [4] y disponible resumidamente para el pre-despacho en [57]. La forma de construir las formulaciones básicas de pre-despacho fue tomada textual de la propuesta en [57]. El ejemplo base consta de 10 unidades, escogidas al azar por sorteo de una variable aleatoria de distribución uniforme. Los datos y detalles topológicos del ejemplo sintético se muestran en el Anexo E.

4.3.1.1 Parámetros de sintonización

A continuación se muestran los resultados de las simulaciones para distintos valores de la longitud del paso “*h*” escogida para el método del subgradiente (ver Anexo C).

Longitud del Paso <i>h</i>	Tiempo Total (seg)	Iteraciones	Tiempo Datos (seg)	Tiempo Pre-solver (seg)	Gap Dual (%)
8	4.876	180	0.17	0	0.0087
12	8.946	317	0.188	0	0.0081
4	12.278	277	0.188	0	0.0089
7	1.329	66	0.172	0	0.0099
6.8	4.237	162	0.187	0	0.0095
20	8.529	287	0.204	0	0.0082

Tabla 4.7. Parámetro fundamental de sintonización.

Nótese que las variaciones del número total de iteraciones se encuentran en el rango [66 – 317] para variaciones del parámetro *h* [4 – 20]. Los tiempos de ejecución por su parte, son proporcionales al número de iteraciones. Se observó que el algoritmo en todos los casos convergió a la misma solución primal cuya función objetivo fue de 3.7175E+05 [us\$]. Sin embargo no así el dual, evidenciado por las diferencias de los valores del gap dual. La Ilustración 4.5 permite visualizar la dispersión de los resultados evidenciando la sensibilidad del parámetro *h* sobre el rendimiento del método.

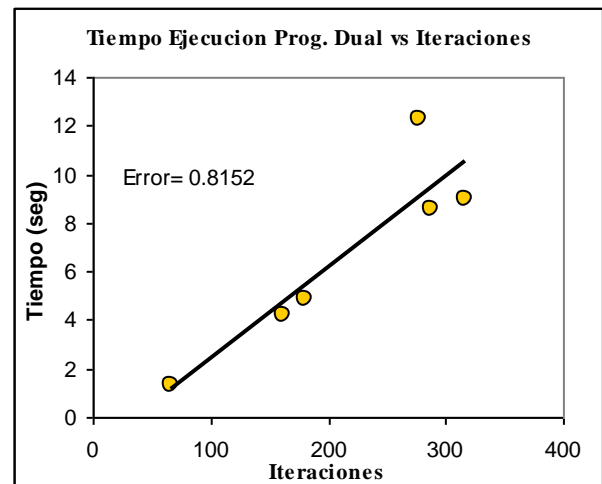
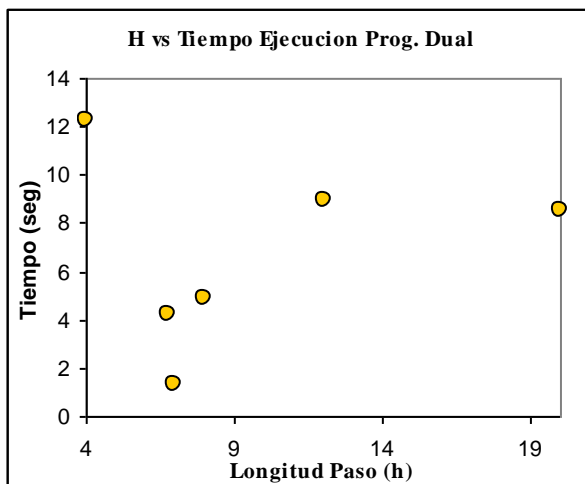


Ilustración 4.5. Efecto de la longitud del paso y la relación “lineal” iteraciones-tiempo.

También se observa que, efectivamente, el esfuerzo computacional en cada iteración del algoritmo de relajación lagrangiana es constante, lo que se ve reflejado en la proporción lineal que guarda con los tiempos totales de ejecución.

4.3.1.2 Efecto del número de unidades

A continuación se muestran los resultados para problemas de pre-despacho con un número de unidades crecientes en el rango [10–100]. Nuevamente, se realizó una muestra aleatoria de las unidades disponibles para generar cada una de las instancias. La demanda se escaló proporcionalmente con la capacidad instalada total de las unidades (ver detalles del método de selección en [57]).

Unidades	Tiempo Total (seg)	F. Obj (us\$)	Iteraciones	Gap Dual (%)	Longitud del Paso
10	2.12	3.72E+05	89	0.0099	7
20	16.717	5.31E+05	130	0.0014	7
40	0.843	1.06E+06	5	0.0054	7
60	3.91	1.66E+06	3	0.0047	7
100	0.704	2.64E+06	3	0.0042	7

Tabla 4.8. Parámetro fundamental de sintonización.

La Ilustración 4.6 permite visualizar nuevamente la dispersión de los resultados. Sin embargo, se evidencia que la relajación lagrangiana es, en definitiva, un atractivo método para resolver problemas de grandes dimensiones. Estos resultados serán debidamente comparados más adelante con los obtenidos por un método de solución primal ejecutado por una conocida rutina comercial.

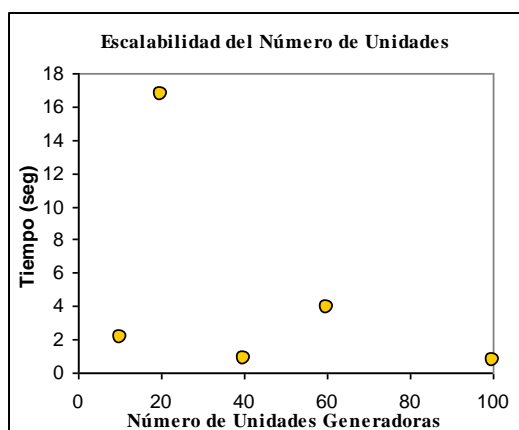


Ilustración 4.6. Respuesta ante número variable de unidades

Los resultados corroboran que, una particularidad de LR, resulta en una reducción del gap dual en proporción a las dimensiones del problema [19].

4.3.1.3 Efecto del número de períodos de tiempo del horizonte de evaluación

Se muestran resultados simples para períodos de evaluación en el rango [4 – 168] hrs. A pesar de ser periodos probablemente “ficticios” para la planificación de la operación de corto plazo, pueden ser reales para proyectos de planificación de mediano y largo plazo. En todo caso, se pretende ilustrar la respuesta del método ante los incrementos en esta dimensión (poco tratado en la bibliografía). Para este caso nuevamente se empleó el ejemplo básico de 10 unidades.

Longitud del Paso	Periodos	Tiempo Total (seg)	Tiempo Dual (seg)	F. Obj (us\$)	Iteraciones	Gap Dual (%)
7	4	16.954	0.713	3.2881E+04	150	0.03
1.5	4	15.094	0.698	3.2881E+04	150	0.013
4	12	48.343	1.668	1.5173E+05	150	0.0167
0.8	12	49.212	1.784	1.5173E+05	150	0.0144
7	24	45.422	1.6433	3.7175E+05	66	0.0099
7	168	1488.4	52.385	2.7921E+06	300	0.08511

Tabla 4.9. Desempeño del algoritmo - número de períodos de tiempo

Los valores de la longitud del paso mostrados fueron los mejores obtenidos para 5 ensayos experimentales. El número de iteraciones en este caso fue reducido en ciertos casos a 150. A pesar de

que, en ocasiones, el algoritmo se detuvo por el número máximo de iteraciones, los resultados se consideran comparables por las similitudes de las brechas duales obtenidas. La Ilustración 4.7 muestra el tiempo de solución del dual en función de los períodos de tiempo.

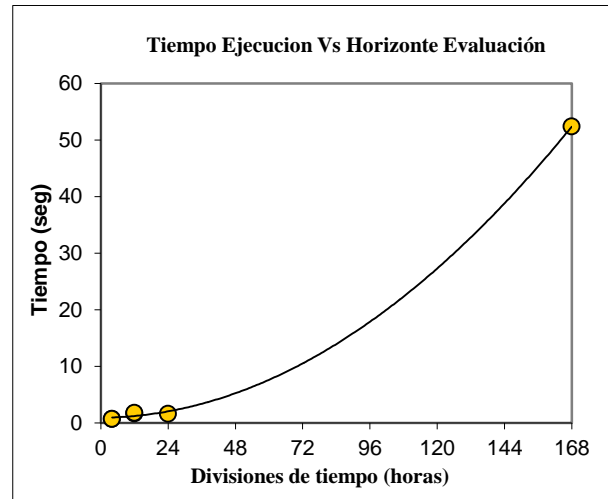


Ilustración 4.7. Respuesta ante número variable de períodos de tiempo

Si bien la evaluación anterior demostró una de las fortalezas de la relajación lagrangeana, ésta en cambio mostró quizás una de sus debilidades. La respuesta ante el número creciente de períodos fue, en este caso, no-lineal con un marcado crecimiento al final (168hrs). En el caso del UC semanal, LR requirió un tiempo de ejecución 30 veces mayor que para el problema diario (dimensiones del problema UC semanal es 7 veces mayor al diario). Además, se registró una mayor dificultad para encontrar soluciones factibles haciendo necesario el ajuste reiterativo del parámetro “h” para cada uno de los casos en estudio como evidencia la Tabla 4.9.

4.3.1.4 Efecto de las restricciones de rampa y reserva

A continuación se muestran los efectos de la inclusión de las restricciones de acoplamiento “reserva en giro”, sobre el desempeño general del algoritmo (y la solución final). El valor de la reserva en giro fue escogido arbitrariamente al 5% y 10% del total de la demanda en cada uno de los períodos de tiempo.

Longitud del Paso	Tiempo Total (seg)	F. Obj (us\$)	Iteraciones	Gap Dual (%)	Reserva Mínima (%)
7	1.62	3.7175E+05	66	0.0099	5
8	4.408	3.7175E+05	57	0.0099	5
10	1.282	3.7175E+05	51	0.0099	5
11	2.153	3.7175E+05	87	0.0099	5
5	3.55	4.0389E+05	150	0.0939	10
1.5	3.74	4.0730E+05	150	0.107	10
10	3.75	4.2788E+05	150	0.1803	10

Tabla 4.10. Efecto de la restricción de reserva en giro

Para este caso, el mejor resultado fue obtenido con $h=10$. Nótese que sin la restricción de reserva, los mejores resultados se habían obtenido para $h=7$. Por lo tanto, no existe un patrón, al menos evidente, de la obtención del mejor valor de este parámetro. Por ello, se confirma que el mejor procedimiento es experimental [27]. Como era de esperarse, la asignación de reserva del 10% encarece notablemente la solución final porque se requiere el despacho de la segunda unidad U197.

Para demostrar los efectos de la inclusión de las rampas en el mismo ejemplo, se realizaron las simulaciones respectivas para ambos casos. Se decidió mantener constante el parámetro de sintonización ($h=10$) para una comparación directa.

Longitud del Paso	Tiempo Total (seg)	F. Obj (us\$)	Iteraciones	Gap Dual (%)	Reserva (%)
10	3.96	4.2788E+05	150	0.177	10
10	1.457	3.7175E+05	57	0.0099	5

Tabla 4.11. Efecto de las tasa de toma de carga

En la Ilustración 4.8 mostrada a continuación se muestran los resultados obtenidos de la Tabla 4.11. Los puntos de color amarillo son los tiempos obtenidos con la restricción de reserva en giro activa y, en consecuencia, los rojos son aquellos obtenidos al resolver el problema sin restricción de reserva.

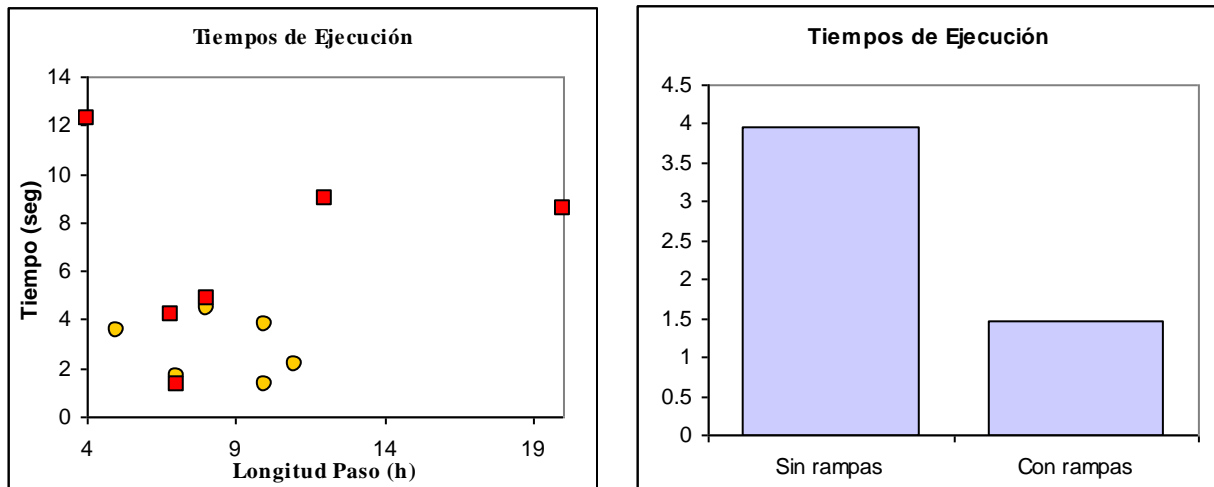


Ilustración 4.8. Efecto de las restricciones de reserva y de rampa

Los tiempos de ejecución se ven reducidos dado que las restricciones de rampa restringen el espacio de soluciones durante el proceso de programación dinámica de las unidades, lo que reduce el tiempo de ejecución del dual.

4.3.1.5 Comparaciones con un modelo comercial MIP

A continuación se presentan los resultados del algoritmo desarrollado con aquellos obtenidos mediante el empleo de una rutina comercial CPLEX [49]. Este es un código de optimización que puede resolver problemas generales de programación entero-mixta haciendo uso del método Branch-and-Cut¹. La explicación y validez de este método escapa de los objetivos del presente trabajo. Los fundamentos del método se pueden encontrar en la referencia [48] y algunos detalles de la implementación en CPLEX, (así como las heurísticas subordinadas que pueden ser invocadas) en [49]. El branch-and-cut de CPLEX es reconocido como uno de las mejores opciones comerciales de programación entero-mixta en cuanto a la robustez y desempeño. Además, ha sido evaluado exitosamente en aplicaciones de UC de dimensiones reales (sistema PJM) en [5]. Por tanto, las comparaciones permitirían:

- Validar los resultados con aquellos obtenidos por un software desarrollado.

¹ Se denomina brach-and-cut a la generalización del conocido método de programación Branch-and-Bound en donde se agregan ciertas desigualdades (planos cortantes) a la relajación LP.

- Comparar el desempeño y efectividad del método de relajación lagrangeana para resolver el pre-despacho con el mostrado por el método de solución primal branch-and-cut.

La Tabla 4.12 muestra los resultados de las simulaciones usando el CPLEX para formulaciones básicas de pre-despacho con un número de unidades en el rango 10-100(se usaron los mismos datos y modelo de pre-despacho que para los casos analizados en la sección 4.3.1.2). Inmediatamente después, en la tabla 4.13, se muestran los resultados obtenidos con la relajación lagrangeana.

Unidades	Tiempo Total (seg)	F. Obj (us\$)	Iteraciones	Tiempo Datos (seg)	Tiempo Pre-solver	Gap (%)
10	3.84	3.7175E+05	427	0.41	0.36	0.01
20	24.31	5.3594E+05	1204	0.53	0.95	0.07
40	5.19	1.0626E+06	910	0.43	1.31	0
60	6.18	1.6611E+06	1256	0.41	2.16	0.01
100	727.63	2.6367E+06	19128	0.68	3.33	0.01

Tabla 4.12. Resultados ejemplos sintéticos con MIP (CPLEX)

Unidades	Tiempo Total (seg)	F. Obj (us\$)	Iteraciones	Gap Dual (%)	Longitud del Paso
10	2.12	3.7175E+05	89	0.0099	7
20	16.717	5.3083E+05	130	0.0014	7
40	0.843	1.0650E+06	5	0.0054	7
60	3.91	1.6636E+06	3	0.0047	7
100	0.704	2.6393E+06	3	0.0042	7

Tabla 4.13. Resultados ejemplos sintéticos del modelo de Relajación Lagrangeana

Los resultados muestran una paridad de ambos métodos tanto en la calidad de la solución como respecto al esfuerzo computacional requerido. La ilustración 4.9 a continuación, compara gráficamente

los resultados de los tiempos de ejecución y los valores de la función objetivo. La ilustración 4.10 muestra los resultados de las iteraciones requeridas por cada uno de los métodos y, finalmente, la ilustración 4.11 compara la calidad de la solución obtenida por el branch-and-cut del CPLEX, graficando la diferencia o ahorro (resta de los valores de la función objetivo) entre ambas soluciones.

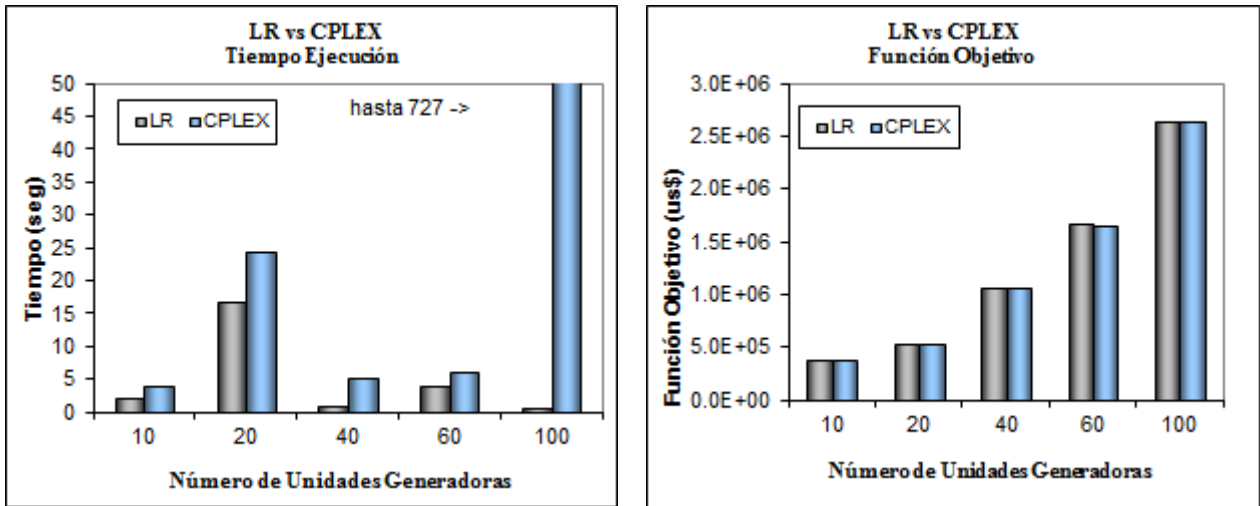


Ilustración 4.9. Comparación Tiempos de Ejecución y Calidad de la Solución en función del número de unidades generadoras

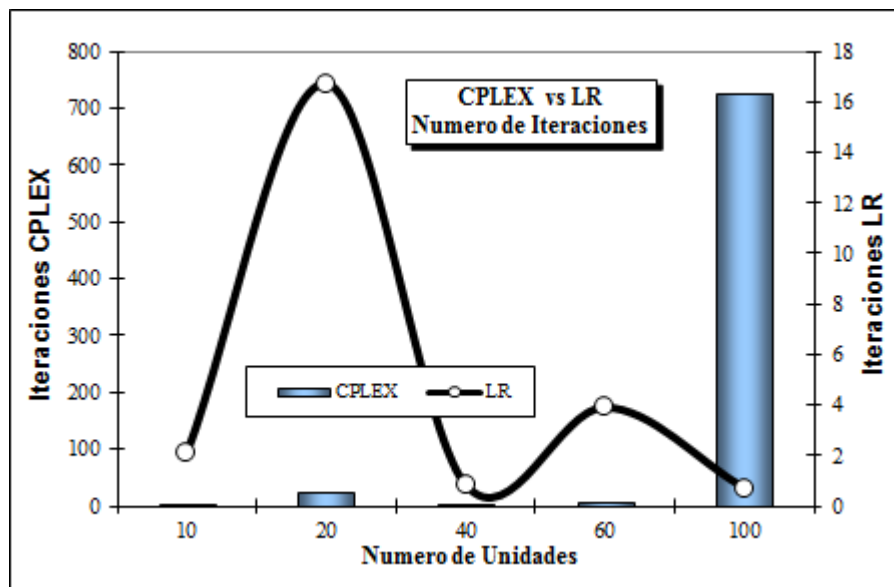


Ilustración 4.10. Comparación del número de Iteraciones en función del número de unidades generadoras

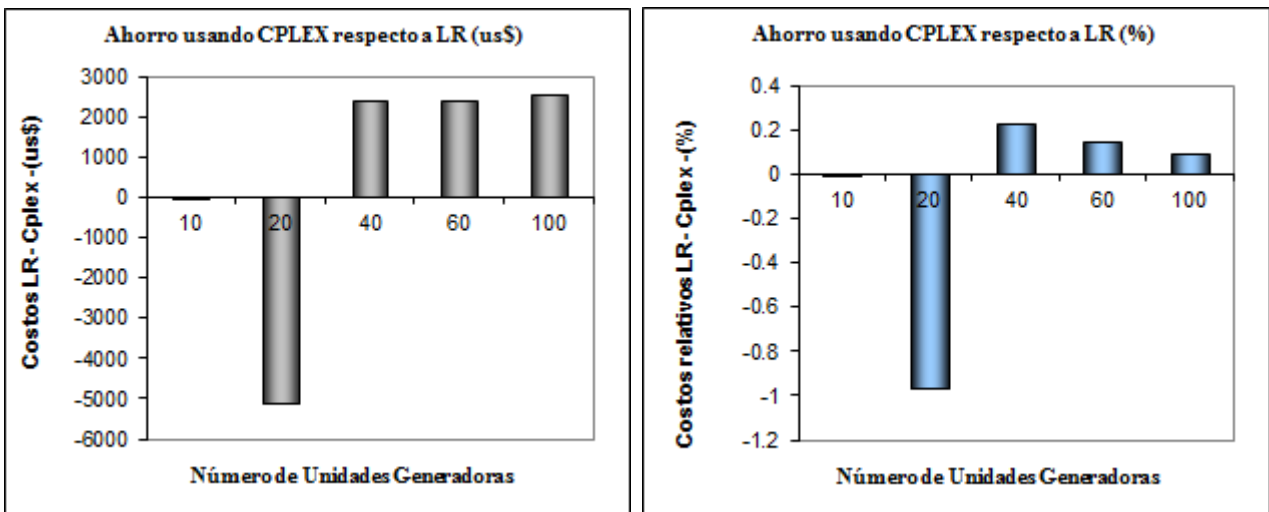


Ilustración 4.11. Comparación del ahorro obtenido al usar el CPLEX respecto a LLR

Las Ilustración 4.9, la Ilustración 4.10 y la Ilustración 4.11, en conjunto con la Tabla 4.12 y Tabla 4.13, demuestran empíricamente que ambos métodos resuelven el pre-despacho obteniendo soluciones de calidades muy similares. Se valida de esta forma el método y la rutina de programación en el Capítulo 3 por comparación.

Adicionalmente, se observa que la calidad de las soluciones del CPLEX fueron en general superiores para problemas de 40, 60 y 100 unidades. Sin embargo, los tiempos de ejecución e iteraciones necesarias fueron notablemente más elevados. Se observa además el “buen” comportamiento de la relajación para dimensiones mayores, porque las curvas del número de iteraciones disminuyeron con la dimensión del problema. Además, se mantuvo constante la calidad de la solución y, en los casos mencionados, fueron muy similares a las obtenidas por el *branch-and-cut*. Quizás lo que no pueda observarse en los gráficos es el tiempo que toma en determinar el valor de la mejor longitud del paso ($h=7$) requerido previamente por la relajación. Sin embargo, una vez obtenido experimentalmente, el desempeño de la metodología ha sido inclusive superior, comparativamente, con el mostrado por el CPLEX.

4.3.2 Comparación estadística

4.3.2.1 Introducción

En la bibliografía de optimización y métodos heurísticos, se han propuesto completas guías para la elaboración y evaluación de experimentos computacionales. Desafortunadamente, difícilmente son seguidas para la elaboración de los reportes en la literatura. En [58] se dan pautas para la medición del rendimiento de experimento con heurísticas. De acuerdo a esta referencia, las comparaciones deberían concluir acerca de: resultados computaciones, robustez y velocidad de convergencia. Las comparaciones deben venir acompañadas de las bases sobre las cuales fueron desarrolladas, por lo que las conclusiones pueden significar poco sin contar al menos con descripción del código, detallada especificación computacional, valores de sintonización. En [59] se mencionan y analizan las metodologías de creación de casos de prueba. Hay tres variantes usuales de creación de casos: casos prácticos (realismo), problemas estándar (compasiones directas) y aleatorio (sintéticos). Los buenos generadores de casos de prueba deben controlar los atributos de los problemas de optimización: número de variables y restricciones, distribución de los coeficientes función objetivo y restricciones, términos constantes de restricciones, dispersión de la matriz de restricciones y correlación entre función objetivo y cada restricción. Las fuentes de obtención de casos de pruebas, según [60] son las siguientes: problemas reales, variantes aleatorias de casos reales, librerías públicas y generador aleatorio de casos. Hay tres formas de creación de problemas sintéticos [59]: generación de coeficientes en forma aleatoria, mediante correlación implícita y correlación explícita. Los resultados de la comparación analítica de los métodos de creación de casos en [59] sugiere, para la creación de casos sintéticos, el uso de la correlación explícita porque resulta la más adecuada para la selección aleatoria multivariable.

4.3.2.2 Motivación

De acuerdo con [60], existen las siguientes alternativas para la creación de casos de prueba:

- Set de datos reales: Presentan como principalmente ventaja la posibilidad de relacionar la capacidad del algoritmo para resolver instancias reales. Sin embargo, los resultados obtenidos pueden ser caso-específicas.
- Variaciones aleatorias de casos reales: es una modalidad poco usada en la bibliografía. Sin embargo, justamente logra superar la limitación del uso de casos específicos reales, al brindarle generalidad a los resultados.
- Librerías públicas de casos de prueba: como principal ventaja se tiene que las librerías de uso público es posible encontrar una variedad de soluciones y de tal forma que se puede validar y contrastar los resultados versus una variedad de métodos y heurísticas. Además, puede que las instancias hayan sido diseñadas cuidadosamente bajo ciertos estándares, de tal forma que la responsabilidad por crear casos interesantes puede ser delegada. La principal desventaja es que no siempre hay disponible librerías para los problemas que se desea resolver, i.e existe una abundante variedad de librerías para el problema de TSP o MKP, pero difícilmente para otros problemas de optimización como el UC; y, además, pueden no ser útiles para investigar algunos factores de incidencia.
- Instancias aleatorias: presentan una variedad de fortalezas:
 - Las características de los problemas están bajo el control explícito del investigador.
 - Si el generador es documentado, las características de las instancias son conocidas. En contraste, como ocurre con algunas las librerías estándares, los orígenes de los casos pueden ser desconocidos.
 - Una vez propuestos, no existen límites sobre la cantidad de instancias que es posible generar.

Sin embargo, no están exentos de dificultades. Aunque en [59] y [60] presentan por separado una lista de los “pitfall” de los generadores aleatorios, en algo coinciden: si no existe un control sobre la correlación entre los valores y distribución de los coeficientes de la matriz de restricciones, entre ellos y con la función objetivo, las instancias generadas pueden alejarse demasiado de la realidad o pueden

terminar siendo poco representativas de los casos reales para los cuales se plantea el algoritmo de solución en estudio.

Los estudios prácticos tienen que limitarse a un espacio acotado de instancias de un determinado problema de optimización. La “sobre-generalización” de un problema, hará que el experimento peligre en caer en la trampa del teorema *no-free-lunch* [61]. No-free-lunch consiste en una serie de teoremas que demuestran estadísticamente que ningún algoritmo es mejor que otro sobre todas las posibles instancias de un problema de optimización. Cada instancia de un problema tiene asociado un valor de la función objetivo. Si llamamos A al conjunto de funciones y B a un conjunto particular de funciones a ser probadas, se define A-B como el conjunto no testado. El teorema NFL implica que si algoritmo K es mejor en promedio que otro algoritmo Z sobre el set A entonces el algoritmo Z será mejor en promedio que K sobre las instancias A-B. El NFL demuestra que la evaluación comparativa es una “suma a cero”. Considerando el NFL, las comparaciones sólo son significativas si se puede asumir que el set de casos prueba usado para las comparaciones son realmente representativas del subconjunto de problemas que se intenta resolver.

En la bibliografía del UC han sido empleadas distintas técnicas para la creación de casos de prueba, que difícilmente pueden enmarcarse en las alternativas bibliográficas citadas anteriormente. De acuerdo con [57], los ejemplos numéricos de UC que pueden ser comparables, son una forma básica del modelo propuesto en [17], la cual consiste en repetir unidades idénticas al conjunto básico de 10 unidades, escalando la demanda por un factor de proporcionalidad.

4.3.2.3 Generador de instancias de UC

El método de generación de casos de UC que se presenta a continuación, cuyas instancias creadas deberán respetar la estructura descrita en el Capítulo 2, intenta rescatar las bondades de cada una de las alternativas de generación de casos expuestas en 4.3.2.1, intentando generalizar tanto como sea posible las posibles instancias del UC.

Los principales fundamentos del generador de casos son los siguientes:

1. Realismo: Todas las unidades empleadas por el generador de instancias se basan en las características técnicas de las unidades en la referencia [4]. El comité de confiabilidad, *Reliability Test System Task Force*, ha propuesto y mejorado continuamente el modelo sintético RTS versión 96 específicamente para estudios como el que se desarrolla en el marco de la presente tesis. El problema que surge con el RTS-96 es el escaso número de unidades (sólo nueve), por lo que los autores para poder generar instancias de mayores dimensiones han tenido que “repetir” los datos para otras semejantes. Como se verá en el próximo punto, esto puede ser mejorado haciendo uso de algunas variables aleatorias.
2. Diversidad de las unidades generadoras: para generar variedad, se emplearán dos métodos:
 - Diversidad tecnológica: Mediante un sorteo aleatorio de distribución uniforme, se escogerá una de las nueve unidades disponibles en [4], la unidad candidata “j”.
 - Diversidad de características propias (Parámetros técnicos): a la candidata “j” seleccionada anteriormente, se le sumará vectorialmente una variable aleatoria cuya función de distribución de probabilidad estará dada por una función gaussiana, centrada en el parámetro del RTS (i.e. $\mu=0$) y cuya dispersión σ será controlada y evaluada durante el presente análisis.
3. Diversidad del requerimiento de energía: En función de las curvas de demandas en la misma referencia, se realizará un sorteo aleatorio de distribución uniforme para escoger alguna de las alternativas de la forma normalizada (en p.u.) de la curva de demanda. La curva de demanda se ajusta a la capacidad de generación mediante un factor de adaptación (f). Este factor también es parte de las variables controladas del experimento.
4. Correlación explícita: Los coeficientes de la matriz de restricciones serán dispuestos acorde con las ecuaciones (2.1)-(2.8), manteniendo la consistencia de la matriz de restricciones con las características de un problema de UC. Los valores de los coeficientes no-nulos de la matriz de restricciones serán determinados por los sorteos aleatorios de tecnologías y diversidad anteriores.

El proceso de creación de casos basado en los fundamentos anteriores se resume en la Ilustración 4.12 mostrada a continuación:

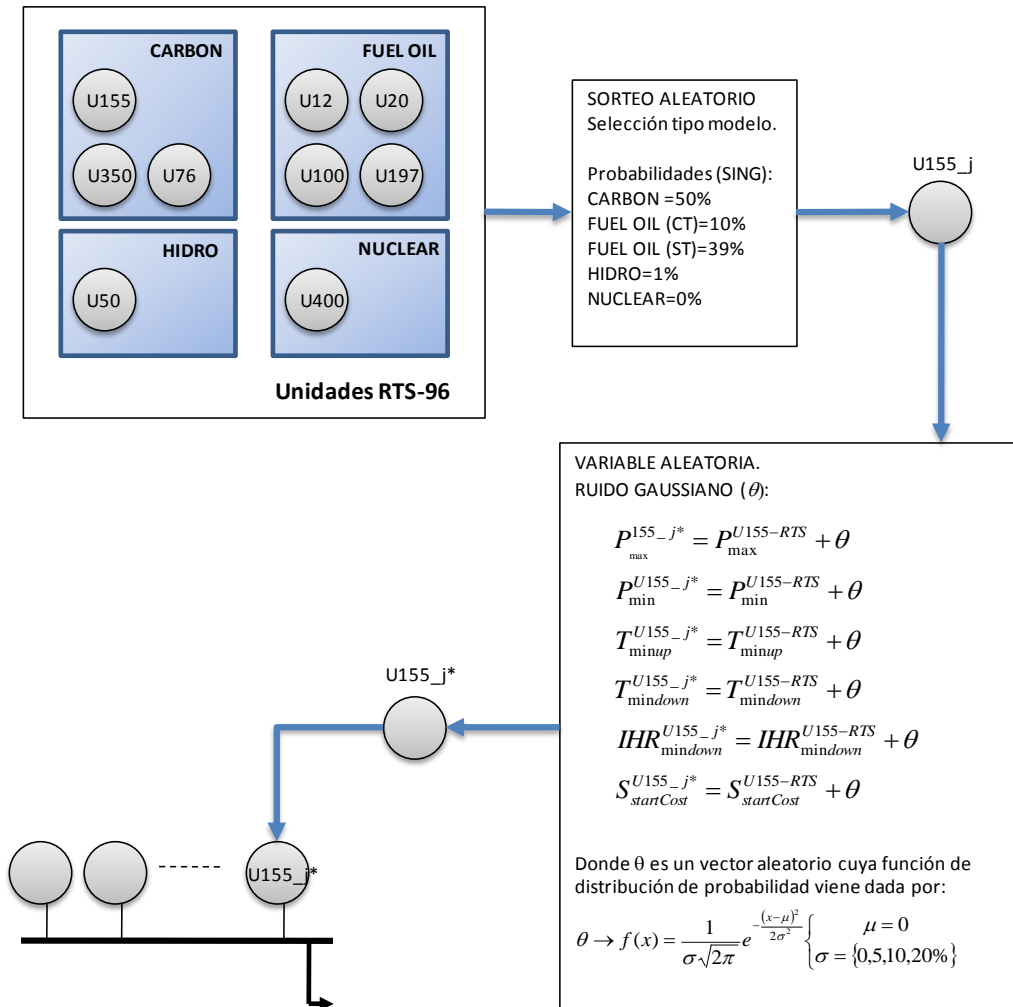


Ilustración 4.12. Generador de instancias de UC. Ejemplo selección unidad U155 RTS-96.

En la Ilustración 4.12 se puede observar que el proceso parte con la información del RTS-96 en [4]. Esta distribución uniforme, o mediante probabilidades discretas, puede estar relacionada con alguna zona geográfica en particular donde la distribución de los recursos e insumos sea conocida o se desee evaluar. Por ejemplo, en la siguiente tabla se muestran las distribuciones para ambos sistemas interconectados en Chile y Estados Unidos.

Tecnología	SING [Chile] [56]	SIC [Chile]	Estados Unidos [62]
Carbón	50%	9%	44%
Diesel-Fuel Oil-Gas	49%	38%	19%
Hidro	1%	53%	20%
Nuclear	0%	0%	17%

Tabla 4.14. Distribución aproximada principales recursos energéticos SING (norte Chile), SIC (centro Chile) y USA.

Una vez realizado el sorteo y seleccionada la unidad “prototipo”, con el objeto de lograr la diversidad, sus parámetros son “alterados” al sumarle un valor θ producto de una muestra aleatoria de distribución normal (4.2):

$$\begin{aligned}
 \text{Parametro}_{RTS}^i &= \text{Parametro}_{RTS} + \theta \\
 \theta &= [X]: f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
 \end{aligned}
 \tag{4.2}$$

Donde $\mu=0$ y σ es la variable de control del experimento. En efecto, será estudiado el resultado sobre los tiempos de ejecución y calidad de la solución para variaciones discretas en el rango [0-20%]. Los datos de las unidades para todos los casos estarán entonces “centrados” en los parámetros del RTS-96 en la Tabla 4.15.

Unidad	Fuel	Fuel Price ²	Start Fuel Price ^{2,3}	Ramp	Hot Start	Cold Start	Incr. Heat ⁴
		mills/kBTU	mills/kBTU	MW/min	MBTU	MBTU	Btu/kWh
U12	Oil #6	6.720	6.720	1	38	68	13219.00
U20	Oil #2	6.192	6.192	3	5	5	14427.00
U50	Hidro	0.000	0.000	0	0	0	0.00
U76	Coal	1.623	1.623	2	596	596	13311.00
U100	Oil #6	6.720	6.720	7	250	566	9877.00
U155	Coal	1.623	6.192	3	260	953	9381.00
U197	Oil #6	6.720	6.720	3	443	775	9620.00
U350	Coal	1.623	6.192	4	1915	4468	9768.00
U400	LWR	5.100	0.000	20	0	0	9436.00

Tabla 4.15. Datos del sistema RTS-96 [4].

Finalmente, la curva de demanda es obtenida mediante el simple proceso a continuación:

- En función de los porcentajes de distribución horaria, diaria y semanal de la demanda máxima anual en [4], se construyen las curvas diarias de demanda (con resolución horaria) para cada uno de los 365 días de un año calendario.
- Se realiza un sorteo aleatorio con probabilidad uniforme, donde cada curva de demanda tiene una probabilidad de ser seleccionada igual a 1/365.
- Cada valor horario de la curva de demanda seleccionada D_{pu} , es multiplicada por la capacidad instalada de generación de la instancia a la cual pertenece y multiplicada a su vez por un factor de escalamiento f del rango [0,1]. Es decir, $D_{MW} = D_{pu} \cdot \sum_i^N P_i \cdot f$. Este factor es otra de las variables

² Los valores en esta columna no pertenecen al RTS-96. Estos valores fueron aproximadamente los vigentes durante el año 2006 en el sistema SING según referencia [51].

³ Por simplicidad, se asumió que el combustible durante el proceso de partida es igual al combustible de operación, lo cual es conocido por no ser necesariamente iguales [57].

⁴ Estos valores corresponden al punto de la curva IHR de mayor potencia de salida.

de entrada controladas del experimento. Evidentemente, el método descrito requiere que sea ejecutado previamente el sorteo de las unidades candidatas.

De esta forma el modelo propuesto permite cumplir con lo siguiente:

- Cumplir con las sugerencias en [58], teniendo precaución de evitar los “pitfall” de la generación aleatoria señalados en [59] y [60].
- Como se sugiere en [59], la diversidad de las instancias estará controlada por los parámetros de las funciones de distribución: dispersión de tecnologías, dispersión del ruido gaussiano a los parámetros operacionales de las unidades, número de unidades y coeficiente de adaptabilidad (f).
- Desde el punto de vista de las curvas de costos marginales sistémicos, el generador de casos permitirá evaluar en forma experimental el desempeño de los principales algoritmos para resolver el UC, partiendo de una aguda curva escalonada -por la presencia de grandes grupos de unidades iguales (caso dispersión nulo)- hacia una curva suavizada producto de la dispersión del IHR de las unidades.
- Estadísticamente, el valor esperado de los parámetros técnicos de las unidades de cada caso será aproximado a una librería fundamentada y especializada para estudios de sistemas de potencia.
- No hay límites de escalabilidad en cuanto al número de unidades. El aumento de las dimensiones no distorsionará la representatividad del problema. Es decir, las instancias no se crean por repetición como tradicionalmente ha sido empleado en la bibliografía del UC [57], evitando una estructura ficticia por bloques simétricos en la matriz de restricciones (lo cual puede beneficiar/perjudicar artificialmente a uno u otro método de solución).
- El proceso es automático, por lo que no está sujeto a intervención involuntaria.
- Las instancias no serán alejadas de la realidad siempre que los valores de dispersión sean razonables.
- La matriz de restricciones tendrá en todo momento las características de dispersión y pesos propias del UC, por el cumplimiento de (2.2)-(2.8).
- La aplicación de estadísticas para el análisis de los resultados es directa [63].

En resumen, el generador de casos propuesto permitirá plantear, luego de aplicar análisis estadístico a la variedad amplia de resultados de casos no alejados de la realidad, conclusiones acerca de la estabilidad de los tiempos del MIP y podrá exponer, por primera vez en forma práctica, las debilidades intuitivas y conceptuales usualmente relacionadas a la relajación lagrangeana.

4.3.2.4 Especificaciones

El proceso de creación y ejecución de casos fue automatizado mediante el uso de las siguientes herramientas, desarrolladas también en el marco de la presente investigación:

- Generador de instancias: Planilla Excel con Macros. Los datos de entrada son la Tabla 4.15 y los parámetros controlados según 4.3.2.3.
- Generador de casos relajación lagrangeana: Planilla Excel con Macros para la generación automática de archivos de texto con formato de entrada para la rutina JAVA de relajación lagrangeana.
- Generador de casos CPLEX (formato .lp): Página web programada en lenguaje PHP que, a partir de un archivo de texto separado por comas, genera el archivo formato CPLEX lp.
- UCOPT: Algoritmos de LR descrito en el Capítulo 3, programado en lenguaje JAVA y compilado con la versión 6.0.

Todas ellas son de libre acceso y se encuentran disponibles en [64].

El proceso completo y los pasos se describen en la Ilustración 4.13 mostrada a continuación.

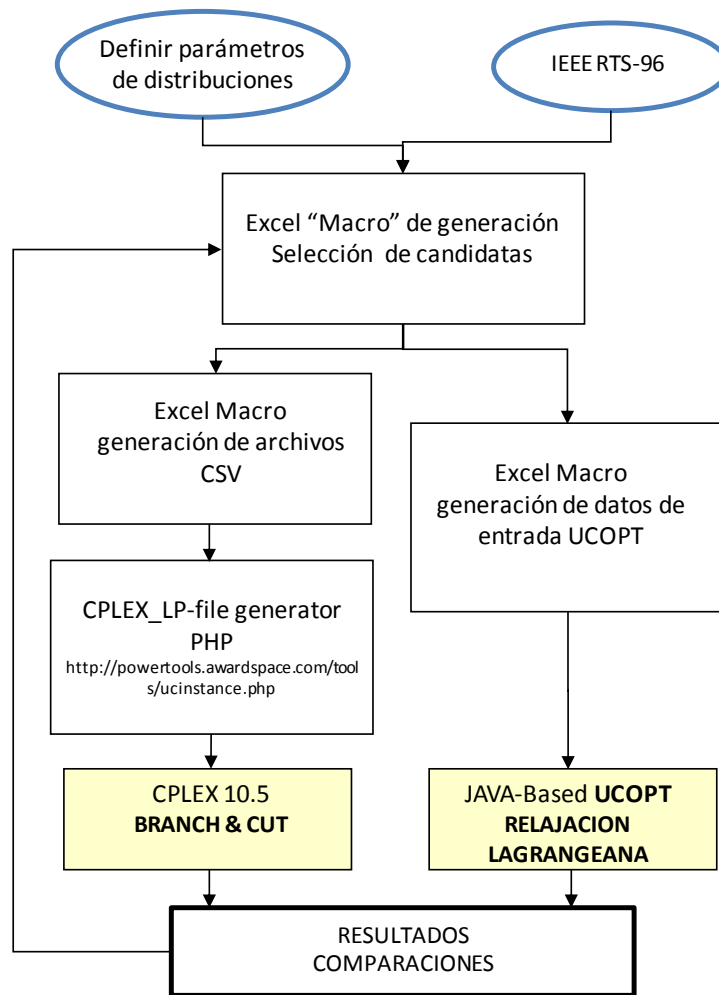


Ilustración 4.13. Proceso de creación, simulación y análisis de resultados de instancias sintéticas de UC.

El ambiente computacional también fue cuidadosamente reservado. Las especificaciones en detalle se muestran en la Tabla 4.16. En cuanto a la sintonización del software requerido, la Tabla 4.17 muestra los valores empleados para la relajación, mientras que la Tabla 4.18 muestra los parámetros configurables del CPLEX.

Característica	Valor
Modelo Comercial	Lenovo T400
Velocidad CPU (GHz)	2.53 GHz
Nº de procesadores	1 (doble core)
Memoria RAM	4 GB
Índice SPEC [50]	18.8 ⁵
Dhrystone	1981,7
Whetstone	640,2
Sistema Operativo	Windows 7
Versión JAVA	6.0

Tabla 4.16. Especificación del ambiente de simulación.

Parámetro	Valor
Gap Dual	0.01%
Método subgradiente	Paso constante
H (subgradiente)	0.9
Tiempo Máximo	10min
Despachos Económicos	Programación lineal (Simplex). GLPK 4.8

Tabla 4.17. Parámetros de simulación UCOPT (Relajación Lagrangeana).

Parámetro	Valor
Gap	0.01%
Prioridad	Soluciones factibles.
Tiempo Máximo	10min
Método LP	Simplex

Tabla 4.18. Parámetros de simulación CPLEX 10.5 (MIP).

De esta forma, se considera que la obtención de un gap menor al 0.01%, por cualquiera de las metodologías, será considerado como el óptimo global del problema.

⁵ Índice "Base" SPEC 2006. Resultados corresponden al procesador Intel T9500, 2,60 Ghz en Dell Precision M6300. Fue escogido por ser el procesador de características más similares.

4.3.2.5 Casos y simulaciones

Mediante el generador de instancias descrito en 4.3.2.3 se construyeron 480 casos de prueba organizados de la siguiente manera.

f=0.3

Unidades	0%	5%	10%	20%	Total
10	20	20	20	20	80
50	20	20	20	20	80
100	20	20	20	20	80
Total	60	60	60	60	240

Tabla 4.19. Número de casos de prueba para f=0.3

f=1.0

Unidades	0%	5%	10%	20%	Total
10	20	20	20	20	80
50	20	20	20	20	80
100	20	20	20	20	80
Total	60	60	60	60	240

Tabla 4.20. Número de casos de prueba para f=1.0

Parámetro	Valor	Nº Casos
f	0.3	240
f	1	240
σ	0%	120
σ	5%	120
σ	10%	120
σ	20%	120
Unidades	10	160
Unidades	50	160
Unidades	100	160

Tabla 4.21. Resumen de casos disponibles para cada parámetro en estudio.

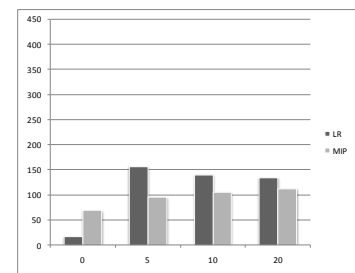
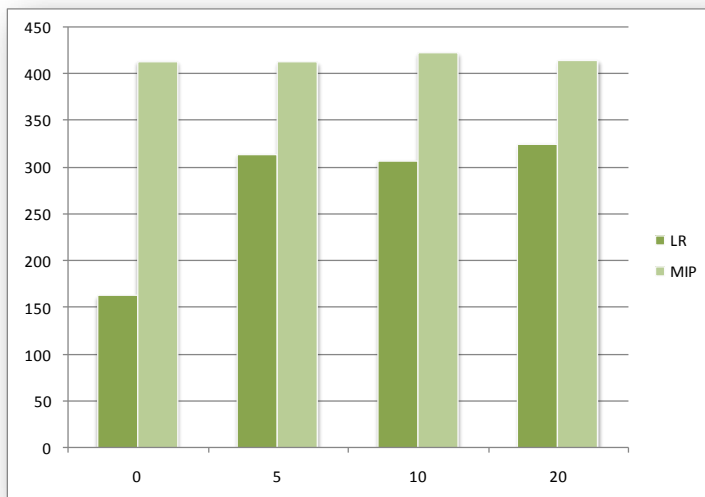
De esta forma el estudio permitirá concluir acerca de la adaptabilidad del sistema (nunca antes estudiada en la bibliografía), la variedad (dispersión) de las características propias de las unidades (tampoco estudiada en la bibliografía) y, por supuesto, del número de unidades.

4.3.2.6 Resultados y comparaciones

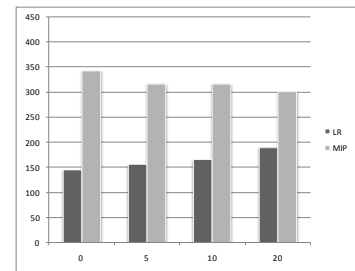
A continuación se presentan los resultados comparativos de las mejor soluciones factibles, tiempos de ejecución, gap final y otros valores relevantes para el análisis de rendimiento de los métodos bajo análisis, con sus respectivos gráficos y estadísticas, según sugerencias en [58].

- SOLUCIONES FACTIBLES Y MEJOR SOLUCIÓN ENTERA:

SOLUCIONES FACTIBLES



f=0.3



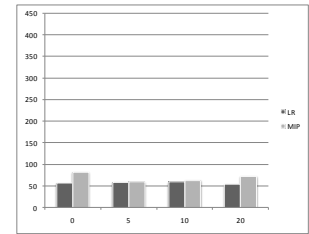
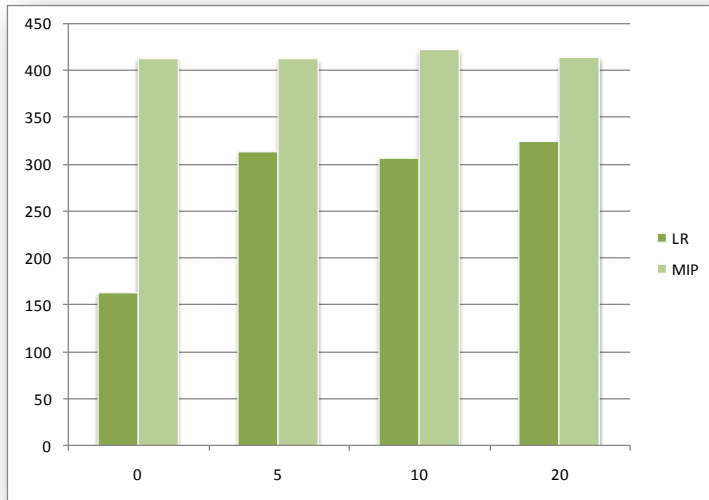
f=1.0

Ilustración 4.14. Número de soluciones factibles en función del coeficiente de adaptabilidad (f)

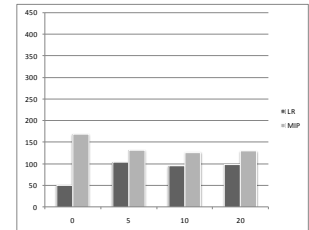
MIP no necesita evaluar muchas soluciones factibles para problemas “fáciles” (capacidad muy por encima de la demanda de energía). MIP consiguió 1501 soluciones factibles, lo que representa un 50% de más soluciones factibles que LR. Además, evaluó en promedio 3.5 soluciones factibles por cada caso, mientras que LR evalúa apenas 2.3 por caso. Se observa que cuando la dispersión de los parámetros es cero, lo que implica que las unidades serán iguales, se ratifican los problemas para encontrar soluciones

factibles cuando coexisten unidades iguales [15]. El problema es más grave mientras mayor sea la capacidad disponible, debido a que se requieren apenas unas “pocas” unidades y éstas son iguales.

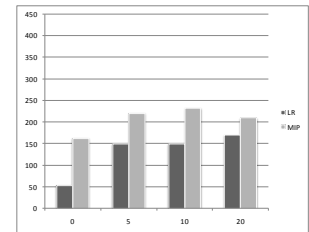
SOLUCIONES FACTIBLES



unidades=10



unidades=50

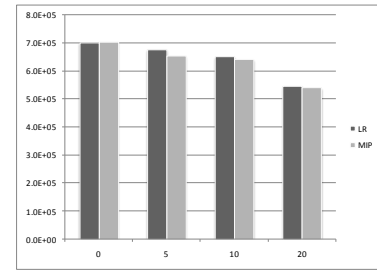
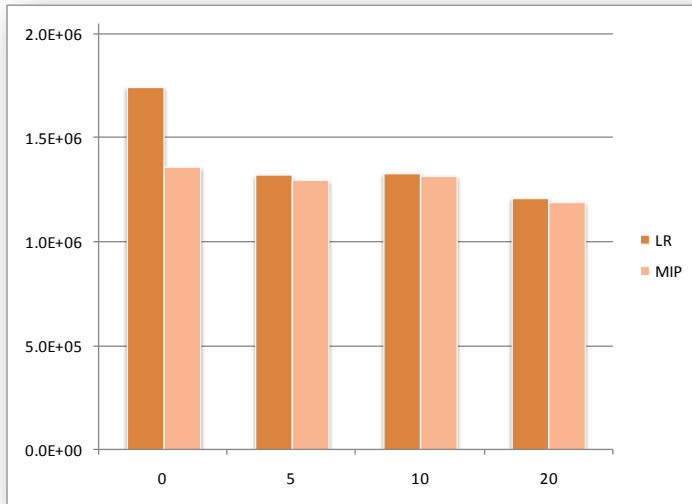


unidades=100

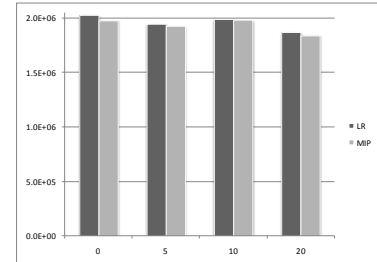
Ilustración 4.15. Número de soluciones factibles en función de la dimensión del problema (número de unidades)

El número de soluciones factibles que evalúa LR crece en proporción casi lineal al número de unidades. En forma global, el número de soluciones factibles del MIP, por su parte, se mantiene invariable ante las variaciones de los parámetros de tamaño y dispersidad.

MEJOR SOLUCIÓN ENTERA



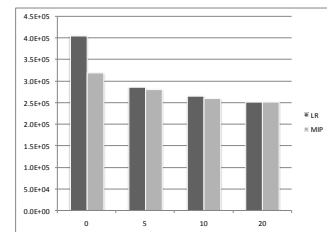
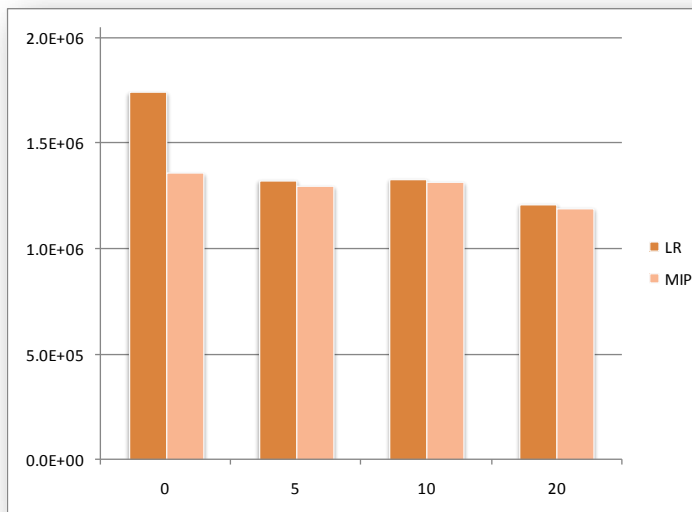
$f=0.3$



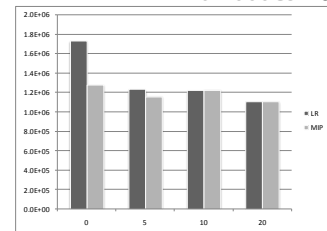
$f=1.0$

Ilustración 4.16. Promedio valor función objetivo en función del coeficiente de adaptabilidad (f)

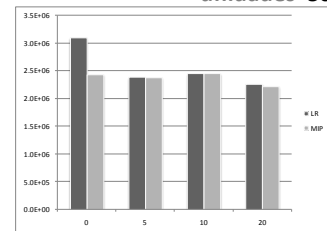
MEJOR SOLUCIÓN ENTERA



unidades=10



unidades=50



unidades=100

Ilustración 4.17. Promedio valor función objetivo en función de la dimensión del problema (número de unidades)

Para cualquiera de las dimensiones analizadas, conforme se observa en la Ilustración 4.16 e Ilustración 4.17, la mejor solución entera que encuentra LR tiene una menor calidad (valor mayor de la función objetivo en problema de minimización) que la arrojada por el optimizador entero-mixto. Estos resultados son consistentes, lo que definitivamente permite dimensionar el sobre costo esperado al utilizar LR. El aumento de la dispersión beneficia ambos métodos, porque existen unidades de menor costo para seleccionar para el despacho. Para las 480 instancias, MIP no logró arrojar la solución óptima (gap menor al 0,01%) en apenas 8 de las 480 instancias analizadas (98.33% de efectividad al resolver los problemas). En contraste, mediante el método LR por su parte, las soluciones aportaron garantías de optimalidad en tan sólo 90 (18.75%) de las instancias analizadas.

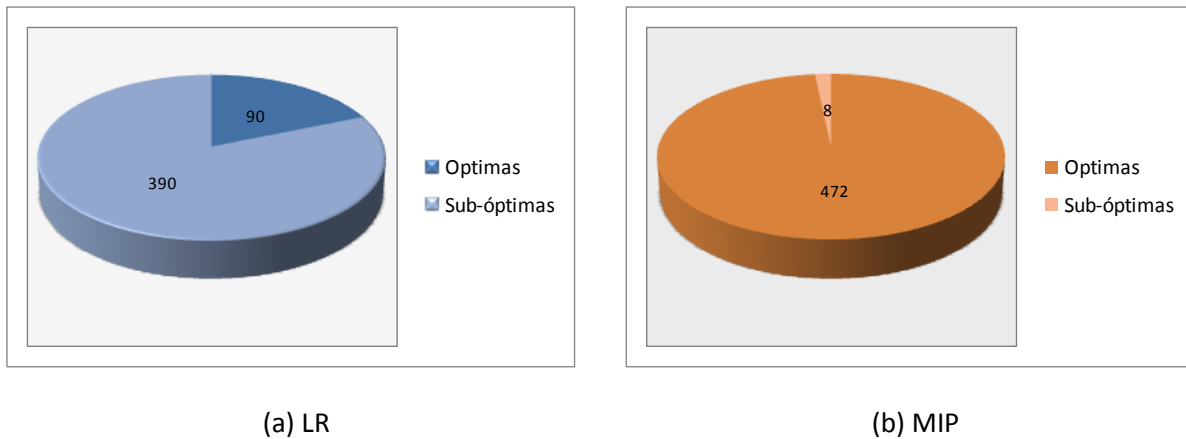


Ilustración 4.18. Número de soluciones óptimas

Quiere decir que resultará difícil mediante el método LR conseguir consistentemente soluciones de una elevada calidad al problema de UC. En la Ilustración 4.18 se muestra el número de soluciones óptimas para las instancias resultas por ambos métodos en función del tiempo máximo de ejecución. Del gráfico, se observa que al asignar los mismos recursos computacionales a ambos métodos, es mucho mayor el número de casos resueltos por MIP (cuya solución es óptima) respecto a LR. El algoritmo LR se detuvo, para el máximo tiempo de ejecución según la de las cuales solo el 18,75% de las soluciones cumplían con la exigencia de optimalidad.

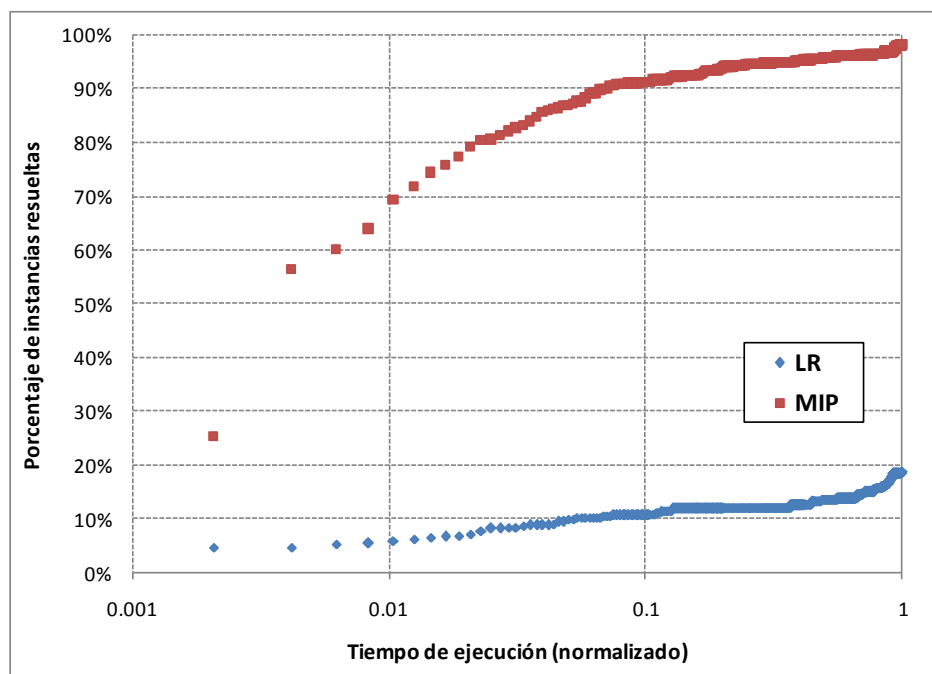
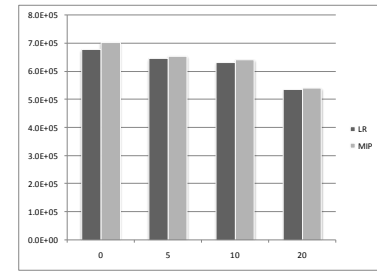
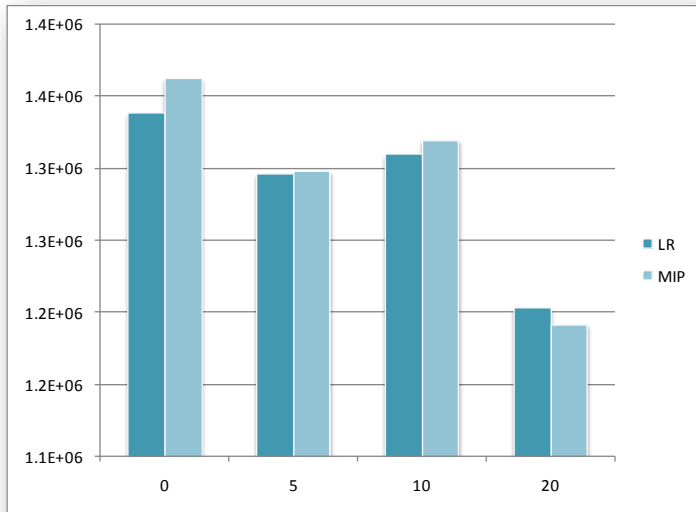


Ilustración 4.19. Casos resueltos en función del tiempo de ejecución.

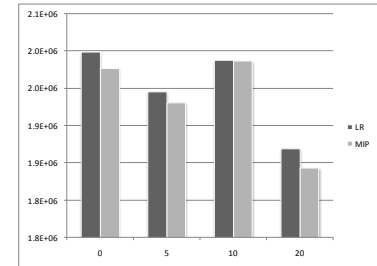
- MEJOR COTA INFERIOR:

De acuerdo con [7] y [48], la integración de LR en la programación entero mixta tiene una interesante aplicación de LR para proporcionar cotas inferior. Esto es especialmente cierto cuando al relajar ciertas restricciones “claves” los problemas desacoplados resultan mucho más fáciles. Sin embargo, aunque conocida en teoría, esta integración no ha sido explotada para problemas de UC, el cual cumple precisamente con la característica conceptual requerida [19]. Por lo tanto, es de mayor interés explorar la calidad de las cotas inferiores. A continuación se evalúan y comparan sistemáticamente los resultados de la maximización del dual respecto de la mejor relajación lineal encontrada mediante el proceso de podado del árbol en el branch-and-bound.

MEJOR COTA INFERIOR



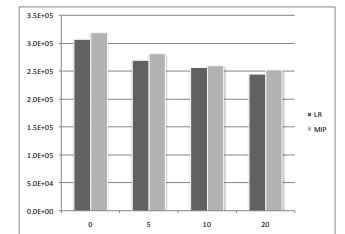
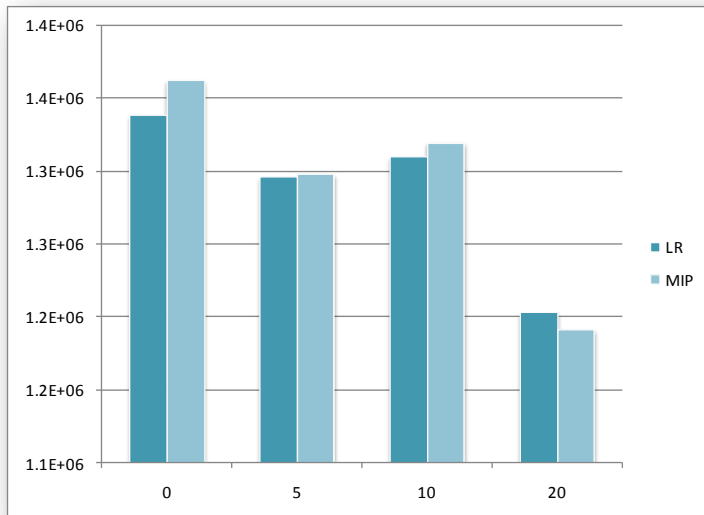
$f=0.3$



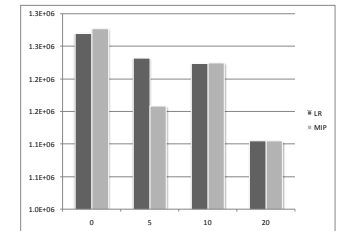
$f=1.0$

Ilustración 4.20. Promedio valor mejor cota inferior en función del coeficiente de adaptabilidad (f)

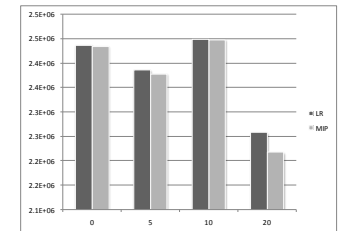
MEJOR COTA INFERIOR



unidades=10



unidades=50



unidades=100

Ilustración 4.21. Promedio valor mejor cota inferior en función de la dimensión del problema (número de unidades)

Es de notar que la mejor cota inferior obtenida por LR, la cual corresponde al máximo del problema dual, resulta mejor que la relajación lineal MIP para algunos de los parámetros controlados del experimento. Si bien las soluciones factibles encontradas por LR son inferiores en calidad a las obtenidas por MIP, para el caso de las cotas inferiores resulta diferente. Como se observa en la Ilustración 4.20, cuando el problema resulta mejor adaptado ($fp=1$), la mejor cota inferior promedio de LR fue siempre mayor a la del MIP.

Unidades	LR	MIP	Diferencia
10	270182.726	277960.227	-2.80%
50	1208122.63	1190337.95	1.49%
100	2382739.24	2369772.91	0.55%

Tabla 4.22. Promedio de la mejor cota inferior en función del número de unidades.

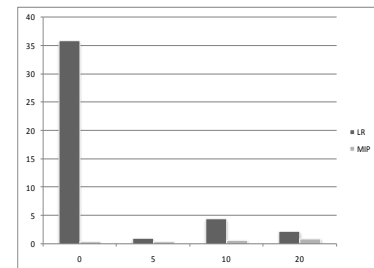
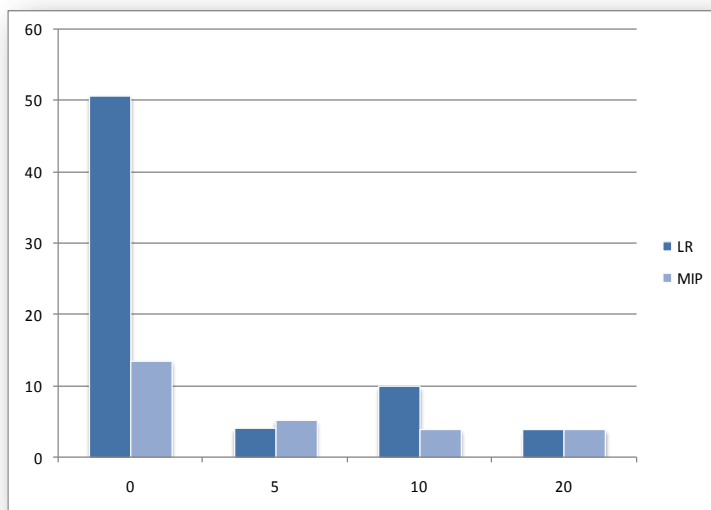
Además, la medida intuitiva de la calidad de LR para sistemas grandes, que como es conocido se ha asociado a la disminución del gap [19], puede observarse que los resultados experimentales permiten concluir que no debe atribuírsele a la mejora de las soluciones factibles respecto al óptimo global, sino que la función dual resulta más cercana al óptimo del problema. De acuerdo a los resultados experimentales mostrados en la Tabla 4.22, es posible deducir que el valor esperado de la función dual aumenta (mejora) respecto de la mayor relación lineal mediante el MIP a medida que el problema crece (juzgando por el número de unidades y, por lo tanto, por el número de variables binarias).

- TIEMPOS DE EJECUCIÓN:

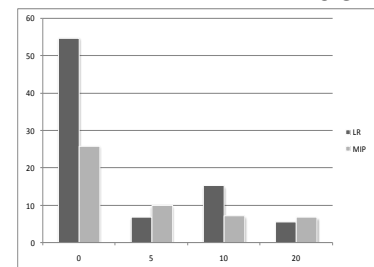
Incluso cumpliendo con lo especificado en [58], resulta difícil realizar comparaciones 1-a-1 de los tiempos de ejecución de dos algoritmos de optimización. Es el presente estudio, la principal razón es promovida por las diferencias de desarrollo entre ambos software. Mientras que CPLEX es un programa comercial con presencia dominante en el competitivo mercado del software, la rutina “amateur” de LR fue desarrollada tomando mínimas precauciones de optimizar el algoritmo, librerías, uso de memoria, etc. Aunque se sugiere al lector explorar escalar los resultados y hacerlos comparables asumiendo que la rutina LR puede ser mejorada, de acuerdo a los objetivos expuestos en el Capítulo 1, la principal

preocupación acerca de la aplicabilidad del MIP por sobre LR viene dada por la irregularidad de los tiempos de ejecución. Esto debido al comportamiento exponencial del tiempo en función de las dimensiones que requiere el método *branch-and-bound* para resolver problemas *np-completos*. Entonces, aunque los gráficos en Ilustración 4.22 e Ilustración 4.23 muestran valores absolutos (medidos en segundos) del tiempo de ejecución de ambas rutinas, la presente sección se concentrará además en investigar las estadísticas que puedan arrojar señales acerca de la dispersión y peores casos.

TIEMPO MEJOR SOLUCION



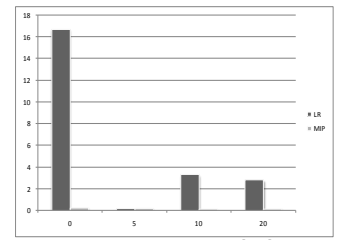
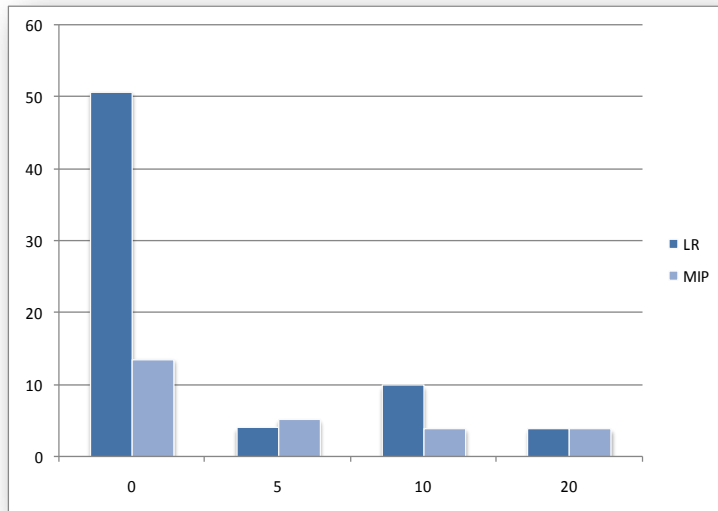
f=0.3



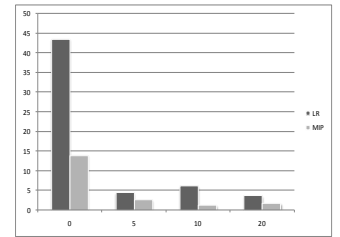
f=1.0

Ilustración 4.22. Promedio de tiempo de ejecución (seg) a mejor solución entera factible en función del coeficiente de adaptabilidad (f)

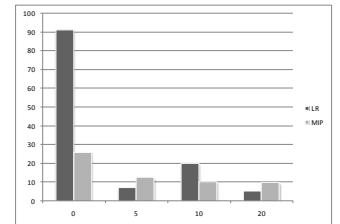
TIEMPO MEJOR SOLUCION



unidades=10



unidades=50



unidades=100

Ilustración 4.23. Promedio de tiempo de ejecución (seg) a mejor solución entera factible en función de la dimensión del problema (número de unidades)

Se puede observar que para los únicos casos para los cuales, en promedio, el algoritmo de relajación lagrangeana en la sección 3.2 logró encontrar la mejor solución entera en menor tiempo respecto al MIP, resulta los casos de mayores dimensiones (100 unidades). Es decir, sólo cuando los sistemas son grandes, existe un valor esperado de ejecución que puede superar en la búsqueda de soluciones factibles a la rutina MIP evaluada.

4.3.3 Aplicación al SING

Los ejemplos mostrados a continuación son una representación de las unidades generadoras disponibles en el SING [56]. El modelo incluirá tantas restricciones como sea posible para representar lo más cercano a la realidad de este sistema eléctrico. Sin embargo, no se pretende mostrar una posición respecto a la forma en que se realiza la operación real del sistema. El objetivo principal es demostrar la aplicación de la rutina en un sentido académico y continuar con la validación de la rutina de programación en la

sección anterior. Como se identificó adecuadamente la dependencia de los parámetros de sintonización, entonces se plantea realizar una estimación experimental de valores de la longitud del paso que permitan obtener soluciones factibles para brechas duales relativas pequeñas además de, por supuesto, realizar la comparación respecto a la rutina comercial MIP.

4.3.3.1 Características fundamentales del SING

El Sistema Eléctrico Interconectado del Norte Grande (SING), tiene una capacidad instalada total de 3596 MW, constituido en un 99.6 % por unidades térmicas y en un 0.4 % por unidades hidráulicas. Del total de unidades sólo dos funcionan usando energía hidráulica (13 MW), por lo que el parque generador se puede considerar netamente térmico. La topología del sistema podría definirse como poco enmallada, aunque tampoco longitudinal, con una barra principal de referencia de la cual se distribuyen las principales líneas del sistema de transmisión 220kV. El sistema equivalente del SING empleado tradicionalmente por el CDEC-SING comprende lo siguiente [56]:

- Un máximo de 322 barras, de las cuales del orden de 185 barras son principales.
- Un total de 107 líneas que constituyen 3.934 km de trazado en niveles de tensión que van desde 66 kV hasta 345 kV. Dentro de estas existen varias líneas de gran longitud (entre 150 a 200 km, y sólo una de 408 km).
- 178 transformadores de los cuales 153 son de dos enrolados y 25 de tres enrollados.
- 33 unidades generadoras las que constituyen 75 posibilidades de configuración que pueden ser utilizadas en la operación del SING.
- Exceso de capacidad instalada de generación. (Capacidad 3596 MW siendo la demanda máxima de registrada de 1897 MW⁶)
- Curvas de demanda constituida esencialmente por consumos de clientes libres (90%) asociados a empresas mineras y clientes industriales. El 10 % restante lo constituyen los consumos de clientes regulados de empresas de distribución que mantienen un crecimiento vegetativo.

⁶ Generación bruta, hora 23 del 21 de diciembre de 2008 [51].

- Durante el año 2006, por razones de estabilidad dinámica post-contingencia, era necesaria la aplicación de límites de generación según [56].

$$P_{i,t}^{\max} \begin{cases} 260 & \text{si } 18 \leq t \leq 24 \\ 240 & \text{demás} \end{cases} \quad (4.3)$$

4.3.3.2 Resultados de los parámetros de sintonización

Como se mencionó previamente, antes de incorporar un algoritmo de relajación lagrangeana, debe realizarse un proceso de sintonización de los parámetros de sintonización para la actualización de las variables duales. A continuación, se muestran resultados de las simulaciones del caso SING de acuerdo a los parámetros en [56] para la primera semana de enero de 2006 con la restricción de limitación de (3.1) activa y para diversos valores de la longitud de paso del subgradiente h .

Longitud del Paso h	Tiempo Total (seg)	Tiempo Dual (seg)	F. Obj (us\$)	Iteraciones	Tiempo Datos (seg)	Gap Dual (%)
1.5	298	22.9	628835.81	100	0.28	0.0297
2.5	212.9	15.5	618711.97	78	0.28	0.008
3.5	221.5	16.99	617473.85	80	0.28	0.00582
4.5	176.5	13.7	618098.61	62	0.28	0.00443
5.5	109.9	11.2	617573.85	38	0.28	0.00464
6.5	192.8	14.129	617573.85	68	0.28	0.00624
7.5	47.93	3.66	618098.61	17	0.28	0.00673
8.5	170.89	13.172	618711.97	59	0.28	0.0084
9.5	216.6	15.773	619153,94	76	0.28	0.0099

Tabla 4.23. Parámetros de sintonización pre-despacho SING

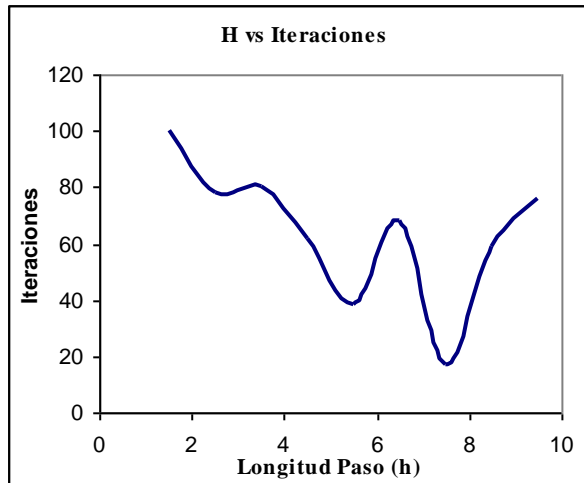


Ilustración 4.24. Parámetros de Sintonización para el pre-despacho del SING

Nótese que, a pesar de haberse escogido intervalos “pequeños” de variación de h , se obtuvieron dos mínimos locales. Resulta muy difícil entonces realizar sugerencias objetivas sobre la escogencia de este parámetro. Nótese además, que para el rango establecido de h el algoritmo convergió a tres soluciones diferentes.

4.3.3.3 Comparaciones con modelo comercial MIP

El objetivo de la presente sección es utilizar el algoritmo de LR sintonizado en 4.3.3.2 para aplicarlo a la solución de problemas típicos del SING y comparar los resultados respecto al mismo problema resultado mediante el optimizador comercial CPLEX [49]. Se debe señalar que las condiciones exactas de operación del SING para los períodos citados no son conocidas en detalle. En función de los resultados de los programas de generación diarios/semanales disponibles al público en [56], se plantearon los escenarios aquí analizados, asumiendo una disponibilidad de unidades, combustibles, potencias máximas de despacho y mantenimientos, acordes con los resultados publicados. Por consiguiente, tampoco es el objetivo comparar los resultados respecto de los obtenidos por el organismo en [56]. Sin embargo, ambos métodos cuentan con exactamente los mismos supuesto y, en consecuencia, los mismos datos operativos.

Los resultados mostrados a continuación corresponden a las simulaciones usando como referencia el caso real de operación del SING del día 15 de Abril de 2009 [56], considerando los supuestos antes mencionados. Los resultados comparativos de tiempos de ejecución, el valor de la función objetivo y gap se muestran en la Tabla 4.24.

Resultados	MIP	LR	Diferencia relativa de LR respecto MIP
Mejor Solución	1.4745E+06	1.4775E+06	0.20%
Mejor Cota inferior	1.4744E+06	1.4705E+06	-0.26%
Gap	0.009%	0.474%	0.46%
Tiempo mejor solución	0.56	1.34	139.29%

Tabla 4.24. Resultados de rendimiento Caso SING por ambos métodos

Estos resultados muestran una solución entera factible por LR que resulta en un costo de operación 0.2% mayor que la obtenida por MIP. La diferencia de los tiempos requeridos de ejecución, aunque notable en términos relativos, muestran que ambos métodos resolvieron en forma eficiente el problema operativo. Sin embargo, aunque la diferencia es tan solo de un 0.2%, vale la pena analizar por qué la solución es más costosa. Las respuestas se encuentran en el análisis detallado de la solución. La Ilustración 4.25 muestra el despacho obtenido por MIP, mostrando los valores ordenados por orden de mérito (orden de menor a mayor costo variable medio). En la base (zona inferior de la figura) se encuentran los generadores con mayor prioridad (menor costo de operación) y en el tope aquellos con menor prioridad (mayor costo de operación).

Generador	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DEUTZ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CUMMINS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TGIQ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TGTAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
U16-TG+U16-TV(Diesel)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ZOFRI_1-6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.204	0	0	0
SUIQ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1.207	0	0
TG1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TG2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ZOFRI_2-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	0
MIIQ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	1.5	0
M1AR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.811	3	3	2.809	1.463
M2AR	0	0	0	0	0	0	0.066	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3
GMAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.172	6	6	6	6	6
TG3(Diesel)	34.17	10	0	0	0	0	0	0	10	10	10	12.46	11.16	0	0	22.79	33.6	16.02	35	35	35	35	35	35
CTM3-TG+CTM3-TV(Diesel)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MAIQ	6	5	5.19	0	0	0	0	0	0	0	0	6	6	5	5	6	6	6	6	6	6	6	6	6
TG2B+0.5TV2C(Diesel)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5E-13	0
TG1A+TG1B+TV1C(Diesel)	332	330.5	332	326.4	323.5	331.9	332	325.2	304.2	308.5	330.9	332	332	314.7	329.9	332	332	332	332	332	332	332	332	332
U10	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35
U11	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35
MIMB	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
U12(Carbon)	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79	79
U13(Carbon)	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80
CTTAR(Carbon)	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140	140
U14(Carbon)	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127
U15(Carbon)	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121
CTM1(Carbon)	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154	154
CTM2(Carbon)	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163	163
NTO1(Carbon)	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127	127
NTO2(Carbon)	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131	131
TG11	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182
CAVA	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
CHAP	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Ilustración 4.25. Solución del pre-despacho del SING por ambos métodos

El código de colores en la figura, corresponde a la siguiente notación:

- Celdas en color gris claro indican periodos donde generadores son “despachados” por ambos métodos.
- Celdas en color gris oscuro indican periodos donde generadores son “despachados” solo por el método LR.
- Celdas en color rojo indican periodos donde generadores son “despachados” solo por el método MIP.
- Los valores numéricos corresponden al nivel (potencia en MW) para abastecer la demanda por MIP (escogido arbitrariamente por ser valores numéricos irrelevantes para el presente análisis).

La Ilustración 4.25 muestra que la diferencia fundamental entre ambas soluciones se centra en el despacho de la unidad generadora de ciclo combinado “TG2B+0.5TV2C(Diesel)”. Durante todo el período de evaluación, este generador es despacho por LR (celdas resaltas en color gris), mientras que MIP, en su lugar, encuentra una solución de menor costo despachando unidades de mayor costo de operación, pero

a su vez mayor utilización (nivel de generación en MW) de generadores de menor costo (ver detalles de ambas soluciones en Anexo F). En otras palabras, la solución por LR propone mayor “desplazamiento” de energía por generadores de menor costo (o base) para lograr introducir el bloque de energía por “TG2B+0.5TV2C(Diesel)”, la cual tiene como potencia mínima de operación un valor “elevado” (125MW). Por qué LR no explora la posibilidad de sustituir un generador de elevado mínimo técnico por “pequeños” generadores de mayor costo pero que permitan mayor uso de los generadores en base, lográndose de esta forma un menor costo global de operación? La respuesta viene por la fase de programación dinámica individual de generadores de LR. La decisión de despacho (on/off) de los generadores, está basada en la función lagrangeana desacoplada

$$\frac{\partial \text{Cop}_{i,t}}{\partial p_{i,t}} - \hat{\lambda}^t \cdot P_{i,t} = 0 \quad (4.4)$$

Si el costo marginal del generador ($\beta_{i,t}$) es menor que el multiplicador, es decir se cumple que $\beta_{i,t} \leq \hat{\lambda}_t$ entonces el generador debería ser despachado (ver detalle de metodología LR en Anexos A y 2. Nótese que $\hat{\lambda}_t$ es común para todos los generadores. Por ejemplo, para que un generador como “TG3(Diesel)” sea despachado, el valor del multiplicador debe ser mayor que $\hat{\lambda}_t \geq 68.94$. Si el valor del multiplicador, durante el proceso iterativo, alcanza este valor, la unidad “TG2B+0.5TV2C(Diesel)” será despachada también porque su costo ($\beta_{i,t}$) es menor ($\beta_{i,t} = 61.33 \leq \hat{\lambda}_t$). No hay entonces incentivo alguno retirar (no despachar) una unidad con menor costo que el valor del multiplicador. Esto reduce el espacio de exploración de LR, aun cuando se obtenga se maximice el dual, al no explorarse todas las combinaciones de unidades que puedan retirarse para un valor dado del multiplicador. La no-convexidad del UC, sugiere que entre estas combinaciones pueden encontrarse el óptimo global (o una solución de menor costo como se demuestra en el presente ejercicio). En la Ilustración 4.26 se muestra el despacho resaltando en color gris oscuro los periodos de tiempo en que tendrían que ser despachados los generadores si los multiplicadores alcanzaran los valores suficientes para lograr el despacho de las unidades más costosas obtenidas según el método MIP. Los valores en el tope de las columnas en la Ilustración 4.26

corresponden al valor de $\hat{\lambda}_t$ para lograr el despacho de las unidades obtenidas según el método MIP. En este caso y como fue explicado, LR no exploraría las combinaciones de despacho de las unidades y periodos en las celdas resaltas en color gris oscuro.

Generador	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DEUTZ																								
CUMMINS																								
TGIQ																								96.41
TGTAR																								8
U16-TG+U16-TV(Diesel)																							92.2	0
ZOFRI_1-6																							0.204	0
SUIQ																							4	1.207
TG1																							0	0
TG2																							0	0
ZOFRI_2-5																							5	5
MIIQ																							81.34	3
M1AR																							3	3
M2AR																							2.811	3
M3AR																							3	3
GMAR																							3	3
TG3(Diesel)																							80.45	3
CTM3-TG+CTM3-TV(Diesel)																							3	3
MAIQ																							6	6
TG2B+0.5TV2C(Diesel)																							6	6
TG1A+TG1B+TV1C(Diesel)																							6	6
U10																							6	6
U11																							6	6
MIMB																							6	6
U12(Carbon)																							6	6
U13(Carbon)																							6	6
CTTAR(Carbon)																							6	6
U14(Carbon)																							6	6
U15(Carbon)																							6	6
CTM1(Carbon)																							6	6
CTM2(Carbon)																							6	6
NTO1(Carbon)																							6	6
NTO2(Carbon)																							6	6
TG11																							6	6
CAVA																							6	6
CHAP																							6	6

Ilustración 4.26. Predespacho del SING mostrando el espacio de soluciones no explorado por LR

4.3.3.4 Relación entre las potencias mínimas de operación y la calidad de la solución por relajación lagrangeana

Esta sección presenta los resultados de sensibilidad de los tiempos de ejecución en función de las potencias mínimas de operación. Dada la formulación del problema de UC simplificado presentado en la sección 2.1, las restricciones de potencia mínima son directamente responsables de las condiciones de integralidad. Siendo esta restricción determinante en el planteamiento del UC, se desea entonces investigar la influencia que tienen los valores de potencia mínima en la solución del problema empleando los métodos de solución LR y MIP.

Para realizar la sensibilidad, se utilizó como referencia el caso real de operación del SING del día 15 de Abril de 2009 [56]. El parámetro de sensibilidad " ω " se utiliza como valor de ajuste de la potencia

mínima de operación en función de la capacidad máxima de generación y pertenece al rango mostrado en la ecuación (4.5).

$$P_{min} = \omega * P_{max} \quad 0 \leq \omega \leq 1 \quad (4.5)$$

De esta forma, se tomaron 5 valores (en forma arbitraria) contenidos en el rango de “ ω ” anteriormente definido. Los valores de “ ω ” y los resultados de: tiempos de ejecución y calidad de la solución (medidos con el valor de la función objetivo final y “gap”); son mostrados en la Tabla 4.25 y Tabla 4.26 para MIP y LR, respectivamente.

Caso	ω	Pmin	MIP (seg)	LR (seg)
1	0	0%	0.02	0.003
2	0.5	50%	0.3	0.667
3	0.7	70%	0.39	0.625
4	0.85	85%	0.48	0.673
5	0.9	90%	0.47	0.651

Tabla 4.25. Tiempos de ejecución en función de la Potencia Mínima de generación (Caso SING)

Caso	ω	Pmin	MIP (%)	LR (%)
1	0	0%	0.00%	0.00%
2	0.5	50%	0.01%	0.20%
3	0.7	70%	0.01%	0.38%
4	0.85	85%	0.01%	3.08%
5	0.9	90%	0.01%	4.70%

Tabla 4.26. Calidad de la solución en función de la Potencia Mínima de generación (Caso SING)

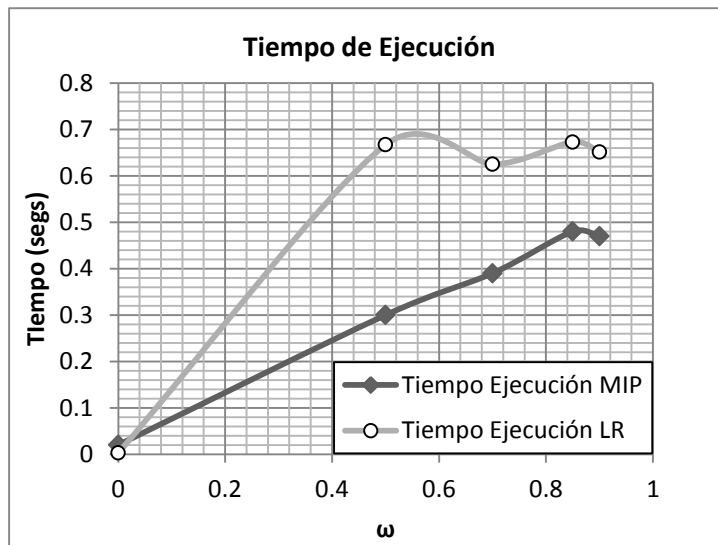


Ilustración 4.27. Tiempo de ejecución (segundos) en función del parámetro de sensibilidad “ ω ”

(Potencia mínima de generación como “porcentaje” de la potencia máxima)

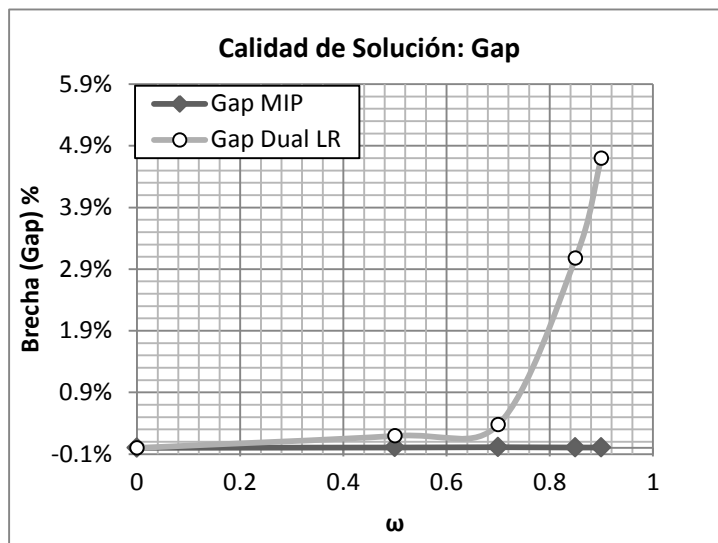


Ilustración 4.28. Brecha (o gap, medido en %) en función del parámetro de sensibilidad “ ω ” (Potencia

mínima de generación como “porcentaje” de la potencia máxima)

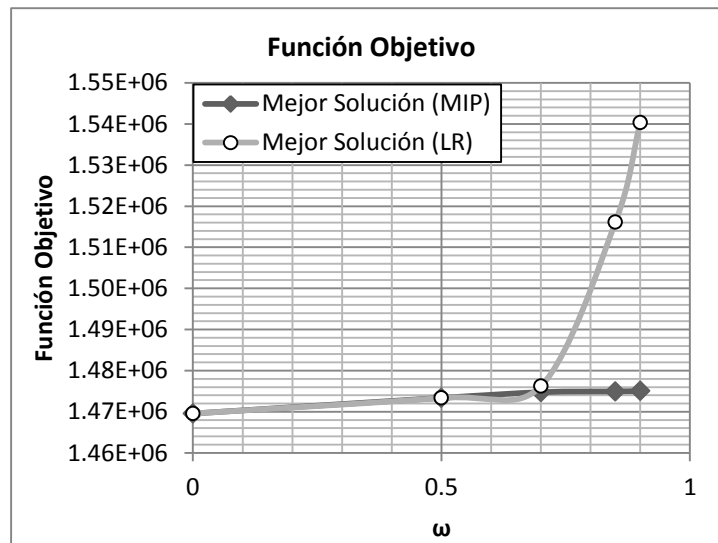


Ilustración 4.29. Función objetivo final (o mejor solución entera) en función del parámetro de sensibilidad “ ω ” (Potencia mínima de generación como “porcentaje” de la potencia máxima)

Los resultados gráficos en Ilustración 4.27, Ilustración 4.28 e Ilustración 4.29, permiten formular el siguiente análisis de resultados:

1. Si las unidades generadoras poseen (en global) un margen estrecho de operación (valor de w cercano a 1 en presente experimento), el problema de UC, juzgando por los tiempos de ejecución, resulta “más difícil” de resolver. Se señala que afecta inclusive al MIP como se muestra en la Ilustración 4.27. Se observa que el método MIP requiere un tiempo de ejecución de 30% y hasta un 60% mayor al resolver problemas donde el margen de operación ($P_{max}-P_{min}$) es menor (valores desde $\omega=[0.7-0.9]$) respecto al tiempo de ejecución para problemas con un margen de operación del 50% ($\omega=0.5$).
2. La brecha dual “inconciliable” con el método LR aumenta a medida que se estrecha el margen de operación (aumenta valor de ω). Esto se evidencia en la tendencia constante-creciente observada por LR en la Ilustración 4.28, incluso cuando se usó un amplio margen de iteraciones (500). Los valores alcanzan un “peak” máximo de 4.7% para $\omega = 0.9$, lo cual cuestiona la calidad de la solución al deteriorarse las pruebas de optimalidad (posible sobre costo teórico del 5%). Esta conclusión resulta de relevancia para un sistema eléctrico como el SING donde, en

promedio, las potencias mínimas operación se considera “elevado” dada las características del parque generador, conforme lo descrito en la sección 4.3.3.1. Para el caso en análisis, el promedio ponderado de las potencias mínimas (como porcentaje de la potencia máxima de despacho) supera el 50% (55.1% [56]).

3. Dado que se dispone de la solución entera factible mediante el método MIP, se determina además con certeza que el valor del gap dual elevado no corresponde a un “fallo” de LR en la maximización del dual, sino que la solución factible encontrada resulta hasta 4.43% mayor (más costosa), para el caso más extremo ($\omega = 0.9$), que “otra” solución factible (encontrada por el método MIP, ver Tabla 4.27). Nótese además que el gap fue del 4.7% (ver Tabla 4.26), clara indicación de que la brecha dual es ocasionada por la pobre calidad de la solución entera factible encontrada por LR.

Caso	ω	Mejor Solución (MIP)	Mejor Solución (LR)	Diferencia (MIP-LR en %)
1	0	1.470E+06	1.470E+06	0.00%
2	0.5	1.473E+06	1.473E+06	0.00%
3	0.7	1.475E+06	1.476E+06	0.10%
4	0.85	1.475E+06	1.516E+06	2.79%
5	0.9	1.475E+06	1.540E+06	4.43%

Tabla 4.27. Comparación de la mejor solución entera factible MIP vs. LR en función de la Potencia

Mínima de generación (Caso SING)

4.3.3.5 Recomendaciones, sugerencias y futuros desarrollos

En vista de los resultados obtenidos, existen evidencias prácticas que muestran que el método MIP resulta más efectivo para resolver el problema de UC en sistemas medianos. Resulta importante diferenciar con sistemas de dimensiones muy grandes, como CAISO, MISO, UCTE, etc, donde el crecimiento exponencial del tiempo y los requerimientos de memoria para resolver un problema único (sin descomposición) permanecen siendo un problema irresuelto.

Sin embargo, LR aún puede ser empleado para obtener soluciones al UC (aunque no necesariamente para sistemas medianos). Las siguientes sugerencias relacionadas con LR se realizan en función de la experiencia obtenida con las simulaciones de prueba presentadas previamente. Se advierte sin embargo, que los procedimientos aquí presentados no son de ninguna forma genéricos y que para cualquier variación de disponibilidades, demanda, reserva y otros, requieren hacer probablemente reajustes de sintonización.

- Para el **método de paso constante (soluciones factibles)**: Usar un paso de longitud que en ningún caso supere el rango $[0 - \lambda_{\text{esperado}}]$, donde $\lambda_{\text{esperado}}$ puede corresponder al valor esperado del multiplicador de la restricción de potencia (i.e. costo linealizado de la unidad marginal). Para el SING con las restricciones activas analizadas se recomienda utilizar parámetros en el rango 1.5-9.5 para facilitar la obtención de soluciones factibles.
- Para el **método de diferencias constantes (búsqueda directa del óptimo)**: Usar valores sugeridos en la bibliografía preferiblemente debido a que su relación es aún menos directa. Por ejemplo, se recomienda revisar la sugerencia en [15].
- **Para el lagrangeano aumentado**: Emplear valores de $c < 1$ que no afecten notablemente la forma de la curva. Es fundamental que el segundo parámetro ε cumplan siempre con: $c \leq \frac{1}{2\varepsilon}$ y $\varepsilon > 0$.
- **Sugerencias específicas de SING**: Los mejores resultados se obtuvieron para valores de paso en el rango $[1.5 - 9.5]$. La relajación lagrangeana aumentada para valores escogidos de c en el rango $[0.02 - 0.001]$ y ε tal que se cumpla con $c \leq \frac{1}{2\varepsilon}$, mostró tener desempeños muy similares. Sin embargo, se recomienda su uso y los rangos propuestos, por la estabilidad que aporta a la convergencia del método.

4.4 Resumen

En el presente capítulo se mostraron los resultados numéricos de las simulaciones realizadas, así como el análisis de rendimiento de las referencias citadas. Los resultados muestran que:

- La revisión bibliográfica, con los factores de normalización, muestran un claro mejor desempeño de los métodos matemáticos por sobre los heurísticos.
- Que para los casos sintéticos se pudo sintonizar LR para producir el óptimo global de los problemas pequeños. Misma solución obtenida por MIP en un tiempo de ejecución similar.
- La comparación estadística en cambio no deja dudas que MIP, para sistemas medianos es superior en cuanto al tiempo de ejecución (menor tiempo para solución factible), obteniendo soluciones.
- El caso real del SING muestra también un mejor desempeño de MIP sobre LR y, expone además que la capacidad de producir soluciones factibles de LR es especialmente susceptible cuando se reduce el margen de operación (definido como la diferencia entre P_{min} y P_{max}). Aunque LR es aplicable al SING, la mejor solución factible puede resultar hasta 4.43% más costosa que MIP.

Capítulo 5

Discusión y Conclusiones

La presente tesis mostró las comparaciones prácticas entre los métodos de programación entero-mixta (MIP) y relajación lagrangeana (LR) para resolver el problema del Predespacho. Los principales aportes son:

- Plantear y aplicar por primera vez un método de generación de instancias aleatorias y analizar el desempeño de LR y MIP en una plataforma computacional uniforme.
- Revisar resultados numéricos en la bibliografía especializada y hacer uso de benchmark computacionales para normalizar los tiempos de ejecución, de tal forma que se puedan comparar en forma más justa la efectividad de diversos métodos matemáticos y metaheurísticos para resolver el UC.
- Exponer la debilidad de LR para cerrar el Gap Dual y obtener soluciones de calidad cuando los sistemas poseen generadores térmicos con margen estrecho de regulación (diferencia entre p_{min} y p_{max}).
- Aplicar y comparar la efectividad de LR y MIP para resolver el Predespacho del sistema chileno SING.

La forma de realizar las comparaciones de UC se basaron en tres (3) métodos de comparación de algoritmos de optimización: 1) histórico (bibliográfico) 2) Sintético-aleatorio y 3) Práctico aplicado a un sistema real (SING). Los resultados mostrados permiten concluir que el método de solución MIP resulta más efectivo (menor valor esperado de los tiempo de ejecución, mayor calidad de solución a juzgar por el menor gap, mayor efectividad para encontrar soluciones factibles) para resolver problemas de UC de medianas dimensiones que el tradicional método LR.

La comparación bibliográfica de resultados mostró que ninguna de las diversas metodologías logró una mejora superior al 1% (juzgando por el valor de la función objetivo) respecto a LR. En este primer método de comparación, LR probó ser el método más eficiente para resolver el UC. Mejorar tan solo un 0.7%, la solución que ésta entrega puede costar, de acuerdo al resultado para las dimensiones 38 unidades, hasta 3000% del tiempo de ejecución si se intenta con otra metodología. Sin embargo, este primer análisis no incluyó resultados por MIP comparables, con la excepción del método combinado *branch-and-bound* y constraint logic programming (CLP) [25] que demostró un desempeño muy similar a LR.

Los resultados obtenidos para los casos sintéticos sencillos por el modelo LR, desarrollado en el contexto del presente trabajo y detallado en Capítulo 3, fueron comparados con los obtenidos por el programa comercial de optimización (CPLEX) basado en MIP. Los tiempos de ejecución fueron similares (diferencias en el orden de 0-8 segs) para problemas de dimensiones reducidas (10 unidades), pero LR confirmó ser más eficiente para sistemas de mayor escala (100 unidades) en este ejercicio (más de 700 segundos). Se probó además que ambos obtuvieron el óptimo global del problema básico de pre-despacho (10 unidades – 24hrs) lo que sirvió como validación de la rutina LR desarrollada. Entre las distintas pruebas realizadas a casos sintéticos resueltos mediante LR se cuentan: solución de ejemplos de 10 hasta 100 unidades, períodos de tiempo en el rango 4-168 hrs, restricciones de reserva, rampa y límite común de generación. Todas requirieron de ajustes del parámetro fundamental de sintonización del subgradiente “h”. LR tuvo dificultad para resolver problemas con un número elevado de períodos de tiempo (1 semana ó 168 períodos), requiriendo hasta 30 veces mayor tiempo de ejecución respecto al problema diario (7 veces menor). Éste último resultado se considera un aporte del presente trabajo considerando que la bibliografía consultada se centra en la comparación del UC diario (24 períodos).

En cuanto al análisis estadístico de ambos métodos, el método LR encontró soluciones factibles para 429 instancias de las 480 (89,4%), cumpliendo con el gap objetivo de 0,01% en tan sólo 90 casos (18,8%), mientras que el MIP encontró la solución óptima para el 98,3% de los casos. Los resultados obtenidos

verifican que para el rango de 10 a 100 unidades, la mejor solución entera factible que encuentra LR tiene una menor calidad (valor mayor de la función objetivo en problema de minimización) que la encontrada por el optimizador MIP. Los resultados permitieron determinar que el sobre costo esperado de las soluciones mediante LR fue de 6,03%. El aumento de la dispersión en los parámetros técnicos beneficia ambos métodos al obtener la mejor solución factible en menores tiempos de ejecución. De esta forma se muestra que no sólo la calidad de la solución por LR se ve afectada por unidades similares, sino que también afecta el rendimiento del método MIP. Se comprueba que a medida que los problemas son más difíciles (juzgando por el coeficiente de adaptabilidad y por el número de unidades), la cota inferior de LR resulta en promedio mayor (mejor) a la cota inferior promedio por MIP.

Finalmente, se demostró que, para casos reales de sistemas medianos (SING), distintos valores del parámetro de sintonización llevó a la relajación (LR) a converger a soluciones diferentes. Este problema surge como uno de los mayores retos de implementación con este método. Como nota positiva en favor de LR, se recuerda que la variación del parámetro de sintonización "h" en el rango 1.5-9.5 siempre generó soluciones factibles en un número muy reducido de iteraciones (menor a 100). La calidad de la solución del MIP resultó apenas mejor que LR (función objetivo 2% menor que LR). El tiempo de ejecución de LR resultó 139.3% mayor que el tiempo requerido por MIP (aunque ambos menor a 1.34 segundos). Sin embargo, el problema de calidad de la solución puede ser mucho más profundo dado que, por definición de la metodología LR, no es posible explorar combinaciones que involucren el despacho de unidades de mayor costo y evitar despachar unidades que "desplacen" energía de generadores base (de menor costo). Esto resultó evidente al analizar la calidad de la solución LR cuando las restricciones de mínimos técnicos de las unidades obligan que la solución óptima entera se aleje de la relajación lineal. Si bien para el caso SING la resolución mediante MIP también se dificulta (experimentando un aumento entre 30% y 60%), este efecto es comparativamente menor al deterioro de la calidad de la solución observada mediante LR que alcanzó soluciones de hasta 5% mayor costo de generación. Conceptualmente, este hallazgo se explica porque los valores de actualización de los

multiplicadores en LR actúan como “precios” a los que los generadores responden individualmente, sin observar que existen combinaciones entre todos (para ese mismo precio) suficientes para abastecer la demanda, lo que se traduce en imposibilidad para explorar zonas del espacio de soluciones factibles. Esto se agudiza especialmente cuando se introducen “grandes” bloques de energía debido a elevados mínimos técnicos. Este problema no es evidente y no se encuentra señalado en la bibliografía revisada, por lo que se considera una de las contribuciones del presente trabajo.

Futuros desarrollos se concentrarán en la integración LR con MIP. LR podría aportar las cotas inferiores que, como se demostró, resultan mejores (mayores) que la simple relajación lineal empleada por el método MIP. Adicionalmente, el método LR puede entregar soluciones enteras las cuales podrían inicializar el método MIP y, en combinación con heurísticas simples como *local search* aprovechar las señales de las primeras soluciones enteras (no necesariamente factibles) para generar soluciones factibles de calidad en el mismo nodo raíz y, adicionalmente, orientar el proceso de “*branching*” si se requiere explorar el árbol del *branch-and-bound*. De esta forma, LR reforzaría el método MIP en ambos extremos: con generación de soluciones factibles de calidad y aportando una mejor cota inferior.

Capítulo 6

Bibliografía

- [1] R. Baldick, "The Generalized Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 10, pp. 465-475, 1995.
- [2] P. N. Prasad, "Unit Commitment – A Bibliographical Survey," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1196-1205, Mayo 2004.
- [3] F. Leanez and R. Palma-Benhke, "Comparación de Rendimiento de Metodologías para el Predespacho," *Anales del Instituto de Ingenieros*, Abril 2004.
- [4] A. r. p. b. t. R. T. S. T. F. o. t. a. o. p. m. Subcommittee, "The IEEE Reliability Test System - 1996"," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1010-1020, Agosto 1999.
- [5] D. Streiffert, R. Philbrick and A. Ott, "A Mixed Integer Programming Solution for Market Clearing and Reliability Analysis," in *Power Engineering Society General Meeting*, 2005.
- [6] H. Pinto, F. Magnago, S. Brignone and O. Alsac, "Security Constrained Unit Commitment: Network Modeling and Solution Issues," in *Power Systems Conference and Exposition. PSCE 06.*, 2006.
- [7] M. L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science (pre-1986)*, no. 27, Enero 1981.
- [8] F. Zhuang and F. D. Galiana, "Towards a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 763-773.
- [9] H. Sasaki, M. Watanabe, J. Kubokawa, N. Yorino and R. Yokoyama, "A Solution Method of Unit Commitment by Artificial Neural Networks," *IEEE Transactions on Power Systems*, vol. 7, no. 3, Agosto 1992.

- [10] C. Benavides, I. Alarcón, D. Espinoza, F. Leanez and R. Palma-Behnke, "Modelo de Predespacho Orientado a Sistemas Medianos con Funciones de Costo no Convexas y Rampas de Arranque y Parada," *Anales del Instituto de Ingenieros*, Agosto 2008.
- [11] G. Morales-Espana, J. Latorre and A. Ramos, "Tight and Compact MILP Formulation for the Thermal Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1-12, 2013.
- [12] G. Morales-Espana, J. Latorre and A. Ramos, "Tight and Compact MILP Formulation of Start-Up and Shut-Down Ramping in Unit Commitment," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1288-1296, 2013.
- [13] M. Anjos and A. Vannelli, "Tight Mixed Integer Linear Programming Formulations for the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39-46, 2012.
- [14] L. Wu, "A Tighter Piecewise Linear Approximation of Quadratic Cost Curves for Unit Commitment Problems," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2581-2583, 2011.
- [15] S. Dekarjangpetch, G. B. Sheblé and A. J. Conejo, "Auction Implementation Problems Using Lagrangian Relaxation," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 82-88, Febrero 1999.
- [16] M. Carrión and J. M. Arroyo, "A Computationally Efficient Mixed-Integer Linear Formulation for the Thermal Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 21, p. 1371, Agosto 2006.
- [17] A. Kazarlis, A. G. Bakirtzis and V. Petridis, "A Genetic Algorithm Solution to the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 11, pp. 83-92, Febrero 1996.
- [18] G. B. Sheblé and G. N. Fahd, "Unit Commitment Literature Synopsis," *IEEE Transactions on Power Systems*, vol. 9, no. 1, Febrero 1994.
- [19] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*, Segunda edición ed., NY: Wiley Interscience, 1984, 1996.

- [20] T. Senjyu, K. Shimaburu, K. Uezato and T. Funabashi., "A Fast Technique for Unit Commitment Problem by Extended Priority List," *IEEE Transactions on Power Systems*, vol. 18, no. 2, Mayo 2003.
- [21] M. Madrigal and V. H. Quintana, "An Interior-Point/Cutting-Plane Method to Solve Unit Commitment Problems," *IEEE transactions on Power Systems*, vol. 15, no. 10, Agosto 2000.
- [22] C.-P. Cheng, C.-W. Liu and C.-C. Liu, "Unit Commitment by Lagrangian Relaxation and Genetic Algorithms," *IEEE Transactions On Power Systems*, vol. 15, no. 10, Mayo 200.
- [23] S. Wang, M. Shahidehpour, D. Kirschen and S. Mokhtari, "Short-Term Generation Scheduling with Transmission and Enviromental Constraints using an Augmented Lagrangian Relaxation," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1294-1301, Agosto 1995.
- [24] C. Beltran and F. J. Heredia, "Unit Commitment by Augmented Lagrangean Relaxation: Testing Two Decomposition Methods," *Journal of Optimization Theory and Applications archive*, vol. 112, no. 2, pp. 295-314, Febrero 2002.
- [25] C.-A. Li, R. B. Johnson and A. J. Svoboda, "A New Unit Commitment Method," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 113-119, Febrero 1997.
- [26] C. L. Tseng, S. S. Oren, A. J. Svoboda and R. B. Johns, "A Unit Decommitment Method in Power System Scheduling," *Electrical Power & Energy System*, vol. 19, no. 6, pp. 357-365, 1997.
- [27] W. Ongsakul and N. Petcharak, "Unit Commitment by Enhanced Adaptive Lagrangian Relaxation," *IEEE Transactions on Power Systems*, vol. 19, no. 1, Febrero 2004.
- [28] M. Shahidepour, H. Yamin and Z. Li, *Market Operations In Electric Power Systems*, NY: Wiley-Interscience, 2002.
- [29] C.-L. Tseng, S. S. Oren, C. S. Cheng, C.-A. Li, A. J. Svoboda and R. B. Johnson, "A Transmission-Constrained Unit Commitment Method," in *System Sciences. Proceedings of the Thirty-First Hawaii International Conference*, 1998.

- [30] J. Shaw, "A Direct Method for Security-Constrained Unit Commitment," *IEEE Transactions on Power Systems*, vol. 10, no. 3, Agosto 1995.
- [31] K. Abdul-Rahman, M. Shahidehpour, M. Argangic and S. Mokhtari, "A Practical Resource Scheduling with OPF Constraints," *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 254-259, Febrero 1996.
- [32] C. Murillo-Sánchez and R. J. Thomas, "Thermal Unit Commitment Including Optimal AC Power Flow Constraints," in *Proceedings of the 31st Hawaii International Conference on System Sciences*, 1998.
- [33] R. Gollmer, A. Moller, M. P. Nowak, R. W. and R. Schultz, "Primal and Dual Methods for Unit Commitment in a Hydro-Thermal Power System," in *Proceedings of the 13th Power Systems Computation Conference*, Trondheim, 1999.
- [34] K.-Y. Huang, H.-T. Yang and C.-L. Huang, "A New Thermal Unit Commitment Approach Using Constraint Logic Programming," *IEEE Transactions on Power Systems*, vol. 13, no. 3, Agosto 1998.
- [35] A. Kazarlis, A. G. Bakirtzis and V. Petridis, "A Genetic Algorithm Solution to the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 11, pp. 83-92, Febrero 1996.
- [36] A. Kazarlis, A. G. Bakirtzis and V. Petridis, "A Solution to the Unit-Commitment Problem Using Integer-Coded Genetic Algorithm," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 83-92, Mayo 2004.
- [37] K. A. Juste, K. E. Tanaka and J. Hasegawa, "An Evolutionary Programming Solution to the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 14, no. 4, Noviembre 2000.
- [38] S. Saneifard, N. R. Prasad and H. A. Smolleck, "A Fuzzy Logic Approach to Unit Commitment," *IEEE Transactions on Power Systems*, vol. 12, no. 2, Noviembre 1997.
- [39] S. Su and Y.-Y. Hsu, "Fuzzy Dynamic Programming: An Application to Unit Commitment," *IEEE Transactions on Power Systems*, vol. 6, no. 3, Agosto 1991.

- [40] T. Senjyu, H. Yamshiro, K. Uezato and T. Funabashi, "A Unit Commitment Problem by Using Genetic Algorithm Based on Characteristic Classification," in *Proceedings of IEEE/PES Winter Meeting*, 2002.
- [41] S.-J. Huang and C.-L. Huang, "Application of Genetic-Based Neural Networks to Thermal Unit Commitment," *IEEE Transactions on Power Systems*, vol. 12, no. 2, Mayo 1997.
- [42] F. Zhuang and F. D. Galiana, "Unit Commitment by Simmulated Annealing," *IEEE Transactions on Power Systems*, vol. 5, no. 1, pp. 311-317, 1990.
- [43] A. H. Mantawy, Y. Abdel-Magib and S. Z. Selim, "A Simulated Annealing Algoritm for Unit Commitment," *IEEE Transactions on Power Systems*, vol. 13, no. 1, Febrero 1998.
- [44] A. H. Mantawy, Y. Abdel-Magib and S. Z. Selim, "Integrating Genetic Algorithm, Tabu Search and Simulated Annealing for the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 14, no. 3, Agosto 1999.
- [45] J. Ding, Y.-Z. Lu and J. Chu, "Extremal Optimization for Unit Commitment Problem for Power Systems," in *IEEE Power and Energy Society General Meeting*, San Diego, 2012.
- [46] C. Christober Asir Rajan and M. R. Mohan, "An Evolutionary Programming-Based Tabu Search Method for Solving the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 19, no. 1, Febrero 2004.
- [47] V. Tsurkirov, *Large-Scale Optimization - Problems and Methods*, Luwer Academic Publishers, 2003.
- [48] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, NJ: Wiley Interscience, 1999.
- [49] ILOG, "CPLEX Interactive Optimizer 7.0. Reference Manual, Part III," 2001. [Online]. Available: <http://www.ilog.com>. [Accessed 2006].
- [50] SPEC, "Sitio de referencia SPEC®:," [Online]. Available: <http://www.spec.org/>. [Accessed Enero 2005].

- [51] J. J. Degarra, "Technical Report: Performance of Various Computers Using Standard Linear Equations Software," Tennessee, 2004.
- [52] J. Dongarra, J. Bunch, C. Moller and G. W. Steward, "LINPACK User's Guide," 1979.
- [53] R. P. Weiker, "Dhrystone Benchmark: Rationale for Version 2 and Measurement Rules," *SIGPLAN Notices*, vol. 23, pp. 49-62, 8 Agosto 1988.
- [54] H. J. Curnow and B. A. Wichman, "A Synthetic Benchmark," *Computer Journal*, vol. 19, no. 1, Febrero 1976.
- [55] K. S. Swarup and S. Yamashiro, "Unit Commitment Solution Methodology Using Genetic Algorithm," *IEEE Transactions on Power Systems*, vol. 17, no. 1, Febrero 2002.
- [56] D. d. O. y. Peajes, "Página oficial web CDEC_SING," Enero 2009. [Online]. Available: <http://www.cdec-sing.cl>. [Accessed Enero 2009].
- [57] F. Leanez and R. Palma-Behnke, "Towards More Accurate Unit Commitment Performance Comparisons," in *Transmission & Distribution Conference and Exposition: Latin America, 2006*, 2006.
- [58] R. S. Barr, B. L. Golden and J. Kelly, "Guidelines for Designing and reporting on Computational Experiments with Heuristic Methods," in *Proceedings of International conference on Metaheuristics for Optimization*, 2001.
- [59] R. R. Hill, "An Analytical Comparison of Optimization Problem Generation Methodologies," in *Simulation Conference Proceedings*, 1998.
- [60] R. L. Rardin and R. Uzsoy, "Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial," *Journal of Heuristics*, vol. 7, no. 3, pp. 261-304, Mayo 2001.
- [61] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, Abril 1997.

- [62] E. B. Woodruff, H. B. Lammers and T. F. Lammers, Steam Plant Operation, Octava Edición ed., 2004: McGraw-Hill Professional.
- [63] R. L. Mason, R. F. Gunst and J. L. Hess, Statistical Design and Analysis of Experiments, Segunda edición ed., Hoboken, New Jersey: John Wiley & Sons, Inc., 2002.
- [64] F. Leanez, "Powertools," 2008. [Online]. Available: <http://powertools.awardspace.com/tools/>. [Accessed 1 Abril 2013].

Capítulo 7

ANEXOS

7.1 ANEXO A - Relajación Lagrangeana: Teoremas de Dualidad

7.1.1 Introducción

Es uno de los métodos más conocidos para resolver problemas de optimización de gran escala. Se suele aplicar a problemas que por sus condiciones permite clasificar las restricciones en dos tipos: sencillas y complejas (en el contexto UC se denominan “individuales” y de “acoplamiento”, respectivamente). La idea principal consiste en resolver problemas de optimización sujeto a las restricciones sencillas, “olvidando” o como se suele llamar “relajando”, las restricciones más complejas o complicadas.

7.1.2 Optimización Dual [1]

Existe quizás una confusión alrededor de los conceptos de los teoremas de dualidad. Se pueden citar varios casos particulares de dualidad que dependen de las condiciones del problema, determinados tanto por las características de la función objetivo como del conjunto de restricciones. Por este motivo se presenta una aclaratoria haciendo referencia a los teoremas de dualidad que interesan para problemas de las características del pre-despacho. El problema de optimización que en adelante se utilizará tiene la siguiente forma:

$$\begin{aligned} \max: & f(x) \\ \text{s.a:} & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \tag{A.1}$$

7.1.2.1 Definición

Se le llama dual al artificio matemático que cumple con la propiedad que su función objetivo está limitada por el problema original (denominado primal). Supongamos que el dual se corresponde con la función:

$$\begin{aligned} \min: & f(y) \\ \text{s.a:} & y \in Y \end{aligned} \tag{A.2}$$

Entonces, $f(y) \geq f(x)$ para todos los $x \in X$ y para todos los $y \in Y$. Esto implica que si el primal es factible, el dual no puede ser ilimitado y viceversa, si el dual es factible, el primal está acotado. El dual suele proveer pruebas de suficiencia para determinar el óptimo. Existen particularidades acerca de estas pruebas que serán tratadas individualmente.

7.1.2.2 Dual de una Función Lagrangeana

En general, se puede definir como sigue: si existe un valor factible x y y pueden ser calculados de tal forma que $f(x)=f(y)$, entonces se dice que x es la solución del primal y y es la solución óptima del dual. Si solo provee cotas, se dice que la dualidad es débil. En detalle, el dual del lagrangeano resulta:

$$\begin{aligned} \min: & L^*(u, v) \\ \text{s.a:} & u \geq 0 \end{aligned} \tag{A.3}$$

donde: $L^*(u, v) = \sup \{f(x) + u \cdot g(x) + v \cdot h(x)\} | x \in X$

Este teorema llevó a la unificación de las teorías previas acerca de la dualidad. A continuación se presentan los tres principales teoremas de dualidad.

7.1.3 Teoremas de Dualidad de la Función Lagrangeana

7.1.3.1 Teorema de Dualidad Débil

El lagrangeano dual provee la siguiente cota superior al óptimo de la función objetivo parametrizada, $f^*(b, c)$ ⁷:

⁷ $f^*(b, c) = \text{Sup}\{f(x) : x \text{ in } X, g(x) \leq b, h(x) = c\}$.

$$f^*(b, c) \leq L^*(u, v) + ub + vc \quad (\text{A.4})$$

Más aún, si definimos:

$$x^* = \arg \max \{L(x, u, v)\} | x \in X \quad (\text{A.5})$$

entonces,

$$x^* = \arg \max \{f(x)\} | g(x) \leq b \text{ y } h(x) = h(x^*) \quad (\text{A.6})$$

7.1.3.2 Teorema de Dualidad Fuerte

Se dice que el lagrangeano dual anterior es un dual fuerte para un problema convexo que satisface las condiciones de interioridad de Slater. Esto es, si x^* es la solución óptima, entonces deben existir un conjunto de multiplicadores u^*, v^* tal que $f^* = L^*(u^*, v^*)$.

7.1.3.3 Teorema de equivalencia de un punto Saddle:⁸

Los valores $x^*, u^*, v^* \in \mathfrak{R}^n, \mathfrak{R}^m$, son un punto saddle, si y solo si se cumplen las siguientes condiciones de dualidad fuerte:

x^* está contenido en: $\arg \max \{L(x, u^*, v^*) : x \in X\}$

x^* es factible. Es decir: $g(x^*) \leq 0$ y $h(x^*) = 0$

$u^*g(x^*) \leq 0$

7.1.4 Dual en Programación Lineal (LP) [2]

Dada la siguiente forma general de un problema de optimización:

$$\begin{aligned} \min: & \quad c \cdot x \\ \text{s.a:} & \quad A_1 x = b_1 \\ & \quad A_2 x \leq b_2 \\ & \quad x_i \geq 0 \end{aligned} \quad (\text{A.7})$$

Su correspondiente dual sería:

⁸ Saddle: Punto de inflexión o Punto silla. Suele asociarse con puntos de una función continua donde $\frac{\partial f(x)}{\partial x} = 0$

$$\begin{aligned}
\max: & \quad \pi \cdot b \\
\text{s.a:} & \quad \pi_i \geq 0 \\
& \quad \pi \cdot A_1 = c_1 \\
& \quad \pi \cdot A_2 \leq c_2
\end{aligned}
\tag{A.8}$$

7.1.4.1 Teorema de dualidad en programación lineal

Para un problema LP primal-dual, exactamente alguna de las siguientes condiciones se cumple⁹:

- El primal tiene una solución óptima y, por tanto, el dual también y sus funciones objetivos son iguales.
- El primal es infactible, por lo tanto el dual es infactible (no-acotado)
- El primal es no-acotado, por lo tanto el dual es infactible.

7.1.5 Referencias

- [1] Vladimir Tsurkiov, "Large-Scale Optimization - Problems and Methods", Applied Optimization series. Luwer Academic Publishers. Sept, 2003.
- [2] Christos H. Papadimitriou, Kenneth Steiglitz, "Combinatorial Optimization - Algorithm and Complexity", Dover Publications Inc. NY, 1998.

⁹ Condiciones de Khun-Tucker

7.2 ANEXO B – Programación Dinámica Individual de Unidades

7.2.1 Introducción

La programación dinámica es una técnica pionera en solución de problemas de optimización. Ha sido empleada para resolver una vasta gama de problemas operativos. El método ha demostrado efectividad en reducir los esfuerzos computacionales necesarios para encontrar trayectorias óptimas o políticas de operación y control. Fue desarrollada inicialmente por el Dr. Richard Bellman y sus asociados a finales de los 50's. Si bien las bases matemáticas que sustentan el método son complejas, las aplicaciones prácticas suelen ser sencillas por la forma lógica en que se plantean los procedimientos. Existe una abundante bibliografía referente a los fundamentos teóricos del método [1] y a sus aplicaciones a la operación de sistemas eléctricos como el pre-despacho [2]. La técnica de programación dinámica es también una de las pioneras en la solución del problema de asignación de unidades [3]. Sin embargo, la aplicación a grandes sistemas se ha visto limitada por la dimensionalidad del problema. Para ello, existe una variedad de alternativas de reducción del espacio de soluciones que básicamente consisten en la limitación simultánea de estrategias (o arcos) y de estados (nodos) [2] trayendo como principal consecuencia negativa la exploración de un espacio de soluciones reducido que limitan las garantías de encontrar la solución óptima. A pesar de esto, la programación dinámica sí ha podido ser aplicada con éxito a casos reales (o de gran-escala) cuando ha sido planteada como solución a los sub-problemas generados dentro de un esquema de descomposición basado en la relajación de la función lagrangeana (ver detalles de la descomposición en anexo A y las referencias allí citadas). Ha sido ampliamente utilizada para resolver el problema estos sub-problemas denominados como "asignación individual de las unidades". Aunque esta aplicación específica como parte del procedimiento de relajación lagrangeana ha sido propuesta desde la creación del método de relajación lagrangeana, pocas referencias detallan las estrategias más exitosas para su aplicación. De esta forma, a continuación se presentan las principales consideraciones para realizar una eficiente asignación individual de unidades basadas en programación dinámica, sin recurrir a técnicas heurísticas (ver por ejemplo referencia [4]).

Supóngase el siguiente problema de pre-despacho básico:

$$\min: \sum_t^T \sum_i^I \text{Cop}_{i,t} + \text{Cst}_{i,t} \quad (\text{B.1})$$

$$\text{sujeto a: } \sum_i^I \text{P}_{i,t} = \text{P}_d^t \quad \forall t = 1, \dots, T \quad (\text{B.2})$$

La función lagrangeana original se plantea de la siguiente forma:

$$L_r = \min: \sum_t^T \sum_i^I \text{Cop}_{i,t} + \text{Cst}_{i,t} + \lambda^t \left[\text{P}_d^t - \sum_i^I \text{P}_{i,t} \right] \quad (\text{B.3})$$

Se puede “relajar” al suponer conocidos los multiplicadores de las restricciones de acoplamiento (

$\lambda^t = \hat{\lambda}^t$), resultando: (ver anexo A)

$$L_r = \sum_i^I \min: \sum_t^T \left[\text{Cop}_{i,t} + \text{Cst}_{i,t} - \hat{\lambda}^t \cdot \text{P}_{i,t} \right] \quad (\text{B.4})$$

Entonces, para cada $i=1, \dots, I$ se debe resolver el siguiente problema:

$$\min: \sum_t^T \left[\text{Cop}_{i,t} + \text{Cst}_{i,t} - \hat{\lambda}^t \cdot \text{P}_{i,t} \right] \quad (\text{B.5})$$

Sujeto a: todas las restricciones operativas de las propias de la unidad “i”

La presencia de variables binarias de decisión en los costos de operación y los costos de partida conforman un problema de optimización no-convexo entero-mixto. Aunque existen diversas técnicas de solución para esta clase de problemas, la más utilizada ha sido la programación dinámica en el sentido del tiempo (forward dynamic programming) [1]. Se plantea encontrar la asignación óptima de la unidad en los períodos $t=1, \dots, T$ teniendo para cada t , dos (2) posibles estados de operación: encendido, apagado. La ilustración B.1 muestra el esquema de programación dinámica individual tradicional.

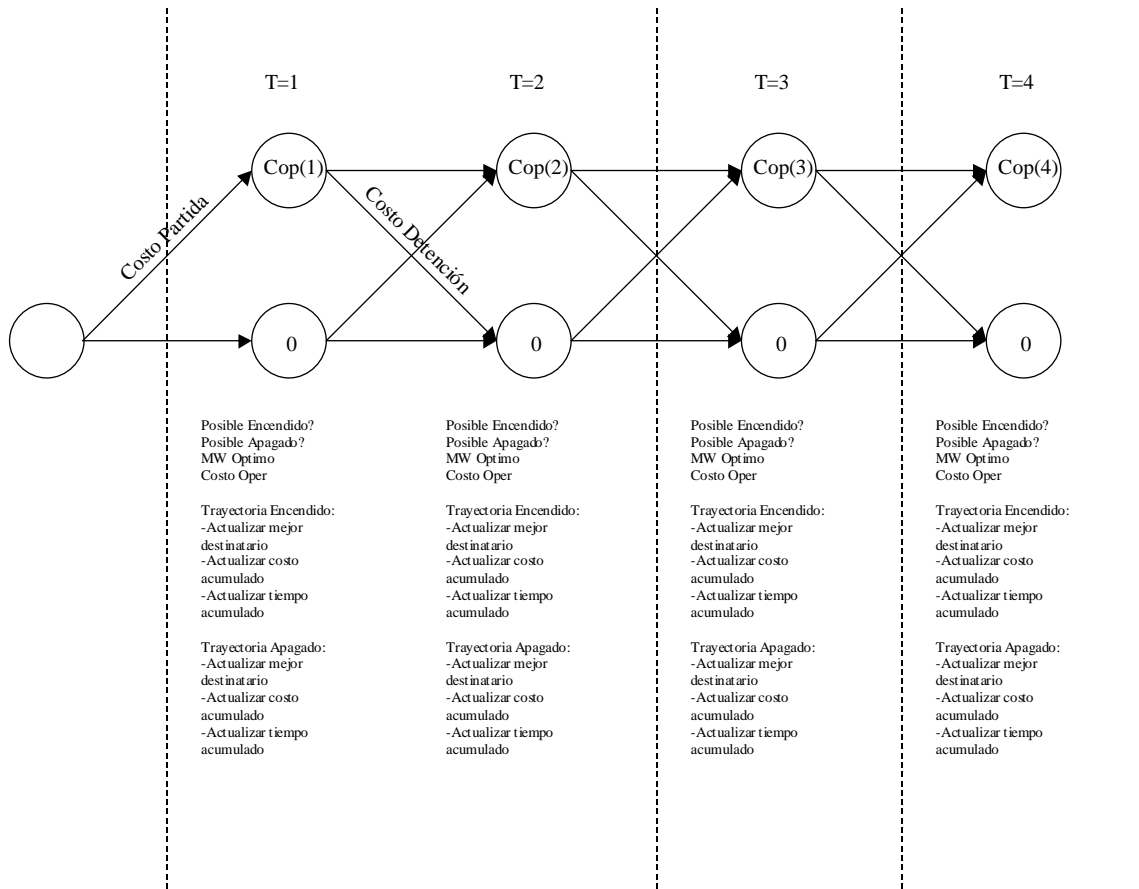


Ilustración B.1. Esquema de Programación Individual de las Unidades

7.2.2 Algoritmo de Solución

La técnica abordada consiste en el registro de 2 trayectorias históricas. El algoritmo de solución propuesto en detalle es el siguiente:

1. Para $t=1$: Determinar estados posibles en función de la condición inicial, tiempos mínimos de operación/detención, salidas programadas, etc.
2. Determinar el nivel de generación ($P_{i,t}$) óptimo a partir de la siguiente expresión: (considerando los límites de generación de la unidad)

$$\frac{\partial \text{Cop}_{i,t}}{\partial P_{i,t}} - \hat{\lambda}^t \cdot P_{i,t} = 0 \quad (\text{B.6})$$

3. Determinar los costos de operación para el ($P_{i,t}$). Si $\text{Cop}_{i,t} < 0$, considerar la posibilidad de encender la unidad. Si $\text{Cop}_{i,t} \geq 0$ hacer $t = t + 1$ y volver a 1.

4. Determinación de la mejor trayectoria para llegar al estado de encendido:
 - Calcular: costo acumulado($t-1$) + costo de encendido - costo mantener encendido.
 - Si >0 registrar mejor “venir desde apagado”.
 - Si <0 registrar mejor “venir desde encendido”.
5. Determinación de la mejor trayectoria para llegar al estado de apagado:
 - Calcular: costo acumulado ($t-1$) encendido + costo de detención - costo mantener apagado.
 - Si >0 registrar mejor “venir desde apagado”.
 - Si <0 registrar mejor “venir desde encendido”.
6. Actualizar el costo y el tiempo de operación acumulado en las dos (2) trayectorias.
7. Si $t = T$, hacer: acumulado encendido - costo acumulado apagado. Si >0 , hacer que trayectoria hacia apagado sea el despacho de la unidad. Si <0 , hacer que trayectoria hacia encendido sea el despacho de la unidad. Finalizar.
8. Si $t < T$ volver a 2.

7.2.3 Referencias

- [1] Stuart E. Dreyfus & Averill M Law, “The Art and Theory of Dynamic Programming”, New York, USA, Academic Press, 1977.
- [2] G. B. Sheblé & G. N. Fahd, “Unit Commitment Literature Synopsis”, IEEE Transactions on Power Systems, Vol. 9, No 1, pp. 128-135, Feb 1994.
- [3] Allen J. Wood & Bruce F. Wollenberg. “Power Generation Operation and Control”. John Wiley and Sons, Second Edition, NY 1996.
- [4] Weerakorn Ongsakul & Nit Petcharaks. “Unit Commitment by Enhanced Adaptive Lagrangian Relaxation”. IEEE Transactions on Power Systems, Vol. 19, No 1, Feb 2004.

7.3 ANEXO C - Método Subgradiente

7.3.1 Introducción

El método del subgradiente es un algoritmo sencillo para resolver problemas de optimización de funciones convexas no-diferenciables. En la referencia [1] puede encontrarse un resumen de sus variantes y algunas aplicaciones simples. Fue desarrollado originalmente en la ex Unión Soviética por Shor en los años 70's [2]. El método es muy similar a los métodos de gradiente (derivativos de primer y segundo orden [3]) con algunas notables excepciones. Por ejemplo, el subgradiente avanza por pasos de longitud predefinida mientras que los derivativos lo hacen por una recta en dirección del gradiente. No es un método descendente y por lo tanto es posible encontrar valores mayores de la función objetivo en iteraciones consecutivas en problemas de minimización. Suelen ser mucho más lentos que los métodos de gradiente con la ventaja de poder ser aplicados a una variedad mucho mayor de problemas. Han mostrado una variedad de aplicaciones en algoritmos de programación genéricos y, en particular, a métodos de descomposición dual.

El subgradiente de una función f es cualquier vector g que satisface la siguiente desigualdad:

$$f(x_2) \geq f(x_1) + g(x_2 - x_1) \quad (C.1)$$

Entonces, supóngase que se desea resolver el siguiente problema de optimización:

$$\begin{aligned} \text{Maximizar: } & f(x) \\ & x \in X \end{aligned} \quad (C.2)$$

Se define el método del subgradiente como la búsqueda iterativa de la solución a (C.2) de la siguiente forma:

$$x^{k+1} = x^k + \alpha_k \cdot g^k \quad (C.3)$$

donde g^k es cualquier subgradiente de f . En efecto, si f es diferenciable, la escogencia ideal sería:

$g^k = \nabla f(x^k)$, convirtiéndose entonces en un método de gradiente convencional con la excepción del control de longitud α_k .

El método del subgradiente no garantiza que el resultado de la siguiente iteración sea mayor en problemas de maximización. Por lo tanto, es usual llevar registro de la mejor solución encontrada hasta entonces. En métodos descendientes no es necesario llevar tal registro sino que se considera al punto actual como el mejor hasta el momento.

7.3.2 Reglas de control de los pasos

En la literatura, existe una variedad notable de proposiciones para controlar el tamaño de los pasos α_k . Se supone dependiente de las características del problema y su determinación involucra usualmente procedimientos heurísticos. A continuación se resumen algunos de las principales formas de control tomados de [1]:

- *Tamaño* del paso constante: $\alpha_k = h$, con $h = \text{constante}$.
- *Longitud* del paso constante: Se refiere a mantener la norma eucleniana entre soluciones consecutivas constantes en cada iteración. Es decir:

$$\alpha_k = \frac{h}{\|g^k\|_2}, \text{ tal que: } \|x^{k+1} - x^k\|_2 = h \quad (\text{C.4})$$

- *Series sumables* pero cuadrados no sumables: se escoge el parámetro α_k que satisface la condición:

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty \quad \text{y} \quad \sum_{k=1}^{\infty} \alpha_k = \infty \quad (\text{C.5})$$

- *Series no sumables diminutivas (diminishing step size rule)*: el parámetro α_k tal que se satisface la condición:

$$\lim_{k \rightarrow \infty} \alpha_k = 0 \quad \text{y} \quad \sum_{k=1}^{\infty} \alpha_k = \infty \quad (\text{C.6})$$

Por ejemplo si se escoge: $\alpha_k = h / \sqrt{k}$ con $h > 0$

7.3.3 Convergencia

Para el método del subgradiente la prueba de convergencia no se basa en la observación del crecimiento/decrecimiento progresivo de la función objetivo como en los métodos de gradiente convencional. En su lugar, se suele medir la distancia eucladiana a la solución óptima. Para funciones f que satisfacen las condiciones Lipschitz, el empleo del método del subgradiente permite el cumplimiento de la siguiente desigualdad:

$$f_{\text{mejor}}^k - f^* = \min_{i=1, \dots, k} : f(x^i) - f^* \leq \frac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2 \sum_{i=1}^k \alpha_i} \quad (\text{C.7})$$

donde:

f_{mejor}^k : El mejor resultado obtenido hasta la iteración "k"

f^* : Valor de la función f en la solución óptima

G: Es la cota superior de la norma del sugradiente $\|g\|_2$

Y además, R se define como una constante proporcional a la norma de la solución inicial. Es decir:

$$R = \|x^1 - x^*\|_2^2 \quad (\text{C.8})$$

La demostración de la desigualdad (C.7) escapa del alcance del presente trabajo. Se puede encontrar en las referencias [1] y [2].

Entonces, a partir de (C.7) se pueden deducir las siguientes afirmaciones:

- Para *Tamaño* del paso constante: ($\alpha_k = h$)

$$f_{\text{mejor}}^k - f^* \leq \frac{R^2 + G^2 h^2 k}{2hk} \quad (\text{C.9})$$

Implica que el subgradiente converge a $G^2 h / 2$ si $k \rightarrow \infty$. Es decir, la mejor solución luego de un número muy elevado de iteraciones se alejará del óptimo en $G^2 h / 2$.

- Para *Longitud* del paso constante: ($\alpha_k = h / \|g^k\|_2$)

$$f_{\text{mejor}}^k - f^* \leq \frac{R^2 + G^2 h^2 k}{2hk} \quad (\text{C.10})$$

Implica que el subgradiente converge a $G \cdot h/2$ si $k \rightarrow \infty$. Es decir, el método del subgradiente converge a una solución acotada en $G \cdot h/2$ de la solución óptima.

- *Series sumables pero cuadrados no sumables:*

$$f_{\text{mejor}}^k - f^* \leq \frac{R^2 + G^2 \|\alpha\|_2^2}{2 \sum_{i=1}^k \alpha_i} \quad (\text{C.11})$$

Por lo tanto, para $k \rightarrow \infty$:

$$\lim_{k \rightarrow \infty} \frac{R^2 + G^2 \|\alpha\|_2^2}{2 \sum_{i=1}^k \alpha_i} = 0 \quad (\text{C.12})$$

Queda demostrado entonces que el método del subgradiente converge a la solución óptima.

- *Series no sumables diminutivas (diminishing step size rule):*

Si la serie α_k converge a cero (condición inicial de α_k), entonces el lado derecho de (A.3.7) converge a cero también. Quiere decir que para este caso, el método del subgradiente también converge a la solución óptima. Se sugiere al lector referirse a [1],[2] para ver la respectiva demostración formal.

7.3.4 Referencias

- [1] S. Boyd, L. Xiao, A Mutapcic, "Subgradient Methods", EE392o: Optimization Projects, Professor Stephen Boyd and Professor Z.-Q. Luo home page at Stanford University, Available online at: <http://www.stanford.edu/class/ee392o>, 2003.
- [2] N. Z. Shor. "Minimization Methods for Non-differentiable Functions", Springer Series in Computational Mathematics. Springer, 1985.
- [3] P. E. Gill, W. Murray, M. H. Wright, "Practical Optimization", Academic Press Inc, London, 1981.

7.4 ANEXO D - Relajación Lagrangeana Aumentada

7.4.1 Introducción

La función lagrangeana aumentada se define como la incorporación a la función lagrangeana de un término no lineal, denominado “de aumentación”, con el objeto de modificar a favor las características de un problema de optimización. Se pueden definir como un subconjunto de una clase más general de funciones denominadas “funciones auxiliares” para la solución de problemas de optimización global. Las funciones auxiliares son todas aquellas funciones que convolucionan la función objetivo con las funciones de restricciones, generando un único problema de optimización sin restricciones [1]. Una excelente referencia de los diversos tipos de métodos de aumento puede encontrarse en [2]. Quizás la aplicación principal se concentra en sumar a la función objetivo otra función convexa que involucre las variables del problema y que tienda a “desaparecer” en el óptimo [3]. Por ejemplo, dado el siguiente problema de optimización:

$$\min: x^3 \mid x - 1 = 0 \quad (\text{D.1})$$

La función lagrangeana convencional estaría dada por la convolución lineal:

$$L_{\text{aug}} = x^3 + \lambda(1 - x) \quad (\text{D.2})$$

Una función lagrangeana aumentada podría ser la siguiente:

$$L_{\text{aug}} = x^3 + \lambda(1 - x) + c \cdot (1 - x)^2 \quad (\text{D.3})$$

Entonces, para este sencillo caso se puede observar que el término agregado “desaparecerá” en la solución óptima $x=1$. Sin embargo, el término “de aumentación”, para ciertos valores del “ c ”, convierte al problema original (D.1) en un problema de optimización estrictamente convexo. La figura D.1 muestra la mejora de convexidad introducida por el lagrangeano aumentado a la función $y = x^3$.

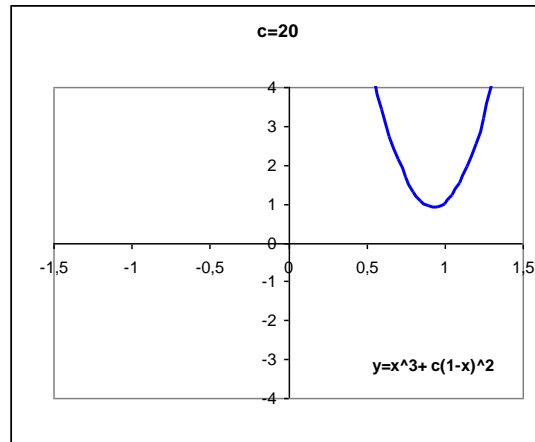
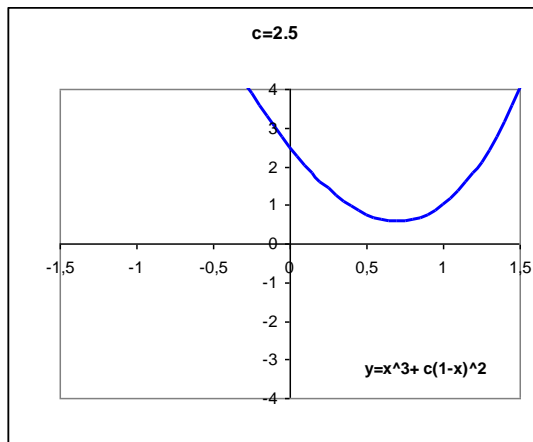
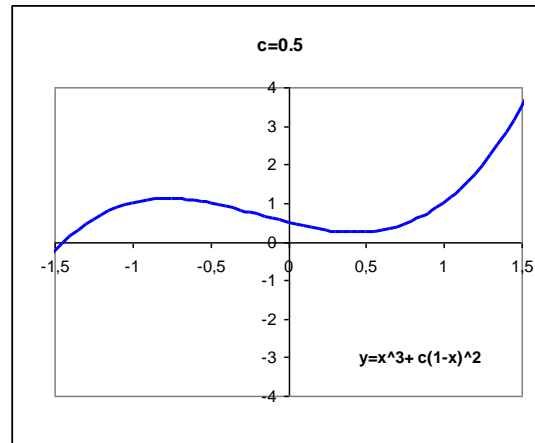
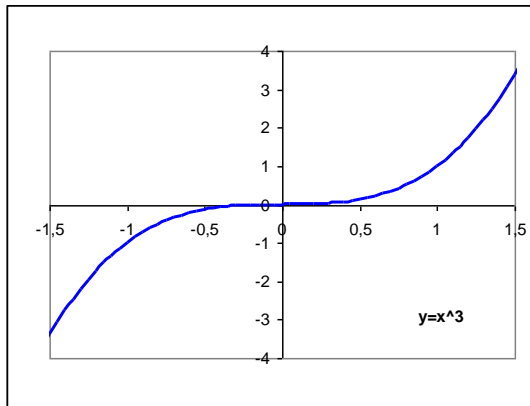


Ilustración D.1. Efecto de Introducción del término cuadrático

Dos dificultades evidentes pueden resultar del empleo de los lagrangeanos aumentados:

1. Si es aplicado a problemas de programación lineal o lineal-entero-mixto, el término “de aumentación” lo transformará en un problema no-lineal.
2. Puede atentar contra el método de descomposición para problemas de estructuras por bloques.

7.4.2 Técnicas de Descomposición

Como las técnicas de optimización suelen definirse dentro de un proceso iterativo, para sobreponer las dificultades en el uso de los lagrangeanos aumentados (mencionadas anteriormente) se pueden aplicar técnicas que se fundamentan en linealización del término agregado en la vecindad de la solución la iteración anterior. Esta propuesta es referida en algunas referencias como técnicas de descomposición y coordinación [5]. Aunque originalmente fueron propuestas para la solución de problemas de

optimización no-separables, futuros desarrollos se concentraron en la aplicación a soluciones a inecuaciones variacionales. Sin embargo, suelen plantear la separabilidad en términos no-lineales para que el lagrangeano aumentado pueda continuar siendo estrictamente convexo.

Existe una variedad métodos de descomposición aplicables a la solución de inecuaciones variacionales.

Uno de los pioneros fue el denominado principio del problema auxiliar, en inglés *Auxiliary Problem Principle* y en adelante APP, presentado por Cohen en 1980 [4]. Fue originalmente desarrollado para el análisis de algoritmos de optimización basados en métodos de gradiente y subgradiente así como también algoritmos de descomposición. En general, el APP consiste en la separación del término

cuadrático $\frac{c}{2} \cdot \|b - Ax\|^2$ en la linealización alrededor de la solución anterior más un término cuadrático

separable convexo. Sin embargo, posteriores estudios se centraron en la ampliación del método al aplicar el operador variacional en un punto variable recibió la denominación de métodos de punto próximo, en inglés *Proximal Point Methods* y en adelante PPM.

Informalmente, el APP consiste en la linealización de cualquier término no separable en el lagrangeano aumentado y, simultáneamente, la adición de términos convexos escogidos de tal forma que sigan

siendo separables. Suponiendo que el término de aumento sea $\frac{c}{2} \cdot \|b - Ax\|^2$, el APP consistirá en la

linealización de este término alrededor de la solución de la iteración anterior $x^{(k)}$ más un término cuadrático separable. Es decir:

$$\frac{c}{2} \cdot \|b - Ax\|^2 \approx c \cdot (b - Ax^{(k)}) \cdot (b - Ax) + \frac{1}{2\epsilon} [b - A(x - x^{(k)})]^2 \quad (D.4)$$

Nótese que (D.4) puede ser estrictamente convexo en función de la escogencia de c y ϵ . En la referencia

[4] se demuestra que el algoritmo iterativo converge a la solución óptima si se escoge valores de c y ϵ

tales que $c \leq \frac{1}{2\epsilon}$.

7.4.3 Referencias

- [1] Yu. G. Evtushenko, A.M. Rubinov, V.G. Zhadan, "General Lagrange-Type Functions in Constrained Global Optimization Part II: Exact Auxiliary Functions", Optimization Methods and Software, vol 16, pp 231-256, 2001.
- [2] A.M. Rubinov, Lagrange-Type Functions in Constrained Non-convex Optimization", Kluwer Academic Publisher, Applied Optimization Series, Vol. 85, Nov. 2003.
- [3] P. E. Gill, W. Murray, M. H. Wright, "Practical Optimization", Academic Press Inc, London, 1981.
- [4] G. Cohen, "Auxiliary Problem Principle and Decomposition of Optimization Problems", Journal of Optimization Theory and Applications, vol. 32, pp. 277-305, 1980.
- [5] L. Murphy, J. Contreras and F.F Wu, "A Decomposition-Coordination Approach for Large-Scale Optimization", Proceedings SIAM Conference on Parallel Processing for Scientific Computing, pp: 78-83, Feb 1995.

7.5 ANEXO E – DATOS DEL SISTEMA RTS-96

Se presentan a continuación los datos de las unidades y de la demanda necesarios para formular problemas de pre-despacho térmico [1].

7.5.1 Datos del RTS-96 necesarios para el pre-despacho

Unit ID	Fuel Type	MW	Net Plant Heat Rate [Btu/kwh]	Incremental Heat Rate Calculated by continuous function [Btu7kwh]
U12	Fossil Steam	2.4	16017	10179
		6	12500	10330
		9.6	11900	11668
		12	12000	13219
U20	Combustion Turbine	15.8	15063	9859
		16	15000	10139
		19.8	14500	14272
		20	14499	14427
U50	Hydro	50	N/A	N/A
U76	Fossil Steam	15.2	17107	9548
		38	12637	9968
		60.8	11900	11576
U100	Fossil Steam	76	12000	13311
		25	12999	8089
		50	10700	8708
		80	10087	9420
		100	10000	9877
U155	Fossil Steam	54.25	11244	8265
		93	10053	8541
		124	9718	8900
		155	9600	9381
U197	Fossil Steam	58.95	10750	8348
		118.2	9850	8833
		157.6	9644	9225
		197	9600	9620
U350	Fossil Steam	140	10200	8402
		227.5	9600	8896
		280	9500	9244
		350	9500	9768
U400	Nuclear Steam	100	12751	8848
		200	10825	8965
		320	10170	9210
		400	10000	9438

Tabla E.1. Curvas de Calor de las Unidades

Unit ID	Hot Start (Mbtu)	Cold Start (Mbtu)	Min Down Time [hr]	Min Up Time [hr]	Start Time Hot [hr]	Start Time Cold [hr]	Warm Star Time [hr]	Ramp Rate MW/Minute
U12	38	68	2	4	2	4	12	1
U20	5	5	1	1	0	0	1	3
U50	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
U76	596	596	4	8	3	12	10	2
U100	250	566	8	8	2	7	60	7
U155	260	953	8	8	3	11	60	3
U196	443	775	10	12	4	7	24	3
U350	1915	4468	48	24	8	12	96	4
U400	N/A	N/A	1	1	N/A	N/A	N/A	20

Tabla E.2. Restricciones de operación

Hour	% of Daily Pick Load	Hour	% of Daily Pick Load
24	67	12	95
1	63	13	95
2	60	14	93
3	59	15	94
4	59	16	99
5	59	17	100
6	60	18	100
7	74	19	96
8	86	20	91
9	95	21	83
10	96	22	73
11	95	23	63

Tabla E.3. Datos de la Demanda (% de la carga máxima diaria)

7.5.2 Construcción de formulaciones de pre-despacho

Como se mencionó en la sección 4.3.1, las formulaciones se hacen tomando de forma aleatoria unidades de las tablas E.1 y E.2. El vector aleatorio de distribución uniforme se realizó con la “semilla” propia de la función ALEATORIO() de MS Excel. Los resultados para el caso sintético en sección se muestran en la tabla a continuación:

Número de Unidades	Unidad ID	Número de Unidades	Unidad ID	Número de Unidades	Unidad ID	Número de Unidades	Unidad ID	Número de Unidades	Unidad ID
1	U20	21	U350	41	U100	61	U155	81	U50
2	U350	22	U350	42	U20	62	U50	82	U197
3	U12	23	U400	43	U20	63	U197	83	U12
4	U100	24	U12	44	U100	64	U12	84	U155
5	U76	25	U76	45	U197	65	U50	85	U350
6	U197	26	U20	46	U76	66	U20	86	U76
7	U155	27	U12	47	U76	67	U12	87	U100
8	U197	28	U20	48	U50	68	U20	88	U12
9	U76	29	U20	49	U12	69	U12	89	U155
10	U76	30	U100	50	U350	70	U350	90	U197
11	U155	31	U350	51	U155	71	U197	91	U12
12	U20	32	U76	52	U155	72	U100	92	U400
13	U50	33	U12	53	U100	73	U76	93	U350
14	U197	34	U350	54	U76	74	U100	94	U50
15	U20	35	U155	55	U197	75	U400	95	U350
16	U12	36	U350	56	U350	76	U20	96	U76
17	U20	37	U100	57	U100	77	U50	97	U155
18	U12	38	U20	58	U350	78	U155	98	U12
19	U350	39	U20	59	U12	79	U197	99	U20
20	U12	40	U100	60	U350	80	U50	100	U20

Tabla E.4. Resultado sorteo Aleatorio de Unidades

7.5.3 Referencias

- [1] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour & C. Singh, "The IEEE Reliability Test System – 1996. A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee", IEEE Transactions on Power Systems, Vol. 14, No.3, pp. 1010-1020, Aug 1999.