



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

MÉTODOS DE OPTIMIZACIÓN BAJO INCERTIDUMBRE APLICADOS AL
PROBLEMA DE COORDINACIÓN HIDRO-TÉRMICA CHILENO

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

CARLOS FRANCISCO PÉREZ PÉREZ

PROFESOR GUÍA:
ANDRÉS WEINTRAUB POHORILLE

MIEMBROS DE LA COMISIÓN:
RODRIGO PALMA BEHNKE
FERNANDO ORDÓNEZ PIZARRO

SANTIAGO DE CHILE
2017

**RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE:** Ingeniero Civil Industrial y grado de
Magíster en Gestión de Operaciones
POR: Carlos Francisco Pérez Pérez
FECHA: 5 de abril de 2017
GUIA: ANDRÉS WEINTRAUB POHORILLE

MÉTODOS DE OPTIMIZACIÓN BAJO INCERTIDUMBRE APLICADOS AL PROBLEMA DE COORDINACIÓN HIDRO-TÉRMICA CHILENO

El presente trabajo, trata sobre metodologías de resolución de problemas estocásticos multi-etapa. El objetivo es comparar el algoritmo *Stochastic Dual Dynamic Programming* (SDDP), con la metodología *Progressive Hedging* (PH). La diferencia entre metodologías radica en que SDDP corresponde a un algoritmo de descomposición vertical o por períodos, mientras que PH representa una metodología de descomposición horizontal o por escenarios.

El estudio se desarrolla en el contexto eléctrico chileno, caracterizado por una red eléctrica compuesta por generadores, demandas y líneas de transmisión. Además de una red hídrica; en la que se modelan todos los generadores hidroeléctricos, flujos hídricos y capacidades de almacenamiento.

Para realizar las pruebas, se utilizan instancias de distintos tamaños y horizontes de planificación, bajo el supuesto uninodal (todas las centrales eléctricas y demandas, conectadas a un mismo nodo) y multinodal.

Las metodologías fueron testeadas mediante un árbol en común. Los resultados muestran que en general, SDDP presenta tiempos de ejecución mucho menores, atribuibles principalmente al lenguaje de ejecución (Fortran vs Python). En cuanto a la exactitud de la solución, PH muestra un mejor desempeño, con diferencias poco significantes.

Un segundo estudio, es la confrontación de generadores de árboles de escenarios. Se compara la metodología actual (nombrada PLP), con dos generadores desarrollados íntegramente en esta investigación. Estas metodologías se basan en probabilidades condicionales por período, y llevan el nombre de Synth y NSG, que corresponden a generación de escenarios sintéticos e históricos, respectivamente. Una de las ventajas de estas metodologías es que consideran tanto la correlación espacial, como la correlación temporal.

Bajo escenarios extremos, Synth muestra los menores costos operativos en un 60 % de las pruebas realizadas, obteniendo costos comparativos hasta un 7% menores.

Por otro lado, bajo escenarios promedios, existe un desempeño muy parecido entre los generadores Synth y PLP, donde PLP muestra mayores ahorros comparativos.

A Dios y mi familia.

Agradecimientos

Primero quiero agradecer a mi familia, que incondicionalmente me ha dado su apoyo y amor en mi formación como persona y profesional.

Al Profesor Andrés por confiar en mí y entregar un tema de tesis tan interesante y con aplicaciones importantes para nuestro país.

Asimismo quiero agradecer a Ignacio Rios, quien fue un gran apoyo y guía en el desarrollo de mi tema de tesis, una persona con un intelecto destacable.

De igual manera, quisiera agradecer al Departamento de Ingeniería Eléctrica (DIE), que a través del profesor Rodrigo Palma, abrió las puertas para desarrollar la investigación, facilitando la información y apoyando con el entendimiento del funcionamiento actual de la distribución eléctrica chilena.

Por último, quiero agradecer a Sebastián Püschel, investigador del Centro de Energía de la Universidad de Chile, quien pasó largas horas aportando en el desarrollo de esta tesis.

Powered@NLHPC: Esta investigación fue parcialmente apoyada por la infraestructura de supercómputo del NLHPC (ECM-02)

Tabla de Contenido

1. Introducción	1
1.1. Motivación	2
1.2. Estado del Arte	3
1.3. Stochastic Dual Dynamic Programming (SDDP)	5
1.4. Organización de contenidos	5
2. Marco Teórico	6
2.1. Árboles y Escenarios	7
2.2. Principio de no-anticipatividad	10
2.3. Formulación extendida	11
2.4. Introducción a Progressive Hedging	12
2.4.1. Intuición	12
2.4.2. Pseudo Código PH	14
2.4.3. Caracterización	14
2.5. SDDP	15
3. Problema eléctrico	18
3.1. Antecedentes y descripción	18
3.2. Reducción de sub-problema bloque-etapa	21
3.3. Incertidumbre	23
3.4. Generación de Escenarios	24
3.4.1. Metodologías de generación de escenarios actuales (Generador de escenarios PLP)	25
3.4.2. Metodología de generación de escenarios propuesta	29
3.4.3. Tratamiento de Data	29
3.4.4. Generador de escenarios Synth - NSG	30
4. Modelo de optimización	34
4.1. Conjuntos	34
4.2. Parámetros	35
4.3. Variables	36
4.4. Restricciones	37
4.5. Función objetivo	39
4.6. Rectificaciones del modelo	40
5. Algoritmo Progressive Hedging	42
5.1. Pseudo-código extendido	42

5.2.	Ajustes y mejoras	44
5.2.1.	Aceleración de la resolución de cada subproblema	44
5.2.2.	Paralelización del algoritmo	45
5.2.3.	Linealizar término cuadrático proximal	46
5.2.4.	Ajustes del parámetro ρ de penalización	48
5.2.5.	Bundling de escenarios	49
5.2.6.	Criterio de término	50
6.	Implementación	52
6.1.	Implementación paralela PYSP	53
6.2.	Comparación de Metodologías	53
6.3.	Recursos computacionales utilizados	56
6.4.	Instancias	57
7.	Resultados	63
7.1.	Convergencia PH	64
7.2.	Cuenca del Laja	68
7.2.1.	Árbol Base	68
7.3.	Sistema completo (SIC - SING)	69
7.3.1.	Caso Uninodal	69
7.3.2.	Caso Multinodal	70
7.3.3.	Análisis de resultados Árbol Base - Sistema completo (SIC - SING)	71
7.4.	Metodologías Generadoras de Escenario	72
7.4.1.	Resultados SYNTH	73
7.4.2.	Resultados NSG	75
7.5.	Comparación de Árbol + Metodología	78
7.5.1.	Hidrologías Húmedas	80
7.5.2.	Hidrologías Medias	82
7.5.3.	Hidrologías Secas	83
8.	Conclusión	86
9.	Bibliografía	88
	Anexos	92
	Anexo A: Pseudo-código Benders Multi etapa	92
	Anexo B: Tipo de centrales de generación eléctrica	95
	Anexo C: Modelo hidráulico - Cuenca del Laja	97
	Anexo D: Resultados de Metodologías de aceleración de resolución de cada subproblema	101
	Anexo E: Resultados de la linealización del término cuadrático proximal	102
	Anexo F: Estrategias para cálculo de ρ	103
	Anexo G: Archivos para paralelización en NLHPC	106
	Anexo H: Convergencia de PH	108
	Anexo I: Años hidrológicos seleccionados para comparación de metodologías	108
	Anexo J: Tiempos de ejecución de lenguajes: Fortran vs Python	111

Índice de Tablas

1.1. Generación eléctrica por tipo de central. Fuente: Comisión nacional de energía CNE [1]	2
3.1. Cantidad de escenarios según aperturas y períodos	33
6.1. Descripción instancia Cuenca del Laja	58
6.2. Parámetros de línea de transmisión - Cuenca del Laja	59
6.3. Parámetros de generadores térmicos - Cuenca del Laja	59
6.4. Parámetros generadores hidroeléctricos - Cuenca del Laja	59
6.5. Características de la demanda - Cuenca del Laja	59
6.6. Descripción instancia Uninodal SIC-SING	61
6.7. Descripción instancia Multinodal SIC-SING	62
7.1. Cuenca del Laja - Árbol base	68
7.2. SIC - SING - Caso Uninodal - Árbol Base	70
7.3. SIC - SING - Caso Multinodal - Árbol Base	70
7.4. Resultados Generador de escenarios SYNTH	73
7.5. Resultados Generador de escenarios NSG	76
7.6. Tiempos de ejecución árbol + Metodología	79
7.7. Tabla resumen escenarios húmedos	81
7.8. Tabla resumen escenarios medios	83
7.9. Tabla resumen escenarios secos	85
A.1. Metodologías de aceleración para cada subproblema	101
A.2. Resultados de tramos para ρ	102
A.3. Convergencia de PH	108
A.4. Años hidrológicos seleccionados	109
A.5. Porcentaje de variación de F.O. por escenario	110
A.6. Tiempos de las pruebas comparativas en relación a C (menos es mejor, desempeño de $C = 1.0$).	111

Índice de Ilustraciones

2.1. Largo de períodos vs incertidumbre.	8
2.2. Árbol de escenario y formulación extendida.	9
2.3. Representación de principio de no anticipatividad, los nodos encerrados en elipses se denominan conjuntos de información.	10
3.1. Representación de Red eléctrica y sus nodos.	19
3.2. Representación de Red hidráulica, con afluentes naturales y sus nodos.	20
3.3. Representación de la carga en el tiempo (8 días) (a)Potencia demandada en cada instante de tiempo. (b) Cantiada de tiempo que la potencia demandada se encuentra sobre un nivel.	21
3.4. Representación aproximada por bloques de demanda de la curva de duración de la carga.	22
3.5. Hidrologías Laja 1960-2013.	23
3.6. Mínima y máxima hidrología Laja 2007 - 2000.	24
3.7. Diagrama de escenarios usado en PLP.	26
3.8. Ejemplo del generador de escenarios que considera todas las posibles combinaciones.	27
3.9. Ejemplo de escenarios improbables.	28
3.10. Estimación de distribución.	31
3.11. Selección de vecinos y estimación de distribución.	32
3.12. Distribuciones condicional, y elección de puntos dentro de cada segmento.	32
5.1. Bundling de Escenarios.	49
5.2. Región de la Formulación extendida compacta.	50
6.1. Representación teórica del árbol común utilizado, sin aperturas desde el segundo período.	54
6.2. Red eléctrica instancia Cuenca del Laja.	58
6.3. Red hidráulica instancia Cuenca del Laja.	60
7.1. Convergencia de PH, Caso SIC-SING Uninodal, 768 escenarios.	64
7.2. Variación porcentual de PH por iteración, Caso SIC-SING Uninodal, 768 escenarios.	65
7.3. F.O por iteración, Caso SIC-SING Uninodal, 768 escenarios.	66
7.4. Variación porcentual de F.O por iteración, Caso SIC-SING Uninodal, 768 escenarios.	67

7.5. Tiempos de ejecución Generador de escenarios SYNTH - Formulación extendida (EF), Caso SIC-SING Multinodal	74
7.6. Tiempos de ejecución Generador de escenarios SYNTH - Formulación extendida (EF), Caso SIC-SING Multinodal	75
7.7. Tiempos de ejecución Generador de escenarios NSG - Formulación extendida (EF), Caso SIC-SING Multinodal	77
7.8. Tiempos de ejecución Generador de escenarios NSG - Formulación extendida (EF), Caso SIC-SING Multinodal	78
7.9. Histograma de Costos operacionales, escenarios húmedos	80
7.10. Variación porcentual de Costos operacionales, escenarios húmedos	81
7.11. Histograma de Costos operacionales, escenarios promedio	82
7.12. Variación porcentual de Costos operacionales, escenarios promedio	82
7.13. Histograma de Costos operacionales, escenarios secos	84
7.14. Variación porcentual de Costos operacionales, escenarios secos	84
A.1. Representación de Red hidráulica - Cuenca del Laja.	98
A.2. Modelo hidráulico completo - Cuenca del Laja.	99
A.3. Modelo hidráulico completo - Cuenca del Laja (continuación).	100

Capítulo 1

Introducción

El mercado eléctrico en Chile, está compuesto por tres sectores cuyas actividades permiten disponer de energía eléctrica en los distintos puntos del mercado (la interconexión física de los componentes de cada uno de estos sectores se denomina sistema eléctrico): Generación, sector que tiene como función el de producir la energía eléctrica a través de distintas tecnologías; Transmisión, su función es la de transmitir, en niveles altos de voltaje, la energía producida a todos los puntos del sistema eléctrico; y Distribución, sector encargado de llevar, en niveles de voltaje más reducidos que los de Transmisión, la energía desde un cierto punto del sistema eléctrico a los consumidores regulados que este sector atiende. Actualmente la producción eléctrica se basa principalmente en la energía hídrica y térmica. En el sistema interconectado central y el sistema interconectado del norte grande (desde ahora en adelante SIC y SING, respectivamente) existe un total de 386 generadores eléctricos y 1.163 líneas de transmisión¹.

De acuerdo al recurso utilizado para producir la electricidad, los generadores eléctricos del sistema SIC-SING se dividen en:

1. Centrales hidroeléctricas: Centrales que utilizan caídas de agua para la generación eléctrica. Recurren a la energía potencial producida por un desnivel natural, pasando el agua por una turbina hidráulica que transmite la energía mecánica a un generador eléctrico transformándola en electricidad.
2. Centrales térmicas convencionales: Este tipo de centrales utiliza combustibles fósil como el carbón, gas natural, petróleo, entre otros, para calentar calderas que generan vapor de agua, este vapor (a alta presión) empuja las palancas de una turbina y mediante un generador eléctrico transforma la energía en electricidad.
3. Centrales eólicas: Transforman la energía cinética del viento, mediante un aerogenerador, en energía eléctrica.
4. Centrales termoelectricas solares: Existen dos tipos de centrales solares, la primera usa paneles fotovoltaicos que captan la energía luminosa del sol transformándola en energía eléctrica. El segundo tipo de centrales de concentración solar, funciona similar a las

¹<http://generadoras.cl/> Generadoras de Chile A.G.

centrales térmicas convencionales, utiliza la energía del sol para calentar un fluido, que produce la evaporación de un segundo fluido para luego empujar las palancas de una turbina generando electricidad.

5. Centrales de biomasa: Utilizan el mismo sistema de generación eléctrica que una central térmica convencional, pero el combustible utilizado en este caso proviene de residuos orgánicos.

A septiembre de 2015 en el sistema SIC-SIG existe una potencia bruta instalada de 20.180,71 [MW]. Y a la misma fecha la producción mediante centrales termoeléctricas convencionales alcanza un 64,08 %, mientras que la producción hídrica un 28,16 %, en la siguiente tabla se puede ver la distribución de la potencia eléctrica despachada [2] [3]:

Tipo de central	Generación en [MW]	Porcentaje
Biomasa	1.701	3,57 %
Central Hidroeléctrica	13.414	28,17 %
Eólica	1.238	2,60 %
Solar	750	1,57 %
Térmica convencional	30.519	64,08 %
Total general	47.624	100 %

Tabla 1.1: Generación eléctrica por tipo de central. Fuente: Comisión nacional de energía CNE [1]

1.1. Motivación

La producción hidroeléctrica se obtiene mediante el uso de caídas de agua, los principales generadores son: embalses, centrales en serie, y puras. Este tipo de producción tiene costos de inversión muy elevados, pero una vez instalada una central hídrica los costos variables son comparativamente despreciables. Por el contrario, los costos variables de operación en la producción térmica son más elevados, debido a que la generación es en base a combustibles como: petróleo, gas, carbón, entre otros.

Un coordinador hidrotérmico es el encargado de determinar la operación óptima del sistema, en otras palabras, se debe decidir cuánto producir con cada central, utilización del agua en los embalses, transmisión de energía por cada línea, para cada etapa dentro del período de planificación, teniendo en cuenta los costos de oportunidad.

Esta estrategia debe minimizar el valor esperado de los costos de operación a lo largo del período, compuesto por los costos de combustible, además de sanciones por incumplimiento de demanda eléctrica. La disponibilidad de cantidades limitadas de energía hidroeléctrica en forma de agua en los embalses, hace el problema de coordinación óptima de uso de agua muy complejo, ya que crea un vínculo entre una decisión operativa de una etapa dada y las futuras consecuencias de esa decisión. Esto es, si se agotan las reservas de energía hidroeléctrica, y se

produce una época de sequía, puede ser necesario el uso de generadores térmicos con costos mucho mayores en el futuro, o incluso dejar la demanda insatisfecha.

Por otro lado, si se mantienen altos los niveles de los embalses por medio de un uso intensivo de producción térmica, y en los períodos siguientes se tienen abundantes hidrologías, se deberá derramar agua del sistema, lo que significa una pérdida de energía y, en consecuencia, mayores costos de operación.

Así es como este trabajo se enfoca en encontrar la operación óptima del sistema eléctrico chileno, tomando en cuenta la existencia de factores aleatorios o estocásticos que se manifiestan dentro de un horizonte de planificación táctico de 10 años, tales como los afluentes naturales a las centrales hídricas.

El principal objetivo es comparar la forma de resolución actual del problema (SDDP), con un enfoque de resolución por escenarios llamado Progressive Hedging (PH). Además de la comparación de metodologías de resolución se realiza una comparación de generadores de árboles de escenario.

1.2. Estado del Arte

La presencia de variables aleatorias dentro de la formulación del problema genera que éste se torne muy complejo de resolver bajo técnicas tradicionales de optimización, en particular se está trabajando con un problema de optimización estocástica mixta, que al ser del tipo multi-etapa, está asociado a una gran cantidad de escenarios que representan las posibles realizaciones de las variables aleatorias en estudio, lo que conlleva a que su tamaño sea significativo y por ende poco tratable por los solvers actuales (el tamaño de las instancias se pueden ver en la sección 6.4). Dado lo anterior, desde hace más de 30 años se han ido desarrollando una serie de modelos matemáticos que permitan determinar las decisiones óptimas a la hora de planificar la producción eléctrica. Así, en primera instancia se desarrollaron modelos determinísticos, donde todos los inputs son datos invariables a lo largo del horizonte de planificación.

La forma de incorporar estocasticidad al problema de Coordinación Hidrotérmica [4] es mediante la discretización de componentes aleatorias generando así árboles de escenarios.

Este modelo puede representar la realidad con distintos grados de exactitud, dependiente del número de fenómenos que se modela y las variables con que se trabaje. En el caso del problema de forma extensa, el número de variables crece de manera proporcional o en progresión aritmética con el número de escenarios para cada etapa del problema, y de manera exponencial o en progresión geométrica con el número de etapas. Es por esto que el problema para instancias con un gran número de etapas y/o escenarios, se caracteriza como de gran escala y la resolución del problema en su forma extensiva resulta un desafío computacional.

Desde un punto de vista metodológico existen diferentes maneras de abordar el problema. Un modelo multi-nodal, en el cual se describe en detalle las redes de transmisión, representa una buena aproximación al problema real. Por otro lado, un diseño uninodal, es decir,

un modelo donde se produce y consume la energía en un solo nodo (por lo que no tiene redes de transmisión), representa una aproximación alejada de la realidad, pero disminuye significativamente la complejidad del problema (revisado en [4]).

En ambos casos, y principalmente debido a la componente estocástica y a su gran tamaño, se ha vuelto necesario utilizar técnicas de descomposición para abordar el problema. Las técnicas más usadas ocupan la estructura escalonada del problema (es decir, que las cierta cantidad de decisiones de un período se relacionan solamente con el siguiente período).

Debido a que este problema es de interés para los organismos reguladores del sistema eléctrico y compañías generadoras en cada país, hay una amplia variedad de modelos y algoritmos para resolver el problema. Algunos de ellos corresponden a:

1. Optimización de la coordinación hidrotérmica con restricciones de irrigación no-convexas: Problema de pequeña escala del Laja, resuleto con programación dinámica dual estocástica [5].
2. Modelo de Gestión Óptima del Laja (GOL), modelo de coordinación hidrotérmica uninodal y mono-embalse basado en programación dinámica estocástica [6].
3. Modelo de Operación Mensual del Sistema Interconectado Central (OMSIC), modelo de coordinación hidrotérmica uninodal y multi-embalse basado en programación dinámica estocástica [7].
4. Modelo de Programación de Largo Plazo (PLP), modelo de coordinación hidrotérmica multi-nodal y multi-embalse basado en programación dinámica dual estocástica [4].
5. Modelo Operación de Sistemas Eléctricos (OSE2000), modelo de coordinación hidrotérmica multi-nodal y multi-embalse basado en programación dinámica dual estocástica [4].
6. Modelo SEEDS desarrollado en el Centro de Energía FCFM, modelo de coordinación hidrotérmica multi-nodal y multi-embalse basado en programación dinámica dual estocástica [4].

Los modelos anteriormente listados utilizan programación dinámica estocástica (SDP), o bien programación dinámica dual estocástica (SDDP) [8]. Esta última es la que se utiliza actualmente en Chile, y corresponde a una aplicación estocástica de la Descomposición de Bender multietapa [9] y [10]. Esta metodología fue desarrollada en los años 90 para el sistema eléctrico de Brasil y desde entonces se ha vuelto muy popular en países o regiones que cuentan con una importante participación de la hidroelectricidad en su matriz energética.

1.3. Stochastic Dual Dynamic Programming (SDDP)

Desde los 90's la aproximación estándar a la solución óptima en Chile ha sido utilizar Stochastic Dual Dynamic Programming [8], que se basa en la aproximación de las funciones de costos futuros esperados, mediante funciones lineales a trozos. Dichas aproximaciones se obtienen de las soluciones duales de la optimización del problema y se pueden interpretar como cortes de Benders multi-etapa en un problema estocástico [10].

La programación dinámica dual descansa sobre el algoritmo de descomposición anidada de Benders [9], que es una metodología general que permite resolver más eficientemente problemas de optimización de gran tamaño cuyas restricciones tienen una estructura por bloques. La metodología descompone el problema completo en un conjunto de subproblemas aprovechando estructuras de bloque-columna en sus restricciones. Una descripción más detallada de esta metodología y su aplicación a problemas de optimización estocástica multietapa se encuentra en Anexo A y en la sección 2.5.

1.4. Organización de contenidos

El presente trabajo de tesis tendrá el siguiente orden: En el capítulo 2 se mostrará el marco teórico en el que se sustenta la investigación, conceptos necesarios para familiarizarse con la optimización estocástica, una breve introducción a PH y a SDDP. En el capítulo 3 se expone el contexto eléctrico chileno, y los generadores de escenarios estocásticos.

En el capítulo 4 se expone el Modelo de Optimización explicado detalladamente, además de modificaciones en su implementación, mientras que el capítulo 5 profundiza el algoritmo PH, mostrando estrategias de mejoras de la metodología. En el capítulo 6 se exhiben las instancias de implementación y los recursos computacionales utilizados, para luego en el capítulo 7 presentar los resultados en la comparación entre algoritmos SDDP y PH. Por último, se desarrollan las conclusiones y se enuncian los trabajos futuros.

Capítulo 2

Marco Teórico

A continuación se presentan conceptos relativos al algoritmo Progressive Hedging que serán utilizados a lo largo del trabajo. Además se busca formalizar las nociones de programación estocástica más relevantes, por lo que puede ser omitido por un lector familiarizado con el contexto.

El trabajo se enmarca en el contexto de investigación de operaciones. En particular está relacionado con la programación estocástica lineal aplicada a problemas de planificación con múltiples etapas. El objetivo de utilizar programación estocástica es encontrar políticas de operación robustas frente a la incertidumbre. En otras palabras, mediante una heurística se debe buscar un plan de acción (cercano al) óptimo en cada escenario, condicional al desarrollo de la naturaleza, a medida que esta se va revelando en cada etapa. Este plan, se explicita en un árbol de decisión que representa la mejor respuesta (en cada escenario) frente a un futuro no anticipado (el principio de no-anticipatividad se explica con detalle en 2.2).

El objeto de estudio es un problema de planificación eléctrica sujeto a incertidumbre externa (hidrologías naturales). Este tipo de problema, debido a su tamaño y naturaleza estocástica, se vuelve intratable computacionalmente bajo enfoques tradicionales, luego necesita ser abordado mediante una heurística que lidie con la naturaleza del problema: variables lineales en un contexto cuadrático y múltiples etapas. Para esto se utiliza el algoritmo Progressive Hedging descrito en 2.4. Finalmente, la manera de representar la incertidumbre es a través de escenarios (ver sección 2.1) y se aplica la esperanza, como regla de decisión neutral al riesgo. Esta medida se utiliza porque es capaz de incorporar de manera explícita todos escenarios considerados y debido a la naturaleza práctica del problema (desde un punto de vista de tiempo de cálculo); un operador de sistema eléctrico hidrotérmico con capacidad de embalse amplia, debe resolver en muchas ocasiones instancias distintas de este mismo problema.

El problema general de programación estocástica se puede formular como sigue:

$$\begin{array}{ll} \underset{x_s}{\text{mín}} & \sum_{s \in S} p_s f_s(x_s) \end{array} \quad (2.1)$$

$$\text{s.t.} \quad x_s \in Q_s \quad \forall s \in S \quad (2.2)$$

En este problema existen variables de decisión x_s para cada escenario $s \in S$, la función objetivo es la esperanza (en la ecuación 2.1), donde p_s representa la probabilidad de ocurrencia del escenario s y las restricciones (conjunto Q_s) son las propias para cada escenario (en la ecuación 2.2). Como se verá más adelante, en la sección relacionada a no-anticipatividad (revisado en detalle en 2.2), en este tipo de formulación el concepto de no-anticipatividad se encuentra de manera implícita en el problema. Esta formulación se denomina “Formulación Extendida Explicita” (ver subsección 2.3).

2.1. Árboles y Escenarios

El concepto de Árbol de Escenario resulta importante a la hora de considerar problemas de programación estocástico y aporta claridad al momento de definir el concepto de no-anticipatividad, en el que se basa PH. Un *escenario* puede ser definido como la representación de la posible evolución de un sistema hacia un estado futuro, el escenario mostrará la hipotética situación de cada parámetro constitutivo de un sistema para cada período de un determinado horizonte de planificación. La división del horizonte de planificación no necesariamente se tiene que hacer en periodos iguales, en las instancias implementadas en este trabajo, varía según el largo del mes.

Bajo este contexto, un árbol de escenarios corresponde al conjunto de escenarios de un problema que mantiene una estructura de precedencia para cada etapa, un escenario se define como un posible estado experimentado por el sistema completo en estudio, caracterizando todos los valores numéricos de los parámetros relevantes para la toma de decisiones, incorporando la incertidumbre a partir de las probabilidades de ocurrencia que cada uno de estos tendrá asociado. Es decir, un escenario es una tupla compuesta de un conjunto de valores numéricos determinados para todas las fuentes de incertidumbre, para todas las etapas temporales.

Ejemplo 2.1.1 En un problema con horizonte de planificación de 4 etapas ($t \in \{0, 1, 2, 3\}$), con dos fuentes de incertidumbre X_t, Y_t . Sean los parámetros aleatorios X_1, X_2, X_3, Y_1, Y_2 e Y_3 , correspondientes a las posibles realizaciones de cada v.a. en cada etapa, luego el escenario i se define como la siguiente tupla:

$$S_i = \{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)\} \quad \forall i \in S \quad (2.3)$$

Cada uno de estos escenarios tiene una probabilidad asociada. Esta probabilidad se calcula a partir de las distribuciones asociadas a las variables aleatorias y, en el caso que se consideran independiente, la probabilidad del escenario simplemente corresponde al producto de las probabilidades para cada variable aleatoria.

Usualmente se cuenta con información histórica sobre alguna variable (en el caso de este trabajo se cuenta con la hidrología históricas de los últimos 50 años) o se estima de alguna manera la distribución de las mismas. En el caso de contar con variables aleatorias continuas, por lo habitual, se realiza una discretización para, de esta forma, obtener una estructura de

árbol tradicional. Del mismo modo, el intervalo de tiempo entre cada una de las etapas puede ser distinto, pudiendo así representar en un inicio periodos más cortos (menos incertidumbre en las primeras etapas) para luego en las etapas finales, contemplar periodos de tiempo mayores, con tal de generar una mayor agregación de los parámetros y variables, pues la incertidumbre aumenta en la medida que se avanza en el tiempo, existiendo mayor error de predicción en la medida que se estime el valor de un parámetro en periodos más distantes al actual.

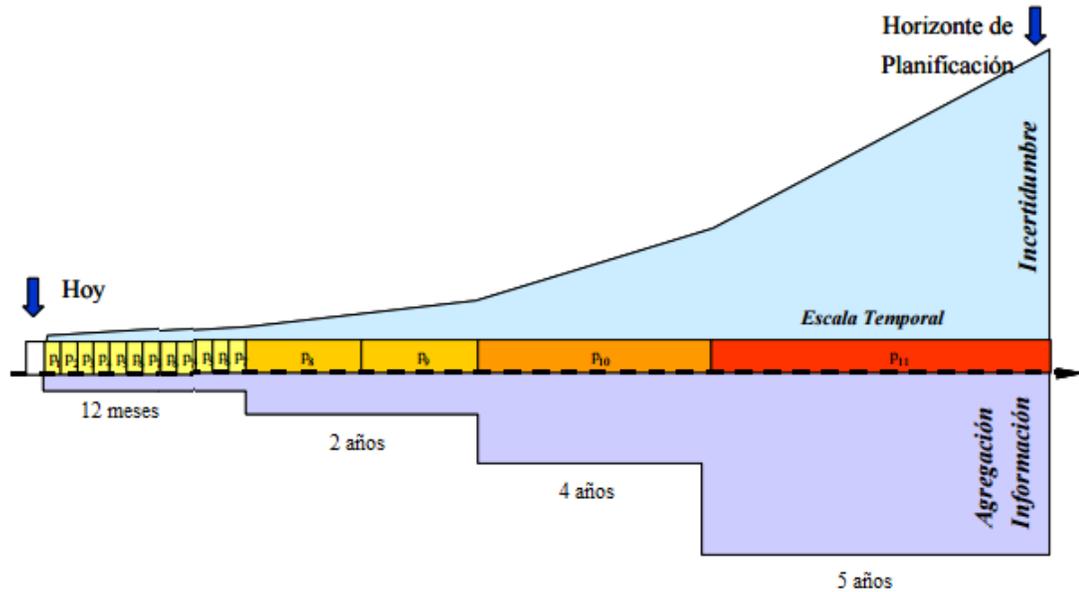


Ilustración 2.1: Largo de períodos vs incertidumbre [11].

En el período inicial (etapa 0), los parámetros sujetos a incertidumbre son idénticos para todos los escenarios recreados, lo que permite llevar a cabo la construcción de lo que se conoce como *Árbol de Escenarios*. Éste consiste en la representación compacta del conjunto de escenarios que se han generado para trabajar el sistema en estudio, donde el primer nodo (raíz) corresponde al estado actual del sistema, y luego, cada rama que se derive de éste corresponderá a una posible evolución del mismo, para el siguiente periodo. Esto se repite de forma sucesiva hasta alcanzar el número de escenarios deseados a modelar dentro del horizonte de estudio específico.

Su nombre deriva directamente de su condición de grafo conexo y acíclico, donde la raíz representa el estado actual del sistema, y luego cada uno de los nodos posteriores determinará el estado de las variables y parámetros estocásticos para uno o más escenarios. Así, un escenario corresponderá al conjunto de nodos desde la raíz hasta las ramas terminales, tomando un único nodo por periodo.

Ejemplo 2.1.2 Para un problema similar al ejemplo anterior, pero caracterizado por tres

periodos y tres escenarios $i, t \in \{1, 2, 3\}$ con valores en común definidos por:

$$S_1 = \{(X_1, Y_1), (X_2^a, Y_2^a), (X_3^a, Y_3^a)\} \quad (2.4)$$

$$S_2 = \{(X_1, Y_1), (X_2^a, Y_2^a), (X_3^b, Y_3^b)\} \quad (2.5)$$

$$S_3 = \{(X_1, Y_1), (X_2^c, Y_2^c), (X_3^c, Y_3^c)\} \quad (2.6)$$

Luego, el árbol de escenario que representa el ejemplo anterior se puede observar en la ilustración 2.2 (izquierda):

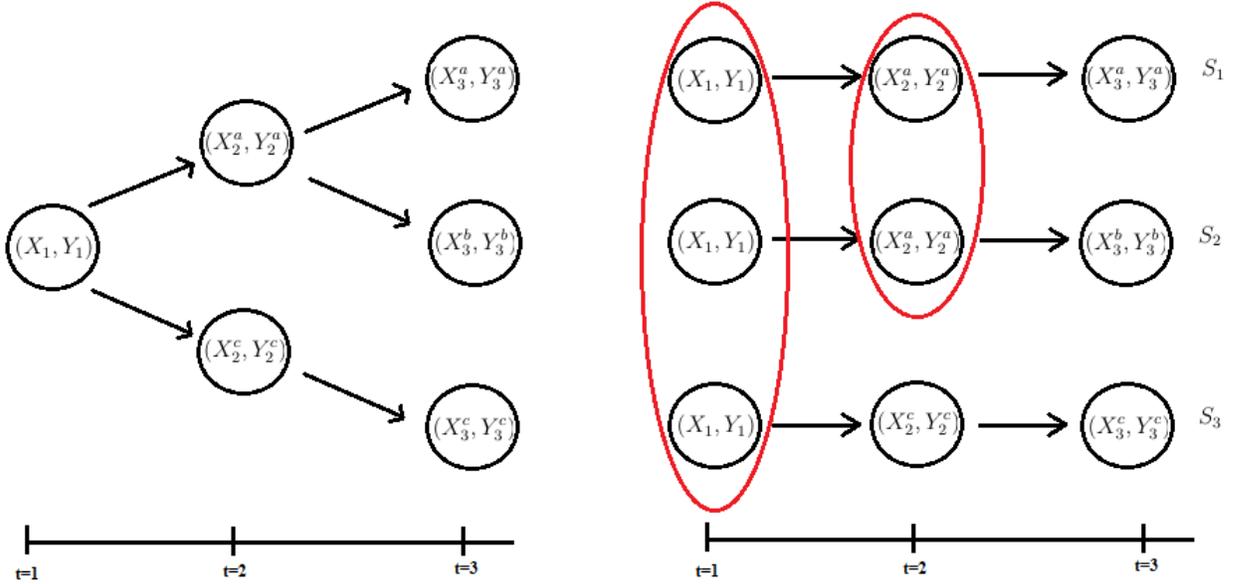


Ilustración 2.2: Árbol de escenario y formulación extendida.

Como es usual, los tres escenarios comparten el nodo inicial, teniendo igual valores para el primer período, asimismo los escenarios S_1 y S_2 tienen valores idénticos para la segunda etapa. Se puede observar en la figura anterior como definen un árbol de escenarios a la izquierda, mientras que a la derecha se muestra la *formulación extendida*. Los nodos encerrados por las elipses tienen idéntico valor para las variables aleatorias. Esto se conoce como conjuntos de información.

De esta manera se introduce la incertidumbre dentro del modelo. Además se comienza a ver la relación entre la estructura de árbol de escenarios y las variables de decisión. Es así como se incorpora el concepto de *indistinguibilidad*, se dice que dos escenarios son indistinguibles hasta la etapa t si todos sus parámetros aleatorios hasta dicha etapa tienen el mismo valor.

Por lo tanto, si dos escenarios son indistinguibles hasta cierta etapa, en el ejemplo los escenarios S_1 y S_2 hasta el período 2, esto tiene también consecuencias sobre las decisiones que se toman para esos escenarios hasta dicha etapa.

2.2. Principio de no-anticipatividad

Uno de los principios más relevantes dentro del contexto de la optimización estocástica donde se utiliza la técnica de Árboles de Escenarios es el principio de no-anticipatividad, fue formulado por Rockafellar y Wets [11], y se hace cargo de la relación entre la estructura de un Árbol de Escenarios y las variables de decisión asociadas al problema. El cual señala que:

Definición 2.2.1 “Si dos escenarios distintos son indistinguibles hasta cierta etapa dentro del horizonte de planificación, entonces los valores de las variables de decisión para dichos escenarios también debe ser igual hasta esa etapa”.

El principio de no anticipatividad hace bastante sentido desde el punto de vista intuitivo, pero presenta profundas implicancias tanto a nivel teórico como de formulación de problemas de optimización estocástica.

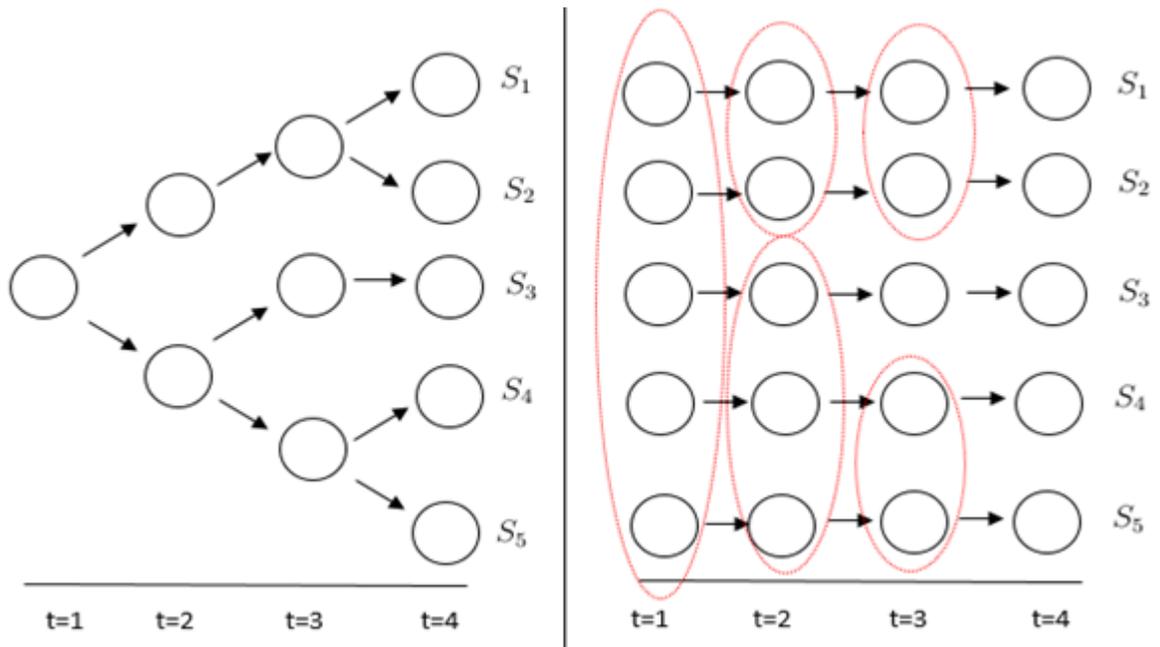


Ilustración 2.3: Representación de principio de no anticipatividad, los nodos encerrados en elipses se denominan conjuntos de información.

Así, la condición necesaria para que las decisiones de los distintos escenarios hasta una determinada etapa sean las mismas se observa gráficamente en la ilustración 2.3, los nodos rodeados por las elipses punteadas indican qué decisiones deben ser idénticas en cada una de las etapas en estudio (restricciones de no-anticipatividad), dada la realización de las diversas variables aleatorias. Definiendo estos conjuntos de información que corresponde a grupos de nodos del árbol de decisión, como se puede ver en la ilustración 2.3, izquierda. Al contar con este principio se puede realizar una nueva formulación del problema, escribiendo de manera desacoplada las decisiones para los escenarios (ya sin utilizar la formulación implícita dada

en un árbol) y posteriormente incorporar las restricciones de no-anticipatividad.

Precisando algunas nociones para los árboles y sus escenarios, se denota al súper-conjunto N que contiene a todos los conjuntos de información presentes en el árbol de escenarios, mientras que cada uno de sus elementos (conjuntos de información) se indexan según la etapa correspondiente del problema y la cantidad de conjuntos de información de la misma N_t . Además $N_t \cap N$ corresponde al conjunto de información para la etapa t . Es claro que N_t puede contener más de un conjunto de información. Algunas propiedades son:

1. **Todo escenario pertenece a lo menos a un nodo:**

$$\forall s \in S \exists t \in T : s \in N_t$$

2. **La cardinalidad de los nodos es decreciente con respecto a tiempo:**

$$|N_t| \geq |N_{t+1}| : t \in T \quad N_t \in N \quad 1 \leq |N_t| \leq |S|$$

3. **En la etapa inicial existe un único conjunto de información que contiene a todos los escenarios:**

$$N_1 = S$$

4. **Cada conjunto de información será singleton en la etapa final:**

$$|N_T| = 1 \quad \forall N_T \in N$$

2.3. Formulación extendida

Tomando como base el principio de no-anticipatividad, el problema de programación estocástica se puede formular de diferentes maneras:

Explícita: En esta formulación, se presentan las variables indexadas según escenarios, con la condición de no anticipatividad como restricción explícita del modelo.

- $\sum_{N_t \in N} |N_t|$ Variables según decisión, a lo más S por etapa.

- $\sum_{t \in T} \sum_{N_t \in N} (|N_t| - 1)$ Restricciones por decisión, para expresar la no-anticipatividad.

Por lo que un problema en su formulación explícita se puede escribir como sigue:

$$\min_{x_s} \quad \sum_{s \in S} p_s f_s(x_s) \quad (2.7)$$

$$\text{s.t.} \quad x_s \in Q_s \quad \forall s \in S \quad (2.8)$$

$$x_s^t = x_{s'}^t \quad \forall s, s' \in N_t, \forall t \in T \quad (2.9)$$

Compacta: En ésta se presentan las variables según conjuntos de información.

$-\sum_{N_t \in N} 1$ Variables según decisión.

- No se necesitan restricciones de no anticipatividad, puesto que las variables las implementan de forma implícita.

Lógicamente la formulación explícita presentará un peor desempeño que la compacta en la medida que aumente el número de escenarios incluidos, debido al mayor tamaño que ésta presenta, tanto en filas como en columnas.

2.4. Introducción a Progressive Hedging

El algoritmo Progressive Hedging PH (Rockefeller y Wets, 1989 [12]), es un algoritmo de descomposición para problemas estocásticos con estructura de árbol de escenarios, tiene un resultado exacto en el sentido que converge al óptimo global para problemas convexos. Su aplicación es favorable para problemas de gran tamaño, en los que su formulación extendida, es intratable directamente, luego se descompone en problemas más pequeños para su resolución.

2.4.1. Intuición

La intuición detrás del algoritmo es que si se pudiese resolver el problema estocástico original a partir de resolver cada escenario por separado, y posteriormente tomar esperanza de la solución obtenida, solo se resolverá un total de $|S|$ problemas. Dichos problemas serán sencillos y determinísticos, lo que permite un mayor rendimiento en términos computacionales.

El cambiar la minimización por el cálculo de la esperanza de los escenarios, no es directo, dadas las restricciones de no-anticipatividad descritas en la forma explícita de la sección anterior (ecuación 2.9), ya que éstas acoplan las decisiones a tomar entre los distintos escenarios.

Ahora si se observa el problema en su forma explícita que para cada solución factible, debido a la restricción de no-anticipatividad, se tiene que:

$$x_s^t = x_{s'}^t \quad \forall s, s' \in N_t, \forall t \in T \quad (2.10)$$

$$= \mathbb{E}[x_s] \quad (2.11)$$

Lo que equivale a señalar que es posible reformular la restricción de no-anticipatividad notando que todas las variables que se encuentran en cada uno de los nodos del árbol de escenarios son iguales, y por lo tanto, iguales a la esperanza de cada uno de los conjuntos de información.

Dado lo anterior, es posible desacoplar la dependencia inherente entre los diversos escenarios utilizando la técnica de relajación lagrangeana sobre este grupo de restricciones. Se

supondrán conocidos los multiplicadores asociados (los penalizadores $\widehat{\omega}_s$ que penalizan la función objetivo en la medida que la solución se desvíe de la restricción de no-anticipatividad), lo que implica que la función objetivo se convierte en:

$$\min_{x_s \in Q_s \forall s \in S} \sum_{s \in S} p_s f_s(x_s) + \sum_{s \in S} \widehat{\omega}_s (x_s - \mathbb{E}(x_s)) \quad (2.12)$$

Luego, reagrupando términos y cambiando la notación ($\widehat{\omega}_s = p_s \omega_s$ y $\mathbb{E}(\omega) = 0$), se obtiene:

$$\min_{x_s \in Q_s \forall s \in S} \sum_{s \in S} p_s \{f_s(x_s) + \omega_s x_s\} \quad (2.13)$$

De esta manera se logra que 'mín' y 'E' sean intercambiables, es decir se puede descomponer el problema en una serie de problemas más pequeños:

$$\sum_{s \in S} p_s \{ \min_{x_s \in Q_s} f_s(x_s) + \omega_s x_s \} \quad (2.14)$$

Sólo resta calcular, iterativamente, los multiplicadores ω_s presentes en la función objetivo. Para asegurar la convergencia y evitar el problema de convertir los sub-problemas en no-acotados se agrega un término proximal cuadrático que fuerza a las variables a moverse en la dirección correcta. Por lo tanto, se procede a utilizar la relajación lagrangeana aumentada, cuya característica principal consiste en el uso de un término cuadrático para llevar a cabo la penalización, la cual corresponde a la diferencia que presenta cada una de las soluciones obtenidas en relación al promedio de las soluciones calculadas en la iteración previa, obteniéndose así finalmente una función objetivo de este estilo:

$$\sum_{s \in S} p_s \{ \min_{x_s \in Q_s} f_s(x_s) + \omega_s x_s + \frac{\rho}{2} (x_s - \bar{x})^2 \} \quad (2.15)$$

Se busca que las variables sean forzadas a ir hacia el promedio \bar{x} . De lo contrario, al minimizar puede ocurrir que un multiplicador ω_s vuelva no-acotada a una variable (con un costo total negativo, en el caso de minimización), llevando a que el problema no converja.

Ejemplo 2.4.1.1 Sea la función objetivo lineal, para algún escenario s :

$$\min_{x_s \in Q_s} f_s(x_s) = c_s x_s$$

Entonces, se puede tener que: $c_s^i + \omega_s^i < 0$ para cierta variable i , por lo que, sin el término cuadrático proximal, el problema se vuelve no acotado:

$$\begin{aligned} \min_{x_s \in Q_s} c_s x_s + \omega_s x_s &= \min_{x_s \in Q_s} x_s (c_s + \omega_s) \\ &> 0 \end{aligned}$$

2.4.2. Pseudo Código PH

A continuación se expone el pseudo código de PH:

Algoritmo Progressive Hedging

- 1: Resolver cada subproblema correspondiente a su escenario $s \in S$: $\min_{x_s \in Q_s} f_s(x_s)$;
- 2: Calcular la solución global de cada nodo: $\bar{x} = \sum_{N_t: s \in N_t} p_s x_s$;
- 3: **if** Las soluciones se parecen lo suficiente $\{\|x - \bar{x}\| < \varepsilon\}$ **then**
- 4: Stop;
- 5: **else** Continuar al paso siguiente;
- 6: **end if**
- 7: Actualizar los penalizadores: $\omega = \rho(x - \bar{x}) + \omega_{i-1}$;
- 8: Resolver cada uno de los escenarios penalizados:
$$\min_{x_s \in Q_s} f_s(x_s) + \omega_s x_s + \frac{\rho}{2} \|x_s - \bar{x}\|^2 ;$$
- 9: Volver a 2 ;

En términos sencillos, el algoritmo busca resolver de forma iterativa cada uno de los escenarios, penalizando en cada iteración con respecto a la diferencia que presenta la solución obtenida en relación al promedio o esperanza de todos los escenarios, es decir, se penaliza con respecto a la distancia que se tiene al óptimo global. Luego, la convergencia del algoritmo se medirá como la distancia euclidiana entre la solución de cada uno de los escenarios y la solución global. El algoritmo finaliza cuando todas las soluciones obtenidas han alcanzado un criterio de cercanía suficiente (iguales o muy cercanas según se establezca con el parámetro ε) en relación a la solución global. Otro criterio de detención es la cantidad de iteraciones, en el caso de no obtener convergencia en un número I de iteraciones, se fijaran las variables que convergieron (o que se encuentren a una distancia menor a ε), y se resolverá el problema en su forma extensiva, para obtener las variables restantes.

2.4.3. Caracterización

Las principales características de Progressive Hedging son:

- Es un algoritmo de descomposición horizontal o por escenarios. Por el contrario, la descomposición vertical corresponde a resolver el problema etapa a etapa. La mayoría de los enfoques anteriores para resolver el problema de coordinación hidrotérmica corresponden a enfoques de descomposición vertical.
- No está limitado por la cantidad de escenarios a considerar, como lo está la formulación extendida que crece mucho más rápidamente en dificultad (tamaño y tiempo de computación) al incrementar los escenarios.

- El algoritmo relaja la restricción de no-anticipatividad, penalizándola en la función objetivo junto a un término proximal cuadrático.
- Posee una demostración basada en Relajación Lagrangeana Aumentada, lo que la hace exacta (converge al óptimo global) en el caso convexo.
- Es una heurística para problemas no convexos (con variables enteras).
- Fácilmente paralelizable, basta resolver cada subproblema en distintos núcleos computacionales, de forma iterativa.
- Su desempeño está fuertemente ligado a lo rápido que se pueda resolver cada subproblema (cada escenario). Puesto que el resto del algoritmo, en términos simples, es calcular promedios de decisiones (\bar{x}) y sumas lineales sobre el mismo penalizador (ω). Para más detalles sobre el algoritmo revisar [13] y [12].

En el capítulo 5 se expone en detalle la metodología PH, profundizando en sus mejoras y ajustes.

2.5. SDDP

El método Stochastic Dual Dynamic Programming (o SDDP) desarrollado por Pereira y Pinto [8] corresponde a un algoritmo para resolver problemas multi-etapa de programación estocástica. Este algoritmo construye soluciones factibles en un ambiente de programación dinámica usando una sub-aproximación de funciones (convexas) de costos futuros.

Esta aproximación es construida a partir de cortes de Benders y ha sido usada en problemas de planificación eléctrica, siendo hoy en día el método estándar para resolver estos problemas en Chile. Debido a esto, SDDP será el punto de comparación para las soluciones obtenidas con PH. A continuación se muestra un ejemplo del funcionamiento del algoritmo. Para más detalles revisar anexo A.

Ejemplo 1.1

A continuación se muestra un problema de coordinación hidrotermica simplificado en dos etapas, donde la segunda etapa presenta incertidumbre reflejada en 3 posibles escenarios y a cada escenario se asigna igual probabilidad de ocurrencia $\frac{1}{3}$:

$$\begin{array}{rcccccc}
\text{mín}_g & \{c_1^T \cdot g_1 & +\frac{1}{3}(c_2^T \cdot g_{21} & +c_2^T \cdot g_{22} & +c_2^T \cdot g_{23})\} & & \\
s.a. & \vdots & \vdots & \vdots & \vdots & & \\
& \sum g_1 & \vdots & \vdots & \vdots & = d_1 & \\
& v_1 - v_0 + K \cdot g_1 & \vdots & \vdots & \vdots & = a_1 & \\
& & \sum g_{21} & \vdots & \vdots & = d_2 & \\
& -v_1 + & v_{21} + K \cdot g_{21} & \vdots & \vdots & = a_{21} & \\
& & & \sum g_{22} & \vdots & = d_2 & \\
& -v_1 + & & v_{22} + K \cdot g_{22} & \vdots & = a_{22} & \\
& & & & \sum g_{23} & = d_2 & \\
& -v_1 + & & & v_{22} + K \cdot g_{22} & = a_{23} &
\end{array}$$

Donde:

$-c_t^T$: Representa el costo variable de generación en periodo t.

$-g_{ts}$: Generación en el período t, en el escenario s.

$-d_t$: Demanda en el período t.

$-v_t$: Volúmen en el período t.

$-a_{ts}$: Afluente natural en el período t, escenario s.

$-K$: Rendimiento de central hídrica.

En este problema se busca minimizar los costos totales de generación para dos etapas, de tal manera de satisfacer la demanda objetivo d_t . Además se tienen restricciones para el caudal turbinado, en donde se muestra que el total turbinado no debe superar las reservas de aguas en los embalses v_t más el afluente natural a_{ts} .

Un subproblema se crea a partir del conjunto de variables, función objetivo y restricciones asociadas a un bloque, estableciendo un encadenamiento de subproblemas maestro-esclavo. Esto se hace de manera que en un subproblema no aparezcan variables asociadas a un esclavo, y en este último las variables asociadas a un maestro se transforman en parámetros (para el ejemplo que se muestra a continuación, la variable v_1 se entrega como parámetro). El problema que se muestra a continuación representa el problema de segunda etapa para un escenario i (problema esclavo):

$$y_i(v_i) = \min_{g_{2i}} \{c_2^T \cdot g_{2i}\} \quad (2.16)$$

$$s.a. \sum g_{2i} = d_2 \quad (2.17)$$

$$K \cdot g_{2i} = a_{2i} + v_1 \quad (2.18)$$

Cada subproblema es resuelto de forma individual, ligándose iterativamente con su esclavo a través de un punto solución y con su maestro a través de los denominados cortes de Benders (el problema de ejemplo, solo tiene dos etapas, por lo que el problema de 2da etapa no tiene un esclavo asociado). Una vez resuelto un subproblema, dado un punto solución del maestro, si éste resulta factible se construye un corte de optimalidad que es agregado al maestro como una restricción. Dicha restricción constituye una aproximación del valor óptimo del subproblema esclavo en la función objetivo del maestro. Si el subproblema esclavo resulta infactible, se construye un corte de factibilidad que limita la región factible del subproblema maestro de manera que su resolución no arroje un punto que provoque la infactibilidad del esclavo.

Luego, el subproblema maestro a resolver se podría escribir de la siguiente forma general:

$$\min_{g_1} \{c_1^T \cdot g_1\} \quad (2.19)$$

$$s.a. \sum g_1 = d_1 \quad (2.20)$$

$$v_1 + K \cdot g_1 = a_1 + v_0 \quad (2.21)$$

$$y \geq \frac{1}{3} \sum_{i=1}^3 O_{ij}(v_1), \quad j = 1, \dots, J \quad (2.22)$$

$$F_k(v_1) \geq 0, \quad k = 1, \dots, K \quad (2.23)$$

La expresión de los cortes de optimalidad O_{ij} y factibilidad F_k son funciones lineales afines y son calculados a partir de la solución primal y dual de los subproblemas esclavos. El corte de optimalidad O_{ij} , es el corte j obtenido de la solución del subproblema esclavo i . El corte de factibilidad F_k es el corte k obtenido de algún subproblema esclavo que resultó infactible.

Para tener una explicación en detalle de como funciona el algoritmo, revisar anexos A.

Capítulo 3

Problema eléctrico

3.1. Antecedentes y descripción

El problema eléctrico a resolver, se desarrolla en el marco del sistema eléctrico chileno. Debido a la importancia que tiene en la economía, el sistema cuenta con un planificador central, quien se encarga de definir los niveles de embalses y producciones en cada central eléctrica. Por lo que se diferencia de un mercado eléctrico competitivo, donde las decisiones de tipo operacionales son tomadas por las empresas que buscan maximizar sus beneficios.

Para el caso en que existe una coordinación central, las empresas generadoras solo deben efectuar las tradicionales funciones relacionadas con la operación de sus plantas: partida, parada, mantenimientos, entre otras.

Como se expuso en la sección introductoria, existe un total de 386 generadores eléctricos y 1.163 líneas de transmisión, considerando el SIC y SING [2] [3], los principales tipos de centrales son Térmicas e Hídricas, siendo estas últimas de gran interés, ya que además de tener costos de operación (comparativos) bajos, pueden acumular energía en forma de agua en las centrales de tipo embalse. Para conocer más sobre los diferentes tipos de centrales revisar Anexo B.

Haciendo una descripción en el contexto eléctrico, a lo largo de Chile se despliegan líneas de transmisión, que son unidas mediante nodos. Dichos nodos (o barras en el lenguaje eléctrico) pueden representar:

- Demandas: Todas las demandas eléctricas domiciliarias e industriales, son simplificadas y agrupadas en puntos de demanda (nodos), por lo que no se desarrolla el problema de distribución menor (llevar los niveles de voltaje más reducidos).
- Generadores eléctricos: Algunos nodos representan generadores eléctricos, cabe notar que en un mismo nodo se puede agrupar más de un generador.

- **Nodos pivotes:** Este tipo de nodos, representan la unión de líneas de transmisión, pero no están relacionados con demandas ni con generadores.

Cabe señalar, que un mismo nodo, puede tener asociado generadores eléctricos y una demanda a la vez (un ejemplo simple de esto es el caso Uninodal). En el Lenguaje de Teoría de grafos, la red eléctrica anteriormente descrita se puede definir como un grafo (típicamente) cíclico y conexo, la ilustración 3.1 lo representa:

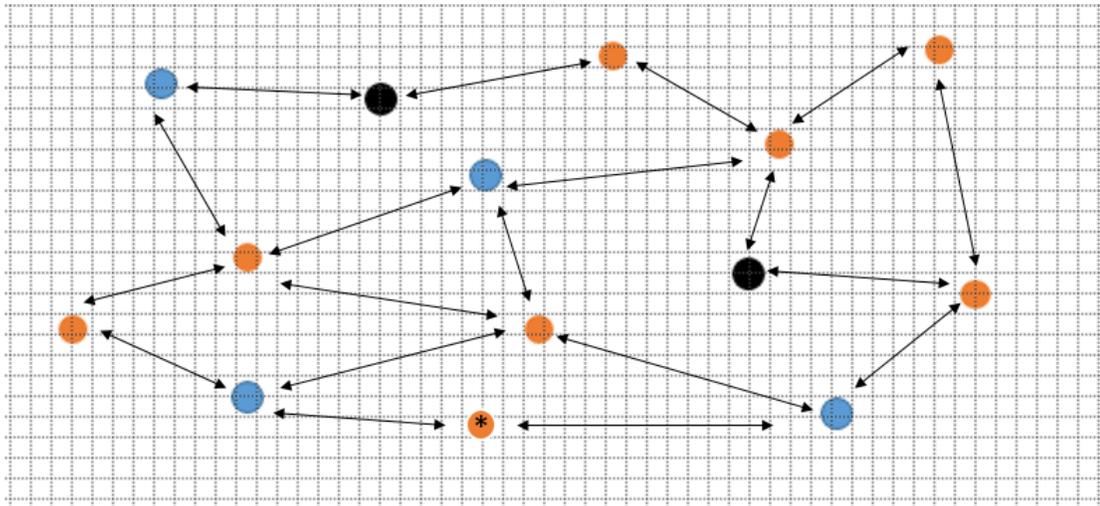


Ilustración 3.1: Representación de Red eléctrica y sus nodos.

En donde los círculos anaranjados representan generadores eléctricos, los círculos azules representan demandas, mientras que los negros son nodos pivotes.

Por otro lado, en el sistema eléctrico existe un modelo subyacente, que es representado por la red hídrica que conforman los generadores Hidroeléctricos. Esta red hídrica obedece a la naturaleza fluvial de las aguas. Si una central A está río arriba de otra central hídrica B , el caudal que pasa por A posteriormente alimentará a la central B .

Los distintos tipos de afluentes hídricos que pueden alimentar a una central son:

- **Afluentes naturales:** Corresponden a irrigaciones naturales que llegan a las centrales y que no provienen de una central anterior.
- **Vertimientos:** Las centrales hidroléctricas, pueden decidir derramar los excedentes hídricos no utilizados (en la medida que existan las instalaciones que lo permitan), vale decir, si una central de pasada necesita un total de $q[m^3]$ para satisfacer la demanda, y el afluente natural que la alimenta entrega una cantidad $Q[m^3]$ (donde $q < Q$), entonces el exceso de caudal será derramado ($Q - q$). El vertimiento también se puede dar al superar la cota máxima de un embalse, pero

este caso es muy poco común.

- Caudal Turbinado: el caudal turbinado corresponden al caudal generado por una central aguas arriba, que luego alimenta a una central aguas abajo.
- Filtraciones: Las filtraciones corresponden a una particularidad que se da en las centrales tipo embalses, en donde el agua embalsada se filtra a través del suelo (típicamente en suelos rocosos), que posteriormente alimenta a las centrales aguas abajo.

Nuevamente basándose en Teoría de grafos, para instancias de gran tamaño, la red hídrica forma un “Bosque”¹ o un grafo acíclico dirigido, no conexo.

Luego, basándose en la figura anterior, y quitando todos los nodos que no representan centrales hídricas (el nodo en la ilustración 3.1, marcado con asterisco representa una central no-hídrica), agregando además, los afluentes naturales (nodos en verde) y dirección de caudales por central. La red hídrica queda representada por la ilustración 3.2.

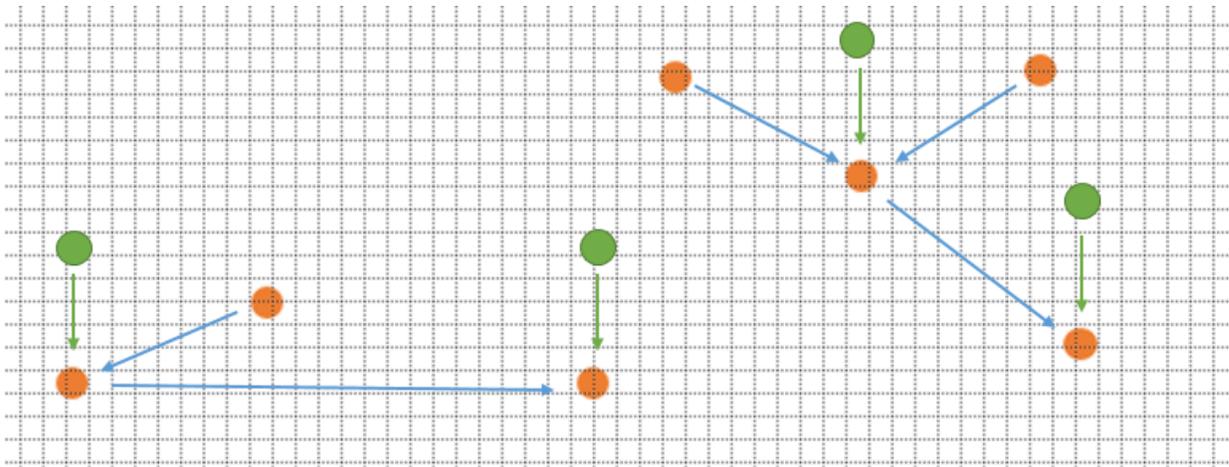


Ilustración 3.2: Representación de Red hidráulica, con afluentes naturales y sus nodos.

Otro punto a tener en cuenta en el problema eléctrico, es que la demanda se considera determinística (debido a la dominancia de la estocasticidad hídrica, en el horizonte de duración del problema), restando así complejidad al problema y haciéndolo abordable desde el punto de vista computacional. Por lo tanto, la única fuente de incertidumbre para generar los escenarios, son los afluentes naturales.

Para revisar en más detalle la notación que se utiliza para las redes hidráulicas, en anexos C se expone el modelo de flujos de la Cuenca del Laja (cuenca utilizada para resolver un modelo simplificado en la instancia de menor tamaño).

¹Un bosque es definido por un conjunto de árboles. Un árbol se define como un grafo acíclico dirigido.

3.2. Reducción de sub-problema bloque-etapa

Para modelar el comportamiento de la demanda y poder adaptar de forma más simple sus fluctuaciones, se procede a realizar una división de las etapas en bloques temporales más pequeños. En la ilustración 3.3 (a) se puede ver el comportamiento de la potencia demandada en el tiempo, y su periodicidad por efecto del consumo en el día y la noche, dicha curva recibe el nombre de Curva de Carga ².

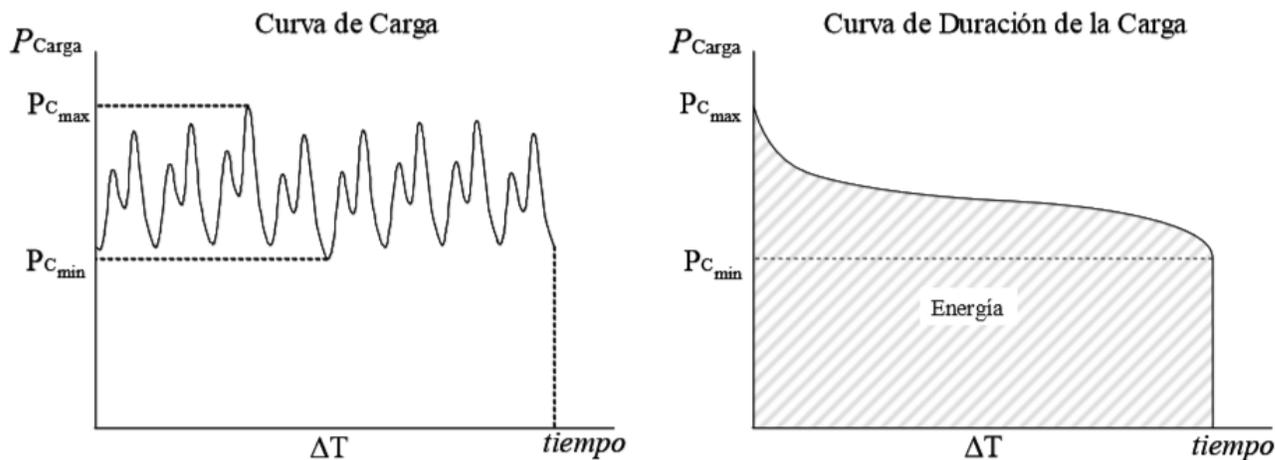


Ilustración 3.3: Representación de la carga en el tiempo (8 días) (a) Potencia demandada en cada instante de tiempo. (b) Cantidad de tiempo que la potencia demandada se encuentra sobre un nivel.

En la ilustración 3.3 (b) se muestra la misma demanda, pero ordenada de forma decreciente con respecto al tiempo, esta curva se conoce como Curva de Duración de Carga. Si bien esta representación es más simple que (a), aun no es útil ya que la potencia presenta un valor continuo en el tiempo. Por lo tanto se procede a aproximar dicha curva a una función constante por tramos, con potencia y duración prefijada en cada uno de los tramos, denominados bloques de demanda, como se muestra en la ilustración 3.4. Mientras mayor sea el número de bloques más precisa será la aproximación.

²Las ilustraciones utilizadas en esta subsección fueron extraídas de [4]

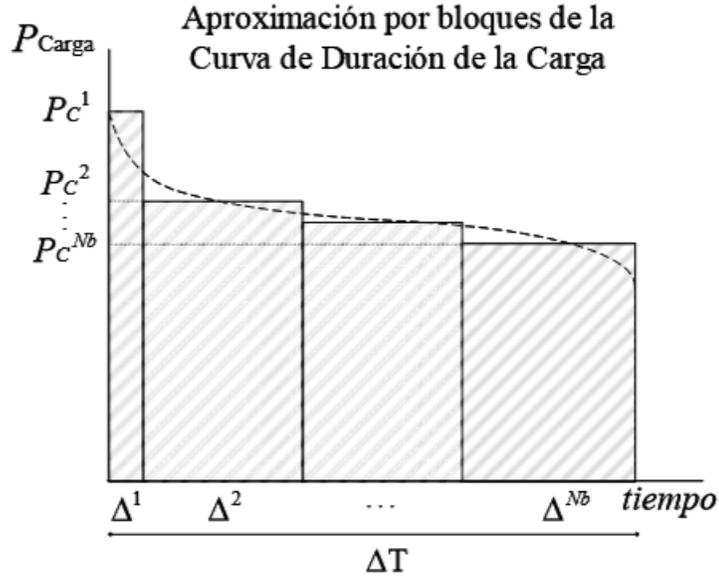


Ilustración 3.4: Representación aproximada por bloques de demanda de la curva de duración de la carga.

Tres requisitos importantes deben satisfacer la aproximación por bloques; El primero, es que al menos un bloque debe representar la demanda más alta del período, o demanda punta, ya que de esta manera se puede estudiar si la capacidad máxima del sistema, puede suplir la demanda más alta. Esto se consigue haciendo que el primer bloque tenga una potencia igual a la máxima demanda. El segundo requisito es que la energía total representada, sea la misma. Por último, que el largo total del período, sea igual a la sumatoria de los bloques que lo componen. Formalmente:

$$P_C^1 = P_{Dmax} \quad (3.1)$$

$$\sum_{b=1}^{Nb} \Delta^b = \Delta T \quad (3.2)$$

$$\int_{t \in \Delta T} p(t) \cdot dt = \sum_{b=1}^{Nb} P_C^b \Delta^b \quad (3.3)$$

Donde $p(t)$ representa la Curva de Duración de Carga, ΔT es el largo total del período. Además, Nb simboliza la cantidad total de bloques, Δ^t la duración total del bloque b y P_C^b representa la potencia del bloque b .

3.3. Incertidumbre

Como se ha mencionado en las secciones anteriores, la única fuente de incertidumbre consideradas dentro del el problema de coordinación hidrotérmica, son los distintos afluentes que irrigan las centrales hidráulicas. Dichos afluentes dependen directamente de la cantidad de agua caída en las épocas de lluvia, y el deshielo en el verano.

Es por esto que un año de sequía condiciona enormemente la operación del sistema eléctrico. El objetivo principal de la coordinación hidrotérmica es anticiparse a años de sequía (o años lluviosos), logrando almacenar recurso hídrico en los embales para palear el bajo caudal afluente. Si se logra anticipar los años de sequía, se estarán disminuyendo significativamente los costos de producción, ya que no se recurrirá a la generación exhaustiva con plantas térmicas (generación de alto costo).

Actualmente se cuenta con la información histórica de caudales afluentes desde 1960³. Para ilustrar el comportamiento de los caudales a través de los años, la siguiente imagen muestra las fluctuaciones del afluente del Laja, desde 1960 hasta 2013:

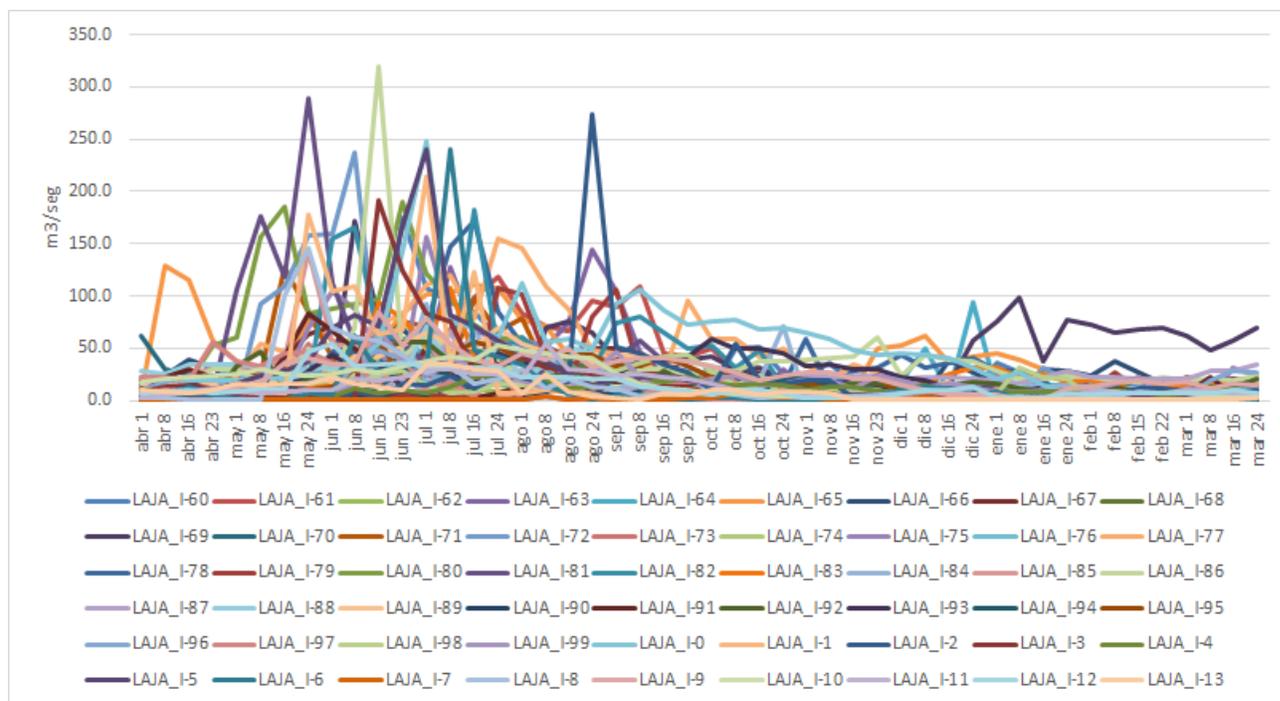


Ilustración 3.5: Hidrologías Laja 1960-2013.

Se puede observar en la ilustración 3.5 que los meses donde el caudal es mayor (meses de lluvia), corresponde a Mayo, Junio, Julio y parte de agosto.

³Cabe notar que en una buena cantidad de centrales se hicieron aproximaciones de los caudales afluentes reales, con su promedio, ya que no se contaba con información exacta.

Como se pudo ver en la sección 2.1, la forma de abordar la incertidumbre es mediante la formulación de Árboles de Escenario. Para construir dichos árboles, se cuenta únicamente con la información histórica de afluentes. Para ejemplificar como se comporta la variabilidad de los afluentes, la ilustración 3.6 muestra el año con menor y mayor caudal para el Laja (2007 y 2000 respectivamente).

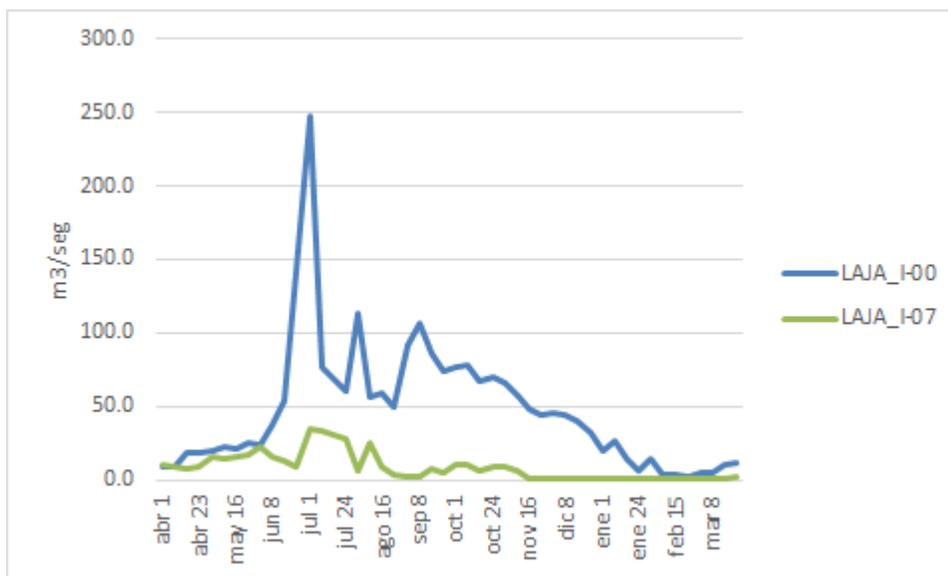


Ilustración 3.6: Mínima y máxima hidrología Laja 2007 - 2000.

Al ver la significativa diferencia entre dichos años, es imperativo desarrollar un generador de árboles de escenario que pueda capturar dicho comportamiento.

La sección 3.4 trata sobre las metodologías de generación actual de escenario y además muestra propuestas innovadoras sobre generadores de escenario.

3.4. Generación de Escenarios

En Chile, tradicionalmente los modelos para la incertidumbre hidrológica usados en el sector eléctrico, utilizan para su construcción la información contenida en una base de datos que almacena el registro histórico de los caudales afluentes semanales promedio, para cada una de las centrales hidroeléctricas. Actualmente la estadística utilizada en el SIC contiene la información de caudales entre los años 1960 y 2010. Por indicación del Artículo 190 de la ley general de servicios eléctricos, D.S N° 327/1997: *“...la estadística para los gastos afluentes en centrales hidroeléctricas deberá ser representativa de las respectivas centrales y considerar, como mínimo, una muestra de 40 años hidrológicos. Esta estadística deberá ser actualizada periódicamente, de modo que siempre incluya los datos correspondientes al año hidrológico*

anterior al que precede a aquel en que se esté efectuando la programación de la operación” [14]. Por lo tanto, se hace obligatorio el uso de las hidrologías históricas, para la generación de los escenarios hidrológicos.

Las series de caudales contenidas en la estadística utilizada, se han obtenido por mediciones realizadas en las distintas cuencas del país. Sin embargo, para algunos afluentes no se cuenta con mediciones directas de los caudales, habiendo sido obtenidos por métodos indirectos, a partir de adición y substracción de otros afluentes de la misma cuenca [7].

3.4.1. Metodologías de generación de escenarios actuales (Generador de escenarios PLP)

El método generador de escenarios descrito en [15] y empleado en el SIC para resolver el problema de coordinación hidrotermica estocástico, puede ser separado en dos etapas: Selección de secuencias y selección de aperturas. El primer paso consiste en definir, para cada año en la data histórica, una secuencia completa de hidrologías, para todo el periodo de estudio. Luego, si el horizonte de planificación es de dos años, se define un total de N secuencias iniciales, concatenando las secuencias por años sucesivos, por ejemplo, la primera secuencia será definida por los años 1960-1961, la segunda por los años 1961-1962, y así sucesivamente hasta la última secuencia hidrológica, definida por la hidrología de 2010-1960.

Luego, para cada una de estas secuencias, en cada una de las etapas, se considera un total de M aperturas para capturar la estocasticidad en cada etapa. En la ilustración 3.7 se representa cómo los escenarios son generados para PLP. Las líneas azules representan las secuencias centrales, mientras que las líneas rojas representan las aperturas en cada nodo.

En la instancia central con la que se trabajó, el número inicial de secuencias es igual al número de años de la data histórica ($N=50$), y el número de aperturas en cada etapa es el mismo ($M=50$).

Para poder resolver este enfoque de generación de escenarios, se debe hacer una descomposición por etapas. Para cada etapa se resuelve un número de problemas igual a la cantidad de aperturas en el nodo (para este caso, se generan aperturas igual a las 51 realizaciones históricas). Luego, la solución obtenida en cada apertura, ponderada por su probabilidad, representan la solución óptima en el nodo.

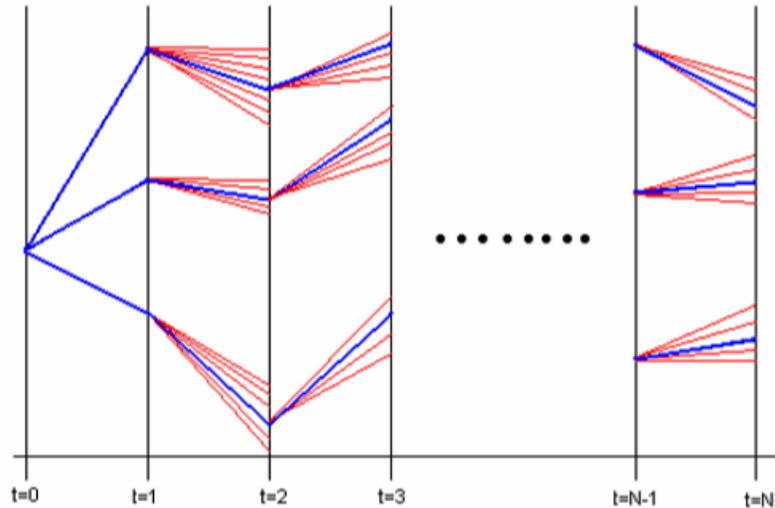


Ilustración 3.7: Diagrama de escenarios usado en PLP. Ilustración extraída de [14]

Una de las características centrales de este método generador de escenarios es que sólo considera N escenarios para todo el horizonte de planificación, y usa las aperturas para incluir la incertidumbre en cada etapa. Por lo tanto, este generador es adecuado para métodos de optimización basados en la descomposición vertical (por etapas), como Benders o SDDP.

Uno de los principales problemas de esta metodología es que no toma en cuenta la correlación temporal entre períodos consecutivos, ya que considera todas las realizaciones de la data histórica para construir las aperturas, esto puede generar aperturas muy poco probables (como se puede ver en la ilustración 3.9), consiguiendo sobre-restringir el problema, generando soluciones sub-óptimas. Adicionalmente, la probabilidad de que una hidrología real sea igual a una hidrología histórica es muy baja.

Muchos estudios han sido desarrollados para mejorar la generación de escenarios en el problema de coordinación hidrotérmico chileno. En el estudio realizado por Eugenio Palacios [14] la distribución obtenida por un modelo PARMA (periodic autoregressive-moving-average) es usado para asignar probabilidad a cada observación en la data histórica, y luego estas probabilidades son consideradas para realizar muestreos y generar las aperturas en cada etapa. Otra alternativa expuesta en [15], donde se utiliza un método de clustering para encontrar las hidrologías históricas que mejor representen la data en su totalidad. Luego, estas hidrologías son utilizadas para generar aperturas en cada etapa y secuencia.

Los métodos descritos más arriba son incontables con algoritmos de descomposición por escenario, que requieren escenarios para todo el periodo de planificación. Una alternativa para abordar esto nace intuitivamente, y consiste en considerar todas las posibles combinaciones de la información histórica, como se describe en [16]. Un ejemplo de esta aproximación es expuesto en la Ilustración 3.8 donde los escenarios para la central Ralco, son generados mediante este método. Notar que por simplicidad, para este ejemplo, solo se han trazado 4 ramificaciones para la primera etapa, y desde esa etapa, se han considerado todas las posibles

combinaciones para las etapas siguientes.

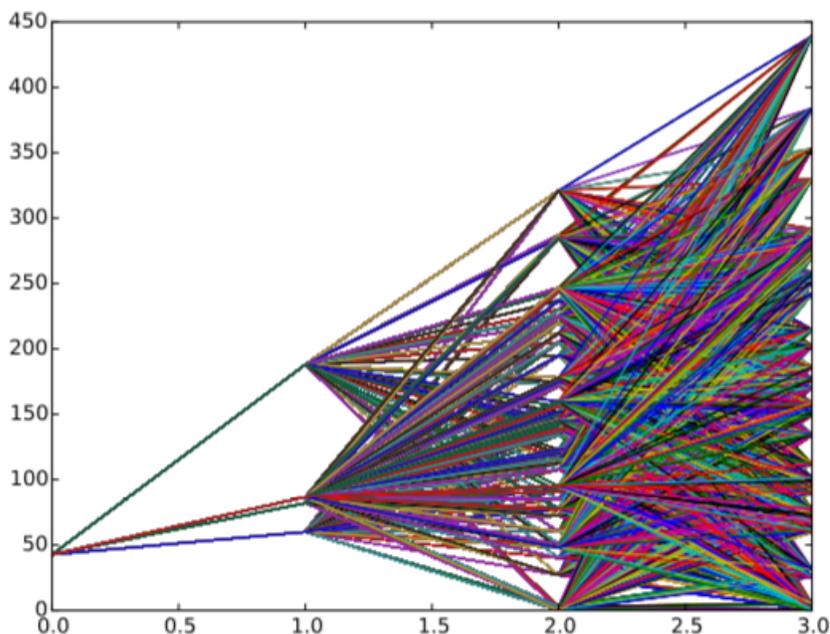


Ilustración 3.8: Ejemplo del generador de escenarios que considera todas las posibles combinaciones.

Es claro que este enfoque tiene muchos inconvenientes: Primero, el número de escenarios crece exponencialmente con el número de etapas, haciendo el problema imposible de resolver. De hecho, el número de escenarios, $|S|$, obtenidos con dicho enfoque es:

$$|S| = |H|^{T-1}$$

Donde H es la cantidad de años para los que se tiene data histórica, y T es la cantidad de etapas consideradas. Para este caso, se generan H aperturas, en cada nodo.

Adicionalmente, tomar todas las posibles combinaciones omite la existencia de correlación temporal y espacial. Por lo tanto, se pueden generar escenarios en los cuales su ocurrencia es muy poco probable en la realidad, como el que se presenta en la Ilustración 3.9. Este escenario es generado combinando alternadamente el mínimo y máximo histórico afluente de octubre, noviembre y diciembre, en Rapel. Además, los datos observados de 2010 también son representados para contrastar con la realidad. Es importante remarcar que el período entre fines de 2010 y los primeros meses de 2011 fueron los con mayor sequía en la historia de Chile.

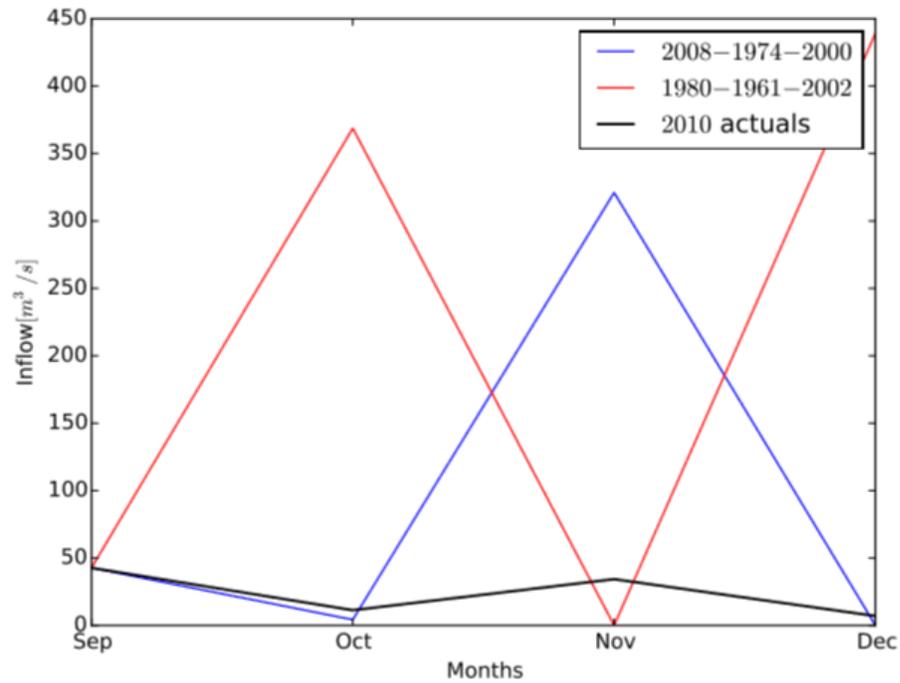


Ilustración 3.9: Ejemplo de escenarios improbables.

Para lidiar con la gran cantidad de escenarios, en [16] se propone una metodología de clustering para escoger un menor número de escenarios, para resolver el problema de planificación hidrotérmica utilizando PH. En el presente trabajo, no se desarrolla extensamente dicha metodología, ya que el problema se vuelve intratable debido a la gran cantidad de combinaciones de escenarios y además, en variadas ocasiones ha sido objeto de estudio [4] [14] [15] [16].

3.4.2. Metodología de generación de escenarios propuesta

Para generar árboles de escenarios que reflejen en algún grado las diferentes correlaciones, a continuación se presenta un método de generación de escenarios, presentado por Ríos [17] e implementado como árbol base en el presente trabajo. La idea principal detrás de este enfoque, es mirar la última observación disponible y encontrar en qué percentil de la distribución histórica se encuentra. Luego, se considera observaciones de la data histórica (del siguiente período), que pertenezca al mismo percentil para construir el escenario de la siguiente etapa. Luego se mueve hacia adelante en las etapas y se realiza el mismo proceso para cada escenario. Como resultado el número de escenarios puede ser muy grande. Por lo tanto se utilizan metodologías de clustering y de reducción de escenarios para hacer decrecer el número de escenarios y proveer un problema de optimización estocástica simple para los planificadores.

3.4.3. Tratamiento de Data

Correlación Espacial

Como se expuso en la sub sección 3.4.1, la correlación espacial es un aspecto muy importante a considerar para generar los escenarios de afluentes hídricos. Esto ha sido enfrentado en la literatura considerando las mismas hidrologías históricas para cada afluente [4] [14] [15] [16]. Sin embargo, esto no puede ser hecho si los escenarios son generados con distribuciones sintéticas⁴. Otra complicación común de utilizar distribuciones sintéticas es la cantidad de datos históricos disponibles para estimar. Esto es aún más complejo si se pretende condicionar sobre el estado actual del sistema en cada momento.

Para lidiar con este problema, se propone transformar la data histórica de cada afluente en cierta cuenca, a un afluente estándar, los que son arbitrariamente escogidos. Luego, se pueden generar distribuciones sintéticas y escenarios para estos afluentes representativos, y posteriormente, usando una transformación inversa obtener escenarios para cada afluente original.

En esta aplicación se utiliza una transformación lineal que permite con un simple cálculo ir hacia adelante y atrás en la transformación.

Correlación Temporal

Como una primera aproximación se asume que el afluente natural de un afluente a puede ser convertido a otro afluente a_R usando la siguiente expresión:

$$\omega_{a_R}^t = \sum_{\tau < t} \alpha(t, \tau) \omega_{a_R}^\tau + \sum_{\tau \leq t} \beta(t, \tau) \omega_a^\tau + \varepsilon_t \quad (3.4)$$

Donde $\omega_{a_R}^t$ representa el afluente resultante para el período t . Notar que 3.4 es una exten-

⁴Los escenarios sintéticos corresponden a generaciones a partir de la data histórica, pero que no representan ninguna realización hídrica pasada.

sión de un Modelo Periódico Autoregresivo (PARMA), de esta forma se induce la correlación temporal, además, se incluyen los efectos de los afluentes vecinos en la estimación del afluente hídrico ω_{aR}^t . Para estimar los coeficientes α y β se formula el problema de optimización:

$$\min_{\alpha, \beta} \sum_{t \in H} \|\varepsilon_t\|_p \quad (3.5)$$

Donde p corresponde a una norma dada, y H es el conjunto de observaciones históricas.

Asumiendo que cada periodo t está compuesto por un mes y un año, $t = (m, y)$, el problema se puede replantear en forma matricial:

$$W_{aR}^y = A\omega_{aR}^y + B\omega_a^y + \varepsilon \quad (3.6)$$

Donde $\omega_a^y \in \mathcal{M}_{12 \times 1}$ es un vector con las hidrologías mensuales⁵ para el año y y afluente a , $A, B \in \mathcal{M}_{12 \times 12}$, A, B son matrices triangulares inferiores. Para no tener singularidades en la matriz B , se agrega la siguiente restricción:

$$\beta(t, \tau) = 0 \quad \forall \tau \notin \{t-1, t\} \quad (3.7)$$

Finalmente, teniendo la matriz inversa B^{-1} se puede convertir los escenarios del afluente representativo al afluente orginial, usando la siguiente expresión:

$$\omega_a^y = B^{-1}(I - A)\omega_{aR}^y + \varepsilon \quad (3.8)$$

Donde I es matriz identidad de $\mathcal{M}_{12 \times 12}$.

3.4.4. Generador de escenarios Synth - NSG

Los generadores de escenarios que se exponen a continuación, corresponden a un desarrollo hecho íntegramente en esta investigación. Si bien estos generadores de escenarios, pueden ser resueltos con distintas metodologías (ej. SDDP), este estudio se centra en la resolución con PH.

El enfoque para generar escenarios puede ser resumido en los siguientes pasos:

1. Estimar la distribución.
2. Encontrar observaciones vecinas.
3. Estimar distribución condicional.

⁵El procedimiento es exactamente el mismo si se considera la data semanal.

4. Calcular puntos de escenario.
5. Avanzar al periodo siguiente y repetir desde 2.

A continuación se explica en detalle, paso a paso:

1. **Estimar distribución:** Después de convertir cada afluente individual en un afluente representativo para cada cuenca, el siguiente paso es estimar la distribución de los afluentes naturales en cada periodo (semana o mes) usando toda la data histórica para el afluente representativo (H^p). Esto puede ser logrado utilizando cualquier técnica de estimación de distribuciones. En este caso, se implementan dos métodos diferentes: Kernel [18] y la estimación exponencial epi-splines [19] [20]. Es importante remarcar que este paso es ejecutado solo una vez, y como resultado se obtiene una distribución de afluentes hídricos para cada período.

$$f_{\omega_{a^*}}^p(\cdot)$$

Donde $p \in \{1, \dots, 12\}$ si se utilizan meses, y $p \in \{1, \dots, 48\}$ si se estima semanalmente

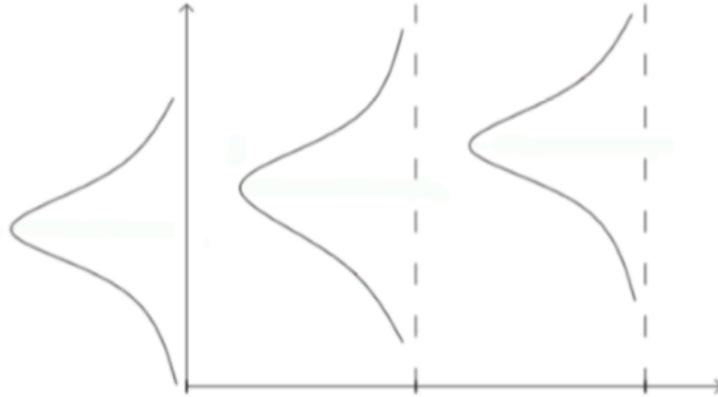


Ilustración 3.10: Estimación de distribución.

Encontrar estimaciones vecinas: Suponiendo que se está en el período p_0 y se quiere generar escenarios para los siguientes P períodos. Además, se asume que el afluente natural en t_0 es conocido, W^{p_0} . Como se busca generar escenarios respecto a la correlación temporal, para obtener escenarios en la siguiente etapa se consideran todos los caminos en la data histórica, que pertenecen al intervalo de una determinada parte de la distribución, como se muestra en la ilustración 3.11 (a).

Estimar distribución condicional: Luego, considerando solamente estos caminos, se estima una distribución de hidrologías para el período t_0+1 , como se muestra en la ilustración 3.11 (b). Esto se logra mediante el uso de la estimación Kernel o epi-spline.

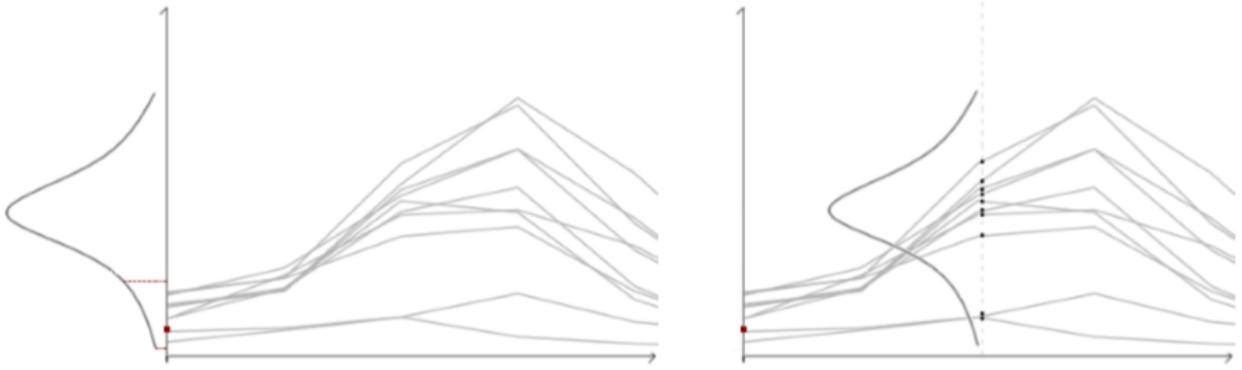


Ilustración 3.11: Selección de vecinos y estimación de distribución.

Calcular puntos de escenarios: Una vez calculada la distribución “condicional” en el período $t_0 + 1$, se puede generar los escenarios, definiendo segmentos de la distribución y calculando el valor condicional esperado, de cada segmento.

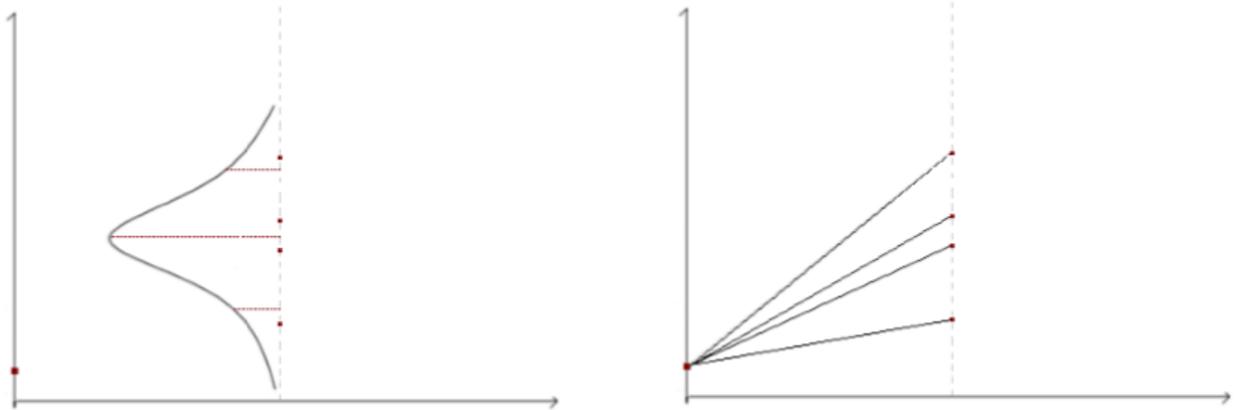


Ilustración 3.12: Distribuciones condicional, y elección de puntos dentro de cada segmento.

Avanzar al período siguiente: Después de calcular cada punto de escenario en la etapa $t_0 + 1$, el siguiente paso es seleccionar cada uno de dichos puntos y repetir el proceso, comenzando desde el paso 2, en otras palabras, observar la distribución estimada para el período $t_0 + 1$ calculada en el paso 1, luego, encontrar los caminos en la data histórica que se encuentran en el intervalo de dicho punto, utilizar esos caminos para estimar la distribución condicional del período $t_0 + 2$, y generar nuevos puntos. Este proceso debe ser repetido para cada punto generado en la etapa. La generación de escenarios termina cuando los escenarios de la última etapa son generados.

La metodología anterior será nombrada más adelante como **Synth** (del inglés *synthetic*) ya que corresponde a una generación sintética de escenarios.

Una segunda metodología desarrollada en esta tesis es **NSG** (una sigla de *Neighbor Scenario Generation* en inglés), que es muy similar a Synth, con diferencia que en el paso 4 (Calcular puntos de escenarios), en vez de calcular el valor condicional esperado de cada segmento, se selecciona la realización histórica más cercana al valor condicional esperado, dentro del intervalo.

Esto se hace para desarrollar las exigencias de la ley general de servicios eléctricos citada con anterioridad en la sección actual, que hace obligatorio el uso de las hidrologías históricas en la generación de escenarios.

Reducción de número de escenarios

Como se expuso en la sección anterior, una de las dificultades al momento de generar escenarios con aperturas por nodos, es que la cantidad de escenarios va creciendo de manera exponencial con el número de períodos. Lo anterior hace que para problemas de numerosas etapas, sea imposible de resolver debido al tamaño. La siguiente tabla muestra la cantidad de escenarios, según las aperturas y el período de planificación:

Tabla 3.1: Cantidad de escenarios según aperturas y períodos

Aperturas por período	Meses	Escenarios
2	12	2.048
3	12	177.147
2	24	8.388.608
3	24	9,41E+10

Se observa que para una baja cantidad de períodos y aperturas, se tiene un gran número de escenarios.

Una de las aplicaciones de los generadores de escenarios NSG y SYNTH, es que la cantidad de aperturas por nodo, puede ir variando etapa a etapa. De esta manera, con el objetivo de disminuir la cantidad de escenarios, se toman más aperturas para los períodos más próximos, donde las decisiones de operación son más importantes, y se hacen menos aperturas para períodos más alejados. La ilustración 2.1, es de ayuda para entender este razonamiento.

Capítulo 4

Modelo de optimización

Para determinar la operación óptima del sistema eléctrico, se utiliza un modelo de programación lineal, el cual busca minimizar los costos de operación a lo largo de horizonte de planificación, penalizando fuertemente los tramos de demanda insatisfecha.

La formulación que se presenta en este capítulo se basa en un trabajo desarrollado por Rigoberto Torres [4] y Andrés Iroume [16]. Pero se exponen modificaciones que muestran cómo se modela actualmente el problema en el Centro de Energía de la Universidad de Chile.

4.1. Conjuntos

Conjuntos básicos:

- T : Etapas de planificación
- B : Bloques de carga
- N : Nodos de la red
- G : Generadores eléctricos
- L : Cargas o centros de consumo
- TL : Líneas de transmisión $(i, j) \in N \times N, i \neq j$

Tipo de generadores:

- G_T : Generadores térmicos
- G_H : Generadores hidráulicos
- G_R : Generadores hidráulicos de embalse
- G_S : Generadores hidráulicos de pasada, conectados en serie con otro generador
- G_P : Generadores hidráulicos de pasada independiente
- G_D : Generadores que no son hídricos ni térmicos (ej: eólicos, solares, biomasa.)

Cabe notar que:

$$\begin{aligned} G_H &= G_R \cup G_S \cup G_P \\ G &= G_H \cup G_T \cup G_D \end{aligned}$$

Conjuntos sobre la red hidráulica:

$$\begin{aligned} \Omega_i^{HR} &: \text{Centrales de embalse aguas arriba de central hídrica } i \quad \forall i \in G_R \\ \Omega_i^{HS} &: \text{Centrales hídricas aguas arriba de central hídrica } i \quad \forall i \in G_S \end{aligned}$$

Conjuntos sobre la red:

$$\begin{aligned} \Omega_i^G &: \text{Conjunto de generadores conectados al nodo } i \quad \forall i \in N \\ \Omega_i^L &: \text{Conjunto de cargas conectadas al nodo } i \quad \forall i \in N \end{aligned}$$

4.2. Parámetros

Parámetros costos y demanda:

$$\begin{aligned} \alpha_i &: \text{Coeficiente lineal de función de costos generador térmico } i \quad \forall i \in G_T \\ \beta_i &: \text{Coeficiente cuadrático función de costos generador térmico } i \quad \forall i \in G_T \\ a_i &: \text{Coeficiente lineal función de costos por carga insatisfecha } i \quad \forall i \in L \\ b_i &: \text{Coeficiente cuadrático función de costos por carga insatisfecha } i \quad \forall i \in L \\ P_i^{t,b} &: \text{Demanda por energía en el nodo } i, \text{ etapa } t \text{ y bloque } b \quad \forall i \in N, \forall b \in B, \forall t \in T \\ \sigma^t &: \text{Tasa de descuento para el período } t \quad \forall t \in T \end{aligned}$$

Parámetros líneas de transmisión:

$$\begin{aligned} r_{i,j} &: \text{Resistencia del equipo de transmisión que conecta } (i, j) \quad \forall (i, j) \in TL \\ x_{i,j} &: \text{Reactancia del equipo de transmisión que conecta } (i, j) \quad \forall (i, j) \in TL \\ \bar{f}_{(i,j)} &: \text{Flujo máximo para línea de transmisión } (i, j) \quad \forall (i, j) \in TL \\ \underline{f}_{(i,j)} &: \text{Flujo mínimo para línea de transmisión } (i, j) \quad \forall (i, j) \in TL \\ \gamma &: \text{Penalizador por transmisión de energía por líneas} \end{aligned}$$

Parámetros para generadores:

$$\begin{aligned} \overline{p}_i^{t,b} &: \text{Potencia máxima del generador } i \text{ en período } t \text{ y bloque } b \quad \forall i \in G, \forall b \in B, \forall t \in T \\ \underline{p}_i^{t,b} &: \text{Potencia mínima del generador } i \text{ en período } t \text{ y bloque } b \quad \forall i \in G, \forall b \in B, \forall t \in T \end{aligned}$$

Parámetros para generadores hídricos:

$$\begin{aligned} \rho_i &: \text{Coeficiente de conversión de caudal turbinado a potencia } i \quad \forall i \in G_H \\ \overline{q}_i^T &: \text{Capacidad máxima de caudal a turbinar por central } i \quad \forall i \in G_H \end{aligned}$$

Parámetros para centrales de embalse:

- ρ'_i : Rendimiento del caudal turbinado según la altura del embalse $i \quad \forall i \in G_R$
- κ'_i : Coeficiente entre diferencia de volumen y filtrado de embalse $i \quad \forall i \in G_R$
- κ''_i : Coeficiente lineal de filtrado de embalse $i \quad \forall i \in G_R$
- q_i^S : Caudal máximo a derramar por embalse $i \quad \forall i \in G_R$
- \bar{V}_i : Volúmen máximo de embalse $i \quad \forall i \in G_R$
- \underline{V}_i : Volúmen mínimo de embalse $i \quad \forall i \in G_R$
- $\delta^{t,b}$: Tiempo en segundos de duración del bloque b y período $t \quad \forall t \in T, \forall b \in B$.
(convierte el volumen en caudal turbinable para embalses)
- V_i^0 : Volumen inicial de embalse $i \quad \forall i \in G_R$

Parámetros estocásticos:

- $w_i^{t,b}$: Afluente natural de la central i en período t y bloque $b \quad \forall i \in G_H, t \in T, b \in B$

4.3. Variables

Variables para red eléctrica:

- $P_{g,t,b}$: Potencia generada por el generador g en el período t y bloque b ,
 $\forall g \in G, \forall b \in B, \forall t \in T$
- $PU_{l,t,b}$: Potencia no suministrada en el punto de demanda l , para el período t y bloque b ,
 $\forall l \in L, \forall b \in B, \forall t \in T$
- $\theta_{n,t,b}$: Ángulo del nodo n en el período t y bloque b ,
 $\forall n \in N, \forall b \in B, \forall t \in T$
- $FP_{i,j,t,b}$: Flujo eléctrico que va desde el nodo i hasta el nodo j en el período t y bloque b ,
 $\forall (i, j) \in TL, \forall b \in B, \forall t \in T$
- $FN_{i,j,t,b}$: Flujo eléctrico que va desde el nodo j hasta el nodo i en el período t y bloque b ,
 $\forall (i, j) \in TL, \forall b \in B, \forall t \in T$

Variables para red hídrica:

- $QT_{h,t,b}$: Caudal turbinado por el generador hidráulico h en el período t y bloque b ,
 $\forall h \in G_H, \forall b \in B, \forall t \in T$
- $QS_{h,t,b}$: Caudal derramado por el generador hidráulico h en el período t y bloque b ,
 $\forall h \in G_H, \forall b \in B, \forall t \in T$
- $QF_{h,t,b}$: Caudal filtrado por el embalse h en el período t y bloque b ,
 $\forall h \in G_R, \forall b \in B, \forall t \in T$
- $QU_{h,t,b}$: Caudal usado embalsado en h en el período t y bloque b ,
 $\forall h \in G_R, \forall b \in B, \forall t \in T$
- $VR_{h,t}$: Volumen del embalse h en el período t
 $\forall h \in G_R, \forall t \in T$
- $VU_{h,t,b}$: Volumen usado embalsado de h en el período t ,
 $\forall h \in G_R, \forall t \in T$

Variable auxiliar:

$$SC_t \quad : \quad \text{Costo total de la etapa } t, \\ \forall t \in T$$

4.4. Restricciones

Restricciones de red eléctrica:

1. Restricciones de flujo para cada línea (relación entre ángulos, reactancia y flujo eléctrico); para que el flujo resultante sea factible desde el punto de vista físico, se debe respetar la ley de voltajes de Kirchhoff (LVK), descrita en la siguiente restricción:

$$FP_{(i,j),t,b} - FN_{(i,j),t,b} = \frac{\theta_{i,t,b} - \theta_{j,t,b}}{x_{i,j}} \quad \forall (i,j) \in TL, \forall t \in T, \forall b \in B \quad (4.1)$$

2. Fijar el ángulo del primer nodo (nodo de referencia):

$$\theta_{1,t,b} = 0 \quad \forall t \in T, \forall b \in B \quad (4.2)$$

3. Flujo transmitido debe respetar cotas mínimas y máximas:

$$\overline{f_{(i,j)}} \leq FP_{(i,j),t,b} - FN_{(i,j),t,b} \leq \underline{f_{(i,j)}} \quad \forall (i,j) \in TL, \forall t \in T, \forall b \in B \quad (4.3)$$

4. Demanda insatisfecha debe ser positiva y menor o igual a la demanda real:

$$0 \leq PU_{l,t,b} \leq P_l^{t,b} \quad \forall l \in L, \forall t \in T, \forall b \in B \quad (4.4)$$

5. Se debe satisfacer la demanda para cada nodo i , teniendo en cuenta las pérdidas en líneas de transmisión:

$$\sum_{g \in \Omega_1^G} P_{g,t,b} + \sum_{l \in \Omega_1^L} PU_{l,t,b} + \sum_{j \in N} \left[FP_{(j,i),t,b} - FN_{(j,i),t,b} - \frac{r_{j,i}(FP_{(j,i),t,b} - FN_{(j,i),t,b})^2}{2} \right] \\ - \sum_{j \in N} \left[FP_{(i,j),t,b} - FN_{(i,j),t,b} - \frac{r_{i,j}(FP_{(i,j),t,b} - FN_{(i,j),t,b})^2}{2} \right] \geq P_i^{t,b} \quad \forall i \in N, \forall t \in T, \forall b \in B \quad (4.5)$$

Restricciones de generadores eléctricos:

6. La generación eléctrica de cada planta debe respetar su capacidad máxima y mínima (existen máximos y mínimos variables según bloque-período):

$$0 \leq \underline{p_g^{t,b}} \leq P_{g,t,b} \leq \overline{p_g^{t,b}} \quad \forall g \in G, \forall t \in T, \forall b \in B \quad (4.6)$$

Restricciones de generadores hidro-eléctricos:

7. Conversión de caudal turbinado a potencia eléctrica:

$$P_{h,t,b} = \rho_h(QT_{h,t,b}) \quad \forall h \in G_H \setminus G_R, \forall t \in T, \forall b \in B \quad (4.7)$$

8. Caudal vertido debe ser positivo:

$$0 \leq QS_{h,t,b} \quad \forall h \in G_H, \forall t \in T, \forall b \in B \quad (4.8)$$

9. Caudal turbinado debe respetar sus cotas máximas y mínimas:

$$0 \leq QT_{h,t,b} \leq \overline{q_h^T} \quad \forall h \in G_H, \forall t \in T, \forall b \in B \quad (4.9)$$

10. Conservación de flujo de centrales hídricas en serie:

$$QT_{h,t,b} + QS_{h,t,b} = \sum_{i \in \Omega_h^{HR}} QF_{i,t,b} + \sum_{j \in \Omega_h^{HS}} [QT_{j,t,b} + QS_{j,t,b}] + \omega_h^{t,b} \quad \forall h \in G_H \setminus G_R, \forall t \in T, \forall b \in B \quad (4.10)$$

Restricciones de generadores tipo embalse:

11. Conversión de caudal turbinado por embalses a potencia eléctrica, para las centrales tipo embalse, se realiza una aproximación que relaciona el volumen del embalse con su producción:

$$P_{h,t,b} = QT_{h,t,b}[\rho_h + \rho'_h(V_{h,t,b} - V_{h,t,b-1})] \quad \forall h \in G_R, \forall t \in T, \forall b \in B \quad (4.11)$$

12. Conversión de caudal filtrado por embalses, al igual que la restricción anterior se hace una aproximación que relaciona el volumen del embalse con las filtraciones:

$$QF_{h,t,b} = \kappa_h'' + \kappa_h'(V_{h,t,b-1} - V_{h,t,b}) \quad \forall h \in G_R, \forall t \in T, \forall b \in B \quad (4.12)$$

13. Cotas de volumen para embalses:

$$\underline{V}_h \leq VR_{h,t} \leq \overline{V}_h \quad \forall h \in G_H, \forall t \in T \quad (4.13)$$

14. Definición de volumen utilizado VU para central tipo embalse:

$$VR_{h,t-1} - VR_{h,t} = VU_{h,t} \quad \forall h \in G_R, \forall t \in T \setminus \{1\} \quad (4.14)$$

15. Volumen utilizado para el período inicial

$$V_h^0 - VR_{h,t} = VU_{h,t} \quad \forall h \in G_R, t = 1 \quad (4.15)$$

16. Relación entre volumen utilizado desde el embalse y el caudal turbinado en cada bloque:

$$VU_{h,t} = \sum_{b \in B} [QU_{h,t,b}(\delta^{t,b})] \quad \forall h \in G_R, \forall t \in T, \forall b \in B \quad (4.16)$$

17. Conservación de flujo de embalses:

$$QT_{h,t,b} + QS_{h,t,b} + QF_{h,t,b} + QU_{h,t,b} = \sum_{i \in \Omega_h^{HR}} QF_{i,t,b} + \sum_{j \in \Omega_h^{HS}} [QT_{j,t,b} + QS_{j,t,b}] + \omega_h^{t,b} \quad (4.17)$$

$$\forall h \in G_R, \forall t \in T, \forall b \in B$$

Costos:

18. Definición de costos por etapa (variable auxiliar):

$$SC_t = \sum_{b \in B} \delta_{t,b} \left\{ \sum_{i \in G_T \cup G_D} [\alpha_i(P_{i,t,b}) + \beta_i(P_{i,t,b})^2] + \sum_{l \in L} [a_l(PU_{l,t,b}) + b_l(PU_{l,t,b})^2] \right\} \quad (4.18)$$

$$\forall t \in T$$

4.5. Función objetivo

Se busca minimizar los costos totales de operación:

$$\text{mín} \sum_{t \in T} \frac{SC_t}{\sigma^t} \quad (4.19)$$

4.6. Rectificaciones del modelo

El modelo utilizado para hacer las pruebas presenta modificaciones con respecto al modelo original, que buscan simplificar su resolución y aumentar la exactitud de los resultados:

Filtrado de embalses

Con el fin de hacer los resultados comparables con otras formas de resolución, para la restricción 4.12 las filtraciones de embalses serán consideradas constantes ($\kappa'_h = 0$) e igual a la media histórica de filtraciones:

$$QF_{h,t,b} = \kappa''_h \quad \forall h \in G_R, \forall t \in T, \forall b \in B \quad (4.20)$$

Caudal turbinado por embalses

En la restricción 4.11 que expone la conversión del caudal turbinado a potencia eléctrica, se considera que un rendimiento constante es suficiente para modelar la conversión de energía, asignando a la constante ρ'_h un valor nulo. En caso contrario, la restricción adopta una forma no lineal (producto de la multiplicación de variables), que involucra el estado inicial y final de los embalses, complicando la resolución del problema de optimización ya que esto puede causar la pérdida de convexidad de la región factible. Por ello, y teniendo en cuenta que el objetivo de este modelamiento, es comparar metodologías de resolución, la restricción que se implementa la misma que las centrales en serie (restricción 4.7) y tiene la siguiente forma:

$$P_{h,t,b} = QT_{h,t,b}(\rho_h) \quad \forall h \in G_R, \forall t \in T, \forall b \in B \quad (4.21)$$

Pérdidas en líneas de transmisión

Al igual que el caso anterior, para evitar la pérdida de convexidad en la restricción 4.5, el parámetro $r_{i,j}$ toma un valor nulo, por lo que no se representan las pérdidas en las líneas de transmisión:

$$\sum_{g \in \Omega_i^G} P_{g,t,b} + \sum_{l \in \Omega_i^L} PU_{l,t,b} + \sum_{j \in N} \left[FP_{(j,i),t,b} - FN_{(j,i),t,b} \right] - \sum_{j \in N} \left[FP_{(i,j),t,b} - FN_{(i,j),t,b} \right] \geq P_i^{t,b} \quad \forall i \in N, \forall t \in T, \forall b \in B \quad (4.22)$$

La aproximación de pérdidas de líneas de transmisión a cero, puede alejar el problema de la realidad, pero teniendo en cuenta que el objetivo es comparar metodologías de resolución para problemas estocásticos, no presenta un alto impacto.

Orden de magnitud de variables

Un error que se presenta en los solvers es la imprecisión de la solución debido a la diferencia de orden de las variables, es por esto que a las variables que representan el volumen de los embalses ($VR_{h,t}$) que son de orden 10^{10} , han sido ponderadas por un factor de escala igual a $1,7e - 7$ que es el que actualmente se utiliza en el modelo de PLP.

Linealización de costos

Por último, se consideran los parámetros de costos cuadráticos β_i y b_l como nulos, en la restricción 4.18, de esta forma, se evita la pérdida de linealidad del problema.

Además, para evitar la aparición de múltiples soluciones óptimas, producto de la creación de subciclos de transmisión de energía, se agrega un coste ficticio a la transmisión γ . Este penalizador es muy pequeño en comparación con los demás costos.

Luego, la variable auxiliar queda definida como sigue:

$$SC_t = \sum_{b \in B} \delta_{t,b} \left\{ \sum_{i \in G_T \cup G_D} \alpha_i(P_{i,t,b}) + \sum_{l \in L} a_l(PU_{l,t,b}) \right\} + \sum_{b \in B, (i,j) \in TL} \gamma(FP_{(i,j),t,b} + FN_{(i,j),t,b}) \quad \forall t \in T \quad (4.23)$$

No se utiliza un flujo resultante como intuitivamente se esperaría ($F_{(i,j),t,b}^{Resultante} = FP_{(i,j),t,b} - FN_{(i,j),t,b}$), y se decide usar variables que representan el flujo positivo $FP_{(i,j),t,b}$ y flujo negativos $FN_{(i,j),t,b}$, de esta manera, el modelo es exactamente igual al hecho para PLP.

Capítulo 5

Algoritmo Progressive Hedging

En el presente capítulo se profundizará en el algoritmo Progressive Hedging para el problema de Coordinación Hidrotérmica, así como definir y caracterizar los ajustes y modificaciones implementados para mejorar la resolución del problema eléctrico. La información contenida en este capítulo busca reforzar lo desarrollado en la sección 2.4.

Lo que se busca con los ajustes y modificaciones expuestas más adelante, es aumentar la velocidad de convergencia del algoritmo, adecuando ciertos parámetros, así como la linealización de términos, permite una mayor velocidad de resolución. El trade-off que se presente es que a una solución rápida puede no ser siempre óptima ya que al acelerar la convergencia se puede forzar que el algoritmo lleve la solución a un punto sub-óptimo.

En este caso, y debido a que las restricciones no-convexas han sido omitidas, se enfrenta a un problema convexo y se puede esperar converger al óptimo (con los parámetros y tolerancia adecuados). El objetivo es encontrar soluciones adecuadas y de manera rápida.

5.1. Pseudo-código extendido

A continuación, se presenta una versión detallada del algoritmo de PH para el caso general, incluyendo diferentes parámetros y tolerancia necesaria. El objetivo es formalizar el marco teórico expuesto en la sección 2.4.

Algoritmo Progressive Hedging - Extendido

- 1: Inicializar: ε , k^{max} (tolerancia) ;
- 2: $k := 0$, $g^* := \infty$;
- 3: $\forall s \in S$ $x_s^k := \operatorname{argmin}_{x_s} f_s(x_s) : x_s \in Q_s$;
- 4: $k := k + 1$;
- 5: $\forall t \in T, \forall N_t \in N$ $\bar{x}_{n,t}^k := \frac{1}{N_t} \sum_{s \in N_t} x_{t,s}^k$;
- 6: $g^k := \sum_{s \in S, t \in T} \|x_{t,s}^k - \bar{x}_{n,t}^k\|$;
- 7: **IF** $g^k < g^* \vee \sum_{s \in S} f_s(x_s^k) > \sum_{s \in S} f_s(x_s^*)$, guardar mejor solución, $x^* := x^k$;

- 8: **IF** $g^k < \varepsilon \vee k > k^{max}$, Ir a 13 ;
 - 9: **IF** $k \leq 1$, $\forall x_s^i \quad \rho_s^i = \rho^i(x, k, s)$;
 - 10: $\forall s \in S, t \in T \quad \omega_{s,t}^k := \rho(x_{t,s}^k - \bar{x}_{n(n,t),t}^k) + \omega_{s,t}^{k-1}$, $(\omega^{(0)} = \vec{0})$;
 - 11: $\forall s \in S \quad x_s^k := \operatorname{argmin}_{x_s} f_s(x_s) + \sum_{s \in S, t \in T} [\omega_{s,t}^k x_{s,t} + \frac{\rho}{2} \|x_{t,s}^k - \bar{x}_{n(n,t),t}^k\|^2]$: $x_s \in Q_s$;
 - 12: Ir a 4 ;
 - 13: Usando x^* como hotstart con las mismas variables fijadas, resolver la formulación extendida compacta: $\min_x \sum_{s \in S} f_s(x_s) : (x \in Q)$;
-

Se debe notar lo siguiente:

- Las tolerancias fijadas en **(1)**, corresponden a la distancia al óptimo y máximo de iteraciones, respectivamente. Ambos valores son pre-definidos antes de iniciar el algoritmo.
- En el paso **(4)** simplemente va contando el número de iteraciones.
- En **(5)** los elementos de la sumatoria tienen igual probabilidad, por lo que los pesos son idénticos, para el cálculo del promedio.
- En **(6)** se lleva a cabo el cálculo de la convergencia de cada variable del algoritmo, la cual corresponde a la distancia euclidiana entre cada escenario, con respecto a su óptimo estimado, en el nodo correspondiente. Junto a lo anterior, se introduce una función $n : S \times t \rightarrow N$, función que se encarga de ubicar cada uno de los escenarios con su nodo correspondiente dentro de cada una de las etapas. De esta forma, se tiene una representación simple a la estructura de árbol original, con nodos padres e hijos. Del mismo modo, se utiliza por primera vez el subíndice n para los promedios que indica un paquete de información determinado.
- En **(7)** se presenta una nueva familia de variables superindexadas por * indicando que son la mejor solución obtenida hasta el momento por el algoritmo. Esto se debe a la naturaleza heurística que no siempre converge o mejora el valor de la función objetivo, con variables enteras, sino que va "saltando" de solución en solución.
- En el paso **(8)** posiblemente alcanzar el valor óptimo (gap cero) tome demasiado tiempo, por lo que se establece un gap determinado (0,05 %) y un límite de tiempo en caso de que la solución no progrese.
- En **(9)** se caracteriza el penalizador ρ como un elemento que es función dependiente de variables que componen el problema, la iteración en que se encuentra y el escenario que se está trabajando. sin embargo, esto varía según la metodología de cálculo del factor ρ utilizadas, aspecto que se trata con profundidad en la sección 5.2.4
- En el paso **(10)** se actualiza el penalizador ω , mientras que en **(11)** se resuelve el problema penalizado, con el término cuadrático proximal incluido.
- Por último, en el paso **(13)** se utiliza la técnica de fijar variables que ya convergieron (o están cerca de converger), para posteriormente resolver la Formulación extendida del

problema.

5.2. Ajustes y mejoras

Los temas desarrollados en esta sección son:

- Aceleración de la resolución de cada subproblema.
- Paralelizar el algoritmo.
- Linealización del término cuadrático proximal.
- Bundling.
- Criterio de término.

5.2.1. Aceleración de la resolución de cada subproblema

Existen tres principales técnicas que permiten acelerar la resolución de cada uno de los subproblemas que se generan a lo largo de las iteraciones del algoritmo:

- **Fortalecimiento del modelo determinístico (o subproblema):** corresponde a ciertas modificaciones y/o adiciones que se realizan sobre la formulación original que presentan como principal característica, el hecho de no dejar fuera espacios factibles de solución, pero permiten encontrar soluciones gap-óptimas más rápido.
- **Hotstarts (o Warmstarts):** Esta técnica consiste en usar la solución de la iteración anterior como punto de partida para cada subproblema. De tal forma de tener un punto de partida cercano al óptimo, lo que tiende a producir tiempos de compunto menores.
- **Gap variable:** Ajustando el gap de optimalidad del solver en cada iteración como criterio de término. Como en cada iteración se busca la convergencia de la solución, es posible que en las primeras iteraciones no sea tan provechoso utilizar gaps muy bajos (o nulos) ya que simplemente se busca una solución aproximada para comenzar a calcular los promedios \bar{x} (difícilmente se encontrará la convergencia de variables en la primera iteración) y pasar a la siguiente iteración. De esta manera, a medida que se va avanzando en las iteraciones se disminuye el Gap, para encontrar soluciones más precisas. De esta forma, los ajustes con respecto a la tolerancia al óptimo son muy útiles para alcanzar soluciones de gran calidad en menor tiempo.

La metodología utilizada para llevar a cabo la comparación entre las distintas combinaciones que se pueden generar con estas técnicas, consistió en utilizar el algoritmo PH para el total de iteraciones necesarias para llegar al óptimo de una instancia pequeña, a un gap del 0.05%. Las combinaciones usadas fueron:

- Sin warmstart, sin Gap ajustable.
- Solo warmstart.
- Solo con Gap ajustable.
- Con warmstart, con Gap ajustable.

Como es de esperar, la mejor combinación obtenida corresponde a la cuarta mencionada, que es la utilización de warmstarts y gap ajustable. En anexos D se pueden ver los resultados para la instancia de menor tamaño Cuenca del Laja.

En relación al uso de warmstart, esta es una estrategia ampliamente conocida y popular dentro de las disciplinas de investigación y gestión de operaciones, teniéndose una amplia documentación al respecto [21]. Para el problema en estudio dicha lógica se confirma, puesto que se obtienen mejoras del orden del 20 % (con respecto al planteamiento sin warmstart ni Gap ajustable) en el tiempo de resolución, de acuerdo a los resultados experimentales en la instancia de menor tamaño.

Con respecto al ajuste variable del gap de optimalidad de cada uno de los subproblemas según iteración se comprobó su utilidad a la hora de alcanzar soluciones de gran calidad en poco tiempo en relación a la búsqueda de un óptimo global de cada instancia, proceso que puede tardar mucho tiempo sin ser significativamente mejor (en términos de valor) a una solución que puede ser encontrada en segundos. La estrategia utilizada consiste en usar un gap elevado en las primeras iteraciones, de tal forma que cada uno de los subproblemas tome poco tiempo en ser resuelto, teniendo cuidado en no tomar un gap muy alto, pues se buscan soluciones que luego al converger en conjunto sean cercanas al óptimo global de la formulación extendida, moviéndose además en la dirección que la penalización les indique, del mismo modo, a través de una serie de experimentos de repetición, se encontró un gap inicial cercano al 15 % que no daña considerablemente la solución, y presenta mejoras desde el punto de vista de tiempo de ejecución.

Luego, iteración a iteración se va disminuyendo el tamaño del gap hasta que el problema converge. Mediante una serie de ensayos se determina trabajar con un gap objetivo (gap en la última iteración) del 0.5 % y 1 %, debido a que cumplen con las condiciones necesarias para llevar a cabo mejoras en los tiempos de resolución.

5.2.2. Paralelización del algoritmo

Debido a PH es un algoritmo de descomposición horizontal¹, presenta facilidades para hacer una implementación computacional paralela y distributiva.

La implementación paralela consiste en enviar cada subproblema (escenario) a un núcleo o clúster determinado, una vez que se tiene la solución para cada escenario, realizar el cálculo

¹Un algoritmo de descomposición horizontal, descompone el problema escenario por escenario, mientras que uno de descomposición vertical, divide el problema etapa por etapa

de promedios y penalizadores, y posteriormente continuar la siguiente iteración. Una forma distributiva quiere decir que si se tienen menos núcleos que sub problemas, se van asignando los subproblemas a los cluster a medida que estos terminan de resolver subproblemas.

La implementación computacional para el algoritmo se encuentra en el software PySP, parte del paquete de software Pyomo² de los laboratorios Nacionales Sandia [22]. Dicho software está orientado a la programación de problemas de optimización, en particular para la implementación del algoritmo Progressive Hedging. Además presenta una estructura estándar para el planteamiento de árboles de escenarios [23].

Después de realizar numerosas pruebas, los resultados obtenidos producto de la paralelización del algoritmo, muestran una notable disminución en el tiempo de ejecución (entre un 45 % y 70 % dependiendo del tamaño de las instancias), en comparación al caso serial y de la formulación extendida. La disminución de tiempo es proporcionalmente mayor, si aumenta el número de escenarios, lo que es consistente con la teoría, puesto que cada procesador o nodo resuelve problemas determinísticos muy simples, y no incurre en grandes tiempos de carga o ejecución, mientras que la formulación extendida debe trabajar todo el problema desde un comienzo.

Cabe notar que en trabajos como [24], se han obtenido mejoras de hasta un 90 % en los tiempos de ejecución.

5.2.3. Linealizar término cuadrático proximal

Una de las complicaciones que presenta PH es que para lograr la convergencia de las variables en cada nodo, se debe restringir agregando un término cuadrático proximal (que obliga que las variables se acerquen a su promedio, paso 11 de la sección 5.1). Lo anterior se traduce en un crecimiento en la complejidad de los subproblemas, lo que evidentemente produce un aumento en el tiempo de cómputo:

$$\sum_{s \in S, t \in T} [\omega_{s,t}^k x_{s,t} + \frac{\rho}{2} \|x_{t,s}^k - \bar{x}_{n(n,t),t}^k\|^2] \quad (5.1)$$

Como se pudo ver en la subsección 2.4.1, el término cuadrático es de suma importancia a la hora de evitar que los subproblemas en resolución se vuelvan no acotados (en el Ejemplo 2.4.1.1 se expone el caso anterior), el término ωx puede ser positivo o negativo, lo que puede causar que la función objetivo crezca indefinidamente, en caso de ser una variable no acotada.

Sin embargo es claro que este término es ficticio, ya que no representa un significado real en el problema de optimización. Luego en el contexto heurístico, si es posible reemplazar este término por una expresión similar pero que no agregue tanta complejidad a los subproblemas, será de gran utilidad.

Luego, a partir de lo anterior, nace la necesidad modificar la el planteamiento original. Ob-

²El software Pyomo antes tenía el nombre Coopr.

servando la expresión 5.1, se puede ver que no existen multiplicaciones cruzadas de variables (ej: xy), solo se ven términos cuadráticos (ej: xx), por lo que la utilización de aproximación lineal por tramos (*piece-wise linearization* en inglés) es una buena alternativa para disminuir la complejidad de los sub problemas.

La implementación de esta linealización fue considerando la importancia de actualizar las pendientes y puntos de quiebre en cada iteración, debido a que el punto mínimo de la curva representa el promedio \bar{x} , que puede ir variando en todas las iteraciones. Del mismo modo, es importante el hecho de que todas las variables que necesiten la linealización del término cuadrático, deben presentar cotas apropiadas, debido a que las pendientes calculadas acotan el problema solo finitamente y no como la curva original $\rho(x - \bar{x})^2$ que siempre puede ser mayor que ωx cuando $x \rightarrow \infty$, entonces puede darse que en un problema de cierta variable no acotada, termine transformando el problema en uno no acotado.

El comportamiento en la práctica de esta aproximación depende fuertemente del número de segmentos escogidos. Si se considera que cada segmento representa una variable nueva con una restricción asociada, es lógico concluir que el número de segmentos es proporcional al número de vértices de poliedro original que representa el subproblema, en los casos que la penalización define una cara. Luego, si se tiene muy pocos segmentos, en el peor de los casos 2, el penalizador tiende a adoptar la forma de V , lo que genera que todas las variables se vean forzadas a caer al promedio, provocando una convergencia anticipada, perjudicando de esta manera a la función objetivo. Asimismo, puede causar una convergencia oscilante, puesto que los promedios cambian durante cada iteración, lo que lleva a generar saltos entre iteración y otra. Por otro lado, si se utilizan demasiados segmentos, el problema se puede complicar al mismo nivel que el subproblema original con término cuadrático.

Debido a lo anterior, es fácil notar que existe un *trade-off* entre el número de tramos a elegir para realizar la linealización (la velocidad de resolución de cada subproblema), y la calidad de la solución obtenida. Afortunadamente esta situación se puede equilibrar, en las primeras iteraciones del algoritmo, al elegir un tamaño muy pequeño del ponderador ρ que contrapesa el efecto negativo de elegir pocos segmentos lineales dentro de la aproximación, ya que esto permite que muy pocas variables se vean forzadas a converger, pero muy velozmente, sin empeorar así el valor de la función objetivo.

$$x^2 - 2x\bar{x} + \bar{x}^2 \rightarrow 2x - 2x\bar{x} + \bar{x}^2 \quad (5.2)$$

Luego de múltiples pruebas, para la instancia de menor tamaño del Laja, se llega a la conclusión de utilizar 5 tramos para la aproximación lineal del parámetro ρ , de esta forma, los subproblemas no se vuelven tan complejos y se resuelven en tiempo razonable. Las pruebas consistieron en resolver el problema para distinta cantidad de tramos para el término ρ , obteniendo el menor gap de optimalidad para 5 tramos. En anexos E se puede ver los resultados para la instancia de la Cuenca del Laja.

5.2.4. Ajustes del parámetro ρ de penalización

El parámetro ρ presente en el algoritmo PH (en la sección 5.1 paso 11), corresponde al ponderador relativo entre forzar la no-anticipatividad y cumplir el término proximal como se puede apreciar en la función objetivo del subproblema modificado:

$$\min_{x_s \in Q_s} f_s(x_s) + \omega_s x_s + \frac{\rho}{2} \|x_s - \bar{x}\|^2 \quad (5.3)$$

Donde ω se va actualizando según:

$$\omega = \rho(x - \bar{x}) + \omega_{i-1} \quad (5.4)$$

Este parámetro es muy sensible dentro del algoritmo como ya fue documentado por Mulvey y Vladimour a finales de los 80's [21]. Del mismo modo, a partir de como se contruye el ponderador ω , el valor de ρ puede interpretarse como el precio fijado por desviarse de la solución ponderada, y así ω puede ser interpretado de forma natural como el cobro extra que se debe incurrir dentro de la función objetivo con tal que al subproblema le sea más conveniente tomar una solución no anticipativa (dentro del espacio factible).

Es claro que este parámetro tiene importantes implicancias en la resolución, por lo tanto, si se puede encontrar un mejor sistema de precios que mejore la relación de las variables entonces el algoritmo convergerá más rápido y con un rendimiento superior en términos de función objetivo y calidad de la solución. El punto de partida de estas estrategias testeadas proviene del estudio publicado por Watson y Woodruff en [25]

Las estrategias testeadas y estudiadas en el presente trabajo fueron, ρ : FIX, CP, SEP, Taylor y combinaciones de ellas. Las primeras tres estrategias fueron expuestas en [25], mientras que la estrategia Taylor fue desarrollo de Badilla en [26]. Finalmente la selección de ρ estuvo subordinada a evitar la convergencia anticipada de las variables debido a la linealización extrema en forma de V (solo dos tramos), la rapidez de cómputo en cada iteración y la velocidad de resolución del problema completo.

En base a lo anterior, los requisitos expuestos fueron alcanzados mediante una estrategia mixta; para las variables con costos asociados (producción de centrales térmicas y potencia no satisfecha) se utilizó SEP, mientras que para las variables sin costos asociado (producción centrales hídricas, ángulos, flujos de potencia, entre otras), se utilizó un ρ fijo (FIX) que fue variando según la cantidad de escenarios.

A modo de conclusión general, el tamaño de ρ para la metodología FIX, crece con el número de escenarios, esto ocurre debido a que con mayor cantidad de escenarios, hay más discrepancias entre éstos, haciendo la convergencia más difícil y disminuyendo la cantidad de variables fijadas por iteración. Por lo tanto, para no tardar muchas iteraciones (y tiempo de cómputo), el valor de ρ va aumentando.

En las variadas implementaciones de la metodología Taylor no se obtuvo resultados sa-

tisfactorios, se observa una oscilación en la convergencia de las variables; mientras más se acercan las variables entre sí, el parámetro ρ causaba que en la siguiente iteración se volvieran a distanciar, por lo que no se conseguía la convergencia. En Anexos F, se puede observar en detalle las estrategias para el cálculo de ρ (FIX, SEP, CP, Taylor).

5.2.5. Bundling de escenarios

El concepto de bundling de escenarios hace referencia a una técnica donde se trabaja con grupos de escenarios que al menos comparten un conjunto de información del árbol original. Si se quiere, en el lenguaje de Teoría de Grafos, corresponde a un sub-árbol de escenarios. En la ilustración 5.1 se puede observar un grupo de tres Bundles para un árbol original de cinco escenarios:

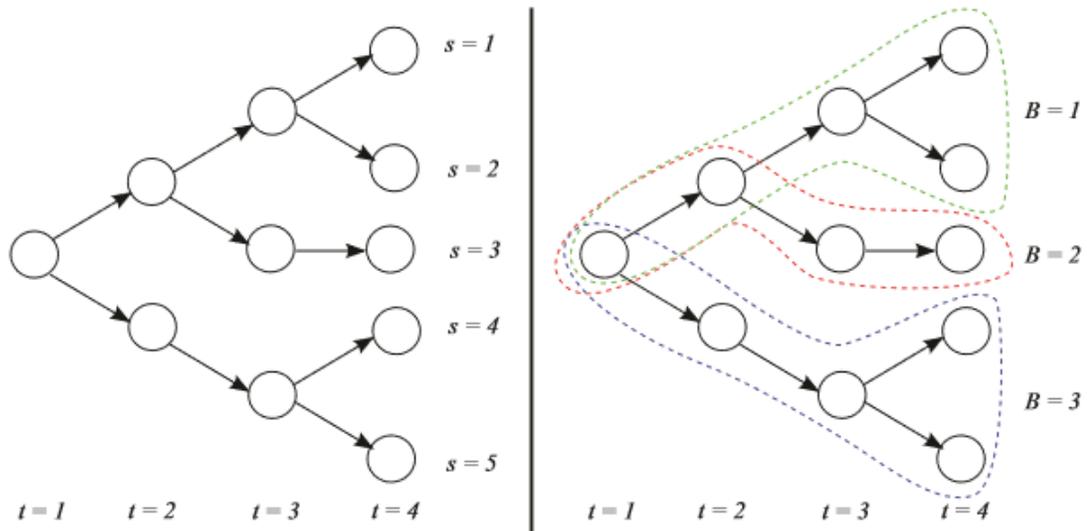


Ilustración 5.1: Bundling de Escenarios.

De esta forma, el problema se puede descomponer en sub-árboles (macro-escenarios), que son problemas más complejos que los sub-problemas (problemas por escenario), pero todavía es fácil resolver su determinista equivalente. Es claro que existe un *Trade-off* entre la cantidad de escenarios agrupados en un bundle y la complejidad de la resolución de cada sub-árbol, en el límite, si se agrupan todos los escenarios en un solo sub-árbol, se estaría resolviendo el árbol original.

El objetivo final de este enfoque consiste en que si los escenarios se resuelven de forma más veloz que el tiempo que se toma el algoritmo PH en realizar el resto de sus pasos (cálculo de promedios, actualización de penalizadores, cálculo de ρ , entre otros), será conveniente reducir el número de veces que se resuelve cada escenario, pues en el caso original se resuelven muchas

iteraciones de problemas sencillos, mientras que en el caso con bundles, se resuelven menos sub-problemas (menos iteraciones), pero los sub-problemas son más complejos y requieren un mayor tiempo de resolución.

Para su aplicación en esta tesis se utiliza la técnica de selección aleatoria para realizar los bundles, que no supere el 5% del total de escenarios, es decir, si se tiene un árbol con 100 escenarios, se realizan 5 *random bundles*, quedando 5 macro escenarios a resolver, y 90 escenarios normales.

Cabe notar que una elección inteligente de bundling de escenarios, puede tener importantes impactos en los tiempos de ejecución, pero ese no es el foco de la presente tesis, por lo que solo se utiliza random bundles.

5.2.6. Criterio de término

El criterio de término del algoritmo (paso 8 en 5.1), busca finalizar luego de alcanzar la tolerancia mínima (ε) para aceptar la convergencia del problema (es decir, cuando todos los escenarios se parecen suficientemente a su óptimo correspondiente en cada nodo). En la práctica éstos conllevan a que el método presente convergencia muy lenta, y que la función objetivo empeore a medida que se fuerce la convergencia.

Dado lo anterior, se establece el número de iteraciones como criterio de término, de tal manera que si alcanzan a converger suficientes variables, se puede proceder a resolver la formulación extendida compacta con variables fijadas, de manera eficiente.

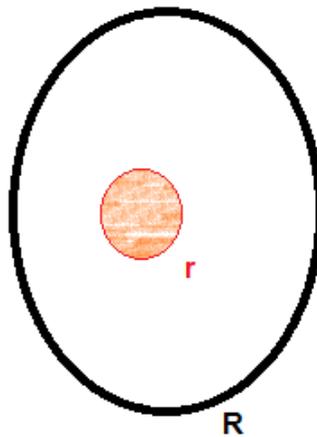


Ilustración 5.2: Región de la Formulación extendida compacta.

En la ilustración 5.2, la región R representa el espacio factible del problema original. Luego de $k^{max} = 20$ iteraciones (o se está a una distancia de $\varepsilon = 0,05$ del óptimo), se fijan

las variables que convergieron, por lo que la región factible se reduce a r (en rojo). Esta región representa un problema mucho más sencillo de resolver para la formulación extendida.

Capítulo 6

Implementación

Para realizar la implementación de PH se utiliza el software Pyomo/PYSP, generado por Sandia National Labs [22], que consiste en un paquete específico para resolución de problemas de optimización estocástica, todo soportado por el lenguaje de programación Python.

El lenguaje para realizar el modelamiento de problemas se denomina Pyomo y el paquete para problemas estocásticos PySP. Este paquete tiene una estructura bien definida para la notación de árboles de escenarios, permitiendo construir múltiples instancias de manera rápida, facilitando los tests.

Además Pyomo cuenta con un módulo dedicado para PH (-runph) que favorece enormemente la implementación de las distintas técnicas para el cálculo de ρ (FIX, SEP, CP), utilización de Warmstart, Bundles, Linealización del término cuadrático proximal, entre otras.

Los generadores de escenarios son implementados mediante scripts programados en Python 2.7.10, además se utilizan scripts para traspasar los escenarios generados en PLP a formatos compatibles con Pyomo.

Los solvers empleados para las corridas de PH fueron Gurobi (gurobi/6.0.4) [27] y cplex (cplex/12.6.1) para su versión de 64 bits. Todo lo anterior soportado en un sistema operativo Unix, Ubuntu (Ubuntu 14.04.4) para instancias pequeñas. Mientras que la implementación para la instancia real del sistema eléctrico chileno (instancia de gran tamaño) se utilizó la infraestructura de supercómputo del NLHPC (del inglés *National Laboratory for High Performance Computing*), del centro de modelamiento matemático (CMM), en un sistema operativo Linux.

Los datos utilizados en estos experimentos fueron entregados por el Departamento de Ingeniería Eléctrica, específicamente el Centro de Energía de la Universidad de Chile. Se accedió a los caudales históricos de los últimos 61 años, además de información específica de centrales, líneas de transmisión y demandas.

6.1. Implementación paralela PYSP

Para la implementación paralela se emplea la aplicación pyro basado en una lógica de memoria distribuida, que permite alcanzar la capacidad total del super computador de NLHPC, mediante el traspaso de mensajes.

Primero se inicia un servidor de nombres (pyomo-ns), que es el encargado de identificar cada uno de los procesadores en los cuales se ejecutará el problema.

Posteriormente, se inicia un servidor de despacho (dispatch-srvr), quien distribuye los distintos problemas en los *workers*.

Por último se inician los *workers* (phsolverserver), encargados de resolver los distintos problemas. Cabe notar que es posible iniciar más de un worker por procesador, pero para la ejecución del problema de gran tamaño se usa un worker por nodo.

En anexos G se pueden ver los archivos empleados para la paralelización, pueden ser de gran utilidad para implementaciones futuras de Progressive Hedging, y sobre todo, su implementación en NLHPC.

6.2. Comparación de Metodologías

Para realizar la comparación de metodologías de resolución se trabaja con dos enfoques.

1.- Árbol Base

El primer enfoque es llamado "Árbol Base" consiste en la comparación de metodologías de resolución del problema estocástico. Se busca comparar el algoritmo PH vs SDDP. Para esto se emplea un árbol de escenario en común, que se construye simplemente considerando las 51 realizaciones históricas de hidrologías, partiendo de un nodo inicial en común. Por lo tanto es un árbol bastante simple en el que solo se debe cumplir la no-anticipatividad en la primera etapa.

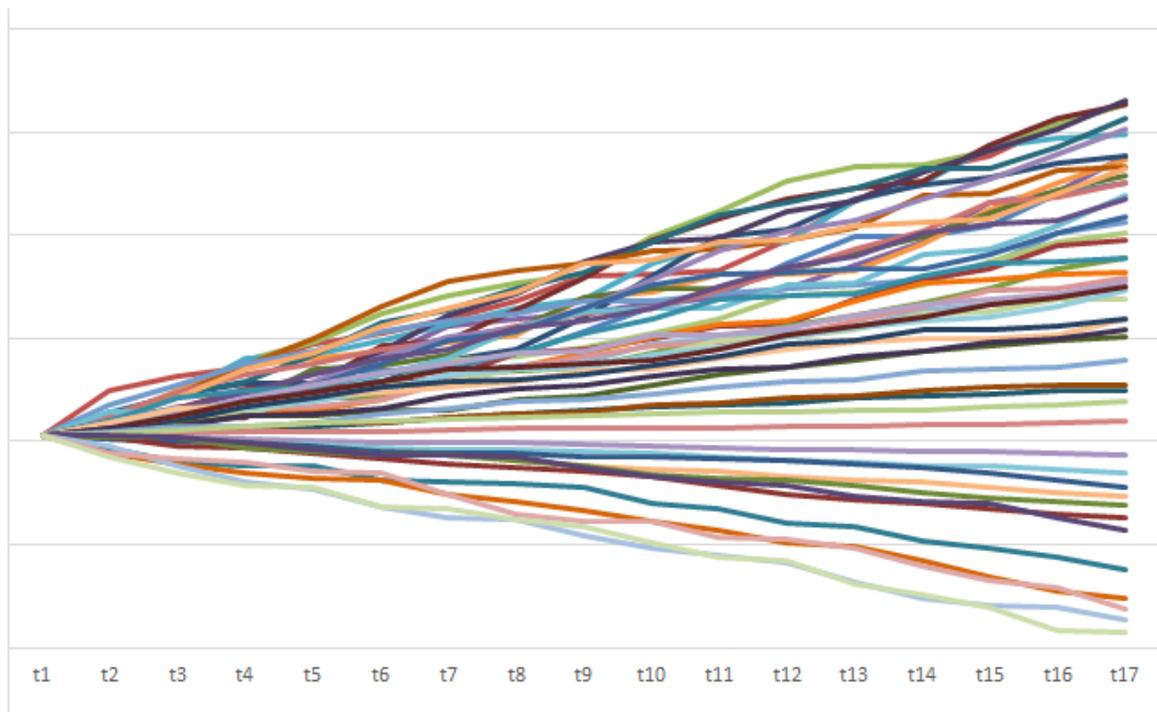


Ilustración 6.1: Representación teórica del árbol común utilizado, sin aperturas desde el segundo período.

Los criterios para comparar las soluciones son: Función objetivo, tiempo de resolución, iteraciones, Gaps, Peor-Mejor escenario.

2.- Comparación Árbol + Metodologías

El segundo enfoque consiste en la comparación entre generadores de escenarios sumado a la metodología de resolución, en otras palabras, se compararán la metodología de generación actual de árboles (llamada generador de escenarios PLP) resuelto con SDDP, con los nuevos generadores SYNTH y NSG, estos últimos siendo resueltos con PH.

SYNTH y NSG consisten en generación sintética de escenarios y generación según el vecino más cercano, respectivamente. Para ver más detalles de estos generadores consultar la sección 3.4.

Una de las principales dificultades de este trabajo de tesis, es la comparación de árboles de escenarios. La mejor forma encontrada de comparar árboles (y metodologías) es someter la solución de primera etapa obtenida, a realizaciones hidrológicas históricas.

En simples palabras, la metodología de comparación consiste en resolver los distintos árboles de escenarios y almacenando la solución de primera etapa. A partir de esta solución de primera etapa (que contiene, costos operacionales de $t=1$ y niveles de embalse al final del período), se resuelven N problemas determinísticos, que parten desde la etapa 2 y recibe como condiciones iniciales, los niveles de embalses al final de la etapa 1. Una vez resuelto los N problemas, que caracterizan hidrologías húmedas, secas y promedios, se procede a analizar los costos operacionales obtenidos.

Ejemplo 1.1

Se utilizará el mismo ejemplo revisado en la sección 2.5. En el siguiente ejemplo se observa un problema de 2 etapas, donde la segunda etapa muestra 3 posibles escenarios con probabilidad de ocurrencia $\frac{1}{3}$:

$$\begin{array}{rcccccc}
 \text{mín}_g & \{c_1^T \cdot g_1 & +\frac{1}{3}(c_2^T \cdot g_{21} & +c_2^T \cdot g_{22} & +c_2^T \cdot g_{23})\} \\
 \text{s.a.} & \vdots & \vdots & \vdots & \vdots & \\
 & \sum g_1 & \vdots & \vdots & \vdots & = d_1 \\
 & v_1 - v_0 + K \cdot g_1 & \vdots & \vdots & \vdots & = a_1 \\
 & & \sum g_{21} & \vdots & \vdots & = d_2 \\
 & -v_1 + & v_{21} + K \cdot g_{21} & \vdots & \vdots & = a_{21} \\
 & & & \sum g_{22} & \vdots & = d_2 \\
 & -v_1 + & & v_{22} + K \cdot g_{22} & \vdots & = a_{22} \\
 & & & & \sum g_{23} & = d_2 \\
 & -v_1 + & & & v_{22} + K \cdot g_{22} & = a_{23}
 \end{array}$$

Donde:

$-c_t^T$: Representa el costo variable de generación en periodo t.

$-g_{ts}$: Generación en el período t, en el escenario s.

$-d_t$: Demanda en el período t.

$-v_t$: Volúmen en el período t.

$-a_{ts}$: Afluente natural en el período t, escenario s.

$-K$: Rendimiento de central hídrica.

1. El primer paso consiste en generar los árboles de escenarios: Si este problema es representado por dos generadores de árboles escenarios distintos: G_1 y G_2 , la única diferencia entre estos dos generadores sería los valores de los afluentes en la etapa 2: a_{2s} , ya que ambos generadores tendrán el mismo nodo inicial (que no tiene incertidumbre).
2. Una vez generado los árboles de escenarios, se calcula el valor óptimo utilizando algún método de optimización bajo incertidumbre (para este caso se utilizan PH y SDDP). La solución de primera etapa es almacenada:

$$\hat{x}_1 = \{\hat{g}_1, \hat{v}_1\}$$

3. Bajo el supuesto de que se conoce lo que pasará después de la etapa 1, es decir, que ya no existe incertidumbre y que se puede encontrar la solución óptima. A partir de la solución de primera etapa, se resuelve el problema sin incertidumbre para la etapa 2

(este ejercicio es extendible a múltiples etapas). El problema queda como sigue:

$$\text{Min}_{x_2} \{c_1^T \cdot \hat{g}_1 + c_2^T \cdot g_2\} \quad (6.1)$$

s.t.

$$\sum g_2 = d_2 \quad (6.2)$$

$$- \hat{v}_1 + v_2 + K \cdot g_2 = a_2 \quad (6.3)$$

Es decir:

$$c_1^T \cdot \hat{g}_1 + \text{Min}_{x_2} \{c_2^T \cdot g_2\} \quad (6.4)$$

s.t.

$$\sum g_2 = d_2 \quad (6.5)$$

$$- \hat{v}_1 + v_2 + K \cdot g_2 = a_2 \quad (6.6)$$

Donde a_2 representa la hidrología real en la etapa 2.

Para hacer la comparación de metodologías de generación se utilizan hidrologías históricas representativas de escenarios secos, húmedos y promedios. Cabe notar que esta metodología de comparación se puede extender a más períodos. Para esta tesis el horizonte de comparación es un año de operación.

6.3. Recursos computacionales utilizados

Para resolver las instancias tanto en su formulación paralela, como extendida se utilizan distintas infraestructuras de cómputo.

Para los test de instancias de menor tamaño se utiliza un laptop cuyas características se presentan a continuación:

Laptop

Sistema Operativo	: Ubuntu 14.04.4
Memoria RAM	: 8.00 GB
N de procesadores	: 4
CPU	: AMD A8-6410.
Solvers	: Gurobi 6.0.4, Cplex 12.6.1
Scripts	: PYTHON 2.7.10, BASH
Middleware	: Pyomo/PYSP Sandia National Labs [22].

Mientras que para instancias de gran tamaño se utiliza la infraestructura de supercómputo del NLHPC, dispuso los siguientes recursos:

NLHPC - Leftrararu

Sistema Operativo	:	Linux
Memoria RAM	:	48.00 GB
N de procesadores	:	120
CPU	:	2 x Intel Xeon E5-2660 10 cores C./U.
Solvers	:	Gurobi 6.0.4, Cplex 12.6.1
Scripts	:	PYTHON 2.7.10, SBASH
Middleware	:	Pyomo/PYSP Sandia National Labs [22].

Por otro lado, para las instancias de comparación PLP-SDDP, los recursos utilizados se enlistan a continuación:

PLP - SDDP

Sistema Operativo	:	Ubuntu server 14.04 LTS
Memoria RAM	:	256.00 GB
N de procesadores	:	64
CPU	:	Intel(R) Xeon(R) CPU E7-8867 v3 @ 2.50GHz
Solvers	:	Cplex 12.6.1
Lenguaje	:	GNU Fortran (Ubuntu 4.9.3-13ubuntu2) 4.9.3

6.4. Instancias

Para realizar las pruebas se trabaja con dos instancias principalmente. Una de menor tamaño, que representa la cuenca del Laja; el objetivo de esta instancia es la calibración del modelo y la definición de metodos de comparación preliminares. La segunda instancia, aborda un problema de mayor tamaño que representa al sistema eléctrico chileno en su totalidad, SIC y SING. Sobre esta última instancia se desarrolla el análisis más significativo del presente trabajo.

A continuación se hace la descripción de cada una de las instancias.

1.- Cuenca del Laja

En la tabla 6.1 se puede ver un resumen de la instancia de menor tamaño.

La red eléctrica involucrada en las simulaciones se puede apreciar a continuación:

Tabla 6.1: Descripción instancia Cuenca del Laja

Item	Cantidad
Demandas	3
Líneas de transmisión	6
Nodos	5
Centrales de Falla	5
Centrales Térmicas	2
Centrales de Embalse	1
Centrales de Pasada en Serie	2
Centrales de Pasada Independientes	0
Total Centrales	10
Centrales Pivotes	0

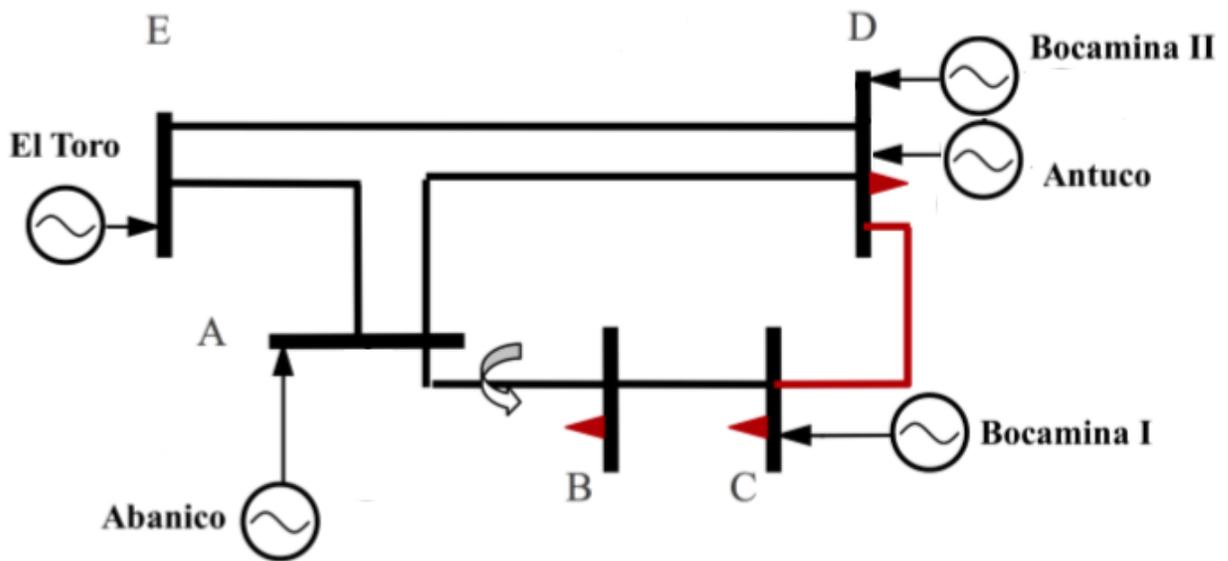


Ilustración 6.2: Red eléctrica instancia Cuenca del Laja.

Con el objetivo de entregar más detalle sobre la instancia, a continuación se exponen los parámetros relativos a ésta:

Por último, la ilustración 6.3 expone la red hidráulica subyacente. La cual muestra la comunicación hídrica entre las filtraciones, generaciones y vertimientos, además de los afluentes

Tabla 6.2: Parámetros de línea de transmisión - Cuenca del Laja

Línea	Límite eléctrico [MW]	Reactancia [ohm]
A-B	350	17.44
A-D	150	6.10991
A-E	400	5.85
B-C	350	13.0751
C-D	240	2.86632
D-E	240	7.24008

Tabla 6.3: Parámetros de generadores térmicos - Cuenca del Laja

Generador	Máx. Capacidad [MW/hr]	Costo Lineal [\$/MWh]	Costo Cuadrático [\$/MWh]
Bocamina I	128	46	0
Bocamina II	350	43.2	0

Tabla 6.4: Parámetros generadores hidroeléctricos - Cuenca del Laja

Generador	Máx. Capacidad [MW/hr]	Máx Caudal [m3/seg]	Parám. de turbinación (ρ, ρ')	Parám. de filtración (κ', κ'')
El toro	450	93.75	(4.8 , 0.0)	(0.0 , 30.8)
Antuco	320	200	(1.6 , 0.0)	-
Abanico	136	113.3	(1.2 , 0.0)	-

Tabla 6.5: Características de la demanda - Cuenca del Laja

Barra	Demanda Media [MW]
B	300
C	300
D	400

naturales para cada central¹.

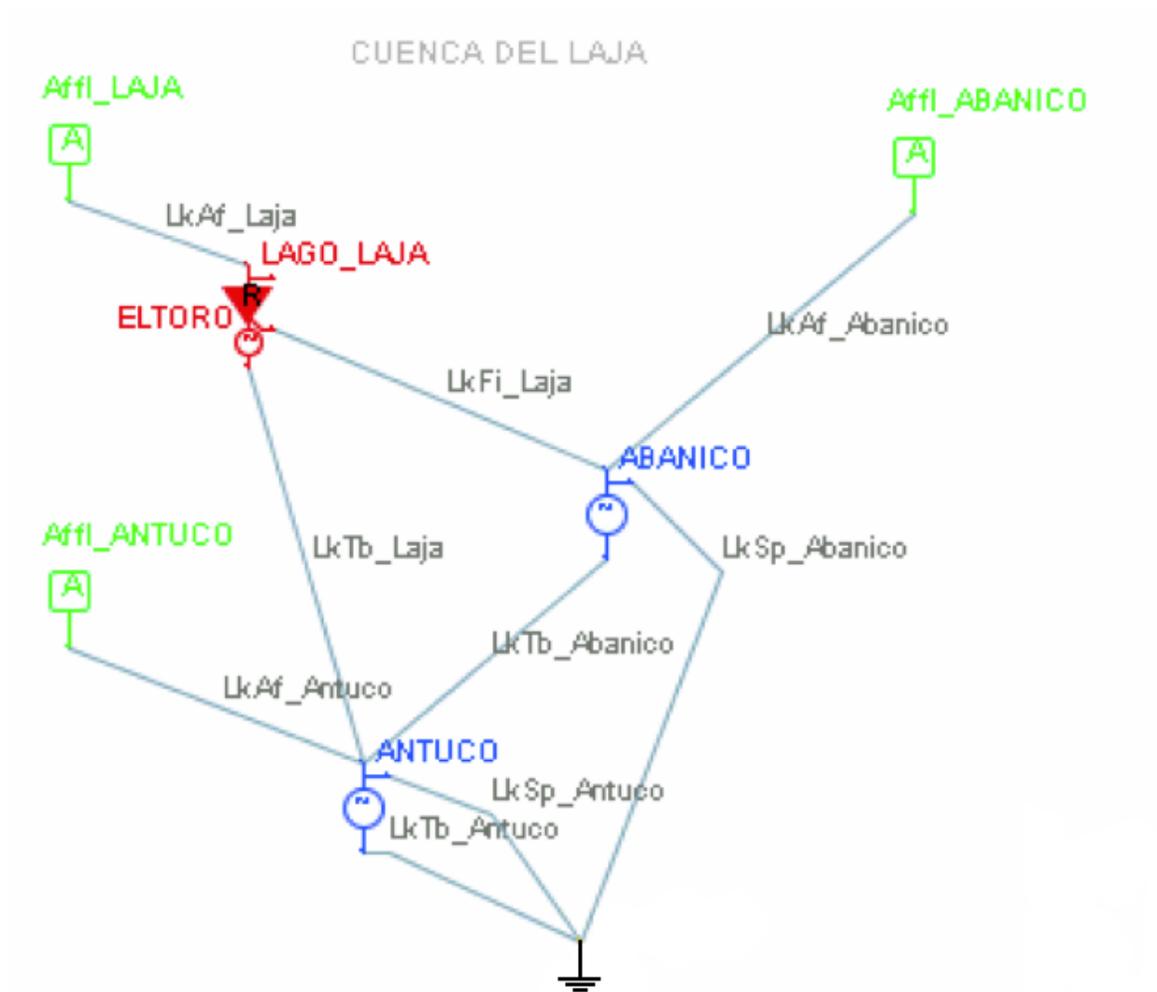


Ilustración 6.3: Red hidráulica instancia Cuenca del Laja.

Donde:

- $LkAf_i$: Irrigación de afluente Natural que alimenta a i.
- $LkTb_i$: Caudal turbinado por i.
- $LkSp_i$: Caudal derramado por i.
- $LkFi_i$: Caudal Filtrado por i.

Para la instancia de menor tamaño, el horizonte de planificación es de 5 meses, con 2 bloques por mes, tomando una tasa de descuento del 10 % mensual.

¹Ilustración aportada por el Centro de Energía

1.- Sistema completo (SIC-SING)

Esta instancia busca representar a cabalidad el Sistema Interconectado Central (SIC), sumado al Sistema Interconectado del Norte Grande (SING); Debido a la extensión, no se documenta todo el detalle de la descripción como en la instancia anterior:

Caso Uninodal

El caso Uninodal representa un problema simplificado, en el cual todas las centrales están conectadas a un mismo nodo (excepto las centrales pivotes), por lo tanto, no existen líneas de transmisión. En la tabla 6.6 se puede ver el caso Uninodal.

Tabla 6.6: Descripción instancia Uninodal SIC-SING

Item	Cantidad
Demandas	1
Líneas de transmisión	0
Nodos	2
Centrales de Falla	1
Centrales Térmicas	306
Centrales de Embalse	9
Centrales de Pasada en Serie	69
Centrales de Pasada Independientes	73
Total Centrales	699
Centrales Pivotes	42

Caso Multinodal

El caso Multinodal presenta la misma cantidad de centrales que el caso anterior, pero conectadas a distintos nodos. Por lo tanto, además de los nodos, se suman líneas de transmisión (red eléctrica) y centrales de falla.

El resumen de dicha instancia se puede apreciar en la tabla 6.7

Para estas instancias de gran tamaño, el período de planificación utilizado es de 103 meses (8.58 años), divididos en 3 bloques cada uno. Además, se utiliza una tasa de descuento anual del 10 %.

Tabla 6.7: Descripción instancia Multinodal SIC-SING

Item	Cantidad
Demandas	182
Líneas de transmisión	413
Nodos	242
Centrales de Falla	242
Centrales Térmicas	306
Centrales de Embalse	9
Centrales de Pasada en Serie	69
Centrales de Pasada Independientes	73
Total Centrales	699
Centrales Pivotes	42

Capítulo 7

Resultados

En este capítulo se exponen los principales resultados del presente trabajo de tesis. Primero se presenta el análisis referente al algoritmo PH, donde se manifiesta tanto su convergencia, como el comportamiento de la función objetivo por iteración.

Segundo, se muestran los resultados para las tres instancias trabajadas: Cuenca del Laja (Multinodal), SIC - SING Uninodal y SIC SING Mutinodal.

Como se mencionó en la sección anterior, se realiza una comparación de metodologías, para la cual se utiliza un árbol base común, resuelto con SDDP, PH y PH paralelo. Las métricas a analizar son: Tiempos de ejecución, N de iteraciones, GAP, Función Objetivo y el desempeño en escenarios extremos.

Posteriormente, se muestran las metodologías generadoras de escenarios NSG y SYNTH (sección 3.4); Para cada una de estas metodologías se estudia la cantidad de aperturas (N de escenarios) con mejor desempeño considerando tiempos de ejecución, función objetivo y número de iteraciones para llegar al óptimo.

El tercer análisis tiene relación con la comparación entre Árbol de Escenario + Metodología de resolución. Para este caso, se hacen comparaciones entre: SDDP + Generador de escenarios PLP, PH + Generador de escenario NSG y PH + Generador de escenario SYNTH.

La forma de comparar los árboles + metodología se divide en tres pasos:

1) Se resuelve el problema eléctrico para cada árbol + Metodología. Una vez encontrado el óptimo, se almacenan las soluciones de primera etapa: niveles de embalses y costos operacionales.

2) Con las soluciones de primera etapa de cada árbol + metodología, se resuelve el problema eléctrico sin incertidumbre. Los problemas resueltos representan realizaciones históricas de años secos, promedio y húmedos.

3) Por último, se comparan las funciones objetivo obtenidas en cada escenario histórico, notando qué metodología tiene mejor desempeño.

7.1. Convergencia PH

Una de las métricas importantes para evaluar el desempeño del algoritmo PH, es la convergencia de la solución por iteración.

En el gráfico 7.1 se puede observar el progreso en la convergencia a medida que aumentan las iteraciones. La forma de medir la convergencia es la distancia euclidiana entre las soluciones, normalizando este valor, para que sea de fácil interpretación.

El caso expuesto, corresponde a la instancia SIC - SING Uninodal, con horizonte de planificación 103 meses (8.58 años), 3 bloques por mes, con 768 escenarios finales. Se utiliza esta instancia como representativa, ya que no es tan compleja como el caso multinodal. En general, el comportamiento de la convergencia es bastante parecido para el resto de las instancias. Para mejorar el desempeño del algoritmo se utilizó Bundles de escenarios del 5% del total de escenarios (38 random bundles), Warmstarts y GAP variable, partiendo con una tolerancia del 15% para la primera iteración y disminuyendo progresivamente hasta un 0,5%. Las metodologías utilizadas para calcular el parámetro ρ es mixta: SEP + FIX.

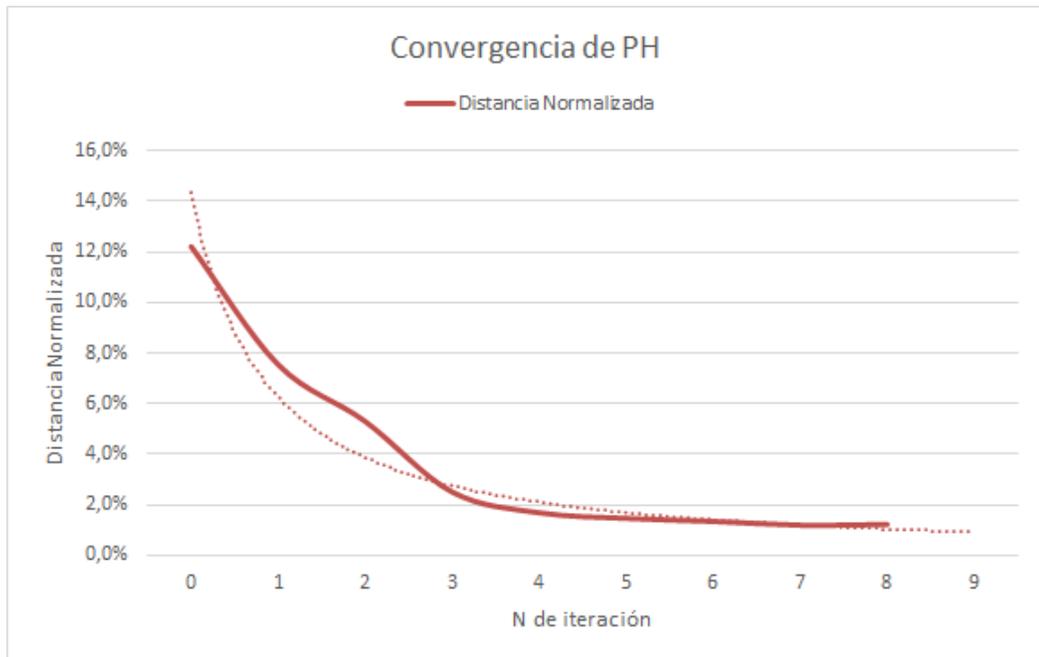


Ilustración 7.1: Convergencia de PH, Caso SIC-SING Uninodal, 768 escenarios.

En la ilustración 7.1 se puede apreciar el progreso en la convergencia. Las primeras 4 iteraciones muestran la mayor disminución en las distancias euclidianas entre las variables. Desde la quinta iteración en adelante, la curva se comienza a aplanar, disminuyendo su progreso por iteración. En el gráfico 7.2, se puede ver la variación porcentual de la convergencia por cada iteración.

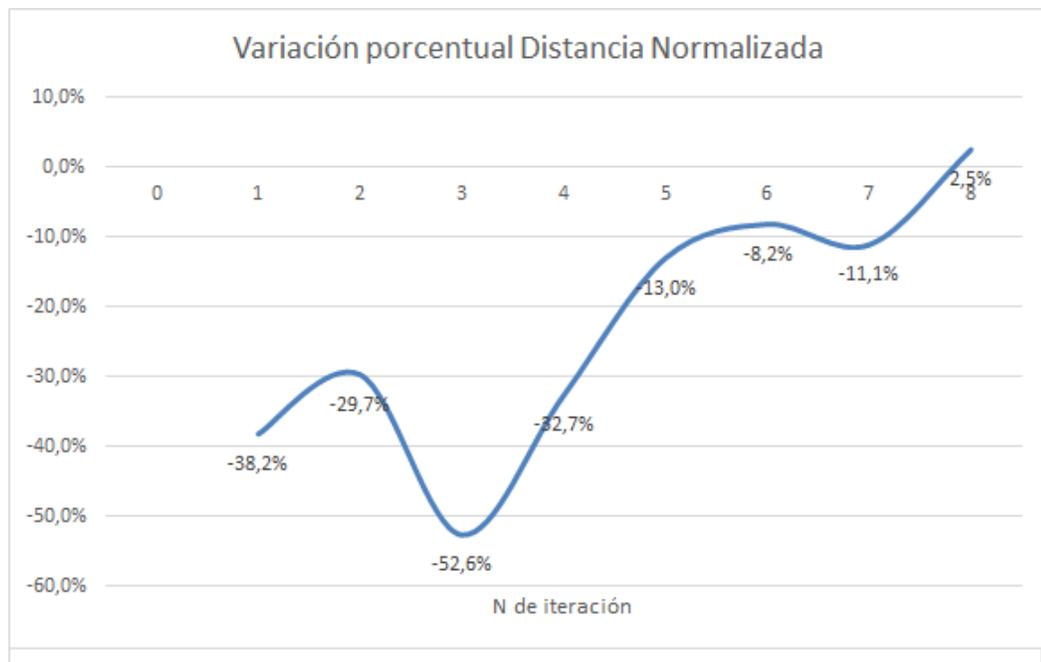


Ilustración 7.2: Variación porcentual de PH por iteración, Caso SIC-SING Uninodal, 768 escenarios.

En el gráfico 7.2 se aprecia que la mayor disminución en las distancias euclidianas ocurre en la tercera iteración. Esto se explica principalmente porque en la iteración 0 el problema es resuelto sin restricción de no anticipatividad. Posteriormente, desde la iteración 1 en adelante, se agrega el penalizador sobre las variables que deben cumplir la no anticipatividad. El penalizador se va ajustando iteración a iteración, forzando la convergencia, logrando que en la tercera iteración se obtenga el mayor progreso.

Desde la iteración 5 en adelante, la velocidad de convergencia disminuye, incluso en la iteración 8 se produce un alejamiento entre las variables, lo que se puede explicar por un sobre ajuste del penalizador.

Después de variadas iteraciones, se pueden generar oscilaciones en la convergencia, sin que éstas sean significativas para la F.O. y variables. Por lo anterior, es importante fijar la cantidad máxima de iteraciones (o tiempo de ejecución), para no generar rutinas ineficientes.

En la ilustración 7.3 se puede ver la variación de la función objetivo. A diferencia del gráfico de la convergencias, se ve una curva creciente, lo que refleja que iteración a iteración la F.O. va "empeorando", debido a que se va forzando la convergencias de las variables, en otras palabras, se resuelven problemas más estricto, a medida que aumentan las iteraciones.

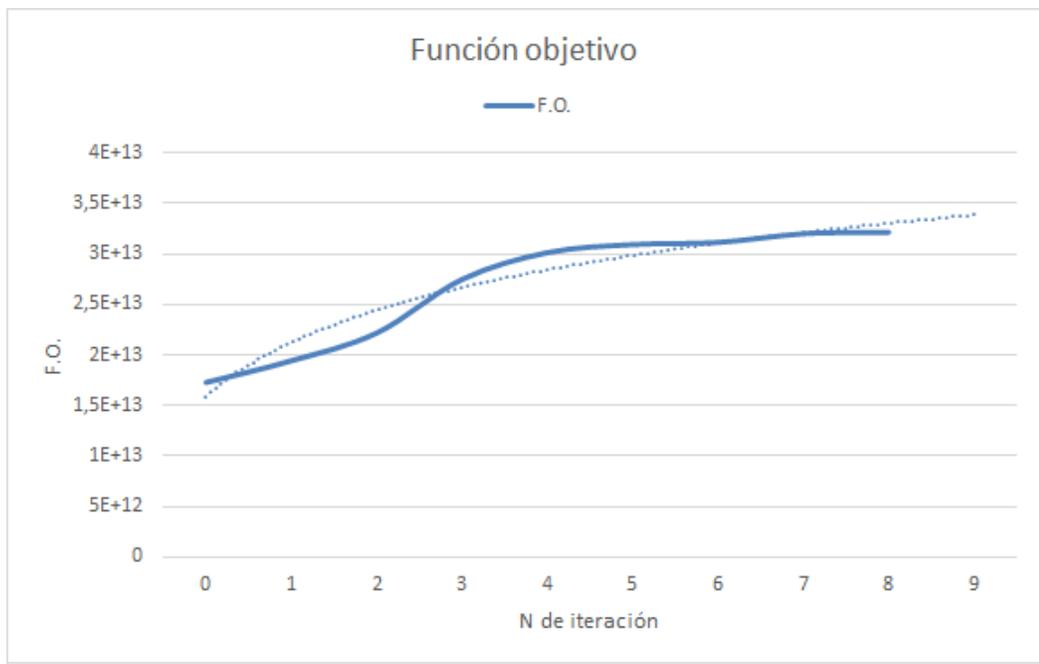


Ilustración 7.3: F.O por iteración, Caso SIC-SING Uninodal, 768 escenarios.

Al igual que para la convergencia, el gráfico 7.4 muestra la variación porcentual de la F.O. por iteración. En la iteración 3 se tiene el mayor aumento porcentual (23,91%), lo que es consistente con la convergencia de las variables, que en la misma iteración muestra su mayor progreso (según gráfico 7.2).

Se observa además que para toda iteración, el crecimiento de la F.O. es positivo. Además, se aprecia que desde la iteración 5 en adelante el crecimiento de la F.O. disminuye considerablemente, en comparación a las iteraciones anteriores.

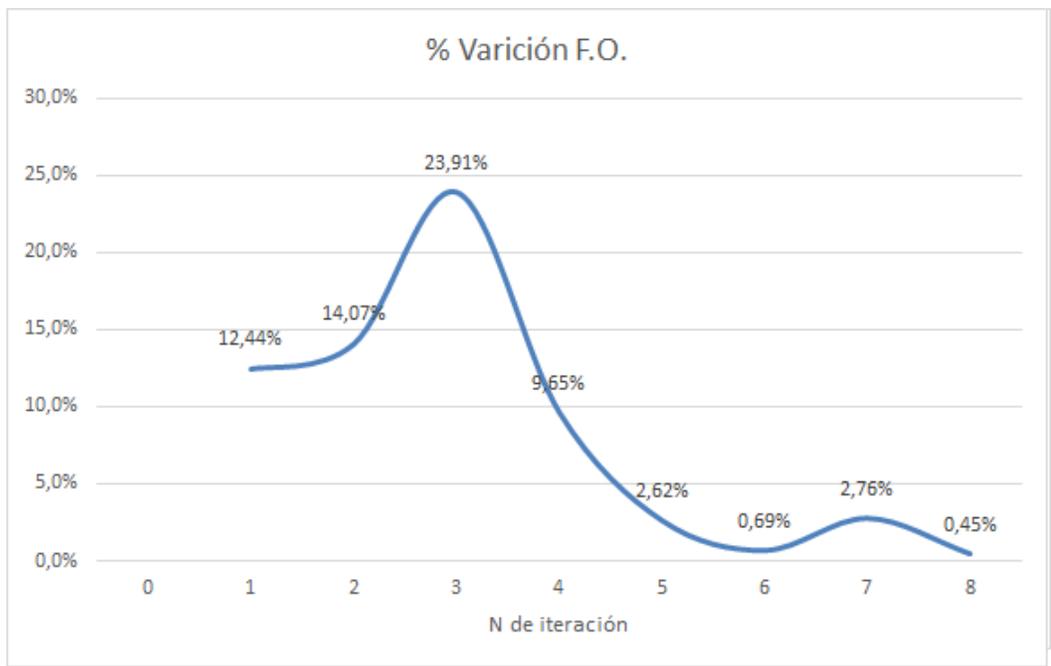


Ilustración 7.4: Variación porcentual de F.O por iteración, Caso SIC-SING Uninodal, 768 escenarios.

Para más detalles sobre la convergencia del algoritmo revisar Anexos H.

7.2. Cuenca del Laja

En esta parte se expone los resultados para la instancia Multinodal de menor tamaño "Cuenca del Laja". El objetivo es comparar las metodologías de resolución PH vs SDDP, analizando tiempos de ejecución, número de iteraciones, Gaps, F.O., peor y mejor escenario. Todos los análisis se hacen en torno a un árbol en común, con 51 escenarios equivalentes a las 51 realizaciones históricas.

Para esta instancia de menor tamaño, el horizonte de planificación es de 5 meses, con 2 bloques por mes. La configuración utilizada para PH es la descrita en las secciones anteriores: GAP ajustable desde el 15% al 1%, Warmstart, ρ mixto: SEP + FIX.

7.2.1. Árbol Base

La tabla 7.1 se exponen los principales resultados de la instancia, para SDDP, PH y PH paralelo:

Tabla 7.1: Cuenca del Laja - Árbol base

	SDDP	PH	PH Paralelo
Tiempo de Ejecución [seg]	2	11,29	15,66
N de Iteraciones	11	2	2
Gap	2,37E+04	0,0	0,0
F.O.	7,50E+11	7,35E+11	7,35E+11
Peor Escenario	8,697E+11	8,44E+11	8,44E+11
Mejor Escenario	7,17E+11	6,46E+11	6,46E+11

Tiempo de Ejecución

Al comparar los tiempos de ejecución, se observa que la metodología SDDP resuelve el problema con mayor velocidad. PH tarda 4,6 veces más en resolver que SDDP, y PH paralelo demora 6,8 veces más. En conclusión, los tiempos de ejecución son menores para SDDP.

Asímismo, se puede ver que el tiempo de ejecución de PH paralelo, es mayor que para PH sin paralelizar. Esto ocurre debido a que para instancias de menor tamaño, como es este caso, no se justifica el tiempo que utiliza PH paralelo en la comunicación entre los núcleos de procesamiento.

N de iteraciones

SDDP resuelve el problema en el máximo de iteraciones seteadas para esta instancia. PH tarda un menor número de iteraciones que SDDP. Cabe destacar que la comparación del número de iteraciones entre metodologías no tiene mucha relevancia, ya que la lógica de iteraciones de SDDP no es igual a la de PH.

Por otro lado, PH y PH paralelo, resuelven en el mismo número de iteraciones (2), sin alcanzar el máximo seteado para esta instancia (máximo: 30).

GAP

En terminos de exactitud, la solución de PH y PH paralelo tienen un mejor desempeño que SDDP, encontrando la solución óptima con total precisión. Mientras que SDDP, presenta un gap de $2,37E+04$, que se ve reflejado en la función objetivo. En el largo plazo, este Gap puede significar importantes diferencias en costos de operación.

Función objetivo

En terminos de función objetivo, PH y PH paralelo, muestran la misma función objetivo, esto ocurre siempre, para todas las instancias resultas. Esta función objetivo es levemente menor que la encontrada por SDDP, y esto se explica por el GAP antes expuesto.

Escenarios extremos

PH muestra un mejor desempeño para escenarios extremos, con menores costos en comparación con SDDP. Esto se explica principalmente porque PH resuelve el problema con mayor exactitud, para esta instancia de menor tamaño.

7.3. Sistema completo (SIC - SING)

En esta parte, se exponen los resultados para la instancia de gran tamaño, que representa la red eléctrica de Chile, SIC-SING.

El análisis se aborda en dos partes; primero se presentan los resultados del Caso Uninodal y posteriormente el Caso Multinodal.

Se realiza una comparación entre metodologías de resolución, SDDP vs PH, bajo un árbol de escenarios en común. Las métricas analizadas son las siguientes: Tiempo de ejecución, N de iteraciones, GAP, F.O. y su desempeño en escenarios extremos.

Para todos los resultados expuestos, se trabaja con un horizonte de planificación de 103 meses (8.58 años) y 3 bloques por mes. La configuración utilizada para PH es: random bundles del 5%, Warmstars y Gap ajustable (partiendo en un 15% hasta un 0,5% desde la tercera iteración en adelante). La metodología para el cálculo de ρ es mixta: SEP + FIX. Se utiliza un total de 120 procesadores.

7.3.1. Caso Uninodal

Como primera aproximación a la instancias de grán tamaño SIC-SING, se exponen los resultados bajo el supuesto Uninodal, es decir, que todas las centrales eléctrica y demandas, se encuentran conectadas a un mismo nodo.

Árbol Base

En la tabla 7.2 se pueden observar los principales resultados de esta instancia. En este caso, se comparan las metodologías de resolución en un árbol de escenarios común, que representa las realizaciones hidrológicas históricas:

Tabla 7.2: SIC - SING - Caso Uninodal - Árbol Base

	SDDP	PH	PH Paralelo
Tiempo de Ejecución [seg]	539	82.590	10.619
N de Iteraciones	36	6	6
Max de iteraciones alcanzado	si	No	No
GAP	1,27E+09	9,68E+08	9,68125E+08
F.O	3,21E+13	3,23E+13	3,23E+13
Peor Escenario	3,72E+13	3,72E+13	3,72E+13
Mejor escenario	2,75E+13	2,75E+13	2,75E+13

7.3.2. Caso Multinodal

En esta sección se exponen los resultados de la instancia de gran tamaño: SIC + SING. Esta instancia es la que mejor respresenta el escenario energético chileno, ya que además de considerar todas las centrales eléctricas, se trabaja bajo el supuesto multinodal.

Debido al gran tamaño de la instancia, no es posible resolver con PH sin paralelizar, ya que la estructura de super cómputo, no cuenta con memoria suficiente para procesar todo en un solo nodo.

Árbol Base

En la tabla 7.3 se pueden observar los principales resultados de esta instancia:

Tabla 7.3: SIC - SING - Caso Multinodal - Árbol Base

	SDDP	PH Paralelo
Tiempo de Ejecución [seg]	1.144	26.920
N de Iteraciones	36	9
Max de iteraciones alcanzado	si	No
GAP	1,78E+09	1,35939E+09
F.O	3,40556E+13	3,43698E+13
Peor Escenario	3,9767E+13	3,97711E+13
Mejor escenario	2,93202E+13	2,93255E+13

7.3.3. Análisis de resultados Árbol Base - Sistema completo (SIC - SING)

Tiempos de ejecución

Respecto a los tiempos de ejecución, tanto en el caso Uninodal, como Multinodal, SDDP resuelve el problema considerablemente más rápido que PH paralelo.

PH paralelo tarda entre, 18 y 22.5 veces más que SDDP en resolver el problema. Lo que sugiere que para instancias de mayor tamaño (como árboles más complejos), esta diferencia en los tiempo de ejecución puede ser determinante en la elección entre una metodología y otra (SDDP vs PH).

Una explicación para estas importantes diferencias, es el lenguaje de compilación que se utiliza, SDDP trabaja con Fortran, mientras que PH utiliza Python, y según testeos realizados por Julia Lag org [28], la velocidad cómputo de Fortran es significativamente más rápida que Python ¹.

Al comparar los tiempos entre PH y PH paralelo, en el caso Uninodal, se observa que PH sin paralelizar tarda 6.7 veces más que PH paralelo. Este es un resultado esperable, ya que la paralelización del algoritmo es favorable para instancias de gran tamaño, debido a que el tiempo utilizado en el traspaso de información entre los núcleos computaciones, se ve justificado.

N de iteraciones

Al igual que para la instancia de menor tamaño, SDDP resuelve en un mayor número de iteraciones que PH, alcanzando su máximo de iteraciones, tanto en el caso Multinodal, como en el Uninodal. Por otro lado PH y PH paralelo, encuentran el óptimo en el mismo número de iteraciones. Este último resultado es obtenido para todas las corridas ejecutadas.

El algoritmo PH, llega a la última iteración cuando ocurre uno de los siguientes sucesos: se alcanza el GAP de optimalidad seteado ó se llega al máximo de iteraciones. Posterior a esto, se fijan las variables que convergieron, y se resuelve la formulación extendida.

Para las instancias expuestas, PH no alcanza el máximo de iteraciones, el algoritmo se detiene debido a que se superó la tolerancia mínima, en otras palabras, las soluciones por escenario, son suficientemente parecidas como para fijar las variables y resolver la forma extendida compacta.

Cabe destacar que las iteraciones de PH y SDDP no son comparables, ya que la lógica de iteraciones, es radicalmente distinta, para cada metodología de resolución.

GAP

Respecto al GAP de optimalidad, se puede ver que PH encuentra una solución más cercana al óptimo, con diferencias poco significativas. Al comparar los GAPs, se obtiene que SDDP

¹Para mayor detalle de las pruebas realizadas, revisar Anexos J

se encuentra entre 0,001 % y 0,00128 % más lejos del óptimo que la solución encontrada con PH.

Para el caso Uninodal, al comparar PH y PH Paralelo, se observa la misma distancia al óptimo.

Función objetivo

Al comparar las F.O. se tiene que SDDP encuentra un valor óptimo menor que PH. Este resultado, se puede atribuir al GAP de optimalidad antes expuesto.

PH y PH paralelo, encuentran igual valor óptimo.

Escenarios extremos

Al comparar los escenarios extremos, se encuentran diferencias poco significativas. Si se compara los escenarios con los costos operacionales más altos, se encuentra que PH presenta menores costos de operación, esta diferencia representa entre el 0,010 % y el 0,08 %. El mismo resultado se obtiene al comparar los escenarios con menores costos operacionales, PH tiene costos operacionales entre un 0,004 % y un 0,018 % menores que SDDP.

7.4. Metodologías Generadoras de Escenario

En esta sección se expone los resultados de las metodologías generadoras de escenario NSG y SYNTH, que fueron desarrolladas en extenso en la sección 3.4.

A grandes rasgos, estas metodologías se componen de 5 etapas:

- 1) Primero, para la etapa t se estima la distribución de los afluentes, en base a la información histórica.
- 2) Posteriormente, esta distribución se divide en segmentos, previamente definidos.
- 3) Para cada uno de las observaciones dentro del segmento, se calcula su probabilidad condicional. Con dichas probabilidades condicionales, se calcula una distribución condicionada para la etapa $t + 1$.
- 4) Con la distribución condicionada se obtienen los valores de hidrologías para la etapa $t+1$.
- 5) Por último, se realiza el mismo proceso para $t + 1$, desde el paso 1) en adelante.

La diferencias entre NSG y SYNTH, radica en que para NSG se utilizan las realizaciones históricas más cercana al valor condicional esperado, mientras que para SYNTH, se utiliza el valor condicional esperado, en otras palabras, SYNTH trabaja con escenarios sintéticos, y NSG con realizaciones históricas.

Para todos los análisis de las metodologías NSG y SYNTH, se resuelve con horizonte de 103 meses (8.58 años), y 3 bloques por mes.

7.4.1. Resultados SYNTH

A continuación se muestran los resultados del generador de escenarios SYNTH, para la instancia de gran tamaño SIC-SING. Se compara el resultado al resolver con la forma extensiva y con PH en paralelo. Para el caso de PH en paralelo, se utilizó Bundles de escenarios del 5 % del total de escenarios (random bundles), Warmstars y GAP ajustable, iniciando con tolerancia del 15 % hasta llegar al 1 %. Se trabaja con metodología mixta de cálculo de ρ : SEP + FIX. Para cada una de las pruebas se utiliza la mayor cantidad de núcleos de acuerdo a la cantidad de escenarios, con un máximo de 120 procesadores. No es eficiente utilizar más núcleos que escenarios, por que quedan procesadores ociosos, sin problemas para resolver.

En la tabla 7.4 se pueden ver los resultados relativos a esta metodología de generación de escenarios:

Tabla 7.4: Resultados Generador de escenarios SYNTH

Formulación Extendida		PH Paralelo SYNTH				
Tiempo [seg.]	F.O	Tiempo [seg.]	Iter.	Nucleos	N de escenarios	F.O.
1539	4,05E+13	1826	1	16	16	4,26E+13
3201	3,66E+13	5436	3	32	32	3,90E+13
8994	3,88E+13	9030	5	48	48	4,11E+13
18030	3,92E+13	3606	2	64	64	3,88E+13
37000	4,32E+13	1850	1	96	96	4,05E+13
$+\infty$		7692	4	120	128	4,12E+13
		14448	5	120	192	4,10E+13
		7728	2	120	256	3,90E+13
		4332	1	120	288	3,85E+13
		34593	6	120	384	3,67E+13
		31141	3	120	432	3,58E+13
		32872	2	120	576	3,54E+13
		48606	5	120	648	3,55E+13
		55300	2	120	768	3,33E+13
		51853	4	120	864	3,21E+13
		107542	7	120	1024	3,04E+13
		138272	5	120	1152	3,13E+13
		155533	6	120	1728	3,11E+13

Tiempos de ejecución

Al comparar los tiempos de ejecución, se observa que para los problemas con menos de 64 escenarios, la formulación extendida resuelve el problema más rápido. Esto se explica porque PH utiliza tiempo en comunicación (traspaso de información) entre los núcleos, almacenamiento de soluciones para warmstarts, cálculo de promedios y penalizadores, que no se justifica para problemas "pequeños".

Desde 64 escenarios en adelante, se puede ver que los tiempos de ejecución para PH son menores. Es más, desde los 128 escenarios, la formulación extendida tarda un tiempo indeterminado, por lo que no logra resolver el problema, debido a su complejidad y tamaño.

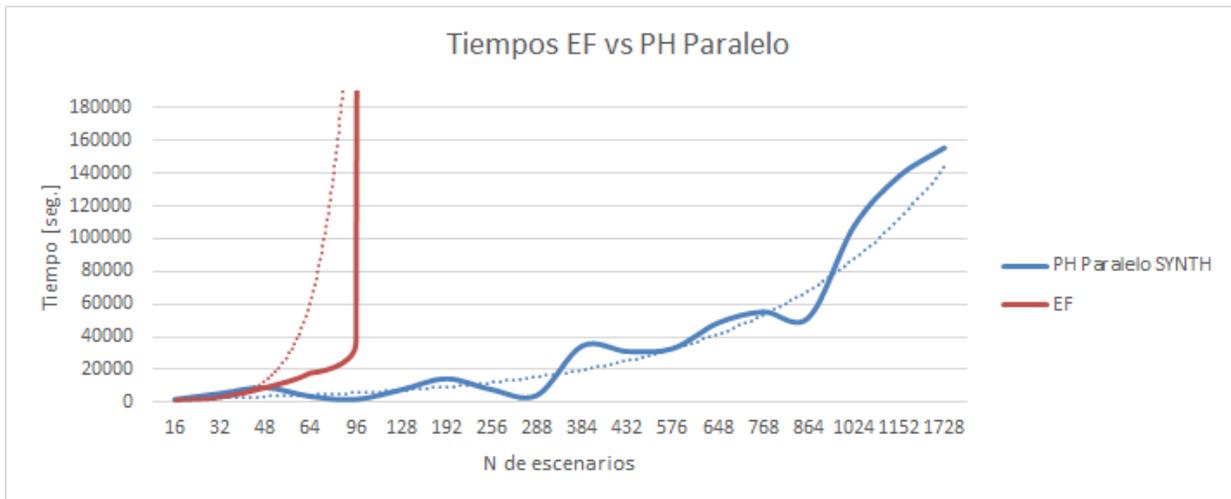


Ilustración 7.5: Tiempos de ejecución Generador de escenarios SYNTH - Formulación extendida (EF), Caso SIC-SING Multinodal

En la ilustración 7.5 se puede observar que para el generador SYNTH, el tiempo de ejecución es creciente con el número de escenarios. Puede generar oscilaciones, como en el caso de los 192 y 384 escenarios, que muestran crecimientos pronunciados en los tiempos, pero que disminuye para una cantidad mayor de escenarios (256 y 432, respectivamente). Una de las explicaciones para estos tiempos más elevados es el número de iteraciones. Como se expone en la tabla 7.4, para estos mismos casos, la cantidad de iteraciones también es mayor.

Función Objetivo

Según la ilustración 7.6 la función objetivo es menor al resolver la formulación extendida, hasta los 64 escenarios, luego, PH paralelo muestra mejores resultados.

Ambas formas de resolución están optimizando el mismo problema, por lo que, desde el punto de vista teórico, deberían encontrar la misma solución. Las diferencias se explican principalmente por la tolerancia, que para ambos casos es del 1% (en la última iteración). Otro factor que puede explicar estas diferencias es que para PH se utiliza GAP ajustable, que para la iteración 0 es del 15%.

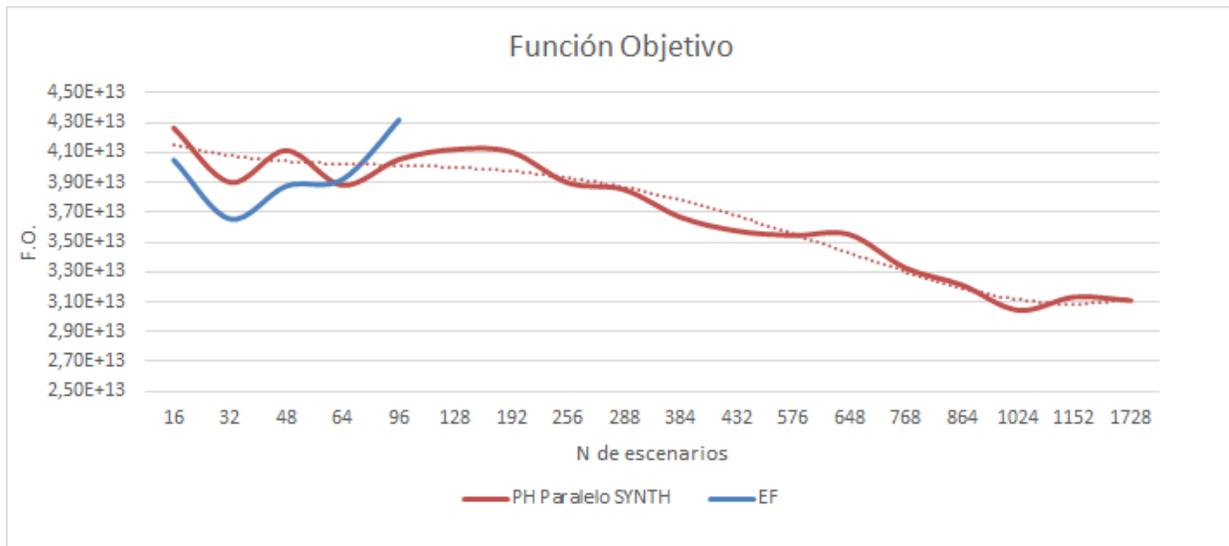


Ilustración 7.6: Tiempos de ejecución Generador de escenarios SYNTH - Formulación extendida (EF), Caso SIC-SING Multinodal

En general la solución obtenida es decreciente con la cantidad de escenarios, encontrando su mínimo en 1024 escenarios. La naturaleza decreciente se puede justificar, porque al utilizar el generador SYNTH, se definen intervalos de la distribución, tomando intervalos que incluyan realizaciones extremas. Mientras menos escenarios, estos escenarios extremos tienen más influencia en el problema, a medida que va aumentando la cantidad de escenarios, este efecto se va suavizando.

Al igual que en el tiempo de ejecución, se pueden observar oscilaciones en la función objetivo, que tienen relación con la tolerancia del problema y la naturaleza del problema resuelto (qué tan restrictivo es).

En conclusión, la configuración óptima encontrada es de 1024 escenarios, que muestra la mejor Función Objetivo $3,04E + 13$ con tiempos de ejecución razonable: 107542 seg. (1,25 días). La cantidad de iteraciones totales es 7.

7.4.2. Resultados NSG

A continuación se expone el análisis del generador de escenarios NSG. La instancia a analizar es SIC-SING Multinodal. Se comparará con la formulación extendida del mismo problema. La configuración utilizada para PH paralelo NSG, es igual al caso SYNTH; Random Bundles de escenarios al 5 %, Warmstars y GAP ajustable, desde el 15 %, decreciendo hasta el 1 % desde la tercera iteración. Para el penalizador ρ se utiliza la metodología mixta: SEP + FIX. Utilizandando la mayor cantidad de núcleos por escenario (con un máximo de 120).

La tabla 7.5, muestra los resultados obtenidos para la metodología NSG:

Tabla 7.5: Resultados Generador de escenarios NSG

Formulación Extendida		PH Paralelo NSG				
Tiempo [seg.]	F.O	Tiempo [seg.]	Iter.	Nucleos	N de escenarios	F.O.
2.285	3,82E+13	9007	5	16	16	4,31E+13
2.730	3,14E+13	7233	4	32	32	2,89E+13
10.038	3,57E+13	5431	3	48	48	3,82E+13
16.107	3,63E+13	1827	2	64	64	3,92E+13
40.862	3,59E+13	16229	9	96	96	3,56E+13
80.322	3,41E+13	21165	11	120	128	3,89E+13
$+\infty$		14443	5	120	192	3,80E+13
		34610	9	120	256	3,72E+13
		30257	7	120	288	3,69E+13
		23063	4	120	384	3,55E+13
		41511	4	120	432	3,61E+13
		82114	5	120	576	3,62E+13
		58369	6	120	648	3,48E+13
		165903	11	120	768	3,32E+13
		138907	9	120	864	3,16E+13
		176842	5	120	1024	3,15E+13
		248846	9	120	1152	2,94E+13
		259203	10	120	1728	3,07E+13

Tiempos de ejecución

Al comparar los tiempos de ejecución se puede observar que hasta los 32 escenarios, el tiempo de resolución para el formulación extendida es siempre menor. Para más de 32 escenarios, los tiempos son significativamente menores para PH Paralelo. Al igual como se expuso anteriormente, esto se explica por que para instancias de mayor tamaño, se justifica el tiempo asignado para comunicación entre núcleos, almacenamiento de soluciones (para warmstarts), cálculo de promedios y penalizadores que tiene que hacer PH, mientras que si las instancias son pequeñas, es ineficiente hacer la división horizontal del problema.

Asímismo, se puede observar que la formulación extendida, resuelve en un tiempo indeterminado el problema con 192 escenarios o más, debido al tamaño del problema. Por otro lado, en la ilustración 7.7, se puede apreciar que el tiempo de ejecución para NSG es creciente con el número de escenarios.

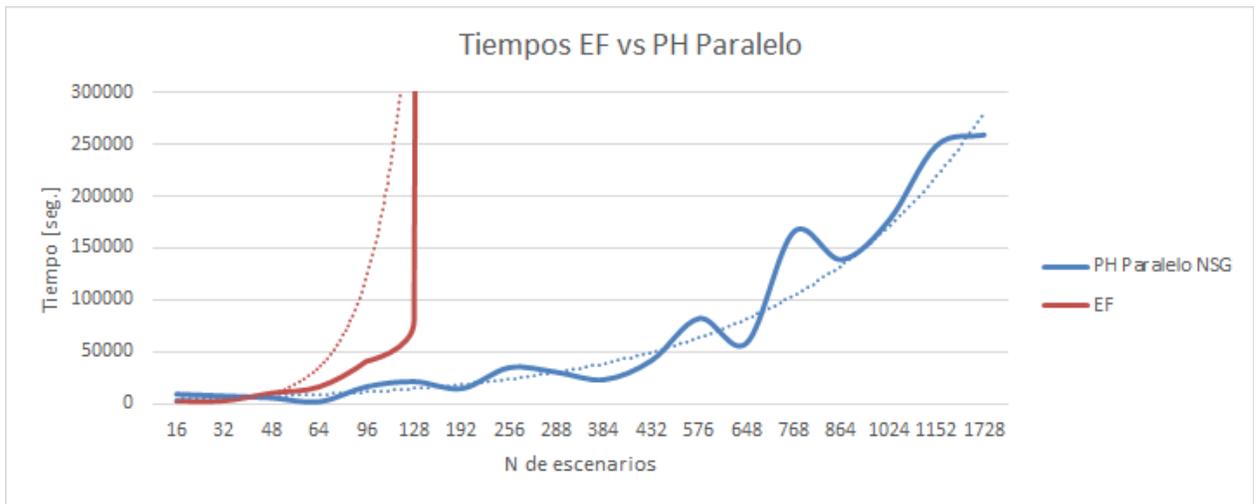


Ilustración 7.7: Tiempos de ejecución Generador de escenarios NSG - Formulación extendida (EF), Caso SIC-SING Multinodal

Además, se puede observar que PH paralelo NSG, presenta oscilaciones en los tiempos de ejecución, explicadas principalmente por la cantidad de iteraciones necesarias para resolver cada instancia. A diferencia del generador SYNTH, las variaciones que se muestran (para 648 y 768 escenarios), es mucho más pronunciada. Esto se puede explicar por la diferencia en la cantidad de iteraciones, para este caso, la instancia de 768 escenarios se resuelve luego de 11 iteraciones, en comparación con 648 escenarios que resuelve en 6.

Función Objetivo

En la ilustración 7.8 se puede observar el comportamiento de la función objetivo, según la cantidad de escenarios. Se puede ver que en general, representa una curva decreciente. Para la instancia de 32 escenarios, se ve un comportamiento inusual, con un valor óptimo muy bajo en comparación con la cantidad de escenarios adyacentes. Esto se explica por la característica de la instancia, que al contar con solo 32 escenarios (menos escenarios que la data histórica disponible), puede representar un problema poco estricto, alejado de la realidad, lo que genera un valor óptimo bajo. El mismo análisis se puede aplicar para la instancia de 96 escenarios. Posterior a esto la curva es decreciente.

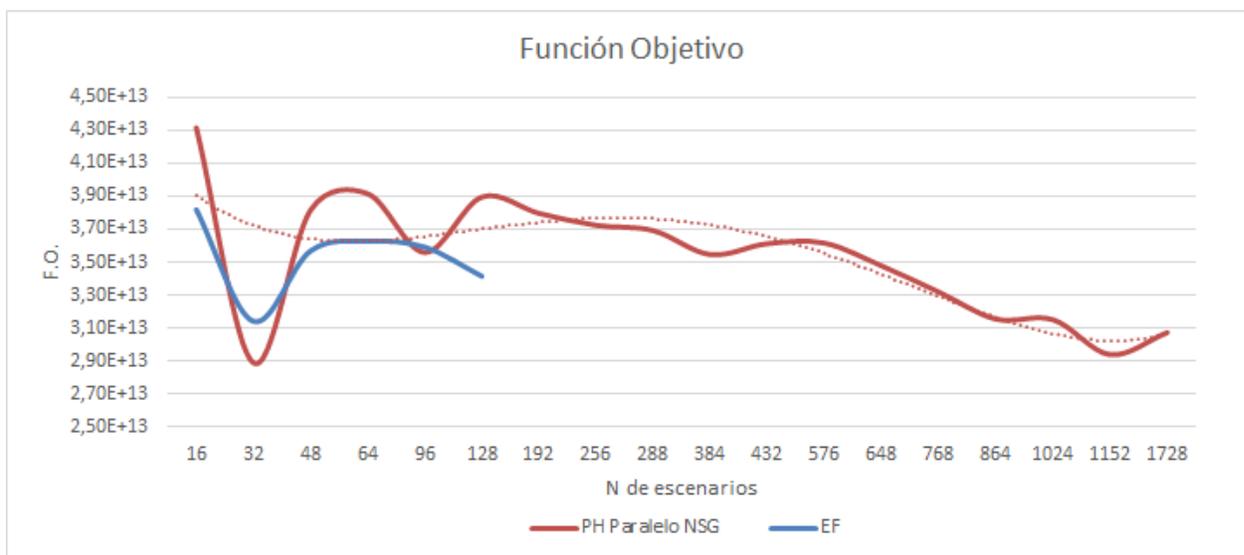


Ilustración 7.8: Tiempos de ejecución Generador de escenarios NSG - Formulación extendida (EF), Caso SIC-SING Multinodal

Hasta los 128 escenarios, se puede hacer una comparación entre la formulación extendida y PH paralelo, ya que para un mayor número de escenarios, se vuelve imposible resolver la formulación extendida con la capacidad de cómputo disponible. Desde el punto de vista teórico, la solución obtenida con la formulación extendida y PH, deberían ser la misma. Al analizar los resultados, se obtiene que las diferencias entre la solución de la formulación extendida y PH, se encuentran dentro del GAP de tolerancia del 1%.

La instancia que muestra el mejor resultado es la de 1152 escenarios, con tiempo de ejecución razonable: 248.846 segundos (2,89 días) y con la segunda mejor función objetivo: 2,94E+13 (la menor función objetivo se encontró con 32 escenarios).

7.5. Comparación de Árbol + Metodología

En esta sección se muestra uno de los resultados más importantes de esta tesis, se compara los Generadores de Escenarios + Metodologías.

Se trabaja con 3 enfoques de resolución:

- Generador árbol PLP, resuelto con SDDP.
- Generador árbol SYNTH, resuelto con PH.
- Generador árbol NSG, resuelto con PH.

Para cada uno de los enfoques se ejecuta el siguiente procedimiento:

1. Resolver el árbol de escenarios, con la metodología asociada. Se almacena la solución

de primera etapa y sus costos operacionales correspondientes.

2. A partir de la solución de primera etapa, resolver problemas deterministas, equivalentes a realizaciones históricas: Lluviosas, promedio y secas.
3. La función objetivo encontrada, sumada a los costos operacionales de primera etapa (parte 1), representan los costos operacionales totales, que es la principal métrica de comparación.

De acuerdo a los resultados expuestos en las secciones 7.4.1 y 7.4.2, se utilizará la configuración con el mejor desempeño para los generadores de árboles. Generador NSG trabaja con 1024 escenarios, mientras que SYNTH, se utiliza un árbol con 1152 escenarios.

Para mejorar la ejecución de PH se utilizan los siguientes ajustes:

- Random bundles del 5 %: Para SYNTH se generan 58 bundles aleatorios, mientras que para NSG 51.
- Warmstars.
- GAP ajustable, iniciando con tolerancia del 15 % en la primera iteración y 1 % desde la iteración 3 en adelante.
- Cálculo de ρ con: SEP para variables con costos operacionales asociados y FIX para variables sin costos.
- Se utilizan 120 procesadores (workers).

La instancia de comparación, es el sistema chileno SIC-SING, Multinodal, con horizonte de planificación de 103 meses (8.58 años), y 3 bloques por mes. Además, se utiliza una tasa de descuento anual del 10 %.

Los tiempos de ejecución para cada enfoque de resolución son los siguientes:

Tabla 7.6: Tiempos de ejecución árbol + Metodología

	NSG + PH	SYNTH + PH	PLP + SDDP
Tiempo (seg)	176.842	138.272	4.321

Como es de esperar, el tiempo de ejecución es mucho menor para PLP+SDDP: NSG + PH tarda 40 veces más que PLP+SDDP y SYNTH + PH tarda 31 veces más.

Al comparar los tiempos de resolución para el algoritmo PH, se observa que NSG tarda un 27 % más que SYNTH. Esto se explica solamente por la naturaleza del árbol generado, ya que ambos problemas tienen la misma configuración de parámetros.

Posterior a estas corridas, se almacena la solución de primera etapa: costos operacionales y niveles de embalses, sometiendo estos resultados a distintas realizaciones hídricas.

7.5.1. Hidrologías Húmedas

Para las hidrologías húmedas se seleccionan las 10 realizaciones históricas con mayor afluente promedio. En este caso se evalúan los costos en un año de operación, considerando 3 bloques por mes:

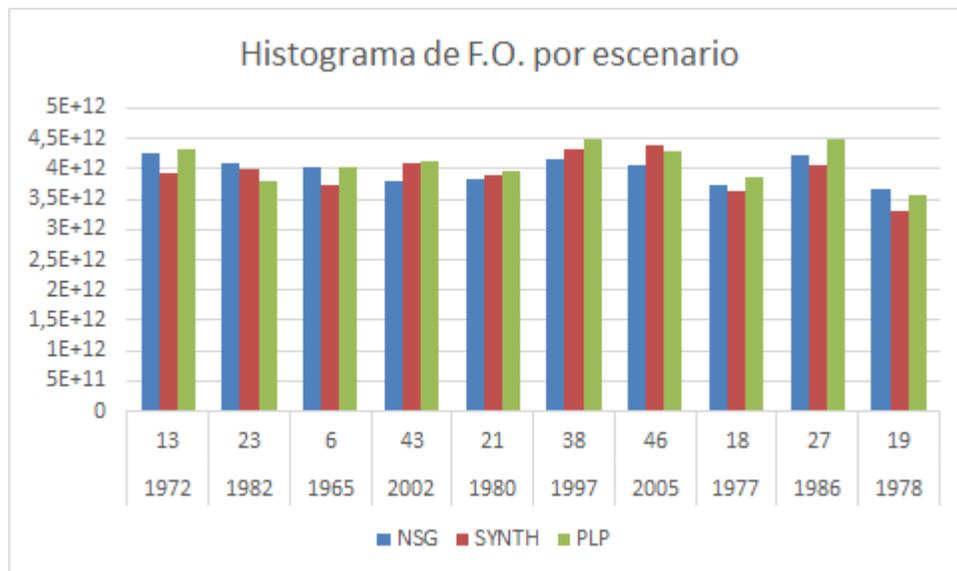


Ilustración 7.9: Histograma de Costos operacionales, escenarios húmedos

En la figura 7.9 se puede ver el histograma de costos operacionales, al someter las soluciones de primera etapa a realizaciones históricas húmedas.

Del gráfico se desprende que para escenarios húmedos, NSG obtiene en 4 ocasiones los costos operacionales más bajos, SYNTH en 5 ocasiones y PLP en 1 (año 1982). Por lo tanto, este primer resultado, sugiere que SYNTH + PLP tiene mejor desempeño en escenarios húmedos.

A continuación se expone el impacto en los costos operativos de cada solución:

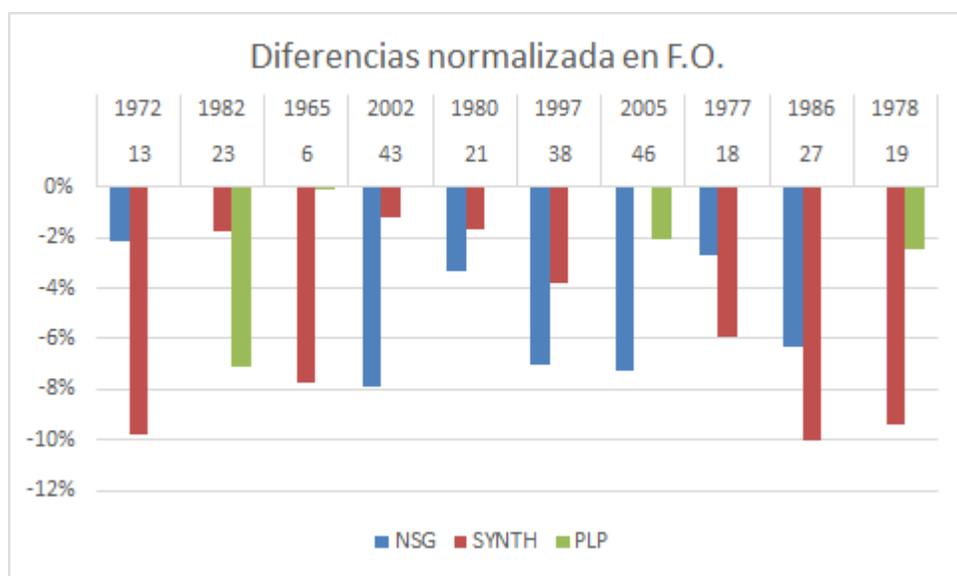


Ilustración 7.10: Variación porcentual de Costos operacionales, escenarios húmedos

La figura 7.10 muestra la diferencia entre funciones objetivos, normalizada. Tomando con valor 0 % el mayor costo operacional por escenario.

La siguiente tabla, muestra un resumen de los principales, resultados referentes a costos:

Tabla 7.7: Tabla resumen escenarios húmedos

	PH paralelo		SDDP
	NSG	SYNTH	PLP
Media	-3,7 %	-5,1 %	-1,2 %
Media condicionada	-5,2 %	-5,7 %	-2,9 %
N de Min. Costos	4	5	1
N de Max. Costos	3	1	6

Se observa que el mejor desempeño lo tiene SYNTH, ahorrando costos promedio de un 5,1 %, seguido por NSG con -3,7 % y por último PLP con un -1,2 %.

Al analizar los escenarios en los cuales las metodologías no representaban el máximo costo, es decir, calculando el porcentaje de mejora, dado que generaban ahorros (Media condicionada al ahorro), el mejor resultado nuevamente lo obtiene SYNTH con un 5,7 % de ahorro en costos.

Al medir la cantidad de veces que se obtuvo el mínimo en costos, SYNTH vuelve a tener el mejor desempeño, alcanzando el mínimo en 5 de 10 ocasiones. Por último, observando cuál fue la metodología que más veces tuvo el máximo en costos, se observa que PLP, tiene el peor desempeño, alcanzando el máximo en 6 ocasiones.

En anexos I se pueden ver las características de las instancias, sus funciones objetivos y mejoras.

7.5.2. Hidrologías Medias

A continuación, se exponen las mismas pruebas de la sección anterior, para escenarios con hidrologías medias. El histograma de funciones objetivo se muestra a continuación:

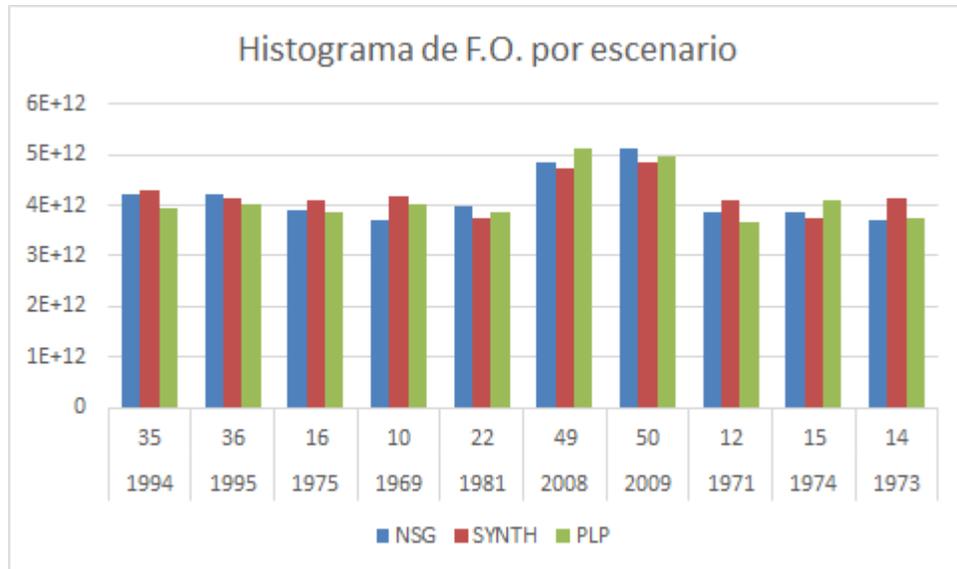


Ilustración 7.11: Histograma de Costos operacionales, escenarios promedio

En este caso, el mejor desempeño lo tiene PLP. Encuentra en la misma cantidad de ocasiones que SYNTH, los mínimos costos, pero además, presenta mayores ahorros. Esto se ve reflejado con mayor claridad en el gráfico 7.12:

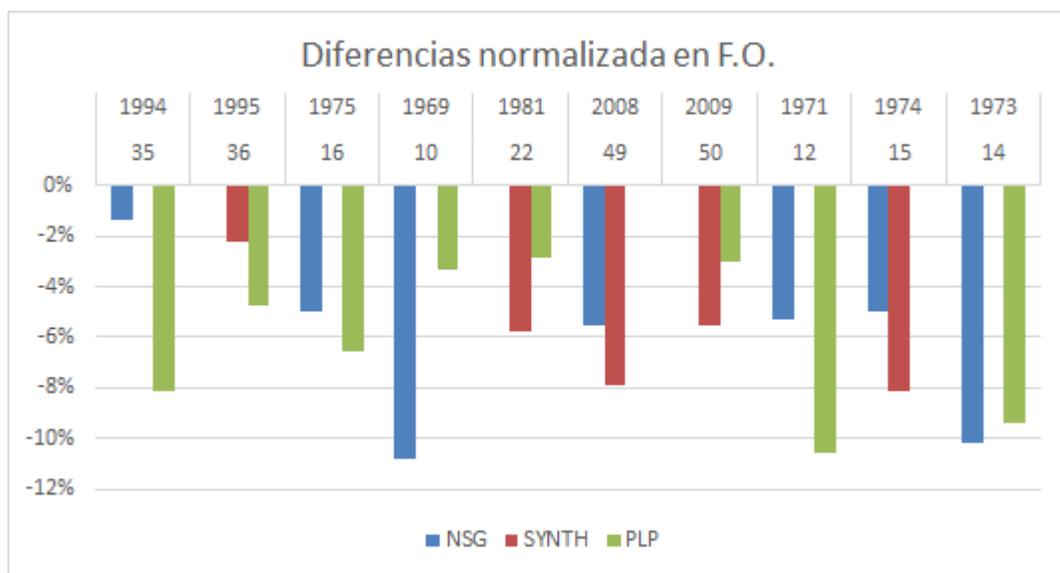


Ilustración 7.12: Variación porcentual de Costos operacionales, escenarios promedio

A modo de resumen, la siguiente tabla muestra los principales indicadores:

Tabla 7.8: Tabla resumen escenarios medios

	PH paralelo		SDDP
	NSG	SYNTH	PLP
Media	-4,3 %	-2,9 %	-4,9 %
Media condicionada	-6,2 %	-5,9 %	-6,1 %
N de Min. Costos	2	4	4
N de Max. Costos	3	5	2

De aquí se desprende que el mejor desempeño para escenarios medios, lo obtiene PLP, alcanzando en 4 veces el mínimo en costos operacionales, presentando un ahorro medio de 4,9 %, y un ahorro condicional (dado que presenta ahorro) de un 6,1 %.

Por otro lado SYNTH presenta resultados erráticos respecto a escenarios promedio, ya que representa el máximo de costos en 5 ocasiones, y el mínimo en 4. Mostrando el peor desempeño entre las metodologías, con un ahorro promedio de -2,9 % y ahorro condicionado del 5,9 %.

Por último, NSG muestra resultados más robustos, ya que alcanza el máximo y mínimo, en menos ocasiones (2 y 3, respectivamente). El ahorro medio es de 4,3 %, mientras que alcanza el máximo en ahorro condicionado: -6,2 %.

7.5.3. Hidrologías Secas

El último tipo de escenarios a trabajar son hidrologías secas. Para este caso se expone un análisis similar a las secciones anteriores.

El gráfico 7.13 muestra los costos operacionales, para cada metodología:

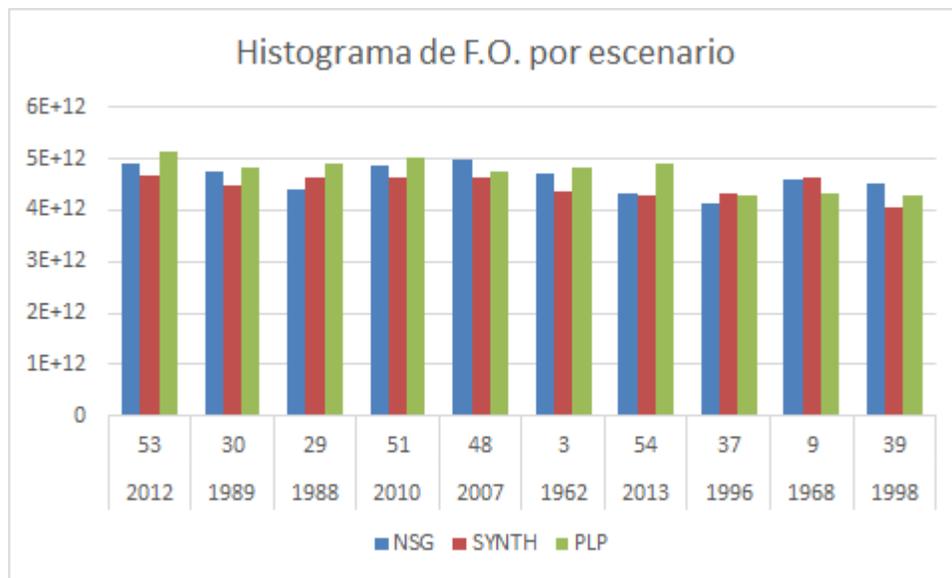


Ilustración 7.13: Histograma de Costos operacionales, escenarios secos

En principio, se puede observar que los mayores costos los obtiene en un mayor número de ocaciones PLP. En el gráfico 7.14, se aprecia con más detalle:

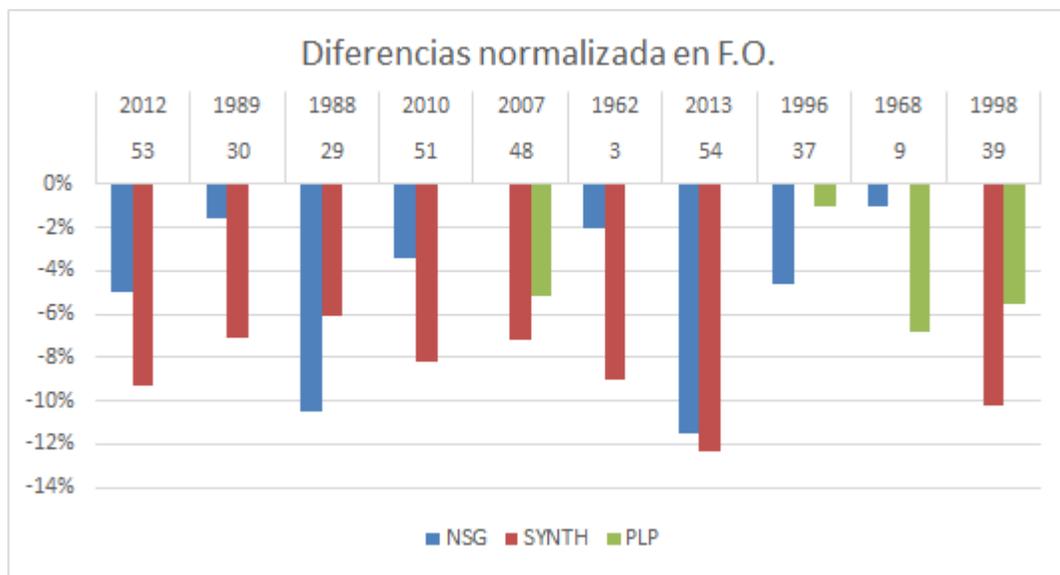


Ilustración 7.14: Variación porcentual de Costos operacionales, escenarios secos

La siguiente tabla, muestra el resumen de los resultados, en la instancias para hidrologías secas:

Se puede concluir que, para escenarios secos, NSG tiene un desempeño regular, alcanzando el mínimo y máximo en la misma cantidad de ocaciones, y mostrando ahorros promedios del 4% del total de los costos.

Tabla 7.9: Tabla resumen escenarios secos

	PH paralelo		SDDP
	NSG	SYNTH	PLP
Media	-4,0 %	-7,0 %	-1,8 %
Media condicionada	-5,0 %	-8,7 %	-4,6 %
N de Min. Costos	2	7	1
N de Max. Costos	2	2	6

Por otro lado, SYNTH muestra los mejores resultados, alcanzando el mínimo de los costos en 7 ocasiones, con un ahorro promedio, por sobre las demás metodologías, de 7% y un ahorro condicionado de 8,7%. Un punto negativo es que, muestra ser el máximo de costos en 2 ocasiones.

Para el caso de SDDP, muestra el peor desempeño, con ahorro promedio de 1,8% y ahorro condicionado de 4,6%. Alcanzando el mínimo de costos en una ocasión y el máximo en 6.

Capítulo 8

Conclusión

Las principales conclusiones referentes al PH se muestran a continuación:

- Las técnicas de aceleración de resolución del algoritmo, como: Warmstart, bundling, paralelización del término cuadrático proximal y GAP ajustable, pueden generar diferencias relevantes en el tiempo de resolución. Según las pruebas realizadas, se obtuvo mejoras de hasta un 20 % en el tiempo.

-Para calcular el término cuadrático proximal, sin duda son de mucha ayuda las técnicas actuales: SEP y CP, pero para variables que no tienen costos operacionales asociados, se vuelve casi obligatorio utilizar FIX. Esta última metodología, presenta el problema que la única forma de obtener el valor del penalizador ρ es 'ensayo y error', obligando a realizar variadas pruebas, para ir generando cotas y llegar al mejor ρ .

- Al comparar PH con y sin paralelizar, siempre obtienen el mismo resultado, considerando Función objetivo, GAP e iteraciones. La única diferencia al paralelizar es el tiempo de resolución, que dependiendo del tamaño de la instancia, puede ser determinante en la elección entre paralelizar o no. Además, si el problema es de gran tamaño, los recursos computacionales son insuficientes (con el desarrollo actual) para resolver el problema sin paralelizar.

Los resultados referentes a generadores de escenario se exponen a continuación:

-Desde el punto de vista teórico, el principal problema que presenta el generador de escenarios PLP, es considerar escenarios improbables, bajo realizaciones reales. Examinar todas las combinaciones de escenarios, para cada apertura del árbol, obliga a crear escenarios con hidrologías muy radicalizadas, cruzando años lluviosos con años de sequía. Lo anterior, genera un problema más estricto que el problema real, lo que solo puede encontrar soluciones subóptimas (peores o igual a la óptima).

-Por otro lado, la construcción de los generadores SYNTH y NSG, consideran tanto la correlación espacial, como la correlación temporal. Además, los generadores SYNTH y NSG, presentan la fortaleza de que la ramificación del árbol, se hace en base a probabilidades condicionales, sin cruzar hidrologías improbables en un escenario real.

-Desde el punto de vista práctico, los generadores NSG y SYNTH tienen mejores resultados en escenarios extremos. Al comparar estas metodologías, SYNTH obtuvo mejores resultados en un mayor número de ocasiones, disminuyendo los costos comparativos, hasta en un 12,3 % para hidrologías secas. En escenarios húmedos, SYNTH muestra disminución de costos de hasta un 10 %, lo que para la operación eléctrica anual representa ahorros significativos.

-Al comparar los generadores bajo escenarios promedio, PLP y SYNTH muestran resultados muy parecidos, presentando mayores ahorros el generador PLP.

-Por último, un aspecto importante al cambiar de un enfoque por períodos a uno por escenarios, es que se pueden modelar fácilmente escenarios de cambio climático (que no están presentes en la data histórica), y que serían de ayuda, dado el comportamiento actual del clima chileno.

Los resultados referentes a las metodologías de resolución son:

-Desde el punto de vista de exactitud, para la mayoría de las instancias comparadas, PH muestra menores GAP que SDDP, al comparar esta distancia porcentual, se vuelve casi insignificante, con diferencias del 0,001 %.

-Analizando los tiempos de ejecución, se observa una diferencia radical entre SDDP y PH-Paralelo, donde PH tarda hasta 23 veces más que SDDP. Lo anterior puede ser determinante al momento de decidir entre qué metodología de resolución se debe utilizar. Para ejemplificar, en una de las instancias de gran tamaño testeada, PH tarda 2,05 días, mientras que SDDP 1,2 horas.

Los principales aspectos que explican estas superlativas diferencias en tiempos de ejecución son los siguientes:

Al momento de aplicar mecanismos de aceleración de resolución para PH, se obtienen mejoras de un 70 % en los tiempos. En otras investigaciones (revisar [24] y [27]) se han alcanzado mejoras de hasta un 90 %, lo que sugiere que para mejorar el desempeño se puede profundizar en técnicas como: cálculo de ρ , bundling, clustering de escenarios, entre otros.

Además, el lenguaje de compilación que utiliza cada metodología presenta diferencias importantes. La velocidad de cómputo de Fortran (lenguaje de SDDP) es significativamente más rápida que la de Python (lenguaje utilizado por PH) ¹.

Por lo tanto, de acuerdo a la investigación realizada, se recomienda cambiar la metodología actual de generación de escenarios y trabajar con un generador alternativo (como SYNTH), y dadas las condiciones actuales, continuar utilizando la metodología de resolución SDDP.

¹Para mayor detalle de las pruebas realizadas, revisar Anexos J

Trabajo Futuro

En este trabajo de tesis, se hizo aproximaciones del problema real, para poder trabajar con un problema convexo. En la realidad, existen ciertas restricciones (como el convenio de riego) que obligan la incorporación de variables enteras. Por lo tanto, el siguiente paso en la comparación de SDDP y PH, es resolver en un problema no convexo, ya que en ese caso, ambas metodologías se comportarían como heurísticas, y no necesariamente obtendrían la solución óptima.

Otra propuesta de investigación, es resolver el problema eléctrico, generando árboles con SYNTH y NSG y utilizando la metodología SDDP. De esta manera, se podrían comparar los resultados con la forma de resolución actual, confrontando íntegramente los generadores de escenario.

Por último, la comparación entre árboles de escenario es un tema abierto y aún no se tiene claridad de cuál es el mejor modo de comparación. En esta tesis, se utiliza la solución de primera etapa, sometida a realizaciones históricas, pero posiblemente, se puedan generar otras formas de comparación, con distintas métricas a evaluar.

Una propuesta, es resolver el árbol de escenario iterativamente hasta cierta etapa n , es decir, partir resolviendo el árbol de escenario, almacenar la solución de primera etapa t , y luego, a partir de esta solución, volver a resolver el árbol, pero partiendo desde la etapa $t+1$, e ir avanzando hasta la etapa n , de tal forma de simular la aplicación real del modelo de coordinación hidrotermica.

Bibliografía

- [1] Comisión nacional de energía. Producción y consumo - producción real por sistema. <http://www.cne.cl/estadisticas/electricidad/>. [En línea], [consulta: 1 Septiembre 2015].
- [2] Comisión nacional de energía. Infraestructura - capacidad instalada de generación. <http://www.cne.cl/estadisticas/electricidad/>. [En línea], [consulta: 1 Septiembre 2015].
- [3] Comisión nacional de energía. Infraestructura - capacidad instalada de transmisión. <http://www.cne.cl/estadisticas/electricidad/>. [En línea], [consulta: 1 Septiembre 2015].
- [4] Torres R. Modelo de coordinación hidrotérmica multinodal y multiembalses para sistemas eléctricos longitudinales en el mediano plazo. Master's thesis, Universidad de Chile, Departamento de Ingeniería Eléctrica, Santiago, Chile, 2006. (Magister en ciencias de la ingeniería, Mención: Ingeniería Eléctrica).
- [5] Pereira-Bonvalleta E. Püschel-Løvengreena S. Matusa M. Moreno R. Optimizing hydrothermal scheduling with non-convex irrigation constraints: Case on the chilean electricity system. *5th International Workshop on Hydro Scheduling in Competitive Electricity Markets. Science Direct.*, 2016. Disponible: <http://www.sciencedirect.com/science/journal/22126716>.
- [6] Juan Carlos Olmedo. Modelo gol (gestión Óptima del laja). <http://web.ing.puc.cl/power/education/GOL%20JC01medo.pdf>. [En línea], [consulta: 1 febrero 2015].
- [7] Minder E. Modelación hidrológica en el sistema interconectado central. Master's thesis, Pontificia Universidad Católica de Chile, Departamento de Ingeniería Eléctrica, Santiago, Chile, 1997. (Tesis Magíster en ciencias de la ingeniería).
- [8] L.M.V.G. Pinto M.V.F. Pereira. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming, North- Holland*, 52(21):359–375, 1991.
- [9] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4, 3:238–252, 1962.
- [10] F. Louveaux J.R. Birge. Introduction to stochastic programmings. *Numerische Mathe-*

matik 4, (2), 1997.

- [11] García J.P. Escudero L.F. Lario F.C., Rodríguez A. Análisis y definición de escenarios en programación estocástica para la gestión de la cadena de suministros en el sector del automóvil. *IV Congreso de Ingeniería de Organización. Sevilla.*, 2001.
- [12] Roger J-B. Wets R.T. Rockafellar. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research.*, 1991.
- [13] Woodruff D. Watson J.P. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problem. *Comput Manag Sci*, 8:355–370, 2011.
- [14] Palacios E. Modelo para manejo de la incertidumbre hidrológica en la planificación de la operación del sic. 2004. (Memoria Ingeniero civil electricista).
- [15] Arredondo M. Tratamiento estadístico de los caudales afluentes en los modelos de planificación del cdec-sic. 2010. (Memoria Ingeniería civil electricista).
- [16] Iroume A. Progressive hedging aplicado a la coordinación hidrotermica. 2012. (Tesis Magíster en Gestión de operaciones).
- [17] Waintraub A. Rios I. Inflow water scenarios for hydro power management. *Technical report*, 2014. No publicado - Universidad de Chile.
- [18] Zucchini W. Applied smoothing techniques part 1: Kernel density estimation. 2003.
- [19] R. J-B Wets. J. O. Royset. Epi splines and exponential epi-splines: Pliable approximation tools. *Technical report*, 2013.
- [20] R. J-B Wets. J. O. Royset. Fusion of soft and hard information in nonparametric density estimation. *Technical report*, 2013.
- [21] Vladimirov H. Mulvey J. Applying the progressive hedging algorithm to stochastic generalized networks. *Ann. Oper. Res.*, pages 399–424, 1991.
- [22] Sandia National Labs, software Package Pyomo.
- [23] Watson J.P. Woodruff D. Hart W., Laird C. Pyomo– optimization modeling in python. *Springer Science + Business Media*, 67, 2012.
- [24] Cristobal Pais. Complete stochastic forestry planning problem using progressive hedging algorithm, 2014. (Tesis Magíster en Gestión de operaciones).
- [25] Woodruff D.L Watson J.P. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computer Management Science*, 2010.
- [26] Badilla F. Problema de planificación forestal estocástico resuelto a través del algoritmo progressive hedging. 2010. (Tesis Magíster en Gestión de operaciones).
- [27] Inc. Gurobi Optimization. Gurobi optimizer reference manual. 2015. Disponible:

<https://gurobi.com>.

- [28] Julia Lang ORG. Tiempos de las pruebas comparativas en relación a c. <http://www.http://julialang.org/>. [En línea], [consulta: 18 Octubre 2016].

Anexos

Anexo A: Pseudo-código Benders Multi etapa

El pseudo-código del algoritmo que se muestra a continuación se aplicará al siguiente problema estocástico multietapa. Las ilustraciones mostradas fueron extraídas de un trabajo de Pereira, 1991 [8]:

$$\begin{array}{ll} \text{Min} & c_1 x_1 + p_1 c_2 x_{21} + p_2 c_2 x_{22} + \dots + p_m c_2 x_{2m} \\ \text{subject to} & A_1 x_1 \geq b_1, \\ & E_1 x_1 + A_2 x_{21} \geq b_{21}, \\ & E_1 x_1 \quad + A_2 x_{22} \geq b_{22}, \dots \\ & E_1 x_1 \quad \quad \quad + A_2 x_{2m} \geq b_{2m}. \end{array}$$

Donde:

- x_{tm} : variable de decisión en etapa i , escenario m

- c_t : costo asociado a la variable x_{tm} en período t .

- A_t , b_{tm} y E_t : son parámetros asociados a las etapas t y escenarios m

- $\alpha_t(x_i)$: es la función de costos futuros, función que refleja todos los costos asociados a los periodos siguientes a la etapa t .

1: Inicialización:

- Definir un set de valores para las variables trascendentes²: \hat{x} for $i = 1, \dots, n$ and $t = 1, \dots, T$.
- $LB = -\infty$ (Cota inferior).
- $UB = \infty$ (Cota superior).
- Definir criterio de parada ε .

²Las variables trascendentes son las variables que tienen implicancia en los períodos siguientes, es decir, las variables que entrega el problema maestro al esclavo.

- $n = 0$ (Número de vértices de la variable dual asociada), $\hat{\alpha}_t(x_t) = 0$ (Función de costos futuros.).
- 2: Resolver el problema de primera etapa (sin considerar las restricciones de etapas siguientes):

$$\begin{aligned} z &= \text{Min} && c_1 x_1 + \hat{\alpha}_1(x_1) \\ &\text{subject to} && A_1 x_1 \geq b_1. \end{aligned}$$

- 3: Almacenar solución de primera etapa \hat{x}_t
- 4: **if** $UB - LB < \varepsilon$: **then**
- 5: Stop.
- 6: **else** Continuar al paso siguiente.
- 7: **end if**
- 8: **Background recursion**
- 9: **for** $t = \{T, T - 1, \dots, 3, 2\}$ **do**
- 10: **for** con $j = \{1, \dots, m - 1, m\}$ **do**
- 11: Resolver el problema de optimización para cada t (periodo), \hat{x}_t, j (escenario):

$$\begin{aligned} \text{Min} &&& c_t x_t + \bar{\alpha}_t(x_t) \\ &\text{subject to} && A_t x_t \geq b_{tj} - E_{t-1} \hat{x}_{t-1j}. \end{aligned}$$

- 12: Sea $\pi_{t-1,i,j}$ el vértice que representa el valor de las variables duales asociadas al problema antes solucionado. Almacenar $\pi_{t-1,i,j}$.
- 13: **end for**
- 14: **end for**
- 15: Calcular el valor esperado del vértice $\pi_{t-1,i,j} = \sum_{j=1}^m p_j \pi_{t-1,i,j}$ y almacenar.
- 16: Construir un hiperplano para los costos futuros aproximados de la etapa $t - 1, \hat{\alpha}_t(x_t)$ (en otras palabras, agregar cortes de Benders)

Una parte importante del algoritmo es cómo determinar el set de valores para las variables trascendentes. Por lo que se propone realizar una recursión hacia adelante (forward recursion) para cada una de las combinaciones de escenario. El pseudo código de dicha parte del algoritmo se muestra a continuación:

- 1: Resolver el problema de primera etapa (sin considerar las restricciones de etapas siguientes):

$$\begin{aligned} z &= \text{Min} && c_1 x_1 + \hat{\alpha}_1(x_1) \\ &\text{subject to} && A_1 x_1 \geq b_1. \end{aligned}$$

- 2: Almacenar solución de primera etapa $\hat{x}_t = \hat{x}_{ti}$ para $i = 1, 2, \dots, n$
- 3: **Foward recursion**
- 4: **for** $t = \{2, 3, \dots, T\}$ (periodos) **do**
- 5: **for** $i = \{2, 3, \dots, n\}$ (escenarios) **do**

6: Resolver el problema de optimización para cada t, j (escenario):

$$\begin{array}{ll} \mathbf{Min} & c_t x_t + \bar{a}_t(x_t) \\ \mathbf{subject\ to} & A_t x_t \geq b_{jt} - E_{t-1} \hat{x}_{t-1}. \end{array}$$

7: Almacenar la solución \widehat{x}_{ti}

8: **end for**

9: **end for**

Anexo B: Tipo de centrales de generación eléctrica

Las centrales de generación eléctrica se clasifican como:

- **Hidroeléctricas:** Corresponde a aquellas centrales que son movidas por agua. Pueden transformar la energía cinética de un cauce de agua (utilizando su caudal) y/o la energía potencial presente en un desnivel.
 - **Centrales de Pasada:** Aquellas que no tienen capacidad de almacenar agua y deben generar de acuerdo a la demanda y disponibilidad que tienen en un momento dado.
 - **Centrales de Embalse:** Las que si pueden almacenar agua con el objeto de generar electricidad en el futuro. De acuerdo a su capacidad se puede tener una regulación de va desde semanas hasta un año.
 - **Centrales Mixtas:** Se puede almacenar agua en pequeñas cantidades (normalmente durante el día) para poder generar en las horas de mayor demanda.
 - **Centrales de Bombeo:** Presentan un desnivel y un estanque en el nivel inferior. Durante las horas de baja demanda se bombea el agua hasta el nivel superior para ser usada durante las horas de demanda alta.
 - **Centrales Mareomotrices y Marinas:** Aquellas que utilizan los flujos de agua de mar o la presión de las olas para generar electricidad.
- **Centrales a Vapor (Térmicas):** Cuentan con una turbina que genera electricidad al ser movida por vapor de agua a alta temperatura y presión. El vapor es generado en calderas que pueden utilizar diferentes combustibles como carbón, petróleo, gas natural u otros. Debido a que necesitan calentar el vapor, estas centrales cuentan con tiempos de partida y apagado largos.
- **Centrales Geotérmicas:** Utilizan la energía térmica presente en el magma terrestre para mover una turbina.
- **Centrales Nucleares:** Su operación es similar a las centrales térmicas pero el combustible corresponde a material radioactivo y funcionan realizan fusión nuclear dentro de los reactores.
- **Turbinas a gas:** Cuentan con una turbina que es activada directamente por los gases de combustión en lugar de ser movida por vapor de agua. Presentan tiempos de encendido menores que las centrales térmicas tradicionales.

- **Centrales de ciclo combinado:** Utilizan una caldera que aprovecha los gases de escape de turbinas a gas. Esta caldera funciona luego como una central a vapor.
- **Motores Diesel:** Se utilizan en sistemas pequeños. Necesitan además del combustible, aire y agua para su refrigeración.
- **Centrales Eólicas:** Utilizan la energía cinética del viento para mover una turbina.
- **Centrales de Cogeneración:** Corresponde a las centrales que genera tanto energía eléctrica como térmica durante su proceso. Esto permite aumentar la eficiencia y se usan tradicionalmente en procesos industriales (como plantas de celulosa).
- **Centrales Solares:** Utilizan la energía solar para generar electricidad. Puede ser a través de una conversión directa en el caso de los paneles fotovoltaicos o a través de una turbina movida por vapor calentado por espejos que concentran la luz del sol.

Anexo C: Modelo hidráulico - Cuenca del Laja

A continuación se muestra la notación utilizada para modelos hidráulicos en el Centro de Energía de la Universidad de Chile, en particular, se expone la cuenca del Laja.

La Ilustración A.1 expone el modelo simplificado, mientras que en A.2 se pueden ver el mismo modelo pero en detalle y más complejo que muestra confluencia de caudales y embalses intermedios que fueron simplificados en A.1.

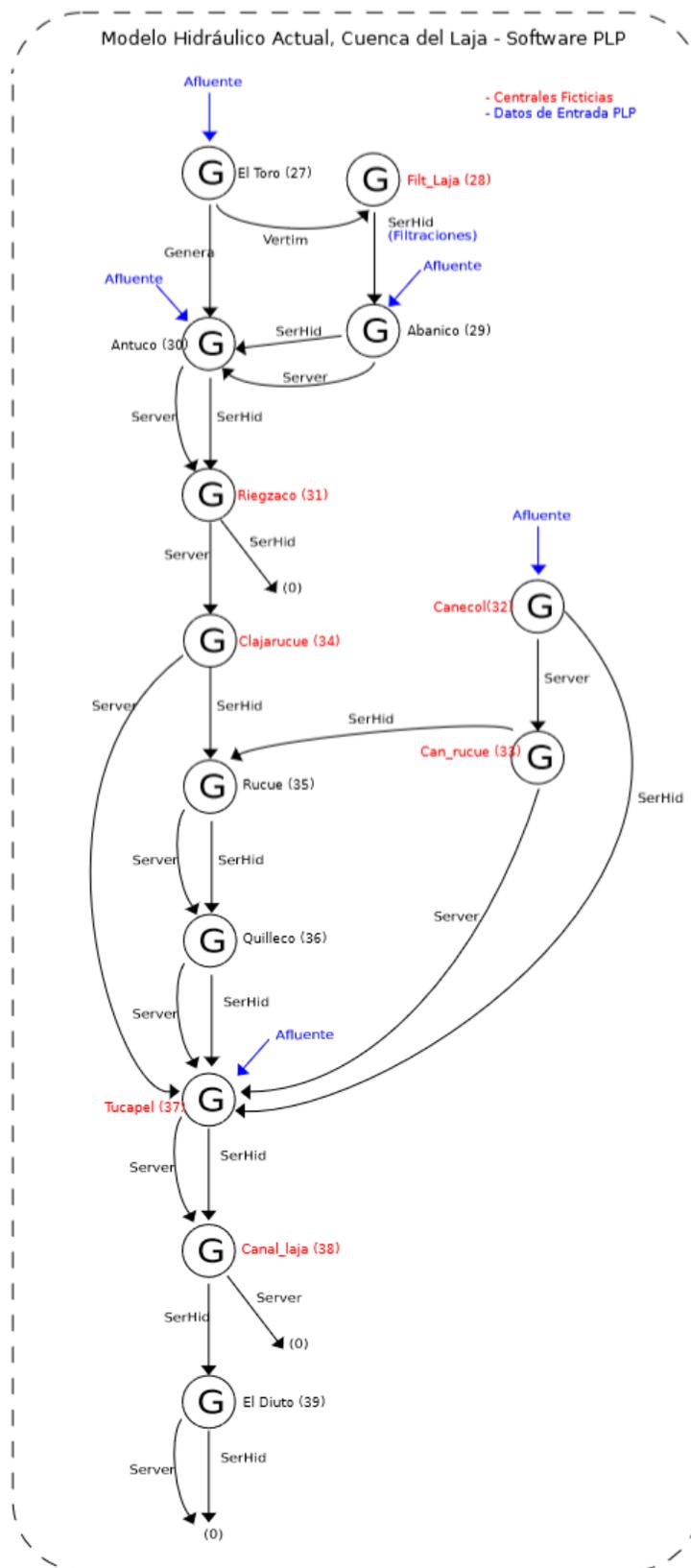


Ilustración A.1: Representación de Red hidráulica - Cuenca del Laja.

Modelo Hidráulico Completo, Concepto de Confluencias

- Afluentes minimizados en el PLP

Convenciones

- q_{capt} = Caudal de captación, laguna o río.
- q_{vert} = Caudal vertido, embalse o laguna.
- q_{alm} = Caudal Almacenado en un embalse o laguna.
- q_{ab} = Caudal aguas abajo después de una captación.
- q_{nt} = Caudal no turbinado, embalses, por vaciado.
- q_{cons} = Caudal de uso consuntivo, Regadíos.
- q_{filtr} = Caudal Filtraciones, embalses.
- q_{turb} = Caudal Turbinado, embalses y Centrales Serie.

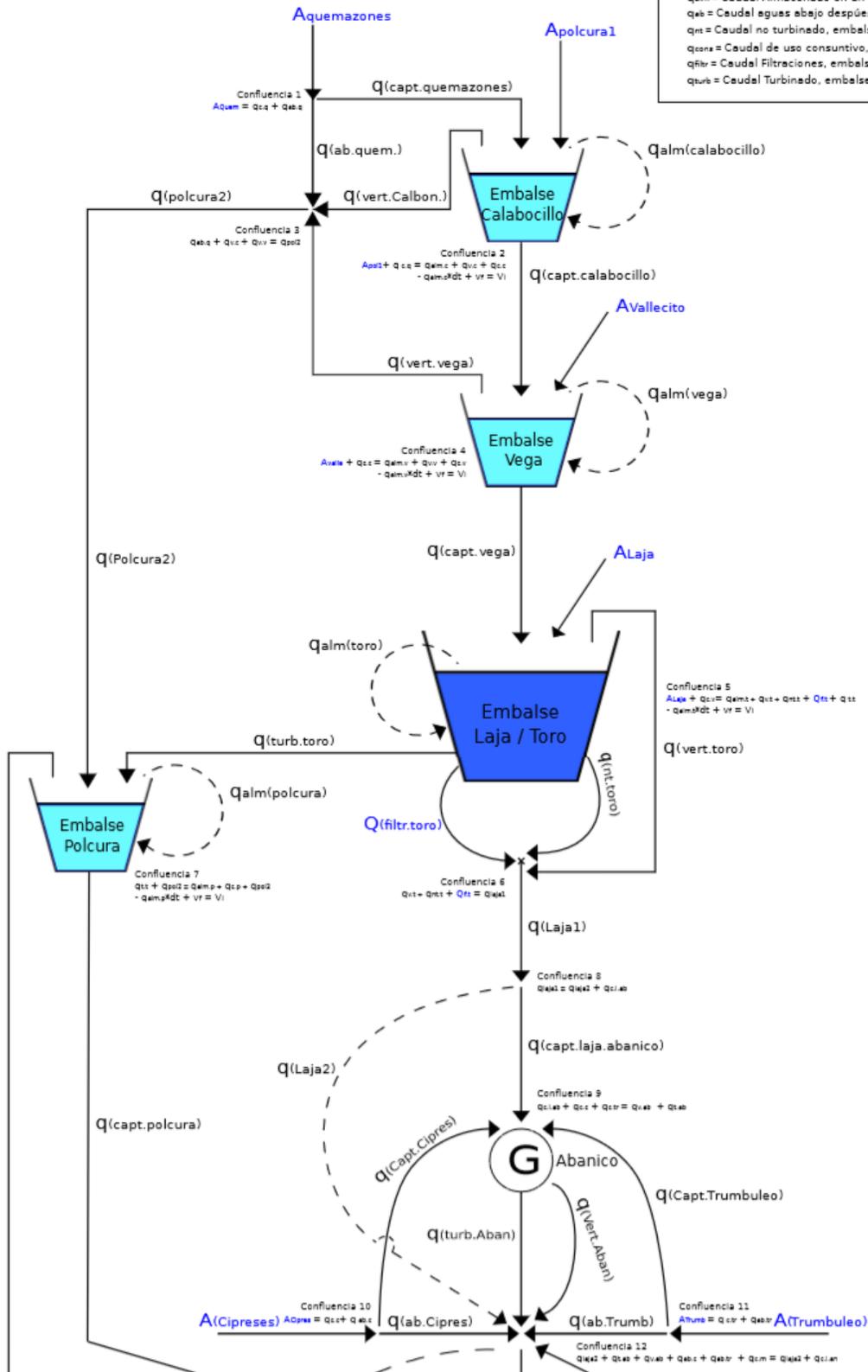


Ilustración A.2: Modelo hidráulico completo - Cuenca del Laja

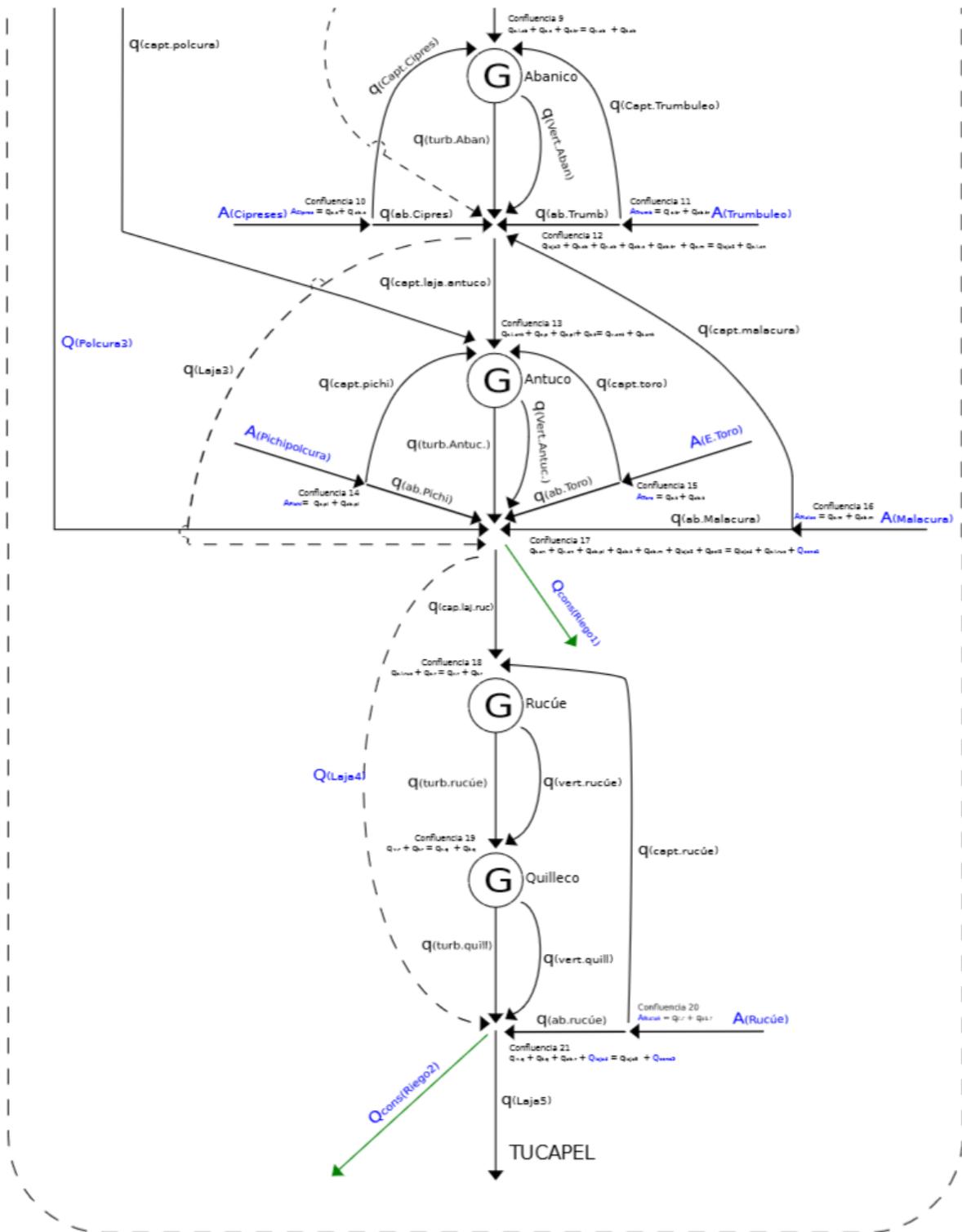


Ilustración A.3: Modelo hidráulico completo - Cuenca del Laja(continuación).

Anexo D: Resultados de Metodologías de aceleración de resolución de cada subproblema

A continuación se exponen los resultados para una instancia menor, que representa la Cuenca del Laja.

Tabla A.1: Metodologías de aceleración para cada subproblema

Combinación de técnicas	Iteraciones	Tiempo [seg.]	F.O.	Mejora de tiempo
Ninguna	233	726.5	3090936.6	0 %
Solo warmstart	190	597.1	3090807.1	18 %
Solo gap ajustable	201	621	3090983.6	15 %
Warmstart y gap ajustable	173	530.3	3090940.1	27 %

Como se puede ver, la mejor solución, considerando tiempos, iteraciones y función objetivo, se encuentra utilizando la estrategia mixta de resolución de subproblemas, es decir, warmstart y gap ajustable.

Se espera una disminución considerable de tiempo para instancias de mayor tamaño.

Cabe notar que para realizar estas pruebas se consideró el resultado obtenido en anexos E, es decir, se utiliza una linealización de 5 tramos para el término cuadrático proximal.

Anexo E: Resultados de la linealización del término cuadrático proximal

Para encontrar la cantidad óptima de tramos en que se puede linealizar el término cuadrático proximal, se utilizó una instancia pequeña, de la Cuenca del Laja.

Como se expuso en la sección 5.2.3, utilizar solo dos tramos, forman una curva en forma de V (o cono invertido) que fuerza de sobremanera la convergencia de las variables. Debido a lo anterior es que los resultados para 2 tramos se encuentra un tiempo de resolución bajísimo, pero con una función objetivo muy lejos del óptimo.

En la siguiente tabla se muestran los resultados:

Tabla A.2: Resultados de tramos para ρ

Tramos	Tiempo de resolución [seg.]	F.O.	Gap Acumulado	Min Gap	Max Gap
2	170.5	5330114	2664.1	31.3	105
3	355.6	3132610	427.9	5	38.41
4	318.4	3129427	384.3	4.5	32.3
5	726.5	3090936.6	380.05	4.47	32.3
6	1575.2	3090550	401.1	5	34.2

Se puede concluir que linealizar el término ρ en 5 tramos es adecuado para esta instancia. Si bien la función objetivo se ve un poco perjudicada en comparación con 6 tramos, se encuentran tiempos de resolución considerablemente menores, que en instancias de mayor tamaño (más períodos y/o centrales y/o bloques) se traduciran en un ahorro significativo de tiempo.

Es más, la opción de 4 tramos no se descarta para instancias aun más complejas.

Anexo F: Estrategias para cálculo de ρ

En esta sección se describen cuatro metodologías de cálculos del parámetro ρ y se entrega la intuición detrás de cada una.

ρ : **FIX**

El modo original de utilizar ρ de acuerdo a la teoría de PH consiste en tomar un número mayor que cero y asignarlo a todas las variables que necesitan cumplir el principio de no-anticipatividad. En este caso, se selecciona un valor $\rho = cte$, que oscila entre 10^{-5} y 10^4 en base a resultados experimentales, obteniéndose rendimientos dependientes del tamaño de la instancia y el tipo de problema de estudio.

ρ : **CP-OFP**

Una interpretación del parámetro ρ es el costo "por desviarse del promedio de las variables en un nodo en común, por lo que se usa un razonamiento similar a la estimación del costo marginal por "desviarse", a esto se le agrega un ponderador k ($k \in [0, 1]$). Así, surge a partir de un enfoque de minimización de costos, donde se estima el costo que se genera por cada subproblema en la medida que este se aleja de las condiciones de no anticipatividad.

Luego el factor *rho* es distinto para cada variable y se calcula según:

$$\rho(i) = k * C_i \tag{A.1}$$

Donde C_i representa el costo de la variable i . Mientras que el factor k es del orden de 10^{-1} ó 10^{-2} . Este razonamiento de costos proporcional (CP) fue extendido para problemas de maximización, donde no está claro cuál es el costo marginal por producto, luego fue denominado "Proporcional a la Función Objetivo"(OFP en inglés). La intuición detrás de esta modificación de ρ es que al existir variables con costos de magnitudes muy distintos entre sí, el costo por desviarse también debe ser - en teoría- proporcionalmente distinto.

Por lo tanto, se debe identificar los costos proporcionales para cada una de las variables modeladas dentro del problema.

ρ : **SEP**

La presente estrategia fue implementada en el problema de coordinación hidrotérmica, junto a una estrategia ρ FIX para variables sin costos asociados:

La sigla SEP significa que el parámetro ρ se selecciona para cada variable, para cada

escenario (Selecting per-element en inglés). Esta estrategia surge de la idea de lograr que el penalizador ω llegue al valor correcto en una sola iteración. Esto se podría alcanzar si para cada escenario se cumple la expresión:

$$\rho = \frac{C_i}{x_s - \bar{x}} \quad (\text{A.2})$$

Ahora aproximandose de una forma mas conservadora, es decir, desde un solo lado (para que las variables enteras no salten de un lado al otro del promedio, formándose ciclos y afectando a otras viariables), se toma una cota en el denominador que permite eliminar la dependencia a cada uno de los escenarios. Así, se utiliza el mínimo y máximo valor de las variables pertenecientes a un mismo nodo, obteniéndose la expresión:

$$\rho(i) = \frac{c_i}{\text{máx}_{s \in N_t} x_s - \text{mín}_{s \in N_t} x_s + 1} \quad (\text{A.3})$$

Para el caso de variables enteras. En el caso de variables continuas, es posible utilizar un denominador más agresivo, obteniéndose:

$$\rho(i) = \frac{c_i}{\text{max}(\sum_{s \in N_t} p_s |x_s - \bar{x}|, 1)} \quad (\text{A.4})$$

ρ : Taylor

La siguiente metodología es un aporte de Fernando Badilla, ex-alumno del MGO y presente la ventaja de poder usarse dinámicamente, es decir, actualiza su valor iteración a iteración:

Intuición Como se busca minimizar la diferencias entre las variables y su promedio en el nodo, lo que se busca es:

$$\min |x - \bar{x}| = \sqrt{(x - \bar{x})(x - \bar{x})}$$

Luego, usando la Serie de Taylor centrada en el promedio:

$$\text{Taylor}(|x - \bar{x}|, \bar{x}) = f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + \frac{1}{2}(x - \bar{x})f''(\bar{x})(x - \bar{x}) \quad (\text{A.5})$$

Removiendo los términos constantes, se puede ver un gran parecido al término de penalización:

$$f'(\bar{x})x + \frac{1}{2}(x - \bar{x})f''(\bar{x})(x - \bar{x}) \quad (\text{A.6})$$

$$\omega x + \frac{\rho}{2}(x - \bar{x})^2 \quad (\text{A.7})$$

Luego, haciendo el cálculo de las derivadas y tomando el límite se obtiene³.

$$\rho_i = \frac{1}{\sum_{k \neq i} (x_k - \bar{x}_{n(k)})^2} \quad (\text{A.8})$$

Un segundo enfoque de esta metodología es sin tomar límites sino despreciando los términos cruzados de diferencias de segundo orden, y en este caso se obtiene:

$$\rho_i = \frac{\sum_{k \neq i} \varepsilon_k^2}{3 \left(\sum_k \varepsilon_k^2 \right)^{\frac{1}{2}}} \quad (\text{A.9})$$

³En esta parte se hacen una serie de cálculos, para mas detalles revisar [26]

Anexo G: Archivos para paralelización en NLHPC

Para paralelizar el problema se utilizarón 3 archivos principalente (además del modelo y la instancia).

1) Archivo BATCH

A continuación se muestra la línea de código del archivo de comunicación con el servidor NLHPC, para este ejemplo se solicitó un total de 3 nodos (60 procesadores), con 20 workers por nodos:

```
#!/bin/bash
#SBATCH --job-name=example
#SBATCH --partition=slims
#SBATCH -N 2
#SBATCH --ntasks-per-node=20
#SBATCH --output=example_%j.out
#SBATCH --error=example_%j.err
#SBATCH --mail-user=carlos2pp@gmail.com
#SBATCH --mail-type=ALL
module load python/2.7.10 cplex/12.6.1

python clear.py
echo $SLURM_JOB_NODELIST >> Nodes.txt
python MultiNode.py
sed -i '$ d' Nodes.txt
cat Nodes.txt
source Nodes.txt
#Ask for the number of nodes requested:
echo "Executing pyomo_ns and dispatch_srvr"
pyomo_ns > ns.out &
export PYRO_HOST=$Node1
dispatch_srvr &> dispatch_srvr.out &

#Se ejecutan los solvers en los dos nodos solicitados
#es importante utilizar srun para lanzarlos, ya que de otra forma
#se ejecutan solo de forma local.

for i in {1..15}; do
  echo "Executing phsolverserver (${i})"
  srun -w $Node1,$Node2 phsolverserver &
done

echo "Executing runph"
runph --instance-directory=data --solver=cplex --solver-manager=phpyro --phpyro-required-workers=30
--linearize-nonbinary-penalty-terms=4 --handshake-with-phpyro --default-rho=1.0 --max-iterations=30
--output-times --scenario-mipgap=0.05 --output-solver-log --enable-ww-extensions --ef-mipgap=0.05
--verbose --traceback --scenario-solver-options="Threads=1"
--solution-writer=pyomo.pysp.plugins.jsonsolutionwriter --ef-mipgap=0.05 --shutdown-pyro
```

2) Archivo clear.py

clear.py simplemente borra el contenido del archivo "Nodes.txt", para asegurar que no quede información de nodos de las corridas anteriores:

```
archivo = open("Nodes.txt", "w")
```

Archivo MultiNode.py

Este archivo lee las variables de entorno de NLHPC, las procesa y posteriormente las almacena en "Nodes.txt". Es claro que este archivo se puede generar de diversas formas:

```
with open("Nodes.txt", 'r') as result:
    a=result.readline()
b=""
c=""
c=a.split("[")
b = a.replace("cn[", "")
b = b.replace("cnf[", "")
b = b.replace("]", "")
b = b.split('-')
if ", " not in a:
    try:
        dist=int(b[1])-int(b[0])
        for i in range(dist-1):
            if 9>=(int(b[0])+i+1):
                b.append("00"+str(int(b[0])+i+1))
            elif 99>=(int(b[0])+i+1)>=9:
                b.append("0"+str(int(b[0])+i+1))
            elif (int(b[0])+i+1)>=100:
                b.append(str(int(b[0])+i+1))
        with open("Nodes.txt", 'w') as result:
            for i in b:
                result.write("Node"+str(b.index(i)+1)+"="+c[0]+i+"\n")
    except:
        print "Just one node found, please check if Node "+c[0]+b[0] + " is correct"
        with open("Nodes.txt", 'w') as result:
            result.write("Node1="+c[0]+b[0]+"\n")
else:
    b=a.split(',')
    with open("Nodes.txt", 'w') as result:
        for i in b:
            try:
                aux=int(i)
                if "cn" in b[0]:
```

Anexo H: Convergencia de PH

A continuación se muestran los resultados de la convergencia del algoritmo PH para la instancia de 768 escenarios, con 38 Random Bundles, GAP ajustable (15% hasta 0,05%), con la metodología mixta: SEP + FIX:

Tabla A.3: Convergencia de PH

Iteración	Dist. Normalizada	F.O.	% Varición F.O.	% Var. Dist. Normalizada
0	12,2 %	1,729E+13		
1	7,54 %	1,944E+13	12,4 %	-38,2 %
2	5,30 %	2,217E+13	14,1 %	-29,7 %
3	2,51 %	2,748E+13	23,9 %	-52,6 %
4	1,69 %	3,013E+13	9,6 %	-32,7 %
5	1,47 %	3,092E+13	2,6 %	-13,0 %
6	1,35 %	3,113E+13	0,7 %	-8,2 %
7	1,20 %	3,199E+13	2,8 %	-11,1 %
8	1,23 %	3,214E+13	0,5 %	2,5 %
9				

Anexo I: Años hidrológicos seleccionados para comparación de metodologías

La siguiente tabla muestra los años hidrológicos seleccionados para representar escenarios, y la correspondiente función objetivo obtenida al resolver con cada una de las metodologías:

Desde el año 1972 hasta el 1978, representan años húmedos. Desde 1994 hasta 1973 años promedio, y desde 2012 hasta 1998, representan años secos.

En la tabla A.5 se pueden observar el ahorro en costos, para cada metodología por escenario.

Tabla A.4: Años hidrológicos seleccionados

Caracterizado	Año	sim. PLP	NSG	SYNTH	PLP
Húmedo	1972	13	4,2466E+12	3,9129E+12	4,3389E+12
Húmedo	1982	23	4,0793E+12	4,007E+12	3,789E+12
Húmedo	1965	6	4,0358E+12	3,7233E+12	4,0323E+12
Húmedo	2002	43	3,8051E+12	4,0809E+12	4,1299E+12
Húmedo	1980	21	3,823E+12	3,8892E+12	3,9547E+12
Húmedo	1997	38	4,1635E+12	4,3086E+12	4,4779E+12
Húmedo	2005	46	4,0542E+12	4,373E+12	4,2828E+12
Húmedo	1977	18	3,7467E+12	3,6224E+12	3,8504E+12
Húmedo	1986	27	4,2084E+12	4,0438E+12	4,4929E+12
Húmedo	1978	19	3,6556E+12	3,3125E+12	3,5662E+12
Promedio	1994	35	4,2394E+12	4,2965E+12	3,9467E+12
Promedio	1995	36	4,2303E+12	4,1382E+12	4,0292E+12
Promedio	1975	16	3,9165E+12	4,1207E+12	3,85E+12
Promedio	1969	10	3,7243E+12	4,1777E+12	4,0388E+12
Promedio	1981	22	3,9712E+12	3,7433E+12	3,8595E+12
Promedio	2008	49	4,847E+12	4,7281E+12	5,1316E+12
Promedio	2009	50	5,1299E+12	4,848E+12	4,9755E+12
Promedio	1971	12	3,8782E+12	4,0941E+12	3,6611E+12
Promedio	1974	15	3,8822E+12	3,7556E+12	4,0864E+12
Promedio	1973	14	3,7172E+12	4,1401E+12	3,7514E+12
Seco	2012	53	4,898E+12	4,6766E+12	5,1564E+12
Seco	1989	30	4,7429E+12	4,4753E+12	4,819E+12
Seco	1988	29	4,4073E+12	4,6235E+12	4,9248E+12
Seco	2010	51	4,8629E+12	4,6179E+12	5,0328E+12
Seco	2007	48	4,9976E+12	4,6368E+12	4,7414E+12
Seco	1962	3	4,7193E+12	4,384E+12	4,8178E+12
Seco	2013	54	4,3286E+12	4,2903E+12	4,8933E+12
Seco	1996	37	4,145E+12	4,3453E+12	4,3021E+12
Seco	1968	9	4,5819E+12	4,6273E+12	4,3124E+12
Seco	1998	39	4,5262E+12	4,0647E+12	4,2767E+12

Tabla A.5: Porcentaje de variación de F.O. por escenario

Caracterizado	Año	sim. PLP	PH paralelo		SDDP
			NSG	SYNTH	PLP
Húmedo	1972	13	-2,1 %	-9,8 %	0,0 %
Húmedo	1982	23	0,0 %	-1,8 %	-7,1 %
Húmedo	1965	6	0,0 %	-7,7 %	-0,1 %
Húmedo	2002	43	-7,9 %	-1,2 %	0,0 %
Húmedo	1980	21	-3,3 %	-1,7 %	0,0 %
Húmedo	1997	38	-7,0 %	-3,8 %	0,0 %
Húmedo	2005	46	-7,3 %	0,0 %	-2,1 %
Húmedo	1977	18	-2,7 %	-5,9 %	0,0 %
Húmedo	1986	27	-6,3 %	-10,0 %	0,0 %
Húmedo	1978	19	0,0 %	-9,4 %	-2,4 %
Promedio	1994	35	-1,3 %	0,0 %	-8,1 %
Promedio	1995	36	0,0 %	-2,2 %	-4,8 %
Promedio	1975	16	-5,0 %	0,0 %	-6,6 %
Promedio	1969	10	-10,9 %	0,0 %	-3,3 %
Promedio	1981	22	0,0 %	-5,7 %	-2,8 %
Promedio	2008	49	-5,5 %	-7,9 %	0,0 %
Promedio	2009	50	0,0 %	-5,5 %	-3,0 %
Promedio	1971	12	-5,3 %	0,0 %	-10,6 %
Promedio	1974	15	-5,0 %	-8,1 %	0,0 %
Promedio	1973	14	-10,2 %	0,0 %	-9,4 %
Seco	2012	53	-5,0 %	-9,3 %	0,0 %
Seco	1989	30	-1,6 %	-7,1 %	0,0 %
Seco	1988	29	-10,5 %	-6,1 %	0,0 %
Seco	2010	51	-3,4 %	-8,2 %	0,0 %
Seco	2007	48	0,0 %	-7,2 %	-5,1 %
Seco	1962	3	-2,0 %	-9,0 %	0,0 %
Seco	2013	54	-11,5 %	-12,3 %	0,0 %
Seco	1996	37	-4,6 %	0,0 %	-1,0 %
Seco	1968	9	-1,0 %	0,0 %	-6,8 %
Seco	1998	39	0,0 %	-10,2 %	-5,5 %

Anexo J: Tiempos de ejecución de lenguajes: Fortran vs Python

Las testeos que se exponen a continuación es parte de un estudio realizado por Julia Lang ORG [28]

La siguiente tabla, muestra la comparación entre tiempos de cómputo para 7 pruebas distintas, tomando como base 1 el tiempo que tarda en ejecutar el lenguaje C:

Tabla A.6: Tiempos de las pruebas comparativas en relación a C (menos es mejor, desempeño de C = 1.0).

Pruebas \ Lenguaje	Fortran gcc 5.1.1	Python 3.4.3	Dif. Porcentual
fib	0,7	77,76	11009 %
parse_int	5,05	17,02	237 %
quicksort	1,31	32,89	2411 %
mandel	0,81	15,32	1791 %
pi_sum	1	21,99	2099 %
rand_mat_stat	1,45	17,93	1137 %
rand_mat_mul	3,48	1,14	-67 %

Las pruebas fueron realizadas en una máquina con las siguientes características: Intel(R) Xeon(R) CPU E7-8850 2.00GHz CPU con 1TB de 1067MHz DDR3 RAM, sistema operativo Linux.

Se observa una diferencia significativa, entre los tiempos de cómputo de Python y Fortran.