



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

**COMPLETE STOCHASTIC FORESTRY PLANNING  
PROBLEM USING PROGRESSIVE HEDGING  
ALGORITHM**

TESIS PARA OPTAR AL GRADO DE MAGISTER EN GESTION DE  
OPERACIONES

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL INDUSTRIAL

**CRISTÓBAL ANDRÉS PAIS MARTÍNEZ**

PROFESOR GUÍA:  
ANDRÉS WEINTRAUB POHORILLE

MIEMBROS DE LA COMISIÓN:  
FERNANDO ORDOÑEZ PIZARRO  
DAVID L. WOODRUFF

SANTIAGO DE CHILE  
2014

*Para Sergio, Pegs, Mari, Bel, Tío Ignacio, Tía Mariel, Flaca, Coca, Ale, Igna,  
Alvarito, Moncho, Nelly, Tía Carmen,  
Don Yayo, JP, Cannobbi, Negro, Maikel, Reid  
y todas las grandes personas que siempre han estado conmigo*

# Tabla de Contenido

<b>1. Introducción.....</b>	<b>1</b>
1.1 Estado del Arte.....	2
1.2 Organización del contenido.....	3
<b>2. Marco Teórico .....</b>	<b>4</b>
2.1 Árboles de Escenarios.....	5
2.2 No anticipatividad.....	7
2.2.1 Formulaciones Extendidas.....	8
2.3 Progressive Hedging .....	9
2.3.1 Intuición.....	9
2.3.2 Pseudo-Código .....	11
2.3.3 Caracterización .....	11
<b>3. Problema forestal.....</b>	<b>13</b>
3.1 Antecedentes.....	13
3.2 Descripción.....	14
3.3 Incertidumbres .....	15
3.4 Generación de Escenarios .....	16
3.5 Modelo .....	23
3.5.1 Formulación Determinista.....	24
3.5.2 Formulación Extendida Estocástica por Escenarios.....	28
3.5.3 Fortalecimientos.....	32
3.6 Instancias .....	35
3.6.1 Componentes determinísticos .....	35
3.6.2 Componentes Estocásticos.....	36
3.6.3 Tamaño.....	37

<b>4. Algoritmo Progressive Hedging.....</b>	<b>38</b>
4.1 Pseudo-código extendido .....	38
4.2 Ajustes y mejoras .....	40
4.2.1 Acelerar la resolución de cada subproblema .....	40
4.2.2 Paralelizar el algoritmo .....	41
4.2.3 Linealizar el término cuadrático proximal .....	43
4.2.4 Ajustar el parámetro $\rho$ de penalización .....	45
4.2.5 Criterio de Término .....	47
4.2.6 Clustering y reducción de Escenarios.....	47
4.2.7 Bundling de Escenarios .....	50
4.2.8 Fijar Variables .....	51
<b>5. Implementación .....</b>	<b>54</b>
5.1 Serial: PYOMO – PYSP .....	54
5.2 Paralela: PYSP .....	55
5.3 Simulación.....	55
5.4 Comparación.....	55
5.5 Hardware y Software (Recursos computacionales).....	56
<b>6. Resultados .....</b>	<b>57</b>
6.1 Problema General.....	57
6.2 Caso Serial .....	59
6.3 Caso Paralelo .....	62
6.4 Caso Bundlings .....	65
6.5 Caso Reducción de escenarios.....	67
6.6 Comparaciones .....	70
<b>7. Calidad de la solución: Modelo Determinístico vs Estocástico.....</b>	<b>72</b>
7.1 Metodología.....	73
7.2 Implementación.....	75

7.2.1	Software Utilizado.....	75
7.2.2	Hardware y software .....	77
7.3	Modelo de optimización Alternativo.....	77
7.4	Resultados.....	78
7.4.1	Problema General .....	79
7.4.2	Simulación.....	80
<b>8.</b>	<b>Conclusiones .....</b>	<b>85</b>
8.1	Trabajo Futuro .....	88
<b>9.</b>	<b>Glosario.....</b>	<b>90</b>
<b>10.</b>	<b>Referencias .....</b>	<b>91</b>
<b>11.</b>	<b>Anexos.....</b>	<b>93</b>
11.1	Restricción de no anticipatividad .....	93
11.1.1	Simple.....	93
11.1.2	Tradicional.....	93
11.1.3	Promedio .....	93
11.1.4	Promedio General .....	94
11.1.5	Binaria.....	94
11.2	Modelo generador de Escenarios.....	94
11.2.1	Ejemplos de árboles de escenarios .....	96
11.3	Reducción de escenarios: Algoritmo Fast Forward .....	99
11.3.1	Teorema de redistribución óptima.....	100
11.4	Resultados de Simulación.....	100
11.5	Resultados Formulación extendida, PH serial y paralelo .....	123
11.6	Estrategias para el cálculo de $\rho$ .....	123

# Índice de Figuras

2.1	Árbol de Escenarios y formulación extendida .....	7
2.2	Representación del principio de no-anticipatividad, donde los nodos encerrados por elipses representan conjuntos de información .....	8
3.1	Diagrama de uno de los bosques de la instancia Millalemu .....	14
3.2	Árbol de Precios y relación del vector Alfa con probabilidades .....	18
3.3	Ejemplo de Árbol de Escenarios para 3 periodos y distintos vectores Alfa .....	19
3.4	Gráfico de precios históricos en UF para las trozas aserrables y trozas pulpables .....	20
3.5	Output gráfico generado por el script generador de Escenarios en MATLAB .....	21
3.6	Esquema de trabajo para la creación de las instancias del problema .....	21
3.7	Output generado por el script en MATLAB en txt (superior) y su representación gráfica equivalente .....	22
3.8	Ejemplo output de parámetros estocásticos para cada escenario .....	22
3.9	Ejemplo de output en formato PySP generado .....	23
3.10	Diagrama de la instancia Millalemu con los 17 bosques .....	36
3.11	Tamaño de las instancias experimentales .....	37
4.1	Esquema de programación en paralelo utilizando el Algoritmo PH y el paquete PySP de Coopr Sandia .....	43
4.2	Diagrama de linealización por tramos.....	44
4.3	Valor del parámetro Rho según instancia .....	46
4.4	Bundling de Escenarios.....	51
6.1	Gráfico con convergencia del algoritmo PH en función del número de iteraciones.....	58
6.2	Convergencia del algoritmo PH para una instancia de test.....	59
6.3	Experimentos realizados con las instancias en formulación extendida y PH serial.....	60
6.4	Gráfico de comparación entre los tiempos de resolución de la implementación serial de PH y la formulación extendida, para las instancias experimentales .....	61

6.5	Cuadro comparativo entre implementación serial y paralela (con 4 núcleos) del algoritmo PH .....	63
6.6	Porcentaje de tiempo de resolución bajo PH paralelo en relación a la formulación extendida y PH serial.....	64
6.7	Comparación rendimiento implementación paralela según n° de procesadores .....	64
6.8	Resultados instancias experimentales utilizando técnica de bundling.....	67
6.9	Resultados reducción de escenarios bajo algoritmo fast-forward .....	69
6.10	Resultados reducción de escenarios para instancias masivas bajo algoritmo fast-forward.....	70
7.1	Esquema de trabajo utilizado para la implementación de la metodología de simulación.....	76
7.2	Ejemplo del comportamiento (teórico) esperado por parte de la ganancia de realizar la formulación estocástica en comparación a un modelo aproximado .....	80
7.3	Extracto de resultados de simulación instancia 50 escenarios .....	81
7.4	Diferencia promedio entre las soluciones evaluadas bajo el modelo determinístico y el estocástico según el número de escenarios considerados .....	82
7.5	Valor de la solución del modelo determinístico en función del número de escenarios trabajados .....	83
11.1	Árbol de seis escenarios para un periodo.....	96
11.2	Árbol de 10 escenarios para 3 periodos .....	97
11.3	Árbol de 24 escenarios para 3 periodos .....	98
11.4	Resultados de simulación para instancia de 10 escenarios .....	102
11.5	Resultados de simulación para instancia 20 escenarios .....	104
11.6	Resultados de simulación para instancia 30 escenarios .....	106
11.7	Resultados de simulación para instancia 50 escenarios .....	108
11.8	Resultados de simulación para instancia 100 escenarios .....	110
11.9	Resultados de simulación para instancia 200 escenarios .....	112
11.10	Resultados de simulación para instancia 300 escenarios .....	114

11.11	Resultados de simulación para instancia 400 escenarios .....	116
11.12	Resultados de simulación para instancia 500 escenarios .....	118
11.13	Resultados de simulación para instancia 729 escenarios .....	120
11.14	Resultados de simulación para instancia 1000 escenarios .....	122
11.15	Tiempos de inicialización para instancias experimentales según método de solución .....	123



## COMPLETE STOCHASTIC FORESTRY PLANNING PROBLEM USING PROGRESSIVE HEDGING ALGORITHM

El trabajo desarrollado en esta tesis se enmarca dentro del área de investigación de operaciones, en el campo de la optimización estocástica para un problema de planificación forestal, el cual se modela como un problema lineal de carácter mixto.

El problema de planificación forestal abordado en esta tesis consiste en la maximización del valor presente del plan de explotación táctico de un conjunto de 17 bosques sujeto a incertidumbres internas (rendimiento de los diversos predios) y externas (precio de los productos y cantidad demandada de estos), incertidumbres que se representan a través de árboles de escenarios. Dentro de las principales decisiones a realizar se encuentra la construcción y/o mejora de caminos de la red interna junto a la cantidad de cada predio que será cosechada, almacenada y vendida durante cada periodo.

Llevar a cabo la resolución de este tipo de problemas estocásticos bajo las técnicas de optimización tradicionales se vuelve inabordable en la medida que el tamaño de la instancia aumenta, lo que justifica la utilización de un algoritmo adecuado para alcanzar resultados en tiempos razonables. La metodología de resolución utilizada se basa en el algoritmo de descomposición por escenarios Progressive Hedging (PH) sobre el cuál se realizan una serie de ajustes según las características del problema con tal de obtener mejores rendimientos. Junto a esto, se realizan una serie de comparaciones de diversas implementaciones del mismo algoritmo con respecto a técnicas de optimización tradicionales, determinando y analizando las ventajas que ofrece PH para el problema en estudio.

Se implementa una metodología de generación de árboles de escenarios basada en un modelo matemático robusto, utilizando como punto de partida un proceso estocástico para representar el movimiento de las variables aleatorias, lo que permite obtener soluciones rigurosas y de calidad.

Se lleva a cabo un estudio detallado respecto a técnicas de reducción de escenarios, así como del desarrollo de una metodología de simulación que permite comparar de forma cuantitativa el rendimiento de modelos estocásticos y determinísticos, permitiendo estimar el número suficiente de escenarios que logran representar de buena manera la realidad, manteniéndose óptimo el trade-off entre calidad de solución y tiempos de cómputo involucrados.

Los resultados concluyen que la utilización del algoritmo PH bajo las configuraciones y ajustes realizados ofrece rendimientos superiores a las técnicas de optimización clásicas y tradicionales para problemas estocásticos de gran escala, con la posibilidad de explotar de gran manera su implementación en paralelo, obteniéndose así rendimientos superiores en la medida que se cuente con mayores recursos computacionales. Se determinan los mejores métodos para llevar a cabo la reducción del número de escenarios de la instancia original, permitiendo así el manejo abordable de instancias que inicialmente se presentan complejas.

Finalmente, la aplicación de la metodología de comparación de rendimiento entre modelos determinísticos y estocásticos fue satisfactoria, determinándose un rango para el número suficiente de escenarios a utilizar dentro del modelo en estudio de tal forma de alcanzar un punto óptimo entre calidad de la solución y el rendimiento computacional del modelo.

# Capítulo 1

## Introducción

En la actualidad el sector forestal nacional alcanza los US\$ 5.300 Millones FOB, cifra que representa un 6.9% del total de exportaciones que realiza el país<sup>1</sup>. Dentro de dicha actividad, destaca la empresa Celulosa Arauco y Constitución, con más de US\$ 1.600 millones exportados, cifra que la convierte en la principal empresa chilena en cuanto a exportación. De este modo, la industria forestal corresponde al segundo mercado exportador más grande del país con un 2,47% del PIB, por debajo del sector minero, representando el 8,6% de las exportaciones chilenas, generando una cantidad considerable de empleos. Del mismo modo, se prevé un aumento en cuanto al nivel de exportaciones para los próximos años, así como del propio mercado interno.

Los principales productos que se manejan en la industria forestal chilena corresponden a tableros, MDF, molduras, pulpas, madera aserrada y celulosa, lo que va de la mano con una composición típica de bosques caracterizada por cerca de un 75% de pino radiata, seguida de un 17% de eucaliptus.

El mercado asiático se mantiene como el principal líder en cuanto a los destinos de las exportaciones forestales chilenas, adquiriendo el 45,7% de las ventas que realiza el país en la actualidad, cifra que representa compras del orden de los US\$ 2.400 millones, cifra que ha venido en aumento hasta la fecha. Los principales destinos de exportación para Chile consisten en China, Estados Unidos y Japón, según orden de prioridad. Los Estados Unidos cuentan con el 8% a nivel mundial (300 millones de hectáreas aproximadamente), por lo que poseen el cuarto patrimonio forestal más extenso del mundo, sólo superado por Rusia, Brasil y Canadá. Del mismo modo, Estados Unidos es el mayor consumidor de productos forestales, ocupando el segundo lugar (después de Canadá) entre los países productores, representando el 15% del comercio mundial.

Los principales elementos de la estructura de costos clásica de una empresa forestal chilena corresponden a los costos de transporte desde la zona de cosecha hasta la zona de venta (46%), seguidos de los costos de cosecha (34%) y finalmente la construcción de caminos (10%) dentro de las unidades de cosecha para establecer la comunicación y traslado de los diversos productos dentro de la red interna.

Una de las grandes ventajas que presenta esta industria en relación a otras industrias de manufactura y/o producción consiste en que la madera como tal es un recurso renovable en la medida que los bosques no sean sobre explotados o dañados gravemente por algún incidente, enfermedad o inclemencia temporal, y así mismo, presenta economías de escala altamente

---

<sup>1</sup> Fuente: INFOR y CORMA

explotables, lo que conlleva directamente a que la toma de decisiones dentro de la industria sea vital para conservar la buena salud de la materia prima y así mismo, realizar un uso eficiente de los recursos con los que se cuenta.

Tomando en cuenta que cerca de un 45% del territorio chileno presenta grandes características para llevar a cabo actividad forestal y que en la actualidad los bosques cubren cerca del 20,7% de la superficie del territorio nacional, es directo darse cuenta de la importancia y oportunidades que presenta este sector industrial dentro de nuestro país.

Así es como este trabajo de tesis se enfoca en la búsqueda de encontrar el mejor plan de explotación posible para un predio, tomando en cuenta la existencia de factores aleatorios o estocásticos que se manifiestan dentro de un horizonte de planificación táctico (4 años), tales como el comportamiento de los precios de mercado de cada uno de los productos que se planean vender. Es así como se aborda dicha problemática a través de un modelo de programación lineal estocástico, incluyendo de forma explícita las incertidumbres relacionadas al mercado, en contraposición a la clásica simplificación bajo cálculos de valores esperados promedio.

## 1.1 Estado del Arte

La presencia de variables aleatoria dentro de la formulación del problema generan 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 multietápico, 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.

Dado lo anterior, desde hace más de 20 años se han ido desarrollando una serie de modelos matemáticos que permitan determinar las decisiones óptimas a la hora de planificar la producción que se debe realizar en cada uno de los bosques. Así, en primera instancia se desarrollaron modelos determinísticos, donde todos los inputs son datos estables a lo largo del horizonte de planificación.

Otros modelos más complejos han sido desarrollado, sin embargo, debido al dinamismo inherente que presentan los mercados mundiales, impulsados por el efecto de la globalización y otros factores económicos, variables tales como el precio de los potenciales productos, así como la demanda que se experimentará en el horizonte de planificación debían ser incluidas como variables aleatorias.

Diversas técnicas han sido adaptadas a problemas estocásticos clásicos bi-etápico compuestos por variables continuas, tales como Relajación Lagrangeana, Descomposición de Benders, Programación Dinámica Estocástica, entre otros. Sin embargo, la utilización de dichas metodologías se vuelve inútil al presentarse un problema multi-etápico, que involucre cientos de escenarios y por ende variables y restricciones. En el problema abordado, no existe forma

directa o natural de condensar las incertidumbres en una única etapa para obtener un problema bi-etápico equivalente, y al mismo tiempo, la presencia de variables binarias impide la utilización de métodos de simulación tipo Montecarlo.

Dado lo anterior es que surge el algoritmo PH, desarrollado por Roger Wets y David Woodruff, de la Universidad de Davis, California. Éste consiste en un algoritmo de descomposición horizontal – es decir, según escenarios –, lo que permite tratar problemas de gran escala al separarlos en una serie de problemas pequeños y de fácil solución. La idea central de este tesis será determinar si esta hipótesis es válida, llevando a cabo una serie de experimentos y diversas configuraciones que permitan comparar el rendimiento de las técnicas tradicionales con respecto al uso del algoritmo.

Una colección respecto de la investigación de operaciones aplicada en la industria forestal se encuentra en [16]. Junto a esto, existe una serie de documentación respecto a diversas aplicaciones del algoritmo PH, así como de su utilización a modo de heurística en una serie de problemas de tipo estocástico en [2], [4], [6], [9], [10] y [19].

## **1.2 Organización del contenido**

El contenido de la presente tesis se desarrolla bajo el siguiente esquema: En el capítulo 2 se presentan los conceptos básicos y fundamentales respecto a la optimización estocástica, junto a una breve introducción respecto al algoritmo Progressive Hedging. En el capítulo 3 se presenta el problema forestal abordado, junto a las modificaciones realizadas, además de las incertidumbres involucradas en su confección. Luego, en el capítulo 4 se desarrolla toda la teoría relacionada al algoritmo PH, describiendo la forma en que éste opera, junto a los distintos ajustes y configuraciones con las que se experimenta. En el capítulo 5 se presentan las diversas implementaciones realizadas a lo largo del trabajo de tesis en relación al algoritmo PH.

A continuación, el capítulo 6 da cuenta de los resultados obtenidos, junto a un análisis de estos. En el capítulo 7 se desarrolla una metodología de simulación que permite contrastar el rendimiento entre modelos determinísticos y estocásticos, presentándose la implementación y resultados obtenidos. Finalmente en el capítulo 8 se desarrollan las conclusiones y desafíos futuros que se desprenden de la investigación realizada.

## Capítulo 2

# Marco Teórico

En el presente capítulo se formalizan los conceptos y definiciones que serán utilizados a lo largo de la presente tesis, así como introducir el funcionamiento del algoritmo Progressive Hedging junto a otros elementos característicos de la programación estocástica, necesarios para el desarrollo del trabajo realizado.

La investigación y trabajo realizados se enmarcan dentro del área de la investigación de operaciones, centrándose en específico en la optimización estocástica entera para problemas de gran escala, donde se deben tomar una serie de decisiones bajo incertidumbre en un periodo de estudio compuesto por múltiples etapas, lo que complejiza de forma significativa la posible resolución del mismo en tiempos razonables. Así, a través tanto de la optimización exacta como por métodos heurísticos, se busca establecer un plan de toma de decisiones robusto que incorpore los elementos estocásticos que se enfrentan, de tal forma que dicho plan de acción sea el “óptimo” ante cualquier escenario posible, condicionado por la realización de cada variable estocástica en cada uno de los periodos en estudio.

Esta planificación queda representada de forma explícita a través de un árbol de decisión, el cual representa la mejor respuesta frente a los diversos escenarios futuros a los que se les haga frente.

El objeto de estudio de la presente tesis consiste en un problema de planificación forestal, el cual se encuentra sujeto a incertidumbres externas (precios de los productos en el mercado) e internas (rendimiento de los rodales). Es debido a estas incertidumbres que el problema se vuelve intratable con los recursos computacionales actuales a través de enfoques de optimización tradicionales, por lo que requiere ser reformulado y abordado a través de una heurística que explote las características y estructura propia del problema, tales como su independencia por escenarios y la ligazón de las variables entre las diferentes etapas. Por ende, se utiliza la heurística conocida como Progressive Hedging, descrita en la sección (2.3)

En particular, se aborda el problema de una firma forestal, la cual es dueña de una serie de bosques y predios, resolviendo en reiteradas ocasiones las distintas instancias para el mismo problema en estudio.

La incertidumbre se incluye en el problema de optimización a través de la creación de una serie de escenarios (2.1) que discretizan los posibles estados de la naturaleza según el periodo en estudio y la forma funcional utilizada para medir el riesgo consiste en la esperanza.

En relación a la estructura general de un problema de optimización estocástica, se tiene que éste puede formularse de la siguiente manera:

$$\min \sum_{s \in S} p_s f_s(x_s) \quad (2.1)$$

$$s. a. \quad x_s \in Q_s \quad \forall s \in S \quad (2.2)$$

$$x_{s,t} = x_{s',t} \quad \forall s, s' \in N_t, \forall t \in T \quad (2.3)$$

En éste, existen una serie de variables de decisión  $x_s$  para cada uno de los escenarios  $s \in S$ , con restricciones propias para cada uno de los escenarios (ecuación (2.2)), así como aquellas que representan el principio de no anticipatividad (ecuación (2.3)) y una función objetivo donde se busca la minimización (en este caso de costos) de la esperanza para una función determinada según los escenarios y su probabilidad de ocurrencia (ecuación (2.1)). Esta formulación clásica es conocida como “Formulación Extendida Explícita” (subsección (2.2.1)) y tal como se verá más adelante, es posible realizar modificaciones con respecto a la forma de modelar las restricciones de no anticipatividad (Anexos sección (11.1)).

## 2.1 Árboles de Escenarios

Dentro del marco teórico relacionado a la optimización estocástica, es habitual representar los posibles valores que pueden alcanzar las principales variables y parámetros del sistema en estudio a través de un número finito de estados que puede experimentar y alcanzar el sistema completo a través de un periodo de tiempo determinado. Esta metodología va de la mano con el hecho de que los problemas que típicamente se abordan en esta disciplina buscan determinar las decisiones a tomar hoy en día a priori, es decir, sin conocer con certeza previa los valores que tomarán los parámetros relevantes para la toma de decisiones en el futuro.

En este contexto, 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 a futuro (hasta un horizonte de estudio determinado), incorporando la incertidumbre a partir de las probabilidades de ocurrencia que cada uno de estos tendrá asociado. Así, cada escenario contiene la serie de valores numéricos de cada una de las fuentes de incertidumbre del problema original, para todas las etapas temporales que se pretendan incluir, realizaciones que determinan una probabilidad de ocurrencia del mismo.

### Ejemplo 2.1.1

En un problema cuyo horizonte temporal de estudio es de 4 etapas ( $t \in \{0,1,2,3\}$ ) que cuenta con 3 fuentes de incertidumbre, sean las variables aleatorias  $X_1, X_2, X_3, Y_1, Y_2$  e  $Y_3$  aquellas correspondientes a las incertidumbres que se realizan en cada etapa. Un escenario quedará definido por el vector de valores numéricos que incluye a cada v.a. en cada periodo:

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

Cada escenario está asociado a una probabilidad de ocurrencia, lo que implica una discretización de las variables aleatorias consideradas en el modelo estocástico original. 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 pronostique el valor de un parámetro en periodos más distantes al actual.

En la etapa 0 (momento actual), 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

Utilizando los elementos del ejemplo anterior, en el caso de que existan cuatro escenarios  $S_1, S_2, S_3, S_4$  y  $S_5$  definidos por los vectores:

$$\begin{aligned} S_1 &= \{(X_1^a, Y_1^a), (X_2^a, Y_2^a), (X_3^a, Y_3^a)\} \\ S_2 &= \{(X_1^a, Y_1^a), (X_2^a, Y_2^a), (X_3^b, Y_3^b)\} \\ S_3 &= \{(X_1^a, Y_1^a), (X_2^b, Y_2^b), (X_3^c, Y_3^c)\} \\ S_4 &= \{(X_1^b, Y_1^b), (X_2^c, Y_2^c), (X_3^d, Y_3^d)\} \\ S_5 &= \{(X_1^b, Y_1^b), (X_2^d, Y_2^d), (X_3^e, Y_3^e)\} \end{aligned}$$

Se presenta en la figura 2.1 cómo estos escenarios definen un árbol de escenarios (izquierda), el cual además se observa en su forma “extendida” (derecha), donde los valores de las variables aleatorias son idénticos para los nodos encerrados en las elipses.

Gracias al concepto de árbol de escenarios, es posible incorporar la incertidumbre al modelo original, y además agrega el concepto de indistinguibilidad entre escenarios, donde se dice que dos escenarios son indistinguibles hasta la etapa  $t$  si todas sus variables aleatorias presentan idéntico valor hasta dicha etapa  $t$ .



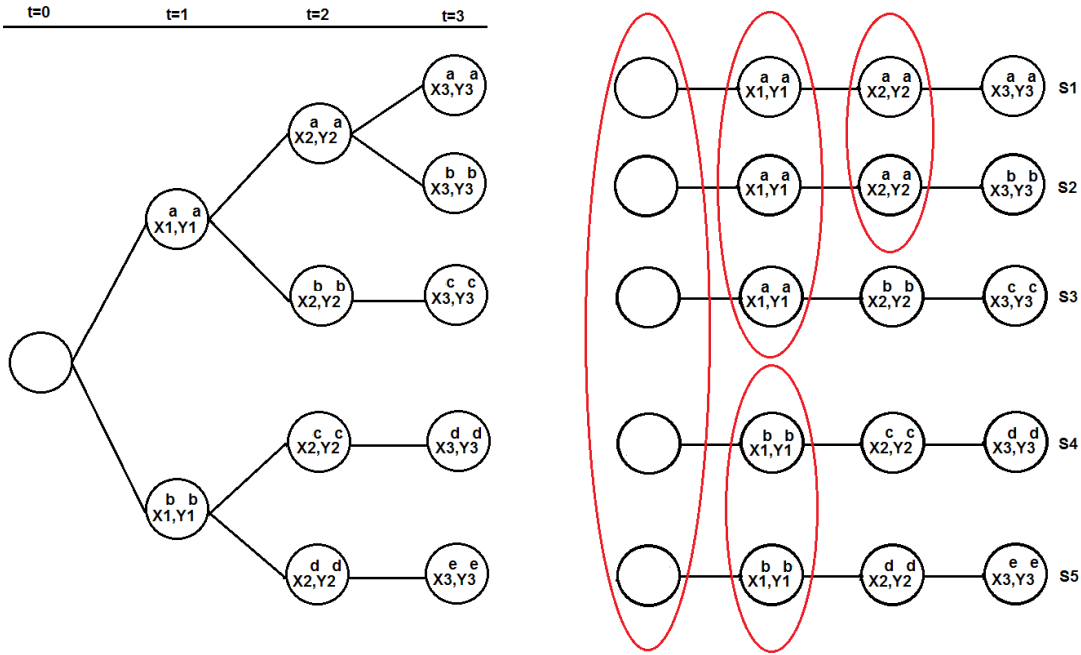


Figura 2.1: Árbol de Escenarios y formulación extendida

## 2.2 No anticipatividad

Uno de los principios más relevantes dentro del contexto de la optimización estocástica donde se utiliza la técnica de Árboles de Escenarios consiste en el llamado principio de no anticipatividad, enunciado y formulado por Rockafellar y Wets en [7], el cual señala que:

*“Si dos escenarios son indistinguibles hasta cierta etapa, entonces las decisiones en esos escenarios, hasta esa etapa, deben ser las mismas”.*

El principio de no anticipatividad verbaliza una intuición bastante natural, sin embargo, presenta profundas implicancias tanto a nivel teórico como de formulación de problemas de optimización estocástica. Así, la condición necesaria para que las decisiones de los distintos escenarios hasta una determinada etapa sean las mismas se observa gráficamente en la figura 2.2, los nodos rodeados por las elipses punteadas indican qué decisiones deben ser idénticas en cada una de las etapas en estudio, dada la realización de las diversas variables aleatorias. Estos grupos de nodos se denominan “Conjuntos de información”, que representan de forma directa a cada uno de los nodos del árbol de escenarios.

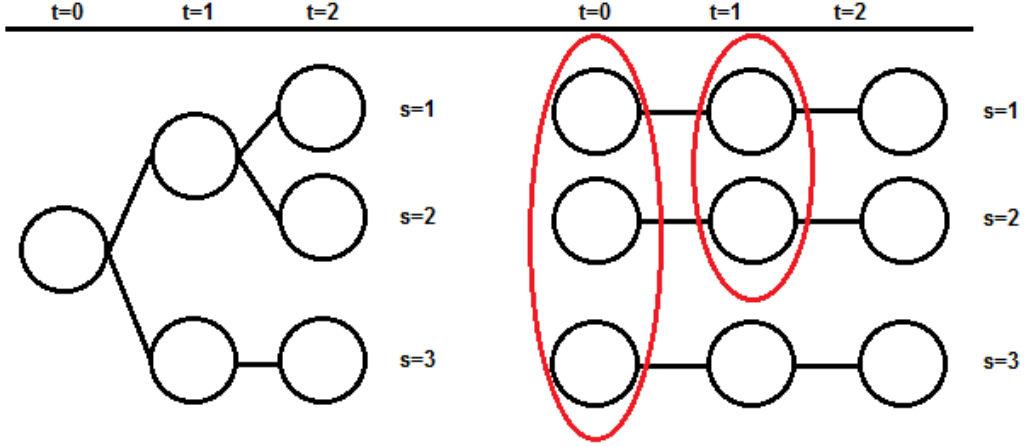


Figura 2.2: Representación del principio de no-anticipatividad, donde los nodos encerrados por elipses representan conjuntos de información

En términos de notación, el super-conjunto que contiene a todos los conjuntos de información presentes en el árbol de escenarios se denomina  $N$ , mientras que cada uno de sus elementos (conjuntos) se indexa según la etapa del problema correspondiente y la cantidad de conjuntos de información de la misma. Del mismo modo, cada grupo de escenarios se denota como  $N_t \in N$ . Cabe señalar que  $N_t$  podrá contener más de un único conjunto de información.

Algunas de sus 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 al tiempo:**

$$|N_t| \geq |N_{t+1}| : \forall t \in T, N_t \in N \subset P(S) \quad 1 \leq |N_t| \leq |S|$$

3. **En la etapa inicial, existe un único conjunto de información, con todos los escenarios:**

$$N_1 = S$$

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

$$\forall N_t \in N, |N_t| = 1$$

### 2.2.1 Formulaciones Extendidas

En base a las implicancias relativas al principio de no anticipatividad, se presentan una serie de formas equivalentes de formular el problema de programación estocástica en estudio:

**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

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

- $\sum_{N_t \in N} 1$  Variables por cada 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. También existen otras maneras que permiten incluir el principio de no anticipatividad dentro del problema, formulaciones que se presentan en Anexos sección (11.1).

## 2.3 Progressive Hedging

El algoritmo Progressive Hedging PH (Rockafellar y Wets, 1989 [4]) consiste en un algoritmo de descomposición utilizado para resolver problemas estocásticos que presentan una estructura de escenarios. La utilidad de su uso radica en que la formulación extendida del problema estocástico se vuelve intratable para instancias de gran tamaño, debido a la cantidad de columnas y filas que se agregan a partir de las restricciones de no anticipatividad, por ende, busca descomponer el problema original en una serie de problemas más pequeños y de fácil resolución.

PH presenta como propiedad el ser exacto para problemas lineales convexos, es decir, converge al óptimo global. Sin embargo, para problemas enteros esta condición no se garantiza, por lo que no se puede asegurar la optimalidad de la solución factible entregada.

### 2.3.1 Intuición

La intuición que subyace en esta descomposición consiste en la posibilidad de resolver el problema estocástico original a partir de la resolución independiente de cada uno de los escenarios modelados, y luego calcular la esperanza de la solución global obtenida. Así, se resolverían  $|S|$  (número de escenarios) problemas sencillos y determinísticos, lo que permite un mayor rendimiento en términos computacionales. No obstante, intercambiar la minimización con el cálculo de la esperanza no es posible debido a la presencia de las restricciones de no anticipatividad en el modelo, dado que éstas acoplan las decisiones a tomar entre los diversos escenarios.

Sin embargo, al observar la estructura de las soluciones factibles obtenidas del problema, a partir de las restricciones de no anticipatividad se tiene que:

$$\begin{aligned} x_s &= x_{s'}, \quad \forall s, s' \in S \\ &= \mathbb{E}(x_s) \end{aligned}$$

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  $\widetilde{w}_s$  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:

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

Realizando un cambio de notación y reagrupando términos ( $\widetilde{w}_s = p_s w_s$  y  $\mathbb{E}(w) = 0$ ), se obtiene:

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

Así, es posible realizar el intercambio entre la minimización y el cálculo de la esperanza, lo que permite descomponer el problema original en una serie de problemas más sencillos y pequeños.

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

En base a lo anteriormente realizado, el paso final consiste en realizar el cálculo iterativo de los multiplicadores  $w$  presentes en la función objetivo. Sin embargo, su convergencia bajo esta formulación no está garantizada, por lo que para evitar esta dificultad (que podría llevar a convertir los sub-problemas en no acotados), se agrega un término proximal cuadrático que obliga a las variables  $X$  a converger 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:

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

Importante es notar que en caso de que no se incluya este término cuadrático se podría producir que al realizar la minimización de la función objetivo, alguno de los multiplicadores  $w$  genere que alguna de las variables sea no acotada (por ejemplo, costo total negativo), y así el problema no logre converger.

**Ejemplo 2.3.1:** En caso de presentar una función objetivo lineal  $f_s(x_s) = c_s x_s$ , entonces puede darse  $c_s^i + w_s^i < 0$  para cierta variable  $i$  minimizando.

### 2.3.2 Pseudo-Código

---

#### Algoritmo Progressive Hedging

---

1. Resolver cada escenario según  $\min_{x_s \in Q_s} f_s(x_s)$ ;
  2. Calcular la solución en cada nodo,  $\bar{x} = \sum_{N_t: s \in N_t} p_s x_s$ ;
  3. *if las soluciones son suficientemente similares then*  
     **Stop**  $\|x - \bar{x}\| < \varepsilon$   
     *end*
  4. Actualizar el factor de penalización  $w_i = \rho(x - \bar{x}) + w_{i-1}$ ;
  5. Resolver cada uno de los escenarios penalizados:  
      $\min_{x_s \in Q_s} f_s(x_s) + w_s x_s + \frac{\rho}{2} \|x_s - \bar{x}\|^2$ ;
  6. 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 épsilon) en relación a la solución global.

### 2.3.3 Caracterización

Las principales características que presenta el algoritmo PH son:

- Es un algoritmo de descomposición horizontal (es decir, por escenarios, la descomposición vertical es por etapas, ver figura 4.1). En general, se han tratado estos problemas mediante descomposición vertical, siendo así una de sus innovaciones principales.

- No está limitado por el número de escenarios considerados, como sí lo está la formulación extendida que crece mucho más rápidamente en dificultad – tamaño y tiempo de resolución – al incrementar los escenarios.
- Relaja la restricción de no anticipatividad poniéndola en la función objetivo más un término proximal cuadrático.
- Posee una demostración basada en Relajación Lagrangeana Aumentada, lo que la hace exacta (converger al óptimo global) en el caso convexo.
- Es una heurística para el caso que involucra variables enteras (problema no convexo).
- Presenta una estructura fácilmente paralelizable, pues basta resolver cada subproblema en distintos núcleos de forma concurrente. Así, es posible utilizar programación distributiva directamente sobre un clúster computacional.
- Su desempeño está fuertemente ligado a lo rápido que se pueda resolver cada uno de los subproblemas (escenarios), dado que el resto del algoritmo consiste básicamente en calcular promedios ( $\bar{X}$ ) y las sumas lineales sobre el mismo ( $w$ ).

Más detalle de la aplicación y comportamiento que presenta el algoritmo en la práctica se encuentra en capítulo 4.

## Capítulo 3

# Problema forestal

El problema a resolver corresponde a la planificación óptima de la explotación de un conjunto de bosques en un horizonte temporal de 4 años, en específico en lo que se refiere a la actividad de cosecha dentro del ciclo forestal (Formación y producción, Manejo y Cosecha Forestales). Luego, en este horizonte es posible explotar una única vez cada una de las unidades forestales, puesto que el recurso no alcanza a renovarse dentro de periodo de estudio.

### 3.1 Antecedentes

El trabajo se basa en el problema presentado por Andalaft, 2002 [1], en la aplicación dentro de una industria real aún utilizada por las empresas forestales. En éste, Andalaft resuelve el complejo problema de optimización sin considerar incertidumbres, utilizando la relajación lagrangeana como principal técnica de resolución, pero incluyendo la instancia completa con los 17 bosques que constituyen Millalemu. Posteriormente el problema de planificación forestal fue abordado por Quinteros, 2009 [20], introduciéndose estocasticidad, reflejada en 18 escenarios de mercado. Simplificando la formulación original (relajando ciertas condiciones del problema original como el tipo de camino a construir/mejorar además de no permitir almacenamiento de stock), se resolvieron instancias a través del algoritmo de Branch & Fix especial para lidiar con las restricciones de no anticipatividad y la presencia de variables binarias.

Luego el problema fue abordado por Badilla, 2010 [21], donde se trabajaron una serie de instancias que incluían la presencia de más incertidumbres (se añade el rendimiento por hectárea), logrando resolver instancias de hasta 324 escenarios, utilizando fortalecimientos y diversas técnicas heurísticas complementarias al algoritmo PH, aun manteniendo las simplificaciones de la formulación original con respecto a los caminos (mejora) y lugares de acopio conocidos como Stocking Yards. Junto a lo anterior, se seleccionó un único bosque de los 17 originales, seleccionándose la instancia de “Los Copihues”, caracterizada por 25 unidades de cosecha, 9 orígenes, 6 intersecciones, 3 salidas, 46 caminos potenciales y 15 existentes.

El problema a tratar incluye las características y restricciones de la formulación original, incluyéndose la posibilidad de construir dos tipos de caminos, así como de mejorar aquellos de tierra a grava, únicamente durante el periodo de verano. Del mismo modo, se permite almacenar productos en zonas específicas conocidas como Stocking yards entre periodos. Finalmente, se buscan resolver instancias correspondientes a un conjunto de bosques (el conjunto original de 17), aumentando la complejidad del problema de forma significativa en comparación a los trabajos anteriores, donde se trabajaba con instancias ficticias bastante menores (en términos del número de rodales, caminos y cantidad de productos en estudio).

### 3.2 Descripción

En el problema de planificación forestal en estudio se desean conocer las decisiones relacionadas a la cantidad de rodales que serán explotados en cada periodo junto a la cantidad de hectáreas que serán extraídas de cada uno de estos, además del número y el tipo de caminos que se construirán y/o mejorarán en cada una de las etapas, de tal manera de determinar la cantidad de cada producto (madera para exportación, aserradero y pulpable) que se deberá producir y transportar a través de la red interna del bosque en estudio, sujeto a la incertidumbre relativa al precio de venta de cada uno de estos, la demanda experimentada (como cota mínima) y el rendimiento que presentará cada uno de los lotes en relación al crecimiento de la materia prima.

El objetivo que se persigue consiste en la maximización del beneficio neto esperado, tomando en cuenta la realización probabilística de cada uno de los escenarios confeccionados.

A modo de complemento visual del problema planteado, se presenta un diagrama donde se grafican – a modo de ejemplo – los diversos elementos que componen el problema forestal para la instancia Los Copihues, en la figura 3.1. En éste se pueden observar los diferentes rodales o unidades de cosecha enumerados desde el 1 hasta el 25, los caminos existentes representados por flechas de color azul, los caminos potenciales señalados por flechas rojas y los correspondientes orígenes o accesos donde se establece la relación entre las unidades de cosecha y los puntos desde los cuáles se pueden explotar, que se representan a través de flechas verdes en el diagrama adjunto. Junto a lo anterior, es posible observar el punto de salida desde el bosque hacia los caminos exteriores, representado por el punto de color naranja.

En el caso de la instancia original, todos los bosques pueden comunicarse entre sí con tal de utilizar salidas particulares para cada uno de los productos, así como los diferentes puntos de almacenamiento de estos.

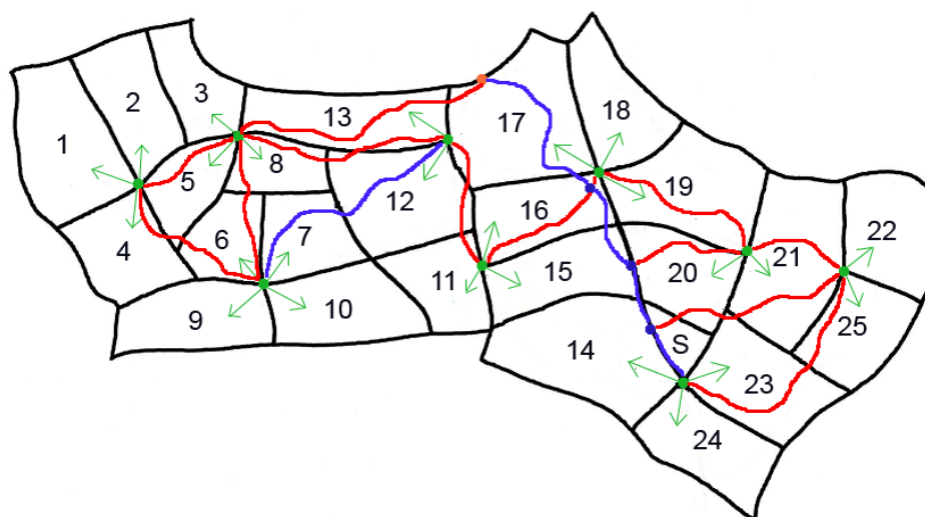


Figura 3.1: Diagrama de uno de los bosques de la instancia Millalemu



Preciso es señalar que en este caso se han reincorporado las restricciones del problema originalmente resuelto por Andalaft [1], en particular, se introducen las siguientes restricciones:

- Se producen tres tipos de productos: para exportación, aserradero y pulpable.
- Existencia de canchas de acopio conocidas como Stocking Yards en los predios, lo que permite almacenar productos de una temporada a la siguiente, bajo un costo asociado.
- Existencia de dos tipos de caminos: tierra y gravilla, lo que permite la construcción de cualquiera de estos durante las temporadas estivales. Del mismo modo, se permite la mejora de un camino de tierra a uno de gravilla, durante las mismas temporadas de verano, reincorporándose así las variables binarias de mejorar los caminos de tierra a gravilla durante estos periodos.

Así, a modo de resumen, se tiene:

1. Se presenta un bosque dividido en una serie de predios o unidades de cosecha, que se caracterizan por presentar árboles de características similares, homogéneos en cuanto a crecimiento y tipo de árbol. (En este ejemplo existen 25 unidades, y una caracterizada por la letra S donde se almacenan los productos).
2. Asociados a estas unidades se encuentran nodos denominados orígenes, desde los cuáles es posible acceder a cada una de éstas para llevar a cabo la cosecha.
3. Se tienen nodos de intersección, puntos del bosque que están relacionados a ciertos caminos (pero que no presentan otra función más que de paso).
4. Nodos de salida, que comunican al bosque con un camino público para transportar los diferentes productos hacia los clientes.
5. Caminos existentes (previamente construidos).
6. Caminos potenciales que podrían construirse dentro del horizonte de planificación.

Importante es señalar que los caminos pueden ser de tierra o gravilla (a diferentes costos) y existe la posibilidad de mejorar los caminos de tierra en verano.

Por ende, las decisiones principales son:

- a) Cuánto cosechar, y desde qué origen según cada unidad por periodo.
- b) Qué caminos se mejoran, construyen y utilizan para transportar los productos.
- c) La cantidad de productos almacenados entre periodos.

### **3.3 Incertidumbres**

Las incertidumbres a tratar dentro del problema son tanto internas como externas, incluyéndose en la primera categoría el rendimiento de madera incierto de cada bosque,

mientras que en la segunda se presentan aquellas relacionadas con el comportamiento del mercado, en particular respecto del precio de cada uno de los productos que se generan y la demanda que cada uno de estos experimentará en el tiempo. En particular tanto el precio como la demanda que experimenta la madera – que en la actualidad es un commodity de exportación – se encuentran sujetos a una serie de factores locales y globalizados, a nivel comercial y financiero.

En relación a la incertidumbre que se recoge en el rendimiento de cada una de las unidades de cosecha, en la actualidad, se utilizan una serie de simuladores dentro de la industria forestal con tal de proyectar el rendimiento de cada rodal según características significativas como el manejo de cada uno de estos y sus condiciones geográficas y de distribución espacial internas. En esta tesis se incorpora la incertidumbre en términos generales de realizar esta proyección, tomando como variable aleatoria el rendimiento global del predio, rendimiento que puede verse afectado principalmente por componentes y fenómenos climáticos que alteren de forma significativa y sostenida la cantidad de lluvias que experimenta el país o zona de estudio, afectando directamente el crecimiento de los árboles. No se incluyen los efectos de potenciales plagas o incendios que podrían acabar con una cantidad significativa de zonas explotables, fenómenos que deben ser estudiados, modelados y formulados de forma complementaria a lo realizado, trabajo que escapa de los objetivos de esta tesis.

La demanda por cada producto se representará a través de una cota mínima a vender/producir por cada uno de los periodos, correlacionada positivamente con el precio que experimenta el producto. Dicha cota representa la presencia de contratos previos con clientes. Cabe señalar que se eliminó la presencia de una cota superior, presente en los trabajos anteriores debido a la falta de sustento de dicha restricción.

Respecto al precio de venta de cada uno de los productos, éste había sido abordado en trabajos anteriores mediante un modelo de camino aleatorio (random walk, [22]) y bajo una serie de técnicas de simulación tipo Montecarlo. Esto se ha modificado en el actual trabajo, utilizándose como base un modelo estocástico de Wiener desarrollado por los profesores Andrés Weintraub y Roger Wets en [17], que ha sido implementado y continuado en el presente trabajo. Así, a través de los precios de venta históricos de cada uno de los productos durante los últimos 20 años, es posible crear una serie de escenarios para diversos periodos de estudio, con una base matemática sólida y fundamentada. La implementación de este método de generación de escenarios se encuentra en la sección (3.4) y fue desarrollada enteramente durante el trabajo de la presente tesis.

### **3.4 Generación de Escenarios**

Los resultados de la aplicación de la metodología de Modelamiento y optimización estocástica basada en árboles de escenarios dependerán en forma significativa de la calidad misma de los escenarios, por ende juega un rol fundamental el establecer la manera en que se generará cada uno de estos.

Una forma recurrente para la generación de escenarios en este tipo de problemas ha sido la confección “manual” de estos, a partir del conocimiento del sistema que presente alguno de los analistas y/o terceros involucrados ya sea en el desarrollo del modelo como en el contexto donde se desenvuelve el sistema en estudio, con respecto a las variables y parámetros relevantes que involucran incertidumbre en sus realizaciones.

Esta metodología, si bien útil en primera instancia, sobre todo para llevar a cabo pruebas y test de los modelos involucrados así como para la confección de sets medianos de escenarios representativos acordes a la visión de los conocedores del problema en estudio, presenta una serie de inconvenientes: dependencia sobre la persona encargada de la confección de los escenarios puesto que es ésta quien presenta el conocimiento necesario para realizar las predicciones y por ende no puede ser reemplazada de forma trivial, alto costo en tiempo (y por ende económico) en la confección de sets con un número de escenarios considerables, mayor probabilidad de error en el tratamiento de los datos y confección de cada escenario (factor humano) y carencia de un sustento/modelo riguroso que justifique el porqué del comportamiento de cada uno de estos.

Dado lo anterior, para el problema forestal en estudio se ha utilizado un modelo matemático para la generación de escenarios de precios de cada uno de los productos que se han incluido en el modelo desarrollado (madera exportable, aserradero y pulpable), otorgándole así mayor rigurosidad y escalabilidad a la creación de los diferentes árboles de precios para cada una de las instancias que se analizan en el presente proyecto.

El modelo matemático fue desarrollado específicamente para el problema forestal en estudio por Weintraub y Wets [17], donde se utiliza como punto de partida un proceso estocástico de Wiener para modelar el comportamiento de los precios asociados a cada uno de los productos en el tiempo. A partir de las propiedades del proceso estocástico señalado, es posible determinar la distribución que sigue el precio de los productos en el tiempo como una log-normal, hecho que permite calcular luego las funciones de densidad y distribución acumuladas de esta variable utilizando métodos numéricos, en función del periodo de tiempo que se necesite y los parámetros característicos del proceso (detalles en Anexos sección (11.2)).

Una vez calculadas las funciones de distribución acumuladas para cada uno de los periodos de interés de la variable precio, se procede a discretizar bajo algún vector  $Alfa_{t,i}$  (donde  $t$  representa el periodo e  $i$  representa el nodo raíz de dicho periodo que dará paso a los nuevos precios en la siguiente etapa temporal) la función de distribución acumulada para  $p(1)$ . El vector  $Alfa_{t,i}$  estará asociado directamente a un vector de probabilidades, donde cada una de sus componentes determinará de forma inequívoca la probabilidad de que ocurra alguno de los precios potenciales del próximo periodo, precios que se calculan directamente como la esperanza condicional de  $p(1)$  dado el intervalo de precios en que se discretiza la función distribución acumulada con el vector  $Alfa_{t,i}$ , siendo así directa la construcción de cada una de las ramas.

El procedimiento anterior se realiza de forma iterativa, modificando únicamente el valor del precio inicial en cada etapa (pasando a ser cada uno de los nodos “terminales” del periodo hasta el cuál se ha confeccionado el árbol de escenarios) y tomando la función distribución acumulada para el periodo correspondiente. Importante es notar que los vectores  $Alfa_{t,i}$  podrían ser todos idénticos o utilizar uno distinto para cada nodo/periodo confeccionado, según se requiera.

### Ejemplo 3.3.1

Sea  $p(0) = 10$  UF, el valor actual de un producto forestal. Sea el vector  $Alfa_{0,1} = [0, 0.2, 0.35, 0.45, 0.6, 0.9, 1]$  asociado al vector de probabilidades  $Prob_{0,1} = [0.2, 0.15, 0.1, 0.15, 0.3, 0.1]$  (notar que corresponde a las distancias entre las componentes del vector  $Alfa_{0,1}$ ).

Así, el árbol de precios hasta el periodo 2 ( $t=1$ ) quedaría:

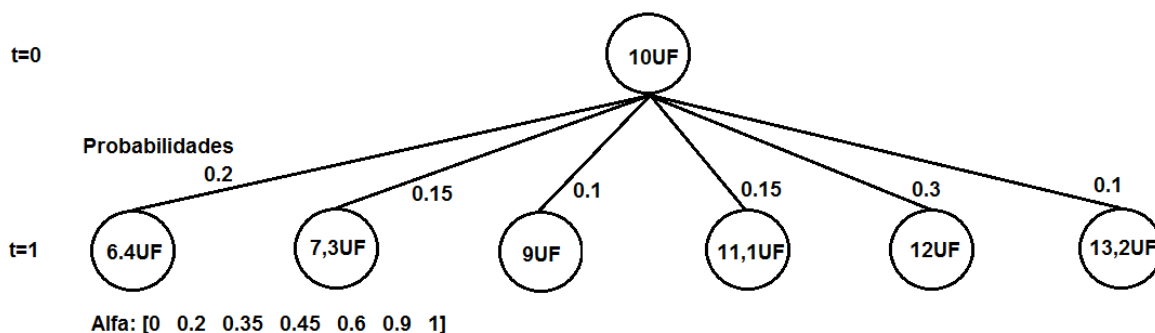


Figura 3.2: Árbol de Precios y relación del vector Alfa con probabilidades

Donde los valores de precios asociados a cada uno de los nodos del segundo periodo se calculan como la esperanza condicional de  $p(1)$  dado que éste se encuentra en cada uno de los intervalos generados por el vector Alfa.

Luego, si se usara el vector  $Alfa_{1,4} = [0, 0.4, 0.6, 1]$  (es decir, generar 3 nodos desde el nodo cuyo valor es 11.1UF en la figura anterior) asociado al vector  $Prob_{1,4} = [0.4, 0.2, 0.4]$ , se tiene el siguiente árbol:

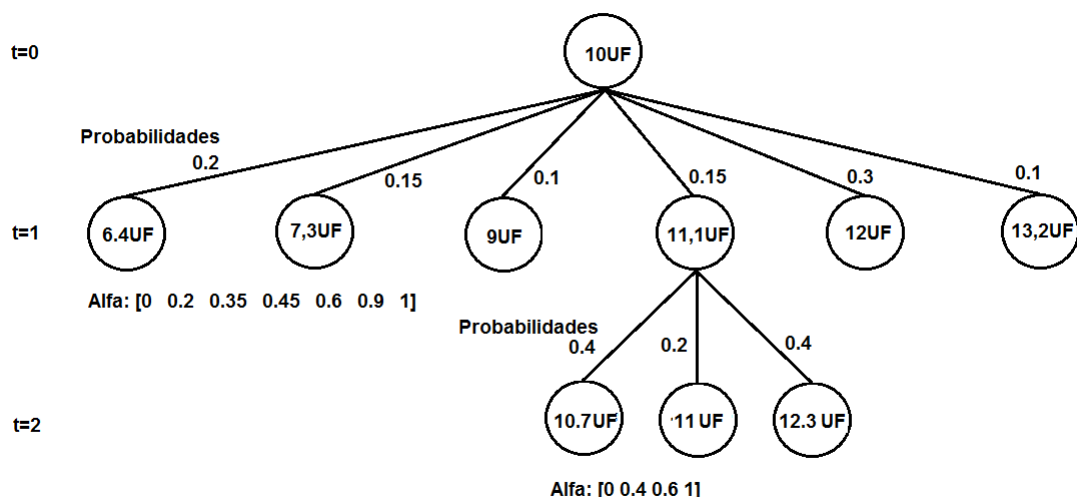


Figura 3.3: Ejemplo de Árbol de Escenarios para 3 periodos y distintos vectores Alfa

Para la confección de los diversos árboles de escenarios a utilizar con cada una de las instancias del problema en estudio, se ha implementado la metodología previamente descrita a modo de rutina en el software matemático MATLAB, donde se requiere como input el vector de precios de los productos en estudio de los últimos 20 años (en UF promedio por cada año), así como los diversos vectores *Alfa* con los que se pretende discretizar cada uno de los nodos por periodo, hasta alcanzar un número determinado de escenarios que se requiera.

El set de datos e información histórica con los que se trabaja en esta tesis contiene todos los valores de los principales productos de la industria forestal chilena y extranjera (tanto en USD como en UF por metro cúbico de producto) desde el año 1977 hasta el año 2013. Estos fueron obtenidos directamente desde la empresa Arauco, dueña de los terrenos de la instancia original Millalemu.

Toda la información está organizada por año,  $\$/m^3$ ,  $UF/m^3$ ,  $\$/UF$  y  $US\$/m^3$ , tanto para el mercado interno como para el mercado de exportación, teniéndose registro exacto del tipo de cambio que presentaba el peso chileno con respecto a la divisa norteamericana.

A modo de ejemplo, se presenta en la figura 3.4 el gráfico para el valor de transacción dentro del mercado interno de las trozas aserrables y trozas pulpables durante el periodo entre 1977-2007 (31 años):

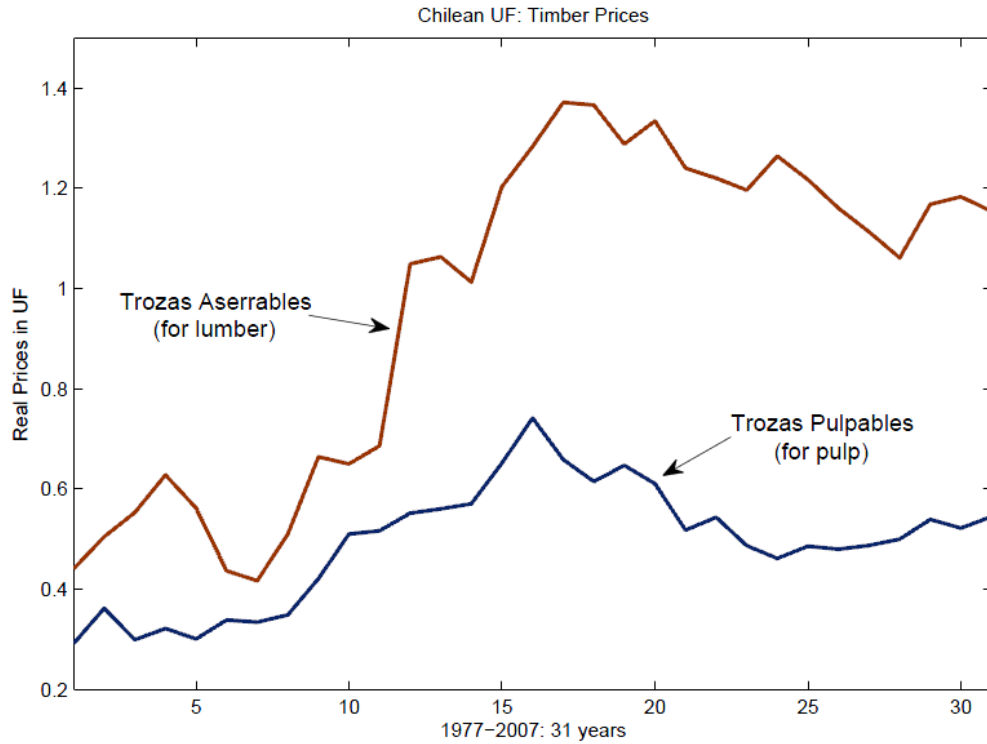


Figura 3.4: Gráfico de precios históricos en UF para las trozas aserrables y trozas pulpables

Con tal de generar distintos árboles de precios para instancias de tamaño, periodos y productos similares, se trabaja con un set preestablecido de vectores *Alfa*, que cumplen con otorgar suficiente probabilidad de ocurrencia a las colas y centro de la distribución acumulada, tomando así en cuenta tanto casos extremos como los de mayor factibilidad (black swans y eventos promedios). Este set está compuesto por 10 subconjuntos, cada uno de los cuáles ramifica un nodo desde uno hasta 10 nuevos nodos en la siguiente etapa (mayor cantidad de nodos hijos implica probabilidades muy menores, lo que conlleva a potenciales errores numéricos dentro del modelo), y al mismo tiempo, con tal de obtener distintos árboles para instancias idénticas, donde cada subconjunto contiene seis vectores *Alfa* distintos, cada uno de los cuáles se elegirá de forma aleatoria dentro de la rutina computacional, generándose así siempre nuevos árboles con diferentes particiones del espacio de precios posibles.

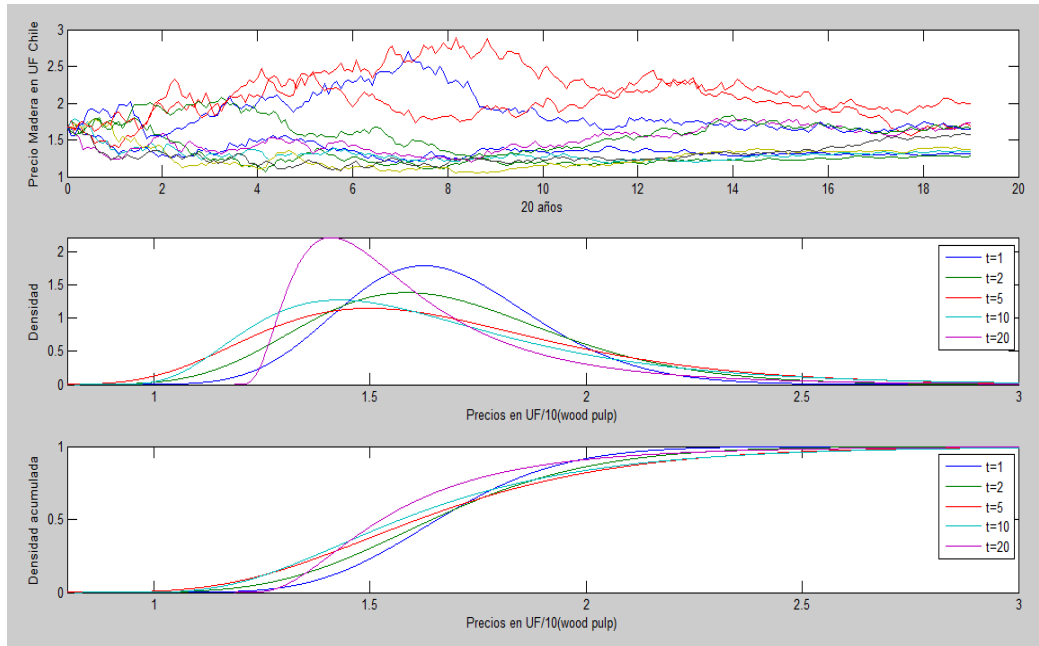


Figura 3.5: Output gráfico generado por el script generador de Escenarios en MATLAB

El típico output obtenido a partir del script computacional previamente señalado se puede observar en la figura 3.5, donde se grafica cada uno de los escenarios simulados para el precio del producto en estudio, así como el gráfico de las funciones de densidad y distribución acumuladas asociadas a cada uno de los periodos en estudio.

Cada uno de los Árboles de precios generados se almacena en un archivo de texto (formato TXT), junto a su relativo árbol de probabilidades, es decir, la colección de vectores de probabilidad condicional que relaciona a cada uno de los nodos con sus respectivos hijos, vectores que permiten así calcular la probabilidad de ocurrencia de cada uno de los escenarios mediante un simple cálculo de multiplicaciones. Una vez creados los Árboles de Escenarios requeridos, estos se procesan mediante un script desarrollado en el lenguaje computacional JAVA, programa que recibe los siguientes inputs: Arcos existentes y potenciales de la instancia en estudio a través de un archivo Arcos.txt, parámetros determinísticos del modelo forestal y los árboles generados previamente en Matlab (figura 3.7, más ejemplos en Anexos subsección (11.3.1)).

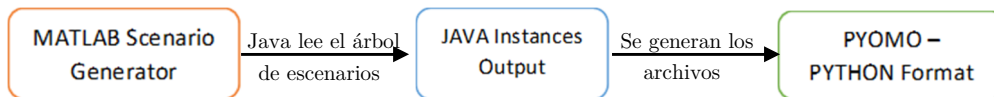


Figura 3.6: Esquema de trabajo para la creación de las instancias del problema

A partir de lo anterior, se obtiene un archivo en el formato de lectura del software de optimización AMPL con todos los parámetros y conjuntos necesarios para realizar la optimización de la formulación extendida y compacta del modelo (incluyendo los conjuntos de información asociados, ver sección de Progressive Hedging). Por otra parte se obtienen todos los archivos en el formato de lectura PySP (en lenguaje Python, ver figura 3.8) necesario para

llevar a cabo la optimización del modelo forestal, donde se incluye el archivo ScenarioStructure.dat que establece la forma en la que se compone el árbol de escenarios a trabajar con la instancia, además de todos los archivos que determinan los parámetros determinísticos y estocásticos que caracterizan a cada uno de los escenarios que se han generado.

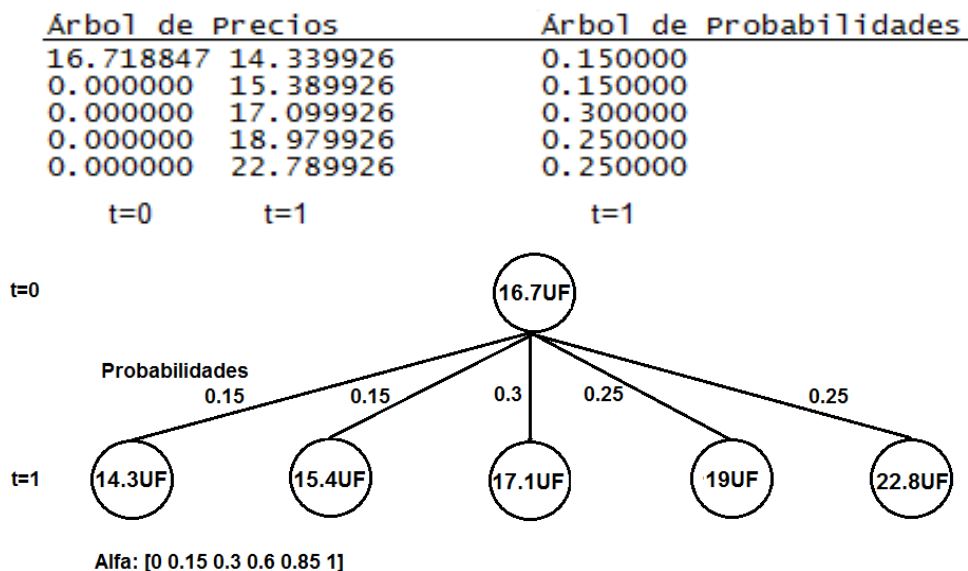


Figura 3.7: Output generado por el script en MATLAB en txt (superior) y su representación gráfica equivalente

En base a la metodología de simulación y generación de Árboles de Escenarios señalada, es posible construir una serie de instancias de prueba para el modelo en estudio de forma eficaz, eficiente y rigurosa, terminando así con la dependencia directa de un analista o experto que se encargue de construir cada uno de los escenarios, junto con los potenciales errores que dicha metodología conlleva, además de aumentar de forma significativa la velocidad con la que se obtienen nuevas instancias, puesto que el rendimiento de los algoritmos desarrollados permite generar árboles del orden de miles de escenarios en unos pocos segundos de procesamiento.

```
#Conjuntos de Información del árbol en estudio
Conjuntos_inf[1][1] := 1 2 3 4 5 6 7 8 9 10;
Conjuntos_inf[1][2] := 1 2 3 4 5 6;
Conjuntos_inf[2][2] := 7 8;
Conjuntos_inf[3][2] := 9;
Conjuntos_inf[4][2] := 10;

#Escenario Probabilidad
param Prob_escenario:=
1 0.05
2 0.020000000000000004
3 0.03
4 0.03
5 0.020000000000000004
6 0.05
7 0.15
8 0.15
9 0.3
10 0.2
;

#Producto Salida Periodo Escenario Valor
param Precio:=
1 S01 1 1 16.718847
1 S01 2 1 14.729926
1 S01 3 1 13.062726
1 S01 4 1 13.395059
1 S01 1 2 16.718847
1 S01 2 2 14.729926
1 S01 3 2 13.732726
1 S01 4 2 13.915059
.
.
.
1 S01 1 10 16.718847
1 S01 2 10 22.789926
1 S01 3 10 21.442726
1 S01 4 10 19.895059
;
```

Figura 3.8: Ejemplo output de parámetros estocásticos para cada escenario



```

1 #Estructura de Escenarios para InstanciaX
2 #escenarios: 3 periodos:2 productos: 2
3
4 set Stages := Stage1 Stage2;
5
6 set Nodes := RootNode
7             |
8             | |
9             | | |
10            | | | |
11           ;
12 param NodeStage := RootNode           Stage1
13                 |
14                 | |
15                 | | |
16                | | | |
17               ;
18 set Children[RootNode]:= Nodo[1][2]
19                 |
20                 | |
21                | | |
22               ;
23 param ConditionalProbability:= RootNode           1.0
24                 |
25                 | |
26                | | |
27               ;
28 set Scenarios:= Escenario1
29                 |
30                 | |
31                | | |

```

Figura 3.9: Ejemplo de output en formato PySP generado

### 3.5 Modelo

En esta sección se presentan los modelos de programación lineal mixta para el estudio del problema de planificación forestal. En particular, se presenta la formulación extendida del modelo estocástico, incluyendo como subíndice la presencia de los diversos escenarios simulados. Cabe señalar que el modelo corresponde a un mix del utilizado por Quinteros [20] y Badilla [21], debido a la presencia de las restricciones y variables relacionadas a los tipos y mejoras de caminos, así como de la posibilidad de almacenar productos durante las temporadas estivales en los centros de acopio, además de la reformulación de ciertas restricciones con tal de obtener un mayor rendimiento a la hora de resolver el problema en estudio.

### 3.5.1 Formulación Determinista

El modelo en su formulación determinista está caracterizado por los siguientes elementos:

- **Conjuntos**

$T$ : Horizonte de Planificación  $\{t\}$ .

$O$ : Nodos orígenes  $\{o\}$ .

$N$ : Nodos de Intersección  $\{n\}$ .

$NO$ : Nodos  $\{no\}$ .

$E$ : Nodos de Salida  $\{e\}$ .

$H$ : Unidades de Cosecha  $\{h\}$ .

$H_o$ : Orígenes  $\{o\}$ , Asociados a las unidades de cosecha  $\{h\}$ .

$O_h$ : Unidades de cosecha  $\{h\}$ , asociadas a los orígenes  $\{o\}$ .

$R^{DE}$ : Caminos Existentes de Tierra  $\{i, j\}$ .

$R^{GE}$ : Caminos Existentes de grava  $\{i, j\}$ .

$R^P$ : Caminos potenciales  $\{i, j\}$ .

$RT = R^{DE} \cup R^{GE} \cup R^P$ : Conjunto de todos los caminos.

$Y$ : Stocking yards  $\{y\}$ .

$R$ : Tipo de camino  $\{r\}$ .

$P$ : Productos  $\{p\}$ .

- **Parámetros**

$A_h$ : Área de la unidad de cosecha  $h$  [Ha]

$a_h^{p,t}$ : Productividad por [Ha] cosechada en la unidad  $h$  para el producto  $p$  en el periodo  $t$  [M<sup>3</sup>]

$U_{i,j}^{r,t}$ : Capacidad del camino  $(i, j)$  tipo  $r$  en el periodo  $t$  [M<sup>3</sup>]

$HC_h^t$ : Costo de cosechar una [Ha] de la unidad  $h$  en el periodo  $t$   $\left[\frac{\text{USD}}{\text{Ha}}\right]$

$PC_o^t$ : Costo de producir un [M<sup>3</sup>] en el origen  $o$  en el periodo  $t$   $\left[\frac{\text{USD}}{\text{M}^3}\right]$

$RC_{i,j}^{r,t}$ : Costo de construir el camino  $(i, j)$  tipo  $r$  en el periodo  $t$  [USD]

$TC_{i,j}^{r,t}$ : Costo de transportar un [M<sup>3</sup>] en el camino  $(i, j)$  tipo  $r$  en el periodo  $t$   $\left[\frac{\text{USD}}{\text{M}^3}\right]$

$UC_{i,j}^t$ : Costo de mejorar el camino  $(i, j)$  en el periodo  $t$   $\left[\frac{\text{USD}}{\text{M}^3}\right]$

$SC_y^t$ : Costo de almacenamiento por [M<sup>3</sup>] del producto  $p$  en el Stocking yard  $y$  durante el periodo  $t$   $\left[\frac{\text{USD}}{\text{M}^3}\right]$

$CAPS_y$ : Capacidad de almacenamiento del Stocking yard  $y$  [M<sup>3</sup>]

$P_{e,p}^t$ : Precio de venta en la salida  $e$  durante el periodo  $t$  del producto  $p$   $\left[\frac{\text{USD}}{\text{M}^3}\right]$

$\underline{Z}^{p,t}, \overline{Z}^{p,t}$ : Cotas de demanda en el periodo t para el producto p [M<sup>3</sup>]

- **Variables de Decisión**

$E_h^t = \begin{cases} 1 & \text{Si la unidad } h \text{ es cosechada en el periodo } t \\ 0 & \sim \end{cases}$

$W_{i,j,r}^t = \begin{cases} 1 & \text{Si el camino } (i, j) \text{ tipo } r \text{ se construye en el periodo } t \\ 0 & \sim \end{cases}$

$V_{i,j}^t = \begin{cases} 1 & \text{Si el camino } (i, j) \text{ se mejora en el periodo } t \\ 0 & \sim \end{cases}$

$X_h^t$  = Área de la unidad h cosechada en el periodo t [Ha].

$Y_{o,p}^t$  = Total del producto p recogido en el origen o durante el periodo t [M<sup>3</sup>].

$S_{y,p}^t$  = Total del product p almacenado en stocking yard y durante el periodo t [M<sup>3</sup>].

$f_{i,j,r}^{p,t}$  = Flujo del producto p transportado por el camino (i,j) de tipo r durante el periodo t [M<sup>3</sup>].

$z_{p,e}^t$  = Cantidad del producto p vendido en el periodo t, en la salida e [M<sup>3</sup>].

- **Restricciones**

1. Cantidad total de producto recolectada por origen

$$\sum_{h \in H} a_h^{p,t} X_h^t = Y_{o,p}^t \quad \forall o \in O_h, p \in P, t \in T$$

2. Límite de área cosechada por unidad

$$\sum_{t \in T} X_h^t - A_h \leq 0 \quad \forall h \in H$$

3. Restricciones de Flujo

a) Flujo de Orígenes

$$Y_{o,p}^t + \sum_{\substack{(i,o) \in RT \\ r \in R}} f_{i,o,r}^{p,t} - \sum_{\substack{(o,j) \in RT \\ r \in R}} f_{o,j,r}^{p,t} = 0 \quad \forall o \in O, p \in P, t \in T$$

b) Flujo de Intersecciones

$$\sum_{\substack{(i,n) \in RT \\ r \in R}} f_{i,n,r}^{p,t} - \sum_{\substack{(n,j) \in RT \\ r \in R}} f_{n,j,r}^{p,t} = 0 \quad \forall n \in N, p \in P, t \in T$$

c) Flujo de Salida

$$\sum_{\substack{(i,e) \in RT \\ r \in R}} f_{i,e,r}^{p,t} - \sum_{\substack{(e,j) \in RT \\ r \in R}} f_{e,j,r}^{p,t} - z_{p,e}^t = 0 \quad \forall e \in E, p \in P, t \in T$$

d) Flujo de Almacenamiento

$$\sum_{\substack{(i,y) \in RT \\ r \in R}} f_{i,y,r}^{p,t} - \sum_{\substack{(y,j) \in RT \\ r \in R}} f_{y,j,r}^{p,t} - S_{y,p}^t = 0 \quad \forall y \in Y, p \in P, t \in T$$

4. Definición de Almacenamiento

$$S_{y,p}^t - S_{y,p}^{t-1} - \sum_{\substack{(i,y) \in RT \\ r \in R}} f_{i,y,r}^{p,t} = 0 \quad \forall y \in Y, p \in P, t \in T > 1$$

5. Capacidad de Almacenamiento

$$\sum_{p \in P} S_{y,p}^t \leq CAPS_y \quad \forall y \in Y, t \in T$$

6. Cotas (superior e inferior) de producción

$$\underline{Z}^{p,t} \leq \sum_{e \in E} z_{p,e}^t \leq \overline{Z}^{p,t} \quad \forall p \in P, t \in T$$

7. Capacidad de los caminos

a) Caminos existentes (Grava)

$$\sum_{p \in P} f_{i,j,R2'}^{p,t} + \sum_{p \in P} f_{j,i,R2'}^{p,t} - U_{i,j}^{R2',t} \leq 0 \quad \forall (i,j) \in R^{GE}, t \in T$$

b) Caminos existentes 2 (Grava)

$$f_{i,j,R1'}^{p,t} = 0 \quad \forall (i,j) \in R^{GE}, t \in T$$

8. Construcción de caminos potenciales

a) Caminos potenciales (tierra)

$$\sum_{p \in P} f_{i,j,R1'}^{p,t} + \sum_{p \in P} f_{j,i,R1'}^{p,t} - U_{i,j}^{R1',t} \left( \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} W_{i,j,R1'}^\alpha - V_{i,j}^\alpha \right) \leq 0 \quad \forall (i,j) \in R^P, t \in T$$

b) Caminos Potenciales (grava)

$$\sum_{p \in P} f_{i,j,R2'}^{p,t} + \sum_{p \in P} f_{j,i,R2'}^{p,t} - U_{i,j}^{R2',t} \left( \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} W_{i,j,R2'}^\alpha + V_{i,j}^\alpha \right) \leq 0 \quad \forall (i,j) \in R^P, t \in T$$

c) A lo más se construye una vez cada camino potencial

$$\sum_{\substack{r \in R \\ \alpha \in T: T \text{ odd}}} W_{i,j,r}^\alpha \leq 1 \quad \forall (i,j) \in R^P$$

9. Mejora de los caminos

a) Caminos Existentes (grava)

$$\sum_{p \in P} f_{i,j,R2'}^{p,t} + \sum_{p \in P} f_{j,i,R2'}^{p,t} - U_{i,j}^{R2',t} \left( \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} V_{i,j}^\alpha \right) \leq 0 \quad \forall (i,j) \in R^{GE}, t \in T$$

b) Caminos Existentes (tierra)

$$\sum_{p \in P} f_{i,j,R1'}^{p,t} + \sum_{p \in P} f_{j,i,R1}^{p,t} - U_{i,j}^{R1',t} \left(1 - \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} V_{i,j}^\alpha\right) \leq 0 \quad \forall (i,j) \in R^{DE}, t \in T$$

c) Mejora desde tierra

$$V_{i,j}^t - \left(1 - \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} W_{i,j,R2'}^\alpha\right) \leq 0 \quad \forall (i,j) \in R^P, t \in T: t \text{ odd}$$

d) No se puede mejorar si el camino se construyó en dicho periodo

$$V_{i,j}^t - \left(\sum_{\alpha \in T: T \text{ odd}, \alpha < t} W_{i,j,R1'}^\alpha\right) \leq 0 \quad \forall (i,j) \in R^P, t \in T: t \text{ odd}$$

10. Capacidad de cosecha y relación de variables

$$X_h^t - A_h E_h^t \leq 0 \quad \forall h \in H, t \in T$$

11. Naturaleza de las variables

$$\begin{aligned} X_h^t, Y_{o,p}^t, S_{y,p}^t, f_{i,j,r}^{p,t}, z_{p,e}^t &\geq 0 \quad \forall e \in E, (i,j) \in RT, t \in T, r \in R, y \in Y, p \in P, o \in O \\ W_{i,j,r}^t, E_h^t, V_{i,j}^t &\in \{0,1\} \quad \forall h \in H, (i,j) \in R^P, r \in R, t \in T: t \text{ odd} \end{aligned}$$

## • Función Objetivo

$$\max T_1 - T_2 - T_3 - T_4 - T_5 - T_6 - T_7$$

donde:

$$\begin{aligned} T_1 &= \sum_{e \in P} \sum_{p \in P} \sum_{t \in T} P_{e,p}^{t,s} z_{p,e}^t \\ T_2 &= \sum_{h \in H} \sum_{t \in T} HC_h^t X_h^t \\ T_3 &= \sum_{o \in O} \sum_{t \in T} \sum_{p \in P} PC_o^t Y_{o,p}^t \\ T_4 &= \sum_{(i,j) \in RT} \sum_{t \in T} \sum_{p \in P} \sum_{r \in R} TC_{i,j}^{r,t} f_{i,j,r}^{p,t} \\ T_5 &= \sum_{y \in Y} \sum_{t \in T} \sum_{p \in P} SC_y^t S_{y,p}^t \\ T_6 &= \sum_{(i,j) \in R^P} \sum_{t \in T: \text{odd}} \sum_{r \in R} RC_{i,j}^{r,t} W_{i,j,r}^t \\ T_7 &= \sum_{(i,j) \in R^P} \sum_{UR^{de}} \sum_{t \in T: \text{odd}} UC_{i,j}^t V_{i,j}^t \end{aligned}$$

La función objetivo busca maximizar el beneficio neto, el cuál está compuesto por los ingresos de la venta de los productos ( $T_1$ ) y seis tipos de costos: costo de cosecha ( $T_2$ ), costo

de producción en los nodos de origen ( $T_3$ ), el costo de transporte ( $T_4$ ), costo de almacenamiento ( $T_5$ ), costo de construcción de caminos ( $T_6$ ) y el costo de mejora de estos ( $T_7$ ).

### 3.5.2 Formulación Extendida Estocástica por Escenarios

Esta formulación difiere de la anteriormente señalada debido a la presencia de un conjunto que representa a los escenarios  $S$ , así como del súper conjunto de los paquetes de información  $N$  y sus respectivos conjuntos indexados según periodo  $N_t$  que permiten formular las restricciones de no anticipatividad en la sección (2.2). Del mismo modo, las variables de decisión y restricciones presentan índices según escenarios, mientras que la función objetivo se modifica incluyendo el cálculo de la esperanza sobre el total de escenarios con que se está trabajando.

Así el modelo es el siguiente:

- **Conjuntos**

- $T$ : Horizonte de Planificación  $\{t\}$ .
- $O$ : Nodos orígenes  $\{o\}$ .
- $N$ : Nodos de Intersección  $\{n\}$ .
- $NO$ : Nodos  $\{no\}$ .
- $E$ : Nodos de Salida  $\{e\}$ .
- $H$ : Unidades de Cosecha  $\{h\}$ .
- $H_o$ : Orígenes  $\{o\}$ , Asociados a las unidades de cosecha  $\{h\}$ .
- $O_h$ : Unidades de cosecha  $\{h\}$ , asociadas a los orígenes  $\{o\}$ .
- $R^{DE}$ : Caminos Existentes de Tierra  $\{i, j\}$ .
- $R^{GE}$ : Caminos Existentes de grava  $\{i, j\}$ .
- $R^P$ : Caminos potenciales  $\{i, j\}$ .
- $RT = R^{DE} \cup R^{GE} \cup R^P$ : Conjunto de todos los caminos.
- $Y$ : Stocking yards  $\{y\}$ .
- $R$ : Tipo de camino  $\{r\}$ .
- $P$ : Productos  $\{p\}$ .
- $S$ : Escenarios  $\{s\}$ .

- **Parámetros**

- $A_h$ : Área de la unidad de cosecha  $h$  [Ha].
- $a_h^{p,t}$ : Productividad por [Ha] cosechada en la unidad  $h$  para el producto  $p$  en el periodo  $t$  [ $M^3$ ].
- $U_{i,j}^{r,t}$ : Capacidad del camino  $(i, j)$  tipo  $r$  en el periodo  $t$  [ $M^3$ ].
- $HC_h^t$ : Costo de cosechar una [Ha] de la unidad  $h$  en el periodo  $t$   $\left[ \frac{\text{USD}}{\text{Ha}} \right]$ .

$PC_o^t$ : Costo de producir un  $[M^3]$  en el origen o en el periodo t  $\left[\frac{USD}{M^3}\right]$ .

$RC_{i,j}^{r,t}$ : Costo de construir el camino  $(i, j)$  tipo r en el periodo t  $[USD]$ .

$TC_{i,j}^{r,t}$ : Costo de transportar un  $[M^3]$  en el camino  $(i, j)$  tipo r en el periodo t  $\left[\frac{USD}{M^3}\right]$ .

$UC_{i,j}^t$ : Costo de mejorar el camino  $(i, j)$  en el periodo t  $\left[\frac{USD}{M^3}\right]$ .

$SC_y^t$ : Costo de almacenamiento por  $[M^3]$  del producto p en el Stocking yard y durante el periodo t  $\left[\frac{USD}{M^3}\right]$ .

$CAPS_y$ : Capacidad de almacenamiento del Stocking yard y  $[M^3]$ .

### Parámetros que dependen de los escenarios:

$P_{e,p}^{t,s}$ : Precio de venta en la salida e durante el periodo t del producto p bajo el escenario s  $\left[\frac{USD}{M^3}\right]$ .

$\underline{Z}^{p,t,s}, \bar{Z}^{p,t,s}$ : Cotas de demanda bajo escenario s, periodo t para el producto p  $[M^3]$ .

### • Variables de Decisión

$E_h^{t,s} = \begin{cases} 1 & \text{Si la unidad h es cosechada en el periodo t} \\ & \text{bajo el escenario s} \\ 0 & \sim \end{cases}$

$W_{i,j,r}^{t,s} = \begin{cases} 1 & \text{Si camino } (i, j) \text{ tipo r se construye en el periodo t} \\ & \text{bajo el escenario s} \\ 0 & \sim \end{cases}$

$V_{i,j}^{t,s} = \begin{cases} 1 & \text{Si el camino } (i, j) \text{ se mejora en el periodo t} \\ & \text{bajo el escenario s} \\ 0 & \sim \end{cases}$

$X_h^{t,s}$  = Área de la unidad h cosechada en el periodo t bajo el escenario s  $[Ha]$ .

$Y_{o,p}^{t,s}$  = Total del producto p recogido en el origen o durante el periodo t bajo el escenario s  $[M^3]$ .

$S_{y,p}^{t,s}$  = Total del producto p almacenado en stocking yard y durante el periodo t bajo el escenario s  $[M^3]$ .

$f_{i,j,r}^{p,t,s}$  = Flujo del producto p transportado por el camino  $(i, j)$  de tipo r durante el periodo t, bajo el escenario s  $[M^3]$ .

$z_{p,e}^{t,s}$  = Cantidad del producto p vendido en el periodo t, en la salida e bajo el escenario t  $[M^3]$ .

- **Restricciones**

1. Cantidad total de producto recolectada por origen

$$\sum_{h \in H} a_h^{p,t} X_h^{t,s} = Y_{o,p}^{t,s} \quad \forall o \in O_h, p \in P, t \in T, s \in S$$

2. Límite de área cosechada por unidad

$$\sum_{t \in T} X_h^{t,s} - A_h \leq 0 \quad \forall h \in H, s \in S$$

3. Restricciones de Flujo

a) Flujo de Orígenes

$$Y_{o,p}^{t,s} + \sum_{\substack{(i,o) \in RT \\ r \in R}} f_{i,o,r}^{p,t,s} - \sum_{\substack{(o,j) \in RT \\ r \in R}} f_{o,j,r}^{p,t,s} = 0 \quad \forall o \in O, p \in P, t \in T, s \in S$$

b) Flujo de Intersecciones

$$\sum_{\substack{(i,n) \in RT \\ r \in R}} f_{i,n,r}^{p,t,s} - \sum_{\substack{(n,j) \in RT \\ r \in R}} f_{n,j,r}^{p,t,s} = 0 \quad \forall n \in N, p \in P, t \in T, s \in S$$

c) Flujo de Salida

$$\sum_{\substack{(i,e) \in RT \\ r \in R}} f_{i,e,r}^{p,t,s} - \sum_{\substack{(e,j) \in RT \\ r \in R}} f_{n,j,r}^{p,t,s} - z_{p,e}^{t,s} = 0 \quad \forall e \in E, p \in P, t \in T, s \in S$$

d) Flujo de Almacenamiento

$$\sum_{\substack{(i,y) \in RT \\ r \in R}} f_{i,y,r}^{p,t,s} - \sum_{\substack{(y,j) \in RT \\ r \in R}} f_{y,j,r}^{p,t,s} - S_{y,p}^{t,s} = 0 \quad \forall y \in Y, p \in P, t \in T, s \in S$$

4. Definición de Almacenamiento

$$S_{y,p}^{t,s} - S_{y,p}^{t-1,s} - \sum_{\substack{(i,y) \in RT \\ r \in R}} f_{i,y,r}^{p,t,s} = 0 \quad \forall y \in Y, p \in P, t \in T > 1, s \in S$$

5. Capacidad de Almacenamiento

$$\sum_{p \in P} S_{y,p}^{t,s} \leq CAPS_y \quad \forall y \in Y, t \in T, s \in S$$

6. Cotas (superior e inferior) de producción

$$\underline{Z}^{p,t,s} \leq \sum_{e \in E} z_{p,e}^{t,s} \leq \overline{Z}^{p,t,s} \quad \forall p \in P, t \in T, s \in S$$

7. Capacidad de los caminos

a) Caminos existentes (Grava)

$$\sum_{p \in P} f_{i,j,R2'}^{p,t,s} + \sum_{p \in P} f_{j,i,R2'}^{p,t,s} - U_{i,j}^{R2',t} \leq 0 \quad \forall (i,j) \in R^{GE}, t \in T, s \in S$$



b) Caminos existentes 2 (Grava)

$$f_{i,j,R1'}^{p,t,s} = 0 \quad \forall (i,j) \in R^{GE}, t \in T, s \in S$$

8. Construcción de caminos potenciales

a) Caminos potenciales (tierra)

$$\sum_{p \in P} f_{i,j,R1'}^{p,t,s} + \sum_{p \in P} f_{j,i,R1'}^{p,t,s} - U_{i,j}^{R1',t} \left( \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} W_{i,j,R1'}^{\alpha,s} - V_{i,j}^{\alpha,s} \right) \leq 0$$

$$\forall (i,j) \in R^P, t \in T, s \in S$$

b) Caminos Potenciales (grava)

$$\sum_{p \in P} f_{i,j,R2'}^{p,t,s} + \sum_{p \in P} f_{j,i,R2'}^{p,t,s} - U_{i,j}^{R2',t} \left( \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} W_{i,j,R2'}^{\alpha,s} + V_{i,j}^{\alpha,s} \right) \leq 0$$

$$\forall (i,j) \in R^P, t \in T, s \in S$$

c) A lo más se construye una vez cada camino potencial

$$\sum_{\substack{r \in R \\ \alpha \in T: T \text{ odd}}} W_{i,j,r}^{\alpha,s} \leq 1 \quad \forall (i,j) \in R^P, s \in S$$

9. Mejora de los caminos

a) Caminos Existentes (grava)

$$\sum_{p \in P} f_{i,j,R2'}^{p,t,s} + \sum_{p \in P} f_{j,i,R2'}^{p,t,s} - U_{i,j}^{R2',t} \left( \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} V_{i,j}^{\alpha,s} \right) \leq 0$$

$$\forall (i,j) \in R^{GE}, t \in T, s \in S$$

b) Caminos Existentes (tierra)

$$\sum_{p \in P} f_{i,j,R1'}^{p,t,s} + \sum_{p \in P} f_{j,i,R1'}^{p,t,s} - U_{i,j}^{R1',t} \left( 1 - \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} V_{i,j}^{\alpha,s} \right) \leq 0$$

$$\forall (i,j) \in R^{DE}, t \in T, s \in S$$

c) Mejora desde tierra

$$V_{i,j}^{t,s} - \left( 1 - \sum_{\alpha \in T: T \text{ odd}, \alpha \leq t} W_{i,j,R2'}^{\alpha,s} \right) \leq 0$$

$$\forall (i,j) \in R^P, t \in T: t \text{ odd}, s \in S$$

d) No se puede mejorar si el camino se construyó en dicho periodo

$$V_{i,j}^{t,s} - \left( \sum_{\alpha \in T: T \text{ odd}, \alpha < t} W_{i,j,R1'}^{\alpha,s} \right) \leq 0$$

$$\forall (i,j) \in R^P, t \in T: t \text{ odd}, s \in S$$

10. Capacidad de cosecha y relación de variables

$$X_h^{t,s} - A_h E_h^{t,s} \leq 0 \quad \forall h \in H, t \in T, s \in S$$

11. Naturaleza de las variables

$$X_h^{t,s}, Y_{o,p}^{t,s}, S_{y,p}^{t,s}, f_{i,j,r}^{p,t,s}, z_{p,e}^{t,s} \geq 0$$

$$\forall e \in E, (i,j) \in RT, t \in T, r \in R, y \in Y, p \in P, o \in O, s \in S$$

$$W_{i,j,r}^{t,s}, E_h^{t,s}, V_{I,j}^{t,s} \in \{0,1\}$$

$$\forall h \in H, (i,j) \in R^P, r \in R, t \in T: t \text{ odd}, s \in S$$

• **Función Objetivo**

$$\max \sum_{s \in S} p_s (T_1^s - T_2^s - T_3^s - T_4^s - T_5^s - T_6^s - T_7^s)$$

donde:

$$T_1^s = \sum_{e \in P} \sum_{p \in P} \sum_{t \in T} P_{e,p}^{t,s} z_{p,e}^{t,s} \quad \forall s \in S$$

$$T_2^s = \sum_{h \in H} \sum_{t \in T} HC_h^t X_h^{t,s} \quad \forall s \in S$$

$$T_3^s = \sum_{o \in O} \sum_{t \in T} \sum_{p \in P} PC_o^t Y_{o,p}^{t,s} \quad \forall s \in S$$

$$T_4^s = \sum_{(i,j) \in RT} \sum_{t \in T} \sum_{p \in P} \sum_{r \in R} TC_{i,j}^{r,t} f_{i,j,r}^{p,t,s} \quad \forall s \in S$$

$$T_5^s = \sum_{y \in Y} \sum_{t \in T} \sum_{p \in P} SC_y^t S_{y,p}^{t,s} \quad \forall s \in S$$

$$T_6^s = \sum_{(i,j) \in R^p} \sum_{t \in T: \text{odd}} \sum_{r \in R} RC_{i,j}^{r,t} W_{i,j,r}^{t,s} \quad \forall s \in S$$

$$T_7^s = \sum_{(i,j) \in R^p UR^{de}} \sum_{t \in T: \text{odd}} UC_{i,j}^t V_{i,j}^{t,s} \quad \forall s \in S$$

La función objetivo presenta su principal modificación en que cada uno de los términos que la componen ahora presentan un índice relacionado con el escenario en que se realizan, puesto que tanto las variables de decisión como las aleatorias y parámetros del modelo dependerán de los escenarios.

Finalmente, las restricciones de no anticipatividad, así como del resto de los elementos requeridos para formular explícitamente el modelo se encuentran en el capítulo 2.

**3.5.3 Fortalecimientos**

Los fortalecimientos corresponden a ciertas modificaciones y/o adiciones que se realizan sobre la formulación original que presentan como principal característica el hecho de que no dejan fuera ninguna solución factible, pero permiten mejorar la cota dual del problema. El

primer par de restricciones presentadas fueron desarrolladas por Badilla [21] y el segundo par pertenecen a Andalaft [1].

### Restricciones Adicionales

1.- Adicional al balance de flujo en los nodos de la red

Condición cero divergencia en la red, o también no hay inventario entre periodos

$$\sum_{h \in H} a_h^{p,t} E_h^{t,s} A_h = \sum_{e \in E} z_{p,e}^{t,s} \quad \forall h \in H, p \in P, t \in T, s \in S$$

En el trabajo de esta tesis, no es válido utilizar esta restricción a priori debido a que el modelo ahora incluye la posibilidad de almacenar productos en las zonas de acopio, por lo que no es posible utilizar este fortalecimiento sin conocer la solución óptima previamente, puesto que limita el espacio de soluciones factibles del problema en estudio.

2.- Capacidad de flujo de caminos existentes

$$f_{i,j,r}^{p,t,s} \leq \sum_{o \in O} \sum_{h \in H_o} a_h^{p,t} E_h^{t,s} A_h \quad \forall (i,j) \in R^{DE} \cup R^{GE}, p \in P, t \in T, s \in S$$

Este fortalecimiento se ha reformulado en el actual trabajo, utilizando las nuevas restricciones de capacidad señaladas en la sección anterior, determinando cotas de flujo consistentes para cada camino existente.

3.- No se construyen caminos aislados (Road-to-triggers, ver Andalaft [1])

$$W_{k,l,r}^{t,s} \leq \sum_{\theta \leq t} \sum_{(i,j) \in N(k,l)} W_{i,j,r}^{\theta,s} \quad \forall (k,l) \in R^P \setminus SC, r \in R, t \in T, s \in S$$

SC = {Caso especial de un camino potencial que conecta un origen con un camino existente}

$$= \{(i,j) \in R^P / [(i, \cdot) \vee (\cdot, i) \in R^{DE} \cup R^{GE} \wedge j \in O] \vee [(j, \cdot) \vee (\cdot, j) \in R^{DE} \cup R^{GE} \wedge i \in O]\}$$

$$N(k,l) = \{\text{Caminos potenciales que conectan con } (k,l)\} \\ = \{(i,j) \in R^P / i = k, l \vee j = l, k\}$$

Este fortalecimiento no fue utilizado en el presente trabajo de tesis debido a la baja de rendimiento que se experimentó al incluirlo.

4.- No se cosechan unidades aisladas (Lot-to-road triggers, ver Andalaft [2])

$$E_h^{t,s} \leq \sum_{\theta \leq t} \sum_{(i,j) \in N(h)} W_{i,j,r}^{\theta,s} \quad \forall h \text{ no conectado}, r \in R, t \in T, s \in S$$

$$\begin{aligned}
N(h) &= \{\text{Caminos potenciales que acceden a unidad } h\} \\
&= \{(i, j) \in R^P / i \in O_h \vee j \in O_h\}
\end{aligned}$$

Al igual que el caso anterior, esta formulación no fue utilizada.

### Lifting

La técnica de lifting (levantamiento o giro de las restricciones) consiste en utilizar una sumatoria de caracter temporal de todos los periodos anteriores al analizado, de tal manera que se disminuye significativamente la cantidad de restricciones del modelo original. Para realizarlo, basta agregar el término  $\sum_{\theta \leq t} \text{Expresión}$  al lado izquierdo de la ecuación. Estos fueron formulados por Andalaft [1].

1. Capacidad de flujo de camino levantado

$$\sum_{p \in P} \sum_{\theta \leq t} f_{i,j,r}^{p,\theta,s} \leq U_{i,j}^{r,t} \sum_{1 \leq \varphi \leq t} W_{i,j,r}^{\varphi,s} \quad \forall (i, j) \in R^P, r \in R, t \in T, s \in S$$

Esta restricción persigue ajustar aún más el modelo, al restringir de manera bastante significativa el flujo total que puede experimentar cada uno de los caminos de la red interna. Se ha reformulado a través de las restricciones modificadas con respecto al modelo original.

2. Capacidad de flujo de caminos existentes

$$\sum_{\theta \leq t} f_{i,j,r}^{p,\theta,s} \leq U_{i,j}^{r,t} \quad \forall (i, j) \in R^{DE} \cup R^{GE}, t \in T, r \in R, s \in S$$

Importante es notar que los lifting 1 y 2 no serían válidos en caso de que la cota fuese impuesta en vez de calculada.

3. No se construyen caminos aislados

$$\sum_{\theta \leq t} W_{k,l,r}^{\theta,s} \leq \sum_{\theta \leq t} \sum_{(i,j) \in N(k,l)} W_{i,j,r}^{\theta,s} \quad \forall (k, l) \in R^P \setminus SC, r \in R, s \in S$$

4. No se cosechan rodales aislados

$$\sum_{\theta \leq t} E_h^{\theta,s} \leq \sum_{\theta \leq t} \sum_{(i,j) \in N(h)} W_{i,j,r}^{t,s} \quad \forall h \text{ no conectado}, r \in R, s \in S$$

## 3.6 Instancias

La instancia utilizada en los experimentos realizados a lo largo de este trabajo de tesis corresponde al bosque de la empresa forestal “Forestal Millalemu”, consistente en 17 bosques que conforman la totalidad de los terrenos pertenecientes a dicha empresa. Los escenarios a utilizar durante los experimentos de la formulación estocástica son obtenidos directamente desde el programa desarrollado en Matlab (ver sección (3.4)), utilizándose como precio raíz el valor de cada uno de los productos en el año 2009.

Se desarrollaron una serie de instancias cuya única diferencia en tamaño se relaciona con el total de escenarios que se incluyen en la formulación estocástica, siendo el resto de los parámetros y variables equivalentes a la formulación determinística.

### 3.6.1 Componentes determinísticos

La instancia original del bosque de la empresa Forestal Millalemu se caracteriza por contener:

- 4 periodos
- 118 unidades de cosecha
- 290 caminos (39 potenciales más 251 existentes)
- 223 nodos en total (118 orígenes, 95 intersecciones)
- 3 tipos de productos
- 2 tipos de camino (tierra y grava)
- Posibilidad de mejorar los caminos durante los periodos estivales
- 3 centros de almacenamiento de productos entre periodos
- 7 salidas

Cada uno de estos elementos sigue las mismas características señaladas en la figura 3.10, siendo en este caso la única diferencia presente el hecho de que existen más bosques contiguos. El esquema completo del bosque se presenta a continuación:

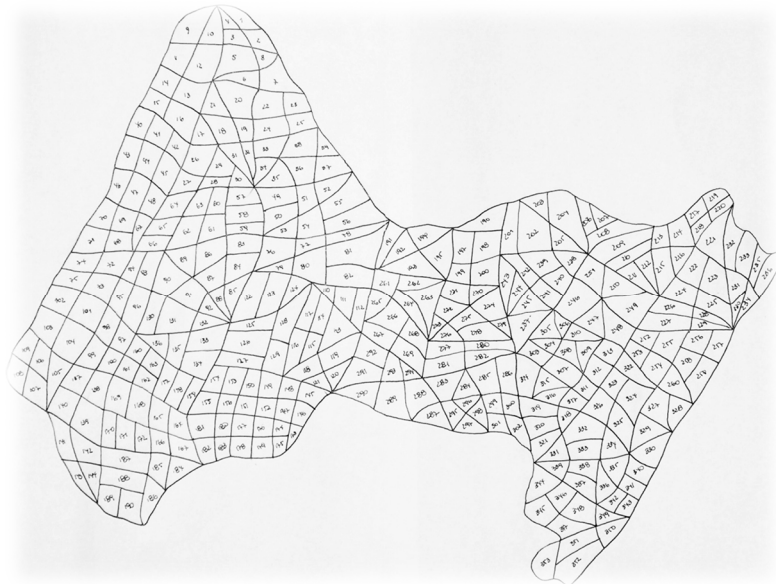


Figura 3.10: Diagrama de la instancia Millalemu con los 17 bosques

### 3.6.2 Componentes Estocásticos

Los componentes estocásticos que se abordan dentro del problema trabajado corresponden a dos tipos principales: Mercado (precio de cada uno de los productos así como la demanda mínima a experimentar) y el rendimiento de cada uno de los bosques y unidades a la hora de producir los diferentes productos que se requieran.

En relación a la incertidumbre planteada con respecto a los precios, ésta fue agregada al problema a través de la metodología planteada en la sección (3.1), obteniéndose así una serie de árboles de escenarios (de diversos tamaños) donde en cada nodo se presenta un triplete que representa el valor de cada uno de los productos en un determinado periodo en estudio. Por otra parte, el mínimo de demanda por cada uno de estos elementos se estipula como un parámetro aleatorio que se encuentra ligado al comportamiento del precio, en base a información histórica, dada la correlación inherente entre ambos elementos.

La incertidumbre relacionada a los rendimientos de los rodales en su etapa adulta sigue el mismo esquema de trabajo desarrollado por Badilla [21], realizándose un camino aleatorio que determina que cada bosque normalmente crece un 3% anual con una desviación de un 5% según el clima que experimente. Junto con lo anterior, cabe hacer notar que las distribuciones han sido winsorizadas, lo que implica que si al simular se obtiene un valor superior o inferior a alguna de las cotas (sobre o bajo un 5%), éste será reemplazado por la cota más cercana, hecho que va de la mano con el limitado efecto climático que se presenta (Navarrete 2009 [22]).

Todas estas fuentes de incertidumbre se combinan en un único árbol de escenarios que contiene toda la información necesaria para llevar a cabo la resolución del modelo. De esta forma, el cálculo del total de escenarios a trabajar en una instancia será la multiplicación directa del número de nodos hijos por periodo que se generen.

En base a todo lo anterior, la rutina descrita en la sección (3.1) genera de forma automática y completamente eficiente todas las instancias y sus archivos asociados, listos para llevar a cabo la llamada a los software de optimización que corresponda o directamente al solver de preferencia. Toda la información relativa a los bosques (camino, nodos, salidas, entre otros) se encuentra embebida dentro de la rutina programada, por ende se obtiene la creación de cada una de las instancias en segundos y sin la necesidad de monitoreo.

### 3.6.3 Tamaño

Debido al gran tamaño de la instancia básica de trabajo, se aprecia que el tamaño del problema determinístico supera las 12 mil variables, que si bien no es un número despreciable, es perfectamente abordable por computadores de escritorio comunes, sin embargo, en la medida que se aumenta el número de escenarios a usar, el crecimiento de variables es explosivo, siendo imposible de resolver en un computador de escritorio una instancia que contenga más de 200 escenarios, presentándose en el caso de 1000 escenarios más de 3.4 millones de variables, así como millones de restricciones.

Del mismo modo, aun siendo un problema abordable con un número disminuido de escenarios, esto no implica que sea simple de resolver, puesto que CPLEX muchas veces no logra obtener una solución con un gap menor a 2% en tiempos menores a 1 hora, lo que da cuenta de la dificultad del problema en estudio.

Nº Escenarios	Variables Continuas	Variables Binarias	Rows	Non-Zeros
1	11946	798	5415	44379
2	19910	1256	9526	80313
3	23892	1596	12333	107802
5	39820	2512	20555	179670
10	59730	4212	34590	317115
20	99550	7612	62660	592005
30	115478	8750	93338	883785
40	119460	9090	96610	999465
50	147334	11248	112242	1189950
107	359927	20406	188017	2206009
110	395920	22447	206819	2426610
216	700941	38478	366680	4369041
343	1082572	57861	566932	6838429
512	1451032	82836	795793	10099437
729	2176548	124254	1193690	15149156
1000	3264822	186381	1790534	22723733

Figura 3.11: Tamaño de las instancias experimentales

## Capítulo 4

# Algoritmo Progressive Hedging

En este capítulo se abordará el algoritmo de Progressive Hedging en profundidad, observando su comportamiento práctico a la hora de llevarse a cabo la resolución del problema forestal completo, así como definir y caracterizar los ajustes y mejoras que se realizaron durante el trabajo de tesis para mejorar el rendimiento del mismo, además de servir como guía para abordar problemas estocásticos enteros de similares características. Más referencias a la teoría subyacente del algoritmo se encuentran en la sección (2.3).

Lo que se pretende lograr a través de la serie de ajustes y técnicas que se mencionan en esta sección corresponde a acelerar la velocidad de resolución del algoritmo, tratando de no perjudicar de gran forma sobre la calidad de la solución misma. Dentro de los métodos más utilizados para mejorar la velocidad de resolución se encuentra la técnica de fijar variables, con tal de luego resolver la formulación extendida compacta de forma más eficiente.

Debido a la presencia de variables enteras dentro del problema en estudio, y recordando que el algoritmo originalmente fue concebido para tratar problemas convexos, es que Progressive Hedging se comporta como una heurística, por lo que no es necesario establecer pasos y condiciones matemáticas rigurosas para llevar a cabo su desarrollo, por ende, se realiza una investigación desde una perspectiva de aplicabilidad más que teórica.

### 4.1 Pseudo-código extendido

Con el objetivo de formalizar el marco teórico expuesto en la sección (2.3), se presenta una versión extendida del pseudo-código que sigue el algoritmo a lo largo de cada una de sus realizaciones, incluyendo detalles característicos de la implementación que fue realizada.

---

#### Algoritmo Progressive Hedging Extendido

---

1. *Inicializar*:  $\varepsilon$  tolerancia;
2.  $k := 0, g^* := \infty$ ;
3.  $\forall s \in S \quad x_s^k := \operatorname{argmin}_{x_s} f_s(x_s) : x_s \in Q_s$ ;
4.  $k := k + 1$ ;
5.  $\forall t \in T, \forall N_t \in N \quad \bar{x}_{n(s,t),t}^k := \frac{1}{|N_t|} \sum_{s \in N_t} x_{t,s}^k$ ;
6.  $g^k := \sum_{s \in S, t \in T} \|x_{t,s}^k - \bar{x}_{n(s,t),t}^k\|$ ;
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.  $\forall s \in S, t \in T \quad w_{s,t}^k := \rho(x_{t,s}^k - \bar{x}_{n(s,t),t}^k) + w_{s,t}^{k-1}, (w^{(0)} = \vec{0});$
  11.  $\forall s \in S \quad x_s^k := \operatorname{argmin}_{x_s} f_s(x_s) + \sum_{s \in S, t \in T} [w_{s,t}^k x_{s,t} + \frac{\rho}{2} \|x_{t,s} - \bar{x}_{n(s,t),t}^k\|^2]: x_s \in Q_s;$
  12. Ir a 4;
  13. Usando  $x^*$  como hotstart con las mismas variables fijadas, resolver  
la Formulación extendida compacta:  $\min_x \sum_{s \in S} f_s(x_s): (x \in Q);$
- 

A partir del pseudo-código anterior se observa:

- En **(5)** se presentan elementos con la misma probabilidad, por ende, los pesos son idénticos.
- En el **paso número (6)** se lleva a cabo el cálculo de la métrica de convergencia del algoritmo, la cual corresponde a la distancia euclidiana entre cada uno de los escenarios con respecto a su óptimo estimado en el nodo correspondiente. Junto a lo anterior, se introduce la función  $n: S \times T \rightarrow N$ , función que se encarga de ubicar a cada uno de los escenarios con su nodo correspondiente dentro de cada una de las etapas. De esta forma, se tiene una representación simple y alternativa a la estructura de árbol original con padres e hijos. Del mismo modo, se utiliza  $n$  como subíndice para los promedios que presenta un paquete de información determinado.
- En **(7)** se presenta una nueva familia de variables que se encuentran superindexadas por  $*$ , indicando que éstas son la mejor solución que se ha obtenido hasta dicho momento por el algoritmo. Esta situación aparece debido a – cómo se mencionó al inicio de esta sección – la naturaleza heurística que presenta el algoritmo al abordar problemas que incluyen variables enteras, y por ende, puede ser que la solución no siempre converja o mejore a lo largo de las iteraciones, “saltando” de solución en solución (no se comporta de forma estable).
- En **(9)** se caracteriza el penalizador  $\rho$  como un elemento que es función dependiente de las 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  $\rho$  utilizada, aspecto que se trata con profundidad en la subsección (4.2.4).
- En el **paso (14)**, debido a que el problema que se está resolviendo es degenerado, posiblemente alcanzar el valor óptimo (cero gap) tome demasiado tiempo, por lo que se establece un nivel de gap determinado (0.001%) y un límite de tiempo asociado en caso de no lograr avanzar.

- Los **pasos (2) y (4)** consisten en la inicialización y actualización de los elementos de la iteración, mientras que posterior al **paso (6)** corresponde al ajuste de fijar variables entre iteraciones, lo que será tratado en la subsección (4.2.8).

## 4.2 Ajustes y mejoras

### 4.2.1 Acelerar 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:

- **Fortalecimientos del modelo determinístico (también llamado subproblema)**  
Se utilizaron los fortalecimientos señalados en la subsección (3.5.3) con tal de comparar las soluciones gap-óptimas que arrojaba el modelo con cada una de las combinaciones posibles entre estos.
- **Hotstars (también llamadas warmstarts)**  
Esta técnica consiste en la utilización de soluciones previas como punto de partida en la iteración actual, de tal forma de tener un punto de partida válido y cercano al óptimo, lo que tiende a generar un ahorro significativo en cuanto a los tiempos de resolución involucrados.
- **Ajuste de Gap según iteraciones**  
Esta técnica consiste en trabajar con un gap variable según la iteración del algoritmo en la cual se encuentre, puesto que debido a que se persigue la convergencia de la solución, no es necesariamente útil comenzar a resolver en la primera iteración con un gap bajo (o nulo), por lo que generar ajustes con respecto a la tolerancia al óptimo que se puede tener para avanzar a la siguiente iteración será muy útil para alcanzar soluciones de gran calidad en menor tiempo.

La metodología utilizada para llevar a cabo las comparaciones entre las distintas combinaciones que se pueden generar con estas técnicas consistió en utilizar el algoritmo PH por el total de iteraciones que fuesen necesarias para alcanzar convergencia, a un gap de 0.05%, utilizando un factor de corte (threshold) de 0.01 y valores idénticos (según la combinación usada) para los otros parámetros. Las combinaciones utilizadas fueron:

- Sin hotstart, sin gap ajustable y sin factor de penalización
- Con hotstart, sin gap ajustable y con penalización
- Con hotstart obtenido de PH, gap ajustable y penalización.

La mejor combinación obtenida corresponde a la tercera mencionada, utilizando además la formulación implícita de los fortalecimientos señalada en la subsección (3.5.3).

En relación al uso de hotstart, esta es una estrategia ampliamente conocida y popular dentro de las disciplinas de investigación y gestión de operaciones, teniéndose una amplia documentación al respecto [6]. Para el problema en estudio dicha lógica se confirma, puesto que se obtienen mejoras del orden del 50% en la mayoría de las instancias experimentales al realizar la comparación con el caso donde esta técnica no es utilizada.

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 tomar horas y horas de resolución, sin necesariamente ser significativamente mejor (en términos de valor) a una solución que puede ser encontrada en términos de segundos. La estrategia utilizada para aplicar esta técnica consiste en comenzar con un gap elevado en las primeras iteraciones, de tal forma que cada uno de los subproblemas tome pocos segundos en ser resuelto, sin embargo, se debe tener cuidado de no tomar un gap muy alto, pues se buscan soluciones coherentes 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, por ende a través de una serie de experimentos de repetición, se encontró que un gap cercano al 20% sería el óptimo para comenzar a resolver subproblemas.

Luego, una vez calculadas dichas soluciones, en la siguiente iteración se disminuye el tamaño del gap, de forma progresiva hasta la última iteración. La idea de esta estrategia es lograr alcanzar soluciones cada vez más cercanas al óptimo global de la formulación extendida, evitando al mismo tiempo el quedar “estancados” en una solución factible por muchas iteraciones, puesto que en problemas altamente combinatoriales como el que se aborda en esta tesis, el usar un mismo gap de forma constante genera muchas soluciones similares, por ende si el gap no fuese modificado con cada nueva iteración, el software de optimización simplemente comprobaría que la solución es factible y se quedaría atrapado en el óptimo local, sin mover la solución según el factor de penalización. Mediante una serie de ensayos, se determinó trabajar con tres gaps a alcanzar finalmente: 2%, 1% y 0.05%, debido a que cumplen con las condiciones necesarias para llevar a cabo una mejora satisfactoria en los tiempos de resolución involucrados.

#### **4.2.2 Paralelizar el algoritmo**

Debido a sus características intrínsecas a la hora de tratar el problema original y generar una descomposición de éste según escenarios (es decir, descomposición horizontal de la instancia original) es que el algoritmo de Progressive Hedging presenta una estructura natural para llevar a cabo programación paralela y distributiva, enviando cada uno de los subproblemas (pequeños y de veloz resolución) a un núcleo o nodo de un clúster determinado, obtener la solución de éste y luego hacer concurrir las soluciones para actualizar el estado de los parámetros del algoritmo y continuar a la siguiente iteración.

Su implementación computacional para el algoritmo PH se encuentra en el software PySP, parte del paquete de software Coopr de los laboratorios Nacionales Sandia [8], pudiéndose así realizar una serie de experimentos en relación al algoritmo y el número de computadores/procesadores que participan para su resolución. Los resultados obtenidos van de la mano con lo esperado, puesto que los tiempos de resolución de las instancias experimentales mejoraron de sobremanera (hasta un 70%) en comparación al caso serial y el de formulación extendida, acentuándose de forma significativa en la medida que el número de escenarios a tratar aumenta, lo que es completamente lógico y consistente con la teoría, puesto que al llevar a cabo programación distributiva, cada procesador o nodo se encarga de resolver problemas individuales muy simples, por lo que no hay un gran tiempo de carga o resolución, mientras que la formulación extendida debe cargar con todo desde un comienzo.

Uno de los inconvenientes que se debe enfrentar a la hora de llevar a cabo programación en paralelo, corresponde al “overhead” o tiempo de inicialización y comunicación que se debe invertir a la hora de llevar a cabo la interacción entre los diferentes núcleos o nodos que se encuentran resolviendo el problema, principalmente a la hora de coordinar, enviar y recibir los problemas y sus soluciones, por lo que la utilización de esta técnica para instancias menores a los 50 escenarios no presenta ninguna utilidad, tal como puede verse en el capítulo 6 de resultados.

El esquema de trabajo utilizado por esta metodología consiste en la presencia de un computador maestro dentro del cual se encuentran los modelos e instancias a tratar, el cual actúa simplemente como monitor de una serie de computadores (o clúster computacional), que se encargan de resolver cada uno de los subproblemas que éste envía. El computador maestro establece un servidor virtual que se encarga de gestionar la conexión con el resto de los “solucionadores” a través de un servidor de despacho, que se encarga de enviar y recibir los subproblemas y sus respectivas soluciones para poder avanzar a través de las iteraciones. Cada uno de los nodos o procesadores establece conexión a este servidor virtual a través de un servidor de resolución (mip server), que presenta una dirección única e inequívoca con dicho recurso computacional. Todo lo anteriormente mencionado fue implementado en una serie de scripts programados en python y ms-dos, complementarios al paquete PySP, de tal forma de poder utilizar esta técnica de forma automática y directa, simplemente determinando de forma previa el número de procesadores/computadores que se pretenden utilizar.

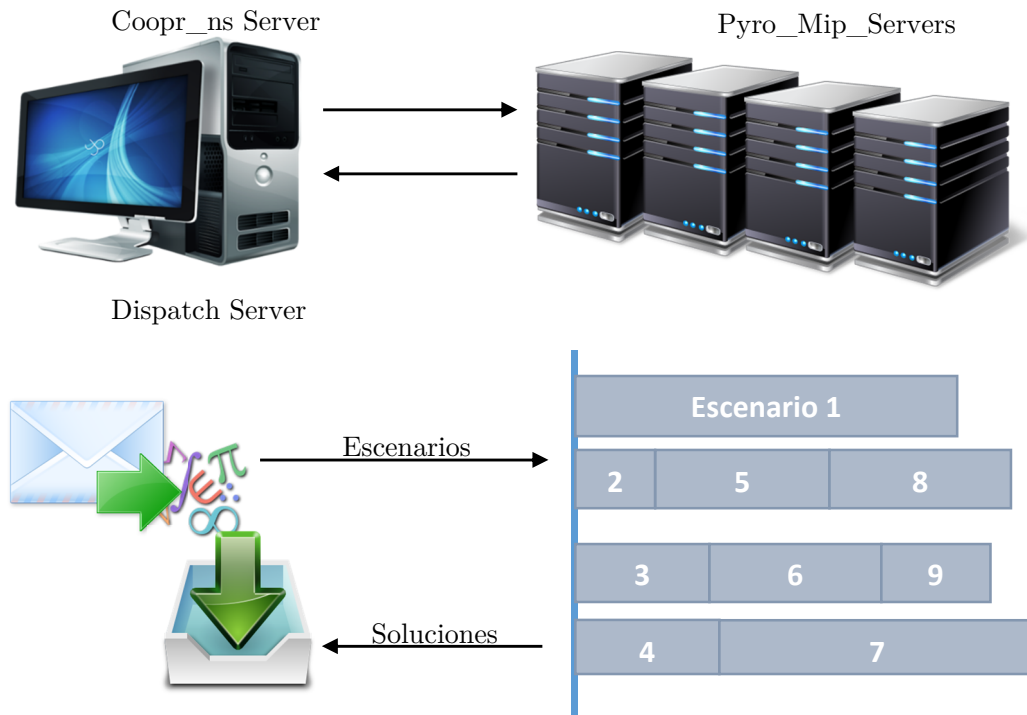


Figura 4.1: Esquema de programación en paralelo utilizando el Algoritmo PH y el paquete PySP de Coopr Sandia

Importante es notar que el script utilizado busca distribuir la carga de trabajo sobre cada uno de los procesadores o nodos “solucionadores” de la forma más equitativa posible, de tal modo que en promedio se tengan todas las soluciones de los escenarios dentro de una ventana de tiempo establecida.

### 4.2.3 Linealizar el término cuadrático proximal

La mayor desventaja que presenta el algoritmo de Progressive Hedging consiste en que es necesario agregar el término proximal cuadrático de penalización al resolver cada uno de los subproblemas, lo que implica que aunque estos originalmente sean pequeños y simples de resolver, terminen siendo problemas complejos (notar que esto se observa en el paso 11 del pseudo-código extendido de la sección (4.1)). Debido a esto, el tiempo de resolución de cada uno de los subproblemas puede verse afectado de forma negativa en cantidades significativas para la obtención de soluciones dentro de tiempos abordables.

El término señalado es necesario para alcanzar la convergencia del algoritmo, pues genera que cada una de las variables por cada escenario se acerque (sea “empujada”) hacia el promedio del nodo respectivo, permitiendo así cumplir con las restricciones de no anticipatividad, necesarias para el problema.

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

Del mismo modo, la presencia de este término es vital a la hora de evitar que los subproblemas en resolución se vuelvan no acotados, debido a la presencia del término  $wx$ , el cual representa las restricciones de no anticipatividad y que puede presentar ambos signos (negativo o positivo), lo que permite que con uno de estos la variable respectiva crezca indefinidamente dependiendo del criterio utilizado en la función objetivo (maximización o minimización) en caso de ser una variable no acotada. Sin embargo, es claro que éste corresponde a un término ficticio que no presenta un significado real e inherente al problema en estudio y que además, al tratarse con variables binarias pierde aún más su significado original (carece de sentido), por ende, dentro del contexto heurístico en el cuál se inmerge el uso del algoritmo para esta tesis, en caso de ser posible reemplazar este término por una expresión de efecto similar, pero con menores implicaciones en los tiempos de resolución involucrados, será totalmente útil.

A partir de esta conjetura es que nace la opción y necesidad de tratar la expresión original. Observando la expresión, es fácil comprender que en ésta se presentan multiplicaciones de variables por sí mismas (términos cuadráticos), pero no existen términos cruzados, lo que da pie a la utilización simple y directa de una aproximación lineal por tramos (conocida como piece-wise linearization en inglés), como se puede observar en la figura 4.2.

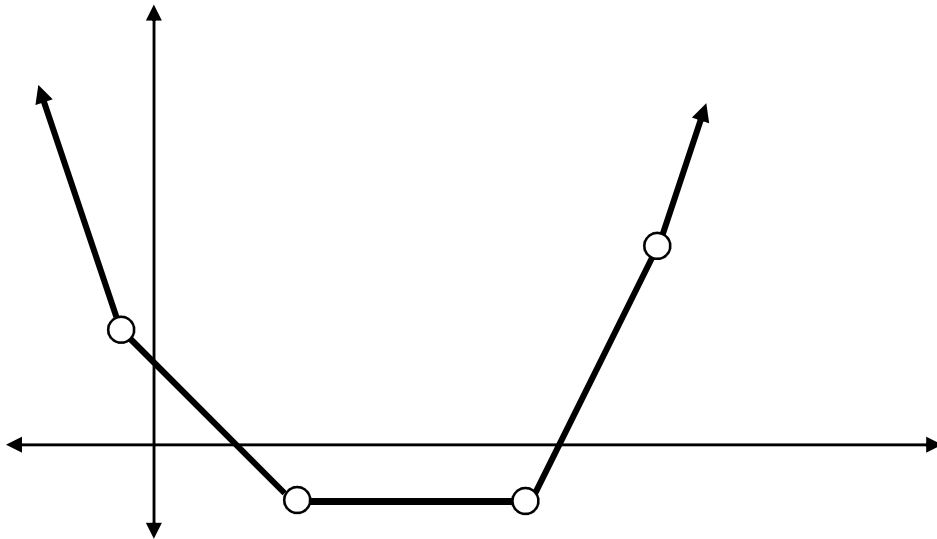


Figura 4.2: Diagrama de linealización por tramos

Este tipo de linealización toma en consideración la necesidad e importancia de actualizar cada una de las pendientes y puntos de quiebre de la linealización para cada uno de los escenarios en cada iteración del algoritmo, puesto que el punto más bajo de la curva representa en efecto el promedio  $\bar{x}$ , el cuál puede experimentar variaciones en todas las iteraciones (la solución puede no ser estable de una iteración a la otra, sobre todo si se utilizan elevados gaps

a la hora de abordar el problema). Del mismo modo, es importante tomar en consideración el hecho de que las variables linealizadas presentan cotas apropiadas, debido a que las pendientes que son calculadas solamente acotan el problema de forma finita, no así la curva original  $\rho(x - \bar{x})^2$ , cuya formulación implica que siempre puede ser mayor que  $wx$  en el caso de que  $x \rightarrow \pm\infty$ , lo que implica que una determinada variable no acotada originalmente, termine transformando el problema en uno no acotado.

El uso de esta técnica es bastante sensible al número de segmentos escogidos para llevar a cabo la linealización de la función original. Considerando que cada segmento de la función original corresponde a una nueva variable con una restricción asociada, es lógico concluir que el número de segmentos es proporcional al número de vértices del poliedro original que representa el subproblema, en aquellos casos donde la penalización definida determina una cara del mismo. Por ende, al tener un número reducido de segmentos, siendo dos el menor posible, la penalización tiende a adoptar la forma de un embudo o uve, con forma de V, lo que genera que todas las variables se vean forzadas a caer al promedio, lo que acarrea una convergencia precoz – con consecuencias negativas en el valor de la función objetivo – además de una convergencia inestable, puesto que los promedios cambian durante cada iteración, lo que lleva a generar saltos en los valores obtenidos entre una iteración y otra. Por otra parte, si se utilizan muchos segmentos, el problema nuevamente se vuelve complejo de resolver, pues se estaría acercando cada vez más al caso original cuadrático.

En base a lo anterior, es lógico notar que existe un trade-off entre el número de tramos a elegir para realizar la linealización, y por ende velocidad de resolución de cada subproblema, y la calidad de la solución obtenida. Sin embargo, esta situación puede contrarrestarse a partir de elegir un tamaño adecuadamente pequeño para el penalizador  $\rho$ , elemento que equipara el efecto negativo de elegir una cantidad disminuida de segmentos lineales dentro de la aproximación, ya que esto permite que muy pocas variables se vean forzadas a converger en cada iteración, pero muy velozmente, sin empeorar así el valor de la función objetivo obtenida.

Complementario a todo lo anteriormente señalado, se observa en la siguiente expresión la linearización trivial de un único segmento aplicada sobre variables binarias, puesto que carece de sentido el perseguir un promedio fraccionario para éstas. La operación matemática utilizada es bastante simple, consistente en realizar la clásica aproximación a partir de la primera derivada de la función original:

$$x^2 - 2x\bar{x} + \bar{x}^2 \rightarrow 2x - 2x\bar{x} + \bar{x}^2$$

#### 4.2.4 Ajustar el parámetro $\rho$ de penalización

El parámetro  $\rho$  presente en el algoritmo PH (observar (2.3.2) paso 4 y/o (4.1) paso 11) corresponde al ponderador relativo entre el cumplimiento del término proximal y el forzar la no anticipatividad de las variables del subproblema modificado, dentro de la función objetivo (ver (2.3.2) paso 5 o (4.1) paso 13). Dentro de los parámetros utilizados para llevar a cabo el algoritmo, es uno de los más sensibles y determinantes en el resultado que arroje el mismo, tal

como fue documentado por Mulvey y Vladimirov a finales de los 80's [6]. Del mismo modo, a partir de cómo se construye el ponderador  $w$ , el valor de  $\rho$  puede interpretarse como el precio fijado que se debe abordar por desviarse de la solución esperada, así,  $w$  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, y por ende válida.

A partir de esto, es lógico tener en consideración la premisa de mejora del rendimiento del algoritmo al modificar este parámetro, en caso de poder encontrar un mejor sistema de precios que permita generar una relación más estable y de mejor calidad entre las variables, así el algoritmo convergerá de forma más veloz y con un rendimiento superior en términos de función objetivo y calidad. El origen de estas estrategias testeadas proviene del estudio publicado por Watson y Woodruff en [10].

Las estrategias testeadas y estudiadas durante el transcurso del presente trabajo consisten en tratar con un parámetro  $\rho$  Fijo (FIX), CP, SEP, Taylor y una serie de combinaciones de las anteriormente mencionadas. Las tres primeras estrategias son expuestas en [10], mientras que la estrategia de Taylor fue desarrollada por Fernando Badilla, ex alumno del Magíster en Gestión de Operaciones. Junto a lo anterior, el método para seleccionar el valor de  $\rho$  se desarrolla bajo el anhelo de evitar una convergencia precoz de todas las variables producto de la linealización extrema en forma de embudo (sólo dos tramos), la rapidez de su cómputo en cada iteración y la velocidad de resolución del problema completo.

En base a lo anteriormente planteado, los requisitos expuestos fueron alcanzados al seleccionar un valor del parámetro  $\rho$  fijo, con valores bastante pequeños en la medida que las instancias no contengan una gran cantidad de escenarios, para luego aumentar en órdenes de potencias de 10 en la medida que se trabaja con árboles de escenarios más masivos.

N° Escenarios	Valor $\rho$	N° Escenarios	Valor $\rho$
1	0.0000001	50	0.001
2	0.0000001	107	0.001
3	0.0000001	110	0.001
5	0.00001	216	0.001
10	0.00001	343	0.001
20	0.0001	512	0.001
30	0.0001	729	0.001
40	0.0001	1000	0.001

Figura 4.3: Valor del parámetro Rho según instancia



Al observar el cuadro precedente, es posible notar que el tamaño del parámetro  $\rho$  óptimo crece en conjunto con el número de escenarios, lo que se explica debido a que en la medida que aumenta la cantidad de escenarios en uso, existe una mayor discrepancia entre los valores que estos presentan, lo que claramente dificulta la convergencia del algoritmo, disminuyendo el número de variables fijadas por cada iteración. Por ende, se aumenta el valor del parámetro  $\rho$ , contrarrestando así este efecto, lo que se traduce en un menor número de iteraciones, con el correspondiente ahorro en tiempo.

#### 4.2.5 Criterio de Término

Al observar el algoritmo extendido en la sección (4.1) paso número 8, se observa que se utiliza el alcanzar un mínimo de tolerancia (caracterizada por el parámetro  $\varepsilon$ ) como punto de corte para determinar que el algoritmo ha convergido, lo que se traduce en que todos los escenarios en estudio se parecen suficientemente al óptimo correspondiente del árbol que está siendo procesado. Sin embargo, esto en la práctica presenta una serie de inconvenientes que conllevan a que el método presente una convergencia muy lenta, tomando mucho tiempo en resolver una instancia y, al mismo tiempo, el valor de la función objetivo comienza a empeorar en la medida que se fuerce la convergencia. A partir de esto es que se establece como criterio alternativo de término el alcanzar un determinado número de iteraciones del algoritmo, de tal manera que se alcancen a fijar suficientes variables y se pueda resolver la formulación extendida compacta con variables fijadas de manera eficiente.

A partir de una serie de experimentos con una gran cantidad de instancias oscilantes entre 1 y 1000 escenarios, se obtuvo un valor para  $\varepsilon = 0.01$ , mientras que el total de iteraciones se estableció en  $k^{max} = 10$ . Del mismo modo, se establece como criterio alternativo el terminar cuando todas las variables binarias alcancen la convergencia, pues así el problema se transforma en un simple problema lineal, sin embargo, dicha condición no fue alcanzada en un número significativo de las instancias testeadas.

#### 4.2.6 Clustering y reducción de Escenarios

Uno de los desafíos más importantes con los que se debe lidiar a la hora de abordar problemas de programación estocástica corresponde a la gran dimensionalidad de los árboles de escenarios con los que se está trabajando, esto debido a que se busca representar la realidad de la forma más fiel posible, por lo que se hace necesario tomar en consideración un sinnúmero de posibilidades. Sin embargo, es posible que no sea necesario e imperativo el utilizar todo el conjunto original de escenarios definido por el árbol confeccionado, sino que tomando un subconjunto representativo de éste sea posible mantener en gran medida la información y/o distribución de la data original, obteniéndose así una solución de gran calidad en un menor tiempo de resolución y que requiere una menor cantidad de recursos computacionales para ser obtenida.

Así, es claro que el subconjunto de escenarios seleccionado deberá contar con una serie de requisitos esenciales, de tal forma que la solución obtenida tenga validez teórica y sea comparable a la que se obtendría bajo la instancia original. Estos requisitos son:

- Conservar los escenarios extremos o eventos cola (black swans) dentro de las posibles realizaciones de las variables aleatorias consideradas.
- Encontrar soluciones válidas en tiempos razonables.
- Representar de forma satisfactoria la estocasticidad del problema original, según alguna métrica útil.

La primera alternativa que surge de forma natural consiste en la confección de Clusters, pues es una aplicación sencilla y que ofrece una primera aproximación natural al problema de generar un subconjunto representativo de un set de datos original. La idea consiste en generar una serie de clusters o grupos que en la práctica representan un conjunto de escenarios que comparten características similares en cuanto al comportamiento de sus variables aleatorias, logrando así resumir la información contenida en un gran número de escenarios en un solo vector. Sin embargo, uno de los problemas que presenta esta técnica de reducción de escenarios consiste en que presenta un sesgo a mantener escenarios promedios, dejando de lado eventos extremos (como black swans) donde los precios puedan dispararse o caer abruptamente, lo que pese a tener una baja probabilidad de ocurrencia, puede llevar a grandes pérdidas si es que no son tomados en cuenta a la hora de procesar el modelo.

Con tal de hacer frente a la situación mencionada, es posible forzar al algoritmo a mantener un determinado número de eventos extremos de la instancia original, conservando una baja probabilidad de ocurrencia.

Del mismo modo, es preciso señalar que para mantener la validez de los escenarios generados, no se debe utilizar el valor de un “escenario promedio” como típicamente se realiza al utilizar esta técnica, sino que se procederá a calcular los diferentes clusters que representan de mejor manera el conjunto de escenarios original, y luego, se determinará el escenario más cercano a cada uno de estos, conservándose así los escenarios que cumplan con esta última condición. Así, se evita trabajar con medias de los datos originales, manteniendo la distribución y comportamiento de la instancia creada.

Gracias a esta técnica, es posible hacer frente al crecimiento explosivo de escenarios que se produce en la medida que se trabaja con un mayor horizonte de planificación y/o se aumenta el número de nodos hijos por cada nodo según etapa.

Como alternativa a la metodología clásica basada en la creación de clusters, se han utilizado otras técnicas para llevar a cabo la reducción de escenarios, cuyo punto de partida se

centra en realizar una medida (bajo alguna métrica adecuada) de la distribución de probabilidad de la instancia original, con tal de luego llevar a cabo una representación de ésta con un menor número de escenarios, conservando de la mejor manera posible las características inherentes a dicha distribución. Se han desarrollado una serie de algoritmos que abordan este tipo de problema, utilizando diversas métricas para llevar a cabo la medición de las distribuciones de probabilidad y la distancia que separa una instancia de otra, siendo el aporte más destacado el abordado por Heitsch y Römisch [3], donde se utiliza una aplicación de la métrica de Fortet-Mourier para desarrollar dos algoritmos que se encargan de llevar a cabo la selección de aquellos escenarios (del set original) que mantienen de mejor manera (y con mayor estabilidad) la distribución del conjunto original.

A pesar de la mayor complejidad teórica que presenta esta metodología, en la práctica los algoritmos que se encargan de llevar a cabo la reducción son altamente eficientes, superando en rendimiento a un cálculo de clusters bajo métodos convencionales como k-means. Los más conocidos son los algoritmos fast-forward y back-forward, cuya diferencia radica en la forma en que se van seleccionando los escenarios. En el caso del algoritmo fast-forward, se comienza con un conjunto vacío y se van agregando aquellos escenarios seleccionados, mientras que en back-forward se comienza con el conjunto de todos los escenarios y se eliminan aquellos que no se planean conservar.

### **K-means**

Se utiliza una adaptación específica del algoritmo de clustering k-means clásico, algoritmo que recibe como input el número de clusters que se pretenden encontrar dentro del set de datos original, a través de la búsqueda de los  $K$  puntos medios representativos de dicho conjunto. Éste fue modificado especialmente para este trabajo de tal forma de que se mantengan en todo momento los eventos extremos (asociados a las probabilidades más bajas de ocurrencia) del conjunto original, utilizándose como criterio el mantener al menos un 50% de los casos cuya probabilidad se encuentra dentro del quintil más bajo de ocurrencia.

El pseudo-código del algoritmo programado es el siguiente:

---

#### **Algoritmo K – means modificado**

---

1. Inicializar:  $S$  conjunto de escenarios, número de clusters ( $K$ ) a seleccionar;
  2. Crear  $C_1, \dots, C_k \in S$ ,  $K$  clusters al azar,  $\varepsilon$  tolerancia;
  3.  $T = 0$ ;
  4. **while** ( $T \leq K$ ) {
    5. Determinar conjunto de escenarios más cercano a cada centro de cluster  $S_i$ ;
    6. Calcular escenario promedio  $\bar{S}_i$  dentro del cluster  $i$ ;
    7. **if**  $\left| \bar{S}_i - S_i \right| < \varepsilon \Rightarrow S_i$  se selecciona,  $T = T + 1$ ; }
    8. Determinar 50% de los casos extremos, conservarlos;
-

## Fast-Forward

Este algoritmo de reducción se encarga de calcular la distancia (utilizando una métrica probabilística o clásica de distancia si se desea) existente entre cada uno de los escenarios que componen la instancia en estudio, para luego seleccionar aquellos escenarios que mejor representan (en términos de la métrica seleccionada) el conjunto original.

Una vez elegidos los escenarios que conformarán el conjunto reducido, se lleva a cabo una redistribución del espacio probabilístico del árbol original, repartiendo así las probabilidades de ocurrencia de los escenarios no seleccionados al subconjunto objetivo, utilizando el teorema de Redistribución de probabilidades [3] (detalle en Anexos sección (11.3))

El pseudo-código del algoritmo es el siguiente:

---

### Algoritmo Fast – Forward

---

1. Inicializar:  $S$  conjunto de escenarios, número de escenarios ( $k$ ) a seleccionar;
  2. Conjunto objetivo  $T = \emptyset$
  3. **while** ( $T \leq k$ ) {
  4.   Calcular distancia (según métrica elegida) entre escenarios;
  5.   Determinar distribución de probabilidad del árbol original;
  6.   **if**  $S_i = \operatorname{argmin}_{S \in \{1..N\}} [\text{Métrica probabilística}] \Rightarrow T = T \cup \{S_i\};$
  7. Redistribuir el espacio probabilístico de  $S$  en  $T$ ;
- 

### 4.2.7 Bundling de Escenarios

El concepto de bundling de escenarios hace referencia a una técnica donde se trabaja con conjuntos de escenarios que al menos comparten un conjunto de información del árbol de escenarios original. Se presentan así una serie de sub-árboles de escenarios que se intersectan (o no) en determinados nodos a lo largo de su realización.

Así, se persigue trabajar con “macro-escenarios” en lugar de una serie de escenarios simples, lo que permite reducir el tamaño de la instancia de trabajo, resolviéndose un número menor de sub-problemas, sin embargo, estos serán claramente más complejos que los originales, por ende, existe un trade-off entre el número de escenarios que se busca reunir como máximo dentro de un único bundle y el rendimiento a la hora de resolver la instancia, puesto que el resultado podría ser contraproducente, dado que se trabaja con un número menor de sub-problemas, pero más complejos que los originales, ya que cuentan con mayor cantidad de variables y restricciones.

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 clásicos (actualización de penalizaciones, cálculo de promedios, 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 bajo el enfoque de bundling, se resuelven menos iteraciones, en las que los conjuntos de escenarios (macro-escenarios) son más complejos y requieren un mayor tiempo de resolución.

Para su confección, se utilizan dos enfoques en el presente trabajo de tesis, correspondientes a una selección aleatoria y otra basada en la técnica de clustering k-means, donde se realizan bundlings con aquellos escenarios que forman parte de un mismo cluster, pues así se apunta a que presenten características similares y por ende su resolución no debiese complejizarse en gran medida.

La figura 4.3 presenta de forma gráfica la presencia de tres bundles para una instancia correspondiente a un árbol de cinco escenarios:

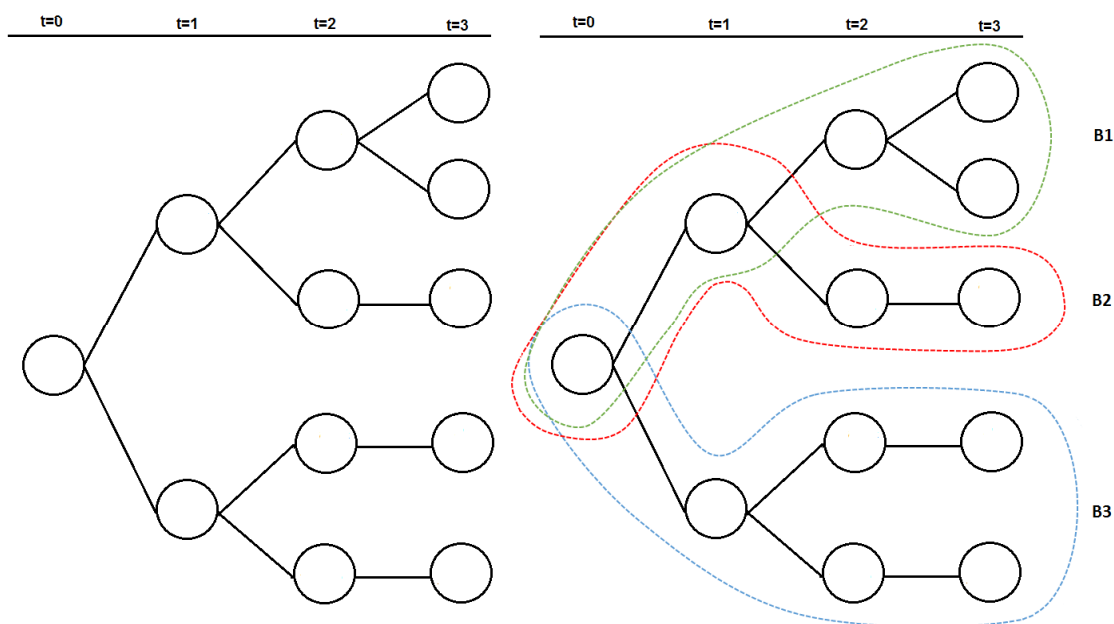


Figura 4.4: Bundling de Escenarios

#### 4.2.8 Fijar Variables

Esta estrategia comienza desde lo planteado por Watson y Woodruff en [10], donde se parte de la idea central de que es necesario acelerar la convergencia que experimenta la heurística PH en problemas no convexos. Así, una forma de abordar este problema consiste en fijar aquellas decisiones que ya hayan experimentado convergencia, es decir, cuyas variables sean iguales en aquellos escenarios que comparten nodos respectivos, con tal de tratar problemas de menor magnitud, obteniéndose así un mayor rendimiento en cuanto a velocidad de resolución.

La necesidad de esta técnica radica en que a lo largo del desarrollo de la heurística las variables enteras no consiguen alcanzar una estabilidad en cuanto a valor, por lo que comienzan a experimentar un “vaivén” o “saltos” entre las diversas soluciones, lo que impide que alcancen la convergencia de forma óptima. Por ende, la estrategia consiste en ir fijando aquellas decisiones que según los escenarios que comparten nodos de información respectivos, hayan alcanzado un estado estable, asumiendo así que dichas decisiones son las definitivas, supuesto que si bien es fuerte, puede mejorarse aplicando una serie de técnicas complementarias. Una de éstas consiste en esperar a que las variables alcancen un estado estacionario de convergencia y que no cambien durante un número determinado de iteraciones, para luego, en caso de alcanzarse dicho número, fijarlas.

Las distintas estrategias utilizadas son:

- **Fijar tras iteración cero:** Corresponde a la estrategia donde se fijan aquellas decisiones que alcanzan la convergencia la primera vez que se resuelve cada uno de los escenarios por separado (ver subsección (2.3.2)), por ende se observa que si aún sin considerar la no anticipatividad existen ciertas decisiones que la cumplen, entonces deben mantenerse.

Claramente el supuesto que realiza esta estrategia es bastante fuerte, pues se asume que la solución final que se alcanzará se encuentra compuesta por la solución inicial fijada, hecho que puede ser totalmente erróneo en algunos casos.

- **Fijar tras ciertas iteraciones convergentes:** Corresponde a fijar las decisiones que han convergido previamente y que se han mantenido así durante un determinado número de iteraciones o retardo. Presenta una mayor solidez como estrategia en comparación a la anterior, debido a que no se realiza un supuesto tan fuerte como el utilizar una solución inicial parcial. Su implementación se realiza de tal manera que es posible distinguir el número de retardos según la etapa que se está resolviendo.
- **Políticas de fijar decisiones positivas (unos) de las negativas (ceros):** Esta estrategia se vuelve útil en el problema en estudio debido a que en éste existen restricciones del tipo “se debe hacer a lo más una vez”, lo que conlleva a que estas variables ligen decisiones dentro de las distintas etapas del árbol. Luego, una estrategia abordada en este tipo de problemas consiste en que al fijar una decisión en valor unitario, se debe ser consecuente y consistente con dicha restricción, fijando luego en cero todos los valores de dicha variable en las etapas restantes, de los escenarios correspondientes.

Además se debe considerar el hecho que esta estrategia incluye fijar en valor cero en todos aquellos nodos de etapas previas que aún no estén totalmente convergidos a dicho valor. Es un criterio “agresivo” pero se vuelve bastante útil a la hora de comparar el rendimiento entre los modelos estocásticos y la formulación extendida.

A partir de los experimentos realizados (ver resultados en capítulo 6), se determina que la estrategia de fijado de variables es fundamental para alcanzar una convergencia de calidad por

parte del algoritmo, pues de otro modo no existe forma de competir con la formulación extendida cuando las instancias son de tamaño reducido, aún bajo una implementación en paralelo.

Claramente existe una cierta desventaja asociada al uso de esta estrategia, materializada en que al utilizar una configuración muy agresiva de fijado de variables, la calidad de la solución puede decaer de forma significativa, y al mismo tiempo, en problemas estocásticos multietápicos no es posible garantizar la existencia de una solución factible del problema extendido al fijar variables, por lo que se realizaron una serie de experimentos previos, con el afán de encontrar la mejor configuración posible, tanto en términos de velocidad y calidad de solución, como de potencial factibilidad.

## Capítulo 5

# Implementación

La implementación computacional se realizó utilizando principalmente el software Coopr/PYSP de los laboratorios Sandia [8], consistente en un paquete de software específico para la resolución de problemas de optimización estocástica.

El entorno de trabajo se ubica bajo el lenguaje de programación Python, siendo PYOMO el paquete principal que permite llevar a cabo el modelamiento y resolución de los modelos estocásticos en estudio, permitiendo una definición natural del problema determinístico y su extensión a una estructura estocástica representada por un árbol de escenarios donde se contenga la información relativa a las posibles realizaciones futuras de las variables aleatorias incluidas. Así, es directo llevar a cabo una serie de experimentos, utilizando la implementación en Matlab del algoritmo generador de escenarios y su posterior formateo bajo JAVA al formato PYSP señalada en la sección (3.4).

El solver utilizado para correr todas las instancias corresponde a CPLEX 12.4 en su versión de 32 bits, el cuál es llamado a través de una serie de scripts programados en Python 2.7 de 64 bits, así como de archivos de batería MS-DOS en Windows versión 8.1 y BASH en LINUX.

Todas las instancias utilizan los 17 bosques y sus parámetros en su totalidad, siendo la única diferencia el total de escenarios a procesar en cada una de éstas.

### 5.1 Serial: PYOMO – PYSP

La implementación serial del problema en estudio se realizó bajo el paquete de modelamiento PYOMO, desarrollándose todos los modelos testeados en su totalidad bajo este formato, a través del lenguaje de programación PYTHON. Así, tanto el modelo determinístico la formulación extendida fueron creados directamente en este entorno de trabajo.

Junto a lo anterior, se utiliza el script “runef” del paquete PYSP de los laboratorios Sandia, script desarrollado en PYTHON que permite crear la formulación extendida de una instancia a partir de la estructura de un árbol de escenarios clásica en formato PYSP, así, su implementación fue totalmente directa a partir del programa en JAVA señalado en la sección (3.4).

Del mismo modo, todas las instancias trabajadas fueron luego generadas en formato LP, cuya sintaxis es la utilizada por el solver CPLEX para la lectura y generación de modelos, lo que permite una mayor velocidad de lectura y resolución de los problemas en estudio.



## 5.2 Paralela: PYSP

La implementación en paralelo del algoritmo de Progressive Hedging se realizó utilizando el script “runph” del paquete de software Coopr de los laboratorios Nacionales de Sandia, script que permite una implementación directa y natural con el paquete “pickle” de Python [23], el que permite establecer la creación de una serie de servidores locales dentro del computador o estación de trabajo multi-núcleo utilizada, permitiendo así utilizar el esquema de trabajo presentado en la subsección (4.2.2).

Así, es posible utilizar programación distributiva dentro de un computador con varios núcleos, enviando cada uno de los subproblemas a distintos puntos de cómputo y estableciendo colas ad-hoc según la carga de trabajo que experimente cada uno de estos, y luego generar la comunicación entre ellos para continuar con la siguiente iteración del algoritmo. Lo mismo al utilizar una serie de estaciones de trabajo (típicamente distribuidas a modo de nodos donde se encuentran cúmulos de procesadores).

Del mismo modo, permite una implementación directa en su código fuente de las técnicas de mejora explicadas en la sección (4.2), tales como el fijado de variables binarias, variación de mipgap y el cálculo del parámetro  $\rho$  según iteración. Junto a lo anterior, presenta la posibilidad de añadir propios módulos de programación por parte de usuarios avanzados (plug-ins), lo que conlleva a un manejo profundo de todos los parámetros (número de bundles y la forma de determinarlos, por ejemplo), customizando de forma completa la implementación del algoritmo, lo que otorga así un sin límite de posibilidades para el usuario (una solución a la medida).

Las llamadas realizadas a estos scripts junto a sus plug-ins se realizan a través de un programa coordinador programado en JAVA, que se encarga de levantar los procesos de los diversos servidores a utilizar, así como de crear los diferentes archivos MS-DOS (Windows) y/o BASH (LINUX) que serán utilizados para llamar a los scripts en PYTHON.

## 5.3 Simulación

Del mismo modo, en el capítulo 7 se lleva a cabo una comparación directa entre el desempeño del modelo determinístico y el estocástico completo bajo una metodología de simulación, simulación que es coordinada por completo por un programa principal desarrollado en JAVA, el cual se encarga de generar el llamado a Matlab para la generación de escenarios, transformar sus outputs al formato PYSP y luego llevar a cabo la creación de todos los scripts y archivos BAT y BASH necesarios para correr los modelos en estudio formulados en PYOMO.

Más detalles de esta implementación en la sección (7.1).

## 5.4 Comparación

Comparando las ventajas y desventajas de cada uno de los enfoques, es de esperarse que la implementación paralela sea la vencedora en cuanto a tiempos de resolución, puesto que se

explota al máximo la capacidad de los procesadores disponibles, sin embargo se debe tomar en cuenta la presencia de “overhead” o tiempo de inicialización y comunicación que se invierte durante este tipo de programación, puesto que es necesario crear las instancias (objetos) que serán enviados a cada uno de los procesadores, coordinarlos y enviarlos a cada uno según el nivel de carga que se presente en estos, y luego, recibir y leer las soluciones obtenidas en cada uno de ellos, por lo que tal cómo se observará en la sección (6.3) de resultados, la implementación serial presenta un mejor rendimiento en la medida que el número de escenarios disminuye (acercándose al problema determinístico).

## 5.5 Hardware y Software (Recursos computacionales)

Las máquinas utilizadas para llevar a cabo todos los experimentos del presente trabajo de tesis corresponden a un computador de escritorio de simple acceso para el público, así como de una estación de trabajo de múltiples núcleos y nodos (cluster computacional) cuyas características se presentan a continuación:

- **Computador de Escritorio**

**Sistema Operativo:** Windows 8.1

**Memoria RAM:** 16 GB DDR4

**CPU:** Intel I7-4700 2.4 GHZ

**Solver:** CPLEX 12.4 32 bits

**Scripts:** MS-DOS, PYTHON 2.7 64 bits, JAVA 64 bits

**Middleware:** Coopr, Sandia National Labs [8]

- **Estación de trabajo multi-núcleo**

**Sistema Operativo:** Linux

**Memoria RAM:** 64 GB

**CPU:** AMD Athlon 2.2 GHZ

**Solver:** CPLEX 12.4 32 Bits

**Scripts:** BASH, PYTHON 2.7, JAVA

**Middleware:** Coopr, Sandia National Labs [8]

## Capítulo 6

# Resultados

En esta sección se presentan los resultados obtenidos durante el trabajo de tesis, de acuerdo a la metodología señalada en el capítulo anterior. En primera instancia se busca realizar una comparación entre los valores de la función objetivo, Gap y tiempo de resolución entregado por la formulación extendida serial clásica, la cual es resuelta directamente en CPLEX, con respecto a la resolución de la misma instancia bajo una implementación serial del algoritmo PH (con todas las mejoras señaladas en la sección (4.2)), para luego desarrollar la misma comparación con respecto a la implementación en paralelo del algoritmo PH.

Las instancias de prueba experimentales consisten en 1, 2, 3, 5, 10, 20, 30, 40, 50, 107, 110, 216, 343, 512, 729 y 1.000 escenarios. Todas contemplan el total de bosques de la instancia original y se han testeado con todas las combinaciones posibles de técnicas de mejora para el caso estocástico.

La metodología para llevar a cabo los experimentos consistió en una reproducción automática de cada una de las instancias mediante una serie de scripts de MS-DOS/BASH (según el sistema operativo), los cuales se encargan de generar las llamadas al paquete de optimización PYOMO y los diferentes scripts de PYTHON involucrados. Luego, cada uno de los resultados se almacena en un formato específico de texto en su propia carpeta individual con toda la información necesaria para llevar a cabo la corrida de la instancia. Por ende, el resultado final de los experimentos corresponde a una macro carpeta que contiene subcarpetas específicas para cada una de las instancias, identificadas por nombres claves asociados al número de escenarios que se están incluyendo, de tal forma que es un proceso inequívoco y automatizado.

### 6.1 Problema General

Típicamente al utilizar el algoritmo de Progressive Hedging se presenta un gran énfasis en cuanto a la métrica de convergencia que se asocia al algoritmo (distancia euclidiana) a la hora de evaluar que tan distintas/similares son las soluciones obtenidas dentro de los diversos escenarios que comparten nodos de información. Con tal de comprender esto, se presenta en las figuras 6.1 y 6.2 un gráfico de convergencia para dos de las instancias experimentales testeadas, donde claramente el objetivo ideal consiste en que el valor decrezca con cada una de las iteraciones que se realicen, pues implica que el problema está alcanzando la convergencia y por ende la solución del mismo.

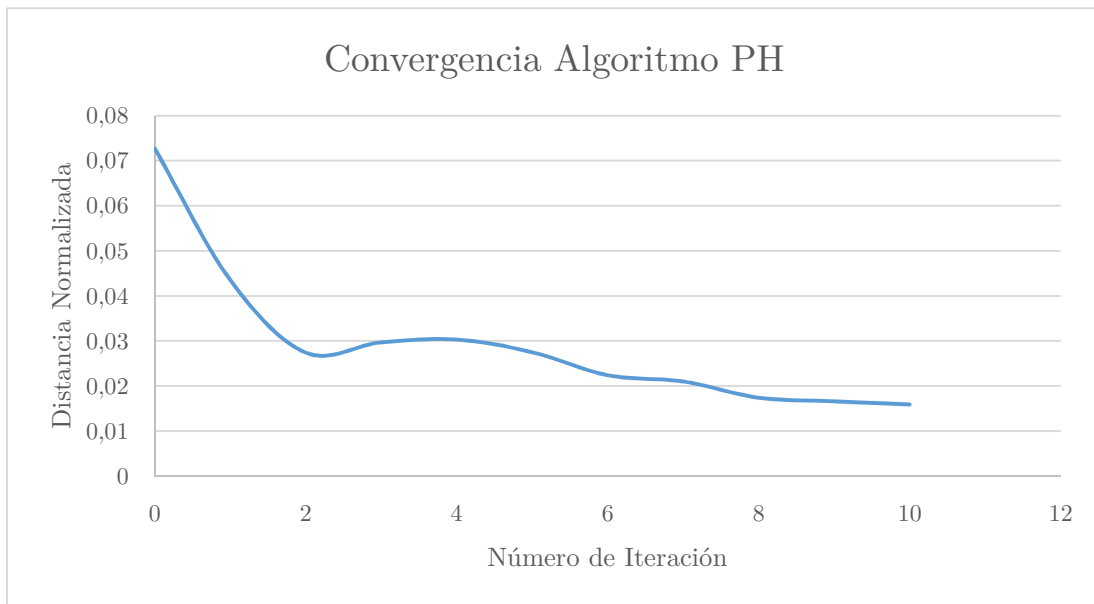


Figura 6.1: Gráfico con convergencia del algoritmo PH en función del número de iteraciones

Sin embargo, esto no siempre es posible en la práctica, puesto que dependiendo de la configuración que se esté utilizando, es factible que la solución realice vaivenes o saltos entre iteraciones, cambiando así el valor obtenido en una serie de variables, lo que genera un retroceso en cuanto a la métrica de convergencia – efecto totalmente no deseado – indicándonos que la configuración utilizada no es la idónea.

En la figura 6.1 se observa cómo se produce un retroceso en cuanto a la convergencia del algoritmo entre las iteraciones 2 y 4. En este caso no se estaba utilizando ninguna estrategia de fijación de variables y se estableció como máximo iterar diez veces, sobre un punto de corte de 0.01 para la distancia normalizada.

Lo ideal es que el valor de la distancia siempre decrezca con cada iteración y clásicamente se experimenta una desaceleración de la convergencia en la medida que se avanza en el número de iteraciones, lo cual es predecible e inocuo, pues al utilizar un valor tan bajo del parámetro  $\rho$  se espera dicho comportamiento. Este comportamiento se observa en la figura 6.2, donde se ve un decrecimiento constante de la convergencia, amparado por el uso de estrategias de fijado de variables.

Complementario a lo anterior, se espera que el valor esperado de la función objetivo tienda a disminuir (si se está maximizando) en la medida que se fuerce la condición de no anticipatividad, en el caso de los experimentos realizados, éste tiende a ser estable durante las primeras iteraciones y luego en las finales disminuye de forma significativa, debido a que se alcanza el valor totalmente convergido y factible del problema, cuando CPLEX realiza la corrida de la Formulación extendida.

Del mismo modo, la cantidad de variables fijadas por iteración tiende a presentar un comportamiento en aumento, pero a tasa decreciente. Debido a las políticas señaladas en la subsección (4.2.8) se tienden a fijar una mayor cantidad de valores unos que ceros.

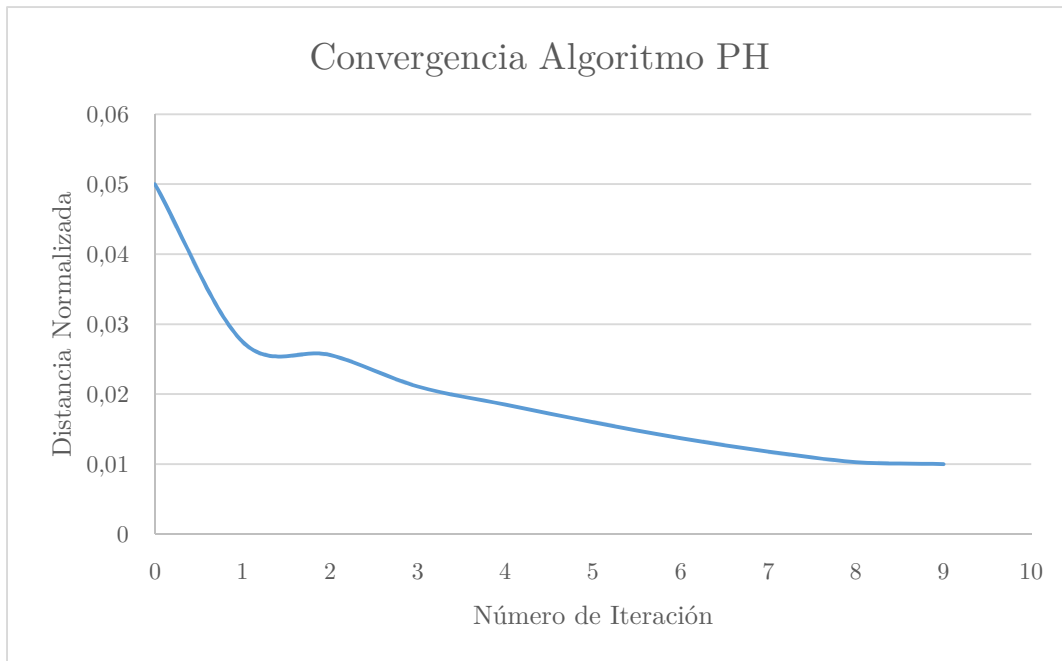


Figura 6.2: Convergencia del algoritmo PH para una instancia de test

## 6.2 Caso Serial

En este caso se contrastó la implementación serial del algoritmo PH utilizando las mejores configuraciones encontradas con respecto al rendimiento de la formulación extendida.

En ninguna de las instancias más pequeñas (menos de 10 escenarios) fue posible superar a CPLEX, debido principalmente a que en la actualidad el algoritmo que se encarga de resolver los problemas de programación lineal ha alcanzado un nivel de eficiencia y eficacia significativas (amparado por un gran avance en cuanto al hardware computacional accesible para público común). Luego en instancias menores a 100 escenarios se logró compensar dicha propiedad, alcanzando (y superando) así la implementación serial de PH los tiempos de resolución involucrados. A partir de los 216 escenarios, se observa una victoria completa para el algoritmo PH, puesto que la formulación extendida es incapaz de resolverse debido a la carencia de memoria que le implica al computador el tratar de resolver directamente instancias de tan elevado tamaño, aun contando con hardware de última generación para computadores de escritorio.

Sin embargo, observando los tiempos involucrados en la resolución de instancias superiores a los 216 escenarios, se vuelve imperante la implementación en paralelo del algoritmo, puesto que alcanzar una solución de calidad toma demasiado tiempo, y por ende, recursos. Basta observar el tiempo de resolución necesario para la instancia de 1.000 escenarios, que ronda las 22 horas 30 minutos en la figura 6.3. El Gap calculado para el algoritmo PH proviene de una estimación realizada entre el valor objetivo y el Gap entregado por CPLEX.

N° de Escenarios	Valor F.Objetivo EF	Gap EF %	Tiempo de Resolución EF [min]	Valor F.Objetivo PH	Gap PH %	Tiempo de Resolución PH [min]	N° de Iteraciones
1	29.776.250	2%	0,07583	28.834.626	5,331%	0,099	1
2	30.344.979	2%	0,17666	29.381.528	5,345%	1,483	9
3	30.454.398	2%	0,26533	29.491.355	5,331%	2,632	10
5	30.493.623	2%	0,54666	29.533.298	5,317%	1,782	4
10	30.464.784	2%	1,28650	29.517.559	5,273%	1,125	6
20	30.629.406	2%	3,98983	29.689.155	5,230%	2,224	7
30	30.663.941	2%	6,12250	29.726.236	5,218%	3,419	5
40	30.587.347	2%	6,64166	29.656.597	5,201%	4,498	5
50	30.678.733	2%	13,5758	29.710.711	5,323%	5,621	6
107	29.906.695	2%	9,47617	29.906.695	2,000%	14,942	4
110	29.886.244	2%	9,29216	29.886.244	2,000%	16,966	1
216	29.983.254	0,52%	141,3000	29.983.254	0,52%	68,340	1
343	OUT OF MEMORY	-	-	29.972.276	2,000%	670,936	1
512	OUT OF MEMORY	-	-	29.981.786	2,000%	852,468	6
729	OUT OF MEMORY	-	-	29.984.276	2,000%	1025,599	4
1000	OUT OF MEMORY	-	-	29.996.688	2,000%	1354,479	7

Figura 6.3: Experimentos realizados con las instancias en formulación extendida y PH serial

Se observa que ambos métodos de resolución presentan un comportamiento similar para instancias pequeñas, e incluso la formulación extendida logra superar a PH serial en las instancias de 107 y 110 escenarios, tanto en términos de tiempo de resolución como del gap alcanzado, gap que supera el 5% en el caso de la formulación PH debido a la estrategia de fijación de variables. Una vez superado dicho umbral, la formulación extendida simplemente carece de posibilidades ante el algoritmo de descomposición, tanto en términos de tiempo de resolución como de la posibilidad de correr el modelo (problemas de recursos computacionales aparecen al realizar la carga de las instancias, no es posible iniciar el proceso de optimización).

Por ende, es posible concluir en primera instancia que el uso de la formulación extendida será recomendable en el caso de contar con instancias de trabajo menores a los 100 escenarios, y a partir de este umbral, se deberá contar con otras técnicas para su resolución, ya sean heurísticas, algoritmos de descomposición o un mix de estos.

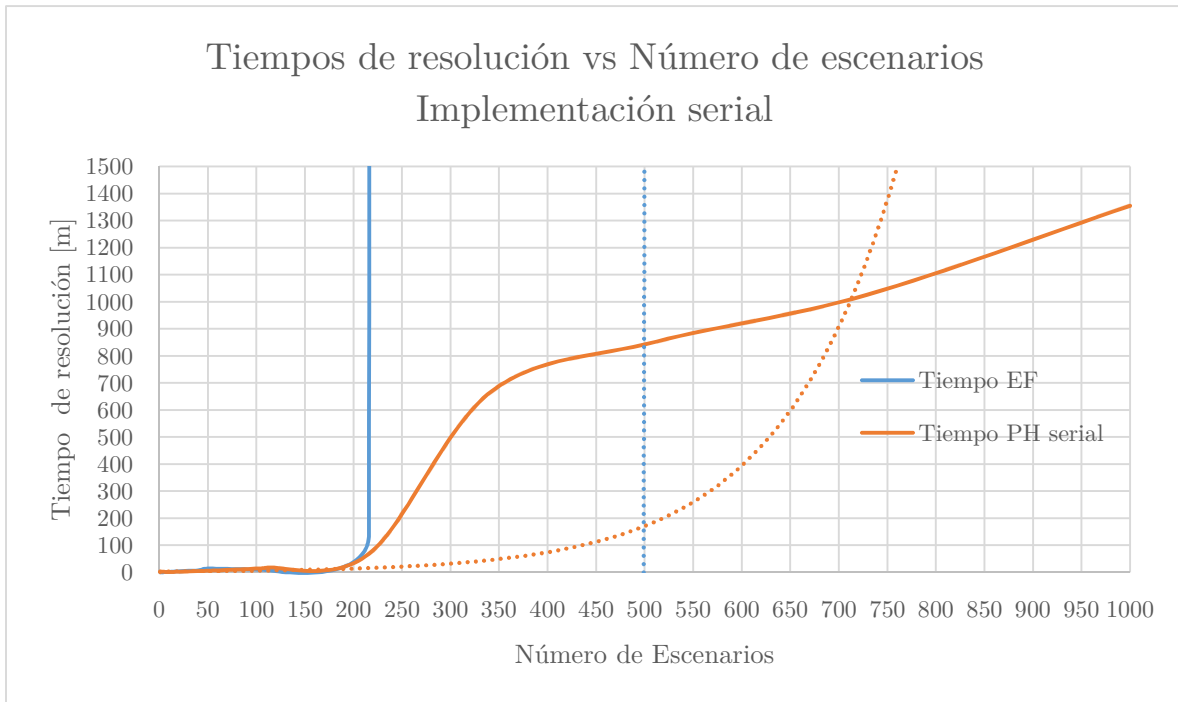


Figura 6.4: Gráfico de comparación entre los tiempos de resolución de la implementación serial de PH y la formulación extendida, para las instancias experimentales

En la figura 6.4 se observa como al ir aumentando el número de escenarios tratados los tiempos de resolución tienden a dispararse de forma exponencial, siendo para la formulación extendida imposible abordar instancias que contengan más de 200 escenarios. Del mismo modo se percibe que al superar los 300 escenarios, la velocidad con que crece la curva de tiempo de resolución tiende a disminuir, esto pues la tendencia apunta a alcanzar una solución cada vez más estable en la medida que se agreguen más y más escenarios, puesto que la diferencia entre estos será cada vez menor (simulando un proceso estocástico continuo) y por ende se tienden a fijar una mayor cantidad de variables en pocas iteraciones, lo que permite alcanzar una mayor eficiencia a la hora de abordar problemas de esta magnitud. Sin embargo, es claramente necesario pasar a una implementación paralela, dado que los tiempos involucrados en la resolución de las instancias son demasiado elevados para llevar a cabo la corrida de los modelos de forma periódica, dado el costo asociado al tiempo involucrado.

La mejor configuración encontrada para esta implementación es:

- Linealización agresiva del término próximas cuadrático, en forma de cono invertido con solo dos trazos para las variables continuas, una para las binarias
- Gap variable según iteración, desde un 20% a un 0.05%
- Fortalecimientos óptimos según subsección [4.2.1]
- Estrategia de Fixed  $\rho$  según subsección [4.2.4]
- Fijar variables luego de la tercera iteración convergente + agresiva

- Máximo de 10 iteraciones
- Punto de convergencia: 0.01

### 6.3 Caso Paralelo

En este caso se contrastó la implementación serial del algoritmo PH utilizando las mejores configuraciones encontradas con respecto al rendimiento de dicha configuración implementada de forma paralela, bajo programación distributiva. El número básico de núcleos con los que se experimentó es de 4 inicialmente, puesto que en la actualidad los computadores de escritorio de acceso público cuentan con este tipo de recursos, sin ser prohibitivos para un usuario corriente. Luego se realizaron pruebas con mayor número de procesadores, tanto en la misma máquina como en clusters computacionales.

En la figura 6.5 es posible observar claramente como el desarrollo de los experimentos presenta un desenlace lógico y acorde a lo esperado teóricamente, puesto que la implementación en paralelo presenta un mejor rendimiento en comparación a la implementación serial en todas las instancias experimentales testeadas, con la única excepción del caso para un único escenario, hecho que se explica de forma directa por la presencia de “overhead” (tiempo de inicialización e intercambio de información) en el caso paralelo.

En todas las instancias se obtuvo un rendimiento de al menos un 10% superior para el caso paralelo (es decir, el tiempo de resolución corresponde al 90% del caso serial), siendo en el mejor de los casos disminuido hasta un 45% del tiempo originalmente invertido (instancia de 110 escenarios). Tomando en cuenta el hecho de que estos experimentos se realizaron en un computador de escritorio de simple acceso para un usuario común, y bajo la distribución a únicamente 4 núcleos, se obtienen resultados más que contundentes que avalan la implementación en paralelo del algoritmo.

Al comparar su rendimiento con el obtenido bajo la formulación extendida, se observa que ésta supera de forma significativa a la implementación en paralelo en instancias de 10 o menor número de escenarios, lo que se debe principalmente al hecho de que CPLEX puede manejar dicho tamaño de instancias sin problemas y además dicha implementación carece de tiempos de envío y respuesta de los subproblemas entre procesadores. Luego, a partir de instancias de 20 escenarios, se observa una gran mejoría en cuanto a los tiempos de resolución, siendo los casos más destacados las instancias de 20 a 50 escenarios, obteniéndose tiempos promedios de resolución apenas un 33% del tiempo de resolución original (ver figura 6.6).

Si bien lo anterior no establece una relación lineal entre “número de núcleos y tiempo disminuido” debido a los tiempos de comunicación involucrados, es posible concluir que la implementación en paralelo es la más adecuada para abordar el problema forestal utilizando instancias masivas.



N° de Escenarios	Valor F.Objetivo PH	Gap PH %	Tiempo de Resolución PH [min]	Valor F.Objetivo PH paralelo	Gap PH paralelo %	Tiempo de Resolución PH paralelo [min]	N° de Iteraciones
1	28.834.626	5,331%	0,099	28.834.626	5,331%	0,116	1
2	29.381.528	5,345%	1,483	29.381.528	5,345%	1,261	9
3	29.491.355	5,331%	2,632	29.491.355	5,331%	1,960	10
5	29.533.298	5,317%	1,782	29.533.298	5,317%	1,391	4
10	29.517.559	5,273%	1,126	29.517.559	5,273%	0,680	6
20	29.689.155	5,230%	2,225	29.689.155	5,230%	1,451	7
30	29.726.236	5,218%	3,419	29.726.236	5,218%	2,094	5
40	29.656.597	5,201%	4,498	29.656.597	5,201%	2,555	5
50	29.710.711	5,323%	5,621	29.710.711	5,323%	3,060	6
107	29.906.695	2,000%	14,942	29.906.695	2,000%	7,224	4
110	29.886.244	2,000%	16,966	29.886.244	2,000%	7,596	1
216	29.983.254	0,520%	68,340	29.983.254	0,52%	45,565	1
343	29.972.276	2,000%	670,936	29.972.276	2,00%	603,888	1
512	29.981.786	2,000%	852,468	29.981.786	2,00%	654,182	6
729	29.984.276	2,000%	1025,599	29.984.276	2,00%	842,789	4
1000	29.996.688	2,000%	1354,479	29.996.688	2,00%	1045,256	7

Figura 6.5: Cuadro comparativo entre implementación serial y paralela (con 4 núcleos) del algoritmo PH

Tal como se ha señalado, un punto a tener en consideración consiste en el número de procesadores en simultáneo que están siendo utilizados. En teoría, a mayor número de procesadores, debiese alcanzarse un mejor rendimiento del algoritmo PH en paralelo, puesto que cada uno de los subproblemas asociados a los escenarios serían abordados por distintos procesos y en simultáneo, sin embargo, preciso es señalar que existen ciertos inconvenientes de llevar esto a cabo en un único computador de escritorio (en contraposición a la implementación en un conjunto de computadores o clusters).

Se realizaron los mismos experimentos en computadores de escritorio que cuentan con 8 y 16 procesadores (el resto de las características es similar al descrito en la sección (6.2)), y en contra de lo que se pensaba, los resultados no fueron satisfactorios, puesto que en la medida que se aumenta el número de procesos en paralelo dentro de una única máquina, el uso intensivo de memoria, disco duro y la propia temperatura interna que alcanza la máquina juegan en contra del rendimiento de ésta, obteniéndose soluciones idénticas en tiempos de resolución similares o peores a los que se obtuvieron con una configuración de 4 núcleos, por lo que se entiende así que para computadores de uso personal, éste sería el número adecuado de procesadores a utilizar para la implementación paralela.

N° de Escenarios	Extended Formulation	PH Serial	N°de Escenarios	Extended Formulation	PH Serial
1	153,8462%	117,84512%	50	22,5450%	54,45057%
2	714,1509%	85,05618%	107	76,2298%	48,34525%
3	738,7563%	74,47442%	110	81,7463%	44,77146%
5	254,4512%	78,05836%	216	63,9070%	66,67462%
10	52,8695%	60,41451%	343	INF	90,00697%
20	36,3758%	65,24313%	512	INF	76,73983%
30	34,2153%	61,26438%	729	INF	82,17538%
40	38,4818%	56,82155%	1000	INF	77,17040%

Figura 6.6: Porcentaje de tiempo de resolución bajo PH paralelo en relación a la formulación extendida y PH serial

Del mismo modo, un aspecto a tener en cuenta a la hora de llevar a cabo la implementación tanto serial como en paralelo (en especial esta última) del algoritmo PH, es que se requiere de un tiempo de inicialización para crear todas las variables y parámetros que serán utilizados a lo largo de las iteraciones, así como realizar la lectura del árbol de escenarios bajo la estructura especificada por el software, puesto que éste puede llegar a ser bastante elevado en la medida que el tamaño de la instancia vaya creciendo, siendo en la mayoría de los casos mayor al tiempo de resolución invertido. Esto ocurre principalmente porque los métodos de lectura de datos que utiliza PYOMO no son eficientes en la actualidad, pues se encuentran en pleno desarrollo, así como los métodos que el paquete de software Coopr Sandia implementa, lo que conlleva a tener tiempos de espera elevados en un comienzo. Sin embargo, el tema principal que se aborda a lo largo de este trabajo de tesis corresponde a los tiempos de resolución de cada una de las metodologías, para más detalles respecto a los tiempos de inicialización invertidos, ver anexos sección (11.5).

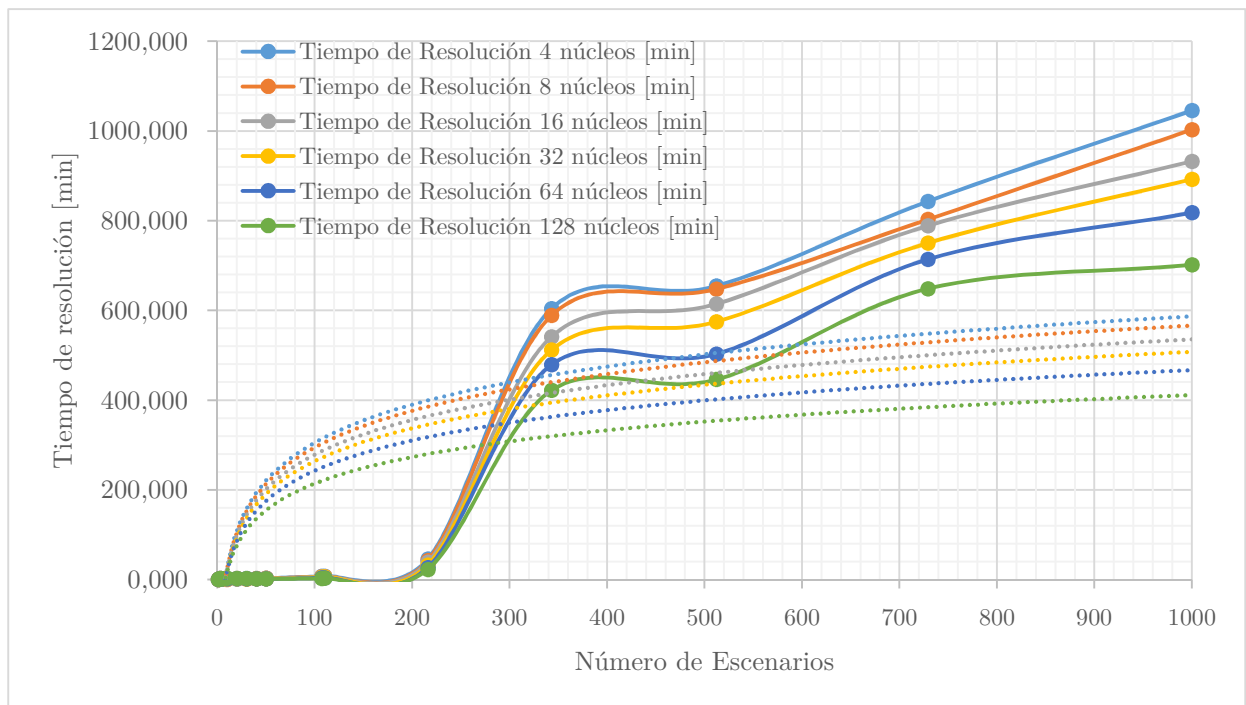


Figura 6.7: Comparación rendimiento implementación paralela según n° de procesadores

Se realizaron experimentos modificando el número de procesadores en paralelo alojados en un cluster computacional, utilizando 4, 8, 16, 32, 64 y 128 núcleos respectivamente, sin embargo, no se encuentran en una única máquina, sino distribuidos en nodos, lo que permite evitar que una única máquina con alta capacidad de cómputo sea saturada por la cantidad de recursos requeridos.

A partir de los experimentos realizados, se concluye que en instancias menores a los 100 escenarios, la mejora en rendimiento no se aprecia de forma significativa, principalmente debido a que al tener un mayor número de procesadores se invierte una mayor cantidad de recursos y tiempo de coordinación y comunicación entre el equipo maestro con las unidades esclavas auxiliares. Aún así, en instancias medianas (50 escenarios) se logra disminuir el tiempo de resolución a menos del 50% en relación al caso basal de 4 núcleos en paralelo.

El mayor rendimiento aparece en instancias que se encuentran entre los 100 y 700 escenarios, alcanzándose reducciones de tiempo de resolución del orden de un 70% para el caso de 128 procesadores en paralelo. Por ende, se compensan los tiempos de comunicación e inicialización de forma significativa, siendo siempre útil el utilizar un número de núcleos menor al número de escenarios, debido a que empíricamente se observa que en la medida que estos superan el universo de escenarios original, se producen disminuciones en el rendimiento global de resolución de la instancia, asociado al overhead inherente a la formulación paralela.

En el caso de instancias masivas – superiores a los 500 escenarios – se observa una reducción marginal decreciente en cuanto a los tiempos de resolución involucrados en la medida que se cuenta con un mayor número de procesadores. La explicación de este comportamiento se asocia a la gran cantidad de recursos de memoria ram y almacenamiento necesarios por este tipo de instancias de gran tamaño, lo que impide su distribución de forma eficiente, así como de la presencia de escenarios de alta complejidad que ralentizan el paso a la siguiente iteración. Detalles de los tiempos de resolución involucrados en Anexos sección (11.6).

Dado lo señalado en la presente sección, la formulación en paralelo bajo la metodología de resolución del algoritmo Progressive Hedging se erige como la más adecuada para tratar instancias masivas del problema forestal en estudio, con la potencialidad de seguir mejorando su rendimiento en el futuro, en la medida que se mejoren las técnicas de distribución y optimización bajo computación paralela, junto a un manejo más eficiente de la memoria disponible en hardware.

## **6.4 Caso Bundlings**

A partir de lo señalado en la subsección (4.2.7), se realizan una serie de experimentos ligados a la creación de bundlings de escenarios, los cuáles se confeccionan a partir de la utilización del algoritmo de clustering k-means, con tal de determinar aquellos escenarios que presentan características similares, y por ende, son más propicios a trabajarse en conjunto.

Se realizaron bundlings agrupando un 75%, 50% y 25% del tamaño original de cada una de las instancias experimentales, bajo configuración serial y en paralelo del algoritmo Progressive Hedging. Las instancias menores a 10 escenarios fueron descartadas debido a la baja utilidad en términos prácticos de llevar a cabo esta técnica, debido a que los tiempos de resolución involucrados presentes en sus formulaciones extendida y bajo algoritmo PH no presentan mejoras significativas.

En instancias menores a los 200 escenarios, se observa una mayor calidad de solución (en términos de valor de la función objetivo) para menores tiempos de cómputo en relación a los casos sin bundling, tanto en la formulación serial como en paralelo, sin embargo, los tiempos finales de resolución obtenidos (cuando se alcanzan los criterios de convergencia) no presentan una mejora significativa en relación al caso original, concluyéndose así que para instancias no masivas, la metodología utilizada no presenta grandes ventajas en relación a las formulaciones clásicas.

En el caso de instancias masivas, se observan tiempos de resolución bastante cercanos a los experimentos previamente realizados, sin embargo, los tiempos de inicialización involucrados (creación de bundles, actualización de parámetros y otros elementos propios del algoritmo PH) perjudican de forma significativa el rendimiento de la técnica, puesto que para instancias del orden de 216 escenarios se requiere de cerca de 340 minutos (5 horas 40 minutos) para poder comenzar a resolver el problema y ya en instancias superiores a los 500 escenarios, los tiempos involucrados superan los 3 días para llevar a cabo el procesamiento y creación de las nuevas instancias bajo las configuraciones presentadas en las secciones anteriores.

Así, aparecen una serie de inconvenientes en relación a la cantidad de memoria ram utilizada por el computador para llevar a cabo la preparación del entorno de trabajo requerido, hecho que ralentiza aún más la resolución de cada una de las instancias, concluyéndose así que el trade-off presente entre la leve mejora obtenida en tiempos reales de cómputo y los minutos invertidos en la inicialización del algoritmo no es óptimo.

N° Escenarios	Tiempo Promedio Inicialización [min]	Tiempo Promedio de resolución [min]	Máxima Diferencia Tiempos de resolución [min]
10	15,0940556	0,29722222	0,00633333
20	30,1165	0,68727778	0,17733333
30	45,3665556	1,06861111	0,00333333
40	60,612	1,64972222	0,37483333
50	75,3505	2,13022222	0,45233333
107	164,676167	5,37622222	0,25774195
110	166,056833	5,71038889	0,49616667
216	339,487	45,78789422	0,54789986
343	738,333333	600,22789421	1,00021557

N° Escenarios	Tiempo Promedio Inicialización [min]	Tiempo Promedio de resolución [min]	Máxima Diferencia Tiempos de resolución [min]
512	4132,23333	644,84578918	1,02659896
729	7048,46	839,11222856	1,24567841
1000	9790,48	1041,99685575	1,45678611

Figura 6.8: Resultados instancias experimentales utilizando técnica de bundling

## 6.5 Caso Reducción de escenarios

Se llevó a cabo la implementación de los algoritmos de reducción de escenarios basados en el método clásico de clustering por k-means y el algoritmo fast-forward. En primera instancia se comparó el rendimiento entre ambos métodos a la hora de llevar a cabo la reducción del tamaño de la instancia original, específicamente en cuanto a tiempos de cómputo asociados para seleccionar y generar el nuevo subconjunto de escenarios a incluir en el modelo en estudio.

En base a los resultados obtenidos, se observa un rendimiento superior del algoritmo fast-forward en relación a la versión modificada de k-means en todas las instancias en estudio. Se presenta un tiempo promedio de 2 segundos para seleccionar los escenarios a utilizar (y redistribuir las probabilidades de aquellos desechados de la instancia original) y llevarlos al formato utilizado para resolver el modelo en el caso del algoritmo fast-forward, mientras que el tiempo promedio involucrado en instancias pequeñas (menores a 100 escenarios) por parte del algoritmo k-means modificado asciende a los 4 minutos, alcanzando en instancias masivas (superiores a los 500 escenarios) tiempos superiores a los 15 minutos de cómputo para llevar a cabo el cálculo de los centroides y posterior selección de los escenarios representativos.

Utilizando las mejores configuraciones encontradas para el algoritmo PH tanto en su versión serial como paralela, se llevó a cabo la resolución de cada una de las instancias en su versión reducida, aplicándose una reducción de un 75%, 50% y 25% para las instancias menores a los 300 escenarios, y luego para el tramo superior, se agrega una reducción de un 10%, así como de un 5% y 1% para el caso de la instancia de 1000 escenarios.

Los mejores resultados se obtienen al aplicar el algoritmo de reducción fast-forward, tanto en términos de calidad de la solución como del tiempo involucrado en la resolución de cada una de las instancias. La calidad de la solución se mide a partir de un factor de infactibilidad, calculado como el número de escenarios del árbol original que es infactible bajo la solución encontrada a partir de la reducción realizada, siendo así posible determinar cuan robusta es la solución alcanzada bajo el método de reducción en estudio.

Para el caso del algoritmo fast-forward, el promedio de infactibilidad se encuentra por debajo de un 10% – es decir, la solución alcanzada a partir de las instancias reducidas es factible de llevar a cabo en un 90% del árbol de escenarios original – mientras que en el caso del algoritmo k-means modificado, ésta se eleva hasta un 15% promedio, por lo que se deduce en primera instancia que la selección de escenarios llevada a cabo bajo el primer algoritmo presenta

una ventaja significativa a la hora de determinar el subconjunto que mejor representa la distribución estocástica original.

En base a los resultados presentes en las figuras 6.8 y 6.9 obtenidos bajo la mejor configuración encontrada en la sección 6.2, es posible ver como en la medida que aumenta el tamaño de la instancia se tiene una disminución progresiva y sostenida del índice de infactibilidad, hecho que está directamente correlacionado con la cantidad de información disponible (y seleccionada) en cada una de las reducciones aplicadas. Así, en la medida que el árbol de escenarios original contenga un mayor número de elementos, se dispone de una mayor cantidad de información, pero también más cercana entre los mismos escenarios, lo que ocasiona que el seleccionar escenarios de instancias masivas conlleve a una gran representación de sus vecinos más cercanos, y así, en la medida que estos se seleccionen de forma adecuada, se podrá representar el universo original de forma altamente satisfactoria.

Esto se observa de forma significativa al observar los resultados de instancias pequeñas (menores a los 100 escenarios), donde el llevar a cabo reducciones no conlleva un gran rendimiento, ya que se obtienen porcentajes de infactibilidad del orden de un 20%, asociados a tiempos de resolución que no son significativamente menores a los originales, ocasionando así que el trade-off inherente entre calidad y tiempo de resolución no sea óptimo, pues en el mejor de los casos (instancia de 50 escenarios), es posible disminuir en un 50% el tiempo de resolución, obteniéndose solamente 2 escenarios infactibles, sin embargo, la solución original requiere aproximadamente 6 minutos para su resolución, por lo que en la práctica la mejora no es significativa.

Situación contraria se observa en las instancias masivas, donde se obtienen resultados de gran calidad en tiempos de resolución significativamente inferiores en relación a la instancia original. Así, es posible observar que en instancias superiores a los 300 escenarios, una reducción a un 75% acarrea en promedio un error de apenas un 1,5% promedio, e incluso reducciones de hasta un 25% conllevan a infactibilidades menores a un 10% promedio, hecho que se ve ampliamente favorecido por tiempos de resolución significativamente inferiores, encontrándose por debajo de un 10% del tiempo originalmente necesario para alcanzar una solución.

Instancia	Reducción	Diferencia % F.Objetivo Original	Paralelo 4 núcleos		Infactibilidad [%]
			Tiempo de resolución [min]	% c/r a instancia original	
10	7	3,74%	0,511333333	45,411%	20,000%
	5	1,06%	1,031166667	91,578%	20,000%
	3	3,74%	0,511333333	45,411%	20,000%
20	15	3,97%	1,187	53,348%	10,000%
	10	3,30%	1,309333333	58,846%	20,000%
	5	3,97%	1,373333333	61,723%	25,000%

Instancia	Reducción	Diferencia % F.Objetivo Original	Paralelo 4 núcleos		Infactibilidad [%]
			Tiempo de resolución [min]	% c/r a instancia original	
30	23	1,63%	1,651166667	48,294%	6,667%
	15	0,09%	1,075833333	31,466%	20,000%
	8	1,35%	0,4599999	13,454%	20,000%
40	30	0,06%	2,094	46,554%	5,000%
	20	2,14%	1,364	30,325%	7,500%
	10	2,06%	0,6811	15,142%	10,000%
50	38	1,15%	2,862166667	50,919%	4,000%
	25	2,94%	1,840833333	32,749%	16,000%
	13	1,78%	1,077333333	19,166%	16,000%
107	81	4,89%	5,898	39,473%	3,738%
	54	4,75%	3,456	23,129%	9,346%
	27	2,20%	1,8845	12,612%	13,084%
110	83	3,79%	5,26587	31,038%	3,738%
	55	0,30%	4,2645	25,136%	9,346%
	28	6,66%	1,8965	11,178%	13,084%
216	162	3,37%	35,7891	52,369%	6,173%
	108	3,65%	24,529	35,893%	6,481%
	54	2,30%	3,437	5,029%	11,574%

Figura 6.9: Resultados reducción de escenarios bajo algoritmo fast-forward

Complementario a lo anterior, se observa que en instancias superiores a los 500 escenarios, el llevar a cabo reducciones que se encuentren entre los 400 y 500 escenarios conlleva una pérdida de factibilidad prácticamente nula, siendo en el peor de los casos de un 1%. En particular, se observan resultados ampliamente satisfactorios al momento de aplicar la técnica de reducción en la instancia de 1000 escenarios, donde reducciones de hasta un cuarto del total original conllevan a una pérdida de factibilidad del orden de un 1% (10 escenarios infactibles), con un correspondiente tiempo de resolución de apenas un 15% del caso original.

Esto marca una primera aproximación respecto del posible número “suficiente” de escenarios a utilizar dentro del problema en estudio, buscando optimizar el trade off inherente a la calidad de la solución y tiempo de resolución involucrado, estudio que será llevado a cabo en profundidad en el capítulo 7.

Instancia	Reducción	Diferencia % F.Objetivo Original	Paralelo 4 núcleos		Infactibilidad [%]
			Tiempo de resolución [min]	% c/r a instancia original	
343	258	0,61%	50,263	7,491%	2,332%
	172	0,05%	2,734666667	0,408%	4,956%
	86	2,11%	6,701833333	0,999%	6,997%
	34	1,21%	2,734666667	0,408%	7,580%
512	384	0,37%	602,455	70,672%	1,563%
	256	3,91%	54,489	6,392%	4,688%
	128	2,68%	51,0795	5,992%	6,250%
	51	2,03%	3,393166667	0,398%	8,203%
729	547	0,30%	697,15	67,975%	0,686%
	365	4,09%	622,456	60,692%	7,682%
	183	0,81%	31,4565	3,067%	11,660%
	73	0,96%	5,596833333	0,546%	13,717%
1000	750	0,01%	912,45	67,365%	0,100%
	500	0,01%	643,21	47,488%	0,200%
	250	0,83%	201,23	14,857%	1,000%
	100	0,96%	13,367	0,987%	8,600%
	50	1,55%	3,380333333	0,250%	15,000%
	10	1,89%	0,654166667	0,048%	16,400%

Figura 6.10: Resultados reducción de escenarios para instancias masivas bajo algoritmo fast-forward

## 6.6 Comparaciones

En base a los resultados obtenidos, es posible determinar que la mejor configuración encontrada para la resolución del modelo en estudio corresponde a la implementación en paralelo del algoritmo Progressive Hedging, utilizando las mejoras propuestas en la sección 4.2, implementación que supera de forma significativa a la formulación extendida resuelta bajo métodos de optimización clásica, así como a su par serial.

El rendimiento alcanzado en la medida que aumenta el número de procesadores disponibles en la unidad de hardware utilizada experimenta un incremento decreciente, principalmente debido a los tiempos de comunicación y coordinación involucrados a la hora de utilizar esta técnica, tal como se observa en la sección 6.3, así, es preciso utilizar un número de núcleos acorde al tamaño del problema que se está abordando.



La reducción de escenarios bajo el algoritmo fast-forward se erige como la indicada para seleccionar de forma óptima el subconjunto de escenarios más representativo con respecto al árbol original, debido a que mantiene su distribución de gran manera y del mismo modo permite alcanzar rendimientos altamente superiores en cuanto a tiempos de cómputo involucrados, sin comprometer la calidad de la solución de forma significativa.

## Capítulo 7

# Calidad de la solución: Modelo

## Determinístico vs Estocástico

Uno de los aspectos más relevantes a la hora de justificar la utilización de técnicas de optimización estocástica que involucran un alto costo computacional, debido a la gran cantidad de restricciones y variables que se requieren dentro del modelo a la hora de trabajar con un elevado número de escenarios, corresponde a que la calidad de la solución obtenida es más robusta y de “mejor nivel” en comparación a aquella que se obtendría al utilizar el modelo determinístico clásico (utilizando por ejemplo el valor esperado para el precio de cada producto en cada una de las etapas de decisión).

Dicha solución será más robusta debido a que ésta toma en cuenta un gran abanico de posibilidades en cuanto a la realización de las variables aleatorias que han sido incluidas dentro del modelo, considerando también aquellos casos extremos (cisnes negros o eventos de cola) cuya probabilidad de ocurrencia es bastante baja pero no nula, lo que permite que las decisiones óptimas a tomar consideren la posibilidad de experimentar uno de estos escenarios, información que se pierde en la medida que se reduce el número de escenarios con el que se está trabajando, hasta prácticamente omitirla en el caso determinístico.

Por otra parte, el concepto de “calidad” de la solución obtenida presenta mayores dificultades en cuanto a su definición, puesto que se deben determinar una o más métricas que permitan comparar las soluciones obtenidas bajo las dos formulaciones (estocástica y determinística), de tal forma que no exista subjetividad al respecto. Una forma clásica de medir la calidad de la solución obtenida bajo el modelo determinístico consiste en determinar si dicha solución es factible en todos los escenarios en estudio (desde aquellos que se utilizaron para determinar el valor esperado de la(s) variable(s) aleatoria(s) incluida(s)), siendo así claramente una solución de baja calidad aquella que presente un menor número de escenarios factibles, es decir, que cubra de forma deficiente las posibles realizaciones futuras de la variable estocástica. Sin embargo, este enfoque no es suficiente para demostrar de forma cuantitativa la diferencia de calidad entre las soluciones obtenidas, por lo que se hace necesario contar con una metodología complementaria que permita comparar las diferentes soluciones obtenidas.

Del mismo modo, uno de los desafíos más complejos al abordar problemas de programación estocástica se encuentra en determinar el número “óptimo” de escenarios a utilizar como input para resolverlo, puesto que existe un fuerte trade-off entre la calidad de la solución obtenida y el tiempo necesario para obtenerla, así como encontrar (en caso de existir) el punto de corte en el cual el problema estocástico comienza a superar de forma significativa a la solución

encontrada bajo el modelo determinístico, es decir, a partir de qué tamaño de árbol de escenarios se hace imperante trabajar con el modelo completo, debido a los resultados que ambos modelos arrojan.

## 7.1 Metodología

La metodología utilizada consiste en un enfoque de simulación donde se busca comparar el desempeño entre dos algoritmos/modelos (llámense A y B) a la hora de resolver el problema bajo ambas formulaciones, determinándose además la factibilidad de dicha solución dentro del universo de escenarios con los que se está trabajando.

Los modelos que se utilizarán durante el trabajo de la presente tesis corresponden al modelo determinístico de N-etapas (A) que utiliza como parámetro el valor esperado de las variables aleatorias calculado a partir del conjunto original de escenarios para introducirlas como elementos determinísticos a lo largo de las N-etapas consideradas y al modelo estocástico (B) el cuál se resolverá utilizando el algoritmo de Progressive Hedging con tal de resolver el universo de escenarios considerados, utilizando como complemento dos técnicas de reducción de escenarios testeadas en una serie de problemas similares (k-means y algoritmo de reducción fast-forward [3]), además de implementar la utilización de programación distributiva o en paralelo con tal de explotar la estructura de descomposición del algoritmo, obteniéndose mejores desempeños.

Junto a lo anterior, se utilizará la mejor configuración encontrada en la sección (6.2) para el algoritmo PH, debido a que demostró presentar el mejor rendimiento para el problema en estudio. Así, se realizará la linearización de los términos de penalización asociados a las variables continuas, se fijarán variables enteras al presentar convergencia durante 3 iteraciones consecutivas, se trabajará con un valor de rho fijo de 0.001, con un punto de corte de convergencia de 0.01 y se comenzará aplicando un gap de gran tamaño (20%) en las primeras iteraciones hasta disminuirlo a un 0.05% en las finales, con tal de mejorar la velocidad de convergencia del algoritmo.

Sean:

$N$  = Número total de iteraciones a realizar.

$I$  = Número de Iteración.

$T$  = Time step.

$FStSolA_{it}$  = First Stage Solution del algoritmo A en la iteración i, Time-step t.

$FStSolB_{it}$  = First Stage Solution del algoritmo B en la iteración i, Time-step t.

$EStSolA_{its}$  = Solución Evaluada sobre escenario s del algoritmo A en la iteración i, Time-step t.

$EStSolB_{its}$  = Solución Evaluada sobre escenario s del algoritmo B en la iteración i, Time-step t.

$FStSolS_{it}$  = First Stage Solution del Escenario seleccionado en la iteración  
i, Time-step t

$P0$  = Valor de las variable aleatorias en el nodo raíz (Precios de los  
productos)

El algoritmo utilizado para llevar a cabo la simulación es el siguiente:

---

### Algoritmo de Simulación

---

```
For (I=1:N){  
  For(T=1:4){  
    Step 0: Se crea un árbol de escenarios a partir del algoritmo generador  
      de escenarios señalado en la sección (3.4).  
    Step 1: Resolver el modelo utilizando los algoritmos A y B.  
    Step 2: Registrar  $FStSolA_{it}$  y  $FStSolA_{it}$   
    Step 3: Seleccionar uno de los escenarios del árbol original al azar  
      (basado en las probabilidades de realización del escenario) y  
      resolverlo. Registrar  $FStSolS_{it}$ .  
    Step 4: Evaluar  $FStSolA_{it}$  y  $FStSolA_{it}$  sobre el escenario seleccionado,  
      obteniéndose  $EstSolA_{its}$  y  $EstSolB_{its}$ . Se comprueba si la  
      solución evaluada es factible.  
    Step 5:  $P0 \leftarrow P1$  (del escenario seleccionado), Actualizar el estado del  
      bosque  
  }}
```

---

Bajo este esquema, se observa que el problema original se resuelve avanzando 4 periodos de tiempo (Time-steps) dentro de cada una de las iteraciones del algoritmo, simulando así la resolución del problema original, cuyo horizonte de planificación es justamente de 4 periodos, lo que además permite comparar directamente las soluciones obtenidas bajo los algoritmos A y B, al evaluarlas sobre un escenario del universo original de trabajo seleccionado de forma totalmente aleatoria, determinándose en primera instancia si dicha solución es factible, y en caso de que ambas lo sean, comparar de forma cuantitativa cuál de ellas es mejor a través del valor de su función objetivo y las diferencias en tiempos de resolución que se registren.

El número de replicaciones deberá ser tal que se puedan realizar test T-student de diferencias de media entre dichas soluciones, de tal forma de poder comparar la “ganancia” que se debiese registrar al utilizar la formulación estocástica vs la determinística en la medida que el universo de escenarios aumente pues a mayor incertidumbre, peor rendimiento presenta el modelo determinístico y mejor desempeño para el caso estocástico debido a la pérdida de

información del primero. Así, para los experimentos realizados, se utilizan 20 replicaciones en cada instancia de trabajo.

Gracias a esta metodología, será posible estimar el valor real y cuantitativo de utilizar un modelo multi-etápico estocástico completo, detallado y realista, con todas las complejidades a las que se enfrenta la industria forestal, además de encontrar una cota estimada respecto al número de escenarios a utilizar con tal de obtener el mejor rendimiento en cuanto a calidad de solución vs tiempo de resolución, puesto que el valor de la solución tenderá a estabilizarse en la medida que se aumente el número de escenarios.

## 7.2 Implementación

### 7.2.1 Software Utilizado

La metodología señalada fue implementada principalmente utilizando el lenguaje de programación JAVA, creándose un script que genera una llamada al software MATLAB para llevar a cabo la creación de cada una de las instancias y transformarlas al formato necesario para utilizarlas posteriormente durante la llamada a los software de optimización.

Una vez creadas las instancias necesarias para llevar a cabo los experimentos, se procede a la lectura de cada uno de los árboles de escenarios creados - outputs del script Matlab presentado en la sección (3.4). Se lleva a cabo el cálculo y creación de la instancia para el Algoritmo A – modelo determinístico que utiliza valores esperados como inputs – así como del Algoritmo B, utilizando el método de reducción de escenarios que el usuario estime conveniente para llevar a cabo la resolución bajo el algoritmo PH.

El script de JAVA realiza una llamada al módulo de optimización PYOMO en Python para resolver el Modelo determinístico y una llamada al módulo de optimización RUNPH del paquete Coopr Sandia [8] para obtener la solución bajo el algoritmo de Progressive Hedging.

Ambas soluciones se registran en carpetas específicas del computador en uso en formato de texto y se procede a la lectura de éstas. El script principal selecciona uno de los escenarios del árbol original de forma aleatoria basándose en las probabilidades de realización de cada uno de estos y evalúa las soluciones de los algoritmos A y B sobre dicho escenario, determinándose además la factibilidad de cada una de éstas.

Junto a lo anterior, JAVA realiza nuevamente una llamada al módulo PYOMO con tal de obtener la solución “real” del escenario seleccionado, obteniéndose así su valor óptimo, lo que permitirá comparar las soluciones evaluadas con la que se obtendría en caso de que dicho escenario se manifestara.

Finalmente, el programa registra el valor de las variables aleatorias en el nodo de la primera etapa (no el raíz) del escenario seleccionado, para utilizarlo como punto de partida para la siguiente iteración y generar el llamado al script de MATLAB generador de escenarios con dicho input, lo que permite crear un nuevo árbol de escenarios desde una etapa posterior a la original, avanzando un periodo en el tiempo.

El esquema de trabajo es el siguiente:

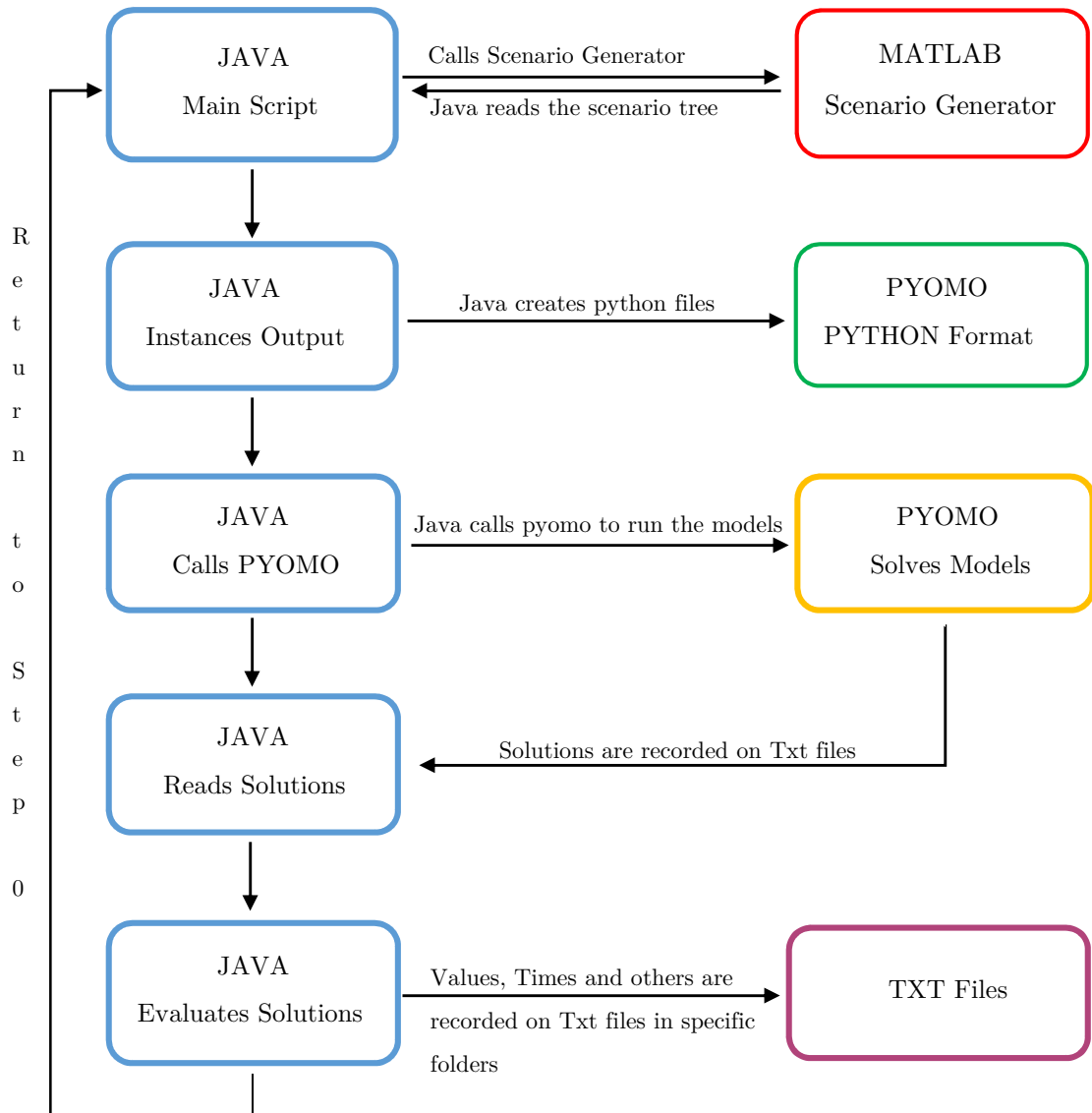


Figura 7.1: Esquema de trabajo utilizado para la implementación de la metodología de simulación

### 7.2.2 Hardware y software

Los experimentos cuyo árbol de escenarios contiene menos de 500 escenarios, fueron llevados a cabo en un computador de escritorio, mientras que aquellos de carácter masivo se implementaron en una estación de trabajo multi-núcleo, con las siguientes características:

- **Computador de Escritorio**  
**Sistema Operativo:** Windows 8.1  
**Memoria RAM:** 16 GB DDR4  
**CPU:** Intel I7-4700 2.4 GHZ  
**Solver:** CPLEX 12.4 32 bits  
**Scripts:** MS-DOS, PYTHON 2.7 64 bits, JAVA 64 bits  
**Middleware:** Coopr, Sandia National Labs [8]
  
- **Estación de trabajo multi-núcleo**  
**Sistema Operativo:** Linux  
**Memoria RAM:** 64 GB  
**CPU:** AMD Athlon 2.2GHZ  
**Solver:** CPLEX 12.4 32 Bits  
**Scripts:** BASH, PYTHON 2.7, JAVA  
**Middleware:** Coopr, Sandia National Labs [8]

El resto de elementos son análogos a los señalados en la sección (6.2).

### 7.3 Modelo de optimización Alternativo

Al observar el modelo originalmente planteado en la sección (3.5), se observa que bajo la metodología de simulación planteada, es posible que la solución del modelo determinístico (o del estocástico en caso de no converger) presente infactibilidad al ser evaluada dentro de un escenario seleccionado al azar, puesto que dicho modelo resuelve un “árbol de escenarios” único y aproximado, por ende, es factible que en caso de evaluar dicha solución en cierto set de escenarios, no se respete la no anticipatividad, o de forma más común, se violen las cotas de producción dada la fuerte restricción asociada a la demanda por productos. Así, a modo complementario, también se trabajará con una modificación del modelo original, agregando dos nuevas variables de holgura asociadas a dos nuevos parámetros de penalización, parámetros que representarán el costo por producir menos que lo pactado con los proveedores y la pérdida por vender a menor precio en caso de producir más de la demanda experimentada.

Dado lo anterior, se agregan los siguientes elementos al modelo expuesto en la sección (3.5):

- **Parámetros**

$\alpha$  = Factor de Penalización por producir menos que la cota inferior  
 $\beta$  = Factor de Penalización por producir por sobre la cota superior

- **Variables de Decisión**

$\bar{H}_{p,e}^{t,s}$  = Variable de holgura, cantidad de producto p vendida en el periodo t en la salida e bajo el escenario s por sobre la dda. superior [M<sup>3</sup>]

$\underline{H}_{p,e}^{t,s}$  = Variable de holgura, cantidad de producto p vendida en el periodo t en la salida e bajo el escenario s por bajo la dda. inferior [M<sup>3</sup>]

Luego, la restricción respecto a las cotas de producción queda:

6. Cotas de producción

$$\underline{Z}^{p,t,s} - \sum_{e \in E} \underline{H}_{p,e}^{t,s} \leq \sum_{e \in E} z_{p,e}^{t,s} \leq \bar{Z}^{p,t,s} + \sum_{e \in E} \bar{H}_{p,e}^{t,s} \quad \forall p \in P, t \in T, s \in S$$

Finalmente, la función objetivo varía su primer término  $T_1^s$  :

$$T_1^s = \sum_{e \in P} \sum_{p \in P} \sum_{t \in T} P_{e,p}^{t,s} (z_{p,e}^{t,s} - \bar{H}_{p,e}^{t,s}(1 - \beta) - \underline{H}_{p,e}^{t,s}(1 - \alpha)) \quad \forall s \in S$$

Gracias a estas modificaciones, la única posibilidad que existe para que el modelo sea infactible, radica en la violación de las restricciones de no anticipatividad, permitiendo así evaluar todas las posibles soluciones obtenidas bajo el enfoque de simulación señalado.

## 7.4 Resultados

En esta sección se presentan los resultados obtenidos durante el trabajo de tesis, de acuerdo a la metodología de simulación señalada en la sección de implementación (3.4). En este caso se persigue realizar una comparación directa entre la solución determinística – donde se utiliza el valor esperado de las variables aleatorias por cada etapa como parámetros conocidos – con respecto a la implementación estocástica bajo el algoritmo PH, utilizando la mejor configuración encontrada en la sección (6.2). Tal como se señala en la sección (7.1), se pretende llevar a cabo una comparación objetiva respecto de la calidad de la solución encontrada por cada una de las formulaciones, al evaluar dicha solución sobre un escenario elegido al azar del universo original que conforma la instancia en estudio, pudiéndose así determinar la diferencia/ganancia que se obtiene al utilizar el modelo estocástico – con toda la complejidad que esto requiere – en relación a la aproximación determinística.



Las instancias de prueba experimentales consisten en las presentadas en la sección (6.1), agregándose además instancias de 200, 300, 400 y 500 escenarios. Todas contemplan el total de bosques de la instancia original y se han testeado con todas las combinaciones posibles de técnicas de mejora para el caso estocástico. Del mismo modo, se han utilizado dos formulaciones del modelo de optimización original, con tal de permitir un estudio más amplio del problema en estudio, asemejándose más a la realidad que experimenta la industria forestal. Una de éstas corresponde a la presentada en la sección (3.5) (original), mientras que la segunda implementa las modificaciones señaladas en la sección (7.3), agregándose las variables de holgura y factores de penalización según la cantidad vendida por cada periodo y producto por salida.

La metodología para llevar a cabo los experimentos consistió en una reproducción automática de cada una de las instancias mediante una serie de scripts de MS-DOS/BASH (según el sistema operativo), los cuales se encargan de generar las llamadas al paquete de optimización PYOMO y los diferentes scripts de PYTHON involucrados, todo esto coordinado por un script principal programado en JAVA. Luego, cada uno de los resultados se almacena en un formato específico de texto en su propia carpeta individual con toda la información necesaria para llevar a cabo la corrida de la instancia. Por ende, el resultado final de los experimentos corresponde a una macro carpeta que contiene subcarpetas específicas para cada una de las instancias, identificadas por nombres claves asociados a la replicación que se está almacenando, el tipo de algoritmo que se está aplicando, entre otras estadísticas, de tal forma que es un proceso inequívoco y automatizado.

#### **7.4.1 Problema General**

Medir y comparar la calidad de la solución obtenida por formulaciones estocásticas y determinísticas es uno de los principales objetivos que debe alcanzarse a la hora de sustentar y avalar el uso de un modelo estocástico complejo, cuyo principal elemento corresponda al tratamiento de una serie de árboles de escenarios, lo que claramente dificulta la resolución del mismo y por ende, los tiempos y costos involucrados en su realización son mayores a los que se experimentaría bajo un modelo aproximado. Sin embargo, las soluciones obtenidas por el modelo determinístico carecen de toda la información que se implementa de forma natural bajo el concepto de árboles de escenarios, lo que genera que éstas no tomen en cuenta realizaciones factibles de las variables aleatorias en juego, sobre todo respecto a los eventos de cola o extremos, que usualmente presentan una probabilidad de ocurrencia tan baja que no son considerados a la hora de realizar el cálculo de los precios esperados, sesgando así al modelo hacia el promedio de posibilidades.

Así, el desafío consiste en determinar la calidad de la solución obtenida bajo cada formulación, para luego demostrar la ganancia obtenida al utilizar el modelo estocástico completo, pudiendo así señalar la utilidad de llevar a cabo realmente la resolución del modelo con todas sus complejidades.

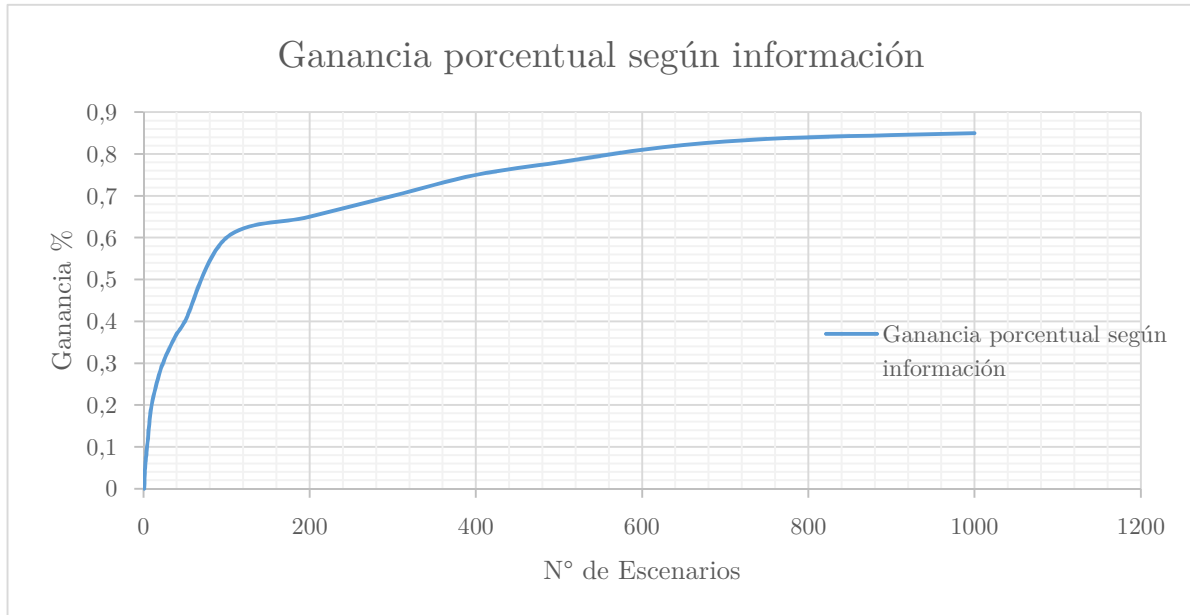


Figura 7.2: Ejemplo del comportamiento (teórico) esperado por parte de la ganancia de realizar la formulación estocástica en comparación a un modelo aproximado.

Junto a esto, otro desafío corresponde a lograr determinar el tamaño óptimo del set de escenarios con el cual se pretende trabajar, en términos de calidad y velocidad de resolución, debido a que la solución debiese experimentar – a partir de algún número de escenarios – un proceso de estabilización, lo que conlleva a concluir que el agregar nueva información (aumentar el número de escenarios) no permite obtener mejoras/ganancias a la hora de determinar las decisiones óptimas que deben llevarse a cabo.

#### 7.4.2 Simulación

Los resultados se organizan de tal manera que se presenta el número de escenarios que componen la instancia en estudio, indicando cuál fue el escenario escogido en cada iteración para realizar el “time-step”, junto a las soluciones de ambas formulaciones, la evaluación de éstas en el escenario seleccionado y la diferencia que presentan (en negativo en caso de ser mejor la solución estocástica). Cabe recordar que sólo se presenta la solución de la primera etapa, mostrándose el valor de los ingresos obtenidos en dicho periodo bajo cada una de las formulaciones.

La configuración del algoritmo PH que fue utilizada corresponde a aquella de mejor rendimiento encontrada en la sección (6.2), utilizándose en primera instancia 4 núcleos para llevar a cabo la resolución en paralelo de las instancias. Del mismo modo, se señala (en caso de ser pertinente) el uso de algoritmos de reducción, indicando tipo (k-means o fast-forward) y el número de escenarios al que se redujo la instancia original.

En base a los resultados obtenidos (ver Anexos sección (11.4)), se observa como a medida que se aumenta el número de escenarios, la solución obtenida por el modelo estocástico resuelto bajo el algoritmo de Progressive Hedging comienza a ganar terreno a la hora de compararla con aquella obtenida bajo el modelo determinístico, esto debido a que la presencia de una mayor cantidad de información para el modelo permite que dicha solución sea más robusta y por lo tanto, al ser aplicada/evaluada sobre alguno de los escenarios del set original, presenta un mejor rendimiento en relación a la que se obtiene de resolver el modelo esperado aproximado.

Esto va de la mano con el comportamiento esperado de la técnica, puesto que al tomar en consideración un mayor abanico de posibilidades, el algoritmo determina una solución que hace frente de forma óptima (o la mejor posible) a todos los escenarios en conjunto, por ende, es de suponer que al evaluar dicho valor sobre cualquiera de los escenarios trabajados, responda de mejor forma en comparación a una solución obtenida bajo una aproximación determinística, que si bien para instancias pequeñas (menores a 20 escenarios, salvo ciertos casos o instancias puntuales) es totalmente justificada, tanto por tiempo como por calidad de la misma, comienza a dejar mucha información de lado en la medida que el tamaño de la instancia crece, lo que claramente no es preferible para una empresa que busque una planificación óptima, pero al mismo tiempo capaz de librar de la mejor forma posible todo tipo de eventos futuros que puedan experimentarse, por muy poco probables que estos sean.

A modo de ejemplo en la figura 7.3 se observan parte de los resultados obtenidos para la instancia de 50 escenarios, donde se puede ver claramente que la diferencia a favor de la formulación estocástica comienza a ser significativa, obteniéndose del orden de magnitud (o cercanos) de 10.000 USD, simplemente por llevar a cabo el desarrollo del modelo estocástico, que si bien toma más tiempo en resolverse, este ni siquiera supera los 3 minutos en promedio, lo que es más que razonable para llevar a cabo la toma de decisiones a nivel táctico.

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.827.194	22,2843	1.792.799	1.826.273	183,84	1.825.793	-32.994
1.715.858	18,5235	1.622.925	1.719.915	176,95	1.719.055	-96.130
1.782.929	21,2335	1.748.252	1.789.917	176,21	1.789.108	-40.856
1.858.176	18,9861	1.818.697	1.863.329	179,25	1.863.200	-44.503
1.827.194	18,9862	1.741.332	1.826.273	186,75	1.826.106	-84.774
1.812.137	23,112	1.743.959	1.818.972	177,64	1.818.855	-74.896
1.760.068	23,2321	1.690.706	1.762.043	187,35	1.761.924	-71.218
1.770.644	16,8762	1.730.463	1.775.577	183,55	1.774.846	-44.383
1.827.194	14,5637	1.748.503	1.826.273	176,22	1.826.226	-77.723
1.775.425	22,4353	1.703.057	1.777.964	187,21	1.777.795	-74.738

Figura 7.3: Extracto de resultados de simulación instancia 50 escenarios

Así, es posible observar en la figura 7.4 como la ganancia (vista como valor negativo en este caso) obtenida por el modelo estocástico completo aumenta de gran manera en la medida

que se incrementa el número de escenarios desde el caso basal determinístico, y luego, comienza a disminuir el incremento porcentual una vez alcanzados los 400 escenarios, observándose una tendencia a que el modelo estocástico alcance un estado estacionario en cuanto a su solución/ganancia, lo que está acorde con lo previsto, dado que al superar cierto umbral de información, el aporte de un nuevo escenario será decreciente en términos marginales, obteniéndose una gran solución, pero que en términos de trade-off de tiempo vs calidad no presenta el mejor rendimiento.

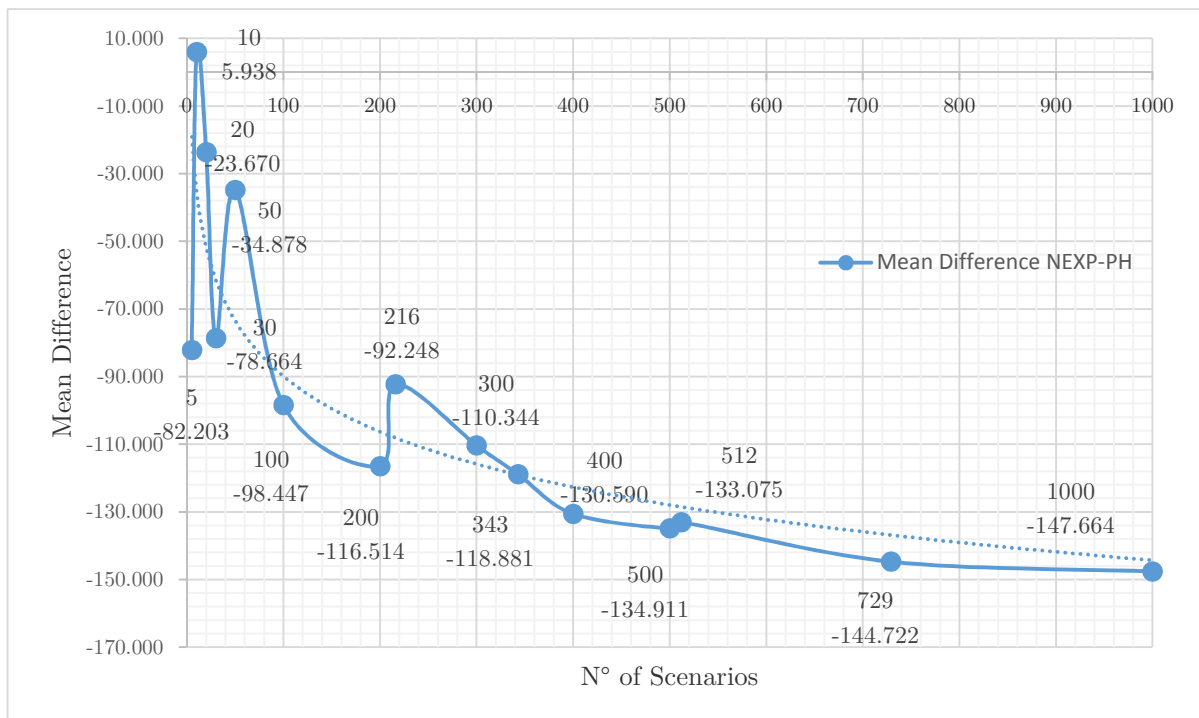


Figura 7.4: Diferencia promedio entre las soluciones evaluadas bajo el modelo determinístico y el estocástico según el número de escenarios considerados

De forma complementaria, al observar el comportamiento de la solución obtenida bajo el modelo determinístico de valor esperado, se puede ver claramente en la figura 7.5 la relación con lo señalado en la figura anterior, puesto que a partir de los 400 escenarios se observa un comportamiento estacionario (salvo pequeños baches cerca de los 500 escenarios) por parte de la solución obtenida, lo que da cuenta de que el número óptimo de escenarios a trabajar para el problema en estudio se encuentra entre los 400 – 500 escenarios como primer punto de corte, puesto que la solución no experimenta mayores variaciones en cuanto a su calidad, mientras que los tiempos de resolución sí que se ven afectados de forma considerable, llegando a ser de hasta 400 minutos más (80% superiores) al resolver instancias de tamaños mayores a 1000 escenarios.

Del mismo modo, el resolver instancias masivas (superiores por sobre los 600 escenarios) comienza a requerir la presencia de mejores componentes de hardware, volviéndose casi impracticable el resolver dichos problemas en computadores de escritorio de uso cotidiano (no prohibitivos en precio) en tiempos razonables, por lo que comienza a ser necesario el tener acceso a estaciones de trabajo profesionales y multi-núcleo.

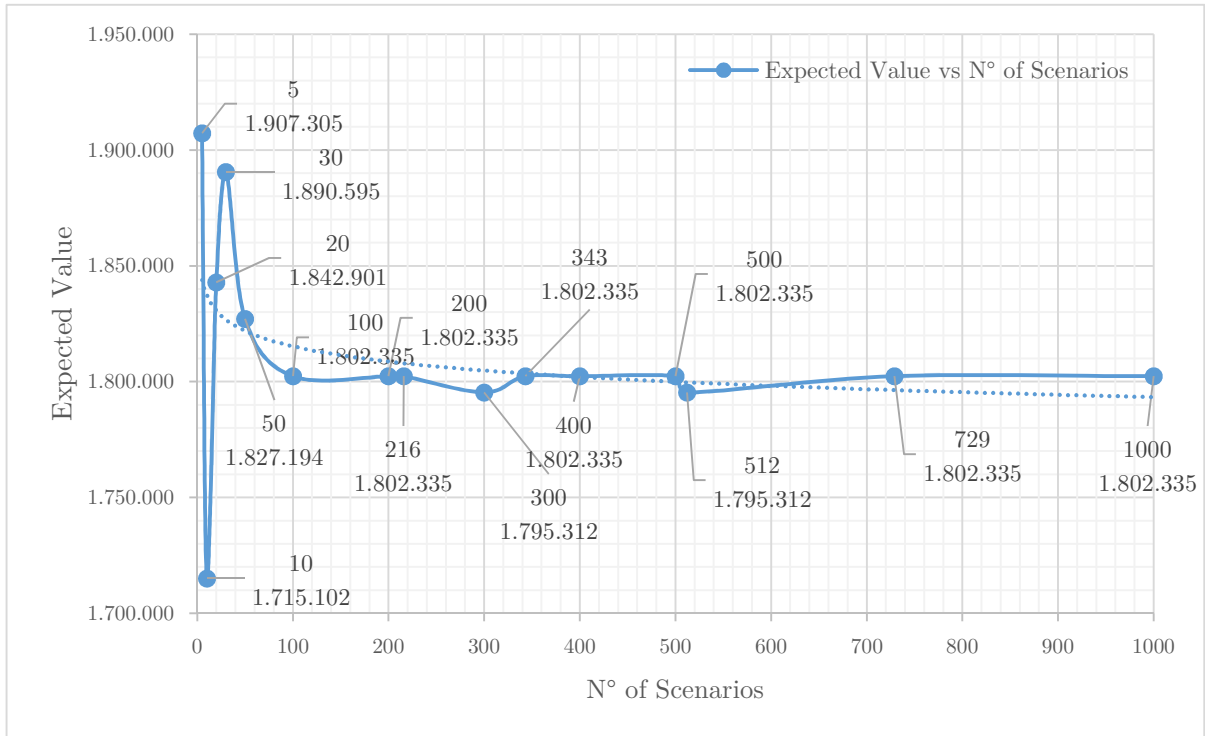


Figura 7.5: Valor de la solución del modelo determinístico en función del número de escenarios trabajados

Por lo anteriormente planteado es posible concluir que la implementación estocástica se justifica en la medida que aumenta el tamaño del árbol de escenarios que se está utilizando para simular la realidad de los mercados de la industria forestal, debido a que al tomar en cuenta todas las posibilidades del universo de escenarios, éste cuenta con un nivel de información mayor al que se reduce la formulación determinística, la cual representa finalmente una simplificación de la realidad. Así, en la medida que aumente el número de escenarios que se incluyen en la instancia, la solución obtenida por el modelo estocástico presentará una calidad mayor, en términos de que al evaluarla en cualquier escenario del árbol original, ésta se desempeña de mejor manera, en comparación al caso determinístico.

Sin embargo, lo anterior no implica que la mejor solución a la pregunta de cómo abordar el problema de forma óptima consista precisamente en aumentar de forma continua el número de escenarios que se incluyen en el árbol original con tal de obtener una solución de mejor calidad. Esto pues en la medida que se aumente el número de escenarios a trabajar, también se incrementará de forma considerable el tiempo requerido para obtener una solución del problema (tal como se vio en las secciones (6.2) y (6.3)), principalmente en cuanto a los tiempos de inicialización y comunicación entre los múltiples procesadores (suponiendo programación distributiva y/o en paralelo).

Es así como se vuelve vital el determinar cuando el problema y modelo en estudio alcanzan un estado estacionario, momento en el cual el agregar mayor cantidad de información se vuelve contraproducente al analizar el trade-off entre calidad y velocidad de resolución, puesto que el

aporte a la calidad de la solución por agregar un escenario se vuelve decreciente en términos marginales, mientras que el agregar un nuevo escenario plantea mayores costos computacionales para inicializar el algoritmo, así como un mayor overhead a la hora de desarrollar la comunicación entre los diversos procesadores/computadores de la estación de trabajo, lo que se traduce en implementaciones poco eficientes y con largos tiempos de espera para alcanzar una solución que no justifica su mejora.

## Capítulo 8

# Conclusiones

El trabajo realizado en la presente tesis incluye la implementación completa de un modelo forestal real que abarca un total de 17 bosques, además de decisiones relacionadas a la construcción y mejora de caminos en la red interna, más la posibilidad de llevar a cabo inventario de los productos (3 tipos distintos) entre periodos. Así, su complejidad supera de forma bastante amplia trabajos similares de los últimos años, donde se habían establecido una serie de simplificaciones con tal de que el problema original se tornara abordable bajo la tecnología de la época.

Se reformularon y modificaron ciertas restricciones en relación al modelo original, condensando así en éstas las mejoras de fortalecimientos y liftings que habían sido implementadas en trabajos precedentes, alcanzándose un mayor rendimiento de resolución.

La generación de escenarios sustentada bajo un modelo matemático permite la obtención de resultados de calidad, una de las grandes carencias que se presenta normalmente en trabajos similares, donde estos se recrean a través de métodos poco exactos y sin bases teóricas sólidas. Se ha llevado a cabo la implementación en software de una metodología de generación de escenarios, creándose instancias de cientos y miles de realizaciones en cuestión de minutos, permitiendo así la experimentación de instancias masivas, con resultados fundamentados y objetivos.

El trabajo realizado en relación a la implementación de las rutinas de generación de escenarios permite replicar dicho modelo en problemas de características similares, siendo así aplicable en trabajos futuros.

Se demuestra la utilidad del algoritmo Progressive Hedging a la hora de resolver problemas de carácter estocástico multietápicos, con una gran cantidad de variables (lineales y binarias) y restricciones, bajo un universo de escenarios de gran magnitud.

Se presenta una documentación detallada y extensa respecto a las mejoras y ajustes que pueden realizarse sobre el algoritmo con tal de mejorar el desempeño de éste.

Los experimentos realizados permiten alcanzar una configuración óptima del algoritmo PH, configuración que permite obtener resultados superiores en términos de rendimiento en relación a una implementación clásica de modelamiento y solución bajo el solver CPLEX, principalmente en términos de tiempo de resolución.

La mejor configuración encontrada para el modelo en estudio consiste en utilizar un factor de penalización pequeño y fijo, bajo un número limitado de iteraciones (10), linealización del término proximal en 2 tramos principales, junto a una estrategia de fijación de variables binarias al alcanzar la convergencia durante 3 iteraciones consecutivas, además de un enfoque de mipgap descendente desde un valor de 20% hasta un 0.001% según las iteraciones realizadas.

Instancias con un gran número de escenarios han sido testeadas y resueltas de forma satisfactoria bajo la implementación del algoritmo PH. Se ha aumentado así el total de escenarios trabajado en relación a trabajos precedentes, desde un máximo de 216 hasta 1000 escenarios en el presente trabajo, es decir, 3,83 veces la cantidad manejada anteriormente, tomando en cuenta que el trabajo actual presenta 17 bosques en vez de 1.

La implementación serial bajo la formulación extendida en CPLEX sólo logra ser superior en instancias menores a 20 escenarios, debido a que el tamaño del problema es bastante pequeño y por ende el algoritmo PH no logra superarlo. Sin embargo, al trabajar con instancias masivas (superiores a 200 escenarios), la formulación extendida es incapaz de resolverse en tiempos razonables, e incluso no es posible cargarla en computadores de escritorio convencionales.

Se desarrolla una implementación en paralelo utilizando los componentes y paquetes principales con los que cuenta el software Coopr, desarrollándose una rutina que permite de forma sencilla seleccionar el total de núcleos a utilizar dentro de un computador o clúster, tomando en cuenta el balanceo de carga de trabajo que cada uno de estos experimentará durante cada iteración.

El esquema de programación en paralelo desarrollado permitirá una implementación directa en problemas similares, generando así un ahorro considerable de tiempo a la hora de realizar una serie de experimentos, con el consecuente uso eficiente de los recursos.

La implementación serial del algoritmo Progressive Hedging presenta un mejor rendimiento en relación a la implementación serial extendida al trabajar con instancias superiores a los 20 escenarios. En relación a esto, la implementación en paralelo del algoritmo PH supera de forma considerable el rendimiento de la serial en la medida que se aumenta el número de núcleos utilizados, alcanzándose ahorros de tiempo cercanos a 90% c/r al caso original.

La técnica de bundling de escenarios permite una mejora significativa en cuanto a los tiempos de resolución, sin embargo los tiempos de inicialización involucrados impiden que ésta se presente como un enfoque eficiente en la práctica para instancias masivas.

Las técnicas de reducción de escenarios implementadas permiten reducir la dimensionalidad del problema original desde instancias masivas hasta tamaños manejables y accesibles para los computadores actuales, manteniendo la estructura estocástica del problema sin perder la robustez y representatividad de la instancia original.



El algoritmo de fast-forward para la reducción de escenarios se presenta como la mejor alternativa para llevar a cabo esta técnica, debido a que mantiene de gran forma la representatividad de la instancia original (manteniendo eventos de cola tipo black swans), lo que conlleva a una solución más robusta y de gran calidad en comparación a otros métodos, así como presentar una gran eficiencia en términos de tiempos de cómputo involucrados.

La metodología de comparación de calidad de la solución encontrada bajo los enfoques estocástico y determinístico desarrollada genera un marco de trabajo para próximos problemas que utilicen un enfoque de resolución similar, permitiendo así realizar comparaciones de rendimiento entre modelos determinísticos y estocásticos dentro del área de optimización.

Se logra dar respuesta para el modelo en estudio a uno de los desafíos más importantes dentro de los problemas de optimización estocástica, consistente en la determinación objetiva y sustentada matemáticamente del número suficiente de escenarios con los que se debiera trabajar, de tal forma de contar con una representación significativa de la realidad, pero que al mismo tiempo permita alcanzar una solución en tiempos de cómputo razonables.

Se observa que la solución estocástica supera a la formulación determinística en términos de calidad de solución en la medida que se agregan escenarios (mayor cantidad de información), alcanzando un punto estacionario, donde el agregar mayor cantidad de escenarios no permite mejorar el rendimiento del modelo en términos del trade-off entre calidad de la solución y tiempo necesario para alcanzarla. Así, se determina el punto de corte donde se indica el número suficiente de escenarios a utilizar para la formulación estocástica.

Se determina que el número suficiente de escenarios a incluir dentro del modelo se presenta entre los 400 y 500 escenarios. Dicho corte se relaciona de forma directa con el comportamiento de la solución del modelo determinístico, que alcanza un estado estacionario una vez superados los 400 escenarios.

Finalmente, es posible señalar que se logra con el objetivo de maximizar el valor esperado del plan de explotación de una instancia real de predios forestales, mejorando de forma significativa los tiempos de resolución en relación a las formulaciones clásicas, así como de incluir una mayor cantidad de escenarios en comparación a trabajos similares anteriores, incorporando además las decisiones del modelo original respecto a la mejora de caminos, posibilidad de stock de productos y el total de bosques de la instancia real.

Las principales contribuciones de este trabajo consisten en:

- La implementación de una metodología de creación y generación de árboles de escenarios para problemas estocásticos de forma satisfactoria y eficiente, lo que permite contar con una gran cantidad de instancias representativas de la realidad de forma automática y carente de errores, en formatos estandarizados y de directa utilización bajo los software de optimización más utilizados en la actualidad.

- La adaptación, configuración e implementación del algoritmo PH de forma óptima dentro de un modelo de planificación forestal, utilizando tanto su versión serial como paralela lo que permite alcanzar rendimientos ampliamente superiores en relación a los métodos clásicos de optimización.
- Se han determinado las técnicas más eficaces a la hora de realizar una reducción de escenarios desde el tamaño original de una instancia, volviendo así abordables instancias que en la actualidad presentan una serie de dificultades al momento de ser trabajadas.
- El desarrollo e implementación de una metodología rigurosa de simulación que permite llevar a cabo la comparación de los modelos determinístico y estocástico en términos de calidad de solución, siendo posible así determinar en términos cuantitativos la ganancia obtenida al llevar a cabo la optimización de un modelo complejo y real versus una simplificación clásica.
- Se determina el número suficiente de escenarios necesarios para representar de forma eficiente “la realidad” dentro del modelo forestal, dándose así respuesta al punto de corte que equilibra el trade-off entre calidad de la solución (según el aporte de información al modelo) y tiempos de resolución involucrados.

## 8.1 Trabajo Futuro

La inclusión de nuevas fuentes de incertidumbre permitirá generar un modelo cada vez más robusto y realista, obteniéndose así un plan de decisiones que toma en cuenta una mayor cantidad de elementos que se presentan en la práctica y que son importantes para la realización de las actividades. Así, la inclusión de los efectos de fenómenos y elementos climáticos sobre el crecimiento y rendimiento de cada una de las unidades y tipo de árboles, la posibilidad de incluir la presencia de potenciales incendios forestales con la correspondiente toma de decisiones respecto a cortafuegos, entre otras variables aleatorias, es un desafío que se queda abierto para futuros trabajos.

En relación al enfoque de generación de escenarios, es preciso realizar una adaptación pertinente de la metodología utilizada en la medida que se incluyan nuevas fuentes de incertidumbre, de tal forma de tomar en cuenta los efectos que se producen entre éstas, tanto a nivel grupal como en grupos específicos, de tal forma de mantener una metodología rigurosa, eficaz y eficiente a la hora de generar e incluir escenarios en formulaciones estocásticas.

La implementación en paralelo del algoritmo Progressive Hedging presenta una serie de oportunidades a la hora de obtener rendimientos superiores en términos de tiempos de resolución en modelos estocásticos masivos. Así, vital es seguir investigando en ésta línea con tal de mejorar el rendimiento a la hora de establecer la comunicación de los problemas y/o

soluciones entre los diversos computadores o núcleos involucrados, así como los tiempos de inicialización inherentes al algoritmo PH. En la misma línea, importante será en el futuro una implementación eficiente del algoritmo en paralelo bajo comunicación asincrónica, donde se vuelve innecesario el alcanzar una solución en todos los escenarios por cada iteración antes de continuar a la siguiente, pudiéndose obtener resultados de gran calidad en menores tiempos de cómputo.

La metodología de comparación bajo un enfoque de simulación presenta una serie de oportunidades a la hora de utilizarla en problemas de características similares y observar el comportamiento en cada uno de estos. Así, se presenta como desafío el seguir experimentando con nuevos problemas, comprobando su efectividad y utilidad práctica en otras áreas, así como la capacidad real que presenta a la hora de determinar el número suficiente de escenarios a incluir.

## Capítulo 9

### Glosario

- GAP: Distancia en términos porcentuales existente entre la mejor solución factible calculada y la mejor cota del óptimo que ha sido encontrada por el solver.
- MIP: Mixed Integer Programming. Problema de optimización lineal que hace referencia a un modelo que incluye variables lineales y enteras (incluida la presencia de variables binarias).
- PH: Progressive Hedging, hace referencia al algoritmo de descomposición por escenarios utilizado. En particular, la traducción correcta al español (consultando a sus autores) corresponde a “protegerse del riesgo”.
- Solver: Corresponde a una rutina de software que se encarga de resolver un problema en particular. Dentro del contexto de optimización, hace referencia a un software que utiliza algoritmos para resolver los problemas de optimización lineal, cuadráticos, entre otros.

## Capítulo 10

# Referencias

1. Andalaft N., Andalaft P., Guignard M., Adrian Magendzo, Alexis Wainer, Andres Weintraub. "A Problem of Forest Harvesting and Road Building Solved Through Model Strengthening and Lagrangean Relaxation", **Operations Research**, Vol.51, N°4, pp. 613-628, 2003
2. Crainic, Teodor Gabriel., Xiaorui, Fu., Progressive Hedging-Based Meta-Heuristics for Stochastic Network Design, **CIRRELT** 2009-03.
3. Holger Heitsch, Werner Römisch "Scenario Reduction Algorithms in Stochastic Programming", **Computational Optimizarion and Applications**, 24, PP187-206, 2003
4. Hvattum, Lars Magnus., Løkketangen, Arne., Using scenario trees and progressive hedging for stochastic inventory routing problems, **J. Heuristics DOI** 10.1007/s10732-008-9076-0.
5. Kirby, M. W., W. Hager, P. Wong. "Simultaneous planning of wildland transportation alternatives". **TIMS Studies in the Management Sciences** 21, pp. 371-387, 1986
6. Mulvey, John M., Vladimirov, Hercules. "Applying the Progressive Hedging Algorithm to stochastic generalized networks, **Ann. Oper. Res.** 31(1991) 399-424.
7. Rockafellar, R.T., Wets R. J-B. "Scenario and policy aggregation in optimization under uncertainty", **Mathematics of Operations Research**, 16, pp. 119-147, 1991.
8. Sandia National Labs, software Package Coop  
<https://software.sandia.gov/trac/coopr>
9. Scalable Heuristics for Stochastic Programming with Scenario Selection, Technical Report, June 2008 Discrete Math and Complex Systems Department, Sandia National Laboratories Albuquerque, NM 87185-1318.
10. Watson, J.P. and Woodruff, D.L. "Progressive Hedging Innovations for a class of stochastic mixed-integer resource allocation problems", **Computer Management Science**, 2010.
11. Watson, J.P., Hart, W., Woodruff, D.L. "PySP: Modeling and Solving Stochastic Programs in Python", **To appear in Mathematical Programming Computation**, 2011.
12. Weintraub, A., D. Navon. "A forest management planning model integrating silvicultural and transportation activities", **Management Science** 22(12) 1299-1309, 1976.

13. Weintraub A. and J. Vera "A Cutting Plane Approach for Chance Constrained Linear Programs", **Operations Research**. Vol. 39, N°5, pp. 776-785, 1991.
14. Weintraub, A., G. J. Jones, A. Magendzo, M. L. Meacham, M. W. Kirby. A heuristic system to solve mixed integer forest planning models. **Operations Research** 42 1010-1024, 1994
15. Weintraub, A., G. J. Jones, A. Magendzo, D. Malchuk. Heuristic procedures for solving mixed-integer harvest scheduling-Transportation planning models. **Canadian Journal Forest Research** 25, pp. 1618-1626, 1995.
16. Weintraub, A., Integer programming in forestry, **Ann. Oper. Res.** (2007) 149:209–216.
17. Weintraub, A., Wets R. J-B., "Harvesting management: Generating wood-prices scenarios", **To appear**
18. Willem K. Klein Haneveld and Maarten H. van der Vlerk, Stochastic integer programming: General models and algorithms, **Ann. Oper. Res.** 85(1999)39–57
19. Yueyue Fan, Changzheng Liu., Solving Stochastic Transportation Network Protection Problems Using the Progressive Hedging-based Method, **Netw Spat Econ** DOI.10.1007/s11067-008-9062-y
20. Forestry management under uncertainty, **Ann. Oper. Res.**, ISSN 0254-5330 May 21, 2009.
21. Badilla, Fernando., Problema de Planificación Forestal Estocástico Resuelto a través del algoritmo Progressive Hedging, Tesis Universidad de Chile, 2010
22. Cosecha Estocástica Óptima de Rodales De Pino Radiata, Eduardo Navarrete, **WORKSHOP INTERNACIONAL EIG2009**. Universidad de La Frontera – Chile.
23. Python Pickle Package Documentation  
<https://docs.python.org/2.7/library/pickle.html>

# Capítulo 11

## Anexos

### 11.1 Restricción de no anticipatividad

Existen distintas formulaciones para expresar las restricciones de no anticipatividad dentro del modelo de optimización:

#### 11.1.1 Simple

Consiste en una secuencia de igualdades que a través de la propiedad de transitividad se cumple:

$$x_t^s = x_t^{s'} \quad \forall s, s' \in N_t : ord(s) = ord(s') + 1 \quad \forall N_t \in N \quad \forall t \in T$$

Bajo esta formulación, se obtiene un número reducido de restricciones lineales:  $\sum_{t \in T} \sum_{N_t \in N} (|N_t| - 1)$ .

#### 11.1.2 Tradicional

Se trabaja con parejas de variables, igualándolas en la medida que pertenezcan al mismo conjunto de información:

$$x_t^s = x_t^{s'} \quad \forall s, s' \in N_t \quad \forall N_t \in N \quad \forall t \in T$$

#### 11.1.3 Promedio

Formulación basada en la propiedad que la suma de las variables debe ser idéntica al promedio de éstas:

$$\sum_{s \in N_t \setminus \{s'\}} x_t^s = (|N_t| - 1)x_t^{s'} \quad \forall s' \in N_t \quad \forall N_t \in N \quad \forall t \in T$$

A partir de ésta, se obtiene un número de restricciones lineales:  $\sum_{t \in T} \sum_{N_t \in N} (|N_t| - 1)$ .

### 11.1.4 Promedio General

Generalización de la formulación precedente, tomando un distinto número de variables a cada lado de la igualdad, para  $1 \leq k \leq \lfloor \frac{|N_t|}{2} \rfloor$ :

$$k \left( \sum_{s \in N_t \setminus \{s'\}} x_t^s \right) = (|N_t| - k)x_t^{s'} \quad \forall s' \in N_t \quad \forall N_t \in N \quad \forall t \in T$$

Bajo ésta, el número de restricciones asciende a:

$$\sum_{t \in T} \sum_{N_t \in N} \lfloor \frac{|N_t|}{2} \rfloor (|N_t| - 1).$$

### 11.1.5 Binaria

Formulación válida solamente para variables binarias, permite una reducción significativa del número de restricciones en la medida que se incluya una variable binaria por cada conjunto de información:

$$\sum_{s \in N_t} x_t^s = |N_t|y_t \quad \forall N_t \in N \quad \forall t \in T \quad (n \leftrightarrow N_t)$$

Así, se logra reducir el total de restricciones a  $\sum_{t \in T} \sum_{N_t \in N} 1$ .

## 11.2 Modelo generador de Escenarios

La metodología utilizada para llevar a cabo al generación de árboles de precios comienza con el modelamiento del proceso de precios de los diversos productos forestales como una ecuación estocástica diferencial (movimiento estocástico) del tipo:

$$\begin{aligned} dp(t) &= \mu(v - p(t))dt + \sigma dw(t)p(t) \\ p(0) &= p_0 \\ t &\geq 0 \end{aligned}$$

dónde  $p(t)$  corresponde al precio que se observa en el periodo  $t$ ,  $p_0$  es el precio en la actualidad,  $v$  es el precio al que se convergerá en el futuro y siendo  $\mu$  y  $\sigma$  constantes del modelo que necesitan ser estimadas, donde  $\mu$  representa la tasa a la que el proceso convergerá al valor "normal" y  $\sigma$  representa la volatilidad del mismo.

Este tipo de modelos se conoce como un proceso browniano geométrico, ajustado con una reversion hacia la media. Del mismo modo, la volatilidad del proceso es modelada a través de un término de difusión que involucra  $w$ , representando un proceso de Weiner tradicional.



La solución del movimiento estocástico planteado consiste en:

$$p(t) = p_0 \exp \left[ - \left( \mu + \frac{1}{2} \sigma^2 \right) t + \sigma w(t) \right] + \mu \nu \int_0^t e^{r(t,s)} ds$$

$$r(t, s) = - \left[ \mu + \frac{1}{2} \sigma^2 \right] (t - s) + \sigma (w(t) - w(s))$$

Sin embargo esta solución es bastante compleja de obtener de forma exacta, por ende se trabaja con una aproximación que en la práctica es imperceptible, siendo así finalmente:

$$p(t) = \nu(1 - e^{-\mu t}) + p_0 \exp \left[ \left( -\mu - \frac{1}{2} \sigma^2 \right) t + \sigma w(t) \right]$$

Analizando esta expresión, es posible notar que debido a que los incrementos de un proceso de Wiener clásico son variables aleatorias independientes de distribución gaussiana, el exponente de la exponencial también lo es y por ende el proceso de precios sigue una distribución log-gaussiana desplazada. Así, el proceso de precios es un proceso log-gaussiano.

Calculando la esperanza y varianza del proceso de precios, bajo las propiedades de un proceso de Wiener se obtiene:

$$E(p(t)) = \nu(1 - e^{-\mu t}) + p_0 e^{-\mu t}$$

$$Var(p(t)) = (p_0 e^{-\mu t})^2 (e^{\sigma^2 t} - 1)$$

Observando la expresión de la esperanza, se observa que existe un término constante de “desplazamiento”  $\nu(1 - e^{-\mu t})$ . Esto permite calcular la función de densidad del proceso de precios a partir de las propiedades clásicas de la distribución log-gaussiana. Así, utilizando la variable auxiliar  $Z_t$ , es posible calcular la función de densidad de  $p(t)$ :

Sea  $Z_t = p_0 \exp[(-\mu - \frac{1}{2}\sigma^2)t + \sigma w(t)]$ , luego bajo las propiedades de una función log-gaussiana, su densidad es:

$$d_{Z_t}(s) = (s\tau\sqrt{2\pi})^{-1} e^{-(\ln s - \theta)^2 / 2\tau^2}, s \in (0, \infty)$$

donde la esperanza y varianza son:

$$E(Z_t) = e^{\theta + \frac{\tau^2}{2}} = p_0 e^{-\mu t + \frac{\sigma^2 t}{2}}$$

$$Var(Z_t) = (E(Z_t))^2 (e^{\tau^2} - 1) = p_0^2 e^{-2\mu t} (e^{\sigma^2 t} - 1)$$

A partir de estas expresiones, es posible calcular los coeficientes de la función de densidad:

$$\theta = \ln p_0 - \mu t$$

$$\tau = \sigma \sqrt{t}$$

lo que permite obtener finalmente una descripción completa de la distribución del proceso de precios para todo t:

$$p(t) = Z_t + \nu(1 - e^{-\mu t})$$

Gracias a lo anterior, es posible tener una descripción de las funciones de densidad y distribución acumulada del proceso, lo que permite establecer un ruptura del espacio probabilístico para el próximo periodo en estudio en el número de nodos hijos que se requieran, calculándose luego las esperanzas condicionales pertenecientes a cada tramo, valor que finalmente representará el valor de la variable aleatoria en estudio en cada nodo hijo del siguiente periodo.

### 11.2.1 Ejemplos de árboles de escenarios

A continuación se presentan una serie de ejemplos de los árboles de escenarios obtenidos a partir del script implementado bajo el software Matlab, haciendo énfasis en el formato intuitivo y natural en el que se presentan los outputs:

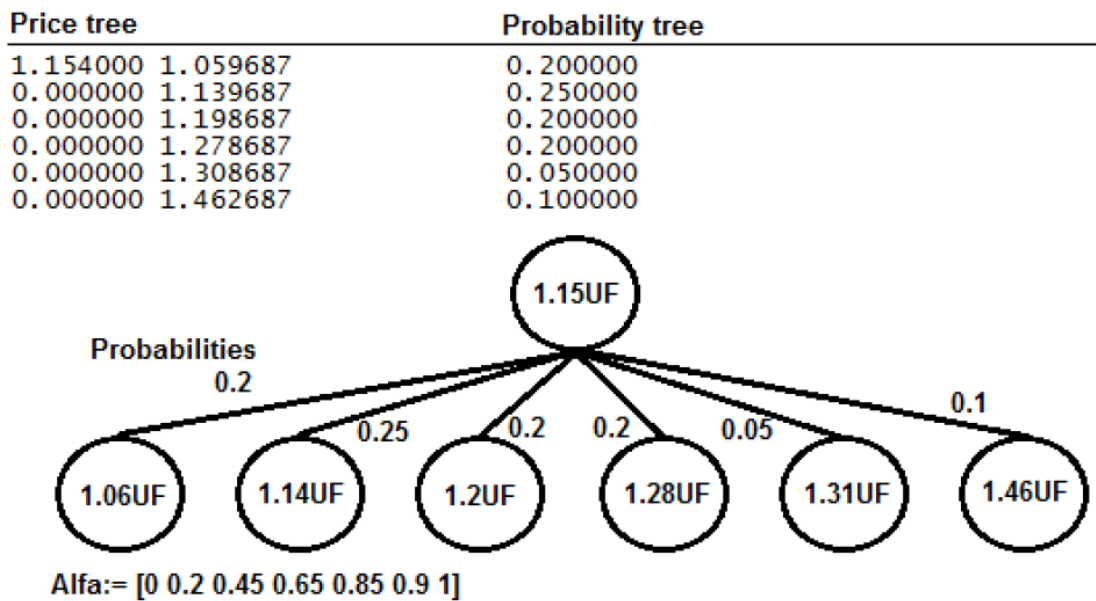


Figura 11.1: Árbol de seis escenarios para un periodo

Price Tree			Probability Tree	
16.718847	15.069926	11.952726	0.250000	0.100000
0.000000	15.679926	18.732726	0.100000	0.800000
0.000000	16.519926	22.922726	0.150000	0.100000
0.000000	17.409926	0.000000	0.150000	0.000000
0.000000	18.089926	15.162726	0.100000	0.450000
0.000000	22.789926	15.842726	0.250000	0.100000
0.000000	0.000000	23.762726	0.000000	0.450000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	16.212726	0.000000	1.000000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	16.962726	0.000000	1.000000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	17.542726	0.000000	1.000000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	21.442726	0.000000	1.000000

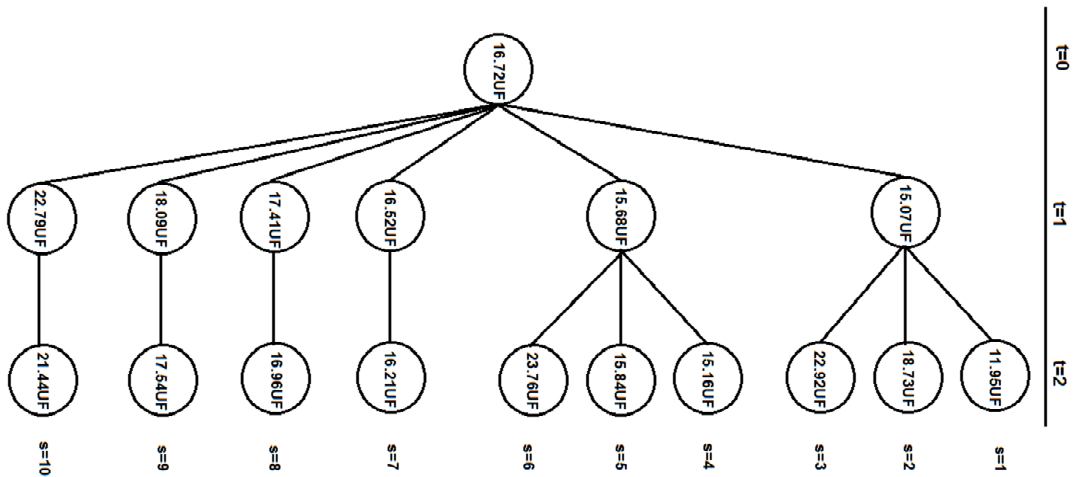


Figura 11.2: Árbol de 10 escenarios para 3 periodos

Price tree			Probability tree	
1.154000	1.059687	0.983936	0.200000	0.250000
0.000000	1.139687	1.073936	0.250000	0.250000
0.000000	1.198687	1.171936	0.200000	0.250000
0.000000	1.278687	1.467936	0.200000	0.250000
0.000000	1.308687	0.000000	0.050000	0.000000
0.000000	1.462687	1.086936	0.100000	0.350000
0.000000	0.000000	1.181936	0.000000	0.250000
0.000000	0.000000	1.308936	0.000000	0.250000
0.000000	0.000000	1.566936	0.000000	0.150000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	1.092936	0.000000	0.250000
0.000000	0.000000	1.194936	0.000000	0.250000
0.000000	0.000000	1.305936	0.000000	0.250000
0.000000	0.000000	1.639936	0.000000	0.250000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	1.062936	0.000000	0.100000
0.000000	0.000000	1.221936	0.000000	0.300000
0.000000	0.000000	1.306936	0.000000	0.200000
0.000000	0.000000	1.495936	0.000000	0.300000
0.000000	0.000000	1.739936	0.000000	0.100000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	1.224936	0.000000	0.350000
0.000000	0.000000	1.333936	0.000000	0.250000
0.000000	0.000000	1.479936	0.000000	0.250000
0.000000	0.000000	1.776936	0.000000	0.150000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	1.350936	0.000000	0.350000
0.000000	0.000000	1.635936	0.000000	0.500000
0.000000	0.000000	1.966936	0.000000	0.150000



Figura 11.3: Árbol de 24 escenarios para 3 periodos

### 11.3 Reducción de escenarios: Algoritmo Fast Forward

Sea  $c(w, \tilde{w})$  una función de distancia tal que  $c: \Omega \times \Omega \rightarrow \mathbb{R}$ , donde  $\Omega$  corresponde a un subconjunto cerrado sobre  $\mathbb{R}^s$ ,  $X \subset \mathbb{R}^m$  corresponde al vector de decisiones del problema estocástico en estudio y la función de distancia cumple con:

$$c(w, \tilde{w}) := \max\{1, h(\|w - w_0\|), h(\|\tilde{w} - w_0\|)\} \|w - \tilde{w}\|, \quad \forall w, \tilde{w} \in \Omega$$

donde la función  $h(\|\cdot - w_0\|)$  representa el crecimiento de constantes que cumplen con ser Lipschitz de la función objetivo  $f_0(\cdot, x)$  en bolas alrededor de  $w_0$  respecto a alguna norma  $\|\cdot\|$  en  $\mathbb{R}^s$ , donde  $w$  representa el vector de valores de las variables aleatorias alcanzado en un escenario determinado.

Sea  $N$  el número total de escenarios con los que cuenta el árbol de la instancia en estudio, sea  $J \subset \{1, \dots, N\}$  el conjunto de escenarios que finalmente serán mantenidos como representantes,  $p_k$  la probabilidad de ocurrencia del escenario  $k$ -ésimo y sea  $n$  el total de escenarios que se busca obtener luego de aplicar el algoritmo de reducción, con  $n < N$ . Así, el algoritmo fast-forward es el siguiente:

---

#### Algoritmo Fast-Forward Extendido

---

1.  $c_{ku}^{[1]} := c(w_k, w_u) \quad k, u = 1, \dots, N.$   
 $z_u^{[1]} := \sum_{\substack{k=1 \\ k \neq u}}^N p_k c_{ku}^{[1]} \quad u = 1, \dots, N.$   
 $u_1 \in \arg \min_{u \in \{1, \dots, N\}} z_u^{[1]}.$   
 $J^{[1]} := \{1, \dots, N\} \setminus \{u_1\}.$
  
- i.  $c_{ku}^{[i]} := \min\{c_{ku}^{[i-1]}, c_{ku_{i-1}}^{[i-1]}\} \quad k, u \in J^{[i-1]}.$   
 $z_u^{[i]} := \sum_{k \in J^{[i-1]} \setminus \{u\}} p_k c_{ku}^{[i]} \quad u \in J^{[i-1]}.$   
 $u_i \in \arg \min_{u \in J^{[i-1]}} z_u^{[i]}.$   
 $J^{[i]} := J^{[i-1]} \setminus \{u_i\}.$

n+1. Redistribución utilizando el teorema de redistribución óptima

---

Algoritmo que surge como solución directa de un problema de optimización donde se busca disminuir la diferencia entre la distribución probabilística del árbol de escenarios original denominada  $P$  y la que se alcanzará en el reducido, denominada  $Q$ .

### 11.3.1 Teorema de redistribución óptima

Dado un conjunto  $J \subset \{1, \dots, N\}$  se tiene el problema de optimización:

$$D_j = \min \left\{ D(J; q) : q_j \geq 0, \sum_{j \notin J} q_j = 1 \right\} = \sum_{i \in J} p_i \min_{j \notin J} c(w_i, w_j)$$

Luego el óptimo se alcanza en:

$$\bar{q}_j = p_j + \sum_{i \in J_j} p_i \text{ para todo } j \notin J$$

$$J_j := \{i \in J : j = j(i)\} \text{ y } j(i) \in \operatorname{argmin}_{j \notin J} c(w_i, w_j) \forall i \in J$$

Gracias a este teorema, es posible determinar la forma en la que se deben distribuir las probabilidades de ocurrencia asociadas a cada uno de los escenarios que no fueron seleccionados bajo el algoritmo de reducción de escenarios fast-forward, manteniendo de la mejor manera posible la distribución probabilística original del árbol de escenarios en estudio.

Se observa que se le agrega la probabilidad de ocurrencia a aquellos escenarios que se encuentren más cercanos en términos de la métrica de distancia utilizada para llevar a cabo los diversos cálculos de la metodología.

### 11.4 Resultados de Simulación

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.715.102	20,1342	1.715.102	1.715.103	8,110	1.715.103	-1
1.747.612	18,5361	1.747.612	1.746.212	9,340	1.749.212	-1.600
1.715.102	20,1342	1.715.102	1.715.103	8,110	1.715.103	-1
1.743.106	20,1257	1.743.106	1.743.107	11,160	1.743.107	-1
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	19,0400	1.747.612	1.745.270	7,880	1.745.270	2.342
32.887.344	19,4185	32.887.344	32.928.001	8,400	32.887.001	343
32.614.894	21,5922	32.614.894	32.677.210	9,700	32.677.210	-62.316
1.715.102	20,1342	2.034.500	1.715.103	7,840	2.021.161	13.340
1.747.612	17,7760	1.747.612	1.745.270	8,320	1.745.270	2.342
1.619.258	8,7768	1.619.258	1.619.259	8,670	1.619.258	0
1.666.314	11,7700	1.666.314	1.666.314	9,020	1.666.313	0
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	18,0875	1.747.612	1.745.270	8,460	1.745.270	2.342
2.142.669	20,5744	2.142.669	2.136.221	8,840	2.136.221	6.448
2.034.500	21,8938	2.034.500	2.021.161	21,920	2.021.161	13.340

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.715.102	20,1342	2.034.500	1.715.103	7,720	2.021.161	13.340
1.747.612	18,6175	1.747.612	1.745.270	7,930	1.745.270	2.342
1.715.102	19,8987	1.715.102	1.715.103	8,300	1.715.102	0
1.743.106	20,7454	1.743.106	1.743.107	11,200	1.743.106	0
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	18,0875	1.747.612	1.745.270	8,460	1.745.270	2.342
2.142.669	17,1932	2.142.669	2.136.221	8,580	2.136.221	6.448
2.034.500	23,4519	2.034.500	2.021.161	7,700	2.021.161	13.340
1.715.102	20,1342	2.034.500	1.715.103	7,840	2.021.161	13.340
1.747.612	18,6573	1.747.612	1.745.270	7,970	1.732.270	15.342
32.887.344	20,0360	32.887.344	32.928.001	8,610	32.887.001	343
32.614.894	22,0692	32.614.894	32.677.210	10,070	32.613.210	1.684
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	18,0875	1.747.612	1.745.270	8,460	1.745.270	2.342
2.142.669	20,5744	2.142.669	2.136.221	8,840	2.136.221	6.448
2.034.500	21,8938	2.034.500	2.021.161	21,920	2.021.161	13.340
1.715.102	20,1342	2.034.500	1.715.103	7,840	2.021.161	13.340
1.747.612	19,7748	1.747.612	1.745.270	8,510	1.745.270	2.342
1.825.343	15,8531	1.825.343	1.825.344	7,260	1.825.344	-1
1.904.592	16,6150	1.904.592	1.838.759	7,560	1.838.759	65.833
1.715.102	20,1342	1.904.592	1.715.103	7,560	1.838.759	65.833
1.747.612	16,9936	1.747.612	1.745.270	7,920	1.745.270	2.342
1.715.102	18,1735	1.715.102	1.715.103	8,150	1.715.102	0
1.743.106	18,5658	1.743.106	1.743.107	11,100	1.743.106	0
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	18,0875	1.747.612	1.745.270	8,460	1.745.270	2.342
2.142.669	20,5744	2.142.669	2.136.221	8,840	2.136.221	6.448
2.034.500	21,8938	2.034.500	2.021.161	21,920	2.021.161	13.340
1.715.102	20,1342	2.034.500	1.715.103	7,840	2.021.161	13.340
1.747.612	18,6573	1.747.612	1.745.270	7,970	1.732.270	15.342
2.142.669	15,9835	2.142.669	2.136.221	9,020	2.136.221	6.448
2.034.500	20,7055	2.034.500	2.021.161	7,630	2.021.161	13.340
1.715.102	20,1342	1.904.592	1.715.103	7,560	1.838.759	65.833
1.747.612	16,9936	1.747.612	1.745.270	7,920	1.745.270	2.342
2.142.669	20,5744	2.142.669	2.136.221	8,840	2.136.221	6.448
2.034.500	21,8938	2.034.500	2.021.161	21,920	2.021.161	13.340
1.715.102	20,1342	2.034.500	1.715.103	7,720	2.021.161	13.340
1.747.612	18,6175	1.747.612	1.745.270	7,930	1.745.270	2.342
1.619.258	8,9750	1.619.258	1.619.259	8,340	1.619.258	0
1.666.314	11,3417	1.666.314	1.666.314	8,410	1.666.313	0

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.715.102	20,1342	1.666.314	1.715.103	8,360	1.666.313	0
1.747.612	17,0202	1.747.612	1.745.270	7,920	1.745.270	2.342
1.655.858	9,2945	1.655.858	1.655.859	8,440	1.655.858	0
1.695.282	19,9314	1.695.282	1.695.283	7,950	1.695.282	0
1.715.102	20,1342	1.695.282	1.715.103	7,670	1.695.282	0
1.747.612	21,2976	1.747.612	1.745.270	8,180	1.745.270	2.342
2.142.669	19,3936	2.142.669	2.136.221	8,680	2.136.221	6.448
2.034.500	27,0446	2.034.500	2.021.161	7,820	2.021.161	13.340
1.715.102	20,1342	2.034.500	1.715.103	7,720	2.021.161	13.340
1.747.612	21,0850	1.747.612	1.733.219	8,210	1.733.219	14.393
32.887.344	22,7828	32.887.344	32.928.001	8,730	32.887.001	343
32.614.894	24,5984	32.614.894	32.677.210	10,100	32.614.210	684
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	19,0400	1.747.612	1.745.270	7,880	1.745.270	2.342
32.887.344	19,4185	32.887.344	32.928.001	8,400	32.887.001	343
32.614.894	21,5922	32.614.894	32.677.210	9,700	32.677.210	-62.316
1.715.102	20,1342	1.904.592	1.715.103	7,560	1.838.759	65.833
1.747.612	16,9936	1.747.612	1.745.270	7,920	1.745.270	2.342
2.142.669	20,5744	2.142.669	2.136.221	8,840	2.136.221	6.448
2.034.500	21,8938	2.034.500	2.021.161	21,920	2.021.161	13.340
1.715.102	20,1342	1.743.106	1.715.103	11,070	1.743.106	0
1.747.612	19,0400	1.747.612	1.745.270	7,880	1.745.270	2.342
1.655.858	9,2945	1.655.858	1.655.859	8,440	1.655.858	0
1.695.282	25,1700	1.695.282	1.695.283	7,630	1.695.282	0

Figura 11.4: Resultados de simulación para instancia de 10 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.842.901	19,6400	1.842.899	1.846.901	80,240	1.846.901	-4.002
1.873.958	20,0700	1.873.956	1.826.894	75,000	1.826.894	47.062
1.776.892	23,2300	1.776.891	1.775.136	76,000	1.775.136	1.755
1.866.919	18,4800	1.866.918	1.790.039	89,000	1.790.039	76.879
1.842.901	19,6400	1.842.899	1.846.901	87,000	1.790.039	52.859
1.745.059	18,8012	1.745.057	1.759.404	77,000	1.759.404	-14.346
31.864.819	20,5693	31.864.817	32.104.008	86,000	32.104.008	-239.190
31.804.859	21,2880	31.804.858	32.030.012	77,000	32.030.012	-225.154
1.842.901	19,6400	1.842.900	1.846.901	85,000	1.718.229	124.671
1.745.059	17,9382	1.745.057	1.759.404	88,000	1.759.404	-14.346
1.690.469	18,6800	1.690.468	1.687.681	90,000	1.687.681	2.786
1.725.651	14,9130	1.725.649	1.718.229	90,000	1.718.229	7.420



N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.842.901	19,6400	1.842.900	1.846.901	88,000	1.790.039	52.860
1.695.059	18,8012	1.695.058	1.700.544	75,000	1.700.544	-5.485
1.776.892	22,4000	1.776.890	1.775.136	88,000	1.775.136	1.754
1.866.919	14,4800	1.866.917	1.790.039	90,000	1.790.039	76.878
1.842.901	19,6400	1.842.899	1.846.901	90,000	1.834.901	7.998
1.873.958	20,0700	1.873.957	1.826.894	83,000	1.826.894	47.063
1.776.892	23,2300	1.776.891	1.775.136	83,000	1.775.136	1.755
1.866.919	18,4800	1.866.917	1.790.039	75,000	1.790.039	76.878
1.842.901	19,6400	1.842.899	1.846.901	89,000	1.846.901	-4.002
1.756.577	15,0000	1.756.576	1.756.525	85,000	1.756.434	142
1.789.415	23,0000	1.789.414	1.797.754	74,000	1.797.683	-8.269
1.879.050	18,0000	1.879.049	1.886.528	76,000	1.886.525	-7.476
1.842.901	19,6400	1.842.899	1.846.901	73,000	1.846.901	-4.002
1.777.525	17,0000	1.777.523	1.785.817	78,000	1.785.819	-8.296
1.780.086	22,0000	1.780.084	1.785.692	73,000	1.785.650	-5.566
1.867.713	22,0000	1.867.711	1.873.823	75,000	1.873.792	-6.081
1.842.901	19,6400	1.842.899	1.846.901	71,000	1.846.901	-4.002
1.884.398	21,0000	1.884.397	1.886.943	71,000	1.886.980	-2.583
1.796.121	19,0000	1.796.120	1.797.762	78,000	1.797.834	-1.714
1.809.016	16,0000	1.809.014	1.808.292	80,000	1.808.239	775
1.842.901	19,6400	1.842.900	1.846.901	71,000	1.846.901	-4.001
1.777.861	15,0000	1.777.859	1.787.433	88,000	1.787.352	-9.493
1.749.600	17,0000	1.749.598	1.754.906	84,000	1.754.921	-5.323
1.736.763	15,0000	1.736.762	1.736.719	79,000	1.736.649	113
1.842.901	19,6400	1.842.899	1.846.901	71,000	1.846.901	-4.002
1.781.359	19,0000	1.781.358	1.783.376	74,000	1.783.385	-2.027
1.824.294	20,0000	1.824.292	1.832.993	85,000	1.833.088	-8.796
1.770.320	23,0000	1.770.319	1.772.985	88,000	1.772.942	-2.623
1.842.901	19,6400	1.842.900	1.846.901	86,000	1.846.901	-4.001
1.796.608	20,0000	1.796.607	1.803.473	89,000	1.803.474	-6.867
1.738.805	18,0000	1.738.804	1.748.705	88,000	1.748.617	-9.813
1.878.648	15,0000	1.878.646	1.886.116	82,000	1.886.139	-7.493
1.842.901	19,6400	1.842.899	1.846.901	87,000	1.846.901	-4.002
1.801.013	16,0000	1.801.011	1.802.569	82,000	1.802.644	-1.633
1.820.166	15,0000	1.820.164	1.829.157	88,000	1.829.235	-9.071
1.806.569	22,0000	1.806.567	1.809.521	83,000	1.809.600	-3.033
1.842.901	19,6400	1.842.899	1.846.901	74,000	1.846.901	-4.002
1.846.725	17,0000	1.846.723	1.846.869	89,000	1.846.949	-226
1.901.117	17,0000	1.901.115	1.903.551	87,000	1.903.584	-2.469
1.741.658	22,0000	1.741.657	1.745.893	74,000	1.745.934	-4.277

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.842.901	19,6400	1.842.899	1.846.901	85,000	1.846.901	-4.002
1.802.604	20,0000	1.802.603	1.803.739	76,000	1.803.665	-1.062
1.889.971	15,0000	1.889.970	1.893.668	86,000	1.893.752	-3.782
1.748.617	21,0000	1.748.615	1.750.148	76,000	1.750.092	-1.477
1.842.901	19,6400	1.842.899	1.846.901	80,000	1.846.901	-4.002
1.790.150	17,0000	1.790.148	1.790.233	74,000	1.790.256	-108
1.869.570	17,0000	1.869.568	1.876.222	77,000	1.876.178	-6.610
1.814.955	22,0000	1.814.954	1.821.250	77,000	1.821.244	-6.290
1.842.901	19,6400	1.842.899	1.846.901	77,000	1.846.901	-4.002
1.880.694	17,0000	1.880.693	1.884.933	87,000	1.884.953	-4.260
1.821.643	19,0000	1.821.642	1.828.674	74,000	1.828.703	-7.061
1.814.812	23,0000	1.814.811	1.823.996	78,000	1.823.985	-9.174
1.842.901	19,6400	1.842.900	1.846.901	83,000	1.846.901	-4.001
1.875.464	20,0000	1.875.462	1.874.551	88,000	1.874.490	972
1.818.960	16,0000	1.818.959	1.818.328	79,000	1.818.349	610
1.832.397	15,0000	1.832.396	1.835.669	89,000	1.835.624	-3.228
1.842.901	19,6400	1.842.900	1.846.901	78,000	1.846.901	-4.001
1.782.346	23,0000	1.782.345	1.786.795	73,000	1.786.862	-4.517
1.811.722	19,0000	1.811.720	1.819.373	72,000	1.819.391	-7.671
1.801.569	17,0000	1.801.568	1.804.264	74,000	1.804.269	-2.701
1.842.901	19,6400	1.842.899	1.846.901	83,000	1.846.901	-4.002
1.885.554	17,0000	1.885.552	1.887.893	74,000	1.887.807	-2.255
1.850.310	17,0000	1.850.308	1.859.429	84,000	1.859.438	-9.130
1.810.472	20,0000	1.810.470	1.811.279	73,000	1.811.181	-711
1.842.901	19,6400	1.842.900	1.846.901	79,000	1.846.901	-4.001
1.863.972	17,0000	1.863.971	1.870.205	71,000	1.870.157	-6.186
1.812.242	18,0000	1.812.241	1.821.470	90,000	1.821.460	-9.219
1.829.365	23,0000	1.829.363	1.838.921	88,000	1.838.832	-9.469

Figura 11.5: Resultados de simulación para instancia 20 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.890.595	18,5000	1.888.840	1.898.869	125,690	1.898.894	-10.054
1.796.104	21,0000	1.793.618	1.805.754	130,000	1.805.744	-12.126
1.864.930	22,0000	1.863.696	1.869.940	126,000	1.869.980	-6.284
1.758.500	22,0000	1.757.377	1.759.357	135,000	1.759.324	-1.947
1.890.595	23,5146	1.887.590	1.898.869	127,000	1.898.784	-11.194
1.820.090	15,0000	1.818.519	1.825.273	130,000	1.825.307	-6.788
1.864.715	23,0000	1.862.509	1.873.368	118,000	1.873.293	-10.784
1.808.292	19,0000	1.806.766	1.810.275	129,000	1.810.361	-3.595

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.890.595	15,1363	1.889.063	1.898.869	135,000	1.898.843	-9.780
1.897.541	16,0000	1.893.589	1.904.070	134,000	1.904.128	-10.539
1.760.716	19,0000	1.758.355	1.768.324	116,000	1.768.334	-9.979
1.848.349	21,0000	1.847.767	1.850.326	126,000	1.850.424	-2.657
1.890.595	17,2415	1.887.763	1.898.869	116,000	1.898.930	-11.167
1.862.952	20,0000	1.861.209	1.867.672	134,000	1.867.638	-6.429
1.877.108	21,0000	1.876.324	1.877.681	128,000	1.877.638	-1.314
1.718.467	18,0000	1.715.939	1.718.945	118,000	1.718.977	-3.038
1.890.595	19,3242	1.886.884	1.898.869	135,000	1.898.872	-11.988
1.827.681	18,0000	1.824.265	1.827.638	133,000	1.827.613	-3.348
1.861.929	15,0000	1.860.849	1.866.246	129,000	1.866.261	-5.412
1.839.599	23,0000	1.836.710	1.841.366	118,000	1.841.273	-4.563
1.890.595	18,1464	1.887.635	1.898.869	135,000	1.898.888	-11.253
1.848.069	16,0000	1.844.236	1.848.619	119,000	1.848.565	-4.329
1.723.361	19,0000	1.721.345	1.725.792	128,000	1.725.804	-4.459
1.875.578	15,0000	1.874.194	1.879.542	128,000	1.879.553	-5.359
1.890.595	22,2454	1.888.689	1.898.869	134,000	1.898.800	-10.111
1.815.707	16,0000	1.814.439	1.823.857	115,000	1.823.918	-9.479
1.905.925	22,0000	1.904.896	1.915.401	121,000	1.915.409	-10.513
1.872.742	17,0000	1.869.648	1.878.189	128,000	1.878.235	-8.587
1.890.595	23,2442	1.887.882	1.898.869	125,000	1.898.945	-11.063
1.732.897	15,0000	1.729.826	1.740.966	120,000	1.740.913	-11.087
1.874.544	23,0000	1.872.408	1.878.019	121,000	1.877.920	-5.512
1.859.199	23,0000	1.857.034	1.858.223	115,000	1.858.251	-1.217
1.890.595	21,2341	1.887.352	1.898.869	130,000	1.898.833	-11.481
1.841.303	16,0000	1.837.315	1.844.197	136,000	1.844.254	-6.939
1.851.247	21,0000	1.847.402	1.853.339	120,000	1.853.394	-5.992
1.887.141	18,0000	1.884.195	1.894.331	124,000	1.894.240	-10.045
1.890.595	20,7765	1.888.658	1.898.869	129,000	1.898.911	-10.253
1.873.274	18,0000	1.870.618	1.875.048	129,000	1.874.970	-4.352
1.870.440	23,0000	1.867.626	1.871.281	122,000	1.871.285	-3.659
1.879.357	18,0000	1.877.379	1.885.861	120,000	1.885.908	-8.529
1.890.595	19,2324	1.889.441	1.898.869	129,000	1.898.787	-9.346
1.774.574	21,0000	1.771.239	1.782.699	128,000	1.782.749	-11.510
1.873.903	19,0000	1.871.486	1.882.107	116,000	1.882.187	-10.701
1.866.416	22,0000	1.862.810	1.875.004	122,000	1.874.967	-12.157
1.890.595	24,1251	1.888.834	1.898.869	131,000	1.898.818	-9.984
1.758.104	21,0000	1.756.967	1.767.504	125,000	1.767.579	-10.612
1.858.258	17,0000	1.856.891	1.861.084	121,000	1.861.001	-4.110
1.895.921	17,0000	1.892.957	1.903.926	126,000	1.903.954	-10.997

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.890.595	15,3412	1.887.648	1.898.869	117,000	1.898.807	-11.159
1.843.492	18,0000	1.840.181	1.847.908	121,000	1.847.963	-7.782
1.865.326	17,0000	1.864.235	1.873.383	119,000	1.873.373	-9.138
1.869.946	21,0000	1.868.143	1.874.825	134,000	1.874.841	-6.698
1.890.595	16,3435	1.886.870	1.898.869	118,000	1.898.962	-12.092
1.806.275	22,0000	1.804.319	1.806.306	135,000	1.806.328	-2.009
1.905.513	18,0000	1.904.283	1.905.497	124,000	1.905.593	-1.310
1.792.478	20,0000	1.791.407	1.794.715	134,000	1.794.670	-3.263
1.890.595	18,3253	1.889.591	1.898.869	129,000	1.898.894	-9.303
1.881.547	15,0000	1.878.101	1.887.857	132,000	1.887.811	-9.710
1.720.728	16,0000	1.716.762	1.729.136	129,000	1.729.123	-12.361
1.766.197	15,0000	1.765.175	1.770.863	135,000	1.770.857	-5.682
1.890.595	21,4315	1.889.269	1.898.869	129,000	1.898.844	-9.575
1.811.852	15,0000	1.811.216	1.814.857	123,000	1.814.800	-3.584
1.803.205	16,0000	1.799.439	1.806.617	119,000	1.806.695	-7.256
1.854.723	16,0000	1.852.827	1.863.497	119,000	1.863.467	-10.640
1.890.595	23,1364	1.889.689	1.898.869	126,000	1.898.953	-9.264
1.868.355	20,0000	1.866.404	1.872.342	119,000	1.872.423	-6.019
1.860.882	15,0000	1.857.323	1.867.255	120,000	1.867.228	-9.905
1.892.081	17,0000	1.889.350	1.894.669	116,000	1.894.569	-5.219
1.890.595	23,2323	1.889.255	1.898.869	127,000	1.898.927	-9.672
1.809.711	17,0000	1.807.036	1.816.572	115,000	1.816.590	-9.554
1.797.597	21,0000	1.795.876	1.806.967	123,000	1.806.950	-11.074
1.878.202	21,0000	1.877.148	1.886.978	117,000	1.886.899	-9.751
1.890.595	19,5415	1.890.181	1.898.869	133,000	1.898.825	-8.644
1.860.682	23,0000	1.858.722	1.861.887	120,000	1.861.962	-3.240
1.727.867	19,0000	1.724.197	1.728.430	118,000	1.728.512	-4.315
1.726.930	23,0000	1.725.463	1.728.484	116,000	1.728.557	-3.094
1.890.595	15,2314	1.888.446	1.898.869	118,000	1.898.915	-10.469
1.876.862	21,0000	1.874.498	1.883.867	117,000	1.883.809	-9.311
1.859.738	15,0000	1.856.889	1.864.978	135,000	1.864.912	-8.023
1.843.097	19,0000	1.839.419	1.853.065	118,000	1.852.992	-13.573

Figura 11.6: Resultados de simulación para instancia 30 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.827.194	22,2843	1.827.192	1.826.273	183,000	1.826.273	919
1.715.858	18,0000	1.715.854	1.719.915	176,000	1.719.943	-4.089
1.782.929	21,0000	1.782.925	1.789.917	176,000	1.789.898	-6.973
1.858.176	18,0000	1.858.174	1.863.329	179,000	1.863.323	-5.149

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.827.194	21,4564	1.827.189	1.826.273	186,000	1.826.273	916
1.812.137	23,0000	1.812.133	1.818.972	177,000	1.819.064	-6.931
1.760.068	23,0000	1.760.065	1.762.043	187,000	1.762.024	-1.959
1.770.644	16,0000	1.770.642	1.775.577	183,000	1.775.638	-4.996
1.827.194	18,4567	1.827.191	1.826.273	176,000	1.826.273	918
1.775.425	22,0000	1.775.420	1.777.964	187,000	1.778.012	-2.592
1.860.060	22,0000	1.860.058	1.862.430	183,000	1.862.383	-2.325
1.741.965	22,0000	1.741.961	1.750.084	189,000	1.750.059	-8.098
1.827.194	19,5645	1.827.192	1.826.273	174,000	1.826.273	919
1.904.118	15,0000	1.904.114	1.913.198	176,000	1.913.221	-9.107
1.740.390	21,0000	1.740.385	1.749.387	177,000	1.749.323	-8.938
1.724.138	15,0000	1.724.136	1.728.987	175,000	1.728.951	-4.815
1.827.194	21,4564	1.827.190	1.826.273	184,000	1.826.273	917
1.721.860	17,0000	1.721.855	1.723.419	190,000	1.723.387	-1.532
1.854.746	17,0000	1.854.744	1.859.154	181,000	1.859.076	-4.332
1.799.294	19,0000	1.799.291	1.806.219	176,000	1.806.138	-6.847
1.827.194	18,5645	1.827.190	1.826.273	188,000	1.826.273	917
1.739.118	20,0000	1.739.115	1.745.629	182,000	1.745.575	-6.460
1.757.713	19,0000	1.757.710	1.767.682	192,000	1.767.587	-9.877
1.805.280	20,0000	1.805.278	1.809.044	176,000	1.808.966	-3.688
1.827.194	18,4567	1.827.192	1.826.273	187,000	1.826.273	919
1.860.723	17,0000	1.860.718	1.861.831	181,000	1.861.906	-1.188
1.861.240	15,0000	1.861.237	1.860.537	187,000	1.860.473	764
1.723.249	20,0000	1.723.246	1.722.745	173,000	1.722.812	434
1.827.194	21,2356	1.827.190	1.826.273	178,000	1.826.273	917
1.870.321	19,0000	1.870.316	1.878.835	183,000	1.878.793	-8.477
1.853.115	16,0000	1.853.112	1.861.401	192,000	1.861.330	-8.218
1.898.539	18,0000	1.898.536	1.906.034	182,000	1.906.033	-7.497
1.827.194	19,5674	1.827.190	1.826.273	180,000	1.826.273	917
1.887.924	21,0000	1.887.919	1.890.079	189,000	1.890.078	-2.159
1.829.713	23,0000	1.829.708	1.835.708	176,000	1.835.692	-5.984
1.730.498	22,0000	1.730.494	1.736.970	191,000	1.736.982	-6.488
1.827.194	19,5648	1.827.190	1.826.273	176,000	1.826.273	917
1.886.812	16,0000	1.886.810	1.894.040	185,000	1.894.005	-7.195
1.851.676	17,0000	1.851.674	1.857.266	189,000	1.857.208	-5.534
1.832.681	20,0000	1.832.677	1.837.444	182,000	1.837.460	-4.783
1.827.194	18,5897	1.827.189	1.826.273	193,000	1.826.273	916
1.792.532	17,0000	1.792.529	1.800.447	187,000	1.800.391	-7.862
1.829.108	22,0000	1.829.104	1.828.868	177,000	1.828.918	186
1.751.444	22,0000	1.751.439	1.756.826	176,000	1.756.917	-5.478

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.827.194	22,4567	1.827.189	1.826.273	180,000	1.826.273	916
1.853.421	16,0000	1.853.417	1.861.605	175,000	1.861.643	-8.226
1.723.452	22,0000	1.723.448	1.730.220	178,000	1.730.184	-6.736
1.883.463	23,0000	1.883.458	1.891.029	187,000	1.891.106	-7.648
1.827.194	15,4567	1.827.190	1.826.273	181,000	1.826.273	917
1.731.880	20,0000	1.731.877	1.737.261	186,000	1.737.337	-5.460
1.797.758	17,0000	1.797.756	1.805.533	183,000	1.805.629	-7.873
1.797.493	23,0000	1.797.490	1.804.491	187,000	1.804.546	-7.056
1.827.194	25,4567	1.827.191	1.826.273	188,000	1.826.273	918
1.841.944	21,5646	1.841.940	1.849.828	173,000	1.849.783	-7.843
1.876.258	22,0000	1.876.253	1.884.565	191,000	1.884.648	-8.395
1.760.660	17,0000	1.760.658	1.761.235	189,000	1.761.182	-524
1.827.194	18,5647	1.827.192	1.826.273	174,000	1.826.273	919
1.769.593	20,0000	1.769.589	1.768.950	177,000	1.768.985	604
1.858.728	17,0000	1.858.726	1.866.921	180,000	1.866.940	-8.214
1.820.072	18,0000	1.820.069	1.819.953	180,000	1.819.867	202
1.827.194	19,4564	1.827.192	1.826.273	184,000	1.826.273	919
1.721.830	23,0000	1.721.827	1.721.471	183,000	1.721.380	447
1.887.659	21,0000	1.887.655	1.896.210	186,000	1.896.163	-8.508
1.719.724	20,0000	1.719.720	1.723.336	189,000	1.723.407	-3.687
1.827.194	21,2365	1.827.192	1.826.273	189,000	1.826.273	919
1.857.966	16,0000	1.857.962	1.860.540	175,000	1.860.502	-2.540
1.883.382	16,0000	1.883.378	1.886.156	185,000	1.886.188	-2.810
1.846.359	16,0000	1.846.356	1.851.027	186,000	1.851.077	-4.721
1.827.194	17,5647	1.827.189	1.826.273	193,000	1.826.273	916
1.747.733	16,0000	1.747.730	1.752.944	188,000	1.753.014	-5.284
1.730.641	18,0000	1.730.638	1.737.056	183,000	1.737.104	-6.466
1.765.755	17,0000	1.765.752	1.771.470	188,000	1.771.415	-5.663
1.827.194	18,4564	1.827.190	1.826.273	180,000	1.826.273	917
1.795.308	23,0000	1.795.306	1.801.744	177,000	1.801.819	-6.513
1.825.805	23,0000	1.825.801	1.831.770	180,000	1.831.820	-6.019
1.863.675	18,0000	1.863.672	1.873.256	182,000	1.873.187	-9.515
1.827.194	19,2654	1.827.189	1.826.273	185,000	1.826.273	916
1.785.475	23,0000	1.785.472	1.785.497	192,000	1.785.433	39
1.798.672	18,0000	1.798.667	1.798.284	173,000	1.798.368	299
1.726.142	15,0000	1.726.139	1.727.273	178,000	1.727.213	-1.074

Figura 11.7: Resultados de simulación para instancia 50 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	20,9000	1.801.440	1.816.114	433,220	1.816.114	-14.674
1.902.430	17,0000	1.901.660	1.904.716	438,000	1.904.687	-3.027
1.854.561	19,0000	1.854.369	1.857.882	432,000	1.857.946	-3.577
1.716.033	16,0000	1.715.317	1.729.101	435,000	1.729.012	-13.695
1.802.335	17,5647	1.801.701	1.816.114	442,000	1.816.114	-14.413
1.771.032	19,0000	1.770.365	1.779.426	435,000	1.779.382	-9.017
1.755.334	22,0000	1.754.714	1.768.562	424,000	1.768.603	-13.889
1.897.533	16,0000	1.897.406	1.906.251	434,000	1.906.226	-8.820
1.802.335	18,2654	1.802.183	1.816.114	428,000	1.816.114	-13.931
1.793.280	21,0000	1.792.726	1.802.185	435,000	1.802.222	-9.496
1.859.333	23,0000	1.858.341	1.869.186	425,000	1.869.131	-10.790
1.844.090	17,0000	1.843.754	1.851.358	442,000	1.851.358	-7.604
1.802.335	16,4564	1.801.613	1.816.114	427,000	1.816.114	-14.501
1.884.773	21,0000	1.884.037	1.897.124	431,000	1.897.134	-13.097
1.785.667	15,0000	1.785.372	1.786.875	427,000	1.786.809	-1.437
1.892.097	22,0000	1.891.911	1.895.563	432,000	1.895.557	-3.646
1.802.335	19,5648	1.801.517	1.816.114	439,000	1.816.114	-14.597
1.888.266	19,0000	1.888.049	1.897.120	437,000	1.897.183	-9.134
1.720.453	23,0000	1.719.838	1.727.498	440,000	1.727.470	-7.632
1.881.100	18,0000	1.880.812	1.894.303	425,000	1.894.339	-13.527
1.802.335	20,2654	1.801.635	1.816.114	438,000	1.816.114	-14.479
1.788.969	15,0000	1.788.231	1.798.370	442,000	1.798.349	-10.118
1.744.254	19,0000	1.743.658	1.748.427	424,000	1.748.348	-4.690
1.856.848	23,0000	1.856.507	1.858.680	441,000	1.858.666	-2.159
1.802.335	22,2565	1.801.827	1.816.114	432,000	1.816.114	-14.287
1.826.826	15,0000	1.826.125	1.839.031	443,000	1.839.131	-13.006
1.875.265	21,0000	1.875.033	1.887.234	425,000	1.887.187	-12.154
1.879.968	19,0000	1.879.475	1.891.036	428,000	1.891.076	-11.601
1.802.335	21,4567	1.801.414	1.816.114	433,000	1.816.114	-14.700
1.863.837	20,0000	1.863.695	1.866.587	424,000	1.866.606	-2.911
1.768.256	21,0000	1.767.336	1.781.302	433,000	1.781.369	-14.033
1.869.732	17,0000	1.869.426	1.880.889	441,000	1.880.860	-11.434
1.802.335	18,4895	1.801.416	1.816.114	432,000	1.816.114	-14.698
1.851.439	19,0000	1.851.280	1.854.662	426,000	1.854.748	-3.468
1.817.897	18,0000	1.817.167	1.822.032	430,000	1.822.084	-4.917
1.840.718	15,0000	1.840.401	1.849.606	432,000	1.849.568	-9.167
1.802.335	16,5698	1.801.465	1.816.114	438,000	1.816.114	-14.649
1.829.938	19,0000	1.829.640	1.829.597	436,000	1.829.689	-49
1.761.076	23,0000	1.760.242	1.769.222	442,000	1.769.128	-8.886
1.875.408	19,0000	1.874.466	1.884.101	443,000	1.884.033	-9.567

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	16,4478	1.801.645	1.816.114	442,000	1.816.114	-14.469
1.871.183	18,0000	1.870.972	1.871.985	435,000	1.871.892	-920
1.791.517	15,0000	1.791.052	1.806.140	427,000	1.806.233	-15.181
1.867.635	17,0000	1.866.865	1.874.034	437,000	1.874.099	-7.234
1.802.335	21,2687	1.802.138	1.816.114	426,000	1.816.114	-13.976
1.885.411	15,0000	1.885.088	1.892.924	432,000	1.892.892	-7.804
1.832.230	20,0000	1.832.055	1.846.743	426,000	1.846.686	-14.631
1.774.116	20,0000	1.773.369	1.785.560	443,000	1.785.609	-12.240
1.802.335	14,2326	1.801.387	1.816.114	426,000	1.816.114	-14.727
1.856.665	18,0000	1.855.665	1.859.910	432,000	1.859.940	-4.275
1.844.019	21,0000	1.843.037	1.854.666	439,000	1.854.750	-11.713
1.824.997	17,0000	1.824.728	1.836.786	435,000	1.836.871	-12.143
1.802.335	18,9879	1.802.059	1.816.114	425,000	1.816.114	-14.055
1.734.785	17,0000	1.734.470	1.737.558	434,000	1.737.588	-3.118
1.770.894	17,0000	1.770.292	1.780.478	434,000	1.780.452	-10.160
1.728.495	22,0000	1.728.074	1.728.814	424,000	1.728.771	-697
1.802.335	18,7415	1.802.086	1.816.114	429,000	1.816.114	-14.028
1.819.942	17,0000	1.819.561	1.828.613	441,000	1.828.626	-9.065
1.770.658	15,0000	1.770.139	1.781.392	429,000	1.781.388	-11.249
1.866.331	17,0000	1.865.526	1.878.744	427,000	1.878.769	-13.243
1.802.335	16,4578	1.801.559	1.816.114	428,000	1.816.114	-14.555
1.902.632	19,0000	1.902.189	1.902.249	424,000	1.902.241	-52
1.899.450	18,0000	1.899.140	1.913.774	426,000	1.913.700	-14.560
1.850.838	21,0000	1.850.729	1.853.387	439,000	1.853.466	-2.737
1.802.335	19,5683	1.802.052	1.816.114	438,000	1.816.114	-14.062
1.778.809	17,0000	1.778.423	1.793.463	424,000	1.793.525	-15.102
1.881.786	15,0000	1.881.249	1.890.798	439,000	1.890.880	-9.631
1.776.706	17,0000	1.776.141	1.790.096	435,000	1.790.025	-13.884
1.802.335	21,2323	1.801.367	1.816.114	426,000	1.816.114	-14.747
1.895.906	22,0000	1.895.205	1.902.088	434,000	1.902.045	-6.840
1.766.680	17,0000	1.766.260	1.769.215	424,000	1.769.260	-3.000
1.758.889	17,0000	1.758.418	1.773.161	427,000	1.773.202	-14.784
1.802.335	18,5687	1.801.521	1.816.114	436,000	1.816.114	-14.593
1.817.019	23,0000	1.816.450	1.819.976	441,000	1.820.003	-3.553
1.889.790	19,0000	1.889.110	1.893.525	438,000	1.893.569	-4.459
1.836.253	18,0000	1.835.302	1.849.343	430,000	1.849.332	-14.030
1.802.335	21,2655	1.801.780	1.816.114	431,000	1.816.114	-14.334
1.761.767	19,0000	1.761.483	1.762.889	442,000	1.762.905	-1.422
1.844.930	23,0000	1.843.964	1.851.191	432,000	1.851.207	-7.243
1.813.198	22,0000	1.812.852	1.820.386	428,000	1.820.442	-7.590

Figura 11.8: Resultados de simulación para instancia 100 escenarios



N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	18,6443	1.800.408	1.813.638	433,220	1.813.735	-13.327
1.758.945	23,0000	1.758.332	1.771.895	431,000	1.771.887	-13.555
1.873.450	22,0000	1.871.967	1.887.034	432,000	1.886.959	-14.992
1.889.181	15,0000	1.887.318	1.889.869	438,000	1.889.951	-2.633
1.802.335	18,5962	1.801.577	1.813.638	435,000	1.813.690	-12.113
1.872.594	15,9686	1.871.569	1.875.259	437,000	1.875.356	-3.787
1.738.070	16,8491	1.736.552	1.744.153	425,000	1.744.163	-7.611
1.791.223	18,2761	1.790.416	1.795.563	431,000	1.795.596	-5.180
1.802.335	17,5150	1.800.583	1.813.638	442,000	1.813.595	-13.012
1.808.673	17,2838	1.808.134	1.808.840	428,000	1.808.859	-725
1.746.111	24,9549	1.745.440	1.759.416	429,000	1.759.404	-13.964
1.747.503	19,0668	1.745.512	1.749.065	436,000	1.749.076	-3.564
1.802.335	17,5150	1.800.703	1.813.638	425,000	1.813.663	-12.960
1.778.565	17,2838	1.777.631	1.794.245	432,000	1.794.214	-16.583
1.803.014	24,9549	1.802.247	1.805.779	435,000	1.805.820	-3.573
1.820.659	19,0668	1.820.365	1.833.473	440,000	1.833.489	-13.124
1.802.335	18,1929	1.800.989	1.813.638	433,000	1.813.589	-12.600
1.793.320	17,7100	1.791.817	1.798.684	431,000	1.798.749	-6.932
1.720.530	20,0094	1.720.260	1.739.860	426,000	1.739.836	-19.576
1.767.097	19,0831	1.765.740	1.784.053	441,000	1.784.057	-18.317
1.802.335	18,5962	1.801.362	1.813.638	443,000	1.813.689	-12.327
1.779.872	15,9686	1.779.640	1.787.203	429,000	1.787.211	-7.571
1.728.387	16,8491	1.727.110	1.728.801	429,000	1.728.828	-1.718
1.780.397	19,0000	1.778.458	1.795.093	437,000	1.795.091	-16.633
1.802.335	24,9549	1.800.511	1.813.638	425,000	1.813.605	-13.094
1.885.973	19,0668	1.885.341	1.895.768	432,000	1.895.765	-10.424
1.790.721	18,5962	1.788.802	1.805.608	428,000	1.805.683	-16.881
1.765.366	15,9686	1.763.422	1.777.676	431,000	1.777.577	-14.155
1.802.335	16,8491	1.801.079	1.813.638	430,000	1.813.583	-12.504
1.772.353	18,2761	1.772.119	1.792.003	432,000	1.791.931	-19.812
1.843.380	18,4890	1.842.523	1.856.183	425,000	1.856.216	-13.693
1.872.980	16,0000	1.872.004	1.886.204	439,000	1.886.180	-14.176
1.802.335	16,8491	1.800.620	1.813.638	424,000	1.813.544	-12.924
1.849.352	18,2761	1.847.413	1.853.908	432,000	1.853.825	-6.412
1.846.729	18,4890	1.845.977	1.865.918	432,000	1.865.894	-19.917
1.733.643	19,1576	1.732.462	1.749.109	426,000	1.749.122	-16.660
1.802.335	17,8945	1.801.917	1.813.638	434,000	1.813.572	-11.655
1.793.784	22,5555	1.792.942	1.801.321	426,000	1.801.242	-8.300
1.861.374	21,2090	1.860.534	1.877.088	424,000	1.877.141	-16.607
1.867.505	15,7431	1.866.786	1.869.835	431,000	1.869.865	-3.079

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	16,8245	1.800.490	1.813.638	432,000	1.813.556	-13.066
1.875.087	20,0000	1.873.396	1.884.074	434,000	1.883.977	-10.581
1.848.953	23,0000	1.847.956	1.849.985	435,000	1.849.924	-1.968
1.758.009	19,0000	1.757.793	1.777.299	440,000	1.777.358	-19.565
1.802.335	18,5962	1.801.194	1.813.638	430,000	1.813.559	-12.365
1.770.839	15,9686	1.770.402	1.779.667	424,000	1.779.707	-9.305
1.848.983	16,8491	1.847.402	1.858.834	431,000	1.858.834	-11.432
1.828.304	18,2761	1.827.210	1.842.233	425,000	1.842.182	-14.972
1.802.335	17,5150	1.801.347	1.813.638	434,000	1.813.613	-12.266
1.828.796	18,2450	1.828.196	1.843.073	436,000	1.843.030	-14.834
1.770.088	24,9549	1.769.635	1.772.149	425,000	1.772.084	-2.449
1.768.842	19,0668	1.767.088	1.785.522	427,000	1.785.430	-18.342
1.802.335	18,5962	1.800.639	1.813.638	429,000	1.813.734	-13.095
1.815.141	15,9686	1.814.765	1.824.613	431,000	1.824.555	-9.790
1.802.216	16,8491	1.801.510	1.813.078	434,000	1.813.157	-11.647
1.849.606	18,2761	1.848.882	1.866.948	438,000	1.866.992	-18.110
1.802.335	18,4890	1.801.022	1.813.638	435,000	1.813.670	-12.648
1.881.758	19,1576	1.880.509	1.900.455	435,000	1.900.418	-19.909
1.887.346	17,8945	1.887.123	1.906.699	432,000	1.906.704	-19.581
1.749.259	22,5555	1.748.181	1.751.404	428,000	1.751.339	-3.158
1.802.335	21,2090	1.800.906	1.813.638	424,000	1.813.697	-12.791
1.744.299	15,7431	1.743.866	1.757.099	427,000	1.757.008	-13.142
1.842.457	16,8245	1.841.267	1.845.694	428,000	1.845.694	-4.427
1.865.507	17,6360	1.864.771	1.877.225	441,000	1.877.307	-12.536
1.802.335	17,5150	1.800.936	1.813.638	428,000	1.813.583	-12.647
1.754.252	17,2838	1.752.970	1.758.447	425,000	1.758.496	-5.526
1.798.868	24,9549	1.797.553	1.817.748	442,000	1.817.747	-20.194
1.778.807	19,0668	1.777.555	1.790.696	428,000	1.790.756	-13.201
1.802.335	18,4890	1.801.837	1.813.638	440,000	1.813.593	-11.756
1.730.532	19,1576	1.729.343	1.743.652	424,000	1.743.727	-14.384
1.876.545	18,4554	1.874.729	1.886.667	435,000	1.886.754	-12.025
1.774.845	22,5555	1.774.081	1.792.932	440,000	1.792.899	-18.818
1.802.335	18,1929	1.801.716	1.813.638	438,000	1.813.647	-11.931
1.834.690	17,7100	1.834.183	1.834.990	439,000	1.835.064	-881
1.752.917	23,0000	1.752.607	1.764.824	425,000	1.764.919	-12.312
1.814.781	21,0000	1.812.985	1.832.690	429,000	1.832.775	-19.790
1.802.335	18,1929	1.801.639	1.813.638	443,000	1.813.721	-12.082
1.742.034	23,0000	1.740.926	1.742.570	425,000	1.742.547	-1.621
1.752.312	22,0000	1.750.791	1.766.260	442,000	1.766.232	-15.441
1.759.008	20,0000	1.757.902	1.758.809	424,000	1.758.761	-859

Figura 11.9: Resultados de simulación para instancia 200 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.795.312	22,2132	1.794.725	1.806.613	854,000	1.806.661	-11.936
1.799.268	19,0000	1.798.827	1.800.778	844,000	1.800.866	-2.039
1.883.534	22,0000	1.882.382	1.886.387	864,000	1.886.348	-3.966
1.787.234	18,0000	1.786.448	1.804.522	852,000	1.804.558	-18.110
1.795.312	22,2132	1.793.577	1.806.613	853,000	1.806.645	-13.068
1.837.649	22,0000	1.836.670	1.842.153	864,000	1.842.094	-5.424
1.757.668	20,0000	1.756.204	1.762.414	846,000	1.762.495	-6.291
1.888.315	17,0000	1.887.804	1.905.408	845,000	1.905.370	-17.566
1.795.312	22,2132	1.793.349	1.806.613	852,000	1.806.593	-13.244
1.814.352	19,0000	1.813.025	1.815.194	860,000	1.815.161	-2.136
1.841.506	21,0000	1.841.009	1.849.047	854,000	1.849.052	-8.043
1.739.884	23,0000	1.738.694	1.757.449	848,000	1.757.421	-18.727
1.795.312	22,2132	1.793.892	1.806.613	862,000	1.806.545	-12.653
1.752.158	22,0000	1.751.745	1.758.699	858,000	1.758.607	-6.862
1.879.672	17,0000	1.879.084	1.886.342	860,000	1.886.432	-7.348
1.898.003	17,0000	1.897.161	1.907.865	847,000	1.907.809	-10.648
1.795.312	22,2132	1.793.386	1.806.613	863,000	1.806.632	-13.246
1.785.443	16,0000	1.783.592	1.789.869	861,000	1.789.939	-6.347
1.825.100	21,0000	1.823.542	1.829.516	857,000	1.829.545	-6.003
1.880.744	18,0000	1.880.295	1.898.203	853,000	1.898.298	-18.003
1.795.312	22,2132	1.793.502	1.806.613	855,000	1.806.640	-13.138
1.881.450	18,0000	1.880.669	1.881.766	845,000	1.881.694	-1.025
1.896.596	16,0000	1.895.255	1.910.231	851,000	1.910.235	-14.980
1.830.131	16,0000	1.829.209	1.834.623	849,000	1.834.564	-5.355
1.795.312	22,2132	1.795.047	1.806.613	844,000	1.806.607	-11.560
1.765.004	21,0000	1.763.435	1.782.401	848,000	1.782.313	-18.878
1.736.960	15,0000	1.735.272	1.742.645	852,000	1.742.571	-7.299
1.746.234	18,0000	1.745.856	1.746.599	850,000	1.746.598	-742
1.795.312	22,2132	1.794.625	1.806.613	848,000	1.806.712	-12.087
1.816.412	21,0000	1.815.758	1.830.420	845,000	1.830.346	-14.588
1.843.121	21,0000	1.841.954	1.846.538	846,000	1.846.517	-4.563
1.720.622	19,0000	1.719.563	1.720.632	848,000	1.720.537	-974
1.795.312	22,2132	1.793.412	1.806.613	865,000	1.806.696	-13.284
1.737.080	15,0000	1.735.482	1.742.991	857,000	1.743.058	-7.576
1.740.914	15,0000	1.739.679	1.759.576	859,000	1.759.558	-19.879
1.724.863	20,0000	1.724.407	1.731.450	863,000	1.731.470	-7.063
1.795.312	22,2132	1.794.027	1.806.613	865,000	1.806.576	-12.549
1.716.858	21,0000	1.716.032	1.738.811	857,000	1.738.833	-22.801
1.837.969	17,0000	1.836.217	1.848.105	860,000	1.848.159	-11.942
1.774.024	23,0000	1.772.440	1.774.508	862,000	1.774.488	-2.048

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.795.312	22,2132	1.793.324	1.806.613	844,000	1.806.645	-13.321
1.766.934	21,0000	1.765.448	1.788.159	856,000	1.788.208	-22.760
1.867.108	18,0000	1.866.257	1.873.687	843,000	1.873.641	-7.384
1.743.144	21,0000	1.741.578	1.758.581	862,000	1.758.542	-16.964
1.795.312	22,2132	1.793.904	1.806.613	850,000	1.806.537	-12.633
1.900.527	17,0000	1.898.786	1.906.844	847,000	1.906.765	-7.979
1.803.459	17,0000	1.801.950	1.807.286	861,000	1.807.311	-5.361
1.737.641	21,0000	1.735.769	1.754.547	864,000	1.754.608	-18.839
1.795.312	22,2132	1.794.156	1.806.613	851,000	1.806.525	-12.369
1.764.097	16,0000	1.763.074	1.776.386	848,000	1.776.457	-13.383
1.801.637	21,0000	1.801.209	1.802.614	854,000	1.802.635	-1.426
1.811.415	18,0000	1.810.528	1.824.121	851,000	1.824.056	-13.528
1.795.312	22,2132	1.794.806	1.806.613	862,000	1.806.592	-11.786
1.732.078	21,0000	1.730.120	1.737.982	856,000	1.737.887	-7.767
1.725.632	21,0000	1.725.235	1.730.518	859,000	1.730.605	-5.370
1.897.297	18,0000	1.896.624	1.917.050	844,000	1.917.034	-20.410
1.795.312	22,2132	1.794.838	1.806.613	854,000	1.806.532	-11.694
1.771.211	17,0000	1.770.804	1.776.337	848,000	1.776.289	-5.485
1.845.816	18,0000	1.845.444	1.866.758	852,000	1.866.823	-21.379
1.865.303	22,0000	1.863.520	1.880.581	864,000	1.880.559	-17.039
1.795.312	22,2132	1.794.645	1.806.613	861,000	1.806.579	-11.934
1.819.836	17,0000	1.818.713	1.820.005	864,000	1.819.969	-1.256
1.774.734	20,0000	1.773.787	1.789.592	850,000	1.789.596	-15.809
1.775.067	16,0000	1.774.334	1.796.706	858,000	1.796.611	-22.277
1.795.312	22,2132	1.793.724	1.806.613	846,000	1.806.690	-12.966
1.735.993	20,0000	1.735.727	1.738.649	865,000	1.738.647	-2.920
1.752.837	21,0000	1.751.148	1.766.229	846,000	1.766.293	-15.145
1.722.909	19,0000	1.721.935	1.727.949	853,000	1.727.936	-6.001
1.795.312	22,2132	1.793.826	1.806.613	848,000	1.806.706	-12.880
1.831.015	19,0000	1.830.582	1.847.807	858,000	1.847.813	-17.231
1.792.018	19,0000	1.790.375	1.799.087	865,000	1.799.030	-8.655
1.892.657	16,0000	1.891.680	1.898.579	860,000	1.898.491	-6.811
1.795.312	22,2132	1.794.548	1.806.613	845,000	1.806.632	-12.084
1.794.633	23,0000	1.794.135	1.807.488	851,000	1.807.465	-13.330
1.876.570	19,0000	1.874.574	1.886.816	857,000	1.886.798	-12.224
1.842.951	23,0000	1.842.573	1.845.260	850,000	1.845.276	-2.703
1.795.312	22,2132	1.794.026	1.806.613	851,000	1.806.656	-12.630
1.818.318	18,0000	1.816.398	1.834.847	858,000	1.834.778	-18.380
1.783.692	23,0000	1.781.841	1.798.879	865,000	1.798.921	-17.080
1.771.301	21,0000	1.770.180	1.779.694	843,000	1.779.750	-9.570

Figura 11.10: Resultados de simulación para instancia 300 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	18,5962	1.800.822	1.815.698	2.414,98	1.815.626	-14.804
1.822.199	17,0000	1.819.848	1.825.012	2.406,00	1.825.074	-5.226
1.895.272	23,0000	1.892.086	1.895.922	2.411,00	1.895.973	-3.887
1.855.869	19,0000	1.853.254	1.873.911	2.407,00	1.873.914	-20.660
1.802.335	17,0000	1.799.200	1.815.698	2.413,00	1.815.662	-16.462
1.759.843	20,0000	1.757.721	1.760.740	2.411,00	1.760.641	-2.920
1.838.884	21,0000	1.835.771	1.848.524	2.409,00	1.848.522	-12.751
1.902.378	18,0000	1.901.543	1.914.894	2.417,00	1.914.821	-13.278
1.802.335	18,0000	1.799.262	1.815.698	2.413,00	1.815.680	-16.418
1.879.729	18,0000	1.878.557	1.881.368	2.411,00	1.881.294	-2.737
1.837.623	15,0000	1.836.517	1.847.988	2.407,00	1.848.067	-11.550
1.828.702	19,0000	1.826.758	1.844.949	2.407,00	1.844.857	-18.099
1.802.335	19,0000	1.799.902	1.815.698	2.415,00	1.815.783	-15.881
1.817.029	21,0000	1.815.314	1.828.554	2.424,00	1.828.615	-13.301
1.885.299	17,0000	1.884.804	1.900.884	2.421,00	1.900.805	-16.001
1.804.174	17,0000	1.803.198	1.818.595	2.410,00	1.818.501	-15.303
1.802.335	16,0000	1.801.013	1.815.698	2.409,00	1.815.672	-14.659
1.781.466	17,0000	1.778.225	1.782.308	2.422,00	1.782.245	-4.020
1.902.704	18,0000	1.899.885	1.920.059	2.408,00	1.920.059	-20.174
1.881.710	23,0000	1.879.916	1.888.565	2.409,00	1.888.588	-8.672
1.802.335	16,0000	1.801.722	1.815.698	2.419,00	1.815.784	-14.062
1.805.448	19,0000	1.805.030	1.822.797	2.405,00	1.822.844	-17.814
1.794.021	23,0000	1.791.775	1.798.703	2.407,00	1.798.732	-6.957
1.868.268	23,0000	1.867.026	1.882.125	2.425,00	1.882.026	-15.000
1.802.335	17,0000	1.801.256	1.815.698	2.404,00	1.815.698	-14.442
1.894.041	22,0000	1.893.382	1.901.886	2.423,00	1.901.813	-8.431
1.854.313	19,0000	1.851.755	1.859.343	2.405,00	1.859.277	-7.522
1.786.689	16,0000	1.782.964	1.792.530	2.418,00	1.792.476	-9.512
1.802.335	22,0000	1.799.770	1.815.698	2.412,00	1.815.764	-15.994
1.824.567	20,0000	1.822.237	1.843.514	2.416,00	1.843.604	-21.367
1.854.354	23,0000	1.852.915	1.862.171	2.417,00	1.862.225	-9.310
1.853.138	19,0000	1.850.116	1.871.030	2.407,00	1.871.061	-20.945
1.802.335	22,0000	1.799.711	1.815.698	2.425,00	1.815.628	-15.917
1.873.940	18,0000	1.872.668	1.893.328	2.414,00	1.893.350	-20.682
1.720.330	20,0000	1.717.094	1.734.976	2.415,00	1.734.995	-17.901
1.767.388	20,0000	1.766.674	1.768.609	2.407,00	1.768.641	-1.967
1.802.335	15,0000	1.799.419	1.815.698	2.415,00	1.815.711	-16.292
1.854.571	23,0000	1.853.177	1.863.115	2.405,00	1.863.109	-9.932
1.878.157	20,0000	1.874.462	1.883.226	2.413,00	1.883.211	-8.749
1.820.098	17,0000	1.819.037	1.833.250	2.411,00	1.833.169	-14.132

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	16,0000	1.799.645	1.815.698	2.412,00	1.815.716	-16.071
1.827.286	23,0000	1.825.283	1.830.414	2.424,00	1.830.456	-5.173
1.845.091	16,0000	1.844.672	1.859.462	2.409,00	1.859.523	-14.851
1.898.175	16,0000	1.895.635	1.903.646	2.424,00	1.903.623	-7.988
1.802.335	15,0000	1.799.590	1.815.698	2.414,00	1.815.749	-16.159
1.760.808	19,0000	1.758.865	1.765.560	2.419,00	1.765.505	-6.640
1.731.749	16,0000	1.728.898	1.735.985	2.418,00	1.736.041	-7.143
1.780.361	21,0000	1.779.897	1.799.489	2.417,00	1.799.437	-19.540
1.802.335	17,0000	1.801.558	1.815.698	2.404,00	1.815.718	-14.160
1.891.188	16,0000	1.888.886	1.895.344	2.416,00	1.895.273	-6.387
1.885.081	17,0000	1.882.020	1.902.469	2.415,00	1.902.524	-20.504
1.881.334	18,0000	1.879.330	1.887.869	2.421,00	1.887.885	-8.555
1.802.335	22,0000	1.800.034	1.815.698	2.425,00	1.815.691	-15.657
1.812.976	18,0000	1.810.035	1.812.960	2.415,00	1.812.998	-2.963
1.753.666	18,0000	1.753.045	1.763.705	2.408,00	1.763.763	-10.718
1.832.224	23,0000	1.828.633	1.845.228	2.422,00	1.845.140	-16.507
1.802.335	17,0000	1.799.636	1.815.698	2.420,00	1.815.601	-15.965
1.896.958	16,0000	1.894.300	1.915.395	2.412,00	1.915.400	-21.100
1.887.217	22,0000	1.885.725	1.897.148	2.409,00	1.897.085	-11.360
1.823.320	16,0000	1.819.836	1.831.137	2.419,00	1.831.192	-11.356
1.802.335	17,0000	1.799.135	1.815.698	2.418,00	1.815.695	-16.560
1.780.193	15,0000	1.779.437	1.800.165	2.421,00	1.800.115	-20.678
1.813.086	16,0000	1.809.641	1.817.972	2.406,00	1.817.924	-8.283
1.827.829	18,0000	1.826.249	1.831.222	2.425,00	1.831.315	-5.066
1.802.335	22,0000	1.799.335	1.815.698	2.406,00	1.815.796	-16.461
1.845.706	21,0000	1.841.836	1.858.845	2.410,00	1.858.830	-16.994
1.730.486	17,0000	1.728.993	1.730.659	2.417,00	1.730.584	-1.591
1.718.664	20,0000	1.716.374	1.733.390	2.425,00	1.733.315	-16.941
1.802.335	16,0000	1.800.996	1.815.698	2.415,00	1.815.629	-14.633
1.807.438	15,0000	1.805.856	1.825.293	2.422,00	1.825.284	-19.428
1.873.800	20,0000	1.871.271	1.891.198	2.413,00	1.891.159	-19.888
1.899.743	22,0000	1.896.000	1.910.236	2.418,00	1.910.306	-14.306
1.802.335	15,0000	1.801.180	1.815.698	2.415,00	1.815.732	-14.552
1.883.085	18,0000	1.881.584	1.893.204	2.418,00	1.893.255	-11.671
1.859.231	20,0000	1.858.127	1.866.326	2.424,00	1.866.424	-8.297
1.730.229	21,0000	1.726.762	1.745.295	2.414,00	1.745.348	-18.586
1.802.335	21,0000	1.800.606	1.815.698	2.404,00	1.815.724	-15.118
1.826.763	19,0000	1.824.948	1.837.689	2.415,00	1.837.657	-12.709
1.884.877	19,0000	1.884.120	1.898.815	2.423,00	1.898.873	-14.753
1.861.324	15,0000	1.860.897	1.876.546	2.421,00	1.876.578	-15.681

Figura 11.11: Resultados de simulación para instancia 400 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	21,5274	1.800.972	1.815.078	2.733,94	1.815.145	-14.173
1.742.380	20,0000	1.740.865	1.764.970	2.734,00	1.765.007	-24.142
1.791.724	21,0000	1.790.900	1.808.889	2.735,00	1.808.848	-17.948
1.753.635	20,0000	1.751.946	1.754.418	2.739,00	1.754.469	-2.523
1.802.335	17,0000	1.801.467	1.815.078	2.739,00	1.815.024	-13.557
1.779.399	21,0000	1.777.870	1.802.110	2.726,00	1.802.090	-24.220
1.796.426	22,2132	1.795.670	1.818.673	2.727,00	1.818.616	-22.946
1.777.438	16,0000	1.775.618	1.781.855	2.727,00	1.781.952	-6.334
1.802.335	21,0000	1.802.070	1.815.078	2.741,00	1.815.123	-13.053
1.782.364	18,0000	1.780.635	1.805.228	2.742,00	1.805.285	-24.650
1.818.209	18,0000	1.817.968	1.831.296	2.738,00	1.831.317	-13.349
1.825.885	23,0000	1.825.198	1.833.344	2.744,00	1.833.317	-8.119
1.802.335	21,0000	1.800.949	1.815.078	2.742,00	1.815.134	-14.185
1.796.143	18,0000	1.795.286	1.813.240	2.724,00	1.813.324	-18.038
1.806.133	22,2132	1.804.201	1.817.819	2.736,00	1.817.828	-13.627
1.845.702	17,0000	1.844.690	1.852.301	2.744,00	1.852.250	-7.560
1.802.335	18,0000	1.801.884	1.815.078	2.730,00	1.815.023	-13.139
1.772.796	22,0000	1.770.798	1.785.969	2.729,00	1.785.936	-15.138
1.901.616	22,2132	1.900.168	1.904.047	2.735,00	1.904.139	-3.971
1.805.915	19,0000	1.805.466	1.821.834	2.740,00	1.821.783	-16.317
1.802.335	17,0000	1.801.056	1.815.078	2.735,00	1.815.115	-14.059
1.799.353	23,0000	1.798.994	1.810.056	2.738,00	1.809.984	-10.990
1.801.540	22,2132	1.800.110	1.808.126	2.744,00	1.808.198	-8.088
1.815.488	21,0000	1.815.275	1.835.974	2.735,00	1.835.925	-20.650
1.802.335	18,0000	1.800.974	1.815.078	2.734,00	1.815.018	-14.044
1.750.428	21,0000	1.749.614	1.760.058	2.738,00	1.759.975	-10.361
1.750.278	22,2132	1.748.900	1.765.250	2.734,00	1.765.230	-16.330
1.880.271	17,0000	1.879.213	1.889.564	2.738,00	1.889.613	-10.400
1.802.335	17,0000	1.801.300	1.815.078	2.733,00	1.815.175	-13.875
1.788.388	21,0000	1.787.649	1.797.937	2.729,00	1.797.938	-10.289
1.718.513	22,2132	1.716.932	1.735.308	2.727,00	1.735.226	-18.294
1.882.440	16,0000	1.881.598	1.900.105	2.737,00	1.900.025	-18.427
1.802.335	21,0000	1.801.966	1.815.078	2.742,00	1.815.030	-13.064
1.744.952	18,0000	1.744.570	1.762.448	2.728,00	1.762.541	-17.971
1.835.874	22,2132	1.834.726	1.836.739	2.730,00	1.836.689	-1.963
1.745.249	21,0000	1.744.893	1.746.448	2.740,00	1.746.371	-1.478
1.802.335	21,0000	1.801.551	1.815.078	2.725,00	1.815.032	-13.481
1.775.694	18,0000	1.775.397	1.788.195	2.730,00	1.788.260	-12.863
1.887.560	17,0000	1.886.115	1.905.716	2.740,00	1.905.753	-19.638
1.781.253	18,0000	1.780.662	1.790.671	2.727,00	1.790.694	-10.032

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	22,2132	1.801.042	1.815.078	2.740,00	1.815.069	-14.027
1.755.135	19,0000	1.754.154	1.766.554	2.744,00	1.766.613	-12.459
1.739.101	23,0000	1.738.307	1.742.330	2.741,00	1.742.419	-4.112
1.741.222	16,0000	1.740.459	1.753.898	2.728,00	1.753.967	-13.508
1.802.335	21,2354	1.801.442	1.815.078	2.724,00	1.815.164	-13.722
1.901.743	16,0000	1.900.622	1.917.355	2.729,00	1.917.292	-16.670
1.850.603	23,0000	1.849.913	1.864.872	2.728,00	1.864.899	-14.986
1.796.569	20,0000	1.795.169	1.809.373	2.743,00	1.809.458	-14.289
1.802.335	23,5677	1.800.680	1.815.078	2.738,00	1.815.034	-14.354
1.902.042	22,0000	1.901.529	1.923.174	2.741,00	1.923.202	-21.673
1.828.252	19,0000	1.827.288	1.830.312	2.725,00	1.830.336	-3.048
1.852.984	22,0000	1.852.673	1.862.819	2.725,00	1.862.740	-10.067
1.802.335	21,2345	1.801.267	1.815.078	2.736,00	1.815.141	-13.874
1.760.014	17,0000	1.758.447	1.774.778	2.739,00	1.774.816	-16.369
1.794.027	15,0000	1.793.595	1.816.994	2.726,00	1.816.982	-23.387
1.733.760	15,0000	1.733.540	1.737.137	2.730,00	1.737.219	-3.679
1.802.335	22,2132	1.801.469	1.815.078	2.731,00	1.815.053	-13.584
1.829.089	17,0000	1.827.986	1.850.319	2.724,00	1.850.227	-22.241
1.744.282	23,0000	1.743.125	1.766.839	2.731,00	1.766.834	-23.709
1.784.600	17,0000	1.783.427	1.785.785	2.741,00	1.785.799	-2.372
1.802.335	16,4568	1.802.117	1.815.078	2.729,00	1.815.173	-13.056
1.821.810	17,0000	1.819.925	1.825.835	2.724,00	1.825.814	-5.889
1.749.841	20,0000	1.749.348	1.752.913	2.743,00	1.752.840	-3.492
1.867.226	22,0000	1.865.811	1.869.729	2.738,00	1.869.676	-3.865
1.802.335	18,4567	1.801.057	1.815.078	2.723,00	1.815.027	-13.970
1.737.637	20,0000	1.737.147	1.757.382	2.727,00	1.757.408	-20.261
1.765.531	20,0000	1.765.097	1.783.462	2.734,00	1.783.430	-18.333
1.876.823	20,0000	1.875.893	1.890.208	2.729,00	1.890.243	-14.350
1.802.335	19,4564	1.801.160	1.815.078	2.735,00	1.815.057	-13.897
1.842.640	19,0000	1.842.322	1.859.594	2.738,00	1.859.667	-17.345
1.869.281	18,0000	1.868.632	1.876.129	2.737,00	1.876.053	-7.421
1.800.823	17,0000	1.800.508	1.804.196	2.726,00	1.804.161	-3.653
1.802.335	22,2132	1.801.162	1.815.078	2.743,00	1.815.166	-14.004
1.823.767	20,0000	1.822.085	1.825.635	2.724,00	1.825.633	-3.548
1.826.146	22,0000	1.825.807	1.838.043	2.732,00	1.838.128	-12.321
1.845.375	16,0000	1.843.902	1.868.100	2.741,00	1.868.043	-24.141
1.802.335	21,2345	1.801.264	1.815.078	2.724,00	1.815.093	-13.829
1.788.431	19,0000	1.786.715	1.808.668	2.737,00	1.808.617	-21.902
1.751.981	17,0000	1.750.094	1.771.160	2.738,00	1.771.072	-20.978
1.848.714	21,0000	1.847.737	1.865.352	2.735,00	1.865.357	-17.620

Figura 11.12: Resultados de simulación para instancia 500 escenarios



N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	21,5274	1.798.409	1.819.383	50.567,37	1.819.291	-20.882
1.870.965	20,0000	1.867.982	1.880.234	50.565,00	1.880.242	-12.260
1.797.864	21,0000	1.791.023	1.807.452	50.574,00	1.807.500	-16.477
1.818.569	20,0000	1.812.113	1.834.174	50.570,00	1.834.164	-22.051
1.802.335	17,0000	1.801.211	1.822.434	50.577,00	1.822.415	-21.204
1.749.240	21,0000	1.743.939	1.767.585	50.577,00	1.767.674	-23.735
1.893.601	22,2132	1.887.305	1.900.262	50.577,00	1.900.271	-12.966
1.740.899	16,0000	1.737.411	1.745.271	50.578,00	1.745.173	-7.762
1.802.335	21,0000	1.798.500	1.816.065	50.561,00	1.816.105	-17.605
1.793.759	18,0000	1.790.600	1.802.805	50.558,00	1.802.719	-12.119
1.782.381	18,0000	1.775.758	1.804.662	50.562,00	1.804.594	-28.836
1.869.399	23,0000	1.864.433	1.876.261	50.564,00	1.876.214	-11.781
1.802.335	21,0000	1.796.964	1.822.495	50.574,00	1.822.430	-25.466
1.858.366	18,0000	1.857.429	1.873.233	50.575,00	1.873.323	-15.894
1.734.030	22,2132	1.728.109	1.736.480	50.567,00	1.736.411	-8.302
1.893.945	17,0000	1.888.958	1.893.929	50.561,00	1.893.919	-4.961
1.802.335	18,0000	1.798.142	1.812.741	50.574,00	1.812.769	-14.627
1.735.490	22,0000	1.734.437	1.744.585	50.577,00	1.744.553	-10.116
1.737.585	22,2132	1.736.604	1.759.225	50.569,00	1.759.217	-22.613
1.805.096	19,0000	1.802.592	1.814.261	50.564,00	1.814.167	-11.575
1.802.335	17,0000	1.796.798	1.823.183	50.577,00	1.823.253	-26.455
1.765.578	23,0000	1.762.561	1.786.059	50.570,00	1.786.024	-23.463
1.751.580	22,2132	1.749.006	1.761.347	50.559,00	1.761.247	-12.241
1.769.767	21,0000	1.768.219	1.788.616	50.569,00	1.788.570	-20.351
1.802.335	18,0000	1.795.884	1.819.835	50.560,00	1.819.886	-24.002
1.797.534	21,0000	1.795.453	1.810.341	50.568,00	1.810.425	-14.972
1.900.685	22,2132	1.899.289	1.911.330	50.578,00	1.911.241	-11.952
1.734.633	17,0000	1.731.859	1.737.241	50.577,00	1.737.169	-5.310
1.802.335	17,0000	1.797.585	1.805.167	50.575,00	1.805.120	-7.535
1.753.827	21,0000	1.748.755	1.756.082	50.569,00	1.756.142	-7.387
1.886.990	22,2132	1.883.931	1.894.445	50.560,00	1.894.536	-10.605
1.897.274	16,0000	1.896.004	1.903.529	50.566,00	1.903.479	-7.475
1.802.335	21,0000	1.795.831	1.812.271	50.558,00	1.812.344	-16.513
1.842.606	18,0000	1.839.657	1.856.588	50.569,00	1.856.529	-16.872
1.901.047	22,2132	1.897.219	1.903.141	50.578,00	1.903.120	-5.901
1.777.790	21,0000	1.773.337	1.779.875	50.567,00	1.779.793	-6.456
1.802.335	21,0000	1.799.939	1.815.607	50.566,00	1.815.686	-15.747
1.845.562	18,0000	1.839.777	1.850.838	50.571,00	1.850.740	-10.963
1.869.828	17,0000	1.864.309	1.878.639	50.563,00	1.878.612	-14.303
1.740.020	18,0000	1.738.709	1.744.609	50.557,00	1.744.570	-5.861

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	22,2132	1.801.273	1.809.952	50.570,00	1.809.945	-8.672
1.822.067	19,0000	1.819.865	1.843.323	50.577,00	1.843.390	-23.525
1.785.042	23,0000	1.784.273	1.795.413	50.575,00	1.795.422	-11.149
1.893.524	16,0000	1.890.173	1.894.653	50.565,00	1.894.593	-4.420
1.802.335	21,2354	1.797.874	1.811.995	50.570,00	1.812.053	-14.179
1.719.163	16,0000	1.714.411	1.729.851	50.559,00	1.729.791	-15.380
1.783.346	23,0000	1.782.360	1.803.900	50.572,00	1.803.889	-21.529
1.872.624	20,0000	1.869.060	1.880.946	50.571,00	1.881.016	-11.956
1.802.335	23,5677	1.795.403	1.816.886	50.566,00	1.816.958	-21.555
1.791.268	22,0000	1.789.744	1.792.629	50.568,00	1.792.539	-2.795
1.836.447	19,0000	1.829.580	1.845.392	50.570,00	1.845.374	-15.794
1.835.301	22,0000	1.829.959	1.844.428	50.565,00	1.844.524	-14.565
1.802.335	21,2345	1.800.347	1.812.573	50.561,00	1.812.616	-12.269
1.790.585	17,0000	1.786.359	1.791.661	50.558,00	1.791.736	-5.377
1.765.495	15,0000	1.760.708	1.782.344	50.569,00	1.782.425	-21.717
1.797.193	15,0000	1.794.655	1.801.966	50.569,00	1.801.997	-7.342
1.802.335	22,2132	1.801.302	1.806.989	50.572,00	1.806.952	-5.650
1.721.411	17,0000	1.718.677	1.722.362	50.571,00	1.722.316	-3.639
1.762.174	23,0000	1.755.608	1.765.641	50.574,00	1.765.651	-10.043
1.785.120	17,0000	1.783.698	1.804.089	50.570,00	1.804.171	-20.473
1.802.335	16,4568	1.799.657	1.819.244	50.578,00	1.819.228	-19.571
1.859.648	17,0000	1.853.053	1.869.187	50.557,00	1.869.151	-16.098
1.866.339	20,0000	1.861.459	1.877.355	50.558,00	1.877.337	-15.878
1.813.914	22,0000	1.811.132	1.827.206	50.560,00	1.827.249	-16.117
1.802.335	18,4567	1.799.792	1.813.409	50.577,00	1.813.335	-13.543
1.755.141	20,0000	1.753.792	1.772.583	50.571,00	1.772.655	-18.863
1.784.420	20,0000	1.781.905	1.806.866	50.558,00	1.806.786	-24.881
1.866.018	20,0000	1.859.239	1.874.343	50.562,00	1.874.292	-15.053
1.802.335	19,4564	1.800.229	1.811.449	50.578,00	1.811.442	-11.213
1.771.875	19,0000	1.769.951	1.775.884	50.561,00	1.775.954	-6.003
1.862.166	18,0000	1.858.117	1.866.007	50.578,00	1.866.103	-7.986
1.756.597	17,0000	1.753.910	1.764.182	50.567,00	1.764.097	-10.187
1.802.335	22,2132	1.799.347	1.815.694	50.563,00	1.815.674	-16.327
1.888.353	20,0000	1.882.573	1.902.977	50.563,00	1.902.944	-20.371
1.892.272	22,0000	1.887.158	1.903.899	50.565,00	1.903.892	-16.734
1.798.647	16,0000	1.797.853	1.803.537	50.570,00	1.803.617	-5.764
1.802.335	21,2345	1.801.582	1.815.855	50.564,00	1.815.951	-14.369
1.778.080	19,0000	1.776.069	1.798.264	50.559,00	1.798.244	-22.175
1.739.403	17,0000	1.734.210	1.760.068	50.562,00	1.759.990	-25.780
1.827.506	21,0000	1.825.802	1.849.052	50.563,00	1.849.045	-23.243

Figura 11.13: Resultados de simulación para instancia 729 escenarios

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	21,5274	1.800.529	1.825.308	50.567,37	1.825.361	-24.832
1.832.234	20,0000	1.830.505	1.851.156	50.565,00	1.851.122	-20.617
1.906.182	21,0000	1.905.614	1.911.370	50.574,00	1.911.405	-5.791
1.818.492	20,0000	1.816.688	1.831.546	50.570,00	1.831.635	-14.947
1.802.335	17,0000	1.800.538	1.825.308	50.577,00	1.825.392	-24.854
1.842.896	21,0000	1.841.063	1.842.836	50.577,00	1.842.868	-1.805
1.721.094	22,2132	1.720.482	1.742.861	50.577,00	1.742.894	-22.412
1.827.896	16,0000	1.827.037	1.836.535	50.578,00	1.836.508	-9.471
1.802.335	21,0000	1.801.369	1.825.308	50.561,00	1.825.355	-23.986
1.809.956	18,0000	1.809.613	1.834.276	50.558,00	1.834.189	-24.576
1.744.442	18,0000	1.743.848	1.749.492	50.562,00	1.749.499	-5.651
1.832.198	23,0000	1.831.516	1.835.529	50.564,00	1.835.615	-4.099
1.802.335	21,0000	1.801.608	1.825.308	50.574,00	1.825.343	-23.735
1.868.937	18,0000	1.866.948	1.883.059	50.575,00	1.883.099	-16.151
1.751.994	22,2132	1.750.485	1.777.668	50.567,00	1.777.712	-27.227
1.895.192	17,0000	1.893.566	1.897.740	50.561,00	1.897.732	-4.166
1.802.335	18,0000	1.800.660	1.825.308	50.574,00	1.825.306	-24.646
1.873.609	22,0000	1.872.641	1.897.081	50.577,00	1.897.038	-24.397
1.816.245	22,2132	1.814.466	1.819.638	50.569,00	1.819.574	-5.108
1.755.441	19,0000	1.754.118	1.772.760	50.564,00	1.772.688	-18.570
1.802.335	17,0000	1.800.962	1.825.308	50.577,00	1.825.340	-24.378
1.899.276	23,0000	1.898.293	1.918.679	50.570,00	1.918.579	-20.286
1.796.822	22,2132	1.796.080	1.816.261	50.559,00	1.816.161	-20.081
1.866.660	21,0000	1.864.741	1.874.293	50.569,00	1.874.357	-9.616
1.802.335	18,0000	1.801.006	1.825.308	50.560,00	1.825.261	-24.255
1.841.212	21,0000	1.840.850	1.862.923	50.568,00	1.862.893	-22.043
1.782.219	22,2132	1.781.452	1.782.989	50.578,00	1.783.016	-1.564
1.906.679	17,0000	1.904.708	1.920.290	50.577,00	1.920.318	-15.610
1.802.335	17,0000	1.802.101	1.825.308	50.575,00	1.825.213	-23.112
1.853.116	21,0000	1.852.691	1.870.957	50.569,00	1.870.999	-18.308
1.756.947	22,2132	1.755.058	1.765.783	50.560,00	1.765.855	-10.797
1.826.349	16,0000	1.825.654	1.846.548	50.566,00	1.846.586	-20.932
1.802.335	21,0000	1.800.376	1.825.308	50.558,00	1.825.265	-24.889
1.842.163	18,0000	1.840.636	1.855.290	50.569,00	1.855.366	-14.730
1.772.089	22,2132	1.771.289	1.791.785	50.578,00	1.791.718	-20.429
1.729.799	21,0000	1.729.565	1.749.701	50.567,00	1.749.781	-20.216
1.802.335	21,0000	1.800.378	1.825.308	50.566,00	1.825.354	-24.976
1.877.492	18,0000	1.876.704	1.900.470	50.571,00	1.900.446	-23.742
1.752.010	17,0000	1.750.967	1.772.112	50.563,00	1.772.092	-21.125
1.877.570	18,0000	1.876.158	1.878.181	50.557,00	1.878.260	-2.102

N-Expected Model			Stochastic Model			Comparison
1st Stage	Solving time [s]	Ev. Solution	1st Stage	Solving time [s]	Ev. Solution	Difference
1.802.335	22,2132	1.801.939	1.825.308	50.570,00	1.825.379	-23.440
1.758.996	19,0000	1.757.277	1.775.252	50.577,00	1.775.259	-17.982
1.881.357	23,0000	1.880.151	1.882.426	50.575,00	1.882.501	-2.350
1.827.108	16,0000	1.826.137	1.848.988	50.565,00	1.849.079	-22.942
1.802.335	21,2354	1.800.958	1.825.308	50.570,00	1.825.313	-24.355
1.894.822	16,0000	1.894.552	1.900.452	50.559,00	1.900.382	-5.830
1.757.265	23,0000	1.755.396	1.760.001	50.572,00	1.759.907	-4.511
1.877.704	20,0000	1.877.107	1.887.433	50.571,00	1.887.456	-10.349
1.802.335	23,5677	1.801.198	1.825.308	50.566,00	1.825.232	-24.034
1.883.315	22,0000	1.881.621	1.899.201	50.568,00	1.899.219	-17.598
1.799.728	19,0000	1.798.980	1.808.095	50.570,00	1.808.184	-9.204
1.796.136	22,0000	1.795.554	1.806.743	50.565,00	1.806.715	-11.161
1.802.335	21,2345	1.801.329	1.825.308	50.561,00	1.825.215	-23.886
1.729.990	17,0000	1.728.018	1.732.463	50.558,00	1.732.511	-4.493
1.770.727	15,0000	1.769.079	1.791.455	50.569,00	1.791.474	-22.395
1.769.711	15,0000	1.768.219	1.791.080	50.569,00	1.791.011	-22.792
1.802.335	22,2132	1.801.506	1.825.308	50.572,00	1.825.327	-23.821
1.875.485	17,0000	1.873.971	1.895.344	50.571,00	1.895.324	-21.353
1.781.604	23,0000	1.780.044	1.787.623	50.574,00	1.787.541	-7.497
1.741.071	17,0000	1.739.753	1.744.724	50.570,00	1.744.792	-5.039
1.802.335	16,4568	1.800.468	1.825.308	50.578,00	1.825.345	-24.877
1.810.903	17,0000	1.809.231	1.826.726	50.557,00	1.826.812	-17.581
1.808.235	20,0000	1.807.291	1.831.751	50.558,00	1.831.697	-24.406
1.736.889	22,0000	1.735.399	1.746.565	50.560,00	1.746.661	-11.262
1.802.335	18,4567	1.801.956	1.825.308	50.577,00	1.825.216	-23.260
1.803.506	20,0000	1.803.182	1.806.654	50.571,00	1.806.650	-3.468
1.876.999	20,0000	1.875.116	1.894.775	50.558,00	1.894.729	-19.613
1.900.452	20,0000	1.899.885	1.902.526	50.562,00	1.902.445	-2.560
1.802.335	19,4564	1.800.985	1.825.308	50.578,00	1.825.276	-24.291
1.877.229	19,0000	1.876.353	1.883.707	50.561,00	1.883.710	-7.357
1.888.259	18,0000	1.886.578	1.901.555	50.578,00	1.901.552	-14.974
1.832.430	17,0000	1.831.296	1.855.822	50.567,00	1.855.796	-24.500
1.802.335	22,2132	1.801.472	1.825.308	50.563,00	1.825.330	-23.858
1.812.828	20,0000	1.810.918	1.817.855	50.563,00	1.817.764	-6.846
1.854.893	22,0000	1.853.769	1.875.446	50.565,00	1.875.516	-21.747
1.828.691	16,0000	1.827.210	1.833.972	50.570,00	1.833.967	-6.757
1.802.335	21,2345	1.801.650	1.825.308	50.564,00	1.825.389	-23.739
1.844.356	19,0000	1.842.504	1.861.338	50.559,00	1.861.438	-18.934
1.794.485	17,0000	1.792.741	1.797.775	50.562,00	1.797.726	-4.985
1.787.572	21,0000	1.786.023	1.798.299	50.563,00	1.798.313	-12.290

Figura 11.14 Resultados de simulación para instancia 1000 escenarios

## 11.5 Resultados Formulación extendida, PH serial y paralelo

N° de Escenarios	Tiempo de Inicialización [min]		
	Extendido	PH serial	PH paralelo
1	0,808	0,7955	0,809666667
2	1,610333333	1,603166667	1,616333333
3	2,4945	2,354666667	2,368333333
5	4,002166667	3,9435	3,994833333
10	7,864833333	7,832	7,932333333
20	15,78766667	15,8075	16,0015
30	23,86066667	23,6745	23,69266667
40	31,737	31,42016667	31,9815
50	39,30916667	39,15316667	39,88233333
107	84,5685	84,03683333	84,5315
110	86,61266667	86,64033333	87,67066667
216	180,264	169,1431667	169,7435
343	OUT OF MEMORY	403,3333333	369,1666667
512	OUT OF MEMORY	2050,5697	2066,116667
729	OUT OF MEMORY	3501,264	3524,23
1000	OUT OF MEMORY	4895,24	4997,42

Figura 11.15: Tiempos de inicialización para instancias experimentales según método de solución

## 11.6 Estrategias para el cálculo de $\rho$

Existen una serie de estrategias utilizadas para llevar a cabo el cálculo del parámetro  $\rho$  dentro de las iteraciones del algoritmo PH. Las tres más utilizadas son:

### 11.6.1 FIX

Consiste en la utilización de un valor fijo establecido para el parámetro  $\rho$ , el cuál se mantiene constante durante todas las iteraciones del algoritmo PH. Se selecciona un valor positivo que oscila entre 0.0001 y 10000 en base a resultados experimentales previos, obteniéndose un mejor rendimiento para cada valor según el tamaño de la instancia y el tipo de problema en estudio.

### 11.6.2 CP - OFP

Realizando un símil a la estimación del costo marginal de un producto, éste se reemplaza por una fracción del ponderador que está asociado a la variable que representa dicho producto. 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 éste se aleja de las condiciones de no anticipatividad “óptimas”.

El factor  $\rho$  se calcula según:

$$\rho(i) = k * C_i$$

expresión que indica que el el parámetro depende del costo asociado a cada variable, así como de un factor que normalmente oscila entre 0.1 y 0.01.

Mismo razonamiento (Costo proporcional, CP) fue aplicado posteriormente en problemas de maximización, donde no es evidente identificar el costo marginal por producto, denominándose en este caso “Proporcional a la Función Objetivo” (OFP en inglés). En este caso en particular, se observa la existencia de variables con costos muy distintos en términos de magnitud, lo que conlleva a que las penalizaciones por desviarse del promedio deban ser – en teoría – proporcionalmente distintas.

Así, se deben identificar los costos proporcionales para cada una de las variables modeladas dentro del problema.

### 11.6.3 SEP

La estrategia de selección por elemento (Selecting per-element en inglés) implica la selección de un valor de  $\rho$  distinto para cada uno de los escenarios. Surge de la idea de forzar al penalizador  $w$  a alcanzar la convergencia en una única iteración. Para lograrlo, una forma sería que el parámetro cumpla con la expresión  $\rho = \frac{c_i}{|x_s - \bar{x}|}$ .

Una aproximación más conservadora consiste en tomar 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:

$$\rho(i) = \frac{c_i}{\max_{s \in N_t} x_s - \min_{s \in N_t} x_s + 1}$$

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

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