Models and Algorithms for Deterministic and Stochastic Optimization Problems

by

Pablo Olaso Redondo

Supervisor:

Celeste Pizarro Romero

October, 2015

Departamento de Matemática Aplicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica

Universidad Rey Juan Carlos

Doña Celeste Pizarro Romero, Profesora Contratada Doctor de la Universidad Rey Juan Carlos del Departamento de Matemática Aplicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica AUTORIZA:

La presentación de la Tesis Doctoral titulada ***Models and Algorithms for Deterministic and Stochastic Optimization Problems*** realizada por Don Pablo Olaso Redondo bajo mi inmediata dirección y supervisión en el Departamento de Matemática Aplicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica y que presenta para la obtención del grado de Doctor por la Universidad Rey Juan Carlos.

En Móstoles, a 24 de Octubre de 2015

Celeste Pizarro Romero

A Ariadne, a Nahia, a Julieta, mis 3 cachorritas

A Karina, mi Amiga, mi Compañera, mi Amor...

Agradecimientos

"Tan chiquí, tan chiquita que es la tierra
si la mi, si la miran desde el sol.
Tan chiquí, tan chiquita que es la infancia,
cuando vi, cuando vino se escapó.
...
Tan chiquí, tan chiquita que es la niña,
tan chiquí, tan chiquita y ya creció."

La canción de Trilce, de Daniel Viglietti, inspirado en la poesía de César Vallejo

Tras más de 3 años aparcada en el desván, pese a estar *casi* terminada, cuando muchos ya la daban por perdida, quizá no sin razón, a pesar de sólo faltarle un *último empujón*, esta tesis por fin ve la luz, y yo con ella. Esto no habría sucedido sin el apoyo incondicional de mi familia. A ellos lo agradezco con toda el alma... Gracias Karina, gracias aita y ama por haberme ayudado a levantarme cuando me parecía imposible. Gracias cachorritas, por crecer tan encantadoras, vosotras que sois mi proyecto más importante, y del que más orgulloso estoy... Con vosotras nunca me dí y nunca me daré por vencido, os lo prometo.

También quiero agradecer a mis amigos, y a mis compañeros de la Rey Juan Carlos y de la Complutense, por todo el apoyo y los ánimos que he recibido siempre.

Por último, gracias Celeste, por no haber renunciado, por haberme acompañado hasta el final.

Contents

Resumen **xiii**

Preface **xxv**

1 Conflict Avoidance: State of the art **1**

1.1 Conflict Avoidance Introduction 1

1.2 Conflict Detection 3

1.3 Conflict Resolution 5

1.3.1 Prescribed 5

1.3.2 Optimization 6

1.3.3 Force field 10

1.3.4 Manual 10

1.3.5 Neural Networks and metaheuristics 11

1.3.6 Others 12

1.4 Similar problems and Review table 13

1.5 Problem description 15

2 Conflict Avoidance: 0-1 linear models for Conflict Detection & Resolution **17**

2.1 Problem description 19

2.2 Collision Avoidance via altitude level changes 20

2.2.1 Conflict Detection 20
2.2.2 Model formulation for conflict resolution 23
2.3 Collision Avoidance via altitude level and speed changes 24
2.3.1 Definitions 24
2.3.2 Conflict Detection 25
2.3.3 Model formulation for conflict resolution 26
2.3.4 Tightening the model 28
2.4 Computational experience 30
2.5 Further discussions and extensions for the proposed models 31
2.5.1 On ascending or descending flight levels 31
2.5.2 Rerouting 38
2.5.3 Arriving at a different time 39
2.5.4 Changing flight level might imply changing speed 40
2.6 Conclusions 41

3 Introduction to Stochastic Programming and Parallel Computing: State of the art 43

3.1 Introduction 44
3.2 Stochastic linear modeling 47
3.2.1 Deterministic linear models 47
3.2.2 Decisions and stages 48
3.2.3 two-stage models 50
3.2.4 Multistage models 51
3.3 Modeling via scenario tree 54
3.3.1 Scenario tree 54
3.3.2 Non-anticipativity principle 57
3.4 Deterministic Equivalent Model 57

3.4.1 Compact representation 59
3.4.2 Representation with splitting variables by scenario groups 60
3.4.3 Representation with splitting variable by scenarios 63
3.5 The value of information in stochastic models 63
3.5.1 The Expected Value of Perfect Information 65
3.5.2 The Value of the Stochastic Solution 65
3.6 Algorithms in Stochastic Linear Programming 67
3.6.1 L-Shaped method for two-stage problem 67
3.6.2 Lagrangean decomposition 72
3.6.3 Obtaining Lagrangean multipliers: Subgradient method 73
3.6.4 Augmented Lagrangean Decomposition 74
3.7 Stochastic Integer Programming 75
3.8 Algorithms in Stochastic Integer Programming 78
3.8.1 L-Shaped method for integer problems 78
3.8.2 Integer simple recourse 81
3.8.3 Stochastic Branch-and-Bound 82
3.9 BFC. Definitions and algorithmic framework for pure 0-1 problems 85
3.10 Applications of Stochastic Programming 90
3.10.1 Transportation 90
3.10.2 Telecommunication 91
3.10.3 Electricity Power Generation 92
3.10.4 Finance 92
3.10.5 Manufacturing 93
3.10.6 Other Applications 94
3.11 Parallel Computing 95
3.11.1 A brief introduction to Parallel Computing 95

3.11.2 Message Passing Interface 96
3.11.3 Speedup 97

4 Metaheuristic algorithms for solving large-scale multistage stochastic mixed 0-1 problems 99

4.1 Introduction. Fix-and-Relax Coordination 99
4.1.1 On the *Fix-and-Relax* scheme 100
4.1.2 Fix-and-Relax model 100
4.1.3 Branching strategy 101
4.1.4 Associated models to scenario group r 101
4.1.5 Two-stage BFC algorithm 103
4.1.6 FRC algorithm implementation 104
4.2 Break stage scenario clustering 106
4.2.1 Associated models to FR level r 108
4.3 Jumping Fix-and-Relax Coordination (FRC-J) 110
4.4 Parallel Jumping Fix-and-Relax Coordination (FRC-PJ) 113
4.4.1 Speedup 114
4.4.2 FRC-PJ 115

5 Computational Experience 117

5.1 Multi-period location-allocation problem under uncertainty 117
5.2 Multistage Stochastic Facility Location Problem 118
5.2.1 0-1 DEM 120
5.3 Computational experience 123
5.3.1 Testbeds description 123
5.3.2 Numerical results 125

6 Conclusions, contributions and future research 133

6.1 Conflict Avoidance in the Air Traffic Flow problem CA 134
6.2 Stochastic Programming 135
6.3 Future work 136

Glossary **140**

Bibliography **142**

List of Tables

1.1 Review table 14

1.2 Review table 15

2.1 Dimensions of the altitude level change problem for model 2.1-2.7 32

2.2 Dimensions of the altitude level and speed changes problem for model 2.8-2.20 33

2.3 Dimensions of the pure 0-1 model 2.1-2.7 34

2.4 Dimensions of the mixed 0-1 model 2.8-2.20 35

2.5 Computational results for the pure 0–1 model 2.1-2.7 36

2.6 Computational results for the mixed 0-1 model 2.8-2.20 37

3.1 six MPI functions 97

5.1 Testbed. Problem and model dimensions. 125

5.2 Computational results. CPLEX vs FRC 127

5.3 Computational results. FRC vs FRC-J 129

5.4 Computational results. FRC-J vs FRC-PJ 131

5.5 Computational results. FRC-SLOC vs FRC-PJ 132

List of Figures

2.1 Illustrative case for three flight routes 22

2.2 Illustrative case for different altitude levels of the routes of the aircraft 1 and 3 22

3.1 Traveling salesperson example 49

3.2 Scenario tree 55

3.3 Non anticipativity principle 58

3.4 Scenario tree 59

3.5 Compact representation 60

3.6 Representation with splitting variables by groups of scenarios 62

3.7 Splitting variables representation by scenarios 64

3.8 Matrix structure for the two stage compact representation 68

3.9 Two stage Benders decomposition 69

3.10 Objective function of a small Stochastic Integer problem 78

3.11 Branch-and-Fix Coordination (BFC) scheme 88

Legend of notation

The criteria used in this thesis for notation for different elements are the following:

Sets.- Capital and calligraphic letters, (for example $\mathcal{F}$, $\mathcal{X}$).

Scalars.- Italic letters, so much capital as small ones, (for example, A, a, X, x).

Vectors.- Small bold letters (for example, $\mathbf{x}$). Its components are denoted by scalars, x_i, whereas different subarrays are denoted by $\mathbf{x}_i$. All vectors, save for otherwise stated, are column vectors; a row vector is denoted by $\mathbf{x}^T$. Particulary, **0** and **1** are vectors with all its components equal to 0 and 1, respectively.

Matrices.- Bold capital letters,(for example, **M**). Same notation is used for its arrays, columns or submatrices, $\mathbf{M}_i$. Its elements are denoted as the scalars, M_{ij}.

Given a matrix **M**, its transposed matrix is $\mathbf{M}^T$. The absolute value of a number x, is denoted by $|x|$. Given a vector $\mathbf{x}$, $|\mathbf{x}|$ denotes its vector dimension, whereas in a set case, $\mathcal{F}$, $|\mathcal{F}|$ refers to its cardinal.

Resumen

La optimización (también llamada programación matemática (PM)) es la rama de las matemáticas que trata sobre encontrar aquella solución que proporcione el mayor beneficio para un problema dado, dicho de otro modo, trata de buscar, de entre todas las posibles soluciones a un problema, aquella que minimice una función dada (o equivalentemente, la maximice, nótese que $\max\{f(x)\}$=$\min\{-f(x)\}$), generalmente se trata de una función real ($f : \mathbb{R}^n \to \mathbb{R}$) y se demomina *función objetivo*. El conjunto de soluciones factibles vendrá definido mediante ecuaciones matemáticas, llamadas restricciones, que las soluciones deben cumplir.

Así pues, dado un problema cualquiera, se debe realizar un modelo matemático consistente en una serie de restricciones y una función objetivo a minimizar, para después resolverlo mediante alguno de los algoritmos proporcionados por el estado del arte. El modelado de un problema dado es de hecho un arte en sí mismo. Se trata de abstraer aquellos aspectos innecesarios, superfluos, y al mismo tiempo representar la realidad lo más fielmente posible, y ello teniendo en cuenta que, dependiendo del enfoque elegido, la resolución del modelo puede no ser viable en la práctica con los recursos computacionales de que dispone la humanidad en su actual estadio de desarrollo. Aspectos que pueden afectar drásticamente a la facilidad de resolución del modelo son:

- El tipo de modelo (fundamentalmente, si es lineal o no-lineal)
- Las variables que entran en juego (el número de variables, si son enteras, continuas o binarias)
- La elección de las restricciones adecuadas. En el trabajo de modelado puede jugar un papel fundamental la búsqueda de nuevas restricciones que permitan hacer el modelo más *robusto* desde el punto de vista matemático, o dicho de otro modo que cumpla ciertas condiciones que permitan a los algoritmos encontrar la solución más fácilmente.

Dentro de la optimización matemática, en esta tesis vamos a transitar por dos áreas que ocupan

un lugar destacado, la Programación Lineal y la Programación Estocástica.

La Programación Lineal trata de aquellos modelos cuyas restricciones y función objetivo son lineales. En general los modelos lineales se pueden resolver en menos tiempo que los no-lineales, especialmente si no tienen variables enteras, o tienen pocas, y cumplen ciertas condiciones.

En esta tesis se aplica la programación lineal a los problemas de elusión de conflictos en el tráfico aéreo, mediante un enfoque distinto al habitual. El modo de tratar estos problemas hasta ahora se basaba principalmente en modelos no-lineales, lo que debido a las limitaciones computacionales mencionadas arriba, no permitía enfrentarse a casos en los que entren en juego muchos aviones o considerar un espacio aéreo amplio (generalmente los modelos tratan 2 o 3 aviones en un espacio limitado). El nuevo enfoque aplicado en esta tesis, en cambio, permite aplicar la programación lineal, lo cual a su vez facilita considerar el plan de vuelo de todos los aviones presentes en un espacio aéreo lo suficientemente amplio, y resolver los posibles conflictos aplicando cambios de velocidad o de altura, e incluso cambiando a rutas alternativas si ello fuera posible.

Por otro lado nos adentramos en el área de la Programación Estocástica. En muchos problemas reales la incertidumbre juega un papel importante y que por tanto debería tenerse en cuenta en el modelo resultante. Sin embargo la incertidumbre no se deja atrapar tan fácilmente, y el como modelarla es aun un problema que dista de estar cerrado, si bien se ha avanzado mucho y existe un enfoque ampliamente aceptado y para el que se han podido desarrollar varios algoritmos que explotan eficientemente sus características particulares.

Antecedentes

Elusión de conflictos en el tráfico aéreo

La detección y resolución de conflictos en el tráfico aéreo, o Air Conflict Avoidance, o simplemente Conflict Avoidance (CA), como se suele denominar en la literatura especializada, consiste básicamente en evitar que los aviones se aproximen demasiado entre sí.

El CA copa actualmente el interés de las compañías aéreas, pues el creciente tráfico aéreo suscita la necesidad de encontrar soluciones automatizadas que simplifiquen el trabajo de los controladores aéreos. Evidentemente, el factor humano juega y jugará por muchos años un papel esencial por la habilidad, difícilmente reproducible por las máquinas de hoy en día, de integrar información, analizarla y tomar las decisiones oportunas. No obstante, el fallo humano también es posible y es por

ello que los sistemas autómatas se usan desde hace tiempo en las cabinas de los aviones y en las torres de control aéreo, tanto en sistemas de alerta de conflictos como de asistencia a la toma de decisiones, proponiendo en su caso soluciones operativas.

Con el incremento de la demanda en el transporte aéreo, se hace urgente la necesidad de automatizar aún más la detección y resolución de conflictos. Se ha estudiado mucho el problema de CA y se han propuesto diversos métodos en la literatura especializada, en su mayor parte desde el punto de vista de la optimización matemática, si bien no únicamente. Como veremos, también se ha abordado el problema desde distintos enfoques como el soft-computing (mediante el uso de redes neuronales, algoritmos genéticos y otras meta-heurísticas).

Comencemos definiendo el *conflicto* de un modo preciso. Asi, podría definirse como aquella situación en que la distancia entre dos o más aviones viola un criterio de separación establecido. La distancia mínima establecida suele ser de 5 millas náuticas horizontalmente o al menos 1000 pies de separación vertical. De este modo queda definido un volumen, la *zona protegida*, alrededor de cada avión, que debe ser respetado por cualquier otro vehículo en todo instante de tiempo. Podríamos definir dicho espacio de un modo más restrictivo (e.g. una esfera de 500 pies de diámetro) o con otros parámetros, por ejemplo, el tiempo necesario para recorrerlo (de este modo tendríamos en cuenta la relación entre distancia y velocidad relativa de los vehículos).

Un planteamiento típico es presentar un modelo dinámico que intente predecir las posiciones de los aviones en el futuro a partir de su situación actual y así decidir si habrá un conflicto. Sin embargo hay que tener en cuenta que cierta incertidumbre es inevitable y dicho modelo debería considerarla a riesgo de no ser del todo fiable su predicción. Por otro lado, como mencionábamos antes, un modelo de este tipo será típicamente no-lineal y deberá restringirse a unos pocos vehículos a fin de ser computable en un tiempo razonable. Otra opción pasa por basarse en información previa tal como un plan de vuelo. Debemos mencionar que existen modelos para calcular las trayectorias óptimas de los aviones en vuelo, y funcionan con bastante eficiencia, sin embargo, aquí la dificultad estriba en que hay que tratar las trayectorias de más de un avión y además el modelo debe evitar conflictos entre ellos, esto sería relativamente fácil de modelar pero el resultado sería irresoluble computacionalmente.

Este problema se ha estudiado ampliamente y desde muchos enfoques, tal como se puede leer en el primer capítulo de esta tesis, que contiene un resumen de gran parte de la bibliografía existente al respecto. Como decíamos al principio, la optimización matemática ha sido la herramienta más utilizada en la resolución de este problema, y esta ha sido también la elección en este trabajo, en particular la programación lineal.

Programación Estocástica

La *optimización* es una herramienta matemática fundamental en la toma de decisiones hoy en día, y se aplica en la más variada gama de problemas, que van desde las finanzas hasta la elusión de conflictos, pasando por los muchos y muy variados problemas de las grandes organizaciones, de planificación, producción y distribución. En muchos casos es posible modelar los problemas a los que nos enfrentamos mediante modelos deterministas, sin embargo, en muchos otros problemas la incertidumbre juega un papel clave que no es posible soslayar. Para muchos problemas de planificación o finanzas, sin ir más lejos, un modelo que no tuviera en cuenta la incertidumbre inherente al mundo financiero podría proporcionar decisiones absolutamente erradas.

La Programación Estocástica (SP, por sus siglas en inglés) surgió para solventar este problema. Si bien ya desde 1955 se viene tratando la incertidumbre en Programación Matemática, su desarrollo no tuvo lugar hasta que los avances en computación lo permitieron, en los años 80.

Una primera aproximación a la SP consistió en obtener los promedios de los parámetros estocásticos y resolver el modelo resultante como si de uno determinista se tratase. Sin embargo, dicho problema *promedio* no tiene necesariamente relación con el escenario que realmente vaya a suceder, es más, la solución obtenida podría ni siquiera ser factible para muchos de los posibles escenarios. Otras aproximaciones se han abordado desde entonces, pero la más extendida en la actualidad es la de modelar mediante *árboles de escenarios*. En ella, los parámetros indeterminados son representados mediante variables aleatorias, y un escenario consiste en una realización concreta de dichas variables. Estas variables suelen ser continuas y esta técnica consistiría en reducirlas, mediante aproximación, a variables discretas con un número limitado de posibles valores, calculando las probabilidades de cada uno de ellos. Surgen aquí numerosos problemas como el de determinar cuál es el número adecuado, lo suficientemente representativo, de escenarios a considerar. A mayor número de escenarios obtendremos una mejor representación de la realidad, pero por otro lado las dimensiones del modelo se dispararán con rapidez alcanzando tamaños inmanejables en la práctica.

Los escenarios se suelen representar mediante un árbol cuyo nodo raíz representa la primera etapa, en la cual se tomarán algunas decisiones. Una vez que se ha tomado una decisión, se dispondrá de cierta información nueva antes de tomar la siguiente decisión, las posibles realizaciones de los parámetros indeterminados en la siguiente etapa se representan mediante sendas ramas que llevan a nuevos nodos en los que nuevas decisiones se han de tomar. Y así hasta los nodos terminales, de modo que el camino seguido desde la raíz hasta cada uno de dichos nodos terminales representará un escenario y se corresponderá con una realización concreta de todos los parámetros indeterminados

del problema.

Por otro lado, al darse distintos posibles escenarios, el concepto de solución óptima pierde sentido, pues las decisiones óptimas para un escenario no tienen por qué serlo para los demás. La estrategia pasa por tanto por buscar soluciones que sean factibles para todos los escenarios sin quedar supeditada a ninguno en concreto. La estrategia clásica consiste en minimizar la esperanza de la función objetivo considerando todos los escenarios.

A pesar de que esta metodología se aplica fundamentalmente a problemas de programación lineal, el tamaño de los modelos alcanza con facilidad tamaños gigantescos, con cientos de miles de variables, e incluso millones. Es más, en muchos casos las variables son enteras o binarias, lo cual hace que estos problemas sean difíciles de resolver. Es por esto que en la literatura se han propuesto una serie de algoritmos que buscan resolver los problemas de Programación Estocástica Entera, como por ejemplo los métodos L-Shaped, Branch-and-Bound estocástico, Branch-and-Fix-Coordinado, y diversas técnicas heurísticas como el algoritmo Fix-And-Relax Coordinado.

Objetivos

En lo que a la elusión de conflictos se refiere, el objetivo planteado es el de crear un modelo matemático que permita resolver el problema de CA de un modo flexible y eficiente, y aplicable a casos con un número considerable de aviones y en un espacio aéreo amplio.

En cuanto a los problemas de Programación Estocástica, el objetivo es desarrollar una aproximación algorítmica capaz de resolver problemas generales de Programación Estocástica multietapa 0-1 Mixta de un modo eficiente, y de este modo permitir resolver casos suficientemente grandes que de otro modo serían dificilmente manejables.

Por último, y para ambas propuesas, realizar una experiencia computacional que permita conocer su calidad.

Metodología

Para la consecución de los objetivos mencionados se ha desarrollado el siguiente plan de trabajo:

Elusión de conflictos en el tráfico aéreo

Se ha realizado en primer lugar una revisión exhaustiva de la literatura existente en torno al problema planteado. De dicho análisis se concluye que la mayoría de los enfoques adoptados hasta ahora adolecen de una alta complejidad en cuanto a su resolución, por lo cuál, aun cuando casi siempre se limitan a una o a lo sumo dos de las tres posibles maniobras de elusión de conflicto (v.g. cambio de velocidad, de altura o giro), no son aplicables a casos en los que se encuentre un número considerable de aviones y un espacio aéreo de amplitud proporcional a cualquiera de los sectores de control en que se divide el espacio aéreo.

Se propone por tanto un nuevo enfoque, basado en el plan de vuelo, que permita modelar el problema mediante programación lineal y sea capaz de detectar y proponer maniobras de elusión de conflictos basadas en cambios de velocidad y altura. El espíritu de este nuevo enfoque está inspirado en la idea de separar tareas: Ya que existen herramientas para calcular las trayectorias de vuelo de un modo bastante realista y eficiente, la solución más sencilla pasaría por crear una nueva herramienta que, a partir de las rutas ya existentes, detecte y resuelva los conflictos. El ideal, de nuevo, sería poder calcular también de un modo preciso, los márgenes en que podría realizarse correctamente la dinámica de vuelo, además del óptimo, pues al dar prioridad a la elusión de conflictos, pierde importancia la eficiencia del vuelo (esto lo dejamos para trabajo futuro). No obstante, aún partiendo de los actuales planes de vuelo y considerando pequeñas alternativas (particularmente cambios de altura), es posible aun evitar los conflictos de un modo bastante coherente y en nuestra opinión no menos realista que las propuestas actualmente existentes en la literatura al respecto, incorporando además la ventaja del gran alcance de esta nueva propuesta. Todo lo anterior conduce a la presentación de dos modelos lineales, uno 0-1 puro y otro 0-1 mixto.

A continuación se desarrolla una experiencia computacional mediante la generación aleatoria de casos de diversa complejidad de resolución, en los que el horizonte espacial y temporal son considerablemente grandes y participan un número considerable de aviones.

Finalmente se analizan tanto la aplicabilidad de los modelos presentados como alguna posible mejora como puede ser la inclusión de rutas alternativas como maniobra de elusión de conflictos.

Programación Estocástica

En primer lugar, se ha partido de una propuesta algorítmica bastante reciente, llamada Fix-And-Relax Coordination (FRC), basada en una heurística, y que ha probado en numerosas ocasiones

su eficiencia y calidad. A partir de ahí, se han propuesto una serie de heurísticas para desarrollar un nuevo algoritmo (FRC-J) que permita resolver problemas grandes con mayor eficiencia. Además se ha desarrollado una versión basada en programación concurrente (FRC-PJ), mediante el uso de las librerías de paso de mensajes (MPI, por sus siglas en inglés), ampliamente utilizadas en el desarrollo de aplicaciones paralelas en entornos de memoria no compartida, fundamentalmente en clusters de computadores.

Por último, se ha realizado una experiencia computacional que permitiera comparar los resultados de los 3 algoritmos, el FRC, el FRC-J y el FRC-PJ, asi como con los tiempos y calidad de soluciones proporcionados por la herramienta CPLEX, que es el standard de facto en la programación lineal. Para ello se toma un problema particular de programación estocástica multietapa 0-1, y se recogen una serie de instancias de tamaño mediano a grande que ya habían sido testadas en la literatura.

Conclusiones

Las principales contribuciones de este trabajo son:

Elusión de conflictos en el tráfico aéreo

- Se propone un nuevo enfoque que permite modelar el problema mediante programación lineal, lo cual facilita encontrar soluciones en tiempos muy breves. Dicho enfoque, en vez de intentar modelar las trayectorias de los aviones basándose en las ecuaciones dinámicas de la física, trata de dividir tareas, y estudiar la elusión de conflictos a partir de un plan de vuelo dado. Dicho plan ya tiene en cuenta la física de vuelo y las preferencias de las aerolíneas para restringir las posibles soluciones. Es suficiente con considerar un (en principio, pequeño) margen de maniobra para acelerar o cambiar de altura. De este modo es posible crear modelos que detecten los posibles conflictos y propongan, si es necesario, las correspondientes maniobras de elusión.
- Se presentan dos modelos lineales, el primero de ellos es 0-1 puro (sin variables continuas) y permite eludir conflictos mediante cambios de altura, mientras que el segundo es 0-1 mixto (es decir, integra tanto variables binarias como continuas) y recurre a cambios de velocidad y de altura, pudiendo dar prioridad a uno u otro en función de las preferencias del usuario. Dichos modelos son lo suficientemente robustos como para poder ser resueltos en un periodo realmente corto de tiempo incluso para casos de gran tamaño, lo cual los hace especialmente útiles para

ser usados en tiempo real.

- La propuesta es flexible, permite que las trayectorias no sean lineales. El plan de vuelo en que se basa puede haber sido calculado de diversos modos: a partir de la extrapolación del vector de velocidad en un momento dado, de los puntos de baliza de los actuales planes de vuelo o bien de planes de vuelo libres basados en trayectorias óptimas calculadas a priori. Por último, no se exige que la velocidad se mantenga constante durante el horizonte temporal como es el caso en parte de la literatura.

- Debido a los tiempos de resolución realmente pequeños que exigen ambos modelos, la propuesta está especialmente indicada para ser usada en el largo plazo y en regiones aéreas amplias, que puedan implicar varios de los sectores de control de tráfico en que se divide el espacio aéreo.

- Finalmente, los modelos son fácilmente escalables y pueden ser ampliados y mejorados con nuevas restricciones. Si bien se bosquejan mejoras a ambos modelos para ciertos casos, como por ejemplo la inclusión de rutas alternativas como maniobra elusiva, se deja como propuesta para trabajo futuro.

De acuerdo con la experiencia computacional presentada, se puede observar que el primer modelo es particularmente robusto y eficiente, por lo que puede aplicarse a problemas aún mayores que el segundo modelo, tanto en cantidad de aviones involucrados como espacio aéreo u horizonte temporal. En cualquier caso, el segundo modelo también muestra buenos resultados, sin embargo, cabe plantear que en muchos casos puede ser suficiente con una propuesta limitada a los cambios de altura como maniobra elusiva, dado que en la mayoría de las situaciones reales, los conflictos son pocos y basta con una ínfima cantidad de maniobras para solucionar los potenciales conflictos. Por otro lado, de acuerdo con parte de la literatura, los cambios de velocidad no se consideran maniobras particularmente eficientes, lo cual es comprensible pues pueden implicar mayor consumo de combustible y por otro lado implican mayor incertidumbre (por la variabilidad de la velocidad del viento, entre otros factores). En resumen, ambos modelos son muy eficientes y aplicables en casos reales, pudiendo predecir y resolver los conflictos para uno o varios sectores de control a la vez con varias horas de antelación.

Como **trabajo futuro** se propone desarrollar las mejoras bosquejadas en esta tesis, buscando el modo de mantener o mejorar la eficiencia computacional de los modelos actuales. También se plantea el desarrollo de una capa intermedia que permita desarrollar los parámetros utilizados por los modelos del presente trabajo a partir de los datos que manejan en la práctica las compañías aéreas y

los centros de control de tráfico aéreo, y en particular que permita obtener tanto las trayectorias como los márgenes de actuación óptimos para la elusión de conflictos.

Programación Estocástica

Se proponen dos nuevos algoritmos, uno de ellos secuencial y el otro paralelo, basados ambos en heurísticas que han probado su efectividad previamente y nuevas heurísticas propuestas por primera vez en esta ocasión. Si bien ambos algoritmos no proporcionan el óptimo al tratarse de heurísticos, se comprueba en la experiencia computacional la calidad de las soluciones obtenidas, llegando a mejorar a CPLEX en algunos casos.

Se toma un problema y una serie de casos de prueba de la literatura existente, el Multi-period location-allocation problem under uncertainty, un problema de decisión estratétiga y táctica cuyo objetivo es decidir la colocación óptima de instalaciones para abastecimiento a clientes. En dicho problema los parámetros aleatorios incluyen la demanda, el número mínimo de clientes en cada período, el número mínimo de instalaciones a abrir en cada período, el coste de apertura y mantenimiento de cada instalación, así como el coste de asignar un cliente a una instalación.

Se consideran sólo variables binarias y los modelos son de enormes dimensiones, con hasta medio millón de variables. Son por tanto problemas difíciles que sin embargo ambos nuevos algoritmos son capaces de resolver en tiempos considerablemente pequeños, especialmente comparados con los tiempos que requieren CPLEX y FRC, que es el algoritmo que se ha usado como referencia.

Cabe decir que los tiempos requeridos por el algoritmo paralelo, FRC-PJ, son particularmente buenos, demostrando que en este tipo de problemas cuyos algoritmos de resolución se basan en la descomposición, la computación paralela permite mejorar enormemente los tiempos de cálculo.

Por otro lado, téngase en cuenta que la ventaja de este tipo de algoritmos no es sólo de eficiencia en cuanto al tiempo de computación, sino también en cuanto a memoria, pues al descomponer el problema principal en varios subproblemas, los requisitos de memoria del sistema son menores.

Por último, se proponen las siguientes líneas de investigación y **trabajo futuro**:

- Desarrollo de nuevas mejoras en los algoritmos presentados recurriendo a otras técnicas conocidas como la *Descomposición Lagrangiana Aumentada*. Dicha descomposición permite obtener mejores cotas del óptimo de la función objetivo en cada nodo del árbol de ramificación, con las

cuales se puede mejorar el mecanismo de poda.

- Incorporar medidas de riesgo a las funciones objetivo. Hasta ahora, los modelos considerados presentan como función objetivo el valor esperado de la función objetivo sobre el conjunto de escenarios. Sin embargo, actualmente se están teniendo en cuenta medidas de riesgo considerando, por ejemplo, semidesviaciones y exceso de probabilidad. Estas aproximaciones son más convenientes bajo la presencia de las variables binarias que los clásicos esquemas de media-varianza.

- Ampliar la experiencia computacional, aplicando los algoritmos presentados en esta tesis y las mejoras que se introduzcan, a problemas aún más grandes y difíciles de resolver.

- Recurrir a otras técnicas de programación paralela que permitan mejorar aún más el algoritmo FRC-PJ, permitiendo dividir el problema en subproblemas a resolver por distintos procesos mediante la fijación de algunas variables a valores distintos para cada proceso.

- Investigar distintos modos de aplicar la heurística planteada en el capitulo 4.4 para evitar que los submodelos más difíciles sean resueltos muy a menudo. Dicha heurística consistiría en cada iteración, en fijar, al ramificar una nueva variable, al valor que obtuvo dicha variable para el submodelo más difícil de resolver, de este modo no será necesario resolverlo de nuevo.

- Desarrollo de una librería que permita resolver problemas generales multietapa 0-1 mixtos de grandes dimensiones, que permita recurrir indistintamente al algoritmo de preferencia según las circunstancias: FRC, FRC-J o FRC-PJ, pudiendo automatizar la selección de la etapa de corte, así como diversas estrategias como pueden ser la selección de la variable a ramificar, el valor a asignar a dicha variable, etc.

Difusión de los resultados

Los principales resultados de esta tesis han sido presentados en:

- XXXIII Congreso Nacional de Estadística e Investigación Operativa. *A parallel computing metaheuristic for solving multistage stochastic mixed integer programs*. Madrid, 2012.

- 12th International Conference on Stochastic Programming. *FRC: A heuristic extension of the BFC approach for solving very large scale multistage mixed 0-1 stochastic programs.* Halifax (Canadá), 2010.

- ECCO XXIII-CO2010 European Chapter on Combinatorial Optimization. *On The Conflict Avoidance for Air Traffic Flow Management Problem, two models.* Málaga, 2010.
- 24th European Conference on Operations Research. *FRC: A heuristic extension of the Branch-and-Fix Coordination approach for solving very large scale multistage mixed 0-1 stochastic problems.* Lisboa, 2010.
- 24th European Conference on Operations Research. *Conflict Avoidance for Air Traffic Flow Management Problem, pure and MIP models.* Lisboa, 2010.
- 4th workshop on Optimization and Variational Analysis. *On solving large-scale stochastic mixed 0-1 linear problems. Seminario invitado.* Elche, 2010.
- 23rd European Conference on Operational Research. *On the collision avoidance for air traffic management problem, a large scale mixed 0-1 program approach.* Sesión invitada. Bonn (Alemania), 2009.
- 23rd European Conference on Operational Research. *On the air traffic flow management problem. A stochastic integer programming approach. Sesión invitada.* Bonn (Alemania), 2009.
- Workshop CORAL 2009 (Conference on Routing and Logistics). *Air traffic flow management. Airports and sectors constraints, with en route problem.* Elche, 2009.

Además, parte del trabajo recogido en esta tesis ha sido publicado en [23, 24].

Preface

Presentation and motivation

Optimization is a very important mathematical tool for helping in decision making and is widely applied in many different areas. However, it is still in continuous development since the problems it deals with are hard to solve and in most cases they have to be highly simplified in order to allow our scarce computational resources finding their solution. This makes problem modeling a true art in which the objective is not only finding a good approximation to reality but avoiding complexity in order to get efficient (easily solvable) models.

In this thesis two important areas in optimization are explored, namely, Linear Programming (LP) and Stochastic Programming (SP).

Conflict Avoidance (CA) in the Air Traffic Flow

In particular LP is applied to the problem of Conflict Avoidance (CA) in the air traffic flow. The CA problem is attracting more and more interest due to the increasing demand of aerial traffic worldwide. That is why new and improved automatic systems for conflict detection and resolution are demanded. The CA problem has been widely studied as can be seen in Chapter 1 and many approaches have been explored, most of them using Mathematical Optimization, though other approaches have been explored, as soft-computing (neural networks, genetic algorithms, ant colony and other heuristics).

Stochastic Programming

Uncertainty is present in many real problems where many key parameters cannot be known a priori. For these problems, Stochastic Programming SP approaches the uncertainty by offering different models (chance-constrained, two-stage and multi-stage models, risk measures...). However, it is not until the 80's when SP is broadly applied, thanks to the advances in computing. SP models easily reach huge dimensions, with hundreds of thousands, even millions of variables, what makes of them hard problems, especially when dealing with integer variables.

So, several algorithms have been proposed in the literature approaching the resolution of Stochastic Integer Programming problems, such as L-Shaped methods, stochastic Branch-and-Bound, Branch-and-Fix-Coordination, and different heuristics as the Fix-And-Relax Coordination algorithm.

Objectives of the thesis

In this thesis a new approach to the CA is introduced, that allow solving the problem both flexible and efficiently. An approach applicable in situations involving a wide aerial space and an appreciable number of aircraft. In addition, new algorithmic frameworks are proposed to solve large-scale **multi-stage** Stochastic mixed 0-1 Programming problems.

Thesis Outline

The thesis is structured as follows:

Chapter 1 presents an overview about the Conflict Avoidance (CA) problem, including an exhaustive state of the art on the subject.

In Chapter 2, the Conflict Avoidance (CA) problem is described, and 2 models are presented: a pure 0-1 and a mixed 0-1 linear, the first of which avoids conflicts by means of altitude changes, while the second's strategy is based on altitude and speed changes. Then, a computational experience is reported, showing that both problems are solvable in really small elapsed times, for what the approach can be used in real time with the help of a state-of-the-art mixed integer linear optimization software. Finally, some improvements are discussed, among others, the inclusion of a third elusion maneuver, namely route changing.

Chapter 3.11 introduces some basic concepts of Stochastic Programming, and presents a state-of-the-art description of SP and current algorithms designed to solve stochastic mixed 0-1 multistage problems. Finally, a brief introduction to parallel programming is outlined, including the MPI library, which is the tool used to implement the parallel algorithm presented in Chapter 4.

In Chapter 4 two algorithms, FRC-J and FRC-PJ, based on matheuristics, are presented for solving large-scale multistage mixed 0-1 problems under uncertainty, where FRC-PJ is a parallel version of FRC-J.

Chapter 5 presents a broad computational experience for assessing the quality of the algorithm frameworks introduced in this thesis. For this purpose, a series of computational experiments have been run for large-scale instances of the NP-hard Multistage Stochastic Facility Location Problem (**MSFLP**)

Finally, Chapter 6 remarks some conclusions and outlines some further research areas.

Diffusion

The main results of this thesis have been published in [23, 24] and presented in several national and international meetings:

- XXXIII National Congress of Statistics and Operations Research. *A parallel computing meta-heuristic for solving multistage stochastic mixed integer programs.* Madrid (Spain), 2012.
- 12th International Conference on Stochastic Programming. *FRC: A heuristic extension of the BFC approach for solving very large scale multistage mixed 0-1 stochastic programs.* Halifax (Canada), 2010.
- ECCO XXIII-CO2010 European Chapter on Combinatorial Optimization. *On The Conflict Avoidance for Air Traffic Flow Management Problem, two models.* Málaga (Spain), 2010.
- 24th European Conference on Operations Research. *FRC: A heuristic extension of the Branch-and-Fix Coordination approach for solving very large scale multistage mixed 0-1 stochastic problems.* Lisbon (Portugal), 2010.
- 24th European Conference on Operations Research. *Conflict Avoidance for Air Traffic Flow Management Problem, pure and MIP models.* Lisbon (Portugal), 2010.

- 4th workshop on Optimization and Variational Analysis. *On solving large-scale stochastic mixed 0-1 linear problems. Seminario invitado.* Elche (Spain), 2010.
- 23rd European Conference on Operational Research. *On the collision avoidance for air traffic management problem, a large scale mixed 0-1 program approach.* Invited Session. Bonn (Germany), 2009.
- 23rd European Conference on Operational Research. *On the air traffic flow management problem. A stochastic integer programming approach. Invited Session.* Bonn (Germany), 2009.
- Workshop CORAL 2009 (Conference on Routing and Logistics). *Air traffic flow management. Airports and sectors constraints, with en route problem.* Elche (Spain), 2009.

Chapter 1

Conflict Avoidance: State of the art

The objective of this first chapter of preliminaries is to provide the reader with a basic knowledge about the Conflict Detection and Resolution (CDR) or Conflict Avoidance (CA) problem, the state of the art on the subject and the main contributions of this work.

1.1 Conflict Avoidance Introduction

Air traffic in Europe and the USA has undergone an astonishing growth during recent years. In 1999 a 50% increase was expected by 2018 over the traffic in that year, see Air Traffic Action Group [5]. In this scenario, the aim of Air Traffic Flow Management consists of extending the airspace allowing the so called "*Free Flight*", where the pilots and the airlines are able to decide freely the flight plan, keeping in touch with the air traffic controller. To maintain safety in the air flow, the CA problem is currently attracting the interest of air transportation service providers and has been studied extensively.

CA is concerned with the following question: Once the flight configuration for a set of aircraft is known, how to draw up a new configuration that prevent the aircrafts from coming too close to each other, i.e. avoid any conflict situation.

There have been built methods for maintaining separation between aircraft in the current airspace system. Humans are an essential element in this process due to their ability to integrate information and make judgments. However, because failures and operational errors can occur, automated systems have begun to appear both in the cockpit and on the ground to provide decision support and to serve as traffic conflict alerting systems. These systems use sensor data to predict

conflicts between aircraft and alert humans to a conflict and can provide commands or guidance to resolve the conflict. Relatively simple conflict predictors have been a part of air traffic control automation for several years, and the Traffic Alert and Collision Avoidance System (TCAS) has been in place onboard domestic transport aircraft since the early 1990s. Together, these automated systems provide a safety net that should provide normal procedures to help controller and pilot when human actions fail to keep aircraft separated beyond established minima.

Recently, interest has grown toward developing more advanced automation tools to detect traffic conflicts and assist in their resolution. These tools could make use of future technologies, such as a data link of current aircraft flight plan information, to enhance safety and enable new procedures to improve traffic flow efficiency.

With the growth of airspace congestion, there is an emerging need to implement these types of tools to assist the human operators in handling the expanding traffic loads and improve flow efficiency.

Unfortunately, the CDR has proven to be a hard problem to solve. To give some idea, the way in which to represent the actual trajectory of an aircraft is by means of a dynamic model that has to take into account, as an example, the following relationships: speed of the aircraft will depend on the wind direction and altitude on which it flies (such that the higher an aircraft flies, the lesser the air is around it and thus it needs to go faster to maintain its position); acceleration depends on the speed (e.g., at lower speeds, a plane can reach higher acceleration ratios) and altitude, and so on. Notice that the aircraft is losing mass throughout the flight as fuel burns, and this influences the speed and acceleration of the aircraft (and, viceversa, the speed influences the consumption of fuel and thus the mass loss), etc. Good introductions to flight dynamics modelization can be found in [101, 133, 225]. Finally, CDR has to deal with the simultaneous trajectories of (possibly) many aircraft. Moreover, we must bear in mind that given the intended trajectories, captured in the flight plans, some uncertainty regarding the actual trajectories of the aircraft is unavoidable, which makes CDR harder to solve. Trying to address all these issues within a mathematical optimization model would lead today to an unmanageable problem (in terms of computing effort, i.e., elapsed time and memory requirements).

To begin with, it is necessary to have a clear definition of a conflict. A conflict is an event in which two or more aircraft experience a loss of minimum separation. In other words, the distance between aircrafts violates a criterion that is considered undesirable. One example criterion is a minimum of 5 nm (nautical miles) of horizontal distance between aircraft or at least 1000 ft of vertical separation (the current en-route separation standard at lower altitudes). As a result, each aircraft has as a safety zone a cylindrical volume of airspace with a 2.5 nm radius and a height of 1,000 ft (500 ft

above and 500 ft below), and they are not allowed to intersect. The safety zone could also be defined as a much smaller region (e.g., a sphere 500 ft in diameter) in the case of tactical collision alerting systems. In any case, the underlying CDR functions are similar, although the specific models and alerting thresholds would probably be different.

The goal for the CDR system is to predict that a conflict is going to occur in the future, assist in the resolution of the conflict situation and communicate the detected conflict to a human operator.

Traditionally, absolute 4D trajectory-based air traffic management and control and relative aircraft-to-aircraft-based spacing concepts have been investigated as alternative pathways. Prevot et al. [200] describe research in European programs, such as PHARE and Co-space and in US, programs such as DAG-TM, and they finally suggest to investigate combinations of these elements.

A recent paper by EUROCONTROL [102] specifies the requirements of Medium-Term conflict Detection (MTCD) (e.g. up to 20 minutes) for Air Traffic Management Systems in detail. The MTCD system is required to detect and notify to the Air Traffic Control Officer (ATCO) a probable loss of the required separation between two aircraft, or aircraft penetrating restricted airspace. This paper assumes that flight data and trajectories are provided to the MTCD, where some uncertainty is expected and has to be taken into account.

Magister [157] presents two different models: The first applies to conflict detection. The second is related to conflict resolution to solving the conflict by lowering one of the two aircraft that are taken into consideration in the conflict. In addition, the same author [158] describes the conflict resolution problem in great detail and makes a quantitative analysis of avoidance procedures.

The remainder of this Chapter is organized as follows. Subsection 1.2 is devoted to the most interesting papers existing in the literature on Collision Detection. Section 1.3 is devoted to the main categories of Conflict Resolution. Section 1.4 presents a few problems similar to CA in the airspace and classifies the literature mentioned in this state of the art in a table. Finally, section 1.5 summarizes the main contributions of this thesis on the CA problem.

1.2 Conflict Detection

Conflict Detection (CD) is the process of detecting conflicts among two or more aircrafts, or between an aircraft and some other airspace constraint such as restricted airspace or regions of bad weather.

Let two aircrafts moving on the same horizontal plane, each following its individual flight plan. The flight plan is assumed to consist of a sequence of waypoints (a waypoint is a reference point in physical space used for purposes of navigation, it consists of a tupla with latitudinal and longitudinal coordinates, plus altitude with respect to a reference geoid) on the plane and a sequence of speeds for moving between them. One can then define the probability of conflict (PC) as the probability where two aircraft will be within an unsafe distance from one another (typically 5 nm outside the TRACON - Terminal Radar Approach Control - and 3 nautical miles inside the TRACON). Conflict detection consists of estimating PC.

An early work, from Chiang et al. [61] use Delaunay and Voronoi Diagrams. With this research, they reduce the conflict detection algorithm from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. A contemporary work, by Erzberger et al. [83] combines deterministic trajectory prediction and stochastic conflict analysis to achieve reliable conflict detection. They formulate error models for trajectory prediction, and describe an efficient algorithm for estimating conflict probability as a function of encounter geometry.

Prandini et al. [199] outline a framework for conflict detection and resolution for pairs of aircrafts moving on the same horizontal plane, and they focuss on the prediction component. They propose a probabilistic framework, thus allowing uncertainty in the aircrafts positions, and they solve the problem resorting to appropriate randomized algorithms.

Prandini et al. [198] deal with aircraft conflict detection at the mid-range and short range levels of the ATMS. Starting from an empirically motivated probabilistic description of the aircraft motion, they propose stochastic models for mid-term and short term prediction of the aircraft positions, thus allowing the corresponding criticality measures to take into account the various sources of uncertainty inherent in the environment. Although they focus on the planar case, they sustain that the extension to the 3-D case is straightforward suggested. However, it is considerably harder to get meaningful bounds for the error of such approximations.

Gandhi et al. [110] describe approaches to detect airborne obstacles on collision course and crossing trajectories in video images captured from an airborne aircraft. The crossing target detection algorithm was also implemented on a pipelined architecture from DataCube and runs in real time. Their work has been successfully tested on flight tests conducted by NASA.

Hu et al. [128] study the problem that consists of evaluating whether the flight plan assigned to an aircraft is safe. They introduce a kinematic model of the aircraft motion in a three dimensional wind field with spatially correlated random perturbations. Then they propose an iterative algorithm based on a Markov chain approximation scheme. The same authors [129] introduce a model of a

two-aircraft encounter with a random field term to address correlation of the wind perturbations to the aircraft motions. Based on this model, they estimate the probability of conflict by using a Markov chain approximation scheme

Jardin [140], presents some algorithms for strategic conflict detection, based on the use of a 4-dimensional space and time grid to represent the airspace. This approach to compute conflict detection was previously introduced by Jardin [138, 139], where he uses a 3-dimensional grid (two horizontal spatial dimensions and time).

Prandini and Hu [197] present a stochastic approximation scheme to estimate the probability that a single aircraft will enter a forbidden area of the airspace within a finite time horizon. A numerical algorithm is also proposed for computing an estimate of the probability that the aircraft might enter an unsafe region of the airspace or come too close to another aircraft.

1.3 Conflict Resolution

Several methods have been proposed to generate a solution to a conflict. Kuchar and Yang [147] present a survey of CDR modeling methods with their own classification. Dowek and Muñoz [78] present a mathematical framework for the formal specification and analysis of conflict detection and resolution algorithms and their properties. For this *State of the Art* six categories have been picked up, namely, Prescribed, Optimized, Force field, Manual, Neural Networks and Others. And finally the works presented in two conferences of aviation that took place in 2009 are commented.

1.3.1 Prescribed

The studies of this category study the standard maneuvers that airborne aircraft are able to carry out to avoid simple conflicts. Resolution maneuvers are fixed during system design based on a set of predefined procedures. NASA [178] and Carpenter and Kuchar [56] assume that a fixed climbing-turn maneuver is always performed to avoid traffic on a parallel runway approach. Prescribed maneuvers may have the benefit that operators can be trained to perform them reflexively. This may decrease response time when a conflict alert is issued. However, prescribed maneuvers are less effective, in general, than maneuvers that are computed in real time since there is no opportunity to modify the resolution maneuver. In many conflicts, it will be necessary to adapt the resolution maneuver to account for unexpected events in the environment, or to reduce the aggressiveness of the

maneuver.

Peng and Lin in [187] study some horizontal maneuvers escape for a traffic alert and collision-avoidance system (TCAS), separately developing speed and flying direction changes.

1.3.2 Optimization

This type of approaches typically combine a kinematic model with a set of cost metrics. An optimal resolution strategy is then determined by finding the no-conflicts trajectories with the lowest cost. For example, the Traffic alert and Collision Avoidance System (TCAS), which is an implementation of the Airborne Collision Avoidance System mandated by the International Civil Aviation Organization, searches through a set of potential climb or descent manoeuvres and selects the least-aggressive one that still provides adequate protection; see [212]. This requires the definition of appropriate cost functions, typically projected separation, or fuel or time, but costs could also cover workload. Developing costs may be fairly straightforward for economic values but difficult when modeling subjective human utilities. Since current interest in this field is generally centered on strategic resolution of conflicts before immediate tactical evasion is required, economic costs and operator workload will be important to the system design.

Some of the models denoted as using Optimized conflict resolution apply techniques such as game theory, genetic algorithms, expert systems, or fuzzy control to the problem. Expert system methods use rule bases to categorize conflicts and decide whether to alert and/or resolve a conflict. These models can be complex and would require a large number of rules to completely cover all possible encounter situations. Additionally, it may be difficult to certify that the system will always operate as intended, and the “experts” used to develop or train the system may in fact not use the best strategy in resolving conflicts. However, a rule base, by design, can be easier than it is an abstract mathematical algorithm for a human to understand or explain.

Krozel and Peters [146] analyze collision avoidance problem in free flight context, taking economics features into consideration. They use relative motion of two aircrafts in a horizontal plane and vectorial and probabilistic calculations to detect if a conflict occurs, so the model is non-deterministic. To solve the collision avoidance problem, first of all, they consider economic factors, like fuel consumption, and time factors, like time required to execute the maneuver and return back to course. The authors also order the different possible maneuvers relative to the cost of these ones, being altitude changes the most economical, and speed changes the worst economical.

Frazzoli et al. [108] solve a planar, multi-aircraft conflict resolution problem, formulated as a nonconvex, quadratically constrained quadratic program, and then approximating it by a convex, semidefinite program. The optimal solution to this convex program is then used to randomly generate feasible and locally optimal conflict resolution maneuvers. Every individual aircraft should then be able to express their preferences at regular time intervals; they are given always conflict-free, straight paths.

Mao et al. [160] set out geometric constructions to solve the problem, including aircraft one-by-one until representing the total number of aircraft, considering the previous aircraft as obstacles and making a sequential process.

Hu has devoted a series of papers and two Thesis to this problem. His work is very interesting for us, since it tackles the most general case: many aircraft encounters in three dimensional space. Particularly, in [126], Hu et al. study the problem of designing optimal conflict-free maneuvers (a maneuver is defined to be a continuous and piecewise C^1 map) for multi-aircraft encounters in a three dimensional environment, proposing an algorithm for solving the two aircraft nonlinear optimization problem. For more than two aircraft, they consider what is called two-legged maneuvers approach, such that a manoeuvre consists of two stages, moving at a constant speed and through a straight line during both stages. The original optimization problem is then reduced to a finite dimensional convex optimization problem with linearly approximated conflict-free constraints on the waypoints and a quadratic objective function. Path flightability is taken into account by introducing an upper bound on the speed and turning angle constraints, which can be expressed by using second order cone expressions. So, the optimization problem becomes a Second Order Cone Programming (SOCP) one.

However, the assumptions on which the proposal are based (namely, every aircraft departs and arrives at the same time, all aircraft move linearly except for one heading angle change in the two-legged manoeuvre, etc.) force to apply the model recursively, which could make it unaffordable as an option in most practical cases, due to the non-linearity of its constraints and objective function. In [127], the same authors study the problem as above, although constrained to the plane, proposing a randomized convex optimization algorithm to find numerically the optimal multi-legged manoeuvres (with an arbitrary number of stages).

Finally, Hu [124], Hu and Sastry ([132], Hu [125], Hu et al. [130]) and Hu et al. [131] study the more general problem of optimal collision avoidance and optimal formation switching for multiple agents moving on a Riemannian manifold.

Pallottino et al. [185] propose two mixed integer models for CDR, the first one allows speed

changes and the other one angle changes, both on the same plane. These models are based on a geometric approach. The second model assumes that the speed is the same for all aircraft, such that each one can manoeuvre only once with an instantaneous heading angle deviation that can be positive (left turn), negative (right turn) or null (no deviation). It does not consider returning to the original route, nor does it explain how the aircraft, after a manoeuvre, reaches its destination. Alonso-Ayuso et al. [16] use the geometric idea proposed in [185], whose first model (velocity changes model) is extended to permit aircraft changing both their speed and altitude levels, resulting in the so-called Velocity and Altitude Changes(VAC) model, which is based on Mixed Integer Linear Optimization (MILO), thus infeasible situations caused by ”head to head” conflicts are avoided. Moreover, all aircraft will be forced to return to the initial configuration when conflict situations are resolved and, finally, a pathological case unresolved in [185] is avoided. These two approaches are intended to be executed repeatedly, each execution within a short time horizon. The trajectories are assumed to be linear over a horizontal plane (even though flight level changes are allowed). Notice that projecting the trajectories onto a plane could appreciably change the actual angles, which makes these models suitable only for small airspace regions in the short term. Later they extended the VC model to a Mixed Integer Non Linear Optimization (MINLO) model by including continuity in the velocity changes, since the original considers that all changes happen instantaneously, see [17]. Later on, in [18], they presented an approximation for coordinating different Air Traffic Controllers Officers (ATCO) in different air sectors. In [19], a two-step approach is presented, the first step being a nonconvex MINLO model based on geometric constructions aimed at minimizing the weighted aircraft angle variations to obtain the new flight configuration, and the second step consisting of a set of unconstrained quadratic optimization models where aircraft are forced to return to their original flight plan as soon as possible once there is no aircraft in conflict with any other. In [22], the Variable Neighborhood Search metaheuristic is used for solving the CDR by turn changes MINLO model previously proposed. In [20], an exact MINLO model is presented and, finally, in [21], a multicriteria scheme based on Goal Programming is presented and a Sequential Mixed Integer Linear Optimization (SMILO) approach is proposed in order to provide a good solution in short computing time for solving the previous models.

Peyronne et al. [190] propose a trajectory using B-splines and a semi-infinite programming formulation for solving the CDR problem via turn changes. It only uses continuous variables, but the computational experiment is restricted to six aircraft in conflict.

Recently, Rey et al. [203] have presented a MILO model where speed changes are used for avoiding conflict situations. Cafieri and Durand [50] also propose a MINLO model based on velocity regulation considering different time instants for performing velocity changes.

Cafieri and Durand [49] focus on mixed 0–1 nonlinear optimization (MINLO) models for conflict avoidance based on speed regulation, while trajectories are kept unchanged and considering different times for velocity changes.

Obstacle avoidance by using the linearized constrained Uninhabited Aerial Vehicle (UAV) dynamic has been modeled by Richards and How [204]. Centralized Model Predictive Control has been widely developed for constrained systems and has been applied to the co-operative control of multiple vehicles. By augmenting the system with a binary ”target state”, that indicates whether the target set is reached or not, the authors end up with a hybrid system at hand. Task completion is then guaranteed by imposing a hard terminal equality constraint on the target state. See also [205].

Christodoulou and Costoulakis [62] propose a MINLP model for solving the conflict problem. The method allows velocity changes and heading angle control to solve all potential conflicts by using standard optimization software, but it can require more computational effort than what it could be affordable.

Ma and Miller [156] present a MILP Trajectory Generation model applied to a rotorcraft performing nap-of-the-earth flight in challenging terrain with multiple known surface threats, i.e. they work on a concrete application of optimal path planning for an autonomous vehicle in an obstacle field in three dimensions.

Schouwenaars et al. [215] discuss the implementation, by using the state-of-the-art optimization engine, of a guidance system based on MILP on a modified, autonomous T-33 aircraft equipped with Boeing’s UCAV avionics package. Their formulation is presented for safe, real-time trajectory generation in a partially-known, cluttered environment.

Mao et al. [162] tackle the problem using instantaneous heading changes as manoeuvres between two aircraft. This paper extends the results of Mao et al. [160] in which the manoeuvres that have been considered are not physically realistic.

A MINLP model proposed by Christodoulou and Kodaxakis [63], with linear objective function and nonlinear constraints only allows speed changes as manoeuvres.

Treleaven [227] assumes that aircraft travel at the same altitude and with the same speed, and uses only horizontal maneuvers for the conflict resolution. This analysis is extended to consider two, three and multiple intersecting flows.

Pannequin et al. [186] present an approach to the problem with severe weather conditions by

using a Nonlinear Model Predictive Control (NMPC) scheme.

Finally, Cetek [58] presents a model that takes into account many physical features like wind speed, relative density at the given flight altitude, gravitational acceleration, mass of the aircraft, aerodynamic drag force, etc. The model is non-linear and it is necessary a great effort to solve the problem, in fact the computational experience that is reported shows very large resolution times (more than 10 minutes), and it is not valid neither for imminent conflicts (short term) nor medium term. In this model, vertical and heading maneuvers are not contemplated (only speed changes), being "head to head" conflicts impossible to solve.

1.3.3 Force field

This type of approaches treat each aircraft as a charged particle and use modified electrostatic equations to generate resolution maneuvers. The repulsive forces between aircraft are used to define the maneuver that each performs to avoid a collision. A force field method, while attractive in the sense that a conflict resolution solution is continuously available by using relatively simple equations, may have some pathologies that require additional consideration before they can be used in operation. For example, force field methods may assume that aircraft continuously maneuver in response to the changing force field, or that aircraft can vary their speed over a wide range. This requires a high level of guidance on the flight deck and increases complexity beyond issuing simple heading vectors, for example. Additionally, sharp discontinuities in the commanded resolution maneuvers may occur that require additional processing or filtering to arrive at physically feasible solutions. Several human-in-the-loop implementations of the force field method, however, appear to have resolved these problems and have shown that the force field based on resolution can be effective when properly applied. See Duong and Hoffman [79], Hoekstra, Van Gent and Ruigrok [121] and Zeghal and Hoffman [244].

It has also been suggested that potencial fields can be used in UAV navigation for obstacle and collision avoidance applications. Sigurd and How [221] proposed a method that provides a way for groups of UAVs to use the gradient of a potencial field to navigate through heavily populated areas safely while still aggressively approaching their targets.

1.3.4 Manual

Some models allow the user to generate potential conflict resolution solutions and obtain feedback as to whether the trial solution is acceptable. These models are denoted as handling a manual

solution in the table. The benefit of a manual solution is that it is generally more flexible in the sense that it is based on human intuition, using information that may not be available to the automation. For example, weather information that is not available to the CDR system may be important when considering a conflict resolution maneuver. Automated solutions that do not take relevant environmental information into account will likely produce nuisance solutions that the human finds unacceptable.

1.3.5 Neural Networks and metaheuristics

In the fifties, artificial neural networks began to be applied to many fields in artificial intelligence like analysis and adaptive control, speech recognition, etc. These networks are based on statistical estimation and optimization and control theory. This field has been used in the CA problem together with other heuristics techniques like genetic algorithms.

Durand et al. [81] have built a neural network with unsupervised learning to compute new trajectories that are close to the optimal trajectory in conflicts with two aircraft. This 3-layer neural network modifies only the heading direction (not more than 45 degrees) of the aircraft if there is a conflict between them. To help the training of the neural network, the authors use a genetic algorithm. This network is not valid for conflicts involving more than two aircraft, however they provide some schemes to solve conflicts involving three aircraft, but not more.

Alam et al. [7] have constructed a neural network that computes near optimal trajectories to solve two aircraft conflicts in a 2D free flight environment, allowing only heading changes maneuvers, since they assume that the aircraft fly at constant speed. A three layer artificial neural network architecture is used to lead the aircraft to the destination point. This neural network does not confront more than two aircraft in a same conflict, and it is not valid in a 3D environment. These are important drawbacks for the problem.

Doshi [77] presents a neural network to predict the position of the aircraft by using event history, being a long history a better choice than a short history since it reduces noise in the model. With this neural network, the conflict detection is obtained by computing separation distances between points of the prediction. An algorithm for determining the incursion distance between two aircraft is presented. It is based on trigonometric analysis and yields SD values useful for danger detection, but it only detects danger conflicts.

Christodoulou and Kontogeorgou [64] present a neural network to predict the optimal speed change for two aircraft in order to avoid an imminent conflict in a 3D environment. The algorithm

combines the neural network with non-linear optimization to obtain the optimal speed change. For each conflict case there is an unique model based on non-linear optimization and it is solved in a 3D environment, but only speed changes are allowed. Vertical maneuvers are not considered since it is assumed that aircraft can fly out of a space divided in layers. It is also assumed that the aircraft fly at constant speed and the motion direction is linear.

Other metaheuristic methods have been used to solve the problem as Ant Colony optimization (see Durand and Alliot [80] and Meng and Qi [167] for solving the problem by performing angle change maneuvers). Genetic Algorithms (see Medioni et al. [166] by also performing turn changes and Vivona et al. [232] for prescribed maneuvers). Particle Swarm optimization (see Gao et al. [111] where again the turn change maneuver is used)

1.3.6 Others

Chiang et al. [61] solve the conflict problem with which they call the *Space-Time Flow* (STF) Method. It is based on an iterative procedure for adding tubes (aircraft) using a graph search in a discretized space-time to route each tube amongst the already routed tubes, which are considered to be obstacles.

Tomlin et al. [226] develop a method to solve the conflict problem by using both speed and heading changes. The algorithm is based on Lie algebra and Hamilton-Jacobi-Isaacs equations.

Goodchild et al. [117] propose a cooperative optimal conflict resolution algorithm based on distributed artificial intelligence, by using a dynamic optimization algorithm

Bicchi and Pallottino [40] use optimal control and game theory to solve the problem. The model assumes that linear velocity is constant and allows to maneuver all the aircraft including air-speed, several angles, heading angle, longitude and latitude as parameters.

Bayen et al. [33] propose a Lagrangian model where the maneuvers are simple instructions such as turn to heading angle, fly direct to a concrete point and speed increase. The model permits aircraft to fly at different altitudes, but not to climb or descend. This model permits a shortcut or detour maneuver that could either shorten or lengthen the flight plan.

1.4 Similar problems and Review table

A problem applied in military aircrafts is presented by Sharma et al. [220] whose model considers aircrafts flying in close proximity to terrain and they have to avoid all fixed obstacles like mountains. The problem is modeled in a 3D environment using a minimax optimal control problem formulated as a nonlinear programming problem. This model has a nonsmooth cost function that is transformed into a smooth cost by introducing additional inequality constraints.

There are other problems treated in the literature that are related to collision avoidance applied in other fields like underwater or automobile routes where the machine have to avoid fixed obstacles in its path. Neural networks have been used to tackle this type of problems, see Ishi et al. [135], Nishida et al. [179] and Ishi et al. [134] applying this results to an underwater robot. Mukai et al. [172] and [173] use their results to generate a new path avoiding the collision with a fixed obstacle by using MILP. Finally, Kim et al. [144] model the same problem as a MILP model but by using a piecewise polynomial approach which is a class of the hybrid dynamical system.

In Table 1.1 the main features of the literature that has been previously reviewed are presented. The Collision Detection Problem (CDP) and Collision Resolution Problem (CRP) are distinguished. With regard to the type of maneuver three types of maneuvers are classified: Horizontal (H), Vertical (V) and Speed changes (S). The papers shown in the table are probably the most relevant among those mentioned in this chapter.

Table 1.1: Review table

Reference	CDP	CRP	Maneuvers	Number of aircrafts	Dimension
[7]	X	X	H	2	2D
[8]	X			N	3D
[33]	X			N	2D
[40]	X	X	H	N	2D
[56]	X	X	H,V	2	2D
[58]	X	X	S	2	2D
[61]	X	X	HV	N	3D
[62]	X	X	HS	N	2D
[63]	X	X	S	N	3D
[64]	X	X	S	N	3D
[72]		X	S,V	N	1D
[77]	X			N	2D
[78]	X	X	H,S,V	N	3D
[81]		X	H	2	2D
[83]	X	X	H,S,V	N	3D
[110]	X			1	2D
[117]		X	S	N	2D
[124]		X	H,S	N	2D
[125]		X	H,S,V	N	3D
[126]		X	H,S,V	N	3D
[127]		X	H,S	N	2D
[128]	X			2	3D
[129]	X			2	3D
[131]		X	H,S,V	N	3D
[130]		X	H,S,V	N	3D
[132]		X	H,S,V	N	3D
[138]	X	X	H	N	3D
[139]		X	H,S,V	N	3D
[140]	X			N	4D
[146]	X	X	H, V, S	N	3D
[156]	X	X	H,S,V	1	3D

Table 1.2: Review table

Reference	CDP	CRP	Maneuvers	Number of aircrafts	Dimension
[185]	X	X	H, S	N	2D
[186]		X	H,S,V	N	3D
[197]	X			1	2D
[198]	X			N	2D
[199]	X			2	2D
[200]		X	H	N	3D
[202]		X	S	N	3D
[204]	X	X	H,V	N	2D
[205]		X	H,V	N	2D
[221]		X	H	N	2D
[226]		X	H	2,3	2D
[161]		X	H,S	3	2D
[160]		X	H	N	2D
[162]		X	H	N	2D
[178]	X	X	H,V	2	2D
[190]		X	H	6	2D
[16]		X	H,V	6	3D
[49]		X	S		3D

1.5 Problem description

Conflict Avoidance has been widely studied as can be seen on this chapter. There are quite a lot of different points of view from which the problem can be tackled. Three different maneuvers are considered to avoid a conflict, namely: velocity, altitude and heading angle changes. To begin with, in this thesis we will consider two maneuvers: speed and altitude changes. But we will study other alternatives in order to take into account independently heading angle changes. Heading angle changes have a nonlinear nature, and most of the models that take them into account are nonlinear, although some exceptions can be found, such as a paper of Pallottino, Feron and Bicchi (2002) [185] that solves the problem with heading angle changes using mixed integer linear programming, but the resulting model does not return aircrafts to their initial direction. And there are also a series of papers, from Alonso-Ayuso et al. [16, 17, 18, 19, 20, 22, 21] extending the approach proposed by Pallottino et al. in an interesting way, although nonlinearity is eventually reintroduced, some heuristics are applied aimed at finding good solutions earlier.

The general approach is by developing a dynamic model trying to predict the future aircraft positions from their current situation and then deciding if a conflict will take place. However, some

uncertainty is inevitable and has to be taken into account in the model at the risk of not offering a reliable prediction. On the other hand, such models are typically non-linear and will be constrained to consider just a few aircraft in order to be solved in a reasonable time. The approach proposed in the next chapter is about basing the model on previous information like a flight plan. Notice that there are models for finding the optimum trajectory for an airborne aircraft and they are efficient enough (see, e.g. [222]), however, it is not easy to extend such models for dealing with more than one aircraft and eluding conflicts among them. It would be relatively easy to create such model but it would be computationally unmanageable.

In the next chapter we present a model based on mixed integer linear programming to solve conflict avoidance problems, using only two maneuvers to avoid conflicts: velocity and altitude changes. It is a dynamic model that can manage a great number of aircraft and that for each time period under consideration yields the altitude (i.e. flying level) and the speed the aircraft should fly so that collisions are avoided.

Chapter 2

Conflict Avoidance: 0-1 linear models for Conflict Detection & Resolution

The Conflict Detection and Resolution Problem for Air Traffic Flow Management consists of deciding the best strategy for airborne aircraft so that there is guarantee that no conflict takes place, i.e., all aircraft maintain the minimum safety distance at every time instant. In this chapter, two integer linear optimization models for conflict avoidance between any number of aircraft in the airspace are proposed, the first being a pure 0-1 linear which avoids conflicts by means of altitude changes, and the second a mixed 0-1 linear whose strategy is based on altitude and speed changes. Several objective functions are established. Due to the small elapsed time that is required for solving both problems, the approach can be used in real time by using state-of-the-art mixed integer linear optimization software.

The main contributions of this chapter are as follows:

1. A new point of view has been adopted, so that it does not tackle the CDR problem by directly modeling the aircraft trajectories nor the physical laws under which the aircraft have to fly. On the contrary, the approach rests on the idea of decomposing the problem, using the aircraft trajectories that actually incorporate such physical laws and the airlines preferences, and with a (hopefully, small) room for maneuver proposes the needed conflict avoidance maneuvers, if any. Additionally, only linear models are required which can be computed in very small elapsed time.

2. Two novel optimization models are proposed. The first one is a pure 0-1 linear model, aimed

at changing altitude levels (i.e., forcing the aircraft to climb or descend in order to avoid conflicts). The second model is a mixed 0-1 linear one that solves the problem by changing aircraft altitude levels and speed. Both models are very tight and, then, require very small elapsed time for solving even large-scale instances, so, they can be used in real time for realistic conflict detection and resolution problems.

3. The approach is flexible, allowing non-linear trajectories, which can be the beacon points of the currently used flight plans, or the future freely decided optimal trajectories in the context of "Free Flight", or the simple straight-line extrapolation of the speed vector at a given instant, as in [16, 185]. Speed is not restricted to be constant as it is the case in many other approaches found in the literature.

4. Additionally, due to its really small elapsed times, our approach is specially suited for being used in long term time horizons as well as in wider airspace regions that may comprise several air traffic control areas in which the aerial space is divided.

5. Finally, both models are scalable and can be easily extended and improved with new features or restrictions, e.g. allowing alternative routes as a third conflict avoidance maneuver, as will be outlined later (although it is left for further research).

Based on the computational experience reported in Section 2.4, we can point out that our first model is tighter than the second one (and, then, it requires smaller computational effort), so, it allows considering wider aerial zones with a higher set of aircraft and a longer time horizon than the second model. Nevertheless, this second model is quite efficient, according to the computational experience to report below. On the other hand, the first model has the drawback of only allowing altitude level changes, a manoeuvre that may not be the preferred choice by passengers because they are more uncomfortable. Nonetheless, in most real-life cases very few of such altitude level changes should be necessary and this model will be useful and enough in most practical situations. Further more, it may be the preferred manoeuvre, as opposed to speed changes, since the latter may imply greater fuel consumption and be more uncertain than the former. Actually, velocity changes are not considered a very efficient maneuver in the literature; see Frazzoli et al. [108], Jardin [138] and Peyronne et al. [190], among others. Moreover, in many papers aircraft velocity is assumed to be constant, see Pallottino et al. [185], Christodoulou and Costoulakis [62], Treleaven [227], Gao et al. [111] and Cafieri and Durand [50], among others. To summarize, the proposed models are both efficient and useful in most real-life situations, the second being more comprehensive than the first one.

The remainder of the chapter is organized as follows: Section 2.1 technically introduces the

problem and some notation. Section 2.2 presents the first model, its preprocessing and its pure 0-1 formulation. Section 2.3 presents the second model, with some new elements, its preprocessing and its mixed 0-1 formulation. Section 2.4 reports the computational results for two testbeds of realistic airborne aircraft conflict instances. In Section 2.5 further ideas and extensions are presented. And, finally, section 2.6 concludes and outlines future work.

2.1 Problem description

A conflict is an event in which two or more aircraft are within an unsafe distance from one another at a given instant. The minimum safety distance is typically 5 nm (nautical miles) of horizontal distance between aircraft outside the TRACON (Terminal Radar Approach Control) and 3 nm inside the TRACON, or at least 1000 feet of vertical separation (the current en-route separation standard at lower altitudes).

Let us consider a set of aircraft $\mathcal{F}$. For each flight $f \in \mathcal{F}$, a dynamic trajectory model is required to project the states into the future in order to predict whether a conflict would occur. This projection may be based solely on current state information (e.g., a straight-line extrapolation of the current speed vector) or may be based on additional procedural information such as a flight plan. In both situations there is generally some uncertainty in estimating the future trajectory. It is represented via a finite sequence of waypoints, $\mathcal{W}_f$. A waypoint is a reference point in the physical space that consists of a tupla with latitudinal and longitudinal coordinates, generally with respect to a reference geoid. At each waypoint, we also know the scheduled speed for moving to the next waypoint. Let also define $\mathcal{W}_f^{'}$ and $\mathcal{W}_f^{-}$ as the sets of all waypoints but the first and the last ones, respectively, to transverse by flight f.

Let us assume that the route path for each aircraft is broken down into segments (not necessarily with equal size), altitude level and speed through each one of these segments, such that the number of waypoints for every aircraft is sufficiently representative of the route. Thus, the distance between two given consecutive waypoints (i.e., the length of a segment) should be less than 5nm (according to the current en-route separation standard at lower altitudes). So, 2nm can be a reasonable distance. In order to justify this choice, assume the extreme situation where two aircraft approach each other head-to-head from an initial distance of 5.1nm (so, no conflict occurs), but, after 1 nm of each route they are at the distance of 3.1nm each other. At this time instant, assume that they turn 180 degrees and come back to their initial waypoints. Each aircraft has been flying 2nm, in total, that is precisely the distance that we have assumed reasonable and no conflict has been detected, in

spite that the aircraft have been in conflict since they have been at a distance of 3.1 nm one from the other. However, this distance is big enough to consider the risk of a real collision. Notice that any more realistic situation is more favorable than the case just described, since the minimum distance between the aircraft would be greater than 3.1 nm. Moreover, the distance between two consecutive waypoints, although important, is case dependent (and, then, an input to the model of choice) and, then, it does not affect the tightness and, so, the validity of the models presented in this work.

Additionally, let $\mathcal{L}_i^f = \{\underline{z}_i^f, \underline{z}_i^f + 1, \ldots, \overline{z}_i^f\}$ denote the set of the allowed altitude levels for aircraft f to traverse waypoint i, for $f \in \mathcal{F}$, $i \in \mathcal{W}_f$. In order to prevent infeasible altitude level changes, let us define $\overline{V}_i^f$ ($\underline{V}_i^f$) as the max (min) number of altitude levels that aircraft f is allowed to climb or descend from waypoint i to the next one, for $f \in \mathcal{F}$, $i \in \mathcal{W}_f^-$. Let also define t_i^f and z_i^f as the scheduled time and altitude of aircraft f while traversing waypoint i in its route, for $f \in \mathcal{F}$, $i \in \mathcal{W}_f$.

We will consider that a conflict takes place if two aircraft traverse two waypoints in their respective routes that are too close to one another, within a small interval of time. To determine the bounds of such interval let us resort to a conservative strategy, and define $m_{A_{i,j}^{f,k}} = \max\{|t_{i+1}^f - t_i^f|, |t_{j+1}^k - t_j^k|\}$ as the smallest time interval that is allowed for aircraft f and k to reach their next waypoints $i+1$ and $j+1$ from the waypoints i and j, respectively, $\forall f, k \in \mathcal{F}, (i,j) \in \mathcal{W}_f \times \mathcal{W}_k$

So, the CDR problem to tackle consists of detecting all conflicts in the *alert zone* (being this one an aerial sector or even the whole airspace) and avoiding them by using a solution provided by very tight 0-1 linear optimization models that are solved by using a state-of-the art optimization engine. The proposed models suggest some changes (as few as possible) in altitude and speed of the aircraft scheduled trajectories.

2.2 Collision Avoidance via altitude level changes

2.2.1 Conflict Detection

The scheme proposed for aircraft conflict detection is similar for the two types of CDR problems to tackle in this work, namely, CA via altitude level changes and CA via altitude level and speed changes. Obviously, it helps to decide if a conflict can be avoided, if any, but it also helps to finding at which pair of waypoints a conflict would occur. Moreover, the conflict detection scheme have some differences between both approaches. The basic idea for the altitude level change scheme is as

follows.

For a pair of aircraft $(f,k) \in \mathcal{F} \times \mathcal{F}$, there is a *potential conflict* at the pair of waypoints $(i,j) \in \mathcal{W}_f \times \mathcal{W}_k$ if the following conditions hold:

1. The waypoints i and j have a smaller distance than the minimum allowed.
2. The time instants are such that $t_i^f < t_{j+1}^k$ and $t_j^k < t_{i+1}^f$. In order to justify this condition, suppose, on the contrary, that e.g., the second inequality does not hold, then, when aircraft k reaches waypoint j, aircraft f is at waypoint $i+1$, at least, and, so, no conflict between the aircraft k and f is possible at the pair of waypoints (i,j)).
3. The altitude levels are such that $\underline{z}_i^f \leqslant \overline{z}_j^k$ and $\underline{z}_j^k \leqslant \overline{z}_i^f$ since, otherwise, the aircraft f and k cannot be at the same altitude level while traversing the waypoints i and j.

Let $\mathcal{P}^{f,k} \subset \mathcal{W}_f \times \mathcal{W}_k$ denote be the set of all *potencial waypoint conflicts* between the aircraft f and k, and $\mathcal{F}^f \subset \mathcal{F}$ be the set of *potential aircraft conflicts* where aircraft f is involved, for $f,k \in \mathcal{F}$. Notice that $k \in \mathcal{F}^f$ iff $\mathcal{P}^{f,k} \neq \emptyset$.

Finally, we can define a partition of the aircraft set $\mathcal{F} = \bigcup_{i \in \mathcal{I}} \mathcal{F}_i$, $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset, \forall i,j \in \mathcal{I}$, where $f \in \mathcal{F}_i \Rightarrow \mathcal{F}^f \subset \mathcal{F}_i, \forall f \in \mathcal{F}, \forall i \in \mathcal{I}$ for splitting the problem into subproblems.

Similarly, for a pair of aircraft $(f,k) \in \mathcal{F} \times \mathcal{F}$, there is a *current conflict* at the pair of waypoints $(i,j) \in \mathcal{W}_f \times \mathcal{W}_k$ if $(i,j) \in \mathcal{P}^{f,k}$ and $z_i^f = z_j^k$.

Finally, let $C\mathcal{P}^{f,k} \subset \mathcal{W}_f \times \mathcal{W}_k$ denote the set of all *current waypoint conflicts* between aircraft f and k, and $C\mathcal{F}^f \subset \mathcal{F}$ be the set of *current aircraft conflicts* where aircraft f is involved, for $f,k \in \mathcal{F}$. Notice that $k \in C\mathcal{F}^f$ iff $C\mathcal{P}^{f,k} \neq \emptyset$.

As an illustration, let us consider the aerial zone depicted in Figure 2.1, where three aircraft cross their paths. Particularly, we can observe that waypoint i_2 is too close to the waypoints j_5 and j_6, which are within the safety disc drawn around waypoint i_2. Suppose that aircraft 1 is scheduled to fly through the waypoints i_2 and i_3 at time instants (e.g., seconds) 33 and 48, respectively, and aircraft 2 is scheduled to fly through the waypoints j_5 and j_6 at the time instances 54 and 71, respectively, (i.e. $t_{i_2}^1 = 33, t_{i_3}^1 = 48$, $t_{j_5}^2 = 54, t_{j_6}^2 = 71$). Then, there is not a potential conflict nor a current conflict at the pair of waypoints (i_2, j_5) (i.e., $(i_2,j_5) \notin C\mathcal{P}^{1,2} \subset \mathcal{P}^{1,2}$), since $t_{j_5}^2 > t_{i_3}^1$. However, there might be a conflict at the waypoints (i_2, j_6). So, the values $t_{j_6}^2$, $t_{j_7}^2$, $t_{i_2}^1$ and $t_{i_3}^1$ should be checked to evaluate it.

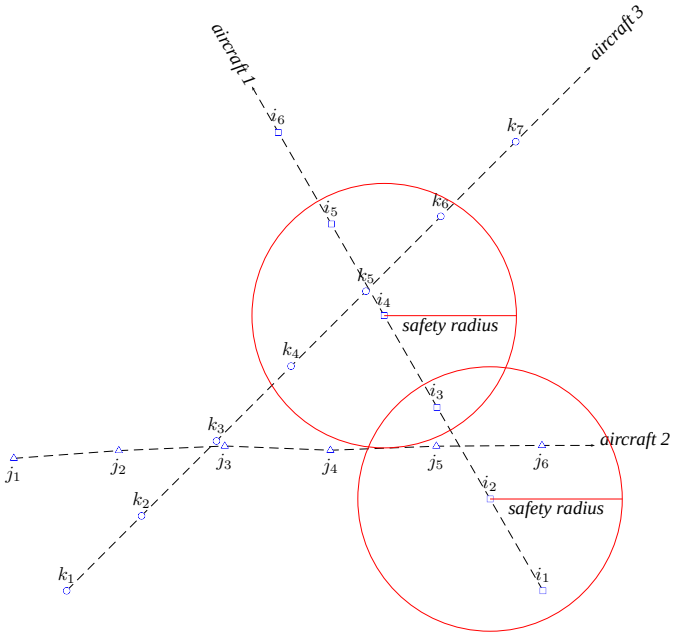


Figure 2.1: Illustrative case for three flight routes

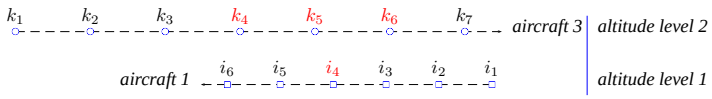


Figure 2.2: Illustrative case for different altitude levels of the routes of the aircraft 1 and 3

On the other hand, we can observe in the figure that waypoint i_4 is too close to the waypoints k_4, k_5 and k_6. So, suppose that e.g., $t^1_{i_4} = 63, t^1_{i_5} = 78, t^3_{k_4} = 65$ and $t^3_{k_5} = 85$. Then, we find out that $t^1_{i_4} < t^3_{k_4} < t^1_{i_5}$ and, so, the first and second conditions given above hold for the pair of waypoints (i_4, k_4) to belong to the sets $\mathcal{P}^{1,3}$ and $C\mathcal{P}^{1,3}$. To check if the third condition hold, the paths of the aircraft 1 and 3 depicted in Figure 2.2 should be analyzed on the axes x and z (i.e., abscissa and height). We can observe that both aircraft fly at different altitude levels and, so, no current conflict takes place, thus $(i_4, k_4) \notin C\mathcal{P}^{1,3}$. However, suppose that $\underline{z}^3_{k_4} = 1, \overline{z}^3_{k_4} = 2$ and $\underline{z}^1_{i_4} = 1, \overline{z}^1_{i_4} = 1$, then $\underline{z}^3_{k_4} = \underline{z}^1_{i_4} = \overline{z}^1_{i_4} < \overline{z}^3_{k_4}$ and, thus, $(i_4, k_4) \in \mathcal{P}^{1,3}$, since aircraft 3 is allowed to fly at altitude level 1 in waypoint k_4 and, so, a conflict may occur at the pair of waypoints (i_4, k_4) if such change is introduced by the model given below.

2.2.2 Model formulation for conflict resolution

The pure 0-1 model that we propose deals with the CDR problem by changing (i.e, climbing or descending) altitude levels for the aircraft in order to avoid *current conflicts*. It considers two objectives in a composite form, i.e., the maximization of rewards for the aircraft flying on the scheduled altitude levels and the minimization of penalizations of altitude level changes for the aircraft flying at levels different from those scheduled ones. Both objectives are optimized at all the given waypoints. So, the model assigns altitude level changes, if any, to the aircraft in order to guarantee that there will be no conflict among them.

Parameters

c_i^f **and** h_i^f, reward and penalization for changing (i.e., climbing or descending) the scheduled altitude level for aircraft f at waypoint i, respectively, $\forall f \in \mathcal{F}, i \in \mathcal{W}_f$.

0-1 variables

$\phi_{i,h}^f$, will take on the value 1 if aircraft f is at altitude level h at waypoint i in its route path and 0, otherwise, $\forall f \in \mathcal{F}$, $i \in \mathcal{W}_f$, $h \in \mathcal{L}_i^f$.

ν_i^f, will take on the value 1 if aircraft f changes its altitude level from waypoint i to the next one and 0, otherwise, $\forall f \in \mathcal{F}$, $i \in \mathcal{W}_f^-$.

The objective function includes two terms, namely, the reward for having the aircraft flying at the scheduled altitude levels and the penalization for flying at different levels than the scheduled ones.

The model is as follows,

$$\max \sum_{f \in \mathcal{F}, i \in \mathcal{W}_f, h = z_i^f} c_i^f \phi_{i,h}^f - \sum_{f \in \mathcal{F}, i \in \mathcal{W}_f^-} h_i^f \nu_i^f \tag{2.1}$$

subject to:

$$\sum_{h\in\mathcal{L}_i^f} \phi_{i,h}^f = 1 \quad \forall f \in \mathcal{F}, i \in \mathcal{W}_f \tag{2.2}$$

$$\phi_{i,h}^f \leqslant \sum_{\ell=\underline{V}_i^f}^{\overline{V}_i^f} \phi_{i+1,h+\ell}^f \quad \forall f \in \mathcal{F}, i \in \mathcal{W}_f^-, h \in \mathcal{L}_i^f \tag{2.3}$$

$$\phi_{i,h}^f \leqslant \sum_{\ell=\underline{V}_{i-1}^f}^{\overline{V}_{i-1}^f} \phi_{i-1,h-\ell}^f \quad \forall f \in \mathcal{F}, i \in \mathcal{W}_f', h \in \mathcal{L}_i^f \tag{2.4}$$

$$\phi_{i,h}^f - \phi_{i+1,h}^f \leqslant \nu_i^f \quad \forall f \in \mathcal{F}, i \in \mathcal{W}_f^-, h \in \mathcal{L}_i^f \tag{2.5}$$

$$\phi_{i,h}^f + \phi_{j,h}^k \leqslant 1 \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} h \in \mathcal{L}_i^f \cap \mathcal{L}_j^k \tag{2.6}$$

$$\phi_{i,h}^f, \nu_i^f \in \{0,1\} \quad \forall f \in \mathcal{F}, i \in \mathcal{W}_f, h \in \mathcal{L}_i^f. \tag{2.7}$$

Constraints (2.2) guarantee that all aircraft traverse every waypoint at only one altitude level. Constraints (2.3)-(2.4) ensure "soft" altitude level changes. Constraints (2.5) give the number of altitude level variations from one waypoint with respect to the next one. Constraints (2.6) avoid the conflicts. Finally, the 0-1 character of the variables is given by (2.7).

Note1: It is assumed in the model that an aircraft can climb and descend without modifying the speed, by an increase and reduction of power, respectively.

Note2: The integrality condition of variable ν_i^f can be relaxed (i.e., let $\nu_i^f \in \mathbb{R}^+$) since it only appears in constraint (2.5), where it is forced to be $\nu_i^f \geqslant 1$ if and only if $\phi_{i,h}^f = 1$ and $\phi_{i+1,h}^f = 0$, and zero otherwise, since it is penalized in the objective function and so it must take on the smallest possible value.

2.3 Collision Avoidance via altitude level and speed changes

2.3.1 Definitions

Hereafter we expand the model presented in Subsection 2.2.2 to take also into account speed changes. To that end, the following additional parameters and variables are defined.

Parameters

$\underline{t}_i^f$ **and** $\overline{t}_i^f$, lower and upper bounds for the feasible time instant at which aircraft f traverses the route segment $i \to (i+1)$, respectively, $\forall f \in \mathcal{F}, i \in \mathcal{W}_f^-$.

$s_{i,j}^{f,k}$, reward for avoiding the conflicts between the aircraft f and k at the waypoints i and j due to time coincidence, $\forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k}$.

Variables

τ_i^f, nonnegative rational variable that represents the time instant at which aircraft f transverses waypoint i, $\forall f \in \mathcal{F}, i \in \mathcal{W}_f$.

$\gamma_{i,j}^{f,k}$, 0-1 variable that takes on the value 1 if there is no conflict between the aircraft f and k at the waypoints i and j due to the timing (and, so, independently at which altitude level they traverse their respective waypoints) and 0, otherwise, $\forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k}$.

$\beta_{i,j}^{f,k}$, instrumental 0-1 variable, it will take on the value 1 if aircraft k arrives at point j before aircraft f arrives at point i, and it zero otherwise.

2.3.2 Conflict Detection

As we mention in Subsection 2.2.1, although the scheme for conflict detection is very similar for the both models that we propose in this work, there are some differences. Let the following slight modification: For a pair of aircraft $(f,k) \in \mathcal{F} \times \mathcal{F}$, there is a *potential conflict* at the pair of waypoints $(i,j) \in \mathcal{W}_f \times \mathcal{W}_k$ if both the conditions 1 and 3 stated in Section 2.2.1 hold and, instead of condition 2, the following one holds too:

- The time instants are such that $t_1^f + \sum_{i'<i} \underline{t}_{i'}^f < t_1^k + \sum_{j' \leqslant j} \overline{t}_{j'}^k$ and $t_1^f + \sum_{i' \leqslant i} \overline{t}_{i'}^f > t_1^k + \sum_{j'<j} \underline{t}_{j'}^k$. In order to justify this condition, suppose, on the contrary, that e.g., the second inequality does not hold, then even if aircraft k reaches waypoint j the soonest possible time instant, aircraft f is at its waypoint $i+1$, at least, and no conflict between the aircraft k and f is possible at the pair of waypoints (i,j).

Similarly to the problem with altitude level changes only, there is a *current conflict* for a pair of aircraft $(f,k) \in \mathcal{F} \times \mathcal{F}$ at the pair of waypoints $(i,j) \in \mathcal{W}_f \times \mathcal{W}_k$ if $(i,j) \in \mathcal{P}^{f,k}$, $t_i^f < t_{j+1}^k$, $t_j^k < t_{i+1}^f$ and $z_i^f = z_j^k$.

2.3.3 Model formulation for conflict resolution

As in the pure 0-1 model, the first term in the objective function rewards the aircraft that do not change their scheduled altitude level, the second term penalizes the number of "jumps" (climbing or descending) of the aircraft taken into consideration and the third term rewards the number of conflict resolutions by avoiding time coincidence. Notice that the model presented in Section 2.2.2 does only consider the first two terms.

The model is as follows,

$$\max \sum_{f\in\mathcal{F},i\in\mathcal{W}_f,h=z_i^f} c_i^f \phi_{i,h}^f - \sum_{f\in\mathcal{F},i\in\mathcal{W}_f^-} h_i^f \nu_i^f + \sum_{\forall f\in\mathcal{F},k\in\mathcal{F}^f,(i,j)\in\mathcal{P}^{f,k}} s_{i,j}^{f,k}\gamma_{i,j}^{f,k} \tag{2.8}$$

subject to constraints (2.2)-(2.5) and

$$\tau_1^f - t_1^f \leqslant \mu \quad \forall f\in\mathcal{F} \tag{2.9}$$

$$t_1^f - \tau_1^f \leqslant \mu \quad \forall f\in\mathcal{F} \tag{2.10}$$

$$\tau_{i+1}^f - \tau_i^f \leqslant \overline{t}_i^f \quad \forall f\in\mathcal{F}, i\in\mathcal{W}_f^- \tag{2.11}$$

$$\tau_{i+1}^f - \tau_i^f \geqslant \underline{t}_i^f \quad \forall f\in\mathcal{F}, i\in\mathcal{W}_f^- \tag{2.12}$$

$$\tau_{|\mathcal{W}_f|}^f - t_{|\mathcal{W}_f|}^f \leqslant \epsilon \quad \forall f\in\mathcal{F} \tag{2.13}$$

$$t_{|\mathcal{W}_f|}^f - \tau_{|\mathcal{W}_f|}^f \leqslant \epsilon \quad \forall f\in\mathcal{F} \tag{2.14}$$

$$\gamma_{i,j}^{f,k} \leqslant \frac{(\tau_i^f - \tau_j^k)}{m_{A_{i,j}^{f,k}}} + m_{i,j}^{f,k}\beta_{i,j}^{f,k} \quad \forall f\in\mathcal{F}, k\in\mathcal{F}^f, \quad (i,j)\in\mathcal{P}^{f,k} \tag{2.15}$$

$$\gamma_{i,j}^{f,k} \leqslant \frac{(\tau_j^k - \tau_i^f)}{m_{A_{i,j}^{f,k}}} + m_{i,j}^{f,k}(1-\beta_{i,j}^{f,k}) \quad \forall f\in\mathcal{F}, k\in\mathcal{F}^f, \quad (i,j)\in\mathcal{P}^{f,k} \tag{2.16}$$

$$\phi_{i,h}^f + \phi_{j,h}^k \leqslant 1 + \gamma_{i,j}^{f,k} \quad \forall f\in\mathcal{F}, k\in\mathcal{F}^f, \quad (i,j)\in\mathcal{P}^{f,k}, \quad h\in\mathcal{L}_{\mathcal{W}_f}^f \cap \mathcal{L}_{\mathcal{W}_k}^k \tag{2.17}$$

$$\tau_i^f \in \mathbb{R}^+ \quad \forall f\in\mathcal{F}, i\in\mathcal{W}_f^- \tag{2.18}$$

$$\phi_{i,h}^f, \nu_i^f \in \{0,1\} \quad \forall f\in\mathcal{F}, i\in\mathcal{W}_f, \quad h\in\mathcal{L}_i^f \tag{2.19}$$

$$\gamma_{i,j}^{f,k}, \beta_{i,j}^{f,k} \in \{0,1\} \quad \forall f\in\mathcal{F}, k\in\mathcal{F}^f, \quad (i,j)\in\mathcal{P}^{f,k}, \tag{2.20}$$

where the parameter ϵ in constraints (2.13) and (2.14) is half the length of the time interval around the scheduled arrival time. Its purpose is to avoid to constrain the aircraft arrival time to an isolated value. The aim of this requirement is to avoid changing scheduled flight times in other air zones, which could lead to new conflicts where they had previously been avoided. The parameter μ in constraints (2.9)

and (2.10) is half the length of the time interval around the scheduled ”departure” time. It will allow a small margin to decide when the aircraft fly into the conflict zone. The parameter $m_{i,j}^{f,k}$ in constraints (2.15) and (2.16) is the smallest possible value, big enough to guarantee that the right-hand-side of both constraints is positive, since their left-hand-side is a 0-1 variable.

Constraints (2.9) and (2.10) set the initial time instant for the aircraft to arrive to the conflict zone. Constraints (2.11) and (2.12) ensure “soft” speed changes. Constraints (2.13) and (2.14) force the aircraft to arrive at their destination waypoints at (almost) their previously assigned time instant. Constraints (2.17) avoid the conflicts together with the auxiliary constraints (2.15) and (2.16), whose purpose is to force the variables $\gamma_{i,j}^{f,k}$ to be zero if aircraft f and k traverse the waypoints i and j, respectively, within a small time interval (i.e., the difference of their time instants be smaller than $m_{A_{i,j}^{f,k}}$). Finally, constraints (2.18)-(2.20) define the character of the variables.

Note 1: As in the pure 0-1 model, the integrality condition of variable ν_i^f can be relaxed (i.e., let $\nu_i^f \in \mathbb{R}^+$), as it can be done with variable $\gamma_{i,j}^{f,k}$ for similar reasons.

Note 2: It is assumed in the model that in case of requiring e.g., a speed increase due to a latitude level change the issue can be addressed via a carefully choice of the parameters of the model, particularly the speed bounds to allow a speed in accordance with the altitude level change.

2.3.4 Tightening the model

Reducing the parameter $m_{i,j}^{f,k}$.

The easiest candidate for the parameter would be the total time considered in the problem, but a tighter candidate can be calculated as follows,

$$m_{i,j}^{f,k} = \frac{\max\left\{|\sum_{s<i}\overline{t}_s^f - \sum_{t<j}\underline{t}_t^k|, |\sum_{s<i}\underline{t}_s^f - \sum_{t<j}\overline{t}_t^k|\right\}}{m_{A_{i,j}^{f,k}}} + 1. \tag{2.21}$$

Again, we can even reduce $m_{i,j}^{f,k}$ by taking into account that so far, the aircraft are forced to arrive at their destination waypoints at their assigned arrival time instants. Then, let us use in expression (2.21) the following formulae: $inf\left\{\sum_{s<i}\overline{t}_s^f, t_{|\mathcal{W}_f|}^f - \sum_{s\geqslant i}\underline{t}_s^f\right\}$ and $sup\left\{\sum_{s<i}\underline{t}_s^f, t_{|\mathcal{W}_f|}^f - \sum_{s\geqslant i}\overline{t}_s^f\right\}$ instead of $\sum_{s<i}\overline{t}_s^f$ and$\sum_{s<i}\underline{t}_s^f$, respectively. Similarly, we can replace $\sum_{t<j}\overline{t}_t^k$ and $\sum_{t<j}\underline{t}_t^k$ with analogous expressions.

Special set of constraints

The above model for collision avoidance via altitude level and speed changes just presented above can be tightened by appending the constraints

$$\beta_{i,j}^{f,k} \leqslant \beta_{i,j-1}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.22}$$

$$\beta_{i,j}^{f,k} \leqslant \beta_{i+1,j}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.23}$$

$$\beta_{i,j}^{f,k} = 1 - \beta_{j,i}^{k,f} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.24}$$

$$\gamma_{i,j}^{f,k} + \beta_{i,j}^{f,k} \leqslant 1 + \gamma_{i,j-1}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.25}$$

$$\gamma_{i,j}^{f,k} \leqslant \beta_{i,j}^{f,k} + \gamma_{i-1,j}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.26}$$

Constraints (2.22)-(2.26) tighten the model allowing an important improvement in the computing time needed for solving the problem. Let us enlighten their "meaning":

(2.22), (2.23): if aircraft k arrives at point j before aircraft f arrives at point i, then it must arrive at point $j-1$ earlier and so $\beta_{i,j-1}^{f,k} = 1$. Likewise, aircraft f will arrive at $i+1$ later, so $\beta_{i+1,j}^{f,k} = 1$

(2.24)**:** if aircraft k arrives at point j before aircraft f arrives at point i (i.e. $\beta_{i,j}^{f,k} = 1$) then it cannot be true the opposite (i.e. $\beta_{j,i}^{k,f} = 1$).

(2.25)**:** if aircrafts f and k do not coincide on time at points i and j respectively (i.e. $\gamma_{i,j}^{f,k} = 1$), and aircraft k arrives at point j before aircraft f arrives at point i (i.e. $\beta_{i,j}^{f,k} = 1$), then aircrafts f and k do not coincide on time at points i and $j-1$ either (i.e. $\gamma_{i-1,j}^{f,k} = 1$).

(2.26)**:** if aircrafts f and k do not coincide on time at points i and j respectively (i.e. $\gamma_{i,j}^{f,k} = 1$), and aircraft f arrives at point i before aircraft k arrives at point j (i.e. $\beta_{i,j}^{f,k} = 0$), then aircrafts f and k do not coincide on time at points $i-1$ and j either (i.e. $\gamma_{i-1,j}^{f,k} = 1$).

We have tried too with an alternative series of constraints:

$$\gamma_{i,j}^{f,k} = \gamma_{i+1,j}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.27}$$

$$\gamma_{i,j}^{f,k} = \gamma_{i,j+1}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.28}$$

$$\gamma_{i,j}^{f,k} = \gamma_{i-1,j}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k} \tag{2.29}$$

$$\gamma_{i,j}^{f,k} = \gamma_{i,j-1}^{f,k} \quad \forall f \in \mathcal{F}, k \in \mathcal{F}^f, (i,j) \in \mathcal{P}^{f,k}. \tag{2.30}$$

Constraints (2.27)-(2.30) actually reduce the LP feasible space, while exclude some non optimal 0-1 solutions, what produces a much tighter model and, then, allows to obtain a smaller elapsed time for solving the problem. To understand their meaning and the reason based on which the excluded 0-1 solutions are not optimal, let us recall first how the variables $\gamma_{i,j}^{f,k}$ work. If the waypoints i and j are too close, the conflict between the aircraft f and k is avoided, since the time instant at which each aircraft traverses the respective waypoint are sufficiently distant, then $\gamma_{i,j}^{f,k} = 1$ and zero, otherwise. So, the above constraints force to avoid a particular set of possible conflicts between the two aircraft f and k (i.e., conflicts in consecutive waypoints), by one and only one of the possible manoeuvres, i.e. changing the altitude level or the speed. For an illustration, consider the situation depicted in Figure 2.1, and suppose that $(i_4, k_4), (i_4, k_5), (i_4, k_6) \in \mathcal{P}^{1,3}$, then if e.g., the potential conflict in (i_4, k_4) is avoided by delaying aircraft 3 so that both aircraft 1 and 3 do not coincide on time at that waypoint, then the potential conflict in (i_4, k_5) should be avoided taken advantage of such delay and not forcing a new maneuver, e.g., forcing aircraft 1 to descend an altitude level.

2.4 Computational experience

We report the results of the computational experience obtained while optimizing the pure 0-1 model and the mixed 0-1 model presented in sections 2.2.2 and 2.3.3, respectively. The models have been implemented in a c++ experimental code and have been optimized by using the sate-of-the-art engine CPLEX v12.1. The computations were carried out in a PC Intel Core 2 Duo 4, 2 GHz and 2 Gbytes of RAM.

Two sets of testbeds of randomly generated instances have been used in our experimentation, 24 instances for the first testbed and 25 instances for the the second one. For each instance 10 simulations have been performed, such that the averages of the computational results are reported. The simulations differ one from the other for each instance in (1) the conflict zone and (2) the arrival time instances of the aircraft (chosen at random throughout a uniform distribution) to the conflict zone along the time horizon through any of the four sides of the conflict zone (all of them with equal probability) and any waypoint of the sides (we have used a normal distribution with a standard deviation equal to 1). A random number of potential altitude levels ranges between 1 and 8 per aircraft.

The second term in the objective function (2.1) has been used for the pure 0-1 model (i.e., minimizing the number of altitude level changes). The constraints (2.22)-(2.26) have been also appended to the mixed 0-1 model. The objective function (2.8) has the following parameters in the testbed: $c_i^f = 0$, $h_i^f = 1$ and $s_{i,j}^{f,k} = 10$, so, the number of altitude levels is minimized and the number of conflict resolutions by speed changing is strongly maximized.

Tables 2.1 and 2.2 show the problem dimensions in the 24 instances in the testbed for the pure 0-1 model and the 25 instances in the testbed for the mixed 0-1 model. The headings are as follows: $|\mathcal{F}|$, number of aircraft; *CZ*, conflict zone side length (in nautical miles); $|\mathcal{T}|$, time horizon (in secs.); $|\bigcup_{f\in\mathcal{F}} C\mathcal{F}^f|$, number of *current aircraft conflicts*; $|\bigcup_{f\in\mathcal{F}} \mathcal{F}^f|$, number of *potencial aircraft conflicts*; $|\bigcup_{f\in\mathcal{F},k\in C\mathcal{F}^f} C\mathcal{P}^{f,k}|$, number of *current waypoint conflicts*; and $|\bigcup_{f\in\mathcal{F},k\in\mathcal{F}^f} \mathcal{P}^{f,k}|$, number of *potencial waypoint conflicts*. We can observe that the number of aircraft, conflict zone side length and time horizon have realistic dimensions.

The number of conflicts that took place in the simulations for each instance has been measured in 4 different ways, namely, the number of *current aircraft conflicts*, the number of *potential aircraft conflicts*, the number of *current waypoint conflicts*, and the number of *potential waypoint conflicts*.

Tables 2.3 and 2.4 show the dimensions of the pure 0-1 and mixed 0-1 models, respectively.

The headings are as follows: *m* and m^*, number of constraints before and after CPLEX preprocessing, respectively; rm: ratio (in %) between m and m^* (i.e., $\frac{m^* \cdot 100}{m}$); *n01* and *nc*, number of 0-1 and continuous variables, respectively; *n* and n^*, number of variables before and after CPLEX preprocessing, respectively; rn, ratio (in %) between n and n^* (i.e., $\frac{n^* \cdot 100}{n}$). We can observe in these tables how high are the dimensions of the models.

Tables 2.5 and 2.6 report the computational results. The headings are as follows: z_{lp}, solution value of the LP relaxation; z_s, solution value of the stronger LP relaxation (i.e., the value of the LP model after appending the cuts identified by CPLEX); z_{ip}, solution value of the original CDR problem; GAP_{lp} and GAP_s, related optimality gaps computed as $\frac{z_{ip}-z_{lp}}{z_{ip}}\%$ and $\frac{z_{ip}-z_s}{z_{ip}}\%$, respectively; *nn*, number of CPLEX branch-and-cut nodes; t_{lp}, t_s and t_{ip}, elapsed times (secs.) to obtain the solution values z_{lp}, z_s and z_{ip}, respectively; t_t, total elapsed time from the starting of the optimization; *nc*, total number of cuts identified and appended by CPLEX.

Note: Some results for the pure 0-1 model, namely z_{lp}, z_s, z_{ip}, GAP_{lp}, GAP_s and *nn*, have not been included in Table 2.5, since they are zero in all instances of the testbed. Additionally, the model is so tight that the LP solution gives integer values for the (0-1) variables and then, the CPLEX branch-and-cut phase is not been required in any of the instances, being the total elapsed time close to zero in 21 out of 24 instances, and very small for the other three remaining instances.

Finally, it is worthy to point out the impressively good total times t_t (in secs.) that have been required for providing the optimal solution of the mixed 0-1 models, see Table 2.6.

2.5 Further discussions and extensions for the proposed models

At this section we discuss some thoughts on the *applicability* of our models and propose some extensions to the proposed models.

2.5.1 On ascending or descending flight levels

We are assuming as a hypothesis that aircraft are exactly at a given flight level while flying through each waypoint, so if an aircraft climbs one level from a given waypoint, say a, to the next one, say b, the ascension must be executed completely while flying the *segment* from a to b. Thus e.g. given a segment 2 miles long, the aircraft must be able to ascend or descend 1000 ft. (the

Table 2.1: Dimensions of the altitude level change problem for model 2.1-2.7

Case	$\lvert\mathcal{F}\rvert$	CZ	$\lvert\mathcal{T}\rvert$	$\lvert\bigcup_{f\in\mathcal{F}} C\mathcal{F}^f\rvert$	$\lvert\bigcup_{f\in\mathcal{F}} \mathcal{F}^f\rvert$	$\lvert\bigcup_{f\in\mathcal{F},k\in C\mathcal{F}^f} C\mathcal{P}^{f,k}\rvert$	$\lvert\bigcup_{f\in\mathcal{F},k\in\mathcal{F}^f} \mathcal{P}^{f,k}\rvert$
p01	25	50	300	15	43	36	270
p02	25	50	600	27	70	79	691
p03	25	100	300	8	20	29	177
p04	25	100	600	12	34	40	345
p05	25	200	600	5	12	18	145
p06	50	200	900	22	45	100	908
p07	50	200	1800	20	67	68	1295
p08	50	200	3600	18	77	65	1650
p09	50	400	1800	10	25	50	681
p10	50	400	3600	12	49	52	1301
p11	65	200	900	36	80	138	1338
p12	65	200	1800	37	125	132	2361
p13	65	200	3600	31	124	107	2588
p14	65	400	1800	20	49	89	1208
p15	65	400	3600	18	69	79	1861
p16	75	200	900	49	100	200	1826
p17	75	200	1800	46	168	187	3026
p18	75	200	3600	39	171	122	3398
p19	75	400	1800	26	58	125	1458
p20	75	400	3600	25	98	98	2471
p21	100	400	3600	43	177	173	4433
p22	100	600	3600	30	93	146	2682
p23	200	400	1800	195	463	868	11610
p24	200	400	3600	163	673	693	17665

Table 2.2: Dimensions of the altitude level and speed changes problem for model 2.8-2.20

Case	$\lvert\mathcal{F}\rvert$	CZ	$\lvert\mathcal{T}\rvert$	$\lvert\bigcup_{f\in\mathcal{F}} C\mathcal{F}^f\rvert$	$\lvert\bigcup_{f\in\mathcal{F}} \mathcal{F}^f\rvert$	$\lvert\bigcup_{f\in\mathcal{F},k\in C\mathcal{F}^f} C\mathcal{P}^{f,k}\rvert$	$\lvert\bigcup_{f\in\mathcal{F},k\in\mathcal{F}^f} \mathcal{P}^{f,k}\rvert$
m01	10	50	300	2	7	6	48
m02	10	50	600	3	9	8	75
m03	10	100	300	1	3	3	21
m04	10	100	600	1	4	5	51
m05	10	200	600	1	2	4	40
m06	20	50	300	9	27	20	162
m07	20	50	600	17	46	48	406
m08	20	100	300	6	13	19	109
m09	20	100	600	6	15	24	203
m10	20	200	600	4	7	16	104
m11	25	50	300	15	43	36	270
m12	25	50	600	27	70	79	691
m13	25	100	300	8	20	29	177
m14	25	100	600	12	34	40	345
m15	25	200	600	5	12	18	145
m16	50	200	900	22	45	100	908
m17	50	200	1800	20	67	68	1295
m18	50	200	3600	18	77	65	1650
m19	50	400	1800	10	25	50	681
m20	50	400	3600	12	49	52	1301
m21	75	200	900	49	100	200	1826
m22	75	200	1800	46	168	187	3026
m23	75	200	3600	39	171	122	3398
m24	75	400	1800	26	58	125	1458
m25	75	400	3600	25	98	98	2471

Table 2.3: Dimensions of the pure 0-1 model 2.1-2.7

Case	m	m*	rm(%)	n	n*	rn(%)
p01	3052	1493	48.9	1081	537	49.7
p02	5222	3758	72.0	1735	1264	72.9
p03	2265	1906	84.2	844	714	84.6
p04	3275	2385	72.8	1231	909	73.8
p05	2007	1428	71.2	787	562	71.4
p06	8634	7431	86.1	3115	2675	85.9
p07	6876	6596	95.9	2631	2509	95.4
p08	6668	5771	86.5	2564	2211	86.2
p09	5558	5425	97.6	2095	2035	97.1
p10	5071	4941	97.4	1941	1883	97.0
p11	11648	11376	97.7	4164	4048	97.2
p12	13082	12788	97.8	4777	4648	97.3
p13	10396	10024	96.4	3963	3799	95.9
p14	9240	8242	89.2	3427	3037	88.6
p15	6867	6698	97.5	2595	2520	97.1
p16	15032	14668	97.6	5296	5140	97.1
p17	16359	15942	97.5	6007	5828	97.0
p18	13847	13592	98.2	5178	5064	97.8
p19	11848	11529	97.3	4400	4262	96.9
p20	10838	9440	87.1	4054	3524	86.9
p21	17117	16583	96.9	6406	6179	96.5
p22	14893	14519	97.5	5519	5353	97.0
p23	67260	65831	97.9	22654	22058	97.4
p24	33092	32789	99.1	11681	11547	98.9

Table 2.4: Dimensions of the mixed 0-1 model 2.8-2.20

Case	m	m*	rm(%)	n01	nc	n	n*	rn(%)
m01	960	746	77.7	353	51	404	313	77.5
m02	1440	1064	73.9	503	71	574	424	73.9
m03	562	453	80.6	219	36	255	208	81.6
m04	1056	796	75.4	367	53	420	319	76.0
m05	957	795	83.1	333	51	384	318	82.8
m06	2783	2085	74.9	1002	136	1138	839	73.7
m07	6039	4215	69.8	2009	239	2248	1508	67.1
m08	2347	1907	81.3	875	131	1006	812	80.7
m09	3971	3064	77.2	1364	192	1556	1189	76.4
m10	2192	1665	76.0	793	117	910	699	76.8
m11	4570	3475	76.0	1612	214	1826	1361	74.5
m12	9918	7012	70.7	3204	365	3569	2412	67.6
m13	3295	2572	78.1	1207	171	1378	1054	76.5
m14	5976	4381	73.3	2054	273	2327	1697	72.9
m15	3265	2600	79.6	1181	178	1359	1084	79.8
m16	15897	12094	76.1	5339	705	6044	4539	75.1
m17	21321	14073	66.0	6920	866	7786	5289	67.9
m18	27032	17451	64.6	8584	1057	9641	6472	67.1
m19	12690	9302	73.3	4262	580	4842	3614	74.6
m20	21820	13947	63.9	7015	883	7898	5450	69.0
m21	29244	21485	73.5	9725	1215	10940	7891	72.1
m22	47939	32576	68.0	15320	1854	17174	11780	68.6
m23	55906	36855	65.9	17455	2132	19587	13376	68.3
m24	26502	19513	73.6	8847	1186	10033	7465	74.4
m25	41860	27504	65.7	13194	1651	14845	10268	69.2

standard vertical distance between flight levels) i.e., fly with a climbing angle α such that $tg(\alpha) = 0.189/2 \Rightarrow \alpha \approx \arctan(.095) \approx 0.094$ radians, or else 5.42^{o}, what seems reasonable. But situations may present in which, for a particular aircraft, it would take more to ascend or descend a flight level than the distance from a to b, (e.g., we may need to use much shorter segments, e.g. the aircraft might difficulties to ascend 1000 ft. in half a mile, at an angle of $\arctan(0.189 * 2) \approx 0.36rad = 20.70^{\text{o}}$).

There is an additional problem: our constraints guarantee that a particular pair of aircrafts avoid all conflicts by flying at different flight levels in the waypoints, but how can we guarantee that they do not come into a conflict between those points while one is descending and the other one is

ascending. We will discuss this issue below and show how to solve the problem and how to relax the hypothesis.

Ascending (descending) in more than one step:

We can take into account that it may take two or more steps to ascend (descend) a flight level, by doing what follows:

re-define the variables $\phi^f_{i,h}$ as $\phi^f_{i,h} = 1$ if aircraft f traverses point i at a height between flight levels $h-1$ and $h+1$. And constraints (2.2)-(2.4) could be replaced by some constraints that took into account some "steps" for changing the altitude level, e.g., let us propose the following set of constraints so that it takes up to two *segments* to ascend or descend a level:

Table 2.5: Computational results for the pure 0–1 model 2.1-2.7

Case	t_{lp}	t_s	t_{ip}	t_t	nc
p01	< 0.01	0	< 0.01	< 0.01	< 0.01
p02	< 0.01	< 0.01	< 0.01	< 0.01	15
p03	< 0.01	< 0.01	< 0.01	< 0.01	1
p04	< 0.01	< 0.01	< 0.01	< 0.01	19
p05	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
p06	< 0.01	< 0.01	< 0.01	< 0.01	44
p07	< 0.01	< 0.01	< 0.01	< 0.01	72
p08	< 0.01	< 0.01	< 0.01	< 0.01	166
p09	< 0.01	< 0.01	< 0.01	< 0.01	1
p10	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
p11	< 0.01	< 0.01	< 0.01	< 0.01	38
p12	< 0.01	< 0.01	< 0.01	< 0.01	10
p13	< 0.01	< 0.01	< 0.01	< 0.01	5
p14	< 0.01	< 0.01	< 0.01	< 0.01	54
p15	< 0.01	< 0.01	< 0.01	< 0.01	4
p16	< 0.01	1	1	1	80
p17	< 0.01	1	< 0.01	< 0.01	14
p18	< 0.01	< 0.01	< 0.01	< 0.01	37
p19	< 0.01	< 0.01	< 0.01	< 0.01	16
p20	< 0.01	< 0.01	< 0.01	< 0.01	19
p21	< 0.01	< 0.01	< 0.01	< 0.01	6
p22	< 0.01	< 0.01	1	1	80
p23	2	18	15	18	311
p24	< 0.01	4	3	4	58

$$\sum_{h\in\mathcal{L}_i^f} \phi_{i,h}^f \leqslant 2 \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f \tag{2.31}$$

$$\phi_{i-1,h}^f \leqslant \sum_{\ell=-1}^{1} \phi_{i+1,h+\ell}^f \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f^{'-} \quad h \in \mathcal{L}_i^f \tag{2.32}$$

$$\phi_{i+1,h}^f \leqslant \sum_{\ell=-1}^{1} \phi_{i-1,h-\ell}^f \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f^{'} \quad h \in \mathcal{L}_i^f \tag{2.33}$$

$$\phi_{i-1,h}^f + \phi_{i+1,h-1}^f \leqslant \phi_{i,h}^f + \phi_{i,h-1}^f \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f^{'-} \quad h \in \mathcal{L}_i^f \tag{2.34}$$

$$\phi_{i-1,h}^f + \phi_{i+1,h+1}^f \leqslant \phi_{i,h}^f + \phi_{i,h+1}^f \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f^{'-} \quad h \in \mathcal{L}_i^f \tag{2.35}$$

$$\phi_{i-1,h}^f + \phi_{i+1,h}^f \leqslant \phi_{i,h}^f + 1 \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f^{'-} \quad h \in \mathcal{L}_i^f \tag{2.36}$$

Table 2.6: Computational results for the mixed 0-1 model 2.8-2.20

Case	z_{lp}	z_s	z_{ip}	$GAP_{lp}(\%)$	$GAP_s(\%)$	nn	t_{lp}	t_s	t_{ip}	t_t	nc
m01	442.02	25.00	25.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m02	702.86	44.00	44.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m03	197.56	20.00	20.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m04	486.63	50.00	50.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m05	387.03	31.00	31.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m06	1490.52	58.00	58.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	6
m07	3892.76	258.00	258.00	2080.65	0.00	0	< 0.01	< 0.01	< 0.01	< 0.01	10
m08	1005.62	33.00	33.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m09	1929.84	210.00	210.00	1405.56	0.00	0	< 0.01	< 0.01	< 0.01	< 0.01	14
m10	962.81	56.00	-56.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m11	2502.83	94.00	94.00	3090.24	0.00	0	< 0.01	< 0.01	< 0.01	< 0.01	2
m12	6635.38	544.00	544.00	1132.49	0.00	0	< 0.01	1	< 0.01	< 0.01	56
m13	1601.03	89.00	89.00	3297.22	0.00	0	< 0.01	< 0.01	< 0.01	< 0.01	0
m14	3254.44	437.00	437.00	772.58	0.00	0	< 0.01	< 0.01	< 0.01	< 0.01	22
m15	1375.16	169.00	169.00	-	-	0	< 0.01	< 0.01	< 0.01	< 0.01	53
m16	8628.41	1244.65	1241.00	779.14	0.20	4	< 0.01	< 0.01	1	1	133
m17	12670.84	4914.69	4821.00	204.00	1.65	3	< 0.01	1	< 0.01	< 0.01	85
m18	16327.43	8069.55	7578.00	126.69	4.81	141	< 0.01	1	5	5	169
m19	6555.91	1993.00	1993.00	447.78	0.00	0	< 0.01	< 0.01	< 0.01	< 0.01	67
m20	12816.86	7624.00	7259.00	76.33	3.96	117	< 0.01	1	2	3	83
m21	17517.54	3214.14	3213.00	514.68	0.04	0	< 0.01	2	2	3	307
m22	29704.67	12036.53	11348.00	172.10	5.80	435	< 0.01	5	24	24	677
m23	33618.61	18632.84	17065.90	104.55	9.41	482	< 0.01	4	30	31	627
m24	14087.53	4059.69	3985.00	308.12	3.63	3	< 0.01	1	1	1	108
m25	24435.39	13424.85	12601.00	104.32	6.22	173	< 0.01	2	9	9	338

Where (2.31) allows for a given aircraft f to traverse a given route point i flying between two flight levels, and constraints (2.32) - (2.36) are intended to guarantee "soft" level changes.

2.5.2 Rerouting

Here we present an idea for allowing the aircraft choosing alternative routes to avoid conflicts.

First, let the following additional parameters:

$\mathcal{R}^f$, set of possible routes to follow for aircraft $f, \forall f \in \mathcal{F}$.

$\mathcal{W}_n^f$, set of ordered route points for aircraft f and route $n, \forall f \in \mathcal{F}, \forall n \in \mathcal{R}^f$.
$\mathcal{W}_n^{f-} = \mathcal{W}_n^f \setminus \{|\mathcal{W}_n^f|\}$ (all the points but the last one)
$\mathcal{W}_n^{f'} = \mathcal{W}_n^f \setminus \{1\}$ (all the points but the first one)
$\mathcal{W}_n^{f'-} = \mathcal{W}_n^f \setminus \{1, |\mathcal{W}_n^f|\}$ (all the points but the first and the last ones)

cr_n^f, cost for using the route $n, \forall f \in \mathcal{F}, n \in \mathcal{R}^f$.

and the following variables:

ρ_n^f, 0-1 variable such that its value is 1 if the aircraft f follows the route $n \in \mathcal{R}^f$ and, otherwise, it is zero, $\forall f \in \mathcal{F}, \forall n \in \mathcal{R}^f$.

Finally, we propose the following model:

$$\min \sum_{f\in\mathcal{F}, n\in\mathcal{R}^f, i\in\mathcal{W}_n^{f-}, h=z_i^f} c_1^f \cdot \phi_{i,h}^f + \sum_{f\in\mathcal{F}, n\in\mathcal{R}^f, i\in\mathcal{W}_n^f, h\in\mathcal{L}_i^f} h_{i,h}^f \cdot \nu_i^f + \sum_{f\in\mathcal{F}, n\in\mathcal{R}^f} cr_n^f \cdot \rho_n^f \quad (2.37)$$

subject to:

$$\sum_{n\in\mathcal{R}^f} \rho_n^f = 1 \quad \forall f \in \mathcal{F} \tag{2.38}$$

$$\sum_{h\in\mathcal{L}_i^f} \phi_{i,h}^f = \rho_n^f \quad \forall f \in \mathcal{F}, \quad n \in \mathcal{R}^f, \quad i \in \mathcal{W}_n^f \tag{2.39}$$

$$\phi_{i,h}^f \leqslant \sum_{\ell=\underline{V}_i^f}^{\overline{V}_i^f} \phi_{i+1,h+\ell}^f \quad \forall f \in \mathcal{F}, \quad n \in \mathcal{R}^f, \quad i \in \mathcal{W}_n^{f-}, \quad h \in \mathcal{L}_i^f \tag{2.40}$$

$$\phi_{i,h}^f \leqslant \sum_{\ell=\underline{V}_{i-1}^f}^{\overline{V}_{i-1}^f} \phi_{i-1,h-\ell}^f \quad \forall f \in \mathcal{F}, \quad n \in \mathcal{R}^f, \quad i \in \mathcal{W}_n^{f'}, \quad h \in \mathcal{L}_i^f \tag{2.41}$$

$$\phi_{i,h}^f - \phi_{i+1,h}^f \leqslant \nu_i^f + (1-\rho_n^f) \quad \forall f \in \mathcal{F}, \quad n \in \mathcal{R}^f, \quad i \in \mathcal{W}_n^{f-}, \quad h \in \mathcal{L}_i^f \tag{2.42}$$

$$\phi_{i,h}^f + \phi_{j,h}^k \leqslant 1 \quad \forall f \in \mathcal{F}, k \in \mathfrak{CF}^f, (i,j) \in \mathfrak{CP}^{f,k} h \in \mathcal{L}_{\mathcal{W}_f}^f \cap \mathcal{L}_{\mathcal{W}_k}^k \tag{2.43}$$

$$\tag{2.44}$$

$$\phi_{i,h}^f \leqslant \rho_n^f \quad \forall f \in \mathcal{F}, \quad n \in \mathcal{R}^f, \quad i \in \mathcal{W}_n^f, \quad h \in \mathcal{L}_i^f \tag{2.45}$$

$$\phi_{i,h}^f, \nu_i^f, \rho_n^f \in \{0,1\} \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}_f, \quad h \in \mathcal{L}_i^f \tag{2.46}$$

The third term in the objective function (2.37), allows us to introduce preferences among routes.

Constraint (2.38) guarantees that every aircraft follows only one route. Constraint (2.39) is to ensure that all flights traverse every route point at only one flight level, for the route selected.

Constraint (2.39) could be replaced by $\sum_{h\in\mathcal{L}_i^f} \phi_{i,h}^f \leqslant 1 \quad \forall f \in \mathcal{F}, \quad n \in \mathcal{R}^f, \quad i \in \mathcal{W}_n^f$ (and constraints (2.38) and (2.45) can be removed to avoid using the variables ρ_n^f). Constraints (2.38) and (2.45) force to select only one route per aircraft. Both of them force variables $\phi_{i,h}^f$ to be 0 when the route they belong to is not selected. Using both constraints helps tightening the model. Finally, the integrality condition for variable ρ_n^f can be relaxed.

2.5.3 Arriving at a different time

Constraints (2.13)-(2.14) force the aircrafts to arrive at (next to) the assigned arrival time. But this may no be a requirement in some situations. In order to allow different arrival times, we could replace constraints (2.13)-(2.14), by the following ones:

$$\tau^f_{|\mathcal{W}_f|} - t^f_{|\mathcal{W}_f|} \leqslant \epsilon + \lambda^f_d \quad \forall f \in \mathcal{F} \tag{2.47}$$

$$t^f_{|\mathcal{W}_f|} - \tau^f_{|\mathcal{W}_f|} \leqslant \epsilon + \lambda^f_a \quad \forall f \in \mathcal{F} \tag{2.48}$$

Where variable $\lambda^f_d \in \mathbb{R}^+$ ($\lambda^f_a \in \mathbb{R}^+$) represent the number of seconds after (before) the arrival time that aircraft f arrives to point $\mathcal{W}_f$. These variables should be penalized in the objective function with some weighting factor. ϵ in these constraints is the threshold.

2.5.4 Changing flight level might imply changing speed

As we said above, ascending, descending or maintaining the flight level might imply flying at a different speed. To take this into account in the models, different lower bounds could be assigned to $\tau^f_{i+1} - \tau^f_i$ as a function of the changes of flight level state. In other words, constraints (2.11)-(2.12) could be replaced by the following ones:

$$(\phi^f_{i+1,h} + \phi^f_{i,h-1} - 1) \cdot \underline{A}^f_i \leqslant \tau^f_{i+1} - \tau^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \quad h \in \mathcal{L}^f_{i+1}, h-1 \in \mathcal{L}^f_i \tag{2.49}$$

$$(\phi^f_{i+1,h} + \phi^f_{i,h+1} - 1) \cdot \underline{D}^f_i \leqslant \tau^f_{i+1} - \tau^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \quad h \in \mathcal{L}^f_{i+1}, h+1 \in \mathcal{L}^f_i \tag{2.50}$$

$$(\phi^f_{i+1,h} + \phi^f_{i,h} - 1) \cdot \underline{C}^f_i \leqslant \tau^f_{i+1} - \tau^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \quad h \in \mathcal{L}^f_{i+1}, h \in \mathcal{L}^f_i \tag{2.51}$$

If the upper bounds were also needed, the following variables and constraints could be added to allow so:

α^f_i, 0-1 variable that takes on value 1 if aircraft f is ascending through segment $i \to (i+1)$.

δ^f_i, 0-1 variable that takes on value 1 if aircraft f is descending through segment $i \to (i+1)$.

ζ^f_i, 0-1 variable that takes on value 1 if aircraft f does not change its flight level through segment $i \to (i+1)$.
$\forall f \in \mathcal{F}, i \in \mathcal{W}^-_f$.

$$(\phi^f_{i+1,h} + \phi^f_{i,h-1} - 1) \leqslant \alpha^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \quad h \in \mathcal{L}^f_{i+1}, h-1 \in \mathcal{L}^f_i \tag{2.52}$$

$$(\phi^f_{i+1,h} + \phi^f_{i,h+1} - 1) \leqslant \delta^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \quad h \in \mathcal{L}^f_{i+1}, h+1 \in \mathcal{L}^f_i \tag{2.53}$$

$$(\phi^f_{i+1,h} + \phi^f_{i,h} - 1) \leqslant \zeta^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \quad h \in \mathcal{L}^f_{i+1}, h \in \mathcal{L}^f_i \tag{2.54}$$

$$\alpha^f_i + \delta^f_i + \zeta^f_i = 1 \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \tag{2.55}$$

$$\tau^f_{i+1} - \tau^f_i \leqslant \overline{A}^f_i \cdot \alpha^f_i + \overline{D}^f_i \cdot \delta^f_i + \overline{C}^f_i \cdot \zeta^f_i \quad \forall f \in \mathcal{F}, \quad i \in \mathcal{W}^-_f \tag{2.56}$$

Where (2.55) is redundant but it is intended to reinforce the model.

2.6 Conclusions

Two novel tight integer linear optimization models for Conflict Detection and Resolution in a set of aircraft in the airspace have been proposed. The first one is a pure 0-1 linear model which avoid conflicts by means of altitude changes, and the second one a mixed 0-1 linear model whose strategy is based on altitude and speed changes. The very small elapsed time for both models shows that they can be used in real time, particularly in the medium and long term, and in wide airspace regions.

The approach is quite flexible and can be applied in the future freely decided optimal trajectories in the context of ”Free Flight”.

Several extensions for improving the performance of both models can be proposed, particularly the possibility of selecting alternative routes and allowing aircraft climbing or descending to the next altitude level in more than one step as well as allowing to relate altitude level changes to speed. It is a subject of future research work.

Chapter 3

Introduction to Stochastic Programming and Parallel Computing: State of the art

The remainder of this thesis aims to contribute to the field of Stochastic Programming (SP). Uncertainty is the key ingredient in many decision problems and SP was created to approach it. However, SP problems are big sized and hard to solve. Here we present a sequential algorithm and a parallel one, based on **metaheuristics** approaches in order to help solving large scale optimization problems under uncertainty (SP problems).

Basic concepts of Stochastic Programming are introduced, as well as a brief state-of-the-art description of Stochastic Programming and current algorithms designed to solve stochastic mixed 0-1 multistage problems. Finally, since our approach involves Parallel Computing (PC), we outline a brief introduction to this field at the end of this chapter.

Chapter 4 presents the Fix-and-Relax Coordination (FRC) version we have implemented for this thesis and the FRC-J and FRC-PJ algorithms, which allow the resolution of large scale stochastic optimization problems by decomposing the original problem into smaller subproblems and coordinately solving them by keeping the so-called non-anticipativity principle. Since both algorithms are **metaheuristic**, optimality is not guaranteed. Finally, Chapter 5 summarizes the results of a broad computational experience that has been carried out using all these algorithms previously presented, for a particular problem: Multi-period location-allocation problem under uncertainty.

The remainder of this chapter is organized as follows: Section 3.1 outlines a brief state of the art on Stochastic Programming. Section 3.2 presents the fundamental concepts of modeling linear

problems under uncertainty. Scenario tree modeling is explained in Section 3.3. Section 3.4 shows the different ways of representing the Deterministic Equivalent Problem (DEM) and Section 3.5 shows the concepts of Expected Value of Perfect Information (EVPI) and Value of the Stochastic Solution (VSS). The main algorithms in Stochastic Linear Programming are presented in Section 3.6. And Sections 3.7 and 3.8 are devoted to Stochastic Integer Programming and its main known algorithms, respectively. The Branch-And-Fix (BFC) algorithm is treated separately in Section 3.9 . Finally, Section 3.11 introduces some basics concepts on Parallel Computing and the libraries that will be used for implementing the parallel algorithm presented in the next chapter.

3.1 Introduction

Optimization is a very useful mathematical decision aid tool and is widely applied in many different areas. Most traditional optimization models are deterministic. However, uncertainty is present in many real problems where some parameters cannot be known a priori. Since the 50's, it is well known that deterministic optimization is not appropriate for capturing the uncertain behavior present in most real situations. Very frequently, mainly in problems with a given time horizon to exploit, some coefficients in the objective function, the *right hand side* (rhs) vector and the constraint matrix are not known with certainty when the decisions have to be made, but some information is available. Financial planning, airline scheduling and production and distribution planning are just a few examples of areas in which ignoring uncertainty may lead to worse or even wrong decisions. Several ways to formalize this uncertainty have been studied leading to different approaches to solve stochastic optimization problems. However, it is not until the 1980s that SP is beeing broadly applied in real-world applications, with the help of new advances in computer technologies that allowed the solution of big size models. This increased the interest in SP, yielding new advances in mathematical theory. New problem formulations appear almost every year and this variety is one of the strengths of the field.

The need to incorporate uncertainty in mathematical programming models resulted in the field of SP which, basically, deals with mathematical programs in which some parameters are random variables. Early work of Beale [34] and Dantzig [69] started this field in 1955, followed by Charnes and Cooper in 1959 (see [60]). Their methods had their roots in statistical decision theory (Wald, 1950) [234], althoug SP focuses on methods of solution and analytical properties instead of constructing derivatives and updating probabilities.

The Deterministic Equivalent Problem (DEM) explained in Section 3.4 was coined by Wets [239], and its first solution was due to Benders [35], by using the so-called Benders Decomposition

(BD); see also Birge and Louveaux [45]; Laporte and Louveaux [150], among others. Sen and Sherali in [219] have proposed a decomposition algorithm based on a branch-and-cut approach to solve **two-stage** stochastic programs having first-stage pure 0-1 variables and 0-1 mixed-integer recourse variables, where a modified BD method is developed. Escudero et al. presented in [92, 93, 95] a general algorithm to solve **two-stage** stochastic mixed 0-1 problems.

In the **multi-stage** stochastic integer optimization problem, at each stage a decision has to be made. So, the decisions for a given stage yielded by the model cannot anticipate the information not yet available, i.e., the corresponding variables must take on the same value under each scenario for a given group of scenarios. In other words the **non-anticipativity** constraints (NAC) must be satisfied (see Wets [240] and Rockafellar and Wets [211]). NAC are further explained in Section 3.3.2.

Alonso-Ayuso et al. [14, 15] addressed **multi-stage** stochastic mixed 0-1 problems where both binary and continuous variables appear at any stage of the time horizon, and where uncertainty appears only in the objective function coefficients and the rhs.

Large-scale **multi-stage** Stochastic mixed 0-1 Problems are in general hard to solve, requiring large computing resources, and there are few approaches in the literature to solve up to optimality such problems where both binary and continuous variables, and uncertainty, appear anywhere in the model. A decomposition methodology (so-named **Branch-and-Fix Coordination (BFC)**), was introduced in Escudero [84] and further developed in Escudero et al. [94, 96, 97, 98, 89]. Moreover, a parallel computing version of the BFC algorithm has been presented recently in Pagès-Bernaus et al. [184]. Other decomposition methods have been proposed in Mulvey and Ruszczynski [177]; Vladimirou [233]; Blomval and Lindberg [48]; Blomval [47]. The Stochastic Dynamic Programming (SDP) **metaheuristic** introduced in Cristobal et al. [68] and Escudero et al. [99] yielded good results. In Dias et al. [76] several parallelization strategies are proposed aimed at speed up the stochastic dynamic programming solution.

The decomposition algorithms used to solve these kind of problems ease the job and, moreover, they can be parallelized allowing the use of greater computer forces. Part of this thesis will focus on this issue. Few papers have appeared on **Stochastic Programming** using parallel programming for stochastic continuous and mixed 0-1 programming, see Birge et al. [43]; Beraldi et al. [36]; Fragniere et al. [106]; Linderoth et al. [152]; Lucka et al. [154]; Al-Khamis and M'Hallah [6]; and Pagès-Bernaus et al. [184], among others.

Classical models present as the objective function the expected value of the objective function over the set of finite scenarios, i.e. the so called risk neutral (RN) approach. However, RN

solutions have the inconvenience of ignoring the variability of such objective function value over the scenarios. So, it does not hedge against the low-probability/high-consequence events (the so-called black swans). Alternatively, risk measures can be added in order to hedge against the impact of most unwanted scenarios. Risk measures are currently been taken into account by considering minimizing, for example, semi-deviations, excess of probability, conditional value-at-risk, expect shortfall and others. These approaches are more convenient under the presence of binary variables than the classical media-variance schemas.

Recently, new risk averse measures have appeared in the literature, e.g., the so-named first- and second-order Stochastic Dominance Constraint (SDC) strategies for a set of profiles, each one included by a threshold for a given function value and some types of shortfall related bounds on reaching it. See [115] and [114] for first-order and second-order SDC integer-linear recourse, respectively, and [89] for the mixture of both strategies in a multiperiod setting. In particular, the Time Stochastic Dominance (TSD) strategy reduces the risk of wrong solutions in a better way than some others under some circumstances, according to the computational comparison reported in e.g., [13]. The strategy also aims to minimize the objective function expected value, see also [90].

Other methodologies aimed at improving the decomposition algorithms currently used in SP solving are Lagrangian Decomposition procedures. Among the most recent approaches, see in [88] where a Multistage scenario Cluster Lagrangian Decomposition (MCLD) approach for obtaining strong lower bounds on the solution value of large sized instances of the multistage stochastic mixed 0-1 problem is presented. The MCLD procedure outperforms the traditional Lagrangian Decomposition scheme based on single scenarios in both the bound's quality and elapsed time.An scheme presented in [88], so-named Lagrangean Progressive Hedging Algorithm LPHA, has its roots in the seminal paper [209] (see also [238]) where PHA is described for the first time. In [91] a specialization of the so-called Cluster Lagrangean Decomposition for obtaining strong (lower) bounds on multistage stochastic (minimization) is applied to a facility location problem under uncertainty.

For a deeper understanding of SP the books [191, 45, 142, 164, 29] provide valuable resources.The survey papers by Schultz et al. [216], Sen [217] and Ph.D. theses by Stougie [223], van der Vlerk [229] and Carøe [55] among many others, are good resources on Stochastic Mixed Integer Programming (SMIP).

Internationally, there is a research community interested specifically in stochastic optimization which information can be found in **http://www.stoprog.org**. This group provides a repository of electronic papers with recent results named *Stochastic Programming E-Print Series*, (URL

http://edoc.hu-berlin.de/browsing/speps/) and information about international conferences, mainly the triennial international conferences on SP, the next edition will be the 14th one, June, 2016 in Buzios, Brazil.

3.2 Stochastic linear modeling

This section deals with the fundamental concepts of modeling mathematical programming problems under uncertainty. The remainder of this thesis is based on it.

3.2.1 Deterministic linear models

A deterministic Linear Programming (LP) problem consists of a set of linear constraints and a linear objective function, such that the problem solution must be subject to such linear constraints and take the optimum value for the objective function. In deterministic LP problems all parameters are considered certain (this is, a controllable model). Such a deterministic problem has the following expression:

$$\begin{aligned} \mathrm{Z} = \min\, & c_1x_1 + c_2x_2 + \ldots c_nx_n \\ \text{s. t. } & a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n = b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n = b_m \\ & x_1, x_2, \ldots, x_n \geq 0 \end{aligned} \tag{3.1}$$

or, using a matricial notation,

$$\begin{aligned} Z = & \min \mathbf{c}^T\mathbf{x} \\ \text{s. t. } & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{3.2}$$

where $\mathbf{x}$ is the n-vector of decision, $\mathbf{c}$ is the n-vector of objective function coefficients, $\mathbf{A}$ is the $m \times n$ matrix of constraints and $\mathbf{b}$ is the m-dimensional column array of independent terms, and

all $\mathbf{c}, \mathbf{A}, \mathbf{b}$ are real known data. The set of solutions that satisfy the model constraints, is defined as $\mathcal{X} = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. An *optimum value* $\mathbf{x}^*$ is a feasible solution where $\mathbf{c}^T\mathbf{x} \geq \mathbf{c}^T\mathbf{x}^*$ for any other feasible solution $\mathbf{x} \in \mathcal{X}$.

According to the type of variables, linear optimization problems can be classified in:

General Linear Programming (LP) problems, where all decision variables are continuous, this is, they take values in the space of real numbers $\mathbb{R}$.

Integer Programming (IP) problems, where all decision variables take integer values. They are known as *pure integer linear problems* and they are denoted as *IP*. Particulary, it is a **0–1 problem** if all integers take values in $\{0, 1\}$.

Mixed Integer Programming (MIP) problems, if there are continuous and integer decision variables. They are known as *mixed integer linear problems*, and they will be denoted as *MIP*. Particulary, if all integer variables take values in $\{0, 1\}$ the problem is known as **mixed 0–1 problem.**

3.2.2 Decisions and stages

Stochastic programs are optimization problems in which some of the model parameters $\mathbf{c}, \mathbf{A}, \mathbf{b}$ of the model (3.2) are considered uncertain. *Recourse programs* are those in which some decisions or recourse actions can be taken after uncertainty is disclosed. The uncertain data in the problem can be represented by random variables. An accurate probabilistic distribution of the random variables is assumed known. Problems with these characteristic appear in multiple disciplines, for example, production and distribution costs usually depend on oil cost, crop production depends on the uncertain weather condition, etc.

It is often the case that making decisions is a matter of time. Let $\mathcal{T} = \{1, 2, \ldots, T\}$ denote the set of time periods in the time horizon. Notice that time periods can be grouped in different decision stages depending on the structure of the problem.

Definition 3.1. A **stage** of a given time horizon, is a set of consecutive time periods in which the realization of some uncertain parameters takes place.

The set of decisions is divided into two groups for the **two-stage** problem:

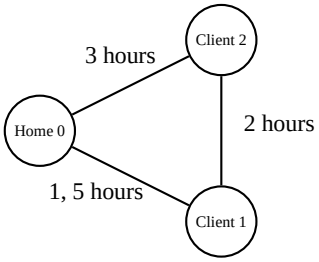


Figure 3.1: Traveling salesperson example

- Decisions that must be taken before the experiment takes place (*first-stage decisions*); the time period when these decisions are taken is called the *first stage*. They are denoted by $\overline{\mathbf{x}}$.
- Decisions that must be taken after the experiment takes place (*second-stage decisions*). They are denoted by $\overline{\mathbf{y}}(\omega)$. Since they depend on the result of the experiment ω and the first stage variables, they can be denoted $\overline{\mathbf{y}}(\omega, \overline{\mathbf{x}})$.

The sequence of events and decisions is thus summarized as: $\mathbf{x} \rightarrow \xi(\omega) \rightarrow \mathbf{y}(\omega, \overline{\mathbf{x}})$.

Observe here that the definitions of first and second stages are only related to the moment in which they are made, before and after the random experiment, and may in fact contain sequences of decisions and events. For example, in agriculture farms, the first stage corresponds to planting and occurs during the whole spring. Second-stage decisions consist of sales and purchases, and it can occur during the summer.

An illustrative example is the following (see Birge and Loveaux, [44]). A traveling salesperson receives one item every day. She visits clients hoping to sell the item. She returns home when a buyer is found or when all clients are visited. Clients buy or do not buy in a random fashion. The decision is not influenced by the previous days' decisions. The salesperson wishes to determine the order in which to visit clients, in such a way as to be at home as early as possible (seems reasonable). Time spent involves the traveling time plus some service time at each visited client. To make things simple, once the sequence of clients to be visited is fixed, it is not changed. Clearly the first stage consists of fixing the sequence and traveling to the first client. The second stage has variable length depending on the successive clients buying the item or not. Now, consider the following example. There are two clients with probability of buying 0.3 and 0.8, respectively and traveling times (including service) as in the graph of Figure 3.1 Assume the day starts at 8 a.m. If the sequence is (1, 2), the first stage goes from 8 to 9:30. The second stage starts at 9:30 and finishes either at 11 a.m. if

Client 1 buys or 2:30 p.m. otherwise. If the sequence is (2, 1), the first stage goes from 8 to 11:00, the second stage starts at 11:00 and finishes either at 2:00 p.m. or at 2:30 p.m. Thus, if sequence (2, 1) is chosen, the first stage may sometimes end after the second stage were finished if sequence (1, 2) where chosen instead (in case Client 1 buys the item).

Traditionally, stochastic problems are classified into ***two-stage*** and ***multi-stage*** *problems* (where three or more stages are considered). In the remainder of this section these models are extended.

3.2.3 two-stage models

In many cases, two stages are enough to modeling a real problem. At the first stage, decisions that cannot be postponed are made (this is, before knowing the uncertainty parameters actual value). The second stage begins when new information about the unknown parameters is available, so decisions are made taking into account the known value of these parameters. Therefore, first stage decisions will be the same for all conceivable scenario, while second stage variables are not anticipated and they will depend on the realization of the uncertain parameters.

Stochastic linear problems with two stages full recourse were independently introduced by Dantzig and Beale in 1955, and can be formulated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} + \mathrm{E}\big[\mathcal{Q}\big(\mathbf{x}, \xi(\omega)\big)\big] \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{3.3}$$

where $\mathcal{Q}(\mathbf{x}, \xi(\omega))$ is the optimal value of the second stage problem

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{q}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} = \mathbf{h} \\ & \mathbf{y} \geq 0 \end{aligned} \tag{3.4}$$

Here, $\mathbf{x}$ and $\mathbf{y}$ are vectors of first and second stage decision variables, respectively. The second stage problem depends on the data $\xi := (\mathbf{q}, \mathbf{h}, \mathbf{T}, \mathbf{W})$, some (eventually all) elements of which can be random. Therefore it is viewed $\xi = \xi(\omega)$ as a random vector. The expectation in (3.3) is taken with respect to the probability distribution of $\xi(\omega)$, which is supposed to be known. Matrices $\mathbf{T}$ and $\mathbf{W}$ are called the *technology* and *recourse* matrices, respectively. If matrix $\mathbf{W}$ is fixed (not random), the

above two-stage problem is called a problem with *fixed recourse*, since second stage problem (3.4) can be viewed as a penalty term for violating the constraint $\mathbf{Tx} = \mathbf{h}$, hence the name *with recourse*. For any x and ξ the function $\mathcal{Q}(\mathbf{x}, \xi)$, although not given explicitly, is a well defined extended real valued function: it takes the value $+\infty$ if the second stage problem (3.4) feasible set is empty, and the value $-\infty$ if the second stage problem is unbounded from below.

By the definition of the function $\mathcal{Q}(\mathbf{x}, \xi)$, this problem can be written as $\mathcal{Q}(\mathbf{x}, \xi) = \mathcal{Q}(\mathbf{h} - \mathbf{Tx})$, where

$$\mathcal{Q}(\chi) := \inf\{\mathbf{q}^T\mathbf{y} : \mathbf{Wy} = \chi, \mathbf{y} \geq \mathbf{0}\} \tag{3.5}$$

and χ denotes $\mathbf{h} - T\mathbf{x}$.

Using the duality theory of Linear Programming (LP) the optimal value $\mathcal{Q}(\chi)$ of the linear program in the right hand side of (3.5) is equal to $\sup\{\pi^t\chi : \mathbf{W}^t\pi \leq \mathbf{q}\}$, unless both systems: $\mathbf{Wy} = \chi, \mathbf{y} \geq 0$ and $\mathbf{W}^t\pi \leq \mathbf{q}$, are infeasible. Consequently,

$$\mathcal{Q}(\mathbf{x}, \xi) := \sup\big\{\pi^T(\mathbf{h} - \mathbf{Tx}) : \mathbf{W}^t\pi \leq \mathbf{q}\big\} \tag{3.6}$$

The feasible set $\mathbf{W}^t\pi \leq \mathbf{q}$ of the dual problem is convex polyhedral. Therefore, for any realization of random data ξ, the function $\mathcal{Q}(\cdot, \xi)$ is convex piecewise linear, whose properties have been extensively studied. For more details, see [44], among others.

3.2.4 Multistage models

The **two-stage** model is a special case of a more general structure, called **multi-stage** stochastic programming model, in which the decision variables and constraints are divided into groups corresponding to stages $t = 1, \ldots, T$. The fundamental issue in such a model is *information structure*: what is known at stage t when decisions associated with t are made.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_T$ be the decision vectors corresponding to time periods (stages) $1, \ldots, T$. At each stage some parameters are revealed, and the following sequence of actions take place:

$$\text{decision } (\mathbf{x}_1) \to \text{observation } \xi_2 = (\mathbf{c}_2, \mathbf{A}_{21}, \mathbf{A}_{22}, \mathbf{b}_2) \to \text{decision } (\mathbf{x}_2) \to \ldots \to \text{observation}$$
$$\xi_T = (\mathbf{c}_T, \mathbf{A}_{T,T-1}, \mathbf{A}_{TT}, \mathbf{b}_T) \to \text{decision } (\mathbf{x}_T)$$

So, the following LP problem is considered:

$$
\begin{array}{lllll}
\min & \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{c}_2^T\mathbf{x}_2 + \mathbf{c}_3^T\mathbf{x}_3 + \cdots + \mathbf{c}_T^T\mathbf{x}_T & & & \\
\text{s.t.} & \mathbf{A}_{11}\mathbf{x}_1 & & & = \mathbf{b}_1 \\
& \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 & & & = \mathbf{b}_2 \\
& \mathbf{A}_{32}\mathbf{x}_2 + \mathbf{A}_{33}\mathbf{x}_3 & & & = \mathbf{b}_3 \\
& \vdots & & & \\
& & & \mathbf{A}_{T,T-1}\mathbf{x}_{T-1} + \mathbf{A}_{TT}\mathbf{x}_T & = \mathbf{b}_T \\
& \mathbf{x}_1, & \mathbf{x}_2, \quad \mathbf{x}_3, & \cdots \quad \mathbf{x}_T & \geq \mathbf{0}
\end{array}
\tag{3.7}
$$

It is called a **multi-stage** stochastic program if $\mathbf{c}_1, \mathbf{A}_{11}$ and $\mathbf{b}_1$ are known, but some (eventually all) of cost vectors $\mathbf{c}_2, \ldots, \mathbf{c}_T$, matrices $\mathbf{A}_{t,t-1}$ and $\mathbf{A}_{tt}$, $t = 2, \ldots, T$, and right hand side vectors $\mathbf{b}_2, \ldots, \mathbf{b}_T$ are random parameters.

The objective is to design the decision process in such a way that the expected value of the total cost is minimized while optimal decisions are allowed to be made at $every$ stage $t = 1, \ldots, T$.

Let ξ_t denote the data which become known at stage t. In the setting of the **multi-stage** problem (3.7), ξ_t is assembled from the components of $\mathbf{c}_t, \mathbf{A}_{t,t-1}, \mathbf{A}_{tt}, \mathbf{b}_t$ some (all) of which can be random, and $\xi_1 = (\mathbf{c}_1, \mathbf{A}_{11}, \mathbf{b}_1)$ is assumed to be known at the first stage of problem (3.7). Let us denote by $\xi_{[t_1,t_2]} = (\xi_{t_1}, \ldots, \xi_{t_2})$, for $1 \leq t_1 \leq t_2 \leq T$, the history of the process from stage t_1 to stage t_2. In particular, $\xi_{[1,t]}$ represents the information available up to time t. The key idea in the above **multi-stage** process is that every decision vector $\mathbf{x}_t$ may depend on the information available at stage t (that is $\xi_{[1,t]}$), but not on the information to be revealed at later stages. This differs **multi-stage** stochastic programs from deterministic multiperiod problems, in which all the information is assumed to be available at the beginning of the time horizon.

There are several ways to formulate a **multi-stage** stochastic program in a precise mathematical form. For instance, $\mathbf{x}_t = \mathbf{x}_t(\xi_{[1,t]})$, $t = 2, \ldots, T$, can be viewed as a function of $\xi_{[1,t]} = (\xi_1, \ldots, \xi_t)$, and the minimization in (3.7) is performed over appropriate functional spaces. If the number of scenarios is finite, this leads to a formulation of the linear **multi-stage** stochastic program as one large structured (deterministic) LP problem. It is also useful to connect dynamics of the **multi-stage** process starting from the end as follows.

Let us look at the problem from the perspective of the last stage T. At that time the values of all problem data, $\xi_{[1,T]}$, are already known, and the values of the earlier decision vectors,

$\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}$, have been chosen. Our problem is, therefore, a simple LP problem

$$\begin{aligned} \min_{\mathbf{x}_T} \quad & \mathbf{c}_T^T \mathbf{x}_T \\ \text{s.t.} \quad & \mathbf{A}_{T,T-1}\mathbf{x}_{T-1} + \mathbf{A}_{TT}\mathbf{x}_T = \mathbf{b}_T \\ & \mathbf{x}_T \geq \mathbf{0}. \end{aligned} \tag{3.8}$$

The optimal value of this problem depends on the earlier decision vector $\mathbf{x}_{T-1}$ and data $\xi_T = (\mathbf{c}_T, \mathbf{A}_{T,T-1}, \mathbf{A}_{TT}, \mathbf{b}_T)$, and is denoted by $\mathcal{Q}_T(\mathbf{x}_{T-1}, \xi_T)$. At stage $T-1$ it is known $\mathbf{x}_{T-2}$ and $\xi_{[1,T-1]}$. Therefore, let the following **two-stage** stochastic programming problem

$$\begin{aligned} \min_{\mathbf{x}_{T-1}} \quad & \mathbf{c}_{T-1}^T \mathbf{x}_{T-1} + \mathbf{E}[\mathcal{Q}_T(\mathbf{x}_{T-1}, \xi_T)|\xi_{[1,T-1]}] \\ \text{s.t.} \quad & \mathbf{A}_{T-1,T-2}\mathbf{x}_{T-2} + \mathbf{A}_{T-1,T-1}\mathbf{x}_{T-1} = \mathbf{b}_{T-1} \\ & \mathbf{x}_{T-1} \geq \mathbf{0}. \end{aligned} \tag{3.9}$$

The optimal value of the above problem depends on $\mathbf{x}_{T-2}$ and data $\xi_{[1,T-1]}$, and is denoted $\mathcal{Q}_{T-1}(\mathbf{x}_{T-2}, \xi_{[1,T-1]})$.

In general, at stage $t = 2, \ldots, T-1$, the problem is:

$$\begin{aligned} \min_{\mathbf{x}_t} \quad & \mathbf{c}_t^T \mathbf{x}_t + \mathbf{E}[\mathcal{Q}_{t+1}(\mathbf{x}_t, \xi_{[1,t+1]})|\xi_{[1,t]}] \\ \text{s.t.} \quad & \mathbf{A}_{t,t-1}\mathbf{x}_{t-1} + \mathbf{A}_{t,t}\mathbf{x}_t = \mathbf{b}_t \\ & \mathbf{x}_t \geq \mathbf{0}. \end{aligned} \tag{3.10}$$

Its optimal value is denoted $\mathcal{Q}_t(\mathbf{x}_{t-1}, \xi_{[1,t]})$ and is called the *cost-to-go* function. On top of all these problems is the problem to find the first decisions, $\mathbf{x}_1$,

$$\begin{aligned} \min_{x_1} \quad & \mathbf{c}_1^T \mathbf{x}_1 + \mathbf{E}[\mathcal{Q}_2(\mathbf{x}_1, \xi_2)] \\ \text{s.t.} \quad & \mathbf{A}_{11}\mathbf{x}_{11} = \mathbf{b}_1 \\ & \mathbf{x}_1 \geq \mathbf{0}. \end{aligned} \tag{3.11}$$

Note that all subsequent stages $t = 2, \ldots, T$ are *absorbed* in the above problem (3.11) into the function $\mathcal{Q}_2(\mathbf{x}_1, \xi_2)$ through the corresponding expected values.

Thus, we obtain the following *nested formulation*:

$$\min_{\mathbf{A}_{11}x_1=\mathbf{b}_1} \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{E}\Big[\min_{\mathbf{A}_{21}\mathbf{x}_1+\mathbf{A}_{22}\mathbf{x}_2=\mathbf{b}_2} \mathbf{c}_2^T\mathbf{x}_2 + \mathbf{E}\big[\cdots + \mathbf{E}[\min_{\mathbf{A}_{T,T-1}\mathbf{x}_{T-1}+\mathbf{A}_{TT}\mathbf{x}_T=\mathbf{b}_T} \mathbf{c}_T^T\mathbf{x}_T]\big]\Big], s.t. \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \geq \mathbf{0}.$$

Observe that the dimensions of this type of problems may become huge, making more difficult the resolution. In the following section we revise the most relevant techniques to solve this type of problems.

3.3 Modeling via scenario tree

3.3.1 Scenario tree

Scenario tree modeling is a technique used to model and interpret the uncertainty. The traditional approach for modeling the uncertainty implies the election of a probability distribution, the estimation of its parameters using historical data and finally the development of a stochastic model. However, this approach may not be appropriate if, for instance, we do not have enough information. In addition, in many applications, it is necessary and possible to consider information that is not reflected in historical data.

Definition 3.2. A **scenario** is a particular realization of the uncertain parameters along the different stages of the time horizon.

Definition 3.3. A **scenario group** for a given stage is the set of scenarios with the same realization of the uncertain parameters up to the stage.

In most real problems, despite existing infinite possible realizations of a random variable, these can be reduced to a finite number of *representative* realizations (or events). Several methodologies are commonly used. Amongst others, we can cite:

1. Neuronal network (used in Supply Chain, Energy, Environmental models, see [39, 59, 165]).
2. Monte Carlo simulation (used in Financial model, see [141], among others).
3. Cluster analysis, (used in Supply Chains, Energy models, see [123] among others).

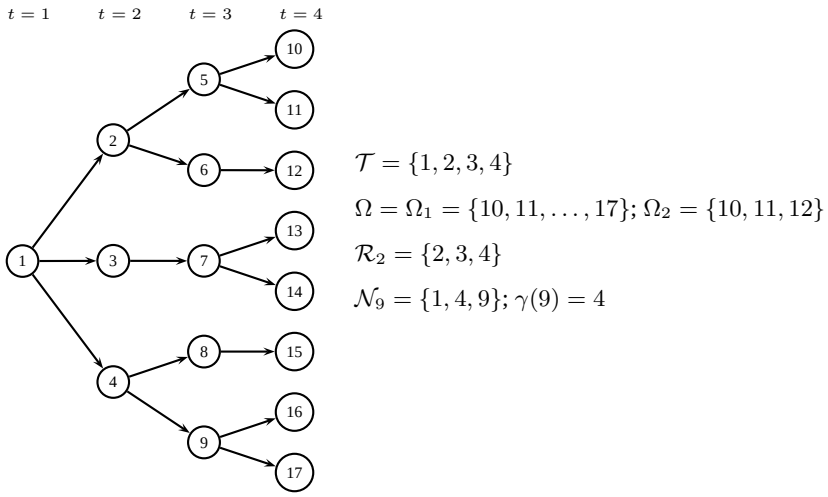


Figure 3.2: Scenario tree

These and others methodologies can help in determining the representative set of events that will be represented by the set of scenarios in the model, see the works of Dantzig and Glynn [70], Dempster and Thompson [74] as well as Di Domenica's PhD Thesis [75], among others.

Many of today approaches for stochastic programming are scenario tree-based approaches (see [27]), to illustrate this concept consider Figure 3.2: each node represents a point in time where a decision can be made. Once a decision is made, some contingencies can happen (e.g. , in this example there are three contingencies for time period $t = 2$), and information related to these contingencies is available at the beginning of the stage (here, time period). This information structure is visualized as a tree, where each root-to-leaf path represents one specific scenario and corresponds to one realization of the whole set of the uncertain parameters. Each node in the tree can be associated with a scenario group, such that two scenarios belong to the same group in a given stage, provided that they have the same realizations of the uncertain parameters up to the stage. Accordingly to the ***non-anticipativity*** principle, see e.g. Birge and Louveaux (1997) [44], both scenarios should have the same value for the related variables with the time index up to the given stage.

Let the following notation related to the scenario tree:

$\mathcal{T}$, set of stages along the time horizon. $\mathcal{T}^- \equiv \mathcal{T} \setminus \{|\mathcal{T}|\}$, being T the last stage in the time horizon.

Ω, set of scenarios.

$\mathcal{R}$, set of scenario groups, so that we have a tree where $\mathcal{R}$ is the set of nodes.

$\mathcal{R}_t$, set of scenario groups in stage t, for $t \in \mathcal{T}$ ($\mathcal{R}_t \subseteq \mathcal{R}$).

Ω_r, set of scenarios in group r, for $r \in \mathcal{R}$ ($\Omega_r \subseteq \Omega$).

$\gamma(r)$, immediate ancestor node of node r, for $r \in \mathcal{R}$.

$\mathcal{V}_r$, set of ancestor scenario groups to group r, including itself, for $r \in \mathcal{R}$. Notice that $(\mathcal{V}_r \subset \mathcal{R})$. Notice also that $\mathcal{V}_r$ can be defined as the set of scenario groups such that $\Omega_r \subseteq \Omega_{r'}$.

$\mathcal{V}^r$, set of successor scenario groups to group r, for $r \in \mathcal{R}$. Notice again that $\mathcal{V}^r$ can be defined as the set of scenario groups such that $\Omega_{r'} \subseteq \Omega_r \forall r' \in \mathcal{V}^r$

w^ω, likelihood that the modeler associates with scenario ω, $P(\xi = \xi^\omega) = w^\omega$, for $\omega \in \Omega$, and $\sum_{\omega \in \Omega} w^\omega = 1$.

w_r, weight factor representing the likelihood associated with scenario group r, for $r \in \mathcal{R}$. Note: $w_r = \sum_{\omega \in \Omega_r} w^\omega$ and $\sum_{r \in \mathcal{R}_t} w_r = 1 \, \forall t \in \mathcal{T}$.

Once the scenario tree is generated, it is necessary taking into account the structure of the tree to extend the problem model. Alternatively, we can solve the deterministic problem associated to each scenario:

$$\begin{aligned} Z^\omega = & \min \mathbf{c}^\omega \mathbf{x}^\omega \\ \text{s.t.} \quad & \mathbf{A}^\omega \mathbf{x}^\omega = \mathbf{b}^\omega \\ & \mathbf{x}^\omega \geq \mathbf{0} \end{aligned} \tag{3.12}$$

In model (3.12), the criterion to select an optimal solution is not clear. Feasible solutions can appear for one scenario and not for another. A solution can have a better value in the objective function for a specific scenario and not for another, etc.

However, *scenario tree* methodology provides feasible solutions under each scenario, but without being subordinated to any of them and with the best objective function expected value for all them.

The most prominent features of a stochastic **multi-stage** model with full recourse are:

- Deterministic models for different scenarios differ some from others, in the objective function coefficients, constraints coefficients and the independent term.
- The number of variables that relate different stages is not significant.

- Constraint matrix is a quasi-stair type.

3.3.2 Non-anticipativity principle

The decisions outcomed by the model must satisfy the non-anticipativity principle that guarantees the independence of the solutions with respect to the information not yet available. The ***non-anticipativity*** *principle*, see Birge and Louveaux [44] and Rockafellar and Wets [210], says that if two different scenarios ω and $\omega^{'}$ are identical until stage τ as to as the disponible information in that stage, then the decisions in both scenarios must be the same too until stage τ.

For each realization of the uncertain parameters in different stages considered along the time horizon, and letting $\xi^{\omega} = (\xi_1^{\omega}, \xi_2^{\omega}, \ldots, \xi_n^{\omega})$, it has been seen that it can be associated a sequence of decisions $\mathbf{x}^{\omega} = (\mathbf{x}_1^{\omega}, \mathbf{x}_2^{\omega}, \ldots, \mathbf{x}_n^{\omega})$, for $\omega \in \Omega$. But these decisions are not independent one to the other.

The *non-anticipativity* principle requires that

$$\mathbf{x}_t^{\omega} = \mathbf{x}_t^{\omega^{'}} \quad \text{if} \quad \xi_{\tau}^{\omega} = \xi_{\tau}^{\omega^{'}}, \forall \tau = 1, \ldots, t,$$

Figure 3.3 represents an example of the *non-anticipativity* principle with $|\Omega| = 4$ scenarios, $S = 3$ decision stages and $T = 3$ time periods. Node 1 of the scenario tree represents the stage, or instant of time in which the first decision must be made. In this first stage the information about uncertain parameters is known with precision, ξ_1. This knowledge is expressed as $\xi_1 : \xi_1^1 = \xi_1^2 = \xi_1^3 = \xi_1^4$. That is, the realizations of the uncertain parameters under each scenario are equivalent in the first stage. So, decisions must be the same too in the first stage for each scenario. These equalities, $\mathbf{x}_1 : \mathbf{x}_1^1 = \mathbf{x}_1^2 = \mathbf{x}_1^3 = \mathbf{x}_1^4$ are represented by the discontinued circle that includes each point stage 1. Once the first decision has been made, two perspectives can take place and the information about the uncertain parameters is available at the beginning of the second stage. The *non-anticipativity* means that it can be found only two versions about the realization of stochastic parameters (and so, the decision $\mathbf{x}_2$). In the figure appears represented the equalities $\mathbf{x}_2^1 = \mathbf{x}_2^2$ and $\mathbf{x}_2^3 = \mathbf{x}_2^4$.

3.4 Deterministic Equivalent Model

Optimization techniques for treating the uncertainty, based on scenario analysis that describes partial or full recourse, constitutes the *Stochastic Optimization*. The scheme of modeling is based on

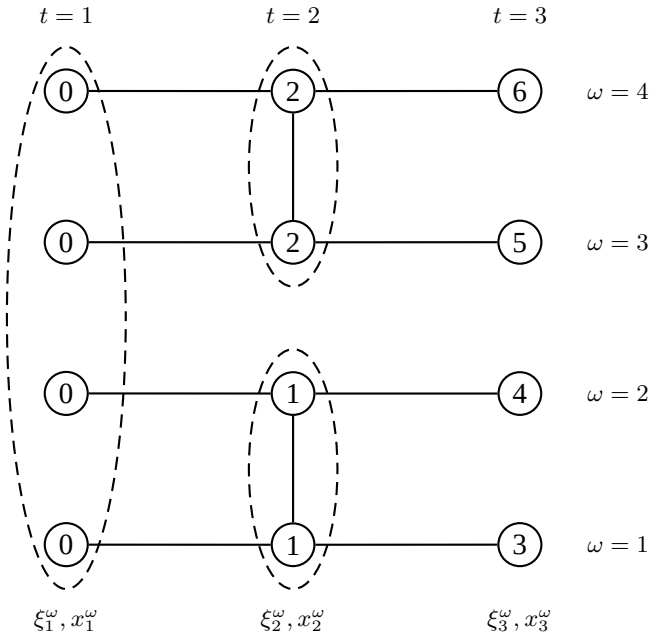


Figure 3.3: Non anticipativity principle

the *non-anticipativity principle* for expressing relationships between the stochastic parameter realization for a given stage and the corresponding decision. The solutions to be obtained are not determined to any particular scenario, but they take into account all of them.

Definition 3.4. The *Deterministic Equivalent Model (DEM)* associated to the full recourse version of the stochastic model associated to problem (3.2), is defined as:

$$\begin{aligned} Z = &\min \sum_{\omega \in \Omega} w^\omega \mathbf{c}^\omega \mathbf{x}^\omega \\ &\text{s.t. } \mathbf{A}^\omega \mathbf{x}^\omega = \mathbf{b}^\omega \quad \forall \omega \in \Omega \\ &\mathbf{x}^\omega \geq \mathbf{0} \quad \forall \omega \in \Omega \\ &\mathbf{x}^\omega \in \mathcal{N} \quad \forall \omega \in \Omega, \end{aligned} \tag{3.13}$$

where $\mathcal{N}$ is the set of solutions that satisfy the *non-anticipativity* principle.

The *non-anticipativity* constraints can be represented by different forms. One of them consists of considering them implicitly into the variables' definition, it is known as *compact representation*. This representation reduces the size of the model in relation to the number of variables. Another

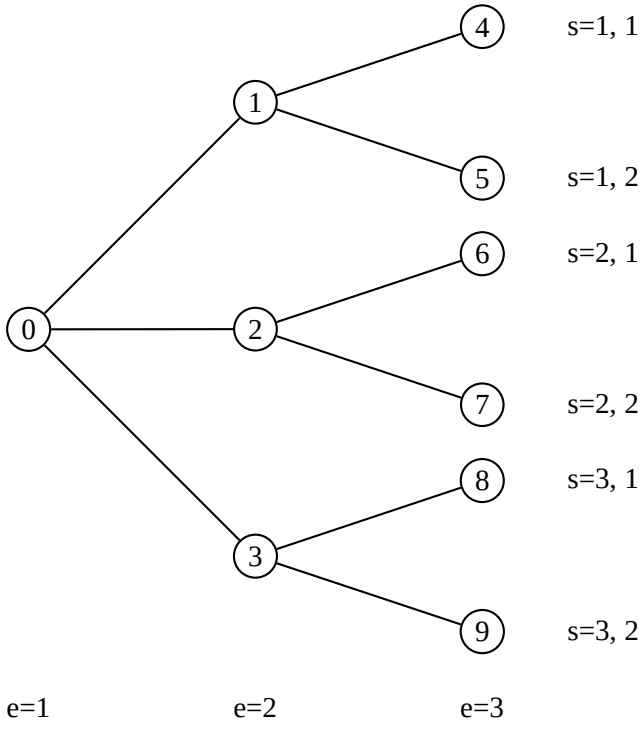


Figure 3.4: Scenario tree

way is a *splitting variable representation*, that can be very amenable depending on the optimizing algorithm to use.

3.4.1 Compact representation

Let us consider a **multi-stage** model with two types of variables:

- which only appear in first stage
- which relate two consecutive stages, so-called *two-staged linked variables*.

For example, the deterministic model with three stages is considered:

$$
\begin{aligned}
\min \quad & \mathbf{h}_1^T\mathbf{x}_1 + \mathbf{h}_{12}^T\mathbf{x}_{12} + \mathbf{h}_2^T\mathbf{x}_2 + \mathbf{h}_{23}^T\mathbf{x}_{23} + \mathbf{h}_3^T\mathbf{x}_3 \\
\text{s.t.} \quad & \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_{12}^1\mathbf{x}_{12} = \mathbf{b}_1 \\
& \mathbf{A}_{12}^2\mathbf{x}_{12} + \mathbf{A}_2\mathbf{x}_2 + \mathbf{A}_{23}^2\mathbf{x}_{23} = \mathbf{b}_2 \\
& \mathbf{A}_{23}^3\mathbf{x}_{23} + \mathbf{A}_3\mathbf{x}_3 = \mathbf{b}_3 \\
& \mathbf{x}_1, \quad \mathbf{x}_{12}, \quad \mathbf{x}_2, \quad \mathbf{x}_{23}, \quad \mathbf{x}_3 \geq \mathbf{0}
\end{aligned}
$$

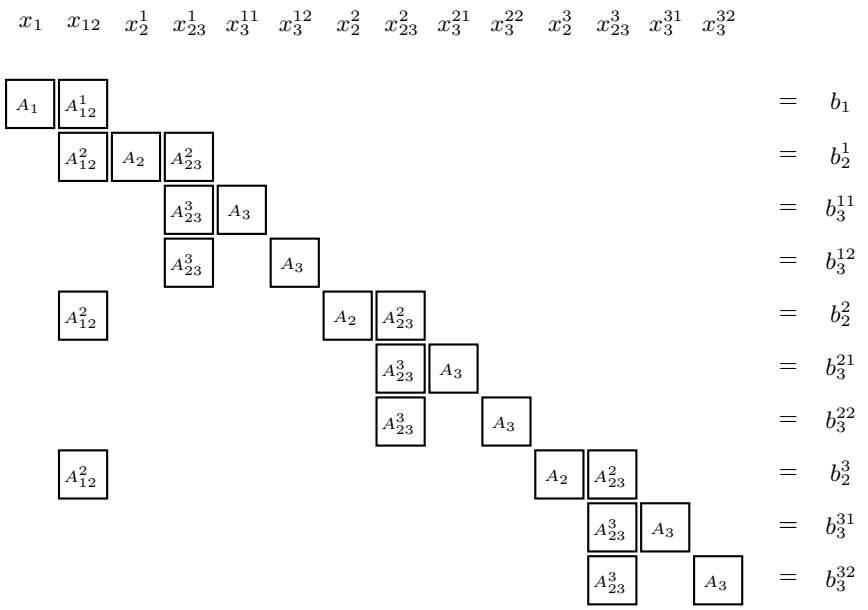


Figure 3.5: Compact representation

where variables with subscript $\mathbf{x}_e$ only appear in stage e, while variables with two subscripts $\mathbf{x}_{e,e+1}$ relate to the stages e and $e+1$. For example, consider the scenario tree represented in Figure 3.4. The compact representation of the deterministic equivalent model is given by the model,

$$
\begin{array}{lrrrrll}
\min \mathbf{h}_1^T\mathbf{x}_1 + \mathbf{h}_{12}^T\mathbf{x}_{12} + \sum_k (\mathbf{h}_2^k)^T\mathbf{x}_2^k + \sum_k (\mathbf{h}_{23}^k)^T\mathbf{x}_{23}^k + \sum_{k,l} (\mathbf{h}_3^{kl})^T\mathbf{x}_3^{kl} & & & & & & \\
\text{s.t. } \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_{12}^1\mathbf{x}_{12} & & & & & = \mathbf{b}_1 & \\
\mathbf{A}_{12}^2\mathbf{x}_{12} & + \mathbf{A}_2\mathbf{x}_2^k & + \mathbf{A}_{23}^2\mathbf{x}_{23}^k & & & = \mathbf{b}_2^k & \forall k \\
 & & \mathbf{A}_{23}^3\mathbf{x}_{23}^k & + \mathbf{A}_3\mathbf{x}_3^{kl} & & = \mathbf{b}_3^{kl} & \forall k,l \\
\mathbf{x}_1, & \mathbf{x}_{12}, & \mathbf{x}_2^k, & \mathbf{x}_{23}^k, & \mathbf{x}_3^{kl} & \geq \mathbf{0} & \forall k,l
\end{array}
$$

In first stage there is only one node in the scenario tree and the same variable is used for all scenarios, say, $\mathbf{x}_1$, in stage 2 there are three nodes and variable $\mathbf{x}_2$ is replaced by three variables $\mathbf{x}_2^k$ $k = 1, 2, 3$. Finally, in stage 3 there are 6 nodes and variable $\mathbf{x}_3$ is replaced by 6 variables $\mathbf{x}_3^{kl}$ for $k = 1, 2, 3, l = 1, 2$. Linking two-staged variables are $\mathbf{x}_{12}$ and $\mathbf{x}_{23}^k$ for $k = 1, 2, 3$. Figure 3.5 shows the matrix structure for the compact representation.

3.4.2 Representation with splitting variables by scenario groups

In the splitting variable representation, the constrains of the *non-anticipativity* principle are explicitly represented. This allows the decomposition of the model in *separated blocks*, which is

frequently an advantage for the resolution of the problem.

Definition 3.5. An **scenario group** for a given stage is the scenario set whose realization of the uncertain parameters is the same until that stage.

To build the splitting variables representation for problem (3.13), we need:

- To create a new variable for each scenario group.
- To include equality constraints among these new variables.

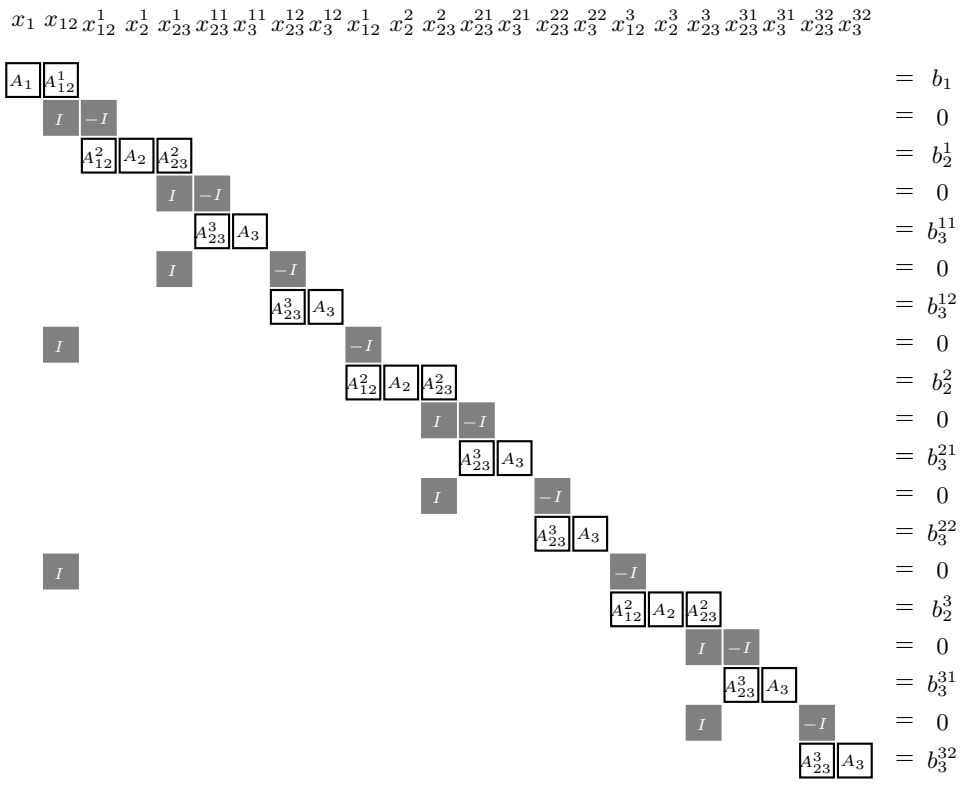


Figure 3.6: Representation with splitting variables by groups of scenarios

The representation with splitting variables by scenarios groups can be expressed:

$$\min\ \mathbf{h}_1^T\mathbf{x}_1 + \mathbf{h}_{12}^T\mathbf{x}_{12} + \sum_k (\mathbf{h}_{12}^k)^T\mathbf{x}_{12}^k + \sum_k (\mathbf{h}_2^k)^T\mathbf{x}_2^k + \sum_k (\mathbf{h}_{23}^k)^T\mathbf{x}_{23}^k + \sum_{k,l} (\mathbf{h}_{23}^{kl})^T\mathbf{x}_{23}^{kl} + \sum_{k,l} (\mathbf{h}_3^{kl})^T\mathbf{x}_3^{kl}$$

s. t.

$$\begin{array}{rrrrrrrrrrrrl}
\mathbf{A}_1\mathbf{x}_1 & + & \mathbf{A}_{12}^1\mathbf{x}_{12} & & & & & & & & & = \mathbf{b}_1 & \forall k \\
 & & \mathbf{x}_{12} & - & \mathbf{x}_{12}^k & & & & & & & = \mathbf{0} & \forall k \\
 & & & & \mathbf{A}_{12}^2\mathbf{x}_{12}^k & + & \mathbf{A}_2\mathbf{x}_2^k & + & \mathbf{A}_{23}^2\mathbf{x}_{23}^k & & & = \mathbf{b}_2^k & \forall k \\
 & & & & & & \mathbf{x}_{23}^k & - & \mathbf{x}_{23}^{kl} & & & = \mathbf{0} & \forall k,l \\
 & & & & & & & & \mathbf{A}_{23}^3\mathbf{x}_{23}^{kl} & + & \mathbf{A}_3\mathbf{x}_3^{kl} & = \mathbf{b}_3^{kl} & \forall k,l \\
\mathbf{x}_1, & & \mathbf{x}_{12}, & & \mathbf{x}_{12}^k & & \mathbf{x}_2^k, & & \mathbf{x}_{23}^k, & & \mathbf{x}_{23}^{kl}, & \mathbf{x}_3^{kl} \geq \mathbf{0} &
\end{array} \tag{3.14}$$

Figure 3.6 shows the structure of this formulation, corresponding to the first scenario group. Complete structure is obtained when it is aggregated into two equals blocks.

3.4.3 Representation with splitting variable by scenarios

This representation is similar to the previous one, but, in this case, new variables for each scenario are created. Explicitly, this can be represented for the problem:

$$
\begin{aligned}
&\min \sum_{\omega \in \Omega} p^{\omega} \left[\mathbf{h}_1^T \mathbf{x}_1^{\omega} + \mathbf{h}_{12}^T \mathbf{x}_{12}^{\omega} + \mathbf{h}_2^{\omega^T} \mathbf{x}_2^{\omega} + \mathbf{h}_{23}^{\omega^T} v x_{23}^{\omega} + \mathbf{h}_3^{\omega^T} \mathbf{x}_3^{\omega} \right] \\
&\text{s.t. } \mathbf{A}_1 \mathbf{x}_1^{\omega} + \mathbf{A}_{12}^1 \mathbf{x}_{12}^{\omega} = \mathbf{b}_1^{\omega} \quad \forall \omega \in \Omega \\
&\qquad \mathbf{A}_{12}^2 \mathbf{x}_{12}^{\omega} + \mathbf{A}_2 \mathbf{x}_2^{\omega} + \mathbf{A}_{23}^2 \mathbf{x}_{23}^{\omega} = \mathbf{b}_2^{\omega} \quad \forall \omega \in \Omega \\
&\qquad \mathbf{A}_{23}^3 \mathbf{x}_{23}^{\omega} + \mathbf{A}_3 \mathbf{x}_3^{\omega} = \mathbf{b}_3^{\omega} \quad \forall \omega \in \Omega \\
&\mathbf{x}_1^{\omega} - \mathbf{x}_1^{\omega'} = \mathbf{0} \quad \forall \omega, \omega' \in \Omega \\
&\mathbf{x}_{12}^{\omega} - \mathbf{x}_{12}^{\omega'} = \mathbf{0} \quad \forall \omega, \omega' \in \Omega \\
&\mathbf{x}_2^{\omega} - \mathbf{x}_2^{\omega'} = \mathbf{0} \quad \forall \omega, \omega' \in \Pi_r, \; r \in \{2,3,4\} \\
&\mathbf{x}_{23}^{\omega} - \mathbf{x}_{23}^{\omega'} = \mathbf{0} \quad \forall \omega, \omega' \in \Pi_r, \; r \in \{2,3,4\} \\
&\mathbf{x}_1^{\omega}, \quad \mathbf{x}_{12}^{\omega}, \quad \mathbf{x}_2^{\omega}, \quad \mathbf{x}_{23}^{\omega}, \quad \mathbf{x}_3^{\omega} \geq \mathbf{0} \quad \forall \omega \in \Omega
\end{aligned}
$$

where p^{ω} is the weight (i.e., probability) assigned to scenario ω, $r \in \{2,3,4\}$ is the set of scenario groups related to stage 2, and $\Pi_2 = \{1,2\}, \Pi_3 = \{3,4\}, \Pi_4 = \{5,6\}$ are the set of scenarios within the groups 2,3 and 4, respectively. Figure 3.7 shows the structure of this formulation.

3.5 The value of information in stochastic models

Stochastic programs have the reputation of being computationally difficult to solve. Many people faced with real-world problems are naturally inclined to solve simpler versions. Frequently used simpler versions are, for example, to solve the deterministic program obtained by replacing all random variables by their expected values or to solve several deterministic programs, each corresponding to one particular scenario, and then to combine these different solutions by some heuristic rule.

A natural question is whether these approaches can sometimes be nearly optimal or whether they are totally inaccurate. The theoretical answer to this question is given by two concepts, namely, the expected value of perfect information and the value of the stochastic solution, see [44] as we presented above.

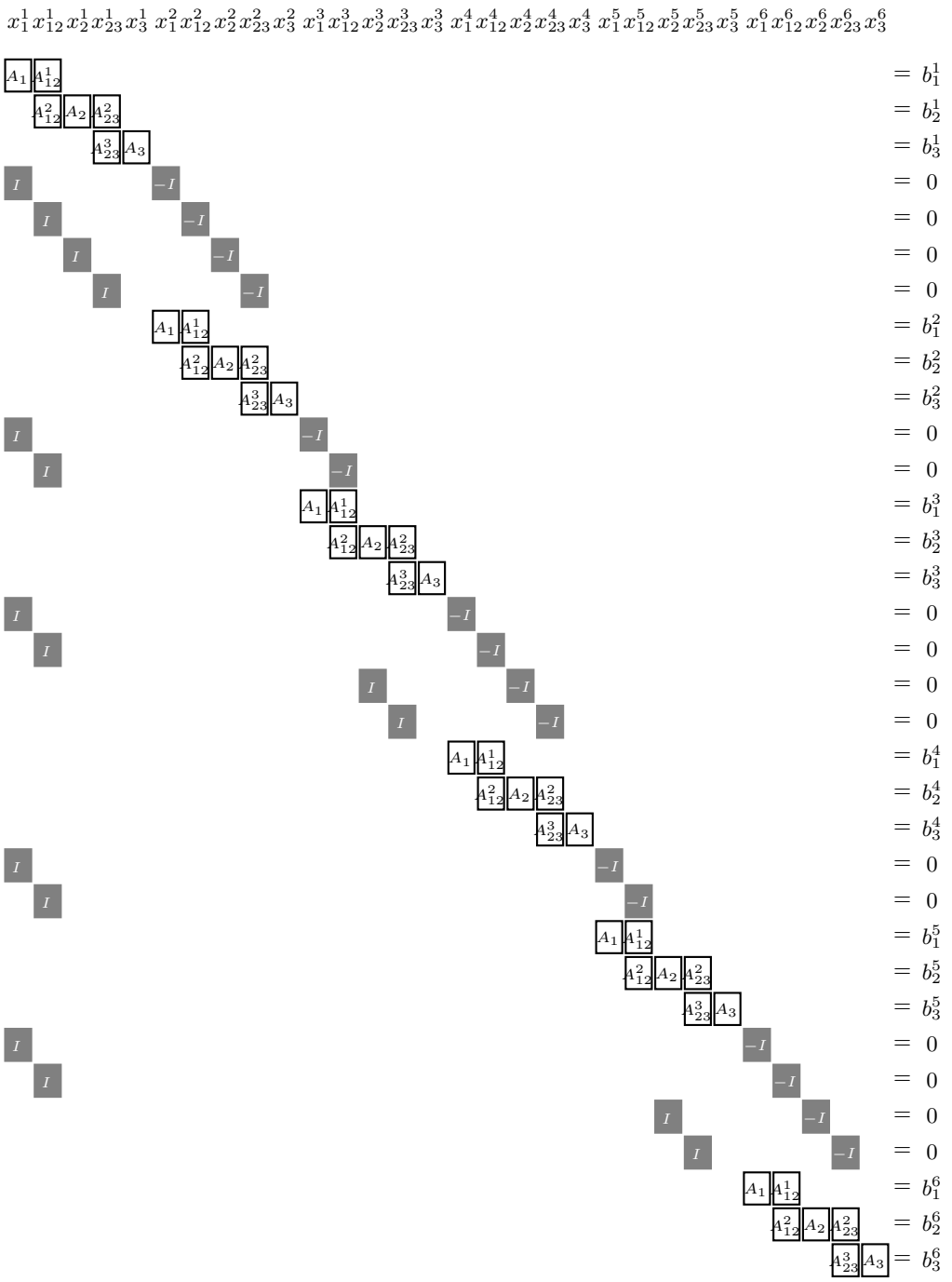


Figure 3.7: Splitting variables representation by scenarios

3.5.1 The Expected Value of Perfect Information

The ***expected value of perfect information*** **(**$EVPI$**)** measures the maximum amount a decision maker would be ready to pay in return for expected complete (and accurate) information about the future. The concept of $EVPI$ was first developed in the context of decision analysis and can be found in a classical reference such as Raiffa and Schlaifer in 1961 [201]. In the 2-stage stochastic programming setting, we may define it as follows. Suppose that the uncertainty can be modeled through a number of scenarios. Let ξ be the random variable each of whose realizations correspond to one of the different scenarios. Define

$$\begin{aligned} \min \quad & z(\mathbf{x}, \xi^{\omega}) = \mathbf{c}^T\mathbf{x} + \min\{\mathbf{q}^T\mathbf{y} | \mathbf{W}\mathbf{y} = \mathbf{h} - \mathbf{T}\mathbf{x}, \mathbf{y} \geq \mathbf{0}\} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{3.15}$$

as the optimization problem associated with one particular scenario ω. Let $\mathbf{x}(\xi^{\omega})$ be optimal solution of this problem and let $z(\mathbf{x}(\xi^{\omega}), \xi^{\omega})$ be the value of the objective function. So, we are in position to compute the expected value of the optimal solution, known in the literature as the ***wait-and-see (WS)*** solution where:

$$WS = E_{\xi}\big[\min_{\mathbf{x}} z(\mathbf{x}, \xi)\big] = E_{\xi}\big[z(\mathbf{x}(\xi), \xi)\big]. \tag{3.16}$$

We may now compare the wait-and-see solution to the so-called ***here-and-now*** solution corresponding to the recourse problem *(RP)* defined earlier in (3.15) and which we now write as:

$$RP = \min_{\mathbf{x}} E_{\xi} z(\mathbf{x}, \xi), \tag{3.17}$$

with an optimal solution, $\mathbf{x}^*$.

The expected value of perfect information is, by definition, the difference between the wait-and-see and the here-and-now solution, namely,

$$EVPI = RP - WS. \tag{3.18}$$

3.5.2 The Value of the Stochastic Solution

For practical purposes, many people would believe that finding the wait-and-see solution or, equivalently, solving the distribution problem is still too much work (or impossible if perfect information is just not available at any price). This is especially difficult because the wait-and-see approach delivers a set of solutions instead of one solution that would be implementable.

A natural temptation is to solve a much simpler problem, namely, the one obtained by replacing all random variables by their expected values. This is called the *expected value problem or mean value problem*, which is simply

$$EV = \min_{\mathbf{x}} z(\mathbf{x}, \overline{\xi}), \tag{3.19}$$

where $\overline{\xi} = E(\xi)$ denotes the expectation of ξ. Let us denote by $\overline{\mathbf{x}}(\overline{\xi})$ an optimal solution to (3.19), called the *expected value solution*. Anyone aware of some stochastic programming concepts or realizing that uncertainty is a fact of life would feel at least a little insecure about advising to make the decision $\overline{\mathbf{x}}(\overline{\xi})$. Indeed, unless $\overline{\mathbf{x}}(\overline{\xi})$ is somehow independent of ξ, there is no reason to believe that $\overline{\mathbf{x}}(\overline{\xi})$ is in any way near to the solution of the recourse problem (3.17).

The value of the stochastic solution is the concept that precisely measures how good or, more frequently, how bad a decision $\overline{\mathbf{x}}(\overline{\xi})$ is in terms of (3.17). We first define the *expected result of using the EV solution* to be

$$EEV = E_{\xi}\big[z(\overline{\mathbf{x}}(\overline{\xi}), \xi)\big]. \tag{3.20}$$

The quantity, *EEV*, measures how $\overline{\mathbf{x}}(\overline{\xi})$ performs, allowing second-stage decisions to be chosen optimally as functions of $\overline{\mathbf{x}}(\overline{\xi})$ and ξ. The value for the stochastic solution is then defined as

$$VSS = EEV - RP. \tag{3.21}$$

A high *VSS* value indicates great advantage using stochastic models in opposition to traditional models based on averages. A small *VSS* value indicates similar solutions in stochastic model and in traditional models.

It is immediate to verify the following inequalities:

$$0 \leq EVPI \qquad \text{and} \qquad 0 \leq VSS.$$

Under general conditions the following inequalities are also verified:

$$EV \leq RP \leq EEV \tag{3.22}$$

$$WS \leq RP \leq EEV \tag{3.23}$$

$$EV \leq WS \tag{3.24}$$

$$EVPI \leq EEV - EV \tag{3.25}$$

$$VSS \leq EEV - EV \tag{3.26}$$

Results of (3.22)–(3.24) are proved in [159], inequality (3.25) is proved in [31] and inequality (3.26) is proved in [42]. See in [42] some other strong lower and upper bounds on the optimal value RP.

3.6 Algorithms in Stochastic Linear Programming

3.6.1 L-Shaped method for two-stage problem

The basic idea of the L-Shaped method, introduced by Van Slyke and Wets in 1969 [231], consist of making an approximation to the non linear term of the objective function. It is well known that calculating this term implies solving all linear problems corresponding to the second stage; this is the reason for trying to avoid several evaluations of the recourse function. The proposed method consists of using that term to build a *master* problem in $\mathbf{x}$ and to evaluate the recourse function only as a subproblem.

It is based in the Benders Decomposition, a classical method in Mathematical Programming developed by Benders in 1962, which allows solving linear problems with big dimensions.

This decomposition consists of splitting the model in two linear problems: The Relaxed Master Problem (RMP), with a set of general constraints, and, the Auxiliar Problem (AP). RMP provides a cost coefficient set for AP, and it receives a new constraint set based on those coefficients. This separation makes easier the dimensional aspect, because at the end there are smaller problems.

Besides, the application of this decomposition strategy allows a lot of scenarios and stages, bigger than the allowed dimension in classic optimization of the global compact problem.

Let us consider the following two-stage stochastic problem:

$$\begin{aligned} \min\ z = \mathbf{c}^T\mathbf{x} + \sum_{\omega\in\Omega} p^\omega \mathbf{q}^{\omega T}\mathbf{y}^\omega & \\ \text{s. a} \qquad \mathbf{A}\mathbf{x} \qquad\qquad = \mathbf{b} & \\ \mathbf{T}^\omega\mathbf{x} + \mathbf{W}\mathbf{y}^\omega = \mathbf{h}^\omega \qquad \forall\omega\in\Omega & \\ \mathbf{x}, \qquad \mathbf{y}^\omega \geq \mathbf{0} \qquad \forall\omega\in\Omega & \end{aligned} \tag{3.27}$$

Figure 3.8 shows the matrix structure for this model.

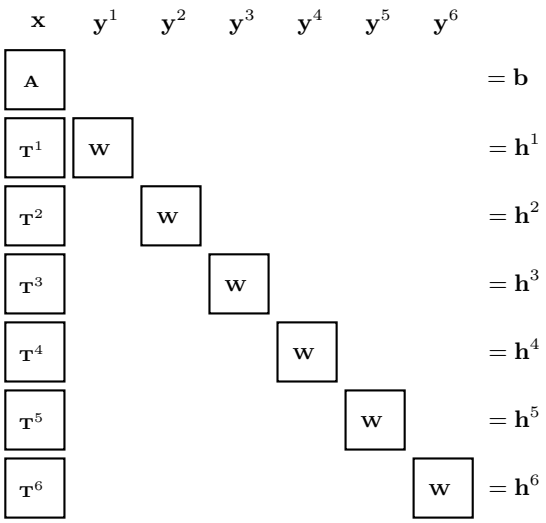


Figure 3.8: Matrix structure for the two stage compact representation

The resolution of (3.27) is equivalent to the resolution of the master problem (3.28):

$$\begin{aligned} \min\ z &= \mathbf{c}^T\mathbf{x} + Q(\mathbf{x}) \\ \text{s. a} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{3.28}$$

with the recourse function $Q(\mathbf{x})$ defined as:

$$\begin{aligned} Q(\mathbf{x}) = \sum_{\omega\in\Omega} p^\omega Q^\omega(\mathbf{x}) = \sum_{\omega\in\Omega} p^\omega \min & \mathbf{q}^{\omega T}\mathbf{y}^\omega \\ \text{s. a} \quad & \mathbf{W}\mathbf{y}^\omega = \mathbf{h}^\omega - \mathbf{T}^\omega\mathbf{x} \qquad \forall\omega\in\Omega \\ & \mathbf{y}^\omega \geq \mathbf{0} \qquad \forall\omega\in\Omega \end{aligned} \tag{3.29}$$

For $\overline{\mathbf{x}}$ fixed, the recourse function $Q(\overline{\mathbf{x}})$ is decomposable in $|\Omega|$ independent problems (see Figure3.9

$$\begin{aligned} Q^\omega(\overline{\mathbf{x}}) = \min \quad & \mathbf{q}^{\omega T}\mathbf{y}^\omega \\ \text{s.a} \quad & \mathbf{W}\mathbf{y}^\omega = \mathbf{h}^\omega - \mathbf{T}^\omega\overline{\mathbf{x}} \\ & \mathbf{y}^\omega \geq \mathbf{0} \end{aligned}$$

Using basic duality theory, $Q^\omega(\overline{\mathbf{x}})$ is equivalent to:

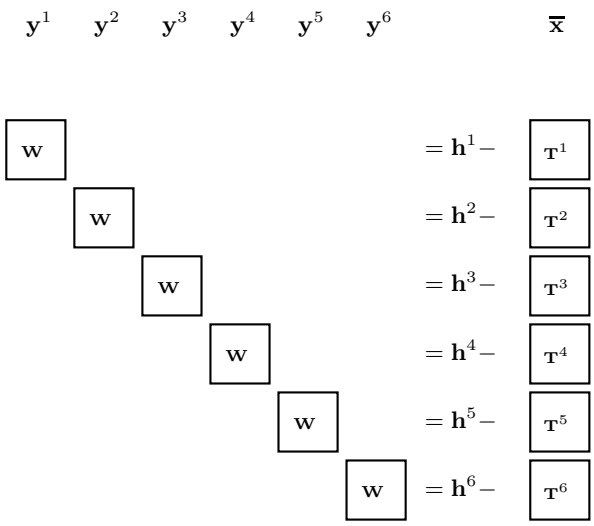


Figure 3.9: Two stage Benders decomposition

$$Q^{\omega}(\overline{\mathbf{x}}) = \max \quad \boldsymbol{\sigma}(\mathbf{h}^{\omega} - \mathbf{T}^{\omega}\overline{\mathbf{x}})$$
$$\text{s.a} \qquad \boldsymbol{\sigma}\mathbf{W}^T = \mathbf{q}^{\omega}$$

Let us assume this problem has solution θ_i^{ω} achieved at dual value $\boldsymbol{\sigma}_i^{\omega}$, $\forall \omega \in \Omega$.

The recourse function trivially satisfies the following constraint:

$$Q^{\omega}(\mathbf{x}) \geq \boldsymbol{\sigma}_i^{\omega}(\mathbf{h}^{\omega} - \mathbf{T}^{\omega}\mathbf{x}).$$

Linearizing around the point of interest $\overline{\mathbf{x}}$, the next equivalent expression is obtained

$$\begin{aligned} Q^{\omega}(\mathbf{x}) \geq \boldsymbol{\sigma}_i^{\omega}(\mathbf{h}^{\omega} - \mathbf{T}^{\omega}\mathbf{x}) =& \boldsymbol{\sigma}_i^{\omega}(\mathbf{h}^{\omega} - \mathbf{T}^{\omega}\mathbf{x} + \mathbf{T}^{\omega}\overline{\mathbf{x}} - \mathbf{T}^{\omega}\overline{\mathbf{x}}) = \\ =& \boldsymbol{\sigma}_i^{\omega}(\mathbf{h}^{\omega} - \mathbf{T}^{\omega}\overline{\mathbf{x}}) + \boldsymbol{\sigma}_i(\mathbf{T}^{\omega}\overline{\mathbf{x}} - \mathbf{T}^{\omega}\mathbf{x}) \\ =& \theta_i^{\omega} + \boldsymbol{\sigma}_i(\mathbf{T}^{\omega}\overline{\mathbf{x}} - \mathbf{T}^{\omega}\mathbf{x}). \end{aligned}$$

Therefore:

$$Q(\mathbf{x}) = \sum_{\omega \in \Omega} p^{\omega} Q^{\omega}(\mathbf{x}) \geq \sum_{\omega \in \Omega} p^{\omega}(\theta_i^{\omega} + \boldsymbol{\sigma}_i(\mathbf{T}^{\omega}\overline{\mathbf{x}} - \mathbf{T}^{\omega}\mathbf{x})).$$

This expression is known in the literature as Benders optimality cut.

If $Q^{\omega}(\overline{\mathbf{x}})$ is infeasible for some $\omega \in \Omega$, $\overline{\mathbf{x}}$ is not a valid first-stage solution. Applying the Farkas' Theorem, a cut can be generated for separating $\overline{\mathbf{x}}$. This cut is known as Benders feasibility cut. One cut can be obtained for each infeasible subproblem.

The master problem is replaced by a Relaxed Master Problem:

$$
\begin{array}{lllll}
\min & \mathbf{c}^T\mathbf{x} + \theta & & & \\
\text{s. t.} & \mathbf{A}\mathbf{x} & = \mathbf{b} & & \\
& \mathbf{D}_t\mathbf{x} & \geq \mathbf{d}_t & \forall t = 1, \ldots, r & \text{(feasibility cuts)} \\
& \mathbf{E}_t\mathbf{x} + \theta & \geq \mathbf{e}_t & \forall t = 1, \ldots, s & \text{(optimality cuts)} \\
& \mathbf{x} \geq \mathbf{0}, \ \theta \in \mathcal{R} & & &
\end{array}
$$

This cuts are added at each iteration of the procedure, using information from the Auxiliary second stage problem.

Algorithm: Benders decomposition

Step 0. Set $r = s = \nu = 0$.

Step 1. Set $\nu = \nu + 1$. Solve the linear problem:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \theta \\ \text{s. t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{D}_t\mathbf{x} \geq \mathbf{d}_t \quad \forall t = 1, \ldots, r \\ & \mathbf{E}_t\mathbf{x} + \theta \geq \mathbf{e}_t \quad \forall t = 1, \ldots, s \\ & \mathbf{x} \geq \mathbf{0}, \ \theta \in \mathcal{R} \end{aligned}$$

and let $(\mathbf{x}^\nu, \theta^\nu)$ denote an optimal solution.

Step 2. For $\omega = 1, \ldots, |\Omega|$, solve the linear problem

$$\begin{aligned} \min \quad & z'_\omega = \mathbf{e}^T\upsilon^+ + \mathbf{e}^T\upsilon^- \\ \text{s.t.} \quad & \mathbf{W}\mathbf{y} + \mathbf{I}\upsilon^+ - \mathbf{I}\upsilon^- = \mathbf{h}^\omega - \mathbf{T}^\omega\mathbf{x}^\nu \\ & \mathbf{y} \geq \mathbf{0}, \upsilon^+, \upsilon^- \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{e}^T = (1, \ldots, 1)$,

If $z'_\omega = 0 \ \forall \ \omega$, then go to Step 3.

If $z'_\omega > 0$ for some w, add a feasibility cut:

- let σ^ν the multipliers of the associated simplex.
- Define $\mathbf{D}_{r+1} = (\sigma^\nu)^T\mathbf{T}^\omega$ and $\mathbf{d}_{r+1} = (\sigma^\nu)^T\mathbf{h}^\omega$.
- Add the so-called feasibility cut $\mathbf{D}_{r+1}\mathbf{x} \geq \mathbf{d}_{r+1}$.
- Set $r := r + 1$ and return to step 1.

Step 3. For $\omega = 1, \ldots, |\Omega|$, solve the linear problem

$$\begin{aligned} Q(\mathbf{x}^\nu)^\omega = \min \quad & z = \mathbf{q}^{\omega T}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{W}\mathbf{y} = \mathbf{h}^\omega - \mathbf{T}^\omega\mathbf{x}^\nu \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Let $\boldsymbol{\sigma}^\nu_\omega$ denote the multipliers associated to the optimal solution for problem ω.

Define $\mathbf{E}_{s+1} = \sum_{\omega=1}^{|\Omega|} p^\omega(\boldsymbol{\sigma}^\nu_\omega)^T\mathbf{T}^\omega$ and $\mathbf{e}_{s+1} = \sum_{\omega=1}^{|\Omega|} p^\omega(\boldsymbol{\sigma}^\nu_\omega)^T\mathbf{h}^\omega$.

Build the optimality cut: $\mathbf{E}_{s+1}\mathbf{x} + \theta \geq \mathbf{e}_{s+1}$.

If $(\mathbf{x}^\nu, \theta^\nu)$ verifies the cut: STOP, $\mathbf{x}^\nu$ is the optimal solution.

Otherwise, set $s := s + 1$, add the cut and go back to Step 1.

◇◇◇

In the previous algorithm, only an optimality cut is added at each iteration. However, the stochastic problem structure allows to introduce several cuts (one por scenario). Birge (1988) propose a multiple cuts version for the L-Shaped method, in which one cut for each realization is added.

3.6.2 Lagrangean decomposition

A methodology available for solving problems with high dimensions is the Lagrangean relaxation. Geoffrion [112] and Fisher [104] apply this method to solve integer programming problems. This technique is based on the dualization of those equations that make more difficult the resolution of the problem. See also Guignard [120]

Let us consider the following splitting variable formulation for a two-stage stochastic program:

$$\begin{aligned} \min\ z = & \sum_{\omega\in\Omega} p^\omega \left(\mathbf{c}^T\mathbf{x}^\omega + \mathbf{q}^{\omega T}\mathbf{y}^\omega\right) \\ \text{s. a} \quad & \mathbf{A}\mathbf{x}^\omega = \mathbf{b} \qquad \forall\omega\in\Omega \\ & \mathbf{T}^\omega\mathbf{x}^\omega + \mathbf{W}\mathbf{y}^\omega = \mathbf{h}^\omega \qquad \forall\omega\in\Omega \\ & \mathbf{x}^\omega - \mathbf{x}^{\omega+1} = \mathbf{0} \qquad \forall\omega\in\Omega \\ & \mathbf{x}^\omega, \mathbf{y}^\omega \geq \mathbf{0} \qquad \forall\omega\in\Omega \end{aligned} \tag{3.30}$$

The non-anticipativity constraints destroy the block structure of the matrix and do not allows to decompose the problem in smaller subproblems. In this case, we consider a *Lagrangean Relaxation* of (3.30), obtained by deleting dualization of the *non-anticipativity* constraints results in the following

model:

$$\begin{aligned} \min\ z = & \sum_{\omega\in\Omega} p^{\omega}\big(\mathbf{c}^T\mathbf{x}^{\omega} + \mathbf{q}^{\omega T}\mathbf{y}^{\omega}\big) + \sum_{\omega\in\Omega} \pi^{\omega T}(\mathbf{x}^{\omega} - \mathbf{x}^{\omega+1}) \\ \text{s. a} \quad & \mathbf{A}\mathbf{x}^{\omega} = \mathbf{b} \qquad \forall\omega\in\Omega \\ & \mathbf{T}^{\omega}\mathbf{x}^{\omega} + \mathbf{W}\mathbf{y}^{\omega} = \mathbf{h}^{\omega} \qquad \forall\omega\in\Omega \\ & \mathbf{x}^{\omega}, \mathbf{y}^{\omega} \geq \mathbf{0} \qquad \forall\omega\in\Omega \end{aligned} \tag{3.31}$$

where π is the vector of the Lagrange multipliers and $\omega' \in \Omega_r$. Generally, we may not be able to find a vector π such that (3.31) coincides with the optimal of the original mixed integer model. However, it is known that (3.31) is a lower bound, for which the best solution is obtained by solving

$$\max_{\pi} \quad L(\pi)$$

which is named *Lagrangian dual*.

It is observed that, for fixed values of the dual variables vector π, (3.31) is separable in subproblems which can be solved independently. From the solution obtained, the multipliers are updated and the dual lagrangean is again optimized with the new values. These multipliers are updated with, e.g., the subgradient method among others (see below).

3.6.3 Obtaining Lagrangean multipliers: Subgradient method

Let π_i denote the value of π at iteration i, so it is obtained iteratively according to the rule:

$$\pi_{i+1}^{\omega} = \pi_i^{\omega} + \beta(\mathbf{x}_i^{\omega} - \mathbf{x}_i^{\omega+1}). \tag{3.32}$$

where $\boldsymbol{v}_i^{\omega} - \boldsymbol{v}_{i-1}^{\omega'}$ is the *stepdirection* (here the subgradient) and β is a scalar which represents the *steplength*. The most used expression for β is as follows,

$$\beta = \frac{\delta(L^* - L_{\pi_i})}{||\mathbf{x}_i^{\omega} - \mathbf{x}_i^{\omega+1}||^2},$$

where L_{π_i} is the optimal solution of the Lagrangean relaxation at iteration i, δ is a scalar which satisfies the inequality $0 < \delta \leq 2$ and L^* is an upper bound for (3.31). Often the succession values of δ is obtained by fixing $\delta = 2$ at the beginning and if the value L_{π_i} does not increase after a given number of iterations, the value of δ is reduced to the half.

Algorithm: Lagrangean decomposition

Step 1. Initialize $i = 0$, $\pi_i = \mathbf{0}$.

Step 2. For the multipliers value π_i, solve the Langrangean Relaxation (RL) and let $\boldsymbol{v}_i$ denote the solution of each problem.

Step 3. If for a tolerance value $\epsilon \geq 0$, it is verified that:

$$||\mathbf{x}_i^{\omega} - \mathbf{x}_i^{\omega+1}||^2 \leq \epsilon$$

stop, the optimal solution for the dual Lagrangean original model is computationally found. Otherwise, go to Step 4.

Step 4. Set $i = i + 1$, and update the multipliers vector by using the subgradient method and go back to Step 2.

◇◇◇

As an alternative we favor the Dynamically Constrained Cutting Plane scheme for updating the Lagrange multipliers.

3.6.4 Augmented Lagrangean Decomposition

The Augmented Lagrangean Decomposition (ALD) introduced by [176] and [213] improves the Lagrangean decomposition in order to force the convergence of the problem by adding an infeasibility penalty term to the objective function. Let us consider the problem (3.30), where the variable $\mathbf{x}$ is split in the variables $\mathbf{x}^{\omega}$ and $\mathbf{x}^{\omega+1}$. By dualizing the non-anticipativity constraints the Lagrangean decomposition is obtained. In order to increase the convergence speed, the following quadratic term is added

$$\frac{\rho}{2} \sum_{\omega \in \Omega} \left\| \mathbf{x}^{\omega} - \mathbf{x}^{\omega+1} \right\|^2$$

to the objective function, where $\rho > 0$ is a penalty weight. Thus, the Augmented Lagrangean decomposition (ALD) becomes,

$$\begin{aligned} L_{\rho}(\pi) = \min & \sum_{\omega \in \Omega} p^{\omega} \left(\mathbf{c}^T \mathbf{x}^{\omega} + \mathbf{q}^{\omega T} \mathbf{y}^{\omega} \right) + \sum_{\omega \in \Omega} \pi^{\omega T} (\mathbf{x}^{\omega} - \mathbf{x}^{\omega+1}) + \frac{\rho}{2} \sum_{\omega \in \Omega} \left\| \mathbf{x}^{\omega} - \mathbf{x}^{\omega+1} \right\|^2 \\ \text{s. a} \quad & \mathbf{A}\mathbf{x}^{\omega} = \mathbf{b} \qquad \forall \omega \in \Omega \\ & \mathbf{T}^{\omega}\mathbf{x}^{\omega} + \mathbf{W}\mathbf{y}^{\omega} = \mathbf{h}^{\omega} \qquad \forall \omega \in \Omega \\ & \mathbf{x}^{\omega}, \quad \mathbf{y}^{\omega} \geq \mathbf{0} \qquad \forall \omega \in \Omega \end{aligned} \tag{3.33}$$

Observe that, in contrast to the Lagrangean decomposition, this problem is not completely separable but, fortunately, it is quasi-separable quadratic.

ALD Algorithm

Step 1. Initialize $i = 0$

Step 2. For the multipliers value π_i, solve the problem (3.33) and let $\bar{\mathbf{x}_i^{\omega}}$ denote the optimal value.

Step 3. If, for given a tolerance value $\epsilon \geq 0$, it is verified:

$$||\bar{\mathbf{x}_i^{\omega}} - \bar{\mathbf{x}_i^{\omega+1}}||^2 \leq \epsilon$$

stop, the optimal computational solution has been found. Otherwise go to Step 4.

Step 4. Update the penalty weight ρ, and the multipliers vector by using, e.g., the subgradient method. Set $i = i + 1$ and go back to Step 2.

◇◇◇

This algorithm can be applied to solve a stochastic complete recourse problem formulated by splitting variables in scenario groups as well as in scenarios.

3.7 Stochastic Integer Programming

The general formulation of a *Stochastic Integer Programming* (SIP) problem is basically equivalent to the linear problem formulation. It is required that some of the variables are integer. Furthermore, in many real applications some or all of the variables are binary, this is, only take 0-1 value.

Just a few properties of stochastic integer problems are known, which is the reason why there are not many efficient methods. However, some techniques proposed in SIP to solve specific problems have been successful. In any case, SIP is very much a field under development.

Formally, a stochastic integer problem in two stages is expressed as:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \mathbf{E}_\xi[\min \mathbf{q}^{\omega^T}\mathbf{y}^\omega] \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{T}^\omega\mathbf{x} + \mathbf{W}\mathbf{y}^\omega = \mathbf{h}^\omega \\ & \mathbf{x} \in \mathcal{X},\ \mathbf{y}^\omega \in \mathcal{Y} \end{aligned}$$

where $\mathcal{X}$ and $\mathcal{Y}$ may have integrality or binary constraints. Using this definition, the equivalent deterministic integer problem can be formulated as:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \mathcal{Q}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

where $\mathcal{Q}(\mathbf{x})$ represents the expected value at the second stage.

Clearly, if integrality constraints only appear in $\mathcal{X}$, the properties of $\mathcal{Q}(\mathbf{x})$ and the feasible region $\mathcal{K}_2 = \{\mathbf{x} : \mathcal{Q}(\mathbf{x}) < \infty\}$ are the same as the continuous case. But when there are integrality constraints in the second stage, these properties are not usually true.

Proposition 3.1. *The expected recourse function of a stochastic integer problem, generally, is not convex and is discontinuous.*

The next simple example illustrates this proposition:

Let us consider the problem with only a variable $\mathbf{x} \geq \mathbf{0}$ in the first stage and an integer recourse function in the second stage:

$$\mathcal{Q}(\mathbf{x}, \xi) = \min\{2\mathbf{y}_1 + \mathbf{y}_2 : \mathbf{y}_1 \geq \mathbf{x} - \xi, \mathbf{y}_2 \geq \xi - \mathbf{x}\}$$

Let us suppose that ξ takes the values 1 or 2 with the same probability, $\frac{1}{2}$. For $\xi = 1$, the optimal solution at the second stage is:

$$\begin{aligned} &\mathbf{y}_1 = 0,\ \mathbf{y}_2 = \lceil 1 - \mathbf{x} \rceil \quad \text{if } \mathbf{x} \leq 1, \\ &\mathbf{y}_1 = \lceil \mathbf{x} - 1 \rceil,\ \mathbf{y}_2 = 0 \quad \text{if } \mathbf{x} \geq 1, \end{aligned}$$

where $\lceil a \rceil$ denotes the minor integer value greater or equal than a. Then, $\mathcal{Q}(\mathbf{x}, 1)$ is discountinous in $\mathbf{x} = 1$ and non convex.

In the feasible regions case, the convexity property does not apply either. Since in the continuous case it is defined a feasible region $\mathcal{K}_2$ of second stage for a fixed value ξ as $\mathcal{K}_2 = \{\mathbf{x} :$ exist $\mathbf{y}$ such than $\mathbf{W}\mathbf{y} = h(\xi) - \mathbf{T}(\xi)\mathbf{x}, \mathbf{y} \in \mathcal{Y}\}$, then the next result applies:

Proposition 3.2. *The feasible region of the second stage $\mathcal{K}_2(\xi) = \{\mathbf{x} : \mathcal{Q}(\mathbf{x}, \xi) < \infty\}$ is generally non convex.*

Proof

Given $\mathcal{K}_2 = \{\mathbf{x} : \mathcal{Q}(\mathbf{x}) < \infty\}$, if $Q(\mathbf{x}, \xi)$ is no convex, then, in general, $\mathcal{K}_2$ is not convex either. □

In some special cases, it is possible to calculate or approximate the function $\mathcal{Q}(\mathbf{x})$ in a reasonable time. However, these cases are exceptions, and, in fact, one of these cases is the recourse simple integer case.

There are three levels of difficulty in solving stochastic integer programs of the above form:

- Evaluating the second-stage cost for a fixed first-stage decision and a particular realization of the uncertain parameters. Note that this involves solving an instance of the second-stage problem which may be an NP-hard integer program and involve significant computational difficulties.
- Evaluating the expected second-stage cost for a fixed first-stage decision. If the uncertain parameters have continuous distribution, this involves integrating the value function $\mathcal{Q}(\mathbf{x}, \cdot)$ of an integer program, and is in general impossible. If the uncertain parameters have a discrete distribution, this involves solving a (typically huge) number of similar integer programs.
- Optimizing the expected second-stage cost. It is well known that the value function of an integer program is non-convex and often discontinuous. Consequently, the expected second-stage cost function $\mathrm{E}[\mathcal{Q}(\cdot, \omega)]$ is non-convex in $\mathbf{x}$. Figure 3.10 illustrates the non-convex nature of the objective function $\mathbf{c}^T\mathbf{x} + \mathrm{E}[\mathcal{Q}(\mathbf{x}, \omega)]$ for a small stochastic integer programming problem with two first-stage variables, see [1]. The optimization of such a complex objective function poses severe difficulties.

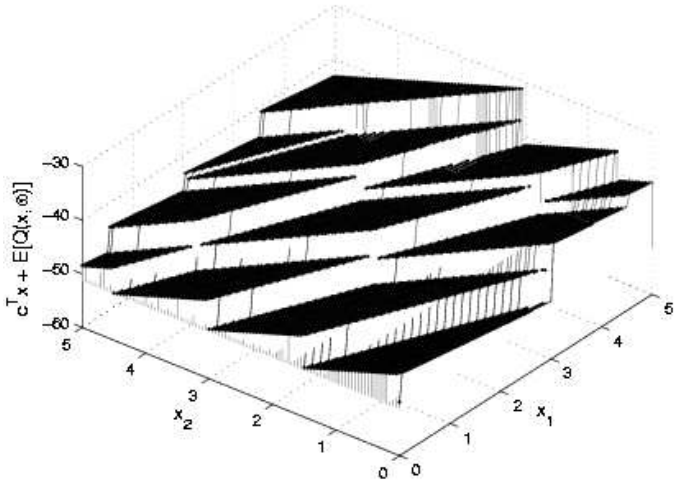


Figure 3.10: Objective function of a small Stochastic Integer problem

3.8 Algorithms in Stochastic Integer Programming

3.8.1 L-Shaped method for integer problems

In this section a general scheme for solving stochastic integer problems is presented. First, let us remember the two-stage definition problem:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + E_\xi Q(\mathbf{x}, \xi) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{X}, \end{aligned}$$

where $Q(\mathbf{x}, \xi) = \min\{\mathbf{q}^T\mathbf{y} | \mathbf{W}\mathbf{y} = \mathbf{h} - \mathbf{T}\mathbf{x}, \mathbf{y} \in \mathcal{Y}\}$. In this case, $\mathcal{X}$ and/or $\mathcal{Y}$ have integrality or binary constraints in $\mathbf{x}$ and/or $\mathbf{y}$, respectively.

The deterministic equivalent problem is defined as:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \mathcal{Q}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{X}, \end{aligned}$$

where $\mathcal{Q}(\mathbf{x})$ is the expected value of the second stage. Let $\bar{\mathcal{X}}$ denote the constraints set in $\mathcal{X}$ which does not defined the type of variables at the first stage.

In a given stage of the algorithm, the following problem is considered:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \theta \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{D}_l\mathbf{x} \geq \mathbf{d}_l \quad \forall l = 1, \ldots, r \\ & \mathbf{E}_l\mathbf{x} + \theta \geq \mathbf{e}_l \quad \forall l = 1, \ldots, s \\ & \mathbf{x} \geq \mathbf{0}, \theta \in \mathcal{R} \end{aligned}$$

This problem is obtained by performing the three following relaxation in the deterministic equivalent model:

1. Integrality conditions are changed to $\mathbf{x} \geq \mathbf{0}$
2. The constraints $\mathbf{x} \in \bar{\mathcal{X}}$ are replaced by *feasibility cuts* and
3. The exact definition of $Q(\mathbf{x})$ is replaced by a polyhedral representation in θ and the so-called *optimality cuts*.

The first stage constraint comprises $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \in \bar{\mathcal{X}}$. Constraints in $\bar{\mathcal{X}}$ are generally relaxed. This is the case when these constraints are not previously known. Also when they are known but there are so many that is not realistic to impose all of them at the beginning.

Definition 3.6. A feasibility cut set is valid in $\mathbf{x}$ if there is a finite number r such that

$$\mathbf{x} \in \{\mathbf{x} : \mathbf{D}_t\mathbf{x} \geq \mathbf{d}_t, \quad l = 1, \ldots, r\}$$

implies $\mathbf{x} \in \bar{\mathcal{X}}$

The L-Shaped and Benders Decomposition changes the exact representation of $Q(\mathbf{x})$ by a polyhedral representation. Later the extension to SIP will be shown .

Definition 3.7. A optimality cuts set is valid in $\mathbf{x}$ if there is a finite number s such that

$$(\mathbf{x}, \theta) \in \{(\mathbf{x}, \theta) : \mathbf{E}_l\mathbf{x} + \theta \geq \mathbf{e}_l, \quad l = 1, \ldots, s\}$$

implies that $\theta = \mathcal{Q}(\mathbf{x})$.

It is assumed that, for a fixed $\mathbf{x}$, $\mathcal{Q}(\mathbf{x})$ can be calculated in a finite number of steps.

Integer L-Shaped algorithm

Step 0. Set $r = s = \nu = 0$, $\bar{z} = \infty$. Fix θ value to ∞ or to a smaller appropriate bound. Make a list of pending nodes which contain only one node corresponding to the initial subproblem.

Step 1. Choose a node from the list as current problem, if there is not, stop.

Step 2. Set $\nu = \nu + 1$. Solve current problem. If there is not a feasible solution, discard the actual node and go to Step 1. Otherwise, let $(\mathbf{x}^\nu, \theta^\nu)$ denote an optimal solution.

Step 3. Verify if some of the relaxed constraints are violated. If some of them are violated, add a feasibility cut , set $r = r + 1$ and go back to Step 2. If $\mathbf{c}^T\mathbf{x}^\nu + \theta^\nu > \bar{z}$, discard the actual problem and go back to Step 1.

Step 4. Verify whether or not some of the integrality constraints are violated. If it is true, create two branches, following the *Branch and cut* procedure, append new nodes to the set of pending nodes and go back to Step 1.

Step 5. Obtain $\mathcal{Q}(\mathbf{x}^\nu)$ and $z^\nu = c^T\mathbf{x}^\nu + \mathcal{Q}(\mathbf{x}^\nu)$. If $z^\nu < \bar{z}$, update $\bar{z} = z^\nu$.

Step 6. If $\theta^\nu \geq \mathcal{Q}(\mathbf{x}^\nu)$, then discard the actual node and go back to Step 1. Otherwise, impose an optimality cut, set $s = s + 1$ and go back to Step 2.

◇◇◇

Proposition 3.3. *Whenever valid feasibility and optimality cuts set exist for a problem, then the L-Shaped method with integer variables finds an optimal solution if it exists, and in a finite number of steps.*

Proof Each one of the three relaxations can be recovered in a finite number of steps. In addition, according to Definition 3.7, $\mathcal{Q}(\mathbf{x})$ can be calculated in a finite number of steps, so Step 5 is finite too. □

The first application of the L-Shaped method for integer variables was proposed by Laporte and Louveaux, [149] for 0–1 variables in the first and second stages. A complete description can be found in Carøe and Tind [54]. A stochastic version of the *branch and cut* method used in the statistic estimation of the recourse function instead of its exact evaluation can be found in Norkin, Ermoliev and Ruszczynski [180].

3.8.2 Integer simple recourse

A two-stage stochastic problem with simple integer recourse can be written as:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + E_\xi[\min \mathbf{q}^{+^T}\mathbf{y}^+ + \mathbf{q}^{-^T}\mathbf{y}^-] \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{y}^+ \geq \xi - \mathbf{T}\mathbf{x} \\ & \mathbf{y}^- \geq \mathbf{T}\mathbf{x} - \xi \\ & \mathbf{x} \in \mathcal{X}, \mathbf{y}^+ \in \mathbb{Z}_+^m, \mathbf{y}^- \in \mathbb{Z}_+^m \end{aligned} \tag{3.34}$$

where $\mathbf{T}$ and $\mathbf{q}$ are known and fixed and where $\mathcal{X}$ defines the set of decision variables at the first stage, which can be continuous or integers and non negative. The expected value function $\mathcal{Q}(\mathbf{x})$ can be aproximated by a separable sum in m components. This is, defining $\chi = \mathbf{T}\mathbf{x}$:

$$\mathcal{Q}(\mathbf{x}) = \sum_{i=1}^{m} \psi_i(\chi_i)$$

where

$$\psi_i(\chi_i) = \mathrm{E}_{\xi_i}\psi_i(\chi_i, \xi_i)$$

Generally, it is known that the expected recourse function is non convex and, if ξ has a discrete distribution, it can be discontinuous. However, there is convexity among values of function ψ evaluated in non necessarily integers, but separated by an integer distance.

Let $\mathbf{x}^0$ denote a point in $\mathbb{R}^n$ and let $\mathbf{i} \in \mathbb{Z}^n$. Let us define $\mathbf{x}^1 = \mathbf{x}^0 + \mathbf{i}$ and for all $\mathbf{j} \in \mathbb{Z}^n$, $\mathbf{j} \leq \mathbf{i}, \mathbf{x}^\lambda = \mathbf{x}^0 + \mathbf{j}$. Equivalently,

$$\begin{aligned} \mathbf{x}^\lambda &= \lambda\mathbf{x}^0 + (1-\lambda)\mathbf{x}^1 \\ \lambda &= \frac{(\mathbf{i}-\mathbf{j})}{\mathbf{i}} \end{aligned}$$

Now, without lost of generality, $\mathbf{x}$ is used as an argument of the function ψ (doing $\mathbf{T}\mathbf{x} = \mathbf{I}\mathbf{x} = \chi$). It is possible to show that

$$\psi(\mathbf{x}^\lambda) \leq \lambda\psi(\mathbf{x}^0) + (1-\lambda)\psi(\mathbf{x}^1)$$

see Birge and Louveaux [44]. It means that a convex linear piecewise function can be written, with points separated by an integer distance. This convex function will be called a $\rho - approx$ in x if it is formed joining points $x \pm k, k$ a integer.

Particular case: $\chi = \mathbf{Tx}$ binary

If the first stage decisions are integer, it is enough that T has integer coefficients in order to ensure that χ satisfies the integrality condition. By definition of $\rho - approx$, solving problem (3.34) is equivalent to solving:

$$\min\Big\{\mathbf{c}^T\mathbf{x} + \sum_{i=1}^{m_2} \rho_i(\chi_i) | \mathbf{Ax} = \mathbf{b}, \chi = \mathbf{Tx}, \mathbf{x} \in \mathcal{X}\Big\}.$$

Since this objective function is piecewise linear and convex, the problem can be solved by a decomposition method such as the L-Shaped method.

3.8.3 Stochastic Branch-and-Bound

The algorithm so-called *Stochastic Branch and Bound* was developed by Norkin, Pflug and Ruszczynski in 1998, [181]. Its application requires the determination of a estimation of the cost function value. Let a general SIP problem:

$$\begin{aligned} \min \quad & F(\mathbf{x}) = \mathrm{E}[f(\mathbf{x}, \omega)] \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \end{aligned}$$

where $\mathcal{X}$ is a finite set of decisions and ω represents posible realizations of the random variable. The *Stochastic Branch and Bound* approach consists of:

- Performing a partition of feasible region $\mathcal{X}$ in subsets of minor size and
- estimating lower bounds of the objective function $F(\mathbf{x})$ inside these subsets.

In each step of the algorithm, a subset with the minor estimated lower bound is chosen for a later partitioning in minors subsets. In contrast to traditional *Branch and Bound* this method does not contain a step in which the algorithm ends with an exact solution, but one stop criteria is chosen and an approximate solution is attained.

Let $\mathcal{X}^p$ denote the actual subsets in which the original set $\mathcal{X}$ has been divided. These subsets are a partition P of $\mathcal{X}$. So, the original problem is divided into the subproblems:

$$\begin{aligned} \min \quad & F(\mathbf{x}) = \mathrm{E}[f(\mathbf{x}, \omega)] \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}^p \end{aligned}$$

and let

$$F^*(\mathcal{X}^p) = \min_{\mathbf{x}\in\mathcal{X}^p} F(\mathbf{x})$$

The following hypothesis are considered:

- There is a function lower bound L of the set of subsets of $\mathcal{X}$ such than for all $\mathcal{X}^p \in P$:

  $$L(\mathcal{X}^p) \leq F^*(\mathcal{X}^p)$$

  and if $\mathcal{X}^p$ is a unitary set then

  $$L(\mathcal{X}^p) = F^*(\mathcal{X}^p)$$

- There is a sequence $\{\xi(\mathcal{X}^p)\}_l$ of random estimators of $L(\mathcal{X}^p)$ which tends to $L(\mathcal{X}^p)$ when l tends to ∞, with probability one.

Stochastic Branch and Bound algorithm

Step 0. Set $r = 0, \mathcal{P}_r = \{\mathcal{X}\}$ and estimate a lower bound $\xi_0(X)$.

Step 1. If the stop criteria is satisfied, stop. Otherwise, choose a subset of lower bound:

$$Y^r = \text{argmin}\{\xi_p(\mathcal{X}^p) : \mathcal{X}^p \in \mathcal{P}_r\}$$

and a approximated solution $\mathbf{x}^r \in \mathcal{Y}^r$.

Step 2. If Y^r is unitary, set $\mathcal{P}_{r+1} = \mathcal{P}_r$. Otherwise, build a partition of subset $\mathcal{Y}^r$:

$$\mathcal{P}_r(\mathcal{Y}^r) = \{\mathcal{Y}_i^r : i = 1, \ldots, n_r\}$$

and from here build a new partition:

$$\mathcal{P}_{r+1} = (\mathcal{P}_r - \mathcal{Y}^r) \cup \mathcal{P}_r(\mathcal{Y}^r)$$

Step 3. Set $r = r + 1$, estimate $\xi_p(\mathcal{X}^p)$ for all subsets $\mathcal{X}^p \in \mathcal{P}_r$, go back to Step 1.

◇◇◇

Observe that in the Stochastic Branch and Bound method, the word *bound* does not have the same meaning as in the deterministic Branch and Bound. In this other method, some branch of the tree are cut when the lower bound are bigger than the actual upper bound of the optimal value of the cost function. However, in the stochastic Branch and Bound approach no branch is deleted definitely, but all branches can be evaluated again in next iterations. A branch can be cut only if it is possible to obtain deterministic lower and upper bounds.

3.9 BFC. Definitions and algorithmic framework for pure 0-1 problems

The **Branch-and-Fix Coordination (BFC)** methodology was introduced by Alonso-Ayuso, Escudero and Ortuño in 2003 [26]. The instances of the mixed 0-1 Deterministic Equivalent Problem (DEM) can have such large dimensions that the plain using of a state-of-the-art optimization engine can make it unaffordable. Alternatively, we can use a variety of schemes, see [85] and references therein. The BFC is aimed at solving **multi-stage** linking constraints in a mixed $0-1$ **Stochastic Programming** problem, and provides an algorithmic scheme for solving large-scale problems.

It is applied to a stochastic model via scenario tree with complete recourse. Let the splitting variable representation of the deterministic equivalent model (DEM):

$$\min \sum_{\omega\in\Omega} w^{\omega}(\mathbf{c}^{\omega}\mathbf{x}^{\omega} + \mathbf{a}^{\omega}\mathbf{y}^{\omega}) \tag{3.35}$$

$$\text{s.t. } \mathbf{A}\mathbf{x}^{\omega} + \mathbf{B}\mathbf{y}^{\omega} = \mathbf{b}^{\omega} \qquad \forall\omega\in\Omega \tag{3.36}$$

$$v_t^{\omega} - v_t^{\omega+1} = \mathbf{0} \qquad \forall\omega\in\Omega_r, \quad r\in\mathcal{G}_t, \quad t\in\mathcal{T}\backslash \tag{3.37}$$

$$\mathbf{x}^{\omega}\in\{0,1\}^n, \qquad \forall\omega\in\Omega. \tag{3.38}$$

where $v=(\mathbf{x},\mathbf{y})$, and $\mathbf{x}$ and $\mathbf{y}$ are, respectively, the vectors of 0–1 and continue variables.

In this representation, it is possible to obtain $|\Omega|$ independent problems if the *non-anticipativity* constraints (3.37) are relaxed:

$$\begin{aligned} &\min \mathbf{c}^{\omega}\mathbf{x}^{\omega} + \mathbf{a}^{\omega}\mathbf{y}^{\omega} \\ &\text{s.t. } \mathbf{A}\mathbf{x}^{\omega} + \mathbf{B}\mathbf{y}^{\omega} = \mathbf{b}^{\omega} \\ &\mathbf{x}^{\omega}\in\{0,1\}^n, \end{aligned}$$

So, $|\Omega|$ *Branch-and-Fix (BF)* trees are created, one for each scenario. Instead of obtaining the optimal solution for each problem independently, BFC is specially designed to coordinate the selection of the branching variable and branching node for each scenario-related BF tree, such that the relaxed **non-anticipativity** constraints (3.37) are satisfied when fixing the appropriate variables to either one or zero. The approach also coordinates and reinforces the scenario-related BF node pruning, the variable fixing and the objective function bounding of the subproblems attached to the nodes.

Twin Node Families (TNF) in the BFC scheme

For the presentation of the *BFC* approach, let $\mathcal{Q}^\omega$ denote the *BF* tree associated with scenario ω, $\mathcal{A}^\omega$ be the set of nodes in $\mathcal{Q}^\omega$ for $\omega \in \Omega$, $\mathcal{I}$ the set of indices of the variables in any vector $\mathbf{x}_t^\omega$, and $(x_t^\omega)_i$ the i-th variable in $\mathbf{x}_t^\omega$, for $t \in \mathcal{T}, i \in \mathcal{I}, \omega \in \Omega$.

Definition 3.8. Two variables, say, $(x_t^\omega)_i$ and $(x_t^{\omega'})_i$ are said to be *common* variables for the scenarios ω and ω' in scenario group r, if $\omega, \omega' \in \Omega_r$, $r \in \mathcal{R}_t$, for $\omega \neq \omega', t \in \mathcal{T}^-, i \in \mathcal{I}$. Notice that two *common* variables have nonzero elements in the ***non-anticipativity*** constraint related to the given scenario group.

Definition 3.9. Any two nodes, say, $a \in \mathcal{A}^\omega$ and $a' \in \mathcal{A}^{\omega'}$ are said to be *twin* nodes with respect to a given scenario group if the paths from their root nodes to each of them in their own *BF* trees $\mathcal{Q}^\omega$ and $\mathcal{Q}^{\omega'}$, respectively, either have not yet been branched on / fixed at their *common* variables or they have the same 0–1 values for their branched / fixed *common* variables $(x_t^\omega)_i$ and $(x_t^{\omega'})_i$, for $\omega, \omega' \in \Omega_r, r \in \mathcal{R}_t, t \in \mathcal{T}_1, i \in \mathcal{I}$.

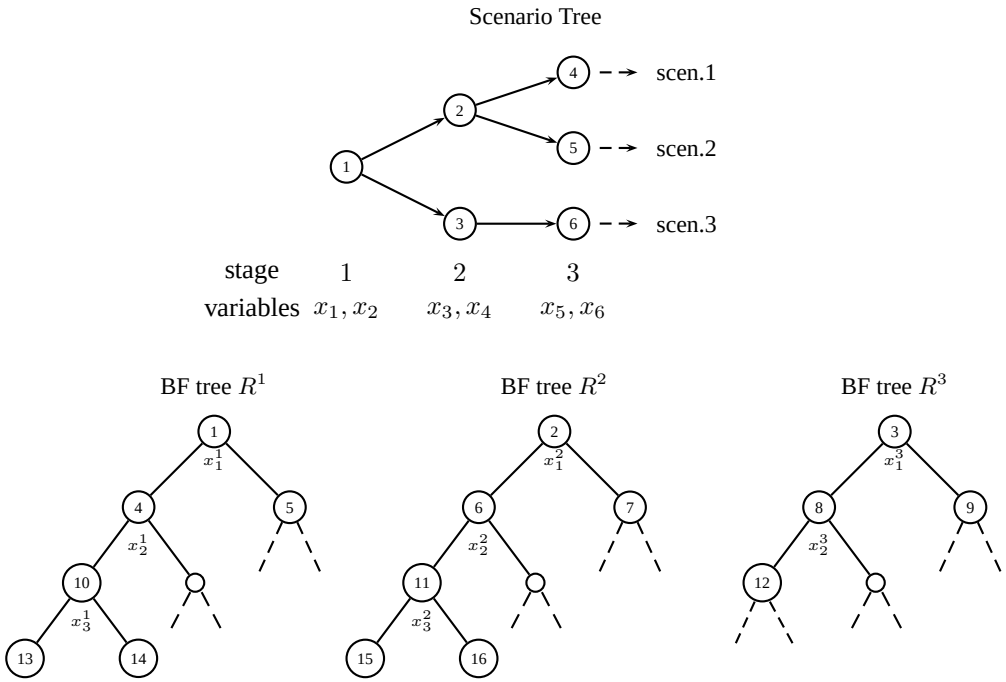
Definition 3.10. A *Twin Node Family (TNF)*, say, $\mathcal{J}_f$ is a set of nodes such that any node is a *twin* node to all the other node members in the family, for $f \in \mathcal{F}$, where $\mathcal{F}$ is the set of (indexes of) the families.

Definition 3.11. A *candidate TNF* is a *TNF* whose members have not yet branched on / fixed at all their *common* variables.

Definition 3.12. A *TNF integer set* is a set of *TNFs* where all variables take integer values, there is one node per each *BF* tree and the ***non-anticipativity*** constraints $(x_t^\omega)_i - (x_t^{\omega'})_i = 0$ are satisfied, $\forall \omega, \omega' \in \Omega_r, r \in \mathcal{R}_t, t \in \mathcal{T}^-, i \in \mathcal{I}$.

Let us consider the scenario tree and the *BF* trees shown in Figure 3.11, where x_h^ω denotes a given variable subscripted h under scenario ω and x_h gives the generic notation for the variable. For illustrative purposes, let the branching ordering $x_1, x_2, \ldots, x_6$. We can see that the first *candidate TNF* is $\mathcal{J}_1$, since the variables from stage 1 are *common* variables to all nodes. Additionally, $\mathcal{J}_2$ is a family that has already been branched on the same value of the *common* variable x_1. It is also a *candidate TNF* since the *common* variable x_2 has not been branched on (and, suppose that it has not been fixed either). Similarly, $\mathcal{J}_3$ is another *candidate TNF*. However, $\mathcal{J}_4$ is not a *candidate TNF* since all the *common* variables for their node members have been already branched on. The family $\mathcal{J}_4$ is split into the families $\mathcal{J}_5$ and $\mathcal{J}_6$ to branch independently on the variables x_3 and x_4, since the nodes 10 and 11 are *twin* nodes for these variables, while node 12 is not. Finally, note that $\mathcal{J}_7$ and $\mathcal{J}_8$ are

also *candidate TNFs*, since the variable x_4 is not yet branched and, on the other hand, it is a *common* variable for the node members of those families.



Branching variables ordering: $x_1, x_2, x_3, x_4, x_5, x_6$

Some Twin Node Families (*TNFs*)

$\mathcal{J}_1 = \{1, 2, 3\}$, $\mathcal{J}_2 = \{4, 6, 8\}$, $\mathcal{J}_3 = \{5, 7, 9\}$, $\mathcal{J}_4 = \{10, 11, 12\}^*$,

$\mathcal{J}_5 = \{10, 11\}$, $\mathcal{J}_6 = \{12\}$, $\mathcal{J}_7 = \{13, 15\}$, $\mathcal{J}_8 = \{14, 16\}$

$*$ A non candidate TNF

Figure 3.11: Branch-and-Fix Coordination (BFC) scheme

Therefore, the aim is to execute $|\Omega|$ phases of *branch and fix*, one for each scenario in a coordinated way. One master problem (MP) is considered which choose the selection of the variable to branch and node to branch. And $|\Omega|$ subproblems are considered, one for each scenario. Figure 3.11 illustrates the Branch-and-Fix trees.

BFC algorithm

Step 1. Solve $|\Omega|$ linear model associated with the scenarios. If the integrality and the ***non-anticipativity*** conditions are satisfied, then the solution is optimal for the stochastic 0-1 problem, stop. Otherwise, go to Step 2.

Step 2. Storing in MP the fractional values of variables and the solution value of each linear problem. Choose in the MP the node and the variable to branch.

Step 3. Use same branching-and-fix variables in all actives twin nodes, this is, active nodes of different trees to satisfy the **non-anticipativity** constraints on the x-variables. Optimize $|\Omega|$ subproblems in each iteration, one for each active node.

Step 4. Update the solution and the active nodes set, if it is empty, stop, the optimal solution has been obtained. Otherwise, go to Step 2.

◇◇◇

Note that the last previous sequence must be executed if some ***non-anticipativity*** constraint is violated, although all variables are 0-1.

3.10 Applications of Stochastic Programming

Due to the fact that many real-life problems have inherent uncertainty, applications for Stochastic Programming (SP) are vast. In this section we simply highlight a few of the applications where both stochastic linear programming (SLP) and stochastic mixed integer programming (SIP) have seen significant success and provide references for further reading. Unlike SLP models, most SIP models started appearing in the literature only in the last few years, due mainly to lack of practical algorithms to tackle these problems. For instance, Bertsimas [37] presents a variety of SP problems with discrete decision variables. A lot of practical problems, such as capacity planning and strategic supply chain planning under uncertainty often involve discrete decision variables. Thus applications for SIP will continue to grow as more practical solution methods for these problems are derived and implemented.

Next it is discussed applications of SP to finance [52, 65, 148, 175, 105, 116, 174, 113, 122], telecommunication [4, 46], and more recently [30, 73], transportation [192, 193], Air Traffic Flow Management [25], electricity power generation [118, 145, 86, 183, 182, 76], facility location [11, 9, 238, 91], and production, supply chain and scheduling [2, 14, 15, 27].

See also several applications in Birge [41], Dantzig and Glynn [71].

3.10.1 Transportation

Many transportation models are commonly formulated as SP models starting from the Ferguson and Dantzig [103] model. In particular, dynamic vehicle allocation has been one of the prominent areas in which SP has been applied. It involves routing a set of vehicles (e.g. trucks, freight cars, planes) to meet demand along routes and to position them for anticipated future demands (loads). The objective is to maximize the total expected returns over given time horizons. See for example Powell [192], and [193], Frantzeskakis and Powell [107] and Powell [194] for various SP dynamic vehicle allocation models. Over the last few years, Powell and Gittoes (1996) [195] and Powell et al. [196] have developed approximations and an adaptive labeling algorithm that effectively approximate the value function at each time period and yield a form of dynamic approximation. Other SP models in transportation include the widely studied stochastic vehicle routing problem. For example Laporte et al. [150] propose the integer L-shaped method for the capacitated vehicle routing problem with stochastic demands, Kenyon and Morton [143] study the stochastic vehicle routing with random travel times, and Laporte et al. [153] propose an exact solution for the *a priori* optimization of the probabilistic traveling salesman problem.

ATFM: Air Traffic Flow Management

Alonso-Ayuso et al. [25] develop an stochastic air traffic Flow Management Model based on the Bertsimas and Stock deterministic model proposed in [38]. The objective of these models is to find a flight planning without violating the capacity of the airports and air-sectors with a minimum delay expected cost. Airport capacities may change depending on weather conditions. On the other hand, the model requires 0–1 variables that model situations as, for example, a flight is going to take off in a given period of time or not, and a flight will be cancelled or not, among others. So, the model is a SIP model. The algorithmic approach is a predecessor of the *Branch-and-Fix Coordination* methodology, presented in Chapter 6.

3.10.2 Telecommunication

The system traffic, performance and reliability of telecommunications systems planning and operations naturally involve uncertainties. Therefore, Stochastic Programming naturally renders itself a viable approach to problems that arise in this field. Sen et al. [218], for example, applied the Stochastic Programing (SP) planning methodology to an industrial-sized network planning problem for Sonet-Switched Network (SSN), and demonstrates improved network performance due to the SP model. This particular problem involves making network design and configuration decisions that require consideration of random point-to-point demands with high variance forecasts in the network. The authors used the stochastic decomposition (SD) method to solve the problem.

Another problem in telecommunications system that is amenable to the SP approach is the server location problem under uncertainty. This type of problems find many real-life applications in situations where facilities or *servers* have to be located at some given potential sites in order to provide some service to potential *clients*. In such problems uncertainty appears not only in the client demands, but also in the client availability and server location costs. For example, Wang et al. in 2003 [236] study the facility location problem for immobile servers with continuous stochastic demands. They present several models and provide heuristics for their solutions. Riis et al. in 2004 [206] study a server location problem for the deployment of mobile switching centers in a telecommunications network and report on the solution of a real large-scale problem instance using the SP approach.

3.10.3 Electricity Power Generation

Electricity power generation is one of the most common areas of application and source of developments for SP methods. One of the problems, usually, referred to as the unit commitment, aims at finding a fuel cost optimal scheduling of startup/shutdown decisions and operation levels for power generation units over some given time horizon. Carøe [55] and Carøe and Schultz [53] study a unit commitment problem in the presence of uncertainty in the load profiles and develop a **two-stage** SIP model with integer first-stage and mixed-integer recourse. They apply a Lagrangean-based decomposition algorithm to solve a problem with real data for a German utility company. The problem has a total of 20.000 integer and 150.000 continuous decision variables with up to 180.000 constraints.

Other examples include the contributions of Pereira and Pinto [188, 189], where decomposition procedures are used for models of the Brazilian power system; Takriti et al. [224] apply the progressive hedging algorithm to a model of the Michigan power system designed for daily scheduling. They report achieving a convergence to near optimal solutions quickly with potential savings over a deterministic procedure of almost $150.000 in generation costs for one sample week.

The recent deregulation of the electricity market has also led to the development of new SP models in this area. For example, Sen et al. [217] develop a SP-based model for power portfolio optimization called DASH. This model is designed to help decision-makers to coordinate production decisions with opportunities in the wholesale power markets. They report that the model selects portfolio positions that perform well on a variety of scenarios generated through statistical modeling and optimization.

3.10.4 Finance

Finance problems inherently involve uncertainty due to the future time nature of financial returns and are therefore, amenable to the SP approach. The goal of SP is to provide a strategy for making decisions that hedge against unforseen scenarios and thus avoid potential losses. An excellent example of SP application to finance is the Russel-Yasuda Kasai Model reported in Cariño et al. in 1994 [51], which won second prize in the 1993 Franz Edelman Award Competition for Management Science Achievement. In the model decisions are made for a Japanese insurance company on how to optimally invest in assets to meet an uncertain liability stream over time. The investment returns are also random and the model includes legal constraints about the use of income to meet liabilities. The

authors model the problem as a **multi-stage** SLP problem and report that the model yield $79 million in its first year of use.

For a list of other successes of application of SP to finance see [67]. Other finance models can be found in [163, 148]. For more recent works, see [207, 208, 228].

3.10.5 Manufacturing

Manufacturing usually involves complex operations in which randomness cannot be ignored. The cost of raw materials, production capacity and demand of products are often random. In fact, uncertainty is inherent in manufacturing operations. In recent years the interest has increased in applying SP to tackle problems in this area, particularly in capacity-planning and expansion and strategic supply chain management under uncertainty.

Eppen et al. [82] provide a capacity-planning model at General Motors formulated as a Stochastic Linear Program aiming to determine the capacity for various products at a series of plants. They maximize the expected profit with a downside risk constraint. Ahmed and Garcia [214] study the dynamic capacity acquisition and assignment problem under uncertainty using the SP approach. Given a set of resources and tasks, this problem seeks to find a minimum cost schedule of capacity acquisitions for the resources and the assignment of resources to task over a given planning horizon. This problem arises, for example, in the integrated planning of locations and capacities of distribution centers (DCs), and the assignment of customers to the DCs, in supply chain applications. The randomness in the problem appear in the assignment costs and the processing requirements for the tasks. They formulate a SIP model and apply a decomposition based branch-and-bound method (Ahmed et al. in 2004, [3]) to solve numerous instances of the problem.

Application of SP to strategic supply chain management under uncertainty seems to have gained interest only in the last few years. Strategic supply chain planning involves the determination of production topology, plant sizing, product selection, product allocation among plants and vendor selection for raw materials. The goal is to maximize the expected profit over a given time horizon for the investment depreciation and operations costs. Uncertainty in strategic supply chain planning may appear in the product net price, product demand, raw material supply cost and production cost. Some recent work in this area include that of Escudero et al. [87], MirHassani et al. [168], Ahmed et al. [2]. In particular, Alonso-Ayuso et al. [26] presents a **two-stage** SP approach for the problem, derive a branch-and-fix coordination (*BFC*) method and report on the solution of large-scale SIP problem instances.

3.10.6 Other Applications

Other **Stochastic Programming** applications include the military (Morton et al. [169], Baker et al. [32]) and network interdiction (Cormican et al. [66], and Woodruff [242]). See too the book from Wallace and Ziemba [235] for more **Stochastic Programming** applications.

3.11 Parallel Computing

This section presents a basic introduction to Parallel Computing and the libraries that are going to be used for implementing the parallel algorithm presented in the next chapter.

3.11.1 A brief introduction to Parallel Computing

Computing consists of running programs by carrying several (ideally, many) computations simultaneously, the underlying idea is dividing large problems into smaller ones, which are then solved concurrently.

The first multiprocessor computers appeared early in the 50s during the last century, but it is not until recent years that interest in parallel computing has grown. For many years computer hardware has been experiencing continuous performance gains. In 1965, Intel co-founder Gordon Moore made the prediction now called *Moore's Law*, that states: "the number of transistors in a dense integrated circuit doubles approximately every two years". However, the physical and mechanical constraints on individual processor speed have been reached, and manufacturers are turning to another solution: multiple processors.

Nowadays, parallel computing is mainly based on multi-core processors and clusters. Computing clustering technology emerged as a result of convergence of a number of computing trends including the availability of low cost microprocessors, high speed networks, and software for high-performance distributed computing. Computer clusters are typically much more cost-effective than single computers of comparable speed or availability. This explains its increasing importance in high performance computing.

As far as software is concerned, several concurrent programming languages, libraries and APIs have been created for programming parallel computers. These can generally be classified based on the assumptions they make about the underlying memory architecture: shared memory (shared between all processing elements in a single address space) and distributed memory (in which each processing element has its own local address space). Some authors distinguish between concurrent (or parallel) computation (when memory is shared among processes), and distributed computation (when memory is distributed among interconnected independent computers).

In a shared memory environment, variables, objects, and data structures are accessible to all the processes. This allows inter-processor communication to be quite faster than the message passing

paradigm used in memory distributed systems, since such communication can be accomplished by just writing data into a memory location where another processor can read from. However, this introduces the problem of guaranteeing the consistency of data that can be accessed by concurrent processes in an indeterminate order. Moreover, shared memory supercomputers are expensive and do not scale easily. This is why, as mentioned above, parallel computing is currently mainly based on memory distributed computer clusters, which are cheaper and easily scalable.

POSIX Threads and OpenMP are two of most widely used shared memory APIs (CUDA is also growing paradigm that takes advantage of GPUs), whereas Message Passing Interface is the most widely used message-passing system API (used in distributed memory architectures). MPI has become a de facto standard for communication among processes that model a parallel program running on a distributed memory system and remains the dominant model used in high-performance computing today.

3.11.2 Message Passing Interface

In this thesis the parallel code has been implemented using C++ and the MPI library, since we understand MPI is the standard in distributed memory programming, and C++ is the best choice when the goal is getting the best computing times. The Message-passing approach makes the exchange of data cooperative among processes. Data must both be explicitly sent and received. An advantage is that any change in the receiver's memory is made with the receiver's participation, thus data is interchanged securely and easily. MPI is a message-passing library specification designed by a group of researchers from academia and industry to function on a wide variety of parallel computers, it defines the syntax and semantics of a core of library routines. MPI uses Language Independent Specifications (LIS) for calls and language bindings, since it is intended for use with any programming language. There are several well-tested and efficient MPI implementations mainly for C, C++ and Fortran, and it has been used too with Python, Perl, java, Matlab, ...

In an MPI environment every process runs independently a copy of the program, and tasks are allocated to processes by their ranks, i.e. each process is assigned a rank and then an instruction can be assigned to it by identifying its rank. MPI library functions include point-to-point send/receive operations, collective communications, combining partial results of computations (gather and reduce operations), synchronizing nodes (barrier operation) as well as obtaining network-related information such as the number of processes in the computing session, current processor identity that a process is mapped to, choosing a graph-like logical process topology, neighboring processes accessible in a

logical topology, and so on. Communication operations can be synchronous and asynchronous.

MPI-2's LIS specifies over 500 functions, however, most applications use only a subset of that standard of no more than 25 functions. In this work we will be concerned with no more than 15 functions. Anyway, the list of indispensable functions, the ones that a programmer cannot do without, are 6 (see Table 3.1), while the others are just to add flexibility, modularity, efficiency, etc. See Gropp et al. [119] for more on MPI.

MPI_Init	Initialize MPI
MPI_Comm_size	Find out how many processes there are running
MPI_Comm_rank	Find out which process I am
MPI_Send	Send a message
MPI_Recv	Receive a message
MPI_Finalize	Finalize the MPI environment

Table 3.1: six MPI functions

3.11.3 Speedup

Ideally, parallelizing a known algorithm and using p processes to run it should *get the job done* p times faster. Reality, however, is more complex and many forces oppose this ideal situation. Message passing inevitably introduces some overhead, and most algorithms include steps that cannot be parallelized, sometimes bottlenecks appear that force some processors to remain idle while waiting for others to finish their calculi. Hence, the use of several processors is not always as efficient as desired. To analyze the efficiency in parallelizing an algorithm, let us outline some ideas taken from the computer science field.

A Turing machine is a simple abstract computational device (due to the mathematician Alan Turing) devoted to understand and analyze the extent of what can be computed on a sequential computer. Additionally, it enables to define the cost of an algorithm (by estimating the number of operations needed to complete it) precisely and is the basis for *complexity* analysis. More advanced models of abstract computing machines have been developed in the field of *computer science*. Among them, the Parallel Random-Access Machine (PRAM) comprises a shared central memory that can be accessed by the different processors or *processing units* (PUs). All PUs execute the same algorithm synchronously and they may access the same memory areas. As the Turing machine, neither the number of PUs nor the memory size are bounded, and any PU is allowed to access any memory location in a time unit. With this baggage, the efficiency of a parallel algorithm can be analyzed.

Let P be a problem of size n, let $t_s(n)$ be the time (alternatively, the number of operations) needed to solve P by the best known sequential algorithm. Now let $t_p(p, n)$ be the time needed by a PRAM algorithm using p PUs to solve P. The ***speedup*** of such PRAM algorithm for problem P is then defined as $s(p, n) = \frac{t_s(n)}{t_p(p,n)}$ and expresses how many times faster is the parallel algorithm compared to the sequential one. The ***efficiency*** is defined as $e(p, n) = \frac{s(p,n)}{p}$, and can be seen as the proportion (from 0 to 1) of the total number of processors that are fully taken advantage of. The speedup and efficiency are widely used to analyze the *goodness* of a parallel algorithm.

For more on parallel programming, see Casanova et al. [57].

Chapter 4

Metaheuristic algorithms for solving large-scale multistage stochastic mixed 0-1 problems

4.1 Introduction. Fix-and-Relax Coordination

The aim of this chapter is to present several frameworks for solving large-scale multistage mixed 0-1 problems under uncertainty in the coefficients of the objective function, the right-hand-side vector and the constraints matrix. A scenario tree scheme is used to represent the Deterministic Equivalent Model of the stochastic mixed 0-1 program with complete recourse. Constraints are modeled by a splitting variable representation via scenarios.

Traditionally, special attention has been given to optimizing the DEM by maximizing the objective function expected value over the set of finite scenarios, subject to the satisfaction of all the problem constraints in the defined scenarios. Currently, small and medium-scale mixed 0-1 DEM are solved by using different types of decomposition approaches. In this chapter, different approaches based on the Fix-and-Relax (FRC) algorithm, are considered for solving large-scale stochastic mixed 0-1 problems. In Section 4.1.1 FRC algorithm is presented. Section 4.3 presents an improved version for FRC algorithm, Jumping Fix-and-Relax Coordination algorithm is introduced for obtaining, hopefully, better results. Finally, Section 4.4 concludes with a parallel version of FRC-J.

Solving the model (3.9) for a given scenario by plain using of state-of-the-art optimization engines may not require too much computing time for solving small and medium scale problems. However, given the potential dimensions of the stochastic version of the problem it is unrealistic to

solve the instances with numerous scenarios as typical dimensions of real-life cases. As an alternative, we propose using a BFC based heuristic approach, so-called Fix-and-Relax Coordination (FRC) that without, obviously, guaranteeing the optimality of the solution, provides good feasible bounds for the optimal one.

4.1.1 On the *Fix-and-Relax* scheme

Let $\mathcal{P}_r$ denote the set of (indexes of the) variables with subscript r (i.e., the variables associated with scenario group r).

As stated in [28], the optimization to be carried out at any time stage of the original problem can be decomposed, by nature, in as many independent models as the number of scenario groups in that stage. Each independent model will be a two-stage model with continuous and 0-1 variables. The first stage submodel will be included by the variables associated with the related scenario group, and the second stage submodel will also be included by the variables associated with the successor scenario groups, such that their nonanticipativity constraints are relaxed. The integrality of the 0-1 variables in set $\mathcal{P}_r$ will not be relaxed while optimizing the model attached to scenario group r. The variables associated with the ancestor scenario groups are already fixed for the problem to solve at each scenario group r. The integrality of the variables associated with the second stage will be relaxed. Moreover, the *Fix-and-Relax Coordination (FRC)* approach must satisfy the constraints (3.37) related to the first stage *common* variables in the independent model attached to the scenario group. Notice that the two stage model will be solved up optimally at the so-called *FR level* r.

4.1.2 Fix-and-Relax model

Let us consider the following MIP model

$$\begin{array}{ll} (MIP): & \min\limits_{x\in\mathcal{X},y\in\mathcal{Y}} cx+ay \\ & \text{s.t. } x_j \in \{0,1\} \qquad \forall j \in \mathcal{P}_r,\ r \in \mathcal{R}, \end{array} \tag{4.1}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the polytopes of the 0-1 variables and continuous variables, respectively, that define the feasible set, and $\mathcal{P}_r\ \forall r \in \mathcal{R}$ is a partition of $|\mathcal{R}|$ elements of the set of the variables $\mathcal{P}$, such that $\mathcal{P} = \cup_{r\in\mathcal{R}}\mathcal{P}_r$ and $\mathcal{P}_r \cap \mathcal{P}_{r'} = \emptyset$, $\forall k, k' \in \mathcal{R}\, r \neq r'$.

Problem (4.1) can be approximated by model

$$\begin{array}{rll} (MIP_r): & \min\limits_{x\in\mathcal{X}, y\in\mathcal{Y}} cx + ay & \\ \text{s.t.} & x_j = \overline{x}_j & \forall j \in \mathcal{P}'_r,\ r' < r, \\ & x_j \in \{0,1\} & \forall j \in \mathcal{P}_r, \\ & x_j \in [0,1] & \forall j \in \mathcal{P}'_r,\ r < r', \end{array} \tag{4.2}$$

where the values $\overline{x}_j$ for $j \in \mathcal{P}'_r$, $r' < r$ in the so-called *FR level* $r > 1$ are retrieved from the solution to the related models $MIP_{r'}$, $r' = 1, \ldots, r-1$.

Since only a reduced subset of (non-fixed) 0-1 variables is kept integer at *FR level* r, MIP_r can be solved with relative time efficiency. See [100, 11].

4.1.3 Branching strategy

We have chosen the *depth first* strategy, see Wolsey (1998) [241], for the *TNF* branching selection and, then, the criterion for branching consists of choosing the candidate *TNF* following the *smallest deterioration* criterion (see section 4.1.6 below for the details) among the two sons of the last branched *TNF*. When there is a guarantee that the incumbent solution could not be produced by the successor of both nodes, then a *backtracking* to the immediate ancestor node is performed.

4.1.4 Associated models to scenario group r

Here we present the models to deal with while solving a given FR level r (remind that each FR level is associated to a particular scenario group r).

Mixed integer model for scenario group r (*FR level r*)

The model to be solved at *FR level* r is as follows,

$$\begin{aligned}
\min \quad & \sum_{r' \in \mathcal{V}^r \cup \{r\}} \sum_{\omega \in \Omega_{r'}} w^{\omega} (c^{\omega}_{r'} x^{\omega}_{r'} + a^{\omega}_{r'} y^{\omega}_{r'}) \\
\text{s.t. } & A^{\omega} x^{\omega} + B^{\omega} y^{\omega} = b^{\omega} && \forall \omega \in \Omega_r \\
& v^{\omega}_{r'} = \overline{v}^{\omega}_{r'} && \forall r' \in \mathcal{V}_r / \{r\} \\
& v^{\omega}_{r} - v^{\omega+1}_{r} = 0 && \forall \omega, \omega+1 \in \Omega_r \\
& x^{\omega}_{r} \in \{0,1\} && \forall \omega \in \Omega_r \\
& 0 \leq x^{\omega}_{r'} \leq 1 && \forall \omega \in \Omega_{r'}, r' \in \mathcal{V}^r \\
& y^{\omega}_{r'} \geq 0 && \forall \omega \in \Omega_{r'}, r' \in \mathcal{V}^r \cup \{r\}
\end{aligned} \tag{4.3}$$

where $v = (x, y)$, $v^{\omega}_{r'}$ is such that $v = (v^{\omega}_{r'} \ \forall \omega \in \Omega_{r'}, r' \in \mathcal{R})$ and the values $\overline{v}^{\omega}_{r'} \ \forall r' \in \mathcal{V}_r / \{r\}$ are retrieved from the solution to the models (4.3), where r is replaced by r', being r' an ancestor scenario group (i.e., *FR level*) to group r. Notice that the **non-anticipativity** constraints (3.37) for scenario group r are $v^{\omega}_{r} - v^{\omega+1}_{r} = 0 \ \forall \omega, \omega+1 \in \Omega_r$.

Scenario set models by relaxing the **non-anticipativity** constraints

The $|\Omega_r|$ independent scenario models while relaxing the **non-anticipativity** constraints from scenario group r are as follows, for each scenario $\omega \in \Omega_r$,

$$\begin{aligned}
\min = \quad & \sum_{r' \in \mathcal{V}^r \cup \{r\}} w^{\omega} (c^{\omega}_{r'} x^{\omega}_{r'} + a^{\omega}_{r'} y^{\omega}_{r'}) \\
\text{s.t. } & A^{\omega} x^{\omega} + B^{\omega} y^{\omega} = b^{\omega} \\
& v^{\omega}_{r'} = \overline{v}^{\omega}_{r'} && \forall r' \in \mathcal{V}_r / \{r\} \\
& x^{\omega}_{r} \in \{0,1\} \\
& 0 \leq x^{\omega}_{r'} \leq 1 && \forall r' \in \mathcal{V}^r : \omega \in \Omega_{r'} \\
& y^{\omega}_{r'} \geq 0 && \forall r' \in \mathcal{V}^r \cup \{r\} : \omega \in \Omega_{r'}
\end{aligned} \tag{4.4}$$

On the LP optimal solution for TNF integer sets

The splitting variable representation for solving the model attached to a given TNF integer set

for scenario group r can be expressed as follows,

$$\begin{aligned}
&\min \sum_{r' \in \mathcal{V}^r} \sum_{\omega \in \Omega_{r'}} w^\omega c^\omega_{r'} x^\omega_{r'} + \sum_{r' \in \mathcal{V}^r \cup \{r\}} \sum_{\omega \in \Omega_{r'}} w^\omega a^\omega_{r'} y^\omega_{r'} \\
&\text{s.t. } A^\omega x^\omega + B^\omega y^\omega = b^\omega && \forall \omega \in \Omega_r \\
&\quad v^\omega_{r'} = \overline{v}^\omega_{r'} && \forall r' \in \mathcal{V}_r \\
&\quad y^\omega_r - y^{\omega+1}_r = 0 && \forall \omega, \omega+1 \in \Omega_r \\
&\quad 0 \leq x^\omega_{r'} \leq 1 && \forall \omega \in \Omega_{r'}, r' \in \mathcal{V}^r \\
&\quad y^\omega_{r'} \geq 0 && \forall \omega \in \Omega_{r'}, r' \in \mathcal{V}^r \cup \{r\}
\end{aligned} \tag{4.5}$$

4.1.5 Two-stage BFC algorithm

The following procedure is executed for each scenario group r to solve the model (4.3), for $r \in \mathcal{R}_t, t \in \mathcal{T}$. Notice that the problem to solve at each FRC iteration, see below, is a **two-stage** problem with continuous and 0-1 variables in the first stage and continuous variables in the second stage. See [84] for more details.

Step 1: Solve the $|\Omega_r|$ MIP submodels (4.4) in order to analyze the *TNF* that comprises the root nodes in the *BF* trees $\mathcal{Q}^\omega \ \forall \omega \in \Omega_r$ associated with *FR level* r. If the **non-anticipativity** constraints (3.37) are satisfied then stop, the optimal solution for *FR level* r mixed 0-1 model has been obtained. Otherwise,

Step 2: Selection of the branching variable. We will see later different possible strategies at this step, according to the *largest small deterioration* criterion. See parameter μ in section 4.1.6.

Step 3: Selection of the *TNF* by branching on the chosen 0-1 variable, according to the *smallest deterioration* criterion. Bounding the just created *TNF* by solving the appropriate scenario related models (4.4). If the bounding value is not better (in this case, smaller) than the incumbent solution value, say, $\overline{Z}_{MIP}$ then the *TNF* is pruned and goto Step 6.

Step 4: If the solution that has been obtained in Step 3 satisfies the **non-anticipativity** constraints (3.37) for the x *common* variables at *FR level* r (i.e., the *TNFs* belong to an integer set), any of the two following situations has happened:

(a) The **non-anticipativity** constraints (3.37) for the y *common* variables at *FR level* r have also been satisfied and, then, a new solution has been found for the mixed 0-1 model (4.3) attached to *FR level* r. The incumbent solution value $\overline{Z}_{MIP}$ can be updated and, additionally, the updating of the active sets at the trees BF $\mathcal{Q}^\omega \ \forall \omega \in \Omega$ can also be performed. In any case, the TNF is pruned. Goto Step 6.

(b) The **non-anticipativity** constraints (3.37) for the y *common* variables have not been satisfied. Goto Step 5.

Otherwise, goto Step 2.

Step 5: Optimize the LP model that results from fixing the x variables related to *FR level* r in model (4.3) to the values given in the TNFs whose associated model has been optimized in Step 3; see model (4.5), where Z_{LP}^{TNF} denotes the solution value. If $Z_{LP}^{TNF} < \overline{Z}_{MIP}$, then the updating of the active node sets and $\overline{Z}_{MIP}$ is performed. Prune the TNF and continue to Step 6.

Step 6: If the sets of active nodes are empty, then stop since the optimality of the *incumbent* solution for *FR level* r model has been proved, if any. Otherwise, goto Step 2.

4.1.6 FRC algorithm implementation

The *FRC* algorithm may have different implementations. Here we present the implementation that we are using in our computational experience. For the presentation of the pseudocode of the *FRC* procedure, let the additional notation:

x_{ri}**,** ith 0-1 variable whose index is in set $\mathcal{P}_r$.

$\overline{x}_{ri}^{\omega}$**,** value of the ith 0-1 variable obtained as the solution of the scenario related model (4.4), for $r \in \mathcal{R}, \omega \in \Omega_r$.

Z_{MIPr} , solution value of the MIP_r mixed integer model defined in (4.3) for *FR level* r.

Z_{MIP}^{FRC}**,** solution value of the original problem given by the proposed approach.

Let us introduce the elements that we use for selecting the variable (following the *largest small deterioration* criterion) and the two descendant *TNF* from a given one (following the *smallest deterioration* criterion), where μ_{ri} is the selection parameter for the ith 0-1 variables to branch jointly in the scenario BF trees in the problem 4.3 for scenario group r.

$$\mu_{ri} = \min\Big\{ \sum_{\omega \in \Omega_r} \overline{x}_{ri}^{\omega},\ |\Omega_r| - \sum_{\omega \in \Omega_r} \overline{x}_{ri}^{\omega} \Big\} \quad \forall i \in \mathcal{P}_r.$$

Additionally, $h_{<i>}$ will denote the ith variable in set $\mathcal{P}_r$ in a non-increasing order of the

μ-parameter. Let the branching parameter λ_{ri} be such that

$$\lambda_{ri} = \begin{cases} 0, & \text{if } \sum_{\omega \in \Omega_r} \overline{x}_{ri}^{\omega} \leq |\Omega_r| - \sum_{\omega \in \Omega_r} \overline{x}_{ri}^{\omega} \\ 1, & \text{otherwise.} \end{cases} \quad \forall i \in \mathcal{P}_r.$$

Procedure *FRC*

$Z_{MIP}^{FRC} := 0$, $t := 1$

While $t \leq |\mathcal{T}|$ do:

For $r \in \mathcal{R}_t$, do:

Build *FR level* model MIP_r (4.3), where the continuous and 0-1 variables for the ancestor *FR levels* are fixed, the 0-1 variables for the model associated with scenario group r are restricted to be 0-1 valued, the integrality constraint is relaxed for the 0-1 variables of the successor scenario groups, and the nonanticipativity constraints for the successor scenario groups are also relaxed.

Obtain Z_{MIPr} by executing the procedure *BFC2*

If $Z_{MIPr} = +\infty$ then stop

If $t = |\mathcal{T}|$ then $Z_{MIP}^{FRC} := Z_{MIP}^{FRC} + Z_{MIPr}$

Endfor

$t := t + 1$

Endwhile

Procedure *BFC2* for *FR level* r

Step 1: Initialize $i := 0$ and $Z_{MIPr} := +\infty$.

Step 2: Solve the LP relaxations of the $|\Omega|$ models (4.4). It can be done in parallel. If the variables from set $\mathcal{P}_r$ have 0-1 values $\overline{x}_{ri}^{\omega}$ $\forall \omega \in \Omega$ and the constraints (3.37) are satisfied, then update the solution value Z_{MIPr} and the related solution and return to the main program since the optimal solution for *FR level* r has been obtained.

Step 3: Update $i := i + 1$.

Step 4: Set $h_{<i>} = \text{argmax}_{j \in \mathcal{P}_r}\{\mu_{rj}, \}$, such that variable j has not been previously branched on, nor fixed at in the current branching path.

Branch $x_{rh_{<i>}} := \lambda_{rh_{<i>}}$, for all scenario in scenario group r.

Step 5: Solve the LP problems LP_r^{ω} $\forall \omega \in \Omega_r$ and compute Z_{LPr}. It can be done in parallel.

If $Z_{LPr} \geq Z_{MIPr}$ then goto Step 7.

If there is any 0-1 variable in set $\mathcal{P}_r$ that either takes continuous values or it takes different values $\overline{x}_{ri}^{\omega}$ for some of the scenario models (4.4) in group r then goto Step 3.

If all the y variables in set $\mathcal{P}_r$ take the same value for all scenario model (4.4) in group r, then update $Z_{MIPr} := Z_{LPr}$, and goto Step 7.

Step 6: Solve LP model (4.5) for satisfying the constraints (3.37) for the *common* y variables in given *FR level* r. Notice that the solution value is denoted by Z_{LP}^{TNF}.

Update $Z_{MIPr} = min\{Z_{LP}^{TNF}, Z_{MIPr}\}$.

If $i < |\mathcal{P}_r|$ then goto Step 3.

Step 7: Prune the current branch.

If $x_{rh_{<i>}} = \lambda_{rh_{<i>}}$, then goto Step 10.

Step 8: Update $i := i - 1$.

If $i = 0$ then save the solution value Z_{MIPr} and the related solution, if any, and return.

Step 9: If $x_{rh_{<i>}} = 1 - \lambda_{rh_{<i>}}$, then goto Step 8.

Step 10: Branch $x_{rh_{<i>}} := 1 - \lambda_{rh_{<i>}}$, for all scenarios in group r.

Goto Step 5.

4.2 Break stage scenario clustering

The relaxation of the ***non-anticipativity*** constraints (3.37) in the model (3.35) results in a set of $|\Omega|$ independent mixed 0–1 models. This way the original problem is decomposed into smaller independent subproblems easier to solve separately, but they may be more subproblems than needed. There are other ways to divide the problem in subproblems and one of them is scenario clustering. Let us the following definition:

Definition 4.1. A ***scenario cluster*** is a set of scenarios where the ***non-anticipativity*** constraints are implicitly defined in the model.

A scenario-cluster partitioning allows a combination of compact and splitting variable representations in the different stages of the problem, keeping the ***non-anticipativity*** constraints implicitly defined in the compact representation for each cluster, while those relating different cluster's variables are explicitly represented in the splitting variable representation.

It is clear that the relaxation of the constraints (3.37) is not required for all pairs of scenarios in order to obtain computational efficiency. By considering scenario *clustering* the quality of the relaxation's solution is reinforced. Moreover, some efficiency in finding the solution can be lost, as we will see later.

By scenario *clustering* the original problem is decomposed in less (though bigger) subproblems, corresponding the extreme cases to the complete scenario partitioning (where each cluster is comprised by just a single scenario) and the original DEM problem with only one cluster containing all scenarios. Finding which scenario clustering is the best option is an open problem and pretty much instance dependent.

Although the criterion for scenario clustering is instance-dependent, we will favor the approach that shows higher scenario clustering for greater number of scenario groups in common, this way there are more ***non-anticipativity*** constraints to be implicitly satisfied, see Escudero et al. [97, 98, 96]. That is why we choose the ***break stage scenario cluster decomposition*** methodology for partitioning the scenario set Ω with respect to a given fixed stage, called break stage, by the relaxation of the **non-anticipativity** constraints up to that stage. The concept of break stage that was introduced in [97], see also [90].

Let Ω^c denote the set of scenarios that belong to cluster c, such that $\Omega^c \cap \Omega^{c'} = \emptyset$, $c, c' \in \mathcal{C} : c \neq c'$ and $\Omega = \cup_{c \in \mathcal{C}} \Omega^c$, and let $\mathcal{C}^r$ be the set of clusters associated to scenario group r, i.e. $p \in \mathcal{C}^r \Leftrightarrow \Omega_r \cap \Omega^p \neq \emptyset$.

Definition 4.2. A **break stage** t^* is a stage such that the set of **scenario cluster**s $\mathcal{C}$ is defined by the set of **scenario group**s $\mathcal{R}_{t^*}$ as follows: $\forall c : (c \in \mathcal{C} \Leftrightarrow \exists! r \in \mathcal{R}_{t^*} : \Omega_r = \Omega^c)$.

Model (3.35) is thus decomposed into submodels by relaxing the **non-anticipativity** constraints until the break stage t^*. Notice that for $t^* = 0$, there is only one cluster that corresponds to the original DEM. So, no decomposition takes place, all the constraints remain implicit, the FRC al-

gorithm will run a single level and the optimum solution will be found. On the opposite, for $t^* = T-1$ the original DEM is decomposed into the full scenario tree, each cluster comprises a single scenario and the FRC algorithm will run T levels. In general, for $t^* = t$ when $t > 0$, the original DEM is decomposed into $|\mathcal{R}_{t^*}|$ submodels, the FRC algorithm will run $t+1$ levels and it is easy to see that the solution found, say Z_t is such that $Z_t >= Z_{t-1}$. Thus, for higher break stages, more efficiency should be achieved (in terms of time and computational resources needed) thanks to decomposition, but worse solutions could be found. Both results have to be balanced when choosing the t^*. Several break stages will be chosen in the computational results presented in Chapter 5 to address this issue.

Given the general *multilinking compact* model for the SP problem:

$$\begin{aligned} &\min Q_E = \sum_{r\in\mathcal{R}} w_r(c_r x_r + a_r y_r) \\ &\text{s.t.} \sum_{r'\in\mathcal{V}_r} (A^r_{t(r')} x_{r'} + B^r_{t(r')} y_{r'}) = b_r \quad \forall r \in \mathcal{R} \\ &\quad x_r \in \{0,1\}, y_r \geq 0 \qquad\qquad \forall r \in \mathcal{R}, \end{aligned} \tag{4.6}$$

where c_r and a_r are the row vectors of the objective function coefficients, A^r_t and B^r_t are the constraint matrices related to stage t, b_r is the *rhs* vector, and x_r and y_r are the vectors of the variables for scenario group r, such that $c_r = c^p_t$, $a_r = a^p_t$, $A^r_t = A^p_t, B^r_t = B^p_t$ and $b_r = b^p_t$, for $r \in \mathcal{R}_t, t \in \mathcal{T}, p \in \mathcal{C}^r$.

The model to consider for each scenario cluster $p \in \mathcal{C}$ can be expressed:

$$\begin{aligned} &\min \sum_{r\in\mathcal{R}:p\in\mathcal{C}^r} w_r(c_r x_r + a_r y_r) \\ &\text{s.t.} \sum_{r'\in\mathcal{V}_r} (A^r_{t(r')} x_{r'} + B^r_{t(r')} y_{r'}) = b_r \quad \forall r \in \mathcal{R} : p \in \mathcal{C}^r \\ &\quad x_r \in \{0,1\}, y_r \geq 0 \qquad\qquad \forall r \in \mathcal{R} : p \in \mathcal{C}^r \end{aligned} \tag{4.7}$$

4.2.1 Associated models to FR level r

Here we present the models to deal with while solving a given FR level r. Notice that the **non-anticipativity** constraints are explicitly declared until stage t^*, while they keep implicit for $t > t^*$ since for $r \in \mathcal{R}_t : t > t^*, |\mathcal{C}^r| = 1$.

Mixed 0-1 model for scenario group r (*FR level r*)

The model to be solved at *FR level* r is as follows,

$$\begin{aligned}
\min \quad & \sum_{r' \in \mathcal{V}^r \cup \{r\}} \sum_{p \in \mathcal{C}^{r'}} w^p (c^p_{r'} x^p_{r'} + a^p_{r'} y^p_{r'}) \\
\text{s.t. } & A^p x^p + B^p y^p = b^p && \forall p \in \mathcal{C}^r \\
& v^p_{r'} = \overline{v}^p_{r'} && \forall p \in \mathcal{C}^r, r' \in \{\mathcal{V}_r \backslash \{r\}\} \\
& v^p_r - v^{p'}_r = 0 && \forall p, p' \in \mathcal{C}^r \\
& x^p_r \in \{0,1\} && \forall p \in \mathcal{C}^r \\
& 0 \leq x^p_{r'} \leq 1 && \forall p \in \mathcal{C}^r, r' \in \{\mathcal{V}^r \backslash \{r\}\} \\
& y^p_{r'} \geq 0 && \forall p \in \mathcal{C}^r, r' \in \mathcal{V}^r \cup \{r\}
\end{aligned} \tag{4.8}$$

where w^p is the probability attributed to p, i.e. $w^p = \sum_{\omega \in \Omega^p} w^\omega$, and $v = (x, y)$, v^p_r are such that $v = (v^p_r \ \forall r \in \mathcal{R}, \ p \in \mathcal{C}^r)$ and the values $\overline{v}^p_{r'} \ \forall r' \in \mathcal{V}_r \backslash \{r\}$ are retrieved from the solution to the models (4.8), where r is replaced by r', such that $r' \in \mathcal{V}_r \backslash \{r\}$. Notice that the **non-anticipativity** constraints are

$$v^p_r - v^{p'}_r = 0 \ \forall p, p' \in \mathcal{C}^r \tag{4.9}$$

Cluster models by relaxing the **non-anticipativity** constraints

When relaxing the **non-anticipativity** constraints from model (4.8) for FR level r, the resulting model can be split into $|\mathcal{C}^r|$ independent cluster models. Each of such models can be optimized separately and concurrently in a different process, and the *nonanticipativity* principle will be satisfied by applying the BFC algorithm. The resulting model for a cluster, say p, is as follows,

$$\begin{aligned}
\min \quad & \sum_{r' \in \mathcal{V}^r \cup \{r\} : p \in \mathcal{C}^{r'}} w^p (c^p_{r'} x^p_{r'} + a^p_{r'} y^p_{r'}) \\
\text{s.t. } & A^p x^p + B^p y^p = b^p \\
& v^p_{r'} = \overline{v}^p_{r'} && \forall r' \in \mathcal{V}_r \backslash \{r\} \\
& x^p_r \in \{0,1\} \\
& 0 \leq x^p_{r'} \leq 1 && \forall r' \in \mathcal{V}^r \backslash \{r\} : p \in \mathcal{C}^{r'} \\
& y^p_r \geq 0 && \forall r' \in \mathcal{V}^r \cup \{r\} : p \in \mathcal{C}^{r'}
\end{aligned} \tag{4.10}$$

On the LP optimal solution for TNF integer sets

The splitting variable representation for solving the LP model attached to a given TNF integer set (see Section 3.9) for FR level r can be expressed as follows,

$$
\begin{aligned}
\min \; & \sum_{r' \in \mathcal{V}^r \setminus \{r\}} \sum_{p \in \mathcal{C}^{r'}} w^p c^p_{r'} x^p_{r'} + \sum_{r' \in \mathcal{V}^r \cup \{r\}} \sum_{p \in \mathcal{C}^{r'}} w^p a^p_{r'} y^p_{r'} & \\
\text{s.t. } & A^p x^p + B^p y^p = b^p & \forall p \in \mathcal{C}^r \\
& v^p_{r'} = \overline{v}^p_{r'} & \forall p \in \mathcal{C}^{r'}, r' \in \mathcal{V}_r \setminus \{r\} \\
& x^p_r = \overline{x}^p_r & \forall p \in \mathcal{C}^r \\
& y^p_r - y^{p'}_r = 0 & \forall p, p' \in \mathcal{C}^r \\
& x^p_r \in \{0,1\} & \forall p \in \mathcal{C}^r \\
& 0 \leq x^p_r \leq 1 & \forall p \in \mathcal{C}^r, r' \in \mathcal{V}^r \cup \{r\} \\
& y^p_{r'} \geq 0 & \forall p \in \mathcal{C}^{r'}, r' \in \mathcal{V}^r \cup \{r\}
\end{aligned}
\tag{4.11}
$$

4.3 Jumping Fix-and-Relax Coordination (FRC-J)

FRC algorithm drastically reduces the computational effort for solving large-scale multi-stage mixed 0-1 SP problems. However, there are still very large-scale problems which remain hard to solve even for this algorithm. This is why we have tried an improved version of FRC. In this section we introduce such algorithm, which we have called Jumping Fix-and-Relax Coordination (FRC-J). The algorithm is based on some metaheuristics aimed at finding earlier solutions and reducing the number of visited nodes at each FRC-J level.

First, let us present the main ideas on which FRC-J is based. Our first objective is to find a feasible solution as fast as possible, i.e, we follow a *depth first* strategy. Then we try to reduce the number of visited nodes at each level and the computing time at each of those nodes.

These are the strategies to follow at each FRC-J level:

1. At each iteration of the Jumping Branch-and-Fix Coordination (BFC-J), when branching, all those variables not yet branched that follow the **non-anticipativity** constraints are fixed to their values (in some sense, we could say that their corresponding TNFs are *jumped down* the BF tree). This *jumping down* strategy aims to find fastly a feasible solution to get an incumbent that will help pruning as many TNFs as possible.

2. Then, when the branching value is selected, instead of ordering variables in a non-increasing order of the μ-parameter, as in Section 4.1.6, an increasing ordering is done, and thus a variable

with the least frequent value is selected (i.e the *largest small deterioration* criterion):

$$h_{<i>} = argmin_{j \in \mathcal{P}_r}\Big\{\mu_{ri}\Big\} \quad s.t. \quad \mu_{ri} = \min\Big\{\sum_{p \in \mathcal{C}^r} \overline{x}^p_{ri},\ |\mathcal{C}^r| - \sum_{p \in \mathcal{C}^r} \overline{x}^p_{ri}\Big\} \quad \forall i \in \mathcal{P}_r.$$

.

The branching parameter $\lambda_{rh_{<i>}}$ is, as before,

$$\lambda_{rh_{<i>}} = \begin{cases} 0, & \text{if } \sum_{p \in \mathcal{C}^r} \overline{x}^p_{rh_{<i>}} \leq |\mathcal{C}^r| - \sum_{p \in \mathcal{C}^r} \overline{x}^p_{rh_{<i>}} \\ 1, & \text{otherwise.} \end{cases}$$

3. Once the branching variable has been selected, it is fixed to $x_{rh_{<i>}} = \lambda_{rh_{<i>}}$. Then, there is no need to find again the MIP solution for those cluster models (4.10) for which the chosen variable already takes on that value ($\overline{x_{rh_{<i>}}} = \lambda_{rh_{<i>}}$) at the last TNF. Then, only a few (half, at most) of the models need to be solved again, saving a lot of computational effort at this point. Moreover, for the way the branching variable and its value are chosen ($h_{<i>}$ and λ_{ri}, see item 2), only the minimum number of such models need to be solved again.,

4. Finally, to help reducing drastically the number of visited nodes, when pruning, all those variables that were before *jumped down*, are now *jumped back* and the previously branched variable is fixed to the opposite value. Notice that doing this *backjumping* eludes searching some branches of the *BF tree* and this excludes the guarantee of finding the optimal solution at each FRC level since it can be skipped, though as we will see later, in the testbed we have worked with in the computational experience, when there is a GAP between FRC-J and FRC, it is very low.

Besides, we are going to use a stack, say S, that serves as a collection of elements, with two principal operations: push, which adds an element i to the collection (let us symbolize it by $S \rightarrow push(i)$), and pop ($S \rightarrow pop(i)$), which removes the last element that was added (this behavior is usually coined as LIFO (last in, first out)). Since the new branching strategy will imply fixing all those variables that already follow the **non-anticipativity** constraints, and the selected variable $h_{<i>}$, this variable will be pushed to the stack. This way, when pruning, the last selected TNF can be retrieved and the previously *jumped* TNFs can be *jumped* back.

Now let us present the FRC-J procedure:

Procedure *FRC-J*

$Z_{MIP}^{FRC} := 0$, $t := 1$

While $t \leq t^*$ do:

For $r \in \mathcal{R}_t$, do:

Build *FR level* model MIP_r (4.8), where the continuous and 0-1 variables for the ancestor *FR levels* are fixed, the 0-1 variables for the model associated with scenario group r are restricted to be 0-1 valued, the integrality constraint is relaxed for the 0-1 variables of the successor scenario groups, and the **non-anticipativity** constraints for the successor scenario groups are also relaxed.

Obtain Z_{MIPr} by executing the Subprocedure for *FR level* r

If $Z_{MIPr} = +\infty$ then stop

If $t = t^*$ then $Z_{MIP}^{FRC} := Z_{MIP}^{FRC} + Z_{MIPr}$

Endfor

$t := t + 1$

Endwhile

Subprocedure for *FR level* r

Step 1: Initialize $i := 0$, $Z_{MIPr} := +\infty$, the stack of fixed variables $S = \emptyset$.

Step 2: Solve the MIP $|\cup_{r\in\mathcal{R}^r} \mathcal{K}^r|$ models (4.10). It can be done in parallel. If the constraints (4.9) are satisfied, then update the solution value Z_{MIPr} and the related solution and return to the main program.

Step 3: Branch all the k TNFs fixing the variables that follow the **non-anticipativity** constraints. Update $S \rightarrow push(i)$ and $i := i + k + 1$.

Step 4: Set $h_{<i>} = \text{argmin}_{j\in\mathcal{P}_r}\{\mu_{rj},\}$, such that variable j has not been previously branched on, nor fixed at in the current branching path.

Branch $x_{rh_{<i>}} := \lambda_{rh_{<i>}}$, for all scenario in scenario group r. Let us remind that there will be no need to solve again the model for the majority of the scenarios.

Step 5: Solve the MIP problems $MIP_r^p \; \forall p \in \mathcal{C}^r$ and compute $Z'_{MIPr} = \sum_{p \in \mathcal{C}^r} MIP_r^p$. It can be done in parallel.

If $Z'_{MIPr} \geq Z_{MIPr}$ then goto Step 7.

If there is any 0-1 variable in set $\mathcal{P}_r$ that takes different values $\overline{x}_{ri}^p$ for some of the scenario models (4.10) in group r then goto Step 3.

If all the y variables in set $\mathcal{P}_r$ take the same value for all scenario model (4.10) in group r, then update $Z_{MIPr} := Z'_{MIPr}$, and goto Step 7.

Step 6: Solve LP model (4.11) for satisfying the constraints (4.9) for the *common* y variables in given *FR level* r. Notice that the solution value is denoted by Z_{LP}^{TNF}.

Update $Z_{MIPr} = min\{Z_{LP}^{TNF}, Z_{MIPr}\}$.

Step 7: Prune the current branch.

If $x_{rh_{<i>}} = \lambda_{rh_{<i>}}$, then goto Step 10.

Step 8: Update $S \rightarrow pop(i)$.

If $i = 0$ then save the solution value Z_{MIPr} and the related solution, if any, and return.

Step 9: If $x_{rh_{<i>}} = 1 - \lambda_{rh_{<i>}}$, then goto Step 8.

Step 10: Branch $x_{rh_{<i>}} := 1 - \lambda_{rh_{<i>}}$, for all scenarios in group r.

Goto Step 5.

4.4 Parallel Jumping Fix-and-Relax Coordination (FRC-PJ)

At this section we present the Parallel Jumping Fix-and-Relax Coordination, a parallel version of the sequential FRC-J algorithm presented in Section 4.3. As we previously mentioned, the process of solving the submodels (4.10) can be parallelized, allowing an important reduction of the computing time needed to solve the algorithm.

Several processes will be working concurrently, all of them performing the same type of tasks except for one special process, which we will call *Master*. The Master will coordinate the whole execution of the program, deciding which subproblems are to be solved, which variables to be fixed, when and how to prune, and so on. Meanwhile, for each scenario cluster to be considered, there is a process dedicated to solve that cluster's model.

4.4.1 Speedup

Let us try to analyze the *speedup* of the FRC-PJ algorithm: Firstly, simplifying the most, let us suppose that all the subproblems were equally hard to solve and it took us the same computing time to solve each of them, and suppose too that this were all the computational effort that we need to take into account, while the time needed to complete the instructions of the algorithm and the extra computing effort added to allow the communications between the different processes were negligible. Then we could conclude that the *speedup* achieved by parallelizing the FRC-J algorithm would be strictly proportional to the number of processes we used, i.e., if there were e.g. $|\mathcal{C}| = 10$ clusters, it would take FRC-PJ a tenth of the time needed by FRC-J to solve the main problem. Of course this is the ideal situation and in the real world we will get lower efficiency.

To dig a bit deeper on the analysis, let us defined:

- t_{jf} and t_{pjf}, respectively, time needed to solve the problem by the FRC-J and the FRC-PJ algorithms.
- t_{np}, total computing time needed to carry out the not *parallelizable* instructions when solving a particular problem by the FRC-J algorithm,
- t_p, time added by the extra instructions needed to communicate and coordinate the different processes by the FRC-PJ.
- $\mathcal{C}^i \subset \mathcal{C}$, set of clusters that need to be solved again at iteration i
- t_i^p, computing time needed to solve the subproblem for cluster $p \in \mathcal{C}$ at iteration $i \in \{1 \dots n\}$, where we suppose that FRC-J needs to solve the $|\mathcal{C}|$ subproblems n times.

Then, $t_{jf} = t_{np} + \sum_{0 \leq i \leq n} \sum_{p \in \mathcal{C}^i} t_i^p$. If for each cluster, times were similar ($t_i^p \backsimeq t_i^{p'} \forall p \in \mathcal{C}, i \in \{1, \dots, n\}$), then, reminding that for some iterations in FRC-J, at least half of the subproblems do not need to be solved again, let us approximate:
$t_{jf} = t_{np} + \sum_{0 \leq i \leq n} \sum_{p \in \mathcal{C}} \frac{t_i^p}{2}$

For FRC-PJ we get the following computing time:
$t_{pjf} = t_{np} + t_p + \sum_{0 \leq i \leq n} \max_{p \in \mathcal{C}} \left\{ t_i^p \right\}$.

Parallelization efficiency depends on t_{np} and t_p values, which are usually low, but for small problems they may be significant. Notice that some of the subproblems may take a lot longer to

be solved than the other ones (we have observed this behavior in the testbed we present in the next chapter), in such cases, it is possible that $t_i^p \geq \sum_{p \in \mathcal{C}^i} t_i^p$. So the efficiency will highly depend on the variability of the computing times needed to solve the different subproblems, the more heterogeneous, the less efficient will be the parallel algorithm.

Anyway, we have explored two ways to tackle this problem and improve the *speedup* of the parallelization:

- Instead of keeping the fastest processes idle, waiting for the other processes to finish their calculation for solving their problems, we could try to go on branching or pruning. However, the slight improvement this strategy could deliver does not make up for the difficult, almost unmanageable, overhead that it would add to the algorithm.
- Another strategy could be setting $\lambda_{rh_{<i>}} = \overline{x}_{ri}^p$ s.t. $p = \max_{k \in \mathcal{C}}\{t_i^k\}$. So, the slowest processes would remain idle whenever possible. However, it is not easy to predict which will be the slowest subproblem to solve a priori. We tackled this problem by using the first iteration time as a reference, but usually, the first solution is the one that takes longer to find, and once we fix one (some) variable(s) and solve again, CPLEX manages to find a solution quite faster. Moreover, after fixing some variables, that *slowest* problem may become the *fastest* one. To summarize, we have tried this strategy with no significant *speedup* improvement, but we believe it should be explored deeper in a later work, particularly involving larger problems that really represent a challenge for the FRC-PJ algorithm.

4.4.2 FRC-PJ

Let us introduce the FRC-PJ procedure. By slightly abusing the notation, let us use cluster p to refer to the process that works with its model, as well as scenario group r to refer to the corresponding level of the algorithm. At a given moment, every process p may be working on a different level r, let us call it r^p. The steps executed by all processes but by the Master are marked with a " ' " (as in *Step 3'*), while the steps not marked (as in *Step 3*) are all executed by the Master except the first step, which is executed by all processes.

Procedure *FRC-PJ* for *FR level* r

Step 1: Initialize MPI environment and variables.

Step 2': Each process reads the particular instance of its subproblem and initializes variables. Set $r^p := 0$.

Step 3: Master waits for any message from the other processes.

Step 3': Each one of the processes $p \in \mathcal{C}^r$ solves its MIP_r^p problem (4.10) concurrently. When finished solving sends the solution to the Master.

Step 4: Master receives the message from a process, say p. If the subproblem was infeasible prune the current branch, i.e., goto Step 10. Otherwise save the solution found and update the value of $Z'_{MIPr} += Z^p_{MIPr}$. If $Z'_{MIPr} \geq Z_{MIPr}$ then prune, i.e. goto Step 10. If not all the processes working on level r^p have yet sent their solutions for the current TNF, goto Step 3. Check if constraints (4.9) are satisfied, and if so, goto Step 8. Otherwise, branch the current TNF:

Step 5: Branch all the k TNFs fixing the variables that follow the **non-anticipativity** constraints. Update $S \rightarrow push(i)$ and $i := i + k + 1$.

Step 6: Set $h_{<i>} = \text{argmin}_{j \in \mathcal{P}_r}\{\mu_{rj},\}$, such that variable j has not been previously branched on, nor fixed at in the current branching path. Branch $x_{rh_{<i>}} := \lambda_{rh_{<i>}}$,. Set $Z'_{MIPr} := 0$. Send a message to all clusters in scenario group r^p. Goto Step 3.

Step 7': All clusters in scenario group r^p shall fix the $k + 1$ variables mentioned in the previous step on their respective models, and goto Step 3'.

Step 8: If all the y variables in set $\mathcal{P}_r$ take the same value for all scenario model (4.10) in group r, then update $Z_{MIPr} := Z'_{MIPr}$, and goto Step 10.

Step 9: Solve LP model (4.11) for satisfying the constraints (4.9) for the *common* y variables in given *FR level* r. Notice that the solution value is denoted by Z^{TNF}_{LP}. Update $Z_{MIPr} = min\{Z^{TNF}_{LP}, Z_{MIPr}\}$.

Step 10: Prune the current branch.

If $x_{rh_{<i>}} = \lambda_{rh_{<i>}}$, then goto Step 13.

Step 11: Update $S \rightarrow pop(i)$.

If $i = 0$ then save the solution value Z_{MIPr} and the related solution, if any, and return.

Step 12: If $x_{rh_{<i>}} = 1 - \lambda_{rh_{<i>}}$, then goto Step 11.

Step 13: Branch $x_{rh_{<i>}} := 1 - \lambda_{rh_{<i>}}$, send the message to all scenarios in group r. Goto Step 3.

Step 14': All clusters in scenario group r^p shall unfix the variables *jumped* in the previous step on their respective models and fix the variable $x_{rh_{<i>}} := 1 - \lambda_{rh_{<i>}}$, and goto Step 3'.

Chapter 5
Computational Experience

In this chapter, a broad computational experience is presented for assessing the quality of the algorithm frameworks introduced in previous chapter. A set of computational experiments have been run for large-scale instances of the NP-hard Multistage Stochastic Facility Location Problem (**MSFLP**) The remainder of this chapter is organized as follows: The Multistage Stochastic Facility Location Problem is firstly introduced in Section 5.1. Section 5.2 formally states the **MSFLP** and introduces a pure $0-1$ DEM associated with it and Section 5.3 reports computational experience, for the algorithms presented, solving a testbed randomly generated.

5.1 Multi-period location-allocation problem under uncertainty

Discrete facility location decisions can be planned as a set of sequential actions to be implemented at different moments of a given time horizon. Multi-period location problems look for sequential location/allocation decisions that fulfil certain coverage levels of demand in some places at each time period. When focusing on essential services, population demand must be serviced from the first time period. However, some applications have been found in which non-essential facilities have to be located, and full coverage is only required at the end of the planning horizon. This is the case, for instance, for the location of libraries, nursing homes, kinder gardens, parking lots, supermarkets, banks, etc. Usually, in these cases budget limitations prevent from imposing full coverage from the first time period, and minimum coverage levels at the different time periods are imposed instead. Different types of multi-period facility location problems have been studied in the literature by numerous authors. The interested reader is referred to [10, 12, 155, 136, 170, 137, 109, 243, 171, 230, 237], to mention just a few.

In this problems, there are several elements that evolve with time, like costs, availability of resources or demands. Historical data is typically used to forecast their values but the fact is that the actual behavior of the system is not deterministic and should be addressed as Stochastic Programming (SP) problems. In the **MSFLP**, we are given a set of potential facilities (e.g., production plants) and a set of customers. At each time period, customers demand of service must be satisfied from an open facility. Thus, at each time period, two types of decisions must be made: the location of the facilities to open and the allocation of customers to open facilities. Uncertain data include facility set-up and maintenance costs as well as customers assignment costs. Furthermore, requests for service from the customers, as well as the minimum number of facilities to open, and the minimum number of allocated customers for open facilities are also uncertain. The objective in the **MSFLP** is to minimize the overall expected costs (i.e. we consider a risk neutral strategy), which in our case, in addition to the above mentioned costs, include penalties for unsatisfied service requests.

The deterministic version of the **MSFLP** is already NP-hard (see [10, 12]). Therefore, the **MSFLP** is an interesting and difficult application of Stochastic Integer Programming since, in practice, very frequently it is an stochastic problem. See [10] for an extensive computational comparison of three formulations for the deterministic version of the **MSFLP**, where the formulation using impulse-step variables produced the best results, that is why this is the formulation used herein.

5.2 Multistage Stochastic Facility Location Problem

In the **MSFLP** a set of *facilities* must be selected (opened) from a given set of potential spots to give service to a given set of *customers*. Let $\mathcal{I}$ denote the index set of facilities, $\mathcal{J}$ the index set of customers, and $\mathcal{T}$ the index set of time periods. Next we describe the modeling hypotheses and the notation:

- At each period $t \in \mathcal{T}$ a decision must be made on the set of facilities to open. Once a facility is opened it remains open until the end of the time horizon.
- There is a *latency*, τ, that represents the number of periods required to make available a facility from the moment it is decided to open it. Throughout we will distinguish between the time period when the decision is made to open a facility and the time period when it becomes available. We assume the latency is the same for all facilities over the time horizon. We further assume that the decision to open the facilities to be available at the first $\tau - 1$ periods in the time horizon has been made before the beginning of it. Notice that, otherwise, no facility would be

available during the first $\tau - 1$ periods. Additionally, let $\mathcal{T}^* \subseteq \mathcal{T}$ and the subset $\mathcal{R}^* \subseteq \mathcal{R}$ of scenario groups such that $\mathcal{T}^* \equiv \mathcal{T} \setminus \{T - \tau + 1, ..., T\}$ and $\mathcal{R}^* \equiv \cup_{t \in \mathcal{T}^*} \mathcal{R}_t$.

- The index set of facilities that are open at some period before the beginning of the time horizon is denoted by $\mathcal{I}^- \subseteq \mathcal{I}$.
  For each $i \in \mathcal{I}^-$, t_i, $(-\tau \leq t_i \leq -1)$ denotes the period (before the beginning of the time horizon) at which it was decided to open facility i.
  Furthermore, for each $i \in \mathcal{I}$, $\mathcal{R}_i^+$ denotes the index set of nodes of the scenario tree where it can be decided to open facility i. This definition depends on whether or not facility i is open before the beginning of the time horizon. That is:

$$\mathcal{R}_i^+ = \begin{cases} \mathcal{R}^* & \forall i \in \mathcal{I} \setminus \mathcal{I}^-, \\ \mathcal{R}^* \cup \{t_i, t_i + 1, ..., -1\} & \forall i \in \mathcal{I}^-. \end{cases}$$

- Customers are progressively assigned to open facilities. At each period $t \in \mathcal{T}$ a minimum number of customers must be assigned to each open facility. Once a customer is assigned it must continue to be assigned in all subsequent periods, although the customer assignment may change from period to period. Moreover, at a given period a customer cannot be assigned to more than one facility. All customers must be assigned at the end of the time horizon.

- Each customer may demand service at any subset of periods. These requests will only be served if the customer is already assigned. Assigned customers do not necessarily have demand at all periods after their first assignment.

- There are set-up and maintenance costs for the open facilities. Assigning a customer to a facility at a given period incurs a cost, even if the customer does not have demand at that period. In addition, a penalty is paid for not servicing unassigned customers with demand. For $j \in \mathcal{J}$, let ρ_j denote the penalty for not servicing the demand of customer j.

- Service requests are assumed to be binary, uncertain and independent. In addition to the demand, other parameters can also be uncertain as the minimum number of customers to be assigned to a facility at each period, the minimum number of facilities to be opened at each period, and the set-up, maintenance and assignment costs. In particular, the following data are assumed to be uncertain:

  d_j^r : coefficient that takes the value 1 or 0 depending on whether or not customer j has demand at period $t(r)$ under scenario group r, $\forall j \in \mathcal{J}$, $r \in \mathcal{R}^-$.

  n^r : minimum number of customers to be serviced at period $t(r)$ under scenario group r, $\forall r \in \mathcal{R}^-$.

ℓ_i^r : lower bound on the number of customers to be assigned to facility i, if it is available at period $t(r)$ under scenario group r, $\forall r \in \mathcal{R}^-$.

m^r : minimum number of facilities to be opened at period $t(r)$ under scenario group r, $\forall r \in \mathcal{R}^*$.

$f_i^{S_r}$: set-up cost for facility i at period $t(r)$ under scenario group r, $\forall i \in \mathcal{I}, r \in \mathcal{R}_i^+$.
Note: As mentioned before, the facilities opened at period $t(r)$ will become available at period $t(r)+\tau$ (thus, at the nodes $\mathcal{R}_{t(r)+\tau}$ of the scenario tree).

$f_i^{M_r}$: maintenance cost for facility i at period $t(r)$ under scenario group r, $\forall i \in \mathcal{I}, r \in \mathcal{R}^-$.
Note that, if it is decided to open a facility at time period t, then its maintenance costs will be incurred from period $t+\tau$ to period T.

c_{ij}^r : assignment cost of customer j to facility i under scenario group r, $\forall i \in \mathcal{I}, r \in \mathcal{R}^-$.

The objective in the **MSFLP** is to minimize the expected overall cost throughout the time horizon. This cost includes facilities set-up and maintenance costs, as well as assignments costs and penalties for unserved customers.

We have followed the so-called *scenario dependent location decision*, where location decisions are gradually made along the planning horizon. This strategy is suitable for situations where both the location and the assignment decisions are considered operational, and it leads to a multi-stage stochastic programming model. This allows high flexibility in the decision process but, on the other hand, comes at the expenses of a DEM which is difficult to solve.

5.2.1 0-1 DEM

Next, we present a DEM for the scenario-dependent location strategy for the **MSFLP** that uses a mixture of impulse and step variables (see [10] and references therein), which is the combination of variables that gave the best numerical results for the deterministic version of the problem. In particular, let us the following sets of binary variables:

$$y_i^r = \begin{cases} 1, & \text{if } \textbf{by} \text{ period } t(r) \text{ it has been decided} \\ & \text{to open facility } i, \text{ under scenario group } r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I}, r \in \mathcal{R}_i^+$$

and

$$x_{ij}^{r} = \begin{cases} 1, & \text{if customer } j \text{ is assigned to facility } i \\ & \text{at period } t(r), \text{ under scenario group } r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, r \in \mathcal{R}^{-}.$$

Remarks:

1. For coherence, we fix $y_i^{t_i} = 1, \forall i \in \mathcal{I}^{-}$ and then, $y_i^r = 1, \forall r \in \mathcal{R}_i^{+}$. Additionally, we consider that $y_i^{\gamma(0)} = 0, \forall i \in \mathcal{I} \setminus \mathcal{I}^{-}$.

2. Recall that $y_i^r = 1$ means that facility i is open by period $t(r)$, but does not necessarily mean that it is yet available at that period.

The compact representation of the **MSFLP** for the scenario-dependent location strategy is as follows,

$$\min \sum_{i\in\mathcal{I}^-} f_i^{S_0} y_i^0 + \sum_{r\in\mathcal{R}^*}\Big[p^r \sum_{i\in\mathcal{I}\setminus\mathcal{I}^-} f_i^{S_r}(y_i^r - y_i^{\gamma(r)}) + \sum_{i\in\mathcal{I}}\Big(\sum_{r':r=\gamma^\tau(r')} p^{r'} f_i^{M_{r'}}\Big) y_i^r\Big] + \sum_{r\in\mathcal{R}^-} p^r \sum_{j\in\mathcal{J}}\Big(\sum_{i\in\mathcal{I}} c_{ij}^r x_{ij}^r + \rho_j d_j^r\big(1 - \sum_{i\in\mathcal{I}} x_{ij}^r\big)\Big) \quad (5.1)$$

subject to

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} x_{ij}^r \geq n^r \qquad \forall\, r\in\mathcal{R}^- \quad (5.2)$$

$$\sum_{j\in\mathcal{J}} x_{ij}^r \geq \ell_i^r y_i^{\gamma^\tau(r)} \qquad \forall\, i\in\mathcal{I}, r\in\mathcal{R}^- : \gamma^\tau(r)\in\mathcal{R}_i^+ \quad (5.3)$$

$$\sum_{i\in\mathcal{I}} x_{ij}^r \leq 1 \qquad \forall\, j\in\mathcal{J}, r\in\mathcal{R}^- : t(r)\in\mathcal{T}\setminus\{T\} \quad (5.4)$$

$$\sum_{i\in\mathcal{I}} x_{ij}^r = 1 \qquad \forall\, j\in\mathcal{J}, r\in\mathcal{R}_T \quad (5.5)$$

$$\sum_{i\in\mathcal{I}} x_{ij}^{\gamma(r)} \leq \sum_{i\in\mathcal{I}} x_{ij}^r \qquad \forall\, j\in\mathcal{J}, r\in\mathcal{R} : t(r) > 1 \quad (5.6)$$

$$x_{ij}^r \leq y_i^{\gamma^\tau(r)} \qquad \forall j\in\mathcal{J}, i\in\mathcal{I}, r\in\mathcal{R}^- : \gamma^\tau(r)\in\mathcal{R}_i^+ \quad (5.7)$$

$$x_{ij}^r = 0 \qquad \forall j\in\mathcal{J}, i\in\mathcal{I}, r\in\mathcal{R}^- : \gamma^\tau(r)\notin\mathcal{R}_i^+ \quad (5.8)$$

$$\sum_{i\in\mathcal{I}\setminus\mathcal{I}^-} (y_i^r - y_i^{\gamma(r)}) \geq m^r \qquad \forall r\in\mathcal{R}^*\setminus\{0\} \quad (5.9)$$

$$\sum_{i\in\mathcal{I}\setminus\mathcal{I}^-} y_i^0 = m^0 \quad (5.10)$$

$$y_i^{\gamma(r)} \leq y_i^r \qquad \forall\, i\in\mathcal{I}\setminus\mathcal{I}^-, r\in\mathcal{R}^*\setminus\{0\} \quad (5.11)$$

$$y_i^r = 1 \qquad \forall i\in\mathcal{I}^-, r\in\mathcal{R}_i^+ \quad (5.12)$$

$$x_{ij}^r \in \{0,1\} \qquad \forall\, i\in\mathcal{I}, j\in\mathcal{J}, r\in\mathcal{R}^- \quad (5.13)$$

$$y_i^r \in \{0,1\} \qquad \forall\, i\in\mathcal{I}\setminus\mathcal{I}^-, r\in\mathcal{R}_i^+. \quad (5.14)$$

The expression (5.1) consists of minimizing the total expected location-allocation cost over the scenarios, according to a risk neutral strategy. Constraints (5.2) force that a minimum number of customers is assigned at period t, for $t\in\mathcal{T}^-$. Constraints (5.3) guarantee that a minimum number of customers are assigned to each available facility at each period. Constraints (5.4) ensure that, at each period, each customer is assigned to one facility at the most, whereas Constraints (5.5) force all customers to be assigned at the end of the time horizon. By constraints (5.6), a customer already assigned

at period $t-1$ will also be assigned at period t, although the facility to whom it is assigned is not necessarily the same at both periods. Constraints (5.7) and (5.8) model that, at any period, customers can be assigned only to available facilities. Constraints (5.9) guarantee that a minimum number of facilities are opened from a time period to the next one along the time horizon $\mathcal{T}^*$. Constraint (5.10) sets the number of facilities m^0 to be opened at period $t = 0$. Finally, constraints (5.11) and (5.12) ensure that if a facility is open at a given period t, then it remains open at all subsequent periods.

5.3 Computational experience

To illustrate the performance and assess the quality of our algorithms we have run a series of computational experiments based on . For these experiments we have implemented the algorithms in C++, using the MPI libraries for the parallel code. We run all programs in ***Solstorm***, a cluster placed in the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway. Solstorm has 2732 CPUs of different types (132 cores at 1.6GHz Intel, 192 cores at 3.0GHz Intel, 984 cores at 2.4GHz Opteron, 1536 cores at 2.2GHz Opteron). CPLEX is used as an auxiliary LP/MIP solver, particularly its version v12.61. In all the experiments a computing time limit of 48h was set. An issue we have had to deal with is that the computational resources of this cluster are shared and, hence, the elapsed times vary from one execution to another. Thus, the algorithms have been run twice and written down the media for each case in the testbed. For each process running FRC-PJ, a single core has been assigned. Likewise, CPLEX was run on a single core.

5.3.1 Testbeds description

To test the formulations and the algorithm, a series of instances have been used, taken from the computational experience reported in Albareda-Sambola et al. [11], where stochastic optimization is used to address a multistage location problem for the first time, to the best of our knowledge. For that work, the computational experiments were conducted in a different type of machine, particularly a PC Intel Core 2 Duo, 2.60 GHz, 3 Gb RAM, Microsoft Visual Studio C++ compiler v6.0., and a computing time limit of 24h was set. These instances were randomly generated. In the testbed, the demands of the customers (d_j^r) and the number of customers that need to be serviced at each time period (n^r) vary among scenario groups in the same time period, while the remaining uncertain parameters (facility and distribution costs, and lower limit on the number of assigned customers to each facility, if open) are generated as deterministic data (i.e., for all scenario groups $r, r' \in \mathcal{R}_t$ then $f_i^{S_r} = f_i^{S_{r'}}$, $f_i^{M_r} = f_i^{M_{r'}}$, etc.).

Four time periods are considered, except for the last three instances, that include five time periods. The number of customers ranges in $\{25, 50, 75\}$, and the number of facilities in $\{8, 10, 15\}$ (see Table 5.1). Also, in all cases, the latency is taken to be $\tau = 1$ since, for the considered lengths of the time horizon, larger latencies would require to make about half of the location decisions a priory regardless the location strategy.

Facility setup costs ($f_i^{S_r}$) are uniformly drawn from $[200, 400]$, while allocation costs (c_{ij}^r) are drawn from $[10, 100]$ and maintenance costs ($f_i^{M_r}$) are uniformly drawn from $[0, 15\frac{|\mathcal{I}|}{T}]$. Penalties for not serving a customer: $\rho_j = \frac{1}{T} \sum_{t \in \mathcal{T}} \max_{i \in \mathcal{I}} c_{ijt}$. To generate the rest of the data, a tentative number of open facilities at each time period was first generated, as $\pi^t \in \{1, \ldots, \lfloor 2 \cdot |\mathcal{I}|/T \rfloor\}$ for $t \in \mathcal{T}$. If $\sum_{t \in \mathcal{T}} \pi^t > |\mathcal{I}|$, the set of values was discarded and regenerated. The minimum number of customers to be assigned at each time period has been generated as $n^r = \left\lfloor (\pi^1 + \cdots + \pi^t)/(\pi^1 + \cdots + \pi^T) \right\rfloor \cdot |\mathcal{J}|$
(Note that, for r with $t(r) = T$ this expression leads to $n^r = |\mathcal{J}|$). Minimum number of facilities to be opened at each time period: $m^r = 1$. Probability of demand: the same probability of demand has been considered for all customers at all time periods. This probability, q, ranges in $\{0.1, 0.5, 0.9\}$ and its value is given in Table 5.1 for all the generated instances.

To build the scenario trees we proceeded as follows. From each node, starting from the root, the number of successor nodes have been randomly selected from $\{1, 2, 3\}$. Then, at the first successor the customer demands d_j^r were randomly generated from a Bernoulli distribution with probability q defined above. In the second and third successors (if they exist) a larger and a smaller probability, respectively, was used. The actual probability of each set of demands is computed assuming that customers requests follow independent Bernoulli variables with probability q and a total weight of 1 is proportionally distributed among the successor nodes. For each set of costs three different regular scenario trees have been generated.

Among the different possible alternatives for the set-up and maintenance facility costs, as well as the minimum number of facilities to open at each time period, the one where the user considers their expected values has been selected, computed as follows:

$$\overline{f}_i^{St} = \sum_{r \in \mathcal{R}_t} p^r f_i^{S_r} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}_i^+$$

$$\overline{f}_i^{Mt} = \sum_{r \in \mathcal{R}_t} p^r f_i^{M_r} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \setminus \{0\}$$

$$\overline{m}^t = \max_{r \in \mathcal{R}_t} \{m^r\} \qquad \forall t \in \mathcal{T}^*$$

Table 5.1: Testbed. Problem and model dimensions.

$Instance$	$\|\mathcal{J}\|$	$\|\mathcal{I}\|$	$\|\mathcal{T}\|$	q	$\|\Omega\|$	$\|\mathcal{R}\|$	n	m	nel	dens (%)
C9-T4	75	15	6	0.9	94	153	171885	196917	1092525	ϵ
C9-T3	75	15	6	0.9	158	250	281505	322533	1785585	ϵ
C9-T2	75	15	6	0.9	153	248	279300	320004	1775610	ϵ
C8-T4	75	15	6	0.5	141	223	250980	287608	1593330	ϵ
C8-T3	75	15	6	0.5	268	429	483915	554541	3076035	ϵ
C8-T2	75	15	6	0.5	117	188	211440	242271	1343520	ϵ
C7-T4	75	15	6	0.1	112	175	196695	225365	1246095	ϵ
C7-T3	75	15	6	0.1	98	158	177525	203384	1127535	ϵ
C7-T2	75	15	6	0.1	120	186	209115	239606	1324065	ϵ
C6-T4	50	10	6	0.9	165	267	134020	163273	851210	ϵ
C6-T3	50	10	6	0.9	213	339	170260	207481	1080890	ϵ
C6-T2	50	10	6	0.9	122	201	100790	122787	641130	ϵ
C5-T4	50	10	6	0.5	95	151	75560	92061	479210	0,01
C5-T3	50	10	6	0.5	147	233	116860	142441	742230	ϵ
C5-T2	50	10	6	0.5	203	319	160160	195161	1015290	ϵ
C4-T4	50	10	6	0.1	292	450	226080	275575	1432280	ϵ
C4-T3	50	10	6	0.1	167	266	133490	162677	847580	ϵ
C4-T2	50	10	6	0.1	184	297	149130	181705	947450	ϵ

As we stated above, the criterion for clustering scenarios in sets is instance dependent, and a trade-off between the size of model (4.10) and the number of NAC (4.9) is needed. Moreover, it has to be taken into account that when the number of scenario groups in common for the scenarios in the same cluster increases, the results tend to improve (see [151]).

5.3.2 Numerical results

Table 5.1 show the dimensions and probabilities of demand of our medium and large scale instances included in the Testbed as well as the dimensions of the associated DEMs. The headings are as follows: $xx - yy$, name of the instance, where xx is the case number and yy is the scenario tree number (see Section 5.3.1 for Testbed trees); $|\mathcal{J}|$, number of customers; $|\mathcal{I}|$, number of facilities; $|\mathcal{T}|$, number of periods in the time horizon; q, probability of demand; $|\Omega|$, number of scenarios under consideration; $|\mathcal{R}|$, number of scenario groups; n, number of (0-1) variables; m, number of constrains; nel, number of nonzero coefficients in the constraint matrix; and $dens$, constraint matrix density (in $\%$). An entry of ϵ in the $dens$ column indicates that the density of the instance is below a threshold value of 0.01.

Tables 5.2-5.5 show the main computational results for the instances presented in Table 5.1.

Let us present them in turn.

Analysis of the dimensions and results of Tables 5.1-5.2

Table 5.2 compares the solution values and elapsed times between CPLEX and FRC. The headings are as follows: Z_{LP}, solution value of the LP relaxation of the DEM; $\overline{Z}^{CPLEX}$ and $\overline{Z}^{FRC}$, solution values (i.e., expected total cost) of the incumbent solutions obtained by CPLEX and our FRC algorithm for DEM, respectively; T^{CPLEX} and T^{FRC}, elapsed times (secs) to obtain the associated incumbent solutions; GG, goodness gap between the solution values $\overline{Z}^{FRC}$ and $\overline{Z}^{CPLEX}$, defined as $\frac{\overline{Z}^{FRC}-\overline{Z}^{CPLEX}}{\overline{Z}^{CPLEX}}$ (in $\%$); TG, time gap between T^{CPLEX} and T^{FRC}, defined as $\frac{T^{CPLEX}}{T^{FRC}}$; GC and GF, quasi-optimality gaps for the solution values obtained by CPLEX and FRC, defined as $\frac{\overline{Z}^{CPLEX}-Z_{LP}}{Z_{LP}}$ (in $\%$) and $\frac{\overline{Z}^{FRC}-Z_{LP}}{Z_{LP}}$ (in $\%$), respectively; t_f^*, break stage used to decompose the original DEM into clusters for that particular instance to be solved by the FRC; nc_f, number of clusters in which the original DEM has been decomposed for that particular instance to be solved by the FRC algorithm.

Notice that $GC = 0$ would mean that the optimal solution is obtained by CPLEX, and $GG = 0$ would mean that both approaches being compared in the corresponding table, have obtained the same solution value. As we have explained above, the break stage t_f^* determines the number of clusters and, for the FRC algorithm (as well as for FRC-J and FRC-PJ,as we will see later), there is a trade-off between the quality of the solution and the elapsed time. For higher t_f^*, lower computing times are needed, but on the other hand worse solutions are found (expressed by higher goodness gap, GG). This is why we have tried different break stages for each instance, and we have selected the solutions that we consider give the better trade-off between the quality of the solution and the computing time. Given the small values of GG and the highly positive value of TG for most of the instances where a CPLEX solution is given, it seems that the FRC option is the most reasonable one.

The cases C4 to C6, with 3 rows each, correspond to medium-size instances with 10 facilities and 50 customers, whose uncertainty in the main parameters is represented by up to 292 scenarios. Their associated DEM have up to 226000 (binary) variables and over 275000 constraints. CPLEX proves the optimality of the solution in all the 9 instances by requiring up to 13854 of elapsed time (instance C6-T3). Notice that this instance has one of the largest probability of demand among the customers. FRC obtains very frequently near-optimal solution with an optimality gap not higher than 3.6$\%$. However, the elapsed time requirements of FRC are very small, being usually one order of magnitude smaller than those required by CPLEX, reaching an outstanding improvement of $TG =$

Table 5.2: Computational results. CPLEX vs FRC

Instance	Z_{LP}	$\overline{Z}^{CPLEX}$	T^{CPLEX}	$\overline{Z}^{FRC}$	T^{FRC}	GG (%)	GC (%)	GF (%)	TG	t_f^*	nc_f
C9T4	19159,02	19644,09	172764	19712,09	8963	0,35	2,53	2,89	19,28	2	4
C9T3	19322	19817,52	172506	20439,74	3175	3,14	2,56	5,78	54,33	4	60
C9T2	19082,51	19580,18	172754	19724,45	8005	0,74	2,61	3,36	21,58	2	8
C8T4	18077,72	18569,88	172390	19237,75	4760	3,60	2,72	6,42	36,22	4	53
C8T3	18988,45	19714,56	172754	19531,52	7037	-0,93	3,82	2,86	24,55	3	38
C8T2	17459,53	17936,94	172364	18441,02	1593	2,81	2,73	5,62	108,20	4	45
C7T4	18998,82	19368,61	130029	19693,88	1809	1,68	1,95	3,66	71,88	4	39
C7T3	18779,62	19058,21	24944	19499,45	1436	2,32	1,48	3,83	17,37	4	38
C7T2	18696,9	19064,1	150318	19507,27	2015	2,32	1,96	4,33	74,60	3	15
C6T4	12931	13096,69	3876	13282,77	678	1,42	1,28	2,72	5,72	3	25
C6T3	13076,12	13260,04	13854	13390,33	470	0,98	1,41	2,40	29,48	4	78
C6T2	12897,23	13024,14	2046	13072,13	448	0,37	0,98	1,36	4,57	2	7
C5T4	12655,5	12845,5	1567	13096,63	288	1,96	1,50	3,49	5,44	3	13
C5T3	12535,82	12676,38	2327	12705,2	380	0,23	1,12	1,35	6,12	3	20
C5T2	13138,6	13272,14	2905	13572,03	1150	2,26	1,02	3,30	2,53	4	74
C4T4	12829,75	12983,22	6242	13277,91	339	2,27	1,20	3,49	18,41	3	38
C4T3	12906,62	13097,14	3404	13405,85	366	2,36	1,48	3,87	9,30	4	62
C4T2	12926,51	13079,39	3745	13139,92	379	0,46	1,18	1,65	9,88	3	28

Elapsed time limit: 172800 secs (48 hours).

108.2 times faster (instance C8-T2).

The cases C7 to C9, with 3 rows each correspond to large-scale instances, with 15 facilities and 75 customers, whose uncertainty in the main parameters is represented by up to 268 scenarios, for which the associated DEM has up to 483000 (binary) variables and 554000 constraints. FRC was able to find good quality solutions for all the instances in less than 3 hours. On the other hand, CPLEX reaches the time limit of 48 hours in 7 out of these 9 instances. Note that the worst time for FRC is for instance C9-T4, that took 8963 seconds to find the incumbent solution, with a goodness gap GG=0.35% and an elapsed time improvement ($TG = 19.28$). In the remaining instances, FRC yields very small goodness gaps. The largest one was 3.6%, for instance C8-T4, which required almost 2 days (172390 seconds) to be solved by CPLEX. Notice that for instance C8-T3, it took FRC 2 hours to find a better solution than CPLEX in 2 days (as we can see for $GG < 0$). Additionally, for instances C9-T4 and C9-T2, the goodness gap GG is not higher than 1%.

Analysis of the results of Table 5.3

Table 5.3 compares the solution values and elapsed times between FRC and FRC-J. As FRC-J is an improved version for FRC algorithm, we want to compare its strengthnesses.

The headings are as follows: Z_{LP}, solution value of the LP relaxation of the DEM; $\overline{Z}^{FRC}$ and $\overline{Z}^{FRCJ}$, solution values (i.e., expected total cost) of the incumbent solutions obtained by FRC and FRC-J, respectively; T^{FRC} and T^{FRCJ}, elapsed times (secs) to obtain the associated incumbent solutions; GG, goodness gap between the solution values $\overline{Z}^{FRCJ}$ and $\overline{Z}^{FRC}$, defined as $\frac{\overline{Z}^{FRCJ}-\overline{Z}^{FRC}}{\overline{Z}^{FRC}}$ (in $\%$); TG, time gap between T^{FRC} and T^{FRCJ}, defined as $\frac{T^{FRC}}{T^{FRCJ}}$; GF and GJ, quasi-optimality gaps for the solution values obtained by FRC and FRC-J, defined as $\frac{\overline{Z}^{FRC}-Z_{LP}}{Z_{LP}}$ (in $\%$) and $\frac{\overline{Z}^{FRCJ}-Z_{LP}}{Z_{LP}}$ (in $\%$), respectively; t_f^* and t_j^* break stages used to decompose the original DEM into clusters for that particular instance to be solved by the FRC and FRC-J algorithms, respectively; nc_f and nc_j, number of clusters in which the original DEM has been decomposed for that particular instance to be solved by the FRC and FRC-J algorithms, respectively.

Notice again that $GG = 0$ would mean that both approaches being compared in the corresponding table, have obtained the same solution value. The quality of the solutions found and the elapsed times, let us insist, are concerned with the break stages, t_f^* and t_j^*, selected to solve each instance. And again, we have tried different break stages for each instance, in order to get what we consider the better trade-off between the quality of the solution and the computing time for comparing both algorithms. While FRC tends clearly to improve its elapsed times for higher break stages and on the other hand FRC-J improvement has a peak and then declines, for some instances. That is why for such instances a different break stage has been selected for both algorithms.

FRC-J finds almost always the same solutions than FRC. Notice that we have selected $t_f^* = t_j^*$ for 14 out of the 18 instances in the testbed, and for all but two of those 14 instances, both algorithms found the same solution ($GG = 0$). For instance C5-T3, instead of selecting the same break stage, FRC finds a slightly better solution ($GG = 0.01$) and the elapsed times are the closest among all the instances, being FRC-J $TG = 1.22$ times faster, which does not seem a significant improvement. On the contrary, for instance C5-T4, FRC-J solution is better ($GG = -0.46$), and the elapsed time is quite better too ($TG = 1.71$).

For instances C9-T3 and C8-T4, the elapsed time needed by FRC-J to find the solution was significantly better for $t_j^* = 3$ than for $t_j^* = 4$, this is why we have selected $t_j^* = 3$ for both instances. Thus, it is no surprise that, in these cases, $GG < 0$. Anyway, it is of interest to note that FRC-J

Table 5.3: Computational results. FRC vs FRC-J

Instance	Z_{LP}	$\overline{Z}^{FRC}$	T^{FRC}	$\overline{Z}^{FRCJ}$	T^{FRCJ}	GG (%)	GF (%)	GJ (%)	TG	t_f^*	nc_f	t_j^*	nc_j
C9T4	19159,02	20054,12	3153	20054,12	804	0,00	4,67	4,67	3,92	3	13	3	13
C9T3	19322	20439,74	3175	20377,29	1294	-0,31	5,78	5,46	2,45	4	60	3	20
C9T2	19082,51	20082,76	2445	20082,07	1320	0,00	5,24	5,24	1,85	3	22	3	22
C8T4	18077,72	19237,75	4760	19134	1160	-0,54	6,42	5,84	4,10	4	53	3	19
C8T3	18988,45	19531,52	7037	19531,53	3147	0,00	2,86	2,86	2,24	3	38	3	38
C8T2	17459,53	18441,02	1593	18441,02	905	0,00	5,62	5,62	1,76	4	45	4	45
C7T4	18998,82	19693,88	1809	19693,88	480	0,00	3,66	3,66	3,77	4	39	4	39
C7T3	18779,62	19499,45	1436	19499,45	747	0,00	3,83	3,83	1,92	4	38	4	38
C7T2	18696,9	19507,27	2015	19507,27	535	0,00	4,33	4,33	3,77	3	15	3	15
C6T4	12931	13282,77	678	13282,76	231	0,00	2,72	2,72	2,94	3	25	3	25
C6T3	13076,12	13390,33	470	13390,09	235	0,00	2,40	2,40	2,00	4	78	4	78
C6T2	12897,23	13072,13	448	13071,88	289	0,00	1,36	1,35	1,55	2	7	2	7
C5T4	12655,5	13096,63	288	13036,03	168	-0,46	3,49	3,01	1,71	3	13	3	13
C5T3	12535,82	12705,2	380	12706,89	312	0,01	1,35	1,36	1,22	3	20	3	20
C5T2	13138,6	13572,03	1150	13572,03	272	0,00	3,30	3,30	4,23	4	74	4	74
C4T4	12829,75	13277,91	339	13277,91	207	0,00	3,49	3,49	1,64	3	38	3	38
C4T3	12906,62	13405,85	366	13405,85	207	0,00	3,87	3,87	1,77	4	62	4	62
C4T2	12926,51	13230,94	301	13230,94	170	0,00	2,36	2,36	1,77	4	71	4	71

Elapsed time limit: 172800 secs (48 hours).

is capable of getting better solutions in significantly smaller elapsed times (e.g. $TG = 4.1$ and $GG = -0.54\%$, for C8-T4), particularly for large sized problems. Concluding, notice that, for all the instances in the testbed, FRC-J is faster than FRC (up to $TG = 4.1$), being the elapsed time less than a half ($TG \geq 2\%$) for 9 out of 18 instances, and what is more important, there are 6 out of the 9 large-scale instances (C7 to C9), where $TG \geq 2$, even improving the solution for C8-T4, where $GG = -0.54\%$ and $TG = 4.1\%$. For all the preceding, it seems that the FRC-J is the most reasonable option faced to FRC.

Analysis of the results of Table 5.4

Table 5.4 compares the solution values and elapsed times between FRC-J and FRC-PJ. As we will see, FRC-PJ gets outstanding better elapsed times.

The headings are as follows: Z_{LP}, solution value of the LP relaxation of the DEM; $\overline{Z}^{FRCJ}$ and $\overline{Z}^{FRCPJ}$, solution values (i.e., expected total cost) of the incumbent solutions obtained by FRC-J and FRC-PJ, respectively; T^{FRCJ} and T^{FRCPJ}, elapsed times (secs) to obtain the associated incumbent solutions; GG, goodness gap between the solution values $\overline{Z}^{FRCPJ}$ and $\overline{Z}^{FRCJ}$, defined

as $\frac{\overline{Z}^{FRCPJ}-\overline{Z}^{FRCJ}}{\overline{Z}^{FRCJ}}$ (in $\%$); TG, time gap between T^{FRCJ} and T^{FRCPJ}, defined as $\frac{T^{FRCJ}}{T^{FRCPJ}}$; GJ and GP, quasi-optimality gaps for the solution values obtained by FRC-J and FRC-PJ, defined as $\frac{\overline{Z}^{FRCJ}-Z_{LP}}{Z_{LP}}$ (in $\%$) and $\frac{\overline{Z}^{FRCPJ}-Z_{LP}}{Z_{LP}}$ (in $\%$), respectively; t_j^* and t_p^*, break stages used to decompose the original DEM into clusters for that particular instance to be solved by the FRC-J and FRC-PJ algorithms, respectively; nc_j and nc_p, number of clusters in which the original DEM has been decomposed for that particular instance to be solved by the FRC-J and FRC-PJ algorithms, respectively; E, efficiency of the parallelization, defined as $\frac{TG}{nc_p}$ (remind from Chapter 3.11.1 that $e(p,n) = \frac{s(p,n)}{p}$).

For this table we let the same break stages t_j^* than in Table 5.3, but for solving the instances with FRC-PJ we have chosen $t_p^* = 4$ for almost all the instances, except for C8-T3 and C4-T4, since the improvement in elapsed time was not significant.

The goodness gap is quite small, no more than the $GG = 1.75\%$ for instance C5-T3 where, however, the elapsed time improvement is quite high ($TG = 19.5$), which is the highest of all the instances. In it is the improvement in the elapsed times what stands out clearly from this table, being $TG \geq 10$ for 14 out of the 18 instances, and $TG = 7.28$ the least improvement accomplished by FRC-PJ (instance C8-T3). Let us point out that both algorithms always find the same solutions, so the gaps $GG > 0\%$ that can be seen in 9 out of the 18 instances are due to the fact that $t_j^* < t_p^*$, but if we had chosen a higher break stage for FRC-J the elapsed time (T^{FRCJ}) would have been worse. Thus, it seems that when the computational resources needed are available, the FRC-PJ is clearly the best option. FRC-PJ can solve large instances in a few seconds when CPLEX needs hours, even days. We can point out cases as C8-T3, where FRC-PJ finds a better solution than CPLEX in 432 seconds vs 2 days.

Notice that efficiency is below 50% ($E < 0.5$) for all the instances, although there are 5 of them that get more than 33% ($E \geq 0.33$), i.e. C9-T4, C7-T4, C7T3, C6-T2 and C5-T3. And the least efficient instance is C6-T3, with $E = 0.18$. These are not big values but it has to be taken into account that many processors have been used, what implies more inter-processor communications. Another important factor that affects parallelization efficiency is the mentioned difference among the subproblems resolution times. Great differences have been observed among those times for some instances. Nevertheless, there is room for further research in order to improve the efficiency of the proposed algorithm.

Table 5.4: Computational results. FRC-J vs FRC-PJ

Instance	Z_{LP}	$\overline{Z}^{FRCJ}$	T^{FRCJ}	$\overline{Z}^{FRCPJ}$	T^{FRCPJ}	GG (%)	GJ (%)	GP (%)	TG	t_j^*	nc_j	t_p^*	nc_p	E
C9T4	19159,02	20054,32	804	20259,04	61	1,02	4,67	5,74	13,18	3	13	4	39	0,34
C9T3	19322	20377,29	1294	20439,74	77	0,31	5,46	5,78	16,81	3	20	4	60	0,28
C9T2	19082,51	20082,07	1320	20166,96	74	0,42	5,24	5,68	17,84	3	22	4	61	0,29
C8T4	18077,72	19134	1160	19237,75	76	0,54	5,84	6,42	15,26	3	19	4	53	0,29
C8T3	18988,45	19531,53	3147	19531,53	432	0,00	2,86	2,86	7,28	3	38	3	38	0,19
C8T2	17459,53	18441,02	905	18441,02	67	0,00	5,62	5,62	13,51	4	45	4	45	0,30
C7T4	18998,82	19693,88	480	19693,88	35	0,00	3,66	3,66	13,71	4	39	4	39	0,35
C7T3	18779,62	19499,45	747	19499,45	51	0,00	3,83	3,83	14,65	4	38	4	38	0,39
C7T2	18696,9	19507,27	535	19624,06	56	0,60	4,33	4,96	9,55	3	15	4	42	0,23
C6T4	12931	13282,76	231	13312,83	13	0,23	2,72	2,95	17,77	3	25	4	63	0,28
C6T3	13076,12	13390,09	235	13390,09	16	0,00	2,40	2,40	14,69	4	78	4	78	0,19
C6T2	12897,23	13071,88	289	13228,61	15	1,20	1,35	2,57	19,27	2	7	4	49	0,39
C5T4	12655,5	13036,03	168	13071,72	19	0,27	3,01	3,29	8,84	3	13	4	35	0,25
C5T3	12535,82	12706,89	312	12929,08	16	1,75	1,36	3,14	19,50	3	20	4	55	0,35
C5T2	13138,6	13572,03	272	13572,03	18	0,00	3,30	3,30	15,11	4	74	4	74	0,20
C4T4	12829,75	13277,91	207	13277,91	23	0,00	3,49	3,49	9,00	3	38	3	38	0,24
C4T3	12906,62	13405,85	207	13405,85	13	0,00	3,87	3,87	15,92	4	62	4	62	0,26
C4T2	12926,51	13230,94	170	13230,94	11	0,00	2,36	2,36	15,45	4	71	4	71	0,22

Elapsed time limit: 172800 secs (48 hours).

Analysis of the results of Table 5.5

Finally, Table 5.5 shows the computational results reported in Albareda-Sambola et al. ([11]), which are compared to the results obtained by the FRC-PJ algorithm.

The headings are as follows: Z_{LP}, solution value of the LP relaxation of the DEM; $\overline{Z}^{FRC-S}$ and $\overline{Z}^{FRCPJ}$, solution values (i.e., expected total cost) of the incumbent solutions obtained by implementation of the FRC used in that paper and FRC-PJ, respectively; T^{FRC-S} and T^{FRCPJ}, elapsed times (secs) to obtain the associated incumbent solutions; GG, goodness gap between the solution values $\overline{Z}^{FRCPJ}$ and $\overline{Z}^{FRC-S}$, defined as $\frac{\overline{Z}^{FRCPJ}-\overline{Z}^{FRC-S}}{\overline{Z}^{FRC-S}}$ (in $\%$); GJ and GP, quasi-optimality gaps for the solution values obtained by FRC and FRC-PJ, defined as $\frac{\overline{Z}^{FRC-S}-Z_{LP}}{Z_{LP}}$ (in $\%$) and $\frac{\overline{Z}^{FRCPJ}-Z_{LP}}{Z_{LP}}$ (in $\%$), respectively;t_p^* break stage used to decompose the original DEM into clusters for that particular instance to be solved by FRC-PJ.

The aim is to compare the goodness of the solutions found, it makes no sense comparing the elapsed times since the algorithms were run in different machines. Anyway, for this table we have chosen smaller break stages to get better solutions, and the elapsed times are included to show that even for these cases good times were achieved.

Table 5.5: Computational results. FRC-SLOC vs FRC-PJ

Instance	Z_{LP}	$\overline{Z}^{FRC-S}$	T^{FRC-S}	$\overline{Z}^{FRCPJ}$	T^{FRCPJ}	GG (%)	GFS (%)	GP (%)	t_p^*
C9T4	19159,02	19957,05	6656	19712,09	872	-1,23	4,17	2,89	2
C9T3	19322	20277,86	9524	20272,28	912	-0,03	4,95	4,92	2
C9T2	19082,51	19856,18	8821	19724,45	1562	-0,66	4,05	3,36	2
C8T4	18077,72	19039,68	7415	19035,88	1007	-0,02	5,32	5,30	2
C8T3	18988,45	19991,31	15575	19531,53	432	-2,30	5,28	2,86	3
C8T2	17459,53	18367,62	7678	18607,08	126	1,30	5,20	6,57	3
C7T4	18998,82	19627,87	3274	19601,23	113	-0,14	3,31	3,17	3
C7T3	18779,62	19307,6	4040	19534,06	108	1,17	2,81	4,02	3
C7T2	18696,9	19548,97	6914	19277,25	959	-1,39	4,56	3,10	2
C6T4	12931	13323,31	807	13312,83	13	-0,08	3,03	2,95	4
C6T3	13076,12	13364,56	1084	13390,09	16	0,19	2,21	2,40	4
C6T2	12897,23	13176,96	715	13071,88	97	-0,80	2,17	1,35	2
C5T4	12655,5	13016,12	381	13036,03	29	0,15	2,85	3,01	3
C5T3	12535,82	12880,07	642	12706,89	49	-1,34	2,75	1,36	3
C5T2	13138,6	13272,15	659	13287,56	74	0,12	1,02	1,13	3
C4T4	12829,75	13007,08	1606	13277,91	23	2,08	1,38	3,49	3
C4T3	12906,62	13101,12	976	13333,35	30	1,77	1,51	3,31	3
C4T2	12926,51	13194,78	660	13139,92	30	-0,42	2,08	1,65	3

As we can see, the solutions found are very similar, being better for FRC-PJ in 11 out of 18 instances ($GG < 0$), with a gap of down to -2.3% for instance C8-T3 and up to 2.08% for instance C4-T4. Notice that even for these good solutions the elapsed times are small, from 16 seconds for the medium sized instance C6-T4 up to the 25 minutes of the large C9-T2. And notice too the gap deviation against FRC-PJ for 7 out of 9 of the largest instances.

Chapter 6

Conclusions, contributions and future research

In this final chapter the conclusions from the research developed in this thesis are explained. The original contributions achieved while pursuing the goals of this thesis are collected. Finally, some future research lines are identified.

Some of the main results of this thesis have been published in [23, 24] and presented in several national and international meetings:

- XXXIII National Congress of Statistics and Operations Research. *A parallel computing meta-heuristic for solving multistage stochastic mixed integer programs.* Madrid (Spain), 2012.
- 12th International Conference on Stochastic Programming. *FRC: A heuristic extension of the BFC approach for solving very large scale multistage mixed 0-1 stochastic programs.* Halifax (Canada), 2010.
- ECCO XXIII-CO2010 European Chapter on Combinatorial Optimization. *On The Conflict Avoidance for Air Traffic Flow Management Problem, two models.* Málaga (Spain), 2010.
- 24th European Conference on Operations Research. *FRC: A heuristic extension of the Branch-and-Fix Coordination approach for solving very large scale multistage mixed 0-1 stochastic problems.* Lisbon (Portugal), 2010.
- 24th European Conference on Operations Research. *Conflict Avoidance for Air Traffic Flow Management Problem, pure and MIP models.* Lisbon (Portugal), 2010.

- 4th workshop on Optimization and Variational Analysis. *On solving large-scale stochastic mixed 0-1 linear problems. Seminario invitado.* Elche (Spain), 2010.
- 23rd European Conference on Operational Research. *On the collision avoidance for air traffic management problem, a large scale mixed 0-1 program approach.* Invited Session. Bonn (Germany), 2009.
- 23rd European Conference on Operational Research. *On the air traffic flow management problem. A stochastic integer programming approach. Invited Session.* Bonn (Germany), 2009.
- Workshop CORAL 2009 (Conference on Routing and Logistics). *Air traffic flow management. Airports and sectors constraints, with en route problem.* Elche (Spain), 2009.

6.1 Conflict Avoidance in the Air Traffic Flow problem CA

With the objective in mind of tackling the CA problem to develop a tool that could help in eluding conflicts both flexible and efficiently, a broad research has been conducted through the vast literature on the subject.

A new point of view has been adopted, based on given aircraft trajectories, that allows developing a linear model capable of detecting conflicts and propose maneuvers for avoiding them. The aim of this new approach is inspired in decomposing the problem. Since it is already at our hand to obtain the flight trajectories in a realistic and efficient way, a good solution could be a model that, taking into account the flight routes, be able of detecting and avoiding all conflicts. Ideally, such trajectories would come with some parameters adding some room for correctly maneuvering, i.e. for reducing or augmenting the speed and for climbing or descending a flight level (this is left as further research). Nevertheless, even with the actual flight plans, and considering a few maneuvers (e.g. climb-descend just one flight level), it is possible to avoid conflicts in a coherent and, in our opinion, as realistic fashion as the approaches currently presented in the literature on the subject, in addition to the benefit of allowing a wide range of action.

Two novel models have been proposed: a pure 0-1 and a mixed 0-1 linear, the first of which avoids conflicts by means of altitude changes, while the second's strategy is based on altitude and speed changes.

Based on the computational experience reported in Section 2.4, it can be said that both models are tight and can be solved in really small elapsed times, being the first, pure 0-1 model, the most

efficient. Although this model only considers flight level changing maneuvers, opposed to the two maneuvers allowed by the second model (flight level and speed changes), it has to be pointed out that speed changes are not considered a very efficient maneuver in the literature (see Frazzoli et al. [108], Jardin [138] and Peyronne et al. [190], among others). Moreover, in many works aircraft are assumed to fly at constant speed (see Pallottino et al. [185], Christodoulou and Costoulakis [62], Treleaven [227], Gao et al. [111] and Cafieri and Durand [50], among others). In addition, in many real cases just a few flight level changes will solve the problem easily.

The approach is also flexible, it allows non-linear trajectories. These given trajectories can be based on procedural information such as a flight plan, as well as the optimal trajectory decided in the future "*Free Flight*" (see [222]), or a straight-line extrapolation of the current speed vector as in [185, 16, 17].

To summarize, both models can be solved in really small elapsed times, even for large-scale instances, so, the approach can be used in real time with the help of a state-of-the-art mixed integer linear optimization software, and the approach can be used in long term time horizons as well as for wide airspace regions that may comprise several air traffic control areas in which the aerial space is divided. Finally, the presented models can be easily extended with new features or restrictions, in particular a third conflict avoidance maneuver has been proposed (selecting alternative routes), although its implementation has been left for further research.

6.2 Stochastic Programming

Two new algorithms have been proposed, one being sequential and the other, based on the first, parallel. They use some heuristics that have proven its effectiveness previously in addition to new heuristics proposed in this thesis. Although both algorithms do not guarantee the solution to be optimum, it can be seen in the computational experience presented in Chapter 5 the quality of the solutions, which improve even CPLEX in some instances.

Both algorithms have been tested by solving a series of large-scale instances of the Multi-period location-allocation problem under uncertainty, a strategic and tactical decision problem which objective is deciding the optimum allocation of locations to fulfil certain coverage levels of demand in some places at each time period. The instances have up to half a million variables and are hard to solve. However, both algorithms manage to solve them in considerably small computing times, specially when compared with the elapsed times required by CPLEX and the FRC algorithm.

The elapsed times required by the FRC-PJ algorithm are particularly good, proving that parallel computing and decomposition algorithms work really well together.

Finally, it has to be taken into account that these kind of algorithms not only allow performance improvements in computing time, but also in memory consumption, since decomposing the main problem in several subproblems reduces the system memory requirements.

6.3 Future work

This thesis yields a series of future lines of research that hopefully will produce interesting results. Some of them are examined below.

Extending the CA models

The models for CA presented in Chapter 2 can be extended and its performance improved. In particular, choosing alternative routes as avoidance maneuver; allowing aircraft climbing or descending to the next altitude level in more than one step; as well as relating altitude level changes to speed.

Computing the aircraft trajectories

An intermediate level to feed the proposed models with the parameters they need from the actual data used by the aerial companies and traffic control centers, yielding the trajectories as well as the constraining parameters for changing speed and altitude levels.

Lagrangean Decomposition

The Lagrangian Decomposition procedures can help to improve the algorithms presented in this thesis. In particular, the specialization of the so-called Cluster Lagrangean Decomposition presented in [91]. In that work it is applied to a facility location problem under uncertainty. The FRC-J and the FRC-PJ algorithms can be improved by using this technique that will allow obtaining strong (lower) bounds at each iteration and so improving the pruning mechanism.

Risk Measures

The models considered so far present as the objective function the expected value of the objective function over the set of finite scenarios, i.e. the so called risk neutral (RN) approach. However, RN solutions have the inconvenience of ignoring the variability of the objective function value over the scenarios. So, it does not hedge against the low-probability/high-consequence events (the so-called black swans). Alternatively, risk measures can be added in order to hedge against the impact of the most unwanted scenarios. Risk measures are currently been taken into account by considering, for example, semi-deviations, excess of probability, conditional value-at-risk and others. These approaches are more convenient under the presence of binary variables than the classical media-variance schemas. However, due to the large number of additional 0-1 and continuous variables and constraints required by these strategies, problems become larger and harder to solve. In particular, the Time Stochastic Dominance (TSD) strategy reduces the risk of wrong solutions in a better way than others under some circumstances, according to the computational comparison reported in e.g., [13]. The strategy also aims to minimize the objective function expected value, see also [90]. It would be of interest solving such problems with the help of the algorithms presented in this thesis.

Computational experience

The algorithms presented in this thesis, as well as the improved versions that will be developed, should be tested by applying them for the resolution of very large-scale problems.

Improving the the parallel approach

As it has been mentioned in Section 5.3.2, the efficiency of the parallel algorithm FRC-PJ leaves room for improvement. To do so, observe that in many instances it has been noticed that some subproblems need considerably higher elapsed times to be solved than the others. As mentioned in Chapter 4, particularly in Section 4.4, it can be "avoided" to solve such subproblems very frequently. Such heuristic would consist in, when branching to a new variable, fixing the value that took that variable in the solution of the hardest problem, so it would not be required solving it again. This approach can be studied in a future work.

Besides, other parallel programming approaches can be tried, such as dividing each subproblem in more subproblems by fixing some variables to different values, and assigning each subproblem to a different process. This will allow more "parallelization" and, hopefully, a greater improvement in computing time.

Library

It would be of great interest developing a library for solving general large-scale multistage stochastic mixed 0-1 problems, by using either of the algorithms that have been implemented for this thesis, namely FRC, FRC-J and FRC-PJ. Some different strategies could also be parameterizable, such as the strategies for selecting the branching variable, the value it has to be fixed at, etc.

Glossary

MSFLP Multi-Period Stochastic Facility Location Problem.. xxiii, 117, 118, 120, 122, 123

Stochastic Programming the field of Mathematical Programming that considers and deals with uncertainty, incorporating it in the mathematical models. 45, 85, 94

***expected value of perfect information* ($EVPI$)** The *expected value of perfect information* measures the maximum amount a decision maker would be ready to pay in return for expected complete (and accurate) information about the future.. 44, 65

metaheuristic A *metaheuristic* is a *higher-level* procedure designed as an efficient approach to a hard optimization problem, aimed at finding a sufficiently good solution. It is especially appropriate for problems with incomplete or imperfect information or limited computation capacity. 43, 45

multi-stage multi-stage problems are stochastic problems in which there are more than two distinguished stages (time periods, such that at the beginning of each stage the values of some uncertain parameters are revealed) where decisions are taken. xxii, 45, 50–52, 56, 59, 85, 93

non-anticipativity In the general formulation of a multi-stage stochastic integer optimization problem, decisions are made stage-wise. At each stage, there are variables corresponding to decisions that have to be made without anticipating the values of some future problem data, i.e., they take on the same value under each scenario in a given group. 45, 55, 57, 85, 86, 89, 102–104, 106–112, 116

scenario cluster A ***scenario cluster*** is a set of scenarios where the ***non-anticipativity*** constraints are implicitly defined in the model.. 107

scenario group A **scenario group** for a given stage is the set of scenarios with the same realization of the uncertain parameters up to the stage.. 54, 107

scenario A **scenario** is a particular realization of the uncertain parameters along the different stages of the time horizon.. 54

two-stage problems where decisions are taken in one of two distinguished stages or time periods (named first and second stage, where first stage is the time period before the uncertain parameters values are known, and the second stage is the time period after such values are revealed). 45, 48, 50, 51, 53, 92, 93, 103

BFC Branch-and-Fix Coordination. 44, 45, 85, 110

BFC-J Jumping Branch-and-Fix Coordination. 110, 112

CA Conflict Avoidance. xxi, xxii, 1, 3, 11, 15, 134, 136

CDR Conflict Detection and Resolution. 1, 2

DEM Deterministic Equivalent Problem. 44, 85, 99, 107, 108, 117, 120, 125–131

FRC Fix-and-Relax Coordination. xiv, xv, xvii, xviii, 43, 99, 100, 103, 104, 107, 108, 110, 111, 126–129, 131, 135, 138

FRC-J Jumping Fix-and-Relax Coordination. xv, xviii, xxiii, 43, 99, 110–114, 126, 128–131, 136, 138

FRC-PJ Parallel Jumping Fix-and-Relax Coordination. xv, xvii, xviii, xxiii, 43, 113–115, 123, 126, 129–132, 136–138

IP Integer Programming. 48

LP Linear Programming. xxi, 47, 48, 51

Master Master Process, responsible for coordinating all the concurrent processes during the parallel algorithm.. 113, 115, 116

MIP Mixed Integer Programming. 48, 100, 103, 111–113

MPI Message Passing Interface. xv, xxiii, 96, 97, 115, 123

SP Stochastic Programming. xxi–xxiii, 43, 44, 46, 90, 93, 108, 110, 118

speedup The speedup and efficiency of an algorithm are widely used to analyze the *goodness* of a parallel algorithm. The speedup of a parallel algorithm p for a problem P of size n is defined as $s(p,n) = \frac{t_s(n)}{t_p(p,n)}$, where $t_s(n)$ is the time (alternatively, the number of operations) needed to solve P by the best known sequential algorithm, and $t_p(p,n)$ the time needed by algorithm p to solve P. 98, 114, 115

stage A **stage** of a given time horizon, is a set of consecutive time periods in which the realization of some uncertain parameters takes place.. 48

TCAS Traffic Alert and Collision Avoidance System. 2

TNF Twin Node Family. 86, 101, 103, 104, 110–112, 116

Bibliography

[1] S. Ahmed. Mean-risk objectives in stochastic programming. Stochastic Programing E-print series, http://hera.rz.hu-berlin.de/speps/, 2004.

[2] S. Ahmed, A.J. King, and G. Parija. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Journal of Global Optimization*, pages 3–24, 2003.

[3] S. Ahmed, M. Tawarmalani, and N.V. Sahinidis. A finite branch and bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, pages 355–377, 2004.

[4] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows:Theroy, Algorithms and Applications*. Prentice-Hall, Englewood Cliffs,NJ, 1993.

[5] Air Traffic Action Group (ATAG), Geneva, Switzerland. *European Air Traffic Forecast 1985-2015*, 1999.

[6] T. Al-Khamis and R. M'Hallah. A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity. *Computers & Operations Research*, 38:1747–1759, 2011.

[7] S. Alam, M. McPartland, M. Barlow, P. Lindsay, and H.A. Abbass. Neural evolution for collision detection & resolution in a 2d free flight environment. Technical Report TR-ALAR-200507012, The Artificial Life and Adaptative Robotics Laboratory, University of New South Wales, 2005.

[8] S. Alam, K. Shafi, H.A. Abbass, and M. Barlow. Forming an intelligent ensemble of conflict detection algorithms in free flight by data mining the scenario space. Technical Report TR-ALAR-200802003, The Artificial Life and Adaptative Robotics Laboratory, University of New South Wales.

[9] Albareda-Sambola, E. FernÃ¡ndez, Y. Hinojosa, and J. Puerto. The multi-period sequential coverage facility location problem. *Computers & Operations Research*, 36:1356–1375, 2009.

[10] M. Albareda-Sambola, A. Alonso-Ayuso, L.F. Escudero, E. Fernández, Y. Hinojosa, and C. Pizarro. A computational comparison of several formulations for the multi-period location-assignment problem. *TOP*, 18:62–80, 2010.

[11] M. Albareda-Sambola, A. Alonso-Ayuso, L.F. Escudero, E. Fernández, and C. Pizarro. Fix-and-relax-coordination for a multi-period location-allocation problem under uncertainty. *Computers & Operations Research*, 40:2878–2892, 2013.

[12] M. Albareda-Sambola, E. Fernández, Y. Hinojosa, and J. Puerto. The multi-period sequential coverage facility location problem. *Computers & Operations Research*, 36:1356–1375, 2009.

[13] A. Alonso-Ayuso, F. Carvallo, L.F. Escudero, M. Guignard-Spielberg, J. Pi, R. Puranmalka, and A. Weinteraub. Medium range optimization of copper extraction planning under uncertainty in future copper prices. *European Journal of Operational Research*, 2013.

[14] A. Alonso-Ayuso, L.F. Escudero, A. Garín, M.T. Ortuño, and G. Pérez. An approach for strategic supply chain planning under uncertainty based on stochastic 0–1 programming. *Journal of Global Optimization*, 26:97–124, 2003.

[15] A. Alonso-Ayuso, L.F. Escudero, A. Garín, M.T. Ortuño, and G. Pérez. On the product selection and plant dimensioning under uncertainty. *Journal of Global Optimization*, 33:307–318, 2005.

[16] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. Collision avoidance in the air traffic management: a mixed integer linear optimization approach. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):47–57, 2011.

[17] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. A mixed 0–1 nonlinear approach for the collision avoidance in atm: velocity changes through a time horizon. *Computers & Operations Research*, 93(12):3136–3146, 2012.

[18] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. On modeling the air traffic control coordination in the collision avoidance problem by mixed integer linear optimization. *Ann. Op. Res.*, 2013.

[19] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. Exact and approximate solving of the aircraft collision resolution problem via turn changes. *Transportation Science. Published online*, 2014.

[20] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution. *TOP*, pages 1–28, 2015.

[21] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. Multiobjective optimization for aircraft conflict resolution. a metaheuristic approach. *European Journal of Operational Research*, 248:691–702, 2016.

[22] A. Alonso-Ayuso, L.F. Escudero, F.J. Martín-Campo, and N. Mladenovic. A vns metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes. *Journal of Global Optimization*, 63:583–596, 2015.

[23] A. Alonso-Ayuso, L.F. Escudero, P. Olaso, and C. Pizarro. Frc: A heuristic extension of the branch-and-fix coordination approach for solving very large scale multi-stage mixed 0-1 stochastic problems. *Technical Reports on Statistics and Decision Sciences. URJC*, 2009.

[24] A. Alonso-Ayuso, L.F. Escudero, P. Olaso, and C. Pizarro. Conflict avoidance: 0-1 linear models for conflict detection & resolution. *TOP*, 21:485–504, 2013.

[25] A. Alonso-Ayuso, L.F. Escudero, and M.T. Ortuño. A stochastic 0-1 program based approach for the air traffic flow management problem. *European Journal of Operational Research*, 120:47–62, 2000.

[26] A. Alonso-Ayuso, L.F. Escudero, and M.T. Ortuño. BFC, a Branch-and-Fix Coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs. *European Journal of Operational Research*, 151:503–519, 2003.

[27] A. Alonso-Ayuso, L.F. Escudero, and M.T. Ortuño. Modeling production planning and scheduling under uncertainty. In S.W. Wallace and W.T. Ziemba, editors, *Applications of Stochastic Programming*, pages 217–252. MPS-SIAM-Series in Optimization, 2005.

[28] A. Alonso-Ayuso, L.F. Escudero, M.T. Ortuño, and C. Pizarro. On a stochastic sequencing and scheduling problem. *Computers and Operations Research*, 34:2604–2624, 2007.

[29] A. Alonso-Ayuso, L.F. Escudero, and C. Pizarro. *Introduction to Stochastic Programming.* Ed. Dikinson, 2009.

[30] J.A. Audestad, A. Gaivoronski, and A. Werner. Modeling market uncertainty and competition in telecommunication environment: Network providers and virtual operators. *Telektronik*, 97:46–64, 2002.

[31] M. Avriel and A.C. Williams. The value of information and stochastic programming. *Operations Research*, 18:947–954, 1970.

[32] S. Baker, D. Morton, R. Rosenthal, and L. Williams. Optimizing military airlift. *Operations Research*, 50:582–602, 2002.

[33] A. M. Bayen, P. Grieder, G. Meyer, and C. J. Tomlin. Lagrangian delay predictive model for sector-based air traffic flow. *Journal of Guidance, Control and Dynamics*, 28(5):1015–1026, September-October 2005.

[34] E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society Series B*, 17:173–184, 1955.

[35] J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[36] P. Beraldi, L. Grandinetti, R. Musmanno, and C. Trik. Parallel algorithms to solve two-stage stochastic linear programs with robustness constraints. *Parallel Computing*, 26:1889–1908, 2000.

[37] D. Bertsimas. A mathematical programming approach to stochastic and dynamic optimization problems.technical report. Technical report, Operations Research Center, MIT, Cambridge, MA, USA, 1994.

[38] D. Bertsimas and S. Stock-Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46:406–422, 1998.

[39] B.Fritzke. http://www.ki.inf.tu-dresden.de/ fritzke/javapaper/. 1997.

[40] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–232, 2000.

[41] John R. Birge. Stochastic programming computation and applications. *INFORMS J. Comput.*, 9(2):111–133, 1997.

[42] J.R. Birge. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming*, 24:314–325, 1982.

[43] J.R. Birge, C.J. Donohue, D.F. Holmes, and O.G. Svintsitski. A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs. *Mathematical Programming*, 75:327–352, 1996.

[44] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming*. Springer, 2nd edition, 2011.

[45] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming. 2nd edition.* Springer, 2011.

[46] S. Bjornestad, A. Hallfjord, and K.O. Joernsten. Discrete optimization under uncertainty, the scenario and policy aggregation technique. *European Journal of Operational Research*, 1992.

[47] J. Blomval. A multistage stochastic programming algorithm suitable for parallel computing. *Parallel Computing*, 29:431–445, 2003.

[48] Jörgen Blomvall and Per Olov Lindberg. A Riccati-based primal interior point solver for multistage stochastic programming—extensions. *Optim. Methods Softw.*, 17(3):383–407, 2002. Stochastic programming.

[49] S. Cafieri, P. Brisset, and N. Durand. A mixed-integer optimization model for air traffic deconfliction. *Proceedings of TOGO (Toulouse Global Optimization). Toulouse, France*, pages 27–30, 2010.

[50] S. Cafieri and N. Durand. Aircraft deconfliction with speed regulation: New models from mixed-integer optimization. *J. Global Optim.*, 58(4):613–629, 2014.

[51] D. Cariño, T. Kent, D. Meyers, C. Stacy, M. Stylvanus, A. Turner, K. Watanabe, and W. Ziemba. The Russel-Yasuda Kasai Model: An asset/liability model for a japanese insurance company using multistage stochastic programming. *Interfaces*, 24:29–49, 1994.

[52] D. Cariño and W. Ziemba. Formulation of the Russell-Yasuda Kasai financial planning model. *Operations Research*, 46:433–449, 1998.

[53] C.C. Carøe and R. Schultz. A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal power system. *Echtzeit-Optimierung groer Systeme, preprint*, SC 98-11, 1998.

[54] C.C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recource. *Mathematical Programming*, 83:451–464, 1998.

[55] Claus C. Carøe. *Decomposition in stochastic integer programming*. Ph.d. thesis, University of Copenhagen, Denmark, 1998.

[56] B. Carpenter and J. Kuchar. Probability-based collision alerting logic for closely-spaced parallel approach. In *35th AIAA Aerosp. Sci. Meet. Exhibit*, Reno, NV, Jan. 1997. paper AIAA-97-0222.

[57] H. Casanova, A. Legrand, and Y. Robert. *Parallel Algorithms*. Chapman & Hall/ CRC Press, 2009.

[58] C.Cetek. Realistic speed change maneuvers for air traffic conflict avoidance and their impact on aircraft economics. *International Journal of Civil Aviation*, 1(1):62–73, 2009.

[59] S. Cerisola. *Descomposicion de Benders para problemas enteros mixtos: Application to a medium term hydrothermal coordination model*. PhD thesis, Universidad Comillas de Madrid, Madrid, 2003.

[60] A. Charnes and A.A. Cooper. Chance constrained programming. *Management Science*, 6:73–79, 1959.

[61] Y. Chiang, J. T. Klosowski, C. Lee, and J. S. B. Mitchell. Geometric algorithms for conflict detection/resolution in air traffic management. In *In 36th IEEE Conference on Decision and Control.*, pages 10–12, 1997.

[62] M. A. Christodoulou and C. Costoulakis. Nonlinear mixed integer programming for aircraft collision avoidance in free flight. *IEEE Melecon 2004, Dubrovnik, (Croacia)*, 1:327–330, 2004.

[63] M. A. Christodoulou and S. G. Kodaxakis. Automatic commercial aircraft-collision avoidance in free flight: The three-dimensional problem. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):242–249, 2006.

[64] M.A. Christodoulou and C. Kontogeorgou. Automatic collision avoidance in commercial aircraft three dimensional flights, using neural networks and non-linear programming. In C.H. Skiadas, editor, *Chaotic Modeling and Simulation*, 2008.

[65] K.J. Cohen and S. Thore. Programming bank portfolios under uncertainty. *Journal of bank research*, 1:42–61, 1970.

[66] K. Cormican, D.P. Morton, and R.K. Wood. Stochastic network interdiction. *Operations Research*, 46:184–197, 1998.

[67] P. Coy. The cfo goes 3d. *Business Week*, 1996. October 28.

[68] M.P. Cristobal, L.F. Escudero, and J.F. Monge. On stochastic dynamic programming for solving large-scale production planning problems under uncertainty. *Computers & Operations Research*, 36:2418–2428, 2009.

[69] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[70] G.B. Dantzig and P.W. Glynn. Parallel processors for planning under uncertainty. *Annals of Operations Research*, 22:1–21, 1990.

[71] George B. Dantzig and Peter W. Glynn. Parallel processors for planning under uncertainty. *Ann. Oper. Res.*, 22(1-4):1–21, 1990. Supercomputers and large-scale optimization: algorithms, software, applications (Minneapolis, MN, 1988).

[72] P. Dell'Olmo and G. Lulli. A new hierarchical architecture for air traffic management: Optimization of airway capacity in a free flight scenario. *European Journal of Operational Research*, 144:179–193, 2003.

[73] M.A.H. Dempster and E. Medova. Envolving system architectures for multimedia network design. *Annals of Operations Research*, 104:163–180, 2001.

[74] M.A.H. Dempster and R.T. Thompson. Parallelization of EVPI based importance sampling procedures for solving multistage stochastic linear programs on MIMD architectures. In *Parallel Optimization Colloquium*. Laboratoire PRISM, University Versalles,Versalles (Francia), 1996.

[75] N. di Domenica. *Stochastic Programming and scenario generation: Decision Modelling simulation and information systems persepective*. PhD thesis, Department of Mathematical Sciences, Brunel University, Uxbridge, Middlesex UB8 3PH, United Kingdom, 2004.

[76] B.H. Dias, M.A. Tomin, A.L.M. Mercato, T.P. Ramos, R.B.S. Brandi, I.Ch. da Silva jr., and J.A.P. Filho. Parallel computing applied to stochastic dynamic programming for long term operation planning of hydrothermal power systems. *European Journal of Operational Research*, 229:212–222, 2013.

[77] A. Doshi. *Aircraft Position Prediction Using Neural Networks*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2005.

[78] G. Dowek and C. Mu noz. Conflict detection and resolution for 1,2,...,n aircraft. Seventh AIAA Aviation Technology, Integration and Operations conference, Belfast (Northern Ireland), 2007.

[79] V. N. Duong and E. G. Hoffman. Conflict resolution advisory service in autonomous aircraft operations. In *16th Digital Avion. Syst. Conf.*, pages 9.3–10–9.3–17, Irvine, CA, Oct. 1997.

[80] N. Durand and J. Alliot. Ant colony optimization for air traffic conflict resolution. In *8th USA/Europe air traffic management research and development seminar (ATM2009)*, 2009.

[81] N. Durand, J.M. Alliot, and F. Médioni. Neural nets trained by genetic algorithms for collision avoidance. *Applied Intelligence*, (13):205–213, 2000.

[82] G.D. Eppen, R.K. Martin, and L. Schrage. A scenario approach to capacity planning. *Operations Research*, 37:517–527, 1989.

[83] H. Erzberger, R. A. Paielli, D. R. Isaacson, and M. M. Eshow X. Conflict detection and resolution in the presence of prediction error, June 1997. prepared for the 1st USA/Europe Air Traffic Management RD Seminar, Saclay, France.

[84] L.F. Escudero. On a mixture of the Fix-and-Relax Coordination and lagrangean substitution schemes for multistage stochastic mixed integer programming. *TOP*, 17:5–29, 2009.

[85] L.F. Escudero. On branch-and-fix coordination and lagrangean substitution and decomposition for multistage stochastic mixed integer programming. *TOP*, 17:5–29, 2009.

[86] L.F. Escudero, J.L. de la Fuente, C. García, and F.J. Prieto. A parallel computation approach for solving multistage stochastic network problems. *Annals of Operations Research*, 90:131–160, 1999.

[87] L.F. Escudero, E. Galindo, C. García, E. Gómez, and V. Sabau. Schumann: A modelling framework for supply chain management under uncertainty. *European Journal of Operational Research*, 119:13–34, 1999.

[88] L.F. Escudero, A. GarÃn, , and A. Unzueta. Cluster lagrangean decomposition in multistage stochastic optimization. *In second revision*, 2015.

[89] L.F. Escudero, A. GarÃn, M. Merino, and G. PÃ©rez. On time stochastic dominance induced by mixed integer-linear recourse in multistage stochastic programs. *European Journal of Operational Research*, 2015.

[90] L.F. Escudero, A. GarÃn, C. Pizarro, and A. Unzueta. Frca, a metaheuristic algorithm for solving multistage stochastic mixed 0-1 problems with the time stochastic dominance risk averse strategy. *on revision*, 2014.

[91] L.F. Escudero, A. GarÃn, C. Pizarro, and A. Unzueta. On the strongest lagrangean bound for multistage stochastic location-assignment problems. *on revision*, 2015.

[92] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. A two-stage stochastic integer programming approach as a mixture of branch-and-fix coordination and benders decomposition schemes. *Annals of Operations Research*, pages 395–420, 2007.

[93] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. A general algorithm for solving two-stage stochastic mixed-integer programs. *Computers & Operations Research*, pages 2590–2600, 2009.

[94] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. On BFC-MSMIP: an exact branch-and-fix coordination approach for solving multistage stochastic mixed 0-1 problems. *TOP*, pages 96–122, 2009.

[95] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. On an exact algorithm for solving large-scale two-stage stochastic mixed-integer problems: theoretical and computational aspects. *European Journal of Operations Research*, pages 105–116, 2010.

[96] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. On BFC-MSMIP strategies for scenario cluster partitioning and twin node families branching selection and bounding for multi-stage stochastic mixed integer programming. *Computers & Operations Research*, 37:738–753, 2010.

[97] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. An algorithmic framework for solving large-scale multistage stochastic mixed 0-1 problems with nonsymmetric scenario trees. *Computers & Operations Research*, 39:1133–1144, 2012.

[98] L.F. Escudero, M.A. Garín, M. Merino, and G. Pérez. On solving strong multi-stage non-symmetric stochastic mixed 0-1 problems. In *Operations Research Proceedings, selected papers presented to OR2011, Zurich, Switzeland, 506-514. Springer*. H.-J. Luth D. Katte and K. Schemdders, editors, 2012.

[99] L.F. Escudero, J.F. Monge, D. Romero Morales, and J. Wang. Expected future value decomposition based bid price generation for large-scale network revenue management. *Transportation Science*, 47:181–197, 2013.

[100] L.F. Escudero and J. Salmerón. On a Fix-and-Relax framework for large-scale resource-constrained project scheduling. *Annals of Operations Research*, 140:163–187, 2005.

[101] B. Etkin and L. D. Reid. *Dynamics of flight : stability and control*. Wiley, New York, 1996.

[102] eurocontrol. Fasti atc manual. Technical report, Eurocontrol, accessed 2011.

[103] A. Ferguson and G. Dantzig. The allocation of aircraft to routes: An example of linear programming under uncertain demands. *Management Science*, 3:45–73, 1956.

[104] M.L. Fisher. The relaxation method for solving integer programming problems. *Management Science*, 27:1–18, 1981.

[105] Stein-Erik Fleten, Kjetil Høyland, and Stein W. Wallace. The performance of stochastic dynamic and fixed mix portfolio models. *European J. Oper. Res.*, 140(1):37–49, 2002.

[106] E. Fragniere, J. Gondzio, and J.P. Vial. Building and solving large-scale stochastic programs on an affordable distributed computing system. *Annals of Operations Research*, 99:167–187, 2000.

[107] L. Frantzeskakisand and W. Powell. A successive approximation procedure for stochastic dynamic vehicle allocation problems. *Transportation Science*, 24:40–57, 1990.

[108] E. Frazzoli, Z.H. Mao, J.-H. Oh, and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *AIAA Journal of Guidance, Control and Dynamics*, 24 i1:79–86, 2001.

[109] R.D. Galvão and E.R. Santibañez-González. A lagrangean heuristic for the p_k-median dynamic location problem. *European Journal of Operational Research*, 58:250–262, 1992.

[110] T. Gandhi, M.T. Yang, R. Kasturi, O.I. Camps, L. Coraor, and J. McCandless. Detection of obstacles in the flight path of an aircraft. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 39, pages 176–191, 2000.

[111] Y. Gao, X. Zhang, and X. Guan. Cooperative multi-aircraft conflict resolution based on coevolution. *International symposium on intrumentation and measurement, sensor network and automation (IMSNA)*, pages 310–313, 2012.

[112] A.M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.

[113] A. Geyer, W. Herold, K. Kontriner, and W.T. Ziemba. The Innovest Austrian Pension Fund Financial Planning Model INNOALM. Technical report, University of Vienna, 2002.

[114] R. Gollmer, U. Gotzes, and R. Schultz. A note on second-order stochastic dominance constraints induced by mixed-integer linear recourse. *Mathematical Programming, Ser. A*, 126:179–190, 2011.

[115] R. Gollmer, F. Neise, and R. Schultz. Stochastic programs with first-order stochastic dominance constraints induced by mixed-integer linear recourse. *SIAM Journal on Optimization*, 19:552–571, 2008.

[116] Jacek Gondzio and Roy Kouwenberg. High performance computing for asset liability management. Report MS-99-004, Econometric Institute, Erasmus University Rotterdam, The Netherlands, 1999. Available at `http://mail.tinbinst.nl/~kouwenbe/ms99004.ps`.

[117] C. Goodchild, M. A. Vilaplana, and S. Elefante. Co-operative optimal airborne separation assurance in free flight airspace. Third USA/Europe Air Trafic Management RD seminar, Napoli (Italy), 2000.

[118] N. Groewe-Kuska, K. Kiwiel, M.P. Nowak, W. Römisch, and I. Wegner. Power management in a hydro-thermal system under uncertainty by lagrangian relaxation. In C.Greengard y A. Ruszczynski, editor, *Decision making under uncertainty: Energy and Power*, pages 39–70. 2001.

[119] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI. Portable Parallel Programming with the Message–Passing Interface*. The MIT Press, 2nd edition, 1999.

[120] M. Guignard. Lagrangean decomposition and lagrangean substitution for stochastic integer programming. Technical report, OPIM Dept., Wharton School, University of Pennsylvania, Philadephia, (USA), 2003.

[121] J. Hoekstra, R. van Gent, and R. Ruigrok. Conceptual design of free flight with airborne separation assurance. In *AIAA Guidance, Navigat., Contr. Conf.*, volume AIAA-98-4239, pages 807–817, Boston, MA, Aug. 1998.

[122] L. Hong, L. Zhong-Hua, and Ch. Xue-Bin. Parallel computing for dynamic asset allocation based on the stochastic programming. In *Information Engineering (ICIE), 2010 WASE International Conference*, volume 2, pages 172–176, 2010.

[123] M. Høyland and S.W. Wallace. Generating scenario trees for multi-stage decision problems. *Management Science*, 47:295–307, 2001.

[124] J. Hu. *A Study of Conflict Detection and Resolution in Free Flight*. M.s. thesis, Department of EECS, University of California, 1999.

[125] J. Hu. *On Optimal Collision Avoidance and Formation Switching on Riemannian Manifolds*. M.a. thesis, Mathematics Department, University of California, 2002.

[126] J. Hu, M. Prandini, and S. Sastry. Optimal coordinated maneuvers for three dimensional aircraft conflict resolution. *AIAA Journal of Guidance, Control and Dynamics*, 25(5):888–900, 2002.

[127] J. Hu, M. Prandini, and S. Sastry. Optimal coordinated motions for multiple agents moving on a plane. *SIAM Journal on Control and Optimization*, 42(2):637–668, 2003.

[128] J. Hu, M. Prandini, and S. Sastry. Probabilistic safety analysis in three dimensional aircraft flight. In *Proceedings of 42nd IEEE Conference on Decision and Control.*, pages 5: 5335–5340, 2003.

[129] J. Hu, M. Prandini, and S. Sastry. Aircraft conflict prediction in the presence of a spatially correlated wind field. *IEEE Transations on Intelligent Transportation Systems*, 6(3):326–340, 2005.

[130] J. Hu, M. Prandini, and C. Tomlin. Interesting conjugate points in formation constrained optimal multi-agent coordination. *American Control Conference, 2005. Proceedings of the 2005*, 3:1871– 1876, 8-10 June 2005.

[131] J. Hu, M. Prandini, and C. Tomlin. Conjugate points in formation constrained optimal multi-agent coordination: A case study. *SIAM J. Control and Optimization*, 45:2119–2137, 2007.

[132] J. Hu and S. Sastry. Optimal collision avoidance and formation switching on rienmanian manifolds. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2, pages 1071–1076, 2001.

[133] D. G. Hull. *Fundamentals of Airplane Flight Mechanics*. Springer Publishing Company, 2007.

[134] K. Ishii, S. Nishida, K. Watanabe, and T. Ura. A collision avoidance system based on self-organizing map and its application to an underwater vehicle. *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, 2:602–607, 2002.

[135] K. Ishii, Syuhei Nishida, and Tamaki Ura. An adaptive learning method for som based navigation system and its application to an underwater robot. Speaking Conference, September 2003.

[136] S. Ratick J. Current and C. ReVelle. Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *European Journal of Operational Research*, 110:597–609, 1997.

[137] M.E. Captivo J. Dias and J. Climaco. Dynamic location problems with discrete expansion and reduction sizes of available capacities. *InvestigaÃ§Ã£o Operacional*, 27:107–130, 2007.

[138] M. R. Jardin. Real-time conflict-free trajectory optimization. Fifth USA/Europe Air Trafic Management RD seminar, Budapest (Hungary), 2003.

[139] M. R. Jardin. *Toward Real-Time en Route Air Traffic Control Optimization.* PhD thesis, Stanford University, (California), 2003.

[140] M. R. Jardin. Grid-based strategic air traffic conflict detection. In *2005 AIAA Guidance, Navigation, and Control Conference and Exhibit.*, 2005.

[141] M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. in approximation algorithms for NP hard problems. Technical report, Dorit Hochbaum, ed. PWS, 1996.

[142] P. Kall and J. Mayer. *Stochastic Linear Programming*. Springer, 2nd edition, 2011.

[143] A.S. Kenyon and D.P. Morton. Stochastic vehicle routing with random travel times. *Transportation Science*, 37:69–82, 2003.

[144] J.H. Kim, S. Hayakawa, T. Suzuki, K. Hayashi, S. Okuma, N. Tsuchida, M. Shimizu, and S. Kido. Modeling of driver's collision avoidance maneuver based on controller switching model. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(6):1131–1143, 2005.

[145] Willem K. Klein Haneveld and Maarten H. van der Vlerk. Optimizing electricity distribution using two-stage integer recourse models. In *Stochastic optimization: algorithms and applications (Gainesville, FL, 2000)*, volume 54 of *Appl. Optim.*, pages 137–154. Kluwer Acad. Publ., Dordrecht, 2001.

[146] J. Krozel and M. Peters. Strategic conflict detection and resolution for free flight. In *Proceedings of the 36th Conference on Decision & Control*, pages 1822–1828, December 1997.

[147] J. K. Kuchar and L. C. Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, 1:179–189, 2000.

[148] M.I. Kusy and W.T. Ziemba. A bank asset and liability management model. *Operations Research*, 34:356–376, 1986.

[149] G. Laporte and F.V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13:133–142, 1993.

[150] G. Laporte, F.V. Louveaux, and L. van Hamme. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50:415–423, 2002.

[151] M. Merino L.F. Escudero, A. Garín and G. Pérez. An algorithmic framework for solving large scale multistage stochastic mixed 0-1 problems with nonsymmetric scenario trees. *Computers & Operations Research*, 39:1133–1144, 2012.

[152] Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The empirical behavior of sampling methods for stochastic programming. *Ann. Oper. Res.*, 142:215–241, 2006.

[153] G. Laporte F.V. Louveaux and H. Mecure. An exact solution for the a priori optimization of the probabilistic traveling salesman problem. *Operations Research*, 39:71–78, 1994.

[154] M. Lucka, I. Melichercik, , and L. Halada. Application of multistage stochastic programs solved in parallel in portfolio management. *Parallel Computing*, 34:469–485, 2008.

[155] Y. Hinojosa M. Albareda-Sambola, E. Fernández and J. Puerto. The single period coverage facility location problem: Lagrangean heuristic and column generation approaches. *TOP*, 18:43–61, 2010.

[156] C.S. Ma and R. H. Miller. Mixed integer linear programming trajectory generation for autonomous nap-of-the earth flight in a threat environment. *Aerospace Conference, 2005 IEEE*, pages 1–9, March 2005.

[157] T. Magister. Conflict detection and resolution in the vicinity of the top of descent point. *Promet*, 14(6):269–275, 2002.

[158] T. Magister. Avoidance maneuvering in the vecinity of the top of descent. *Journal of Aerospace Engineering*, 17(4):176–181, 2004.

[159] A. Mandansky. Inequalities for stochastic linear programming problems. *Management Science*, 6:197–204, 1960.

[160] Z.-H. Mao, E. Feron, and K. Bilimoria. Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules. *IEEE Transactions on Intelligent Transportation Systems*, 2(2):101–109, 2001. ISSN 1524–9050.

[161] Z.H. Mao, D. Dugail, and E. Feron. Space partition for conflict resolution of intersecting flows of mobile agents. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):512–527, 2007. ISSN: 1524-9050.

[162] Z.H. Mao, D. Dugail, E. Feron, and K. Bilimoria. Stability of intersecting aircraft flows using heading-change maneuvers for conflict avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 6(4):357–369, 2005.

[163] H.M. Markowitz. *Porfolio Selection; Efficient Diversification of Investments*. John Wiley, 1959.

[164] K. Marti. *Stochastic Optimization Methods*. Springer, 2005.

[165] T.M. Martinetz and K.J. Schulten. A "neural-gas" network learns topologies. *Artifical Neural Networks,*, pages 397–402, 1991.

[166] F. Médioni, N. Durand, and J. Alliot. Air traffic conflict resolution by genetic algorithms. *Lect. Notes Comput. Sci.*, 1063:370–383, 1996.

[167] G. Meng and F. Qi. Flight conflict resolution for civil aviation based on ant colony optimization. *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium*, 1:239–241, 2012.

[168] S.A. MirHassani, C. Lucas, G. Mitra, and C.A. Poojari. Computational solution of a capacity planning model under uncertainty. *Parallel Computing Journal*, 26:511–538, 2000.

[169] D. Morton, R. Rosenthal, and L. Weng. Optimizitation for military airlift. *Military Operations Research*, 1:49–68, 1996.

[170] W.J. Hopp M.S. Daskin and B. Medina. Forecast horizons and dynamic facility location planning. *Annals of Operations Research*, 40:125–152, 1992.

[171] S. Nickel M.T. Melo and F. Saldanha da Gama. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Computers & Operations Research*, 33:181–208, 2006.

[172] M. Mukai, T. Kawabe, H. Nishira, Y. Takagi, and Y. Deguchi. On vehicle path generation method for collision avoidance using mixed integer programming. In *Proc. of the 2007 IEEE Conference on Control Applications*, 2007.

[173] M. Mukai, J. Murata, T. Kawabe, H. Nisira, Y. Takagi, and Y. Deguchi. Optimal path generation for automotive collision avoidance using mixed integer programming. *SICE Journal of Control, Measurement and System Integration*, 1(3):222–226, 2008.

[174] J.M. Mulvey, G. Gould, and C.Morgan. An asset and liability management system for Towers Perrin-Tillinghast. *Interfaces*, 30:96–114, 2000.

[175] J.M. Mulvey and H.Vladimirou. Stochastic networks, optimization models for investment planning. *Annals of Operations Research*, 20:187–217, 1989.

[176] J.M. Mulvey and A. Ruszczynski. A diagonal quadratic approximation method for large-scale linear programs. *Operations Research Letters*, 12:205–221, 1992.

[177] J.M. Mulvey and A. Ruszczynski. A new scenario decomposition method for large-scale stochastic optimization. *Operations Research*, 43:477–490, 1995.

[178] NASA. Workshop flight deck centered parallel runway approaches in instrument meteorological conditions. In M. Waller and C. Scanlon, editors, *Proceedings of the NASA Workshop on Flight Deck Centered Parallel Runway Approaches in Instrument Meteorological Conditions*, pages 8–19, Hampton, VA, Dec 1996. NASA Conf. Pub. 10191.

[179] S. Nishida, K. Ishii, and T. Furukawa. An adaptive controller system using mnsom. International Congress, 2006.

[180] V.I. Norkin, Y.M. Ermoliev, and A. Ruszczynski. On optimal allocation of indivisibles under uncertainty. *Operations Research*, 46:381–395, 1998.

[181] V.I. Norkin, G.Ch. Pflug, and A. Ruszczynski. A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83:425–450, 1998.

[182] M.P. Novak, R. Schultz, and M. Westphalen. A stochastic integer programming model for incorporating day-ahead trading of electricity into hydro-thermal unit commitment. *Optimization and Engineering*, pages 163–176, 2005.

[183] R. Nürnberg and W. Römisch. A two-stage planning model for power scheduling in a hydro-thermal system under uncertainty. *Optimization in Engineering*, 3:355–378, 2002.

[184] A. Pagès-Bernaus, G. Pérez-Valdés, and A. Tomasgard. A parallelised distributed implementation of a Branch-and-Fix Coordination algorithm. *European Journal of Operational Research*, 244:77–85, 2015.

[185] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):3–11, 2002.

[186] J. J. Pannequin, A. M. Bayen, I. M. Mitchell, H. Chung, and S. Sastry. Multiple aircraft deconflicted path planning with weather avoidance constraints. In *AIAA Guidance, Navigation and Control Conference. Hilton Head, (South Carolina)*, 2007.

[187] L. Peng and Y. Lin. Study on the model for horizontal escape maneuvers in tcas. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):392–398, June 2010.

[188] M.V.F. Pereira and L.M.V.G. Pinto. Stochastic optimization of a multireservior hydroelectric system - a decomposition approach. *Water Resources Research*, 21:779–792, 1985.

[189] M.V.F. Pereira and L.M.V.G. Pinto. Multistage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359–375, 1991.

[190] C. Peyronne, A. Conn, M. Mongeau, and D. Delahaye. Solving air-traffic conflict problems via local continuous optimization. *European Journal of Operational Research*, 2014.

[191] G.Ch. Pflug and A. Pichler. *Multistage Stochastic Optimization*. Springer, 2014.

[192] W. Powell. A comparative review of alternative algorithms for the dynamic vehicle allocation program. In B. Golden and A. Assad, editors, *Vehicle Routing: Methods and Studies*. Noth-Holland, 1988.

[193] W. Powell. Real-time optimization for truckload motor carriers. *OR/MS Today*, 17:28–33, 1990.

[194] W. Powell. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science*, 3:195–219, 1996.

[195] W. Powell and D. Gittoes. An approximate labeling algorithm for the dynamic assignment problem. In M.B.L. Bianco and P. Toth, editors, *Advanced Methods in Transportation Analysis*. Springer,New York, 1996.

[196] W. Powell, W. Snow, and R.K.M. Cheung. Adaptive labeling algorithms for the dynamic assignment problem. *Transportation Science*, 2004.

[197] M. Prandini and J. Hu. Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction. Technical report, IEEE Transactions on Automatic Control, 2008.

[198] M. Prandini, J. Hu, J. Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Trans. on Intelligent Transportation Systems. Special issue on Air Traffic Control*, 1(4):199–220, December 2000.

[199] M. Prandini, J. Lygeros, A. Nilim, and S. Sastry. A probabilistic framework for aircraft conflict detection. In *AIAA Guidance, Navigation and Control*, 1999.

[200] T. Prevot, V. Batiste, E. Palmer, and S. Shelden. Air traffic concept utilizing 4d trajectories and airborne separation assistance. AIAA, Reston (Virginia), 2003.

[201] H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Harvard University, Boston, MA, (USA), 1961.

[202] J. J. Rebollo, I. Maza, and A. Ollero. A two step velocity planning method for real-time collision avoidance of multiple aerial robots in dynamic enviroments. Seventeenth World Congress. The International Federation of Automatic Control, Seoul (Korea), 2008.

[203] D. Rey, C. Rapine, R. Fondacci, and N.E. El Faouzi. Minimization of potentials air conflicts through speed regulation. *Transportation Research Record*, 2300, 2012.

[204] A. G. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. American Control Conference, Anchorage (Alaska), 2002.

[205] A. G. Richards and J. P. How. Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. IEEE American Control Conference, 2003.

[206] M. Riis, A. Skriver, and J. Lodahl. Deployment of mobile switching centers in a telecommunications network: A stochastic programming approach. http://home.imf.au.dk/riis/, 2004.

[207] R.T. Rockafellar and S.P. Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2:21–41, 2000.

[208] R.T. Rockafellar and S.P. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471, 2002.

[209] R.T. Rockafellar and R.J-B Wets. Scenario and policy aggregation in optimisation under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[210] R.T. Rockafellar and R.J-B Wets. Scenario and policy aggregation in optimisation under uncertainty. *Mathematics of Operations Research*, 52:359–375, 1991.

[211] R.T. Rockafellar and Roger J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.*, 16(1):119–147, 1991.

[212] Radio Technical Committee on Aeronautics RTCA. Minimum performance specifications for tcas airborne equipment. Document No. RTCA/DO185, Washington, DC, 1983.

[213] A. Ruszczynski. An augmented lagrangian decomposition method for block diagonal linear programming problems. *Operations Research Letters*, 8:287–294, 1989.

[214] S.Ahmed and R. García. Dynamic capacity acquisition and assignment under uncertainty. *Annals of Operational Research*, pages 267–283, 2004.

[215] T. Schouwenaars, M. Valenti, E. Feron, and J. How. Implementation and flight test results of milp-based uav guidance. In *Aerospace Conference, 2005 IEEE*, pages 1–13, 2005.

[216] R. Schultz, L. Stougie, and M.H.van der Vlerk. Two-stage stochastic integer programming: A survey. *Statistica Neerlandica*, 50:404–416, 1996.

[217] S. Sen. Algorithms for stochastic mixed-integer programming models. In G. Nemhauser and R. Weismantel, editors, *Stochastic Integer Programming Handbook*. North-Holland, 2003.

[218] S. Sen, R.D. Doverspike, and S. Cosares. Network planning with random demand. *Telecommunications Systems*, 3:11–30, 1994.

[219] S. Sen and H.D. Sherali. Decomposition with branch-and-cut approaches for two stage stochastic mixed-integer programming. *Mathematical Programming*, pages 203–223, 2006.

[220] T. Sharma, P. Williams, C. Bil, and A. Eberhard. Optimal three dimensional aircraft terrain following and collision avoidance. *Anziam J.*, (47):C695–C711, 2007.

[221] K. Sigurd and J. How. Uav trajectory design using total field collision avoidance. AIAA Guidance, Navigation and Control conference, 2003.

[222] M. Soler, A.Olivares, and E. Staffetti. Multiphase optimal control framework for commercial aircraft four-dimensional flight-planning problems. *Journal of Aircraft*, 52:274–286, 2014.

[223] L. Stougie. *Design and analysis of algorithms for stochastic integer programming*, volume 37 of *CWI Tract*. Stichting Mathematisch Centrum Centrum voor Wiskunde en Informatica, Amsterdam, 1987.

[224] S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, 1996.

[225] A. Tewari. *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB and Simulink*. Birkhäuser Basel, 2007.

[226] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[227] K. Treleaven. *Conflict Resolution and Traffic Complexity of Multiple Intersecting Flows of Aircraft*. PhD thesis, Faculty of the School of Engineering. University of Pittsburgh, USA, 2007.

[228] S. Uryasev and P.M. Pardalos. *Stochastic Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 2001.

[229] Maarten H. van der Vlerk. *Stochastic programming with integer recourse*. PhD thesis, University of Groningen, The Netherlands, 1995.

[230] T.J. van Roy and D. Erlenkotter. A dual-based procedure for dynamic facility location. *Management Science*, 28:1091–1105, 1982.

[231] R. Van Slyke and R-J.B. Wets. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[232] R. Vivona, D. Karr, and D. Roscoe. Pattern based genetic algorithm for airborne conflict resolution. *AIAA Guidance, Navigation and Control Conference Exhibition, AIAA, Keystone, CO, USA,*, 2006.

[233] Hercules Vladimirou. Computational assessment of distributed decomposition methods for stochastic linear programs. *European Journal of Operational Research*, 108:653–670, 1998.

[234] A. Wald. *Statistical Decision Functions*. J. Wiley, 1950.

[235] S.W. Wallace and W.T. Ziemba. *Applications of Stochastic Programming*. MPS-SIAM Series in Optimization, 2005.

[236] Q. Wang, E. Batta, and C.M. Rump. Facility location models for immobile servers with stochastic demand. *Naval Research Logistics. To appear*, 2003.

[237] A. Warszawski. Multi-dimensional location problems. *Operational Research Quarterly*, 24:165–179, 1973.

[238] J.P. Watson and D. Woodruff. Progressive hedging innovations for a class of stochastic mixed integer resource allocation problems. *Computational Management Science*, 8:355–370, 2011.

[239] Roger J.-B. Wets. Stochastic programs with fixed recourse: the equivalent deterministic program. *SIAM Rev.*, 16:309–339, 1974.

[240] Roger J.-B. Wets. On the relation between stochastic and deterministic optimization. In *Control Theory, numer. Meth., Computer Syst. Mod.; internat. Symp. Rocquencourt 1974, Lecture Notes Econ. math. Syst. 107, 350-361,* 1975.

[241] L.A. Wolsey. *Integer Programming*. John Wiley, 1998.

[242] D.L. Woodruff. *Stochastic Integer Programming and Network Interdiction Models*. Kluwer Academic Press, 2002.

[243] J. Puerto Y. Hinojosa and F.R. Fernández. A multiperiod two-echelon multicommodity capacitated plant location problem. *European Journal of Operational Research*, 123:45–65, 2000.

[244] K. Zeghal and E. Hoffman. Design of cockpit displays for limited delegation of separation assurance. In *18th Digital Avion. Syst. Conf.*, St. Louis, MO, Oct. 1999.