Chapter 5
Scenario decomposition framework for mixed-integer second-stage problem: stochastic Lagrangean decomposition

The most common framework for dealing with uncertainty in optimization models is the two-stage stochastic programming. Typically, two-stage stochastic programming models comprise two types of decisions: first-stage decision that must be taken prior to knowing the realization of the uncertainty, and second-stage decisions that represent recourse measures that can be taken after the uncertainty unveils. The objective is then to minimize both first-stage and expected recourse costs. In some cases, it might also be suitable to consider some sort of risk measure together with the expected recourse cost in order to avoid incurring in high costs for some of the realization of the uncertainties (You et al., 2009).

If some of the stage-two variables are required to be integer, we have what is known as a stochastic integer programming (SIP) problem (Carøe and Schultz, 1999). When this is the case, the stochastic programming problem loses desirable properties such as convexity and continuity of the recourse cost function. In this context, solution methods that rely on the use of dual results from linear programming, such as the L-Shaped algorithm and its variants (Van Slyke and Wets, 1969; Birge and Louveaux, 1988), cannot be directly applied to the stochastic problem with second-stage integer variables. Moreover, the expected recourse function is discontinuous and the set of first-stage decisions that yield second-stage solutions in known to be nonconvex in such cases.

In order to deal with this issue, several researchers have proposed approaches that are capable of dealing with stochastic integer programming problems. These approaches either try to adapt the L-Shaped algorithm into the context of stochastic integer programming problems through the use of convexification techniques for the second-stage problem (see, for example Laporte and Louveaux (1993); Sherali and Fraticelli (2002); Zheng et al. (2012)), enumerative branch-and-bound strategies (see for example Carøe and Schultz (1999);

Norkin et al. (1998); Ahmed et al. (2004)), or else apply dual decomposition methods by means of Lagrangean decomposition approaches (Carøe and Schultz, 1999). In this case, the problem is decomposed into scenario subproblems through the relaxation of non-anticipativity constraints (NAC) and the solution strategy relies on finding the optimal dual multipliers. Several methods have been proposed in the literature for solving the dual problem associated with the Lagrangean decomposition. The techniques available include the classical subgradient method (Held and Karp, 1971; Held et al., 1974), cutting-plane approaches (Kelley, 1960; Redondo and Conejo, 1999), the volume algorithm (Barahona and Anbil, 2000), bundle methods (Lemarechal, 1989; Zhao and Luh, 2002), and augmented Lagrangean methods (Ruszczyński, 1995; Li and Ierapetritou, 2012).

In this chapter we present a comprehensive framework for the multi-product, multi-period supply chain investment planning problem considering network design and discrete capacity expansion under demand uncertainty. In this case, we consider the existence of integer decisions in the second-stage problem, which requires a different approach from the one presented in chapter 4. Some of the novel features that we present include the implementation of a Lagrangean decomposition to solve a large-scale problem from a real world case study, an algorithmic approach for solving the dual Lagrangean problem, and a comparison between the computational performance of different formulations for the nonanticipativity conditions (NAC). Furthermore, we consider a risk measure that allows us to reduce the probability of incurring in high costs while preserving the decomposable characteristic of the problem.

5.1 Mathematical Model

In chapter 4 we have considered a simplified version of the supply chain investment planning model presented in chapter 2. In the same spirit of what was done in chapter 4, we consider here a simplified version of the complete model 2.1 to 2.18 presented in chapter 2, however with some additional features.

In this chapter we consider that, since we are dealing with uncertainty of the demand levels, it might be the case that the base does not have enough tankage available to deal with unexpectedly high demand. When this is the case, transportation ships can be used as temporary tanks in places where marine access is available. This is only done under emergency circumstances due to the high impact on the logistic costs.

(a) Nomenclature

The mathematical model considered in this section uses the same nomenclature presented in Table 4.1, except for the terms that are not comprised in the version of the model 4.1 to 4.10 presented in chapter 4. Table 5.1 lists the additional terms considered.

Parameters	
E_{jt}	Emergency floating tankage cost
F_{jt}	Emergency floating tankage capacity
Variables	
z^{ξ}_{jt}	Emergency tankage contract decision

Table 5.1: Model Additional Notation

(b) Model Formulation

The mathematical model used in this section is very similar to the model 4.1 to 4.10 presented in chapter 4, except for the objective function and constraints 4.5, 4.8, and 4.10 of the second-stage problem. This is due to the fact that in this chapter we removed the continuous variable u^{ξ}_{jpt} representing the unmet demand from constraint 4.5 and added the term $F_{jt}z^{\xi}_{jt}$ to constraints 4.8 and 4.10. In addition, we replaced the term $\sum_{j,p,t} S_{jpt}u^{\xi}_{jpt}$ that represented the shortfall cost by $\sum_{j,t} E_{jt}z_{jt}$, which represents the costs for hiring emergency floating capacity. For the sake of completeness, we state below the second-stage model considered in this chapter.

$$\mathcal{Q}(w,y) = \min_{x,v\in\mathbb{R}^{+},z\in\{0,1\}} \sum_{\xi} P^{\xi}\left(\sum_{i,j,p,t} C_{ijt}x^{\xi}_{ijpt} + \sum_{j,p,t} H_{jpt}v^{\xi}_{jpt} + \sum_{j,t} E_{jt}z_{jt}\right) \tag{5.1}$$

s.t.:

$$\sum_{i} x^{\xi}_{ijpt} + v^{\xi}_{jpt-1} = \sum_{i} x^{\xi}_{jipt} + v^{\xi}_{jpt} + D^{\xi}_{jpt} \quad \forall j \in \mathcal{B}, p \in \mathcal{P}, t \in \mathcal{T}, \xi \in \Omega \tag{5.2}$$

$$v^{\xi}_{jpt} \leq M^{0}_{jp} + M_{jp}\sum_{t'\leq t} w_{jpt'} + F_{jt}z^{\xi}_{jt} \quad \forall j \in \mathcal{B}, p \in \mathcal{P}, t \in \mathcal{T}, \xi \in \Omega \tag{5.3}$$

$$\sum_{i} x^{\xi}_{ijpt} \leq K_{jp}\left(M^{0}_{jp} + M_{jp}\sum_{t'\leq t} w_{jpt'}\right) + F_{jt}z^{\xi}_{jt} \quad \forall j \in \mathcal{B}, p \in \mathcal{P}, t \in \mathcal{T}, \xi \in \Omega \tag{5.4}$$

4.6, 4.7, and 4.9

The objective function 5.1 represents freight costs between nodes, inventory costs, and costs for hiring emergency floating capacity. Equation 5.2 comprises the material balance in distribution bases. Constraint 5.3 defines the storage capacities together with their expansion possibility and the additional emergency capacity that might be necessary. Constraint 5.4 sets the throughput limit for bases, defined by the product of the available storage capacity with the throughput capacity multiplier, and the possibility of expanding them by means of additional floating tankage.

5.2 Solution Algorithm

From our early investigations, we observed that large-scale instances of the stochastic supply chain investment planning problem presented in section 5.1 cannot be solved in full space. In this sense, we consider that scenario-wise Lagrangean decomposition is an alternative to overcome this challenge. In the following section, we detail the algorithmic strategy for solving the aforementioned problem. Our method integrates a scenario-wise decomposition based on Lagrangean decomposition and novel approach for updating the Lagrangean multiplier set.

(a) Lagrangean Decomposition Approach

To illustrate the proposed approach, we will consider that we have the supply chain investment planning problem presented in section 5.1 written in the following compact notation in the reminder of this chapter.

$$v = \min_{x,y} cx + \sum_{\xi} P^{\xi} q y^{\xi} \tag{5.5}$$

s.t.:

$$Ax \leq b \tag{5.6}$$

$$Tx + Wy^{\xi} \leq h^{\xi} \quad \forall \xi \in \Omega \tag{5.7}$$

$$x \in \{0,1\}^n \tag{5.8}$$

$$y^{\xi} \in Y \quad \forall \xi \in \Omega \tag{5.9}$$

where c is a n-dimensional vector, q is a p-dimensional vector, A is a $m \times n$ matrix, b is a m-dimensional vector, T and W are matrices of size $q \times n$ and $q \times p$, respectively, and h is q-dimensional vector. In our context, cx represents our investment costs (i.e., first-stage costs), while $\sum_{\xi} P^{\xi} q y^{\xi}$ represents the costs with freight, inventory, and emergency tankage acquisition. (i.e., second-stage costs). The set of constraints $Ax \leq b$ represents constraints 4.2 and 4.3,

while $Tx + Wy^{\xi} \leq h^{\xi}$ represents constraints 4.7, 4.9, 5.3, and 5.4. Finally, set Y denotes constraints 4.6 and 5.2, as well as restrictions regarding mixed 0-1 variable domains.

As it is commonly known, this class of problem exhibits a block-angular structure that can be exploited in a decomposition fashion, provided that we are able to split it into more manageable pieces. One possible way of making it possible to decompose the problem is to use Lagrangean decomposition (Fisher, 1985; Guignard and Rosenwein, 1989) in the context of stochastic optimization. Such a procedure was first proposed by (Carøe and Schultz, 1999) allowing the problem to be decomposed into scenario subproblems. The idea behind this scenario decomposition approach is to create copies $x^1, \ldots, x^{|\Omega|}$ of the first-stage variables and then rewrite the problem as follows:

$$v = \min_{x,y} \sum_{\xi} P^{\xi} \left(cx^{\xi} + qy^{\xi} \right) \tag{5.10}$$

s.t.:

$$Ax^{\xi} \leq b \quad \forall \xi \in \Omega \tag{5.11}$$

$$Tx^{\xi} + Wy^{\xi} \leq h^{\xi} \quad \forall \xi \in \Omega \tag{5.12}$$

$$y^{\xi} \in Y \quad \forall \xi \in \Omega \tag{5.13}$$

$$x^1 = \cdots = x^{|\Omega|} \tag{5.14}$$

Equalities 5.14 correspond to the nonanticipativity constraints (NAC for short, as defined in the beginning of this chapter). As the name suggests, these constraints state that the first-stage decisions must not depend on any particular scenario which will prevail in the second stage. In other words, it means that we cannot have particular first-stage solutions for specific scenarios given that these solutions must be taken prior to the uncertainty realization.

There are several ways of representing NAC. They can be expressed in aggregated form, where a single constraint is used to express the non-anticipativity property, or considering a disaggregated form, in which individual NAC are used to represent the non-anticipativity between the first-stage variables in a pairwise fashion. In order to be able to decide between these two representations one must consider the inherent trade-off between them. Even though the aggregate constraint yields in general weaker linear relaxations than the conjunction of the NAC, the advantage is that fewer Lagrangean multipliers are needed, which might make it easier to find good values. Using the disaggregate formulation requires a larger number of multipliers, although with the potential benefit of having more control when it comes to the search of good multiplier values.

In our early experimentations, we observed that the disaggregated representation of NAC provided better computational results in our context. Therefore, we will only consider this type of representation hereinafter. Nevertheless, even the disaggregated representation can be done in different manners. In this thesis, we consider two different ways of formulating disaggregated NAC.

The first formulation assigns the scenario copy variables in a sequential fashion. In this way, we can replace condition 5.14 by the following set of constraints:

$$x^{\xi} = x^{\xi+1} \quad \forall \xi = 1, \ldots, |\Omega| - 1 \tag{5.15}$$

The other representation consists of associating the scenario copy variables considering one scenario (say the first scenario) as a reference to other copy variables. By doing this, an asymmetric structure is created regarding the set of constraints that represent the non-anticipativity conditions. In this case, the NAC are formulated as follows:

$$x^{1} = x^{\xi} \quad \forall \xi = 2, \ldots, |\Omega| \tag{5.16}$$

Independent of which representation one might choose, the Lagrangean relaxation with respect to the non-anticipativity condition 5.14 is the problem of finding $x^{\xi}, y^{\xi}, \xi = 1, \ldots, |\Omega|$ such that:

$$D(\lambda) = \min_{x,y} \sum_{\xi} P^{\xi} \left(c x^{\xi} + q y^{\xi} \right) + \sum_{\xi} \lambda^{\xi} s^{\xi} \tag{5.17}$$

s.t.:

$$Ax^{\xi} \leq b \quad \forall \xi \in \Omega \tag{5.18}$$

$$T_{\xi}^{\xi} x + W_{\xi} y^{\xi} \leq h^{\xi} \quad \forall \xi \in \Omega \tag{5.19}$$

$$y^{\xi} \in Y \tag{5.20}$$

where $s^{\xi} = x^{\xi} - x^{\xi+1}, \xi = 1, \ldots, |\Omega| - 1$ for the sequential representation, $s^{\xi} = x^{1} - x^{\xi}, \xi = 2, \ldots, |\Omega|$ for the asymmetric representation, and λ is $(|\Omega|-1)$-dimensional vector. The Lagrangean dual then becomes the problem of finding λ such that

$$v_{LD} = \max_{\lambda} D(\lambda) \tag{5.21}$$

Duality theory establishes that $v \geq v_{LD}$ Guignard (2003). In particular, for nonconvex cases such as MILP models, we may have that $v > v_{LD}$, which implies the existence of a duality gap. This fact is a well known result and can

be found in Carøe and Schultz (1999).

One important property of the Lagrangian dual problem 5.21 is that it is a convex non-smooth program, which splits into separate subproblems for each scenario ξ that can be solved independently. Each scenario subproblem can be then stated in the following form:

$$D^{\xi}(\lambda) = \min_{x,y} P^{\xi}\left(cx^{\xi} + qy^{\xi}\right) + f^{\xi}(\lambda)x^{\xi} \tag{5.22}$$

s.t.:

$$Ax^{\xi} \leq b \tag{5.23}$$

$$T_{\xi}^{\xi}x + W_{\xi}y^{\xi} \leq h^{\xi} \tag{5.24}$$

$$y^{\xi} \in Y \tag{5.25}$$

where $D(\lambda) = \sum_{\xi} D^{\xi}(\lambda)$ and $f^{\xi}(\lambda)$ is given depending on the chosen formulation for the NAC. For the sequential case we have that

$$f^{\xi}(\lambda) = \begin{cases} \lambda^{1}, \text{ if } \xi = 1 \\ -\lambda^{|\Omega|}, \text{ if } \xi = |\Omega| \\ \lambda^{\xi} - \lambda^{\xi-1}, \text{ otherwise} \end{cases}$$

and for the asymmetric formulation, we have that

$$f^{\xi}(\lambda) = \begin{cases} \sum_{\xi=2}^{|\Omega|} \lambda^{\xi}, \text{ if } \xi = 1 \\ -\lambda^{\xi}, \text{ otherwise} \end{cases}$$

Note that each one of the scenario subproblems are completely independent and, thus, this type of decomposition could benefit from the use of parallel computation.

(b) Proposed strategy for solving the Lagrangean Dual

Although computationally convenient, the decomposition framework presented in section 5.2(a) does not solve the original full-space problem. Nevertheless, it is widely known that the Lagrangean dual represents a relaxation of the original problem for any given set of Lagrange multipliers (Guignard, 2003). In this sense, we can concentrate our efforts in finding better multipliers sets, i.e. multipliers yielding tighter relaxations to the original problem that approximate as much as possible the solution of the Lagrangean dual to the solution of the full-space problem.

Typically, Lagrangean decomposition algorithms rely on successively

solving the Lagrangean subproblems for sequentially improving Lagrange multipliers that are obtained based on the use of some information available after solving these subproblems. The algorithm starts with an initial guess for the Lagrange multipliers, which can be obtained by some problem-specific strategy, such as dual values of NAC from the linear relaxation of the complete problem, or even set to prespecified values (e.g, zero). Then, at each iteration the Lagrangean subproblems are solved and a relaxed solution is obtained. The algorithm stops when some of the convergence criteria are satisfied. Otherwise, the Lagrange multipliers are updated and the algorithm proceeds towards the next iteration. Figure 5.1 schematically illustrates the proposed algorithm. The algorithm starts at an initialization step, where the Lagrange multipliers and other parameters are set to their initial value. The algorithm then proceeds with the iterative solution of the Lagrangean dual, followed by the procedure that seeks to obtain a feasible solution from the solution obtained in the Lagrangean dual. This iterative procedure continues until the Lagrangean dual solution and the feasible solution are close enough, as measured by some stoping criteria. If the criteria are not fulfilled yet, the procedure iterates using information from these two previous steps to update the Lagrangean multipliers and iterate once more. Notice that, even though the algorithm follows a classical iterative framework, it has particular features that differ from traditional approaches.

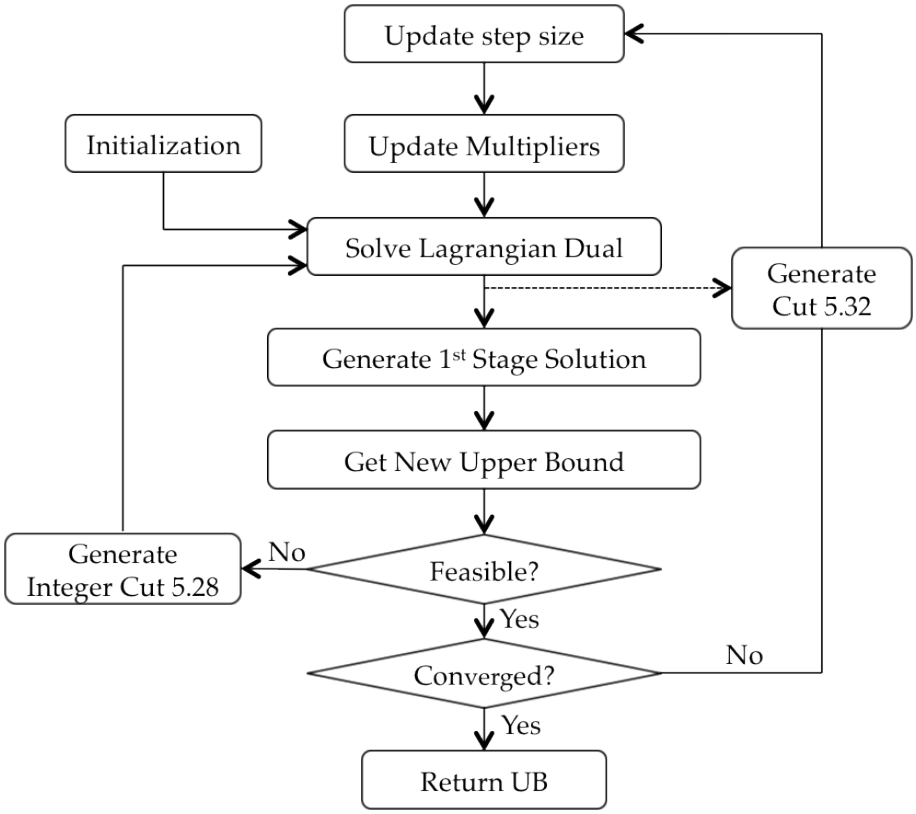

Figure 5.1: Schematic representation of the proposed Lagrangean decomposition

The most common method used to obtain solutions to the Lagrangean dual is the subgradient method (Held and Karp, 1971; Held et al., 1974). The method relies on the use of subgradient information available after solving the Lagrangean relaxation to predict improvement directions for the multipliers, as well as step sizes. Usually, this approach is preferred because of its ease of implementation added to its capability of predicting reasonably good Lagrangean multipliers for many cases. Nevertheless, special care must be taken in terms of selecting good strategies for defining and updating the subgradient step size.

Unfortunately, it is sometimes reported in the literature that the subgradient approach might fail to achieve regarding its convergence. To circumvent this drawback, many researchers have searched for improvements to this technique over the years.

One alternative considered is the use of cutting-planes for approximating the Lagrangean dual function. Cutting-plane strategies are available since the early 60's after the seminal work of Kelley (1960). Nevertheless, this approach might require a large number of iterations in order to yield good approximations of the Lagrangean dual function and thus, good multiplier

updates. This effect is mainly due to the fact that at the early iterations of the algorithm, the number of cutting planes generated is too few to provide good Lagrangean multipliers. One possible way of improving these kind of approaches is to consider trust-regions for the multipliers in the early iterations, so that this effect is controlled (Mouret et al., 2011).

In this chapter we introduce a novel hybrid approach where we seek to combine the ideas from the aforementioned methods into a single framework. The main idea behind the algorithm is to combine cutting plane generation using the dual information obtained from the solution of the Lagrangean dual problem with subgradients that provides approximated ascent directions. Moreover, we use the step size predicted by the subgradient method as a trust-region for the multipliers update process. As will be seen in the computational results section, the combination of these techniques provides effective Lagrangean multiplier updates, while ensuring good convergence properties.

(c) Upper bounding procedure

One specific feature of the proposed algorithm is the particular heuristic that uses information derived from the solution of the Lagrangean dual problem to derive a feasible solution and a valid upper bound to the full-space problem. It should be noted that it is not computationally demanding to calculate an upper bound for the full-space problem, once a first-stage solution is available. This is mainly due to the fact that for a fixed first-stage solution, the full-space problem becomes decomposable in scenarios.

The heuristic is based in the following formation rule. Consider a given iteration K. First, we calculate α_K as

$$\alpha_K = \sum_{\xi \in \Omega} P^{\xi} x_K^{\xi} - \sum_{\xi \in \Omega} P^{\xi}(1 - x_K^{\xi}) \tag{5.26}$$

If $\alpha_K > 0$, the investment (i.e., a combination of location and product in the case of capacity expansion or origin and destination, in the case of network design) is selected to compose the feasible solution. The time period for the selected investment will be the earliest among the scenarios where the investment was decided. We choose the earliest time period as the one to be implemented based on the observation that the costs incurred by recourse actions are typically larger than the increase in first-stage costs due to investing earlier in a given project. In addition to that, one might notice that the existence of more logistic options allows the system to possibly reach more efficient and less costly logistics, which yields economies of scale.

Since we are using a heuristic to generating solutions based on information that comes from scenarios individually, it might be the case that the solution generated is not feasible for the full-space problem. If it is the case, then we use an integer cut to remove this infeasible solution from the search space of the relaxed dual. Let $X_1 = \{j \mid x_j = 1\}$ and $X_0 = \{j \mid x_j = 0\}$. Then, we can write the integer cut as

$$\sum_{j \in X_0} x_j + \sum_{j \in X_1} (1 - x_j) \geq 1 \tag{5.27}$$

and add it to every scenario subproblem. We then solve again the Lagrangean relaxation and proceed with algorithm execution.

(d) Multiplier updating procedure

The most common method for updating the Lagrangean multipliers is the subgradient algorithm (Held and Karp, 1971; Held et al., 1974). This algorithm consists of an iterative method in which at a given iteration K, with a current set of Lagrangean multipliers λ_K a step is taken along the subgradient of $D(\lambda)$. Let s_K the the subgradient vector of dimension $(|\Omega| - 1)$ with components given as $s_K^\xi = x_K^\xi - x_K^{\xi+1}, \xi = 1, \ldots, |\Omega| - 1$ for the sequential representation and $s_K^\xi = x_K^1 - x_K^\xi, \xi = 2, \ldots, |\Omega|$ for the asymmetric representation, where $x_K^\xi, \xi = 1, ..., |\Omega|$ is the solution for the Lagrangean dual given λ_K. Then, the Lagrange multipliers are updated using the subgradient information as follows:

$$\lambda_{K+1}^\xi = \lambda_K^\xi + \theta_K \frac{(UB - LB_K)}{\sum_\xi (s_K^\xi)^2} s_K^\xi \quad \forall \xi \in \Omega \tag{5.28}$$

where UB is an approximation to the optimal value for v and $LB_K = D(\lambda_K)$. The term $\theta_K \in (0, 2]$ is used to correct the error in the estimation of the true optimal value and is usually selected using heuristic rules. The new set of Lagrangean multipliers λ_{K+1} is then used as an input for solving again the Lagrangean dual problem. The procedure continues until reaching the limit in the number of iterations, or unitl some stopping criteria is met, such as minimum improvement in the magnitude of $D(\lambda_K)$ in some norm (say a l_2-norm) of the subgradient vector (Guignard, 2003).

An alternative procedure for updating the Lagrangean multipliers is based on the use of cutting planes to approximate the Lagrangean dual function. In this type of approaches, the solutions obtained from the Lagrangean dual are used to generate supporting hyperplanes (commonly referred as cuts in the optimization literature) that iteratively generate an approximation for

the Lagrangean dual function from which new multipliers are then successively derived. Given a certain iteration K, the Lagrange multipliers can be obtained solving the following auxiliary problem:

$$\max_{\eta,\lambda} \eta \tag{5.29}$$

s.t.:

$$\eta \leq \sum_{\xi} P^{\xi} \left(cx_k^{\xi} + qy_k^{\xi} \right) + \sum_{\xi} \lambda^{\xi} s_k^{\xi} \quad \forall k = 1, \ldots, K \tag{5.30}$$

where 5.30 represents the cuts generated up to iteration K with the information available in each iteration $k = 1, \ldots, K$. The main drawback associated with this type of approach is that it is commonly known that this problem is always unbounded during early iterations of the algorithm (Mouret et al., 2011), making necessary that some valid bounds are imposed on the multipliers in order to prevent unboundedness.

In our approach we seek to combine both subgradient and cutting-plan strategies in a hybrid approach, aiming at developing an efficient manner of updating the Lagrangean multiplier set. In the proposed strategy, the Lagrangean multiplier updates are done by solving the following optimization problem in a given Kth iteration:

$$\max_{\eta,\lambda} \eta \tag{5.31}$$

s.t.:

$$\eta \leq \sum_{\xi} P^{\xi} \left(cx_k^{\xi} + qy_k^{\xi} \right) + \sum_{\xi} \lambda^{\xi} s_k^{\xi} \quad \forall k = 1, \ldots, K \tag{5.32}$$

$$\lambda_{K-1}^{\xi} - \theta_K \frac{(UB - LB_K)}{\sum_{\xi} (s_K^{\xi})^2} |s_K^{\xi}| \leq \lambda^{\xi} \leq \lambda_{K-1}^{\xi} + \theta_K \frac{(UB - LB_K)}{\sum_{\xi} (s_K^{\xi})^2} |s_K^{\xi}| \quad \forall \xi \in \Omega \tag{5.33}$$

The objective function 5.31 and constraint 5.32 correspond to the optimization problem for the traditional cutting plane approach for updating the Lagrange multipliers. However, in the proposed algorithm we use a dynamically updated trust-region for the Lagrangean multipliers in order to circumvent unboundedness issues. To construct this trust-region, we use subgradient information available up to the current iteration to define step sizes, in the same spirit of what is done in the classical subgradient method. However, in our case the multipliers are selected considering an optimization framework rather than heuristically updating its values. Constraint 5.33 represents the aforementioned trust-region.

Note that we use a step length parameter θ_K to adjust the length of the step size. This parameter is dynamically updated during the algorithm execution following the ideas firstly presented in Barahona and Anbil (2000). We define three types of iterations according to the dual solution value $D(\lambda_K)$ obtained in the Kth iteration. The first type is when we observe that $D(\lambda_K) < D(\lambda_{K-1})$. This is what is called a "negative" iteration. In this case we make $\theta_K = \beta_- \theta_{K-1}$, where $0 < \beta_- < 1$. Otherwise, we compute $g_K = s_{K-1} \cdot s_K$ and if $g_K < 0$ it means that a further step in the direction of s_K would have given a smaller value for $D(\lambda_K)$. In this case we call this iteration "zero" and make $\theta_K = \beta_0 \theta_{K-1}$, where $0 < \beta_- < \beta_0 < 1$. Finally, if $g_K \geq 0$, then this iteration is called "positive". In this case we make $\theta_K = \beta_+ \theta_{K-1}$, where $\beta_+ > 1$.

(e) Algorithm statement

We can summarize the proposed algorithm as follows:

Step 1: Initialization:

1.1) Set UB $= \infty$; LB $= -\infty$; $K = 1$

1.2) Set initial Lagrangean multiplier λ_K values;

Step 2: Solve Lagrangean dual Problem:

2.1) Solve each independent subproblem 5.17 to 5.20 ;

2.2) Combine subproblem solutions to get the lower bound $LB_K = \sum_\xi D^\xi(\lambda_K)$;

2.3) If $LB_K > LB$, then $LB = LB_K$. Also store solution for generating cuts later.

Step 3: Generate first-stage solution and derive UB

3.1) Apply the proposed heuristic for generating a first-stage solution x_K;

3.2) Obtain $v(x_K)$ evaluating x_K in 5.5-5.9. If x_K is not feasible, add a integer cut of type 5.27 and return to *Step 2*

3.3) If $v(x_K) < UB$, then $UB = v(x)$;

Step 4: If $UB - LB < \epsilon$ or any other criteria, such as time elapsed or number of iterations are met, stop and return x_K and UB. Otherwise, set $K = K + 1$ and proceed.

Step 5 Lagrange multiplier update:

5.1) Adjust the step length β;

5.2) Solve 5.31 to 5.33 and retrieve the new set of Lagrangean multipliers λ_K. Return to *Step 2*;

5.3 Risk Management

In stochastic programming, where uncertain data are modeled as stochastic processes, the objective function value is a random variable that can be characterized by a probability distribution. Bearing in mind that the objective function is given as a combination of the total first-stage cost and the expected cost of the recourse actions, we actually optimize a function characterizing the distribution of this random variable (i.e., its expected value).

Nevertheless, despite the numerous advantages of representing a random variable by its expected value, it is important to highlight that this is a risk-neutral approach. In other words, it means that the remaining parameters characterizing the distribution associated with random variables are not taken into consideration by the optimization itself. This might lead to cases where, even though the expected cost is optimal, the distribution of the objective function cost might present a significant probability of incurring in higher cost levels.

To control the risk of expected cost distributions with non-desirable properties, such as a high probability of incurring in high costs, risk management constitutes an important issue when formulating stochastic programming models. The most common way of controlling risk is to include in the model formulation a term that represents the measure of the risk associated with a profit distribution. Popular risk measures include variance (Markowitz, 1952), shortfall probability (Browne, 1999), expected shortage (Acerbi et al., 2001), Value-at-Risk (VaR) (Jorion, 2000), and Conditional Value-at-Risk (CVaR) (Rockafellar and Uryasev, 2000).

We chose to use the expected shortage as a risk measure. The reasoning behind this choice is related to the inherent interpretation and computational complexity of each risk measure. On one hand, using the variance as a risk measure is not completely adequate in the present context since it penalizes in the same way scenarios with higher and lower costs, since it only is concerned with deviations from the expected value. On the other hand, risk measures such as shortfall probability, VaR, and CVaR has the drawback of increasing the problem complexity, either by increasing the number of binary variables (in the case of shortfall probability and VaR), or by destroying the decomposable structure of the Lagrangian dual (in the case of VaR and CVaR) [2].

The expected shortage can be defined as the expectation of the cost in the scenarios where the cost is higher than a pre-specified target η. The expected

[2]One might argue that decomposition can be restored by creating additional copy variables. Nevertheless, it would increase the size of the multipliers set, and thus, the complexity of the problem.

shortage is given by:

$$ES(\eta, x) = \frac{1}{SP(\eta, x)} \sum_{\xi} P^{\xi} \max\{cx + qy_k^{\xi} - \eta, 0\} \tag{5.34}$$

where $SP(\eta, x) = \sum_{\xi | cx + qy_k^{\xi} > \eta} P^{\xi}$. One can observe that expression $\max\{cx + qy_k^{\xi} - \eta, 0\}$ is different from zero in all scenarios in which the cost is greater than η. In order to properly calculate the expected value of the cost over such scenarios, it is necessary not to take into account the probability of those scenarios with a cost smaller than η. For this reason, the expectation expression above must be divided by the sum of the probabilities of all scenarios with a cost larger than η. The sum of these probabilities is called the shortfall probability.

The expected shortage can be incorporated into the risk-neutral formulation given in section 5.3 as follows

$$\min_{x,y,r} cx + \sum_{\xi} P^{\xi} qy^{\xi} + \sum_{\xi} P^{\xi} r^{\xi} \tag{5.35}$$

s.t.:

$$Ax \leq b \tag{5.36}$$

$$Tx + Wy^{\xi} \leq h^{\xi} \quad \forall \xi \in \Omega \tag{5.37}$$

$$y^{\xi} \in Y \tag{5.38}$$

$$cx + qy^{\xi} - \eta \leq r^{\xi} \quad \forall \xi \in \Omega \tag{5.39}$$

$$r^{\xi} \geq 0 \quad \forall \xi \in \Omega \tag{5.40}$$

where $r^{\xi}, \forall \xi \in \Omega$ is a continuous and non-negative variable that is equal to $\max\{cx + qy_k^{\xi} - \eta, 0\}$. Once the problem above is solved and the optimal values for variables $r^{\xi}, \forall \xi \in \Omega$ are available, we calculate the expected shortage $ES(\eta, x)$ as given by

$$ES(\eta, x) = \frac{1}{\sum_{\xi | r^{\xi} \geq 0} P^{\xi}} \sum_{\xi} P^{\xi} r^{\xi} \tag{5.41}$$

Note that in this case the block-diagonal structure is preserved, which allow us to use the same ideas presented in section 5.2 as solution strategy. Moreover, only one constraint and one continuous variable is added to each subproblem, which means that there is not significant increase in the complexity of the subproblems. Regarding the selection of target η, there is an inherent trade-off between the level of risk accepted and how much optimality might be compromised in order to reach such desired risk protection. In order to

elucidate this trade-off one might successively solve the problem for different target values and then come up with an Pareto optimal frontier considering the objective function and different target levels.

5.4 Numerical results

In this section, we present the numerical results for two different examples where the proposed framework is applied. All the problems were modeled using AIMMS 3.11 and solved with CPLEX 12.1 (including the ones within the decomposition approach) on Intel i7 1.8GHz CPU with 4GB RAM.

(a) Example 1

The first example consists of a small instance, where a simplified version of the supply chain investment planning problem is considered. The structure of the network considered can be found illustrated in Figure 5.2. It consists of five time periods of one year each, one product, three production sites, and 9 demand points, from which 5 are primary bases with marine access (second row) and four are secondary bases (third row).

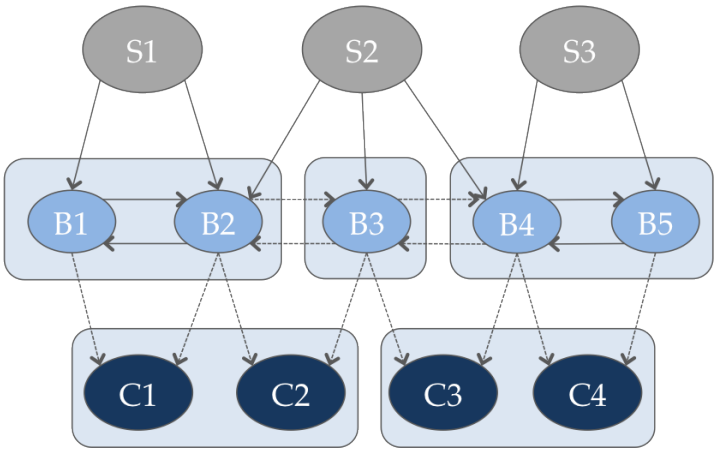

Figure 5.2: Network structure of example 1

In this example, only the primary bases can rely on the use of emergency floating tankage if necessary. We consider twelve options for the network design (arcs represented with dotted lines in Figure 5.2) and that only primary bases are able to have investments in tankage expansion. The bases are organized in five different regions (represented by gray rectangles) based on their geographical proximity. Sixteen demand scenarios are considered, where it is assumed that the demand for each region can either grow or decrease 5% per year.

The equivalent deterministic of the two-stage stochastic problem for this example has 3859 constraints, 2083 continuous variables, 480 binary variables, and 13045 non-zeros. Tables 5.2 and 5.3 give the optimal first-stage decisions concerning the selection and timing of both network design and capacity expansion decision.

Bases	Periods	
	2	4
B1		X
B2	X	
B3	X	
B4	X	

Table 5.2: Example 1 Capacity expansion decisions

Arcs	Periods		
	1	2	5
B1-C1	X		
B2-C1		X	
B2-C2	X		
B3-C2	X		
B3-C3		X	
B4-C3	X		
B4-C4			X
B5-C4	X		

Table 5.3: Example 1 Network design decisions

The optimal expected cost is $ 8718.3 million. If we optimize the problem considering the average values of the random variables, the optimal solution of this case would be suboptimal for the complete stochastic problem with a cost of $ 10134.4. The Value of the Stochastic Solution (VSS) (Birge and Louveaux, 1997), which is given by the absolute difference between the optimal value of the stochastic program and the objective value calculated using the solution of the deterministic problem considering average levels for the stochastic variables, is $ 1416.1 million. The VSS can be seem as a measure of the savings in cost due to the consideration of uncertainty, indicating in this case savings of about 16%. The large savings in this case are related to the high cost of acquiring emergency floating tankage and with the fact that the project selection when the demand is considered to be its average comprises fewer projects, making the floating tankage acquisition more often.

	Sequential Formulation		Asymmetric Formulation	
	Subgradient	Proposed	Subgradient	Proposed
Total Time(s)	1800.0	1037.4	394.4	118.0
Iterations	662	316	136	42
UB($ million)	8718.3	8731.7	8722.8	8726.6
LB($ million)	8487.2	8586.2	8584.9	8555.5
% gap	2.65	1.67	1.58	1.96

Table 5.4: Summary of CPU times(s)

Table 5.4 gives a summary of the number of iterations and the computational time required to reach convergence. We compare 4 different cases where we combine two different formulations for the NAC constraints (namely asymmetric formulation as given in 3.20, and sequential formulation as given in 3.23) and two different algorithms (namely the traditional subgradient algorithm, and our proposed hybrid approach). In this example we used a 2% optimality gap and 1800s as stopping criteria. We consider in this example $\beta_- = 0.8, \beta_0 = 0.99$, and $\beta_+ = 1.2$.

As can be seen in Table 5.4, the asymmetric formulation performs better in terms of computational time when compared to the sequential formulation, independently of which solution technique is used (394.4s versus 1800.0s for the subgradient algorithm and 118.0s versus 1037.4s for our proposed hybrid approach). In addition to that, our proposed algorithm performs better than the traditional subgradient algorithm no matter which formulation is used (1037.4s versus 1800.0s for the sequential formulation and 118.0s versus 394.4s for the asymmetric formulation). The differences in the bounds obtained are due to the use of a 2% gap as one of the stop criteria, which invalidates any comparisons between the bounds obtained with different combinations of formulation and solution techniques.

The best combination observed is the use of the asymmetric formulation combined with the proposed hybrid approach (118.0s). We believe that the faster performance presented by the asymmetric formulation is related to the fact that in this formulation the subproblem for $\xi = 1$ combines the multipliers from all problems, while in the other formulation the multipliers are considered in a somewhat myopic fashion since only two different multipliers are combined in each subproblem. It seems that for this particular case, the penalties provided by the Lagrangean multipliers tend to be more effective in the case of the asymmetric formulation since there are fewer iterations, thus improving convergence. Moreover, when we compare the two different

algorithms, one must bear in mind that the subgradients involving only binary variables provide a very limited set of possibilities (recall that, for each dimension, it can only assume the values of 1, -1 or 0). Provided this fact, and that the subgradient is an estimation to the true ascent direction, the consequences of having poor estimations can be very harmful. Our proposed approach deals with that issue by using the magnitude of the step as a reference combined with the outer-approximation of the Lagrangean dual function to decide the step size update based on an optimization framework. As the results suggest, this strategy tends to provide better decisions in terms of the Lagrange multiplier updates. Figures 5.3, 5.4, 5.5, and 5.6 gives the plots of the convergence profile of each combination, comparing then between the two algorithms and the two formulations. In these pictures, "Asymmetric" represents the use of the asymmetric formulation, "Sequential" represents the use of the sequential formulation, "Proposed" represents our proposed hybrid algorithm, and "Subgradient" the traditional subgradient algorithm.

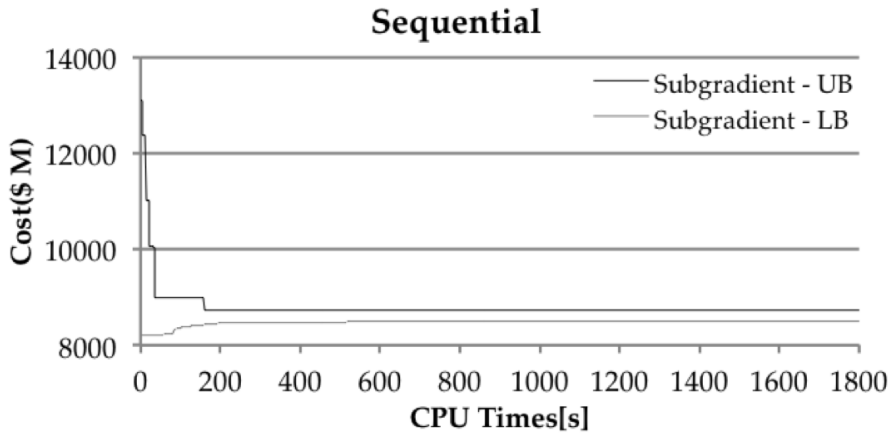

Figure 5.3: Convergence Profile: Subgradient algorithm and sequential formulation

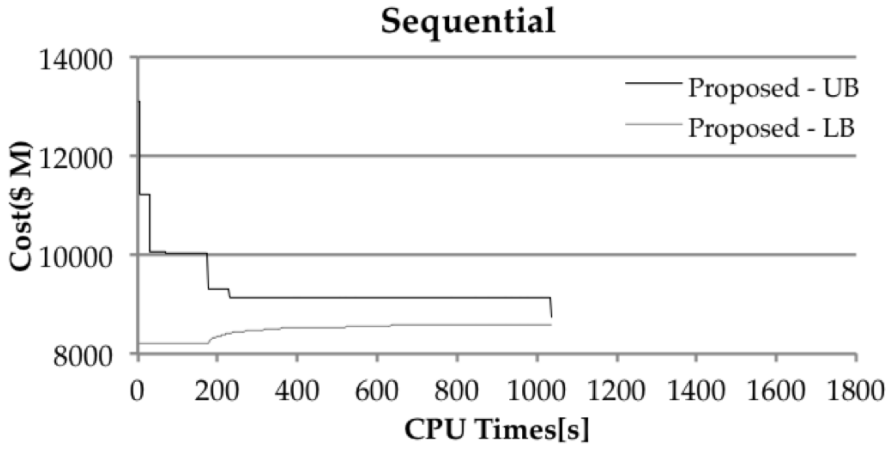

Figure 5.4: Convergence Profile: Proposed algorithm and sequential formulation

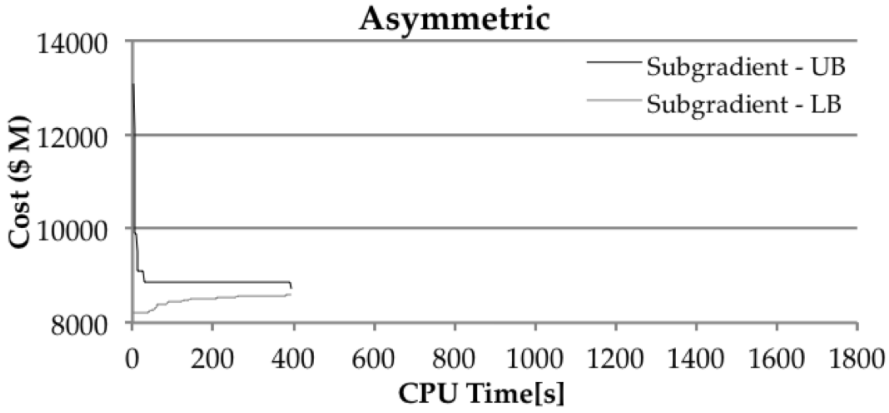

Figure 5.5: Convergence Profile: Subgradient algorithm and asymmetric formulation

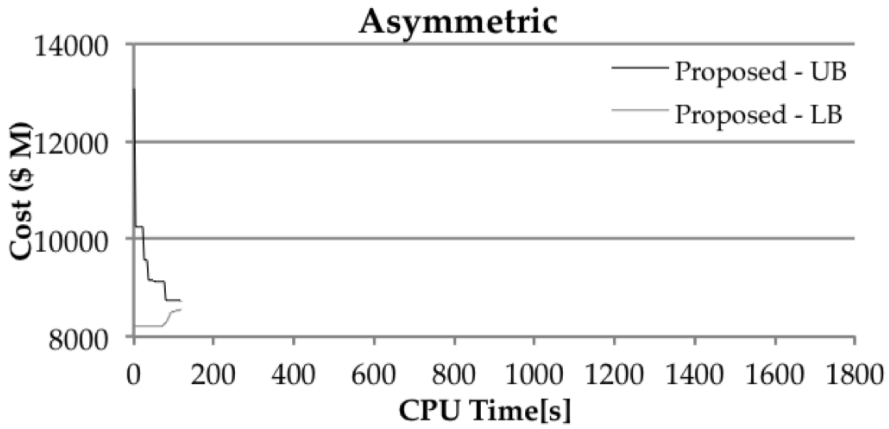

Figure 5.6: Convergence Profile: Proposed algorithm and different algorithms

(b) Example 2

The second example represents the realistic supply chain investment planning problem under demand uncertainty described in section 3.3. In this case we consider four different products to be distributed between 16 locations (13 bases, 1 refinery and 2 international suppliers). A total of 28 projects for tankage expansion and 3 projects for network design were considered under a planning horizon of 5 years, divided quarterly into 20 periods. All computational experiments in this instance were solved considering 7200s (2h) and 2% optimality gap as stopping criteria.

In order to reduce the number of binary variables of the model, we use a multi-scale definition for the time horizon regarding investment (first-stage) decisions and planning (second-stage) decisions. In this sense, we aggregate the investment decisions such that they are considered to be available at the beginning of each semester (i.e., considering a semiannually divided horizon), while the planning decisions are taken considering the original quarterly divided horizon. Such an approach yields an upper bounding approximation to the original problem. However, in our early experimentations this was shown to be an acceptably tight approximation, with differences in the objective function smaller than 0.5% in our case. Figure 5.7 illustrates the different scales used for investment decisions and planning decisions.

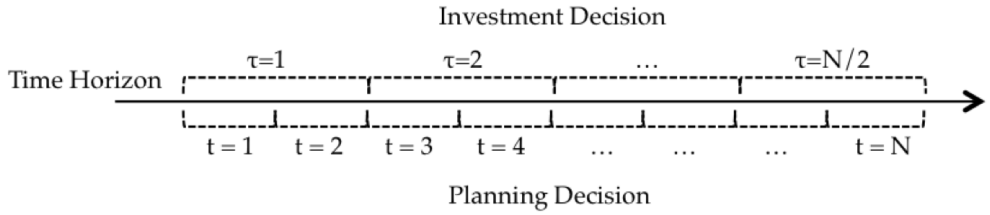

Figure 5.7: Multi-scale approximation representation

Scenarios	Constraints	Binary Var.	Continuous Var.	Non-zeros
25	113822	1890	88283	478780
50	226822	3390	175783	951705
100	452822	6390	350783	1897555
200	904822	12390	700783	3789255

Table 5.5: Deterministic Equivalent Sizes

Scenarios	Sequential Formulation		Asymmetric Formulation		Full Space
	Subgradient	Proposed	Subgradient	Proposed	
25	1203.2	676.5	622.0	482.7	675.5
50	2714.3	1260.9	908.5	507.3	OoT
100	OoT	4625.5	4018.7	1061.5	OoM
200	OoT	OoT	OoT	6151.8	OoM

Table 5.6: Summary of CPU times(s)

Table 5.5 shows the deterministic equivalent size of the problem considered for instances of different sample sizes, while Table 5.6 gives computational results in terms of solution times. The instances were solved using the traditional subgradient and the proposed hybrid algorithms considering the two different formulation for NAC. We also compare this solution times with directly solving the full-space deterministic equivalent problem (column "Full-space").

As can be seem in Table 5.6, the hybrid approach combined with the asymmetric formulation of the NAC outperforms the other possible combinations in terms of computational times. Indeed, for the 200 scenario instance, this is the only algorithm that is able to reach a 2% optimality gap solution before the time limit of 7,200s. The entries "OoM" and "OoT" stands for Out-of-Memory and Out-of-Time, respectively. The entry Out-of-Memory means that the available RAM was not sufficient to deal with the deterministic equivalent in these cases. The entry Out-of-Time means that the time limit was reached by the algorithm before the optimality gap limit. We also consider in this example $\beta_- = 0.8$, $\beta_0 = 0.99$, and $\beta_+ = 1.2$, which are the same values used in Example 1.

We solve this case study with a sample size of 200 scenarios. Figure 5.8 shows the results in terms of the cost distribution. The objective function of the stochastic problem is \$64283.8 million. The solution of the deterministic problem considering the average demand levels for the same 200 scenarios is the suboptimal solution value of \$68236.4 million. The VSS for this scenario sample is thus \$3952.6 million, which represents savings of about 5.8%.

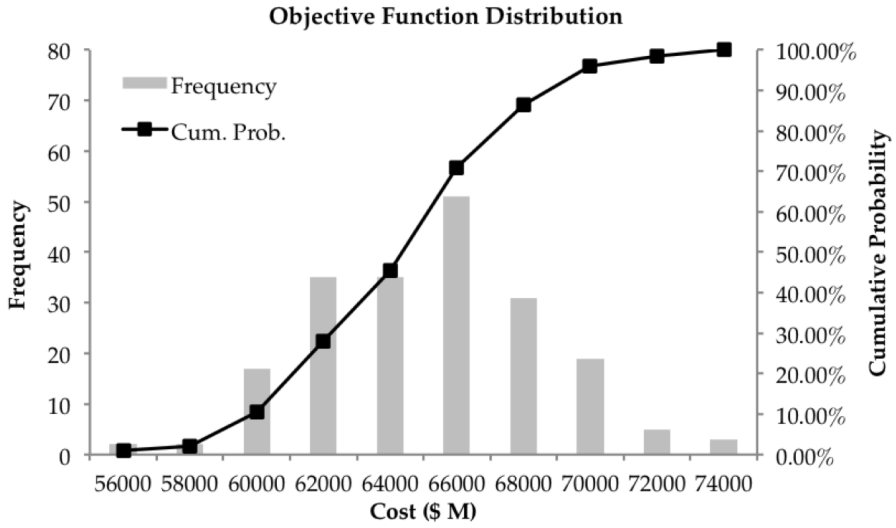

Figure 5.8: Cost distribution for 200 scenarios

In Figure 5.8, the distribution of costs shows that there is a non-negligible probability of incurring in high costs due to the dispersion that the probability distribution presents towards its right-hand side. In order to control the risk of high costs we applied the risk management model presented in section 5.3 for minimizing the expected shortfall. We set the target to $ 68000 million for the following calculations based on the assumption that we would like to avoid possible deviations that exceed the expected cost in more than 5% . Figure 5.9 presents the new distribution of costs for the case when risk is incorporated in the model.

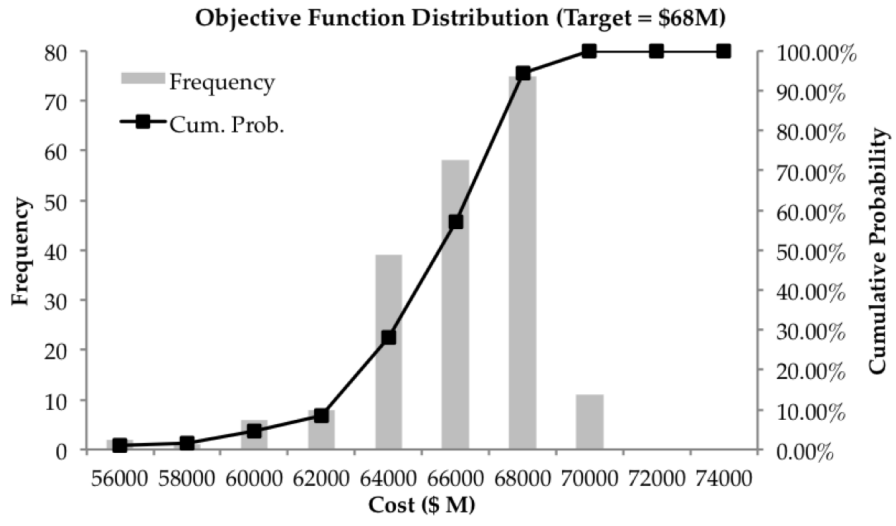

Figure 5.9: Cost distribution for 200 scenarios after risk management

As can be seen in Figure 5.9 the risk management affects the cost

distribution by reducing the the expected shortfall (i.e., expected cost over the target), as well as the probability of incurring in higher costs. In this case, the objective function value increases to \$65588.7 million, or 1.3%. The expected shortfall cost is reduced from \$9433 million to \$3850.0 million (over the target), which represents a reduction of 59.3%. Moreover the probability of shortfall is dropped from 13.5% to 5.5%. Figure 5.10 shows the comparison between the objective function value distribution before and after the risk management technique is applied.

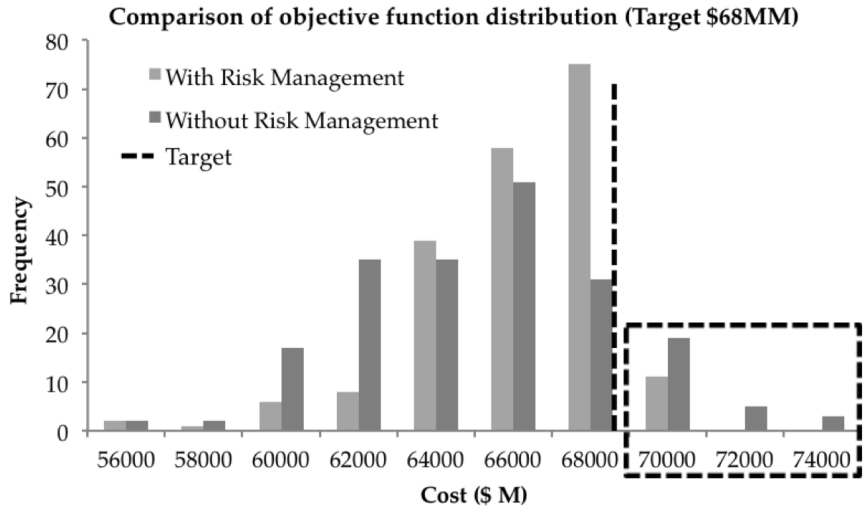

Figure 5.10: Cost distribution comparison

5.5 Conclusions

In this chapter, we presented a two-stage mixed-integer linear stochastic programming approach for the strategic planning of a multi-product, multi-period supply chain investment planning problem under demand uncertainty. We developed a comprehensive framework for solving the problem based on Lagrangean decomposition, exploiting its scenario decomposable structure. In this context, the use of decomposition presented itself as being imperative, due to the large size of the full-space problem. We also presented a novel hybrid algorithmic framework for updating the Lagrangean multiplier set, based on the combination of cutting-plane, subgradient, and trust-region strategies. Numerical results suggests that significant savings in computational times can be achieved by using the proposed strategy.

We also explicitly consider a risk management tool as a mean to reduce the chances of incurring in high costs. We chose the expected shortfall as a risk measures, since it presented itself as being more suitable for the presented

context when compared to several other risk measures. The results suggest that this risk measure can efficiently reduce the high cost risks without increasing the complexity of the problem.