Working Paper

On Augmented Lagrangian Decomposition Methods For Multistage Stochastic Programs

Andrzej Ruszczyński

WP-94-05
February 1994

On Augmented Lagrangian Decomposition Methods For Multistage Stochastic Programs

Andrzej Ruszczyński

WP-94-05
February 1994

IIASA International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 715210 □ Telex: 079 137 iiasa a □ Telefax: +43 2236 71313

Abstract

A general decomposition framework for large convex optimization problems based on augmented Lagrangians is described. The approach is then applied to multistage stochastic programming problems in two different ways: by decomposing the problem into scenarios or decomposing it into nodes corresponding to stages. In both cases the method has favorable convergence properties and a structure which makes it convenient for parallel computing environments.

Keywords: Stochastic Programming, Decomposition, Augmented Lagrangian, Jacobi Method, Parallel Computation.

1 Introduction

Multistage stochastic optimization problems belong to the most difficult problems of mathematical programming. Their size grows very quickly with the number of stages and with the number of events (scenarios) incorporated into the model. Although problems of this type occur frequently in applications (like, e.g., investment planning problems, control of water systems or energy systems), it was a generally held opinion that they are too difficult to be solved in their full formulation. However, recent advances in the theory of stochastic programming and in the computing technology make it possible to develop new methods for solving multistage stochastic programs of remarkable sizes. The purpose of this paper is to describe such an approach which has already proved successful in some applications and appears to have a potential to solve a broad class of problems.

After a brief description of the class of problems under consideration in section 2, we present the general decomposition framework in section 3. The method is applicable to general convex problems with many subproblems and many linking constraints. Our approach is based on augmented Lagrangians and has its roots in the pioneering work [21]. Following [20] we show that properties of the method heavily depend on sparsity of the linking constraints. Next, in section 4, we apply the general framework to multistage stochastic programming problems formulated in a scenario form. The subproblems correspond to scenarios and nonanticipativity constraints are treated as linking constraints. In section 5 we apply the general framework to multistage problems decomposed into particular stages of the decision-making process. Then the equations of dynamics, which relate to the variables from different stages, are treated as linking constraints in the decomposition approach. In both cases we show that the augmented Lagrangian decomposition method has favorable properties with a broad range of parameters guaranteeing convergence and good rate of convergence estimates.

2 Multistage stochastic programming models

In a multistage optimization problem decisions are to be made in stages $t = 1, 2, \ldots, T$ and the decision vector is a collection of subvectors corresponding to successive stages,

$$x = (x(1), x(2), \ldots, x(T)).$$

Decisions in successive time stages have to satisfy two groups of relations. The first group describes the set of feasible actions for each t:

$$x(t) \in X(t), \quad t = 1, 2, \ldots, T, \tag{2.1}$$

where $X(t) \subseteq R^{m_x}$, $t = 1, 2, \ldots, T$. The second group describes the dynamics of the system and relates decisions from different time stages. In the simplest linear model they may read:

$$D(t)x(t-1) + H(t)x(t) = b(t), \qquad t = 1, \ldots, T. \tag{2.2}$$

Here $D(t)$ and $H(t)$, $t = 1, \ldots, T$ are sequences of $m_b \times m_x$ matrices, $b(t)$, $t = 1, \ldots, T$, is a sequence of vectors in R^{m_b} and $x(0)$ is fixed. Obviously, the dimensions need not be the same for different t; we just use one m_x and one m_b for simplicity.

Finally, there is a cost function $c : R^{m_x T} \to R$

$$c(x) = c_1(x(1)) + c_2(x(2)) + \ldots + c_T(x(T)) \tag{2.3}$$

that needs to be minimized.

In stochastic programming, the data $X(t)$, $D(t)$, $H(t)$ and $b(t)$ are random objects defined on some underlying probability space $(\Omega, \mathcal{B}, P)$. We shall call each sequence

$$s_\omega(t) = (X_\omega(t), D_\omega(t), H_\omega(t), b_\omega(t)), \qquad t = 1, \ldots, T,$$

corresponding to an elementary event $\omega \in \Omega$ a *scenario*.

Realizations of the random data associated with time stage t become known at t, so it is reasonable to make the decision $x(t)$ dependent on the information that is already available. Consequently, x is a random vector itself, and (2.1) and (2.2) are relations between random variables that are assumed to hold with probability 1.

However, x cannot be an arbitrary random vector; the dependence of $x(t)$ on ω may result only from the observations carried out up to time t. This is called *non-anticipativity*: for each t decisions $x_\omega(t)$ must be equal for all scenarios ω that have common past and present. Formally, this can be stated as the condition of measurability of $x(t)$ with respect to the σ-subfield $\mathcal{B}(t) \subseteq \mathcal{B}$, generated by $\{s(1), s(2), \ldots, s(t)\}$.

Moreover, the cost (2.3) is a random variable itself and we need to replace it by a scalar-valued function. It is a common practice to use an expected value of the cost as the objective, although other choices are possible, too, as, e.g., mean-variance models. Since we are going to work with a general convex c, using its expectation does not seem very restrictive, because we still retain the flexibility of nonlinear utility functions.

The problem can be now stated as follows:

$$\min \; E\left[c_1(x_\omega(1)) + c_2(x_\omega(2)) + \ldots + c_T(x_\omega(T))\right] \tag{2.4}$$

subject to the constraints

$$D_\omega(t)x_\omega(t-1) + H_\omega(t)x_\omega(t) = b_\omega(t), \qquad t = 1, \ldots, T, \quad \omega \in \Omega, \tag{2.5}$$

$$x_\omega(t) \in X_\omega(t), \qquad t = 1, \ldots, T, \quad \omega \in \Omega, \tag{2.6}$$

with $x(0) = x_0$ fixed. The non-anticipativity constraint can be formulated as follows: for all $\omega, \zeta \in \Omega$ and any $t \in \{1, \ldots, T\}$

$$x_\omega(t) = x_\zeta(t) \;\; \textit{if} \;\; s_\omega(\tau) = s_\zeta(\tau) \;\; \textit{for} \;\; \tau = 1, \ldots, t. \tag{2.7}$$

In other words, decisions corresponding to scenarios which are indistinguishable up to time t should be equal.

We shall assume throughout this paper that the sets $X_\omega(t)$, $t = 1, \ldots, T$, $\omega \in \Omega$ are convex and closed and the functions c_t, $t = 1, \ldots, T$, are convex, which makes (2.4)-(2.6) a convex optimization problem. But even then the problem is too difficult

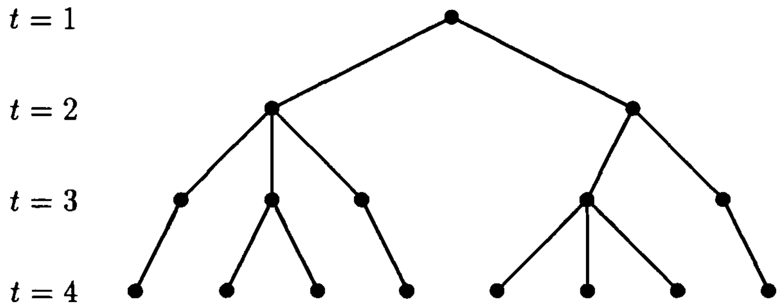

Figure 1: Scenario tree.

to be successfully solved for general distributions of the random data. Therefore in applications we confine ourselves with some approximate distributions comprising only finitely many scenarios. In other words, we assume that Ω is a finite set:

$$\Omega = \{1, 2, \ldots, S\}.$$

Under such an assumption, with the set of scenarios $s_\omega(t)$, $t = 1, \ldots, T$, $\omega \in \Omega$, we can associate a tree $\mathcal{T} = \{\mathcal{N}, \mathcal{A}\}$, where $\mathcal{N}$ is a set of nodes and $\mathcal{A}$ is a set of arcs of $\mathcal{T}$. The set of nodes $\mathcal{N}$ is divided into subsets (levels) $\mathcal{N}_t$, $t = 1, \ldots, T$; the nodes $n \in \mathcal{N}_t$ at level t correspond to different subscenarios $\{s^n(1), \ldots, s^n(t)\}$. At level 1 there is only one node $n = 1$ (the data for stage 1 are known). At level 2 there are as many nodes as different realizations of $s(2)$ that can occur; at level 3 the nodes correspond to different pairs $\{s(2), s(3)\}$, etc. The number of nodes at level T is equal to the number of scenarios S. The arcs join nodes from neighboring levels in such a way that a node n at level t corresponding to subscenario $s^n = \{s^n(1), \ldots, s^n(t)\}$ is connected with all nodes m at level $t + 1$ whose subscenarios $s^m = \{s^m(1), \ldots, s^m(t + 1)\}$ equal s^n up to time t. An example of such a tree for an 8-scenario problem is shown in Fig. 1.

Problems with finitely many scenarios are more amenable for computer solutions, but many difficulties still remain.

First of all, one has to note the remarkable size of the problem. If the scenarios introduced to the model are to reflect uncertainties that occur at successive time stages, then the number S of scenarios grows exponentially with the increase of the time horizon T. Even for relatively small T the dimension of (2.4)-(2.7) may be so large that the whole problem will become intractable by direct solvers.

However, (2.4)-(2.7) has a very special structure which creates a number of possibilities for developing special solution methods.

Existing computational methods for multistage stochastic programming problems can be divided into two main groups. First, there are versions of general-purpose algorithms in which special features of stochastic problems are used to improve data structures and solution strategies [10, 8]. Secondly, we have a number of special *decomposition methods* which exploit the structure of the problem to split it into manageable pieces and coordinate their solution [23]. One can distinguish two classes: *primal* decomposition methods that work with subproblems which are assigned to time stages

[4, 7, 17, 18, 22] and *dual* methods, in which subproblems correspond to scenarios [11, 19, 16].

In this paper we shall use the general theory of augmented Lagrangian decomposition of [20] to develop and analyze two new decomposition methods for multistage stochastic programs. The first one is a dual method proposed for linear multistage stochastic programs in [11] and further developed in [12]. We shall show how to deal with convex objectives and we shall present some results on its convergence and rate of convergence. The second approach is a primal method based on the concept of node decomposition. Again, we shall use the general theory developed in [20] to obtain convergence and rate of convergence results for the method. Alternative decomposition approaches based on augmented Lagrangians are discussed in [3, 6, 19].

3 General decomposition framework

The purpose of this section is to briefly describe the general augmented Lagrangian decomposition method for partially separable convex problems. The approach will then be used in later sections to develop specific methods for multistage stochastic problems.

Let $X_1, X_2, \dots, X_L$ be non-empty closed convex subsets of $R^{n_1}, R^{n_2}, \dots, R^{n_L}$, respectively, and let $f_i : R^{n_i} \to R$, $i = 1, 2, \dots, L$ be convex functions. Next, let A_i be matrices of dimension $m \times n_i$, $i = 1, 2, \dots, L$ and let $b \in R^m$. We consider the convex programming problem:

$$\min \sum_{i=1}^{L} f_i(x_i) \tag{3.1}$$

$$\sum_{i=1}^{L} A_i x_i = b, \tag{3.2}$$

$$x_i \in X_i, \ i = 1, 2, \dots, L. \tag{3.3}$$

The augmented Lagrangian for this problem is defined as:

$$\Lambda(x, \pi) = \sum_{i=1}^{L} f_i(x_i) + \langle \pi, b - \sum_{i=1}^{L} A_i x_i \rangle + \frac{1}{2}\rho \left\| b - \sum_{i=1}^{L} A_i x_i \right\|^2, \tag{3.4}$$

with some penalty parameter $\rho > 0$. As usual, we define the dual functional

$$g(\pi) = \inf_{x \in X} \Lambda(x, \pi)$$

with $X = X_1 \times X_2 \times \cdots \times X_L$, and the dual problem:

$$\max_{\pi \in R^m} g(\pi). \tag{3.5}$$

There are many theoretical and computational advantages of the augmented Lagrangian approach over the ordinary Lagrangian (with $\rho = 0$). For the duality to hold, it is sufficient that the following condition is satisfied.

Constraint Qualification Condition. At least one of the following conditions holds:

(i) at some feasible point x^0

$$\text{ri}\,\{d:\ \exists\alpha > 0 \textit{ such that } x^0 + \alpha d \in X\} \cap \{d:\ Ad = 0\} \neq \emptyset;$$

(ii) X is a polyhedral set.

The fundamental duality result can be formulated as follows (see, e.g., [13, 14]).

Proposition 3.1 *Assume that (3.1)-(3.3) has an optimal solution and the Constraint Qualification Condition is satisfied. Then (3.5) has an optimal solution and*

(a) for every optimal solution $\hat{x}$ of (3.1)-(3.3) and every optimal solution $\hat{\pi}$ of (3.5)

$$f(\hat{x}) = g(\hat{\pi});$$

(b) for every optimal solution $\hat{\pi}$ of (3.5) a point $\hat{x}$ is a solution of (3.1)-(3.3) if and only if

$$\Lambda(\hat{x}, \hat{\pi}) = \min_{x \in X} \Lambda(x, \hat{\pi}). \tag{3.6}$$

An important advantage over the usual Lagrangian duality is that (3.6) is sufficient for primal recovery when the dual solution is known. The major computational advantage is the possibility of solving the dual problem by the following algorithm.

Method of Multipliers

Step 1. For fixed multipliers π^k find a solution x^k of the problem

$$\min_{x \in X} \Lambda(x, \pi^k). \tag{3.7}$$

Step 2. If $Ax^k = b$ then stop (optimal solution found); otherwise set

$$\pi^{k+1} = \pi^k + \rho(b - Ax^k), \tag{3.8}$$

increase k by 1 and go to Step 1.

The following two propositions summarize the fundamental properties of the method of multipliers (see [2, 15]).

Proposition 3.2 *Let the Constraint Qualification Condition be satisfied. Then the sequence $\{\pi^k\}$ generated by the method of multipliers is convergent to a solution $\hat{\pi}$ of (3.5).*

Proposition 3.3 *Assume that f_i, $i = 1, 2, \ldots, L$ are convex polyhedral functions, X_i, $i = 1, 2, \ldots, L$ are convex polyhedral sets and (3.1)-(3.3) has a solution. Then the method of multipliers is convergent in finitely many iterations.*

The simplicity of iteration (3.8) makes the method of multipliers especially attractive for problems with many linking constraints (3.2), where column generation techniques stemming from [5] fail. However, a serious disadvantage is that (3.4) is not separable, so problem (3.7) cannot be split into independent subproblems for x_i, $i = 1, 2, \ldots, L$.

To overcome this difficulty we introduce for $i = 1, 2, \ldots, L$ the functions

$$\Lambda_i(x_i, \tilde{x}, \pi) = f_i(x_i) - \langle A_i^T \pi, x_i \rangle + \frac{1}{2}\rho \left\| b - A_i x_i - \sum_{j \neq i} A_j \tilde{x}_j \right\|^2, \tag{3.9}$$

where $\tilde{x} \in R^n$ is an additional parameter, $n = \sum_{i=1}^{L} n_i$. The main idea of our approach is to replace problem (3.7) with L problems

$$\min_{x_i \in X_i} \Lambda_i(x_i, \tilde{x}, \pi^k), \quad i = 1, 2, \ldots, L, \tag{3.10}$$

and to iteratively update the parameter $\tilde{x}$ by making steps towards the solutions of (3.10). It is not difficult to see that (3.10) is equivalent to minimizing (3.4) with respect to x_i while keeping x_j, $j \neq i$, frozen at $\tilde{x}_j$. However, we are not going to use (3.10) in a sequential fashion, but we shall rather solve it for each i in parallel and then update $\tilde{x}$. This approach is called a *nonlinear Jacobi algorithm.*

We are now ready to describe the method in detail. It should be noted that it is a sub-algorithm for carrying out Step 1 of the method of multipliers in a decomposed fashion. In what follows $\tau \in (0, 1)$ is a parameter of the method.

Jacobi Method

Step 0. Set $\tilde{x}^{k,0} = x^{k-1}$ and $r = 0$.

Step 1. For $i = 1, 2, \ldots, L$ solve (3.10) getting points $x_i^{k,r}$.

Step 2. If $A_i x_i^{k,r} = A_i \tilde{x}_i^{k,r}$, $i = 1, 2, \ldots, L$, then stop; otherwise set for $i = 1, 2, \ldots, L$

$$\tilde{x}_i^{k,r+1} = \tilde{x}_i^{k,r} + \tau(x_i^{k,r} - \tilde{x}_i^{k,r}), \tag{3.11}$$

increase r by 1 and go to Step 1.

Let us now pass to conditions under which the Jacobi method generates sequences $\{x^{k,r}\}_{r=0}^{\infty}$ and $\{\tilde{x}^{k,r}\}_{r=0}^{\infty}$ whose accumulation points are solutions of (3.7). They involve the measure of sparsity of the linking constraints (3.2) defined as follows. For every matrix A_i, let A_{ji} denote its jth row and let

$$V(i, j) = \{k \neq j : \ A_{ji}^T A_{jk} \neq 0\};$$

i.e., $V(i, j)$ is the set of other blocks linked with block i via row j. We can now define the *maximum number of neighbors* as

$$N = \max_{i,j} |V(i, j)|. \tag{3.12}$$

In other words, N is the maximum number of blocks linked by any single constraint, decremented by one. The theorems to follow show that convergence properties of the Jacobi method depend heavily on the number of neighbors N.

Theorem 3.1 *Assume that the assumptions of Proposition 3.1 are satisfied and the sets $X_i, i = 1, 2, \ldots, L$ are bounded. If in the Jacobi method the under-relaxation coefficient satisfies the inequalities*

$$0 < \tau < \frac{1}{N}, \tag{3.13}$$

where N is given by (3.12), then:

(a) for all $i = 1, 2, \ldots, L$ $\lim_{r\to\infty} A_i(x_i^{k,r} - \tilde{x}_i^{k,r}) = 0$;

(b) each accumulation point of the sequence $\{x^{k,r}\}_{r=0}^{\infty}$ is a solution of (3.7).

To estimate the speed of convergence, we need the following assumption on the growth rate of the augmented Lagrangian function ($\hat{X}(\pi)$ denotes the set of solutions of (3.7)).

Quadratic Growth Condition. There exist $\gamma > 0$ and $\delta > 0$ such that for every $x \in X$ with $\mathrm{dist}(x, \hat{X}(\pi)) < \delta$ we have

$$\Lambda(x, \pi) - \hat{\Lambda}(\pi) \geq \gamma[\mathrm{dist}(x, \hat{X}(\pi))]^2.$$

It is clear that this condition is satisfied by linear and quadratic problems (3.1)-(3.3).

We can now formulate our main result on the speed of convergence.

Theorem 3.2 *Let the assumptions of Theorem 1 and the Quadratic Growth Condition be satisfied. Then, for all $r = 0, 1, 2, \ldots$ the following inequality holds*

$$\Lambda(\tilde{x}^{k,r+1}, \pi^k) - \hat{\Lambda}(\pi^k) \leq q\left[\Lambda(\tilde{x}^{k,r}, \pi^k) - \hat{\Lambda}(\pi^k)\right] \tag{3.14}$$

with

$$q = 1 - \frac{\tau(1 - \tau N)}{2\rho\alpha^2 N^2 \gamma^{-1} + 1}, \tag{3.15}$$

and

$$\alpha = \max_{1 \leq i \leq L} \|A_i\|. \tag{3.16}$$

Theorems 1 and 2 have been proved in [20]. We can also find there further refinements of these results for the case when the subproblems (3.10) are not solved till optimality, but with dynamically determined stopping criteria. In [9] the general approach is specialized to linear programming with even broader stepsize range than (3.13) with tighter estimates of the speed of convergence.

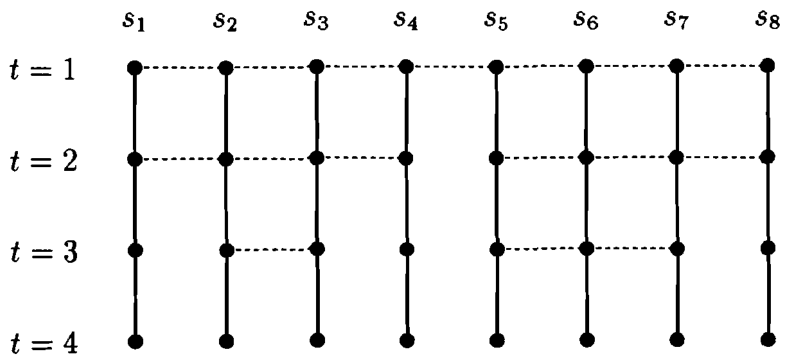

Figure 2: Sequences of decisions and nonanticipativity.

4 Scenario decomposition

We shall now apply the general framework of the previous section to problem (2.4)-(2.7) with the following assignments:

- subproblems correspond to scenarios $i = 1, \ldots, S$ with decision vectors
$$x_i = (x_i(1), x_i(2), \ldots, x_i(T));$$
- relations (2.5) and (2.6) are used to describe the sets X_i in (3.3):
$$x_i \in X_i = \{x_i : \; D_i(t)x_i(t-1) + H_i(t)x_i(t) = b_i(t), \\ x_i(t) \in X_i(t), \; t = 1, \ldots, T\}; \tag{4.1}$$
- non-anticipativity constraints are treated as linking constraints (3.2).

We shall now develop a formulation of non-anticipativity constraints which is convenient for our decomposition approach. Let us define the *last common stage* of scenarios ω and ξ by
$$t^{\max}(\xi, \omega) = \max\{t : \; s_\xi(\theta) = s_\omega(\theta), \; \theta = 1, \ldots, t\}.$$
We shall now order scenarios in Ω by assigning to them numbers $i = 1, \ldots, S$ in such a way that for every i scenario $i+1$ has the largest last common stage with i among all scenarios $j > i$:
$$t^{\max}(i, i+1) = \max\{t^{\max}(i,j) : \; j > i\}.$$
Scenarios in Fig. 2 for the tree of Fig. 1 are ordered in this way.

It is easy to observe that with such an ordering, the bundles of scenarios which are indistinguishable up to some time t form connected subsets of {1,...,S}. In Fig. 2, they are joined by horizontal dotted lines.

Next, for every scenario i and every time period t, we define the *sibling* of i at t as

$$\nu(i,t) = \begin{cases} i+1 & \text{if } t^{\max}(i, i+1) \geq t, \\ \min\{j : \; t^{\max}(i,j) \geq t\} & \text{otherwise.} \end{cases}$$

Time	Scenario							
stage	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	1
2	2	3	4	1	6	7	8	5
3	1	3	2	4	6	7	5	8
4	1	2	3	4	5	6	7	8

Table 1: Siblings of scenarios.

Let us note that a scenario may have different siblings at different time stages. For the example of Fig. 1 and Fig. 2, siblings of scenarios are shown in Table 1.

For every t, the mapping $\nu(i,t)$ defines a permutation of Ω, which maps bundles of indistinguishable scenarios onto themselves. It is easy to observe that $\nu(i,t) \neq i$, if the bundle of scenario i at stage t contains more than one member. The inverse permutation will be denoted by $\nu^{-1}(i,t)$.

Using the mapping $\nu(i,t)$ we can describe the non-anticipativity condition by the constraints:

$$x_i(t) = x_{\nu(i,t)}(t) \text{ for all } (i,t) \text{ such that } i \neq \nu(i,t). \tag{4.2}$$

There is still some redundancy in this set (we can remove one equation for each bundle), but we shall keep all equations (4.2) for convenience.

Thus, the whole problem has the following structure:

$$\min \sum_{i=1}^{S} p_i \sum_{t=1}^{T} c_t(x_i(t)) \tag{4.3}$$

subject to (4.1) and (4.2). This corresponds exactly to the general model (3.1)-(3.3).

The augmented Lagrangian function for (4.1)-(4.3) has the form

$$\begin{aligned} \Lambda(x,\pi) &= \sum_{i=1}^{S} p_i \sum_{t=1}^{T} c_t(x_i(t)) + \sum_{i=1}^{S} p_i \sum_{t=1}^{T-1} \langle \pi_i(t), x_i(t) - x_{\nu(i,t)}(t) \rangle \\ &+ \frac{1}{2}\rho \sum_{i=1}^{S} p_i \sum_{t=1}^{T-1} \| x_i(t) - x_{\nu(i,t)}(t) \|^2 . \end{aligned} \tag{4.4}$$

We introduce scaling factors p_i to the Lagrangian and penalty terms to simplify the resulting subproblems and multiplier iterations. Subproblems (3.10) take on the form:

$$\begin{aligned} \min_{x_i \in X_i} [\Lambda_i(x_i, \tilde{x}, \pi) &= \sum_{t=1}^{T} c_t(x_i(t)) + \sum_{t=1}^{T-1} \langle \pi_i(t) - \pi_{\nu^{-1}(i,t)}(t), x_i(t) \rangle + \\ & \frac{1}{2}\rho \sum_{t=1}^{T-1} \{ \| x_i(t) - \tilde{x}_{\nu(i,t)}(t) \|^2 + \| x_i(t) - \tilde{x}_{\nu^{-1}(i,t)}(t) \|^2 \}]. \end{aligned} \tag{4.5}$$

In other words, the augmented Lagrangian is minimized with respect to the variables associated with scenario i assuming that other variables are temporarily fixed at their values $\tilde{x}_j$ for all $j \neq i$. This is done in parallel for each scenario.

Jacobi Method

Step 0. Set $\pi = \pi^k$, $\tilde{x}^{k,r} = x^{k-1}$ *and* $r = 1$.

Step 1. For $i = 1, \ldots, S$ solve (4.5) with $\tilde{x} = \tilde{x}^{k,r}$ obtaining new points $x_i^{k,r}$.

Step 2. If $x_i^{k,r}(t) = \tilde{x}_i^{k,r}(t)$ for all (i,t) such that $i \neq \nu(i,t)$, then stop; otherwise set

$$\tilde{x}_i^{k,r+1}(t) = \tilde{x}_i^{k,r}(t) + \tau(x_i^{k,r}(t) - \tilde{x}_i^{k,r}(t)), \tag{4.6}$$

increase r by 1 and go to Step 1.

Let us note that in order carry out Step 1 for scenario i, we need to know data from scenarios $\nu(i,t)$ and $\nu^{-1}(i,t)$, $t = 1, \ldots, T-1$. In addition to that, the multiplier iteration (3.8) has the form

$$\pi_i^{k+1}(t) = \pi_i^k(t) + \rho(x_i(t) - x_{\nu(i,t)}(t)).$$

So, at scenario i we can update both $\pi_i(t)$ and $\pi_{\nu^{-1}(i,t)}(t)$ (or directly their difference occuring in (4.5)) using the already available data $x_i(t)$, $x_{\nu(i,t)}(t)$ and $x_{\nu^{-1}(i,t)}(t)$. The fact that each multiplier is updated by two subproblems does not matter, because they use the same data. Consequently, both levels of the method: the multiplier update and the Jacobi iteration, can be carried out in a distributed fashion. All these features make our approach especially convenient for parallel computing environments.

Let us now pass to convergence conditions and to the speed of convergence. We immediately see that each constraint (4.2) links variables from only two scenarios. Therefore, the number of neighbors in (3.12) equals

$$N = 1.$$

By Theorem 3.1, apart from the Constraint Qualification Condition, it is sufficient for convergence that the under-relaxation coefficient in (4.6) satisfies the inequalities

$$0 < \tau < 1.$$

This is a very mild requirement.

Assuming additionally the Quadratic Growth Condition, from Theorem 3.2, we obtain the guaranteed ratio of convergence:

$$q = \frac{\tau(1-\tau)}{4\rho\gamma^{-1} + 1}. \tag{4.7}$$

The number 4 in the denominator follows from the observation that the constraint matrix of (4.2) has submatrices A_i which, after removing empty rows and columns, can be permuted to the form

$$A_i = \begin{bmatrix} 1 & & & \\ -1 & & & \\ & 1 & & \\ & -1 & & \\ & & \ddots & \\ & & & 1 \\ & & & -1 \end{bmatrix}.$$

Thus $\|A_i\| \leq \sqrt{2}$ so $\alpha^2 \leq 2$ in (3.15). The best estimate of the ratio (4.7) can be obtained for $\tau = \frac{1}{2}$:

$$q = \frac{1}{16\rho\gamma^{-1} + 4}.$$

For polyhedral cost functions c_t and polyhedral sets $X_i(t)$, $t = 1, \ldots, T$, $i = 1, \ldots, S$, we can additionally observe that (locally) $\gamma = \beta^{-1}\rho$ with some $\beta > 0$ independent of ρ. Then the ratio becomes independent of the penalty parameter ρ:

$$q = \frac{1}{16\beta + 4}.$$

The above results constitute a promising theoretical fundament for an efficient practical method for convex multistage stochastic problems. The computational results of [12] and [1] provide practical evidence for that.

5 Node decomposition

We shall now apply the general framework of section 3 to problem (2.4)-(2.7) with the following assumptions:

- explicit non-anticipativity constraints are removed from the problem by decreasing the number of decision variables;
- equations of dynamics (2.5) are treated as linking constraints.

Let us start by removing explicit non-anticipativity constraints. To achieve that we shall use the scenario tree $\mathcal{T} = \{\mathcal{N}, \mathcal{A}\}$, as described in section 2 and illustrated in Fig. 1. We denote by $a(n)$ the *ancestor* of node n, i.e. the node at the previous level with which n is connected and by $\mathcal{S}(n)$ the set of *successors* of n, $\mathcal{S}(n) = \{m : n = a(m)\}$.

A node n at level t of the tree corresponds to the bundle Ω_n of scenarios which are indistinguishable up to time t. By the non-anticipativity condition (2.7), all decisions $x_\omega(t)$, $\omega \in \Omega_n$, must be equal. We denote their value by x_n.

Next, for each node $n \in \mathcal{N}$, we define probability $\bar{p}_n$ as follows: for each *terminal node* $n \in \mathcal{N}_T$ we set $\bar{p}_n = p_\omega$, where $\omega \in \Omega$ is the event that corresponds to leaf n. For other nodes we define $\bar{p}_n = \sum_{m \in \mathcal{S}(n)} \bar{p}_m$.

Finally, with a slight abuse of notation, for a node n corresponding to event ω at stage t we define:

$$\begin{aligned} D_n &= D_\omega(t), \\ H_n &= H_\omega(t), \\ b_n &= b_\omega(t), \\ X_n &= X_\omega(t), \\ c_n(\cdot) &= c_t(\cdot). \end{aligned}$$

Using this notation we can rewrite (2.4)-(2.7) as follows:

$$\min \sum_{n \in \mathcal{N}} \bar{p}_n c_n(x_n) \tag{5.1}$$

$$D_n x_{a(n)} + H_n x_n = b_n,\ n \in \mathcal{N}, \tag{5.2}$$

$$x_n \in X_n,\ n \in \mathcal{N}, \tag{5.3}$$

where $x_{a(1)} = x(0)$. This corresponds again to the general format (3.1)-(3.3).

The augmented Lagrangian for (5.1)-(5.3) has the form:

$$\begin{aligned} \Lambda(x,\pi) = &\sum_{n \in \mathcal{N}} \bar{p}_n c_n(x_n) + \sum_{n \in \mathcal{N}} \bar{p}_n \langle \pi_n, b_n - D_n x_{a(n)} - H_n x_n \rangle \\ &+ \frac{1}{2}\rho \sum_{n \in \mathcal{N}} \bar{p}_n \|b_n - D_n x_{a(n)} - H_n x_n\|^2. \end{aligned} \tag{5.4}$$

Again, the introduction of scaling factors $\bar{p}_n$ simplifies subproblems (3.10)

$$\begin{aligned} \min_{x_n \in X_n} [\Lambda_n(x_n, \tilde{x}, \pi) = &\ c_n(x_n) - \langle H_n^T \pi_n + \sum_{m \in \mathcal{S}(n)} p_{m|n} D_m^T \pi_m, x_n \rangle \\ &+ \frac{1}{2}\rho \|b_n - D_n \tilde{x}_{a(n)} - H_n x_n\|^2 \\ &+ \frac{1}{2}\rho \sum_{m \in \mathcal{S}(n)} p_{m|n} \|b_m - D_m x_n - H_m \tilde{x}_m\|^2], \end{aligned} \tag{5.5}$$

where $p_{m|n} = \bar{p}_m / \bar{p}_n$ is the probability of getting to node m from node n.

Jacobi Method

Step 0. Set $\pi = \pi^k$, $\tilde{x}^{k,r} = x^{k-1}$ *and* $r = 1$.

Step 1. For $n \in \mathcal{N}$ solve (5.5) with $\tilde{x} = \tilde{x}^{k,r}$ obtaining new points $x_n^{k,r}$.

Step 2. If $D_n x_{a(n)} + H_n x_n = b_n$ for all $n \in \mathcal{N}$ then stop; otherwise set for $n \in \mathcal{N}$

$$\tilde{x}_n^{k,r+1} = \tilde{x}_n^{k,r} + \tau(x_n^{k,r} - \tilde{x}_n^{k,r}), \tag{5.6}$$

increase r by 1 and go to Step 1.

Let us observe that in order carry out Step 1 for node n we need to know data from the predecessor node $a(n)$ and from successors $m \in \mathcal{S}(n)$. In addition to that, the multiplier iteration (3.8) has the form

$$\pi_n^{k+1} = \pi_n^k + \rho(b_n - D_n x_{a(n)} - H_n x_n).$$

So, at node n we can update π_n and all π_m, $m \in \mathcal{S}(n)$, using the already available data $x_n(t)$, $x_{a(n)}$ and x_m, $m \in \mathcal{S}(n)$. Again, the fact that each multiplier is updated by two subproblems does not matter, because they use the same data. Consequently, both levels of the method: the multiplier update and the Jacobi iteration, can be carried out in a distributed fashion with communication along the branches of the scenario tree. This is very convenient for parallel computing environments.

Let us now pass to convergence conditions and to the speed of convergence. We immediately see that each constraint (5.2) links variables from only two nodes, so the number of neighbors in (3.12) equals

$$N = 1.$$

By Theorem 3.1, similar to scenario decomposition, it is sufficient for convergence that the under-relaxation coefficient in (5.6) satisfies the inequalities

$$0 < \tau < 1$$

and the Constraint Qualification Condition holds. Assuming additionally the Quadratic Growth Condition, from Theorem 3.2 we obtain the guaranteed ratio of convergence:

$$q = \frac{\tau(1-\tau)}{2\alpha^2\rho\gamma^{-1} + 1} \tag{5.7}$$

with α defined as in (3.16). Let us estimate α. Assume that $\mathcal{S}(n) = \{m_1, m_2, \dots, m_l\}$. The submatrix A_n of the constraint matrix of (5.2), after removing empty rows, has the form

$$A_n = \begin{bmatrix} H_n \\ D_{m_1} \\ D_{m_2} \\ \vdots \\ D_{m_l} \end{bmatrix}.$$

Thus

$$\|A_n\|^2 \le \|H_n\|^2 + \sum_{m \in \mathcal{S}(n)} \|D_m\|^2.$$

Therefore it is sufficient to use in (5.7)

$$\alpha^2 = \max_{n \in \mathcal{N}} \left[\|H_n\|^2 + \sum_{m \in \mathcal{S}(n)} \|D_m\|^2 \right].$$

The best estimate of the ratio (4.7) can be obtained for $\tau = \frac{1}{2}$:

$$q = \frac{1}{8\alpha^2 \rho \gamma^{-1} + 4}.$$

Again, for polyhedral cost functions c_t and polyhedral sets $X_i(t)$, $t = 1, \ldots, T$, $i = 1, \ldots, S$, we can additionally observe that (locally) $\gamma = \beta^{-1}\rho$ with some $\beta > 0$ independent of ρ. Then the ratio becomes independent of the penalty parameter ρ:

$$q = \frac{1}{8\alpha^2 \beta + 4}.$$

6 Conclusions

The general decomposition framework based on augmented Lagrangians has a number of features which make it particularly promising for large scale problems:

- the dual updates are simple, so the method can be applied to problems with many linking constraints;
- convergence properties of the method improve with the sparsity of the linking constraints.

These general properties become especially important for multistage stochastic programming problems, where large dimension and special structure create a potential for decomposition approaches.

Two ways of applying the general framework to multistage stochastic programming problems have been described:

- scenario decomposition with non-anticipativity treated as a linking constraint;
- node decomposition where the equations of dynamics are linking constraints.

In both cases the resulting problem formulations have many subproblems and very many linking constraints. The constraints, however, are very sparse. Owing to that, in both cases, the method turns out to be particularly simple and robust, with a broad stepsize range guaranteeing convergence and good speed of convergence estimates. It is worth stressing that in both cases parameters of the method are determined by general structural properties of the problems, not by numerical values of the data.

The scenario decomposition method has larger subproblems, which correspond to whole scenarios, and it has to deal with many variables which are just copies of each other. This, however, makes the method very general and flexible. Any causal form of a scenario subproblem is allowed, also with time delays. The objective and constraints need not be separable in time. A practical advantage of the scenario decomposition method is that it allows for an easy introduction of uncertainties into existing deterministic models. One does not have to build a new model, it is sufficient to run in parallel copies of an existing model with different scenario data and to coordinate non-anticipativity of decisions.

The node decomposition method is far more specialized. It has much smaller subproblems, but more of them. The structure of the equations of dynamics is exploited explicitly. The method appears to be better suited for very large problems with many time stages, where scenario subproblems are too large. It requires, however, more modeling effort than the scenario decomposition approach: the node subproblems have to be formulated in the most convenient way. For problems with time delays, new variables may be introduced to keep the number of neighbors equal to 1.

Finally, it has to be stressed that in both cases the structural properties of the problem are reflected in the communication pattern of the method, which is convenient for distributed computing environments.

References

[1] A.J. Berger, J.M. Mulvey and A. Ruszczyński, "Solving stochastic programs with convex objectives", technical report SOR 93-16, Department of Civil Engineering and Operations Research, Princeton University, Princeton 1993.

[2] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, (Academic Press,1982).

[3] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation* (Prentice-Hall, Englewood Cliffs, 1989).

[4] J.R. Birge, "Decomposition and partitioning methods for multistage stochastic linear programs", *Operations Research* 33(1985) 989-1007.

[5] G.B. Dantzig and P. Wolfe, "Decomposition principle for linear programs", *Operations Research* 8(1960) 101-111.

[6] M. Fortin and R. Glowinski, "On decomposition-coordination methods using an augmented Lagrangian", in: *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, M. Fortin and R. Glowinski (eds.), North-Holland, Amsterdam, 1983, pp. 97-146.

[7] H.I. Gassmann, "MSLiP: A computer code for the multistage stochastic linear programming problem", *Mathematical Programming* 47(1990) 407-423.

[8] J. Gondzio and A. Ruszczyński, "A sensitivity method for basis inverse representation in multistage stochastic linear programming problems", *Journal of Optimization Theory and Applications* 74(1992) 221-242.

[9] M. Kallio, A. Ruszczyński and S. Salo, "A regularized Jacobi method for large-scale linear programming", working paper WP-93-61, IIASA, Laxenburg, 1993.

[10] I.J. Lustig, J.M. Mulvey and T.J. Carpenter, "Formulating stochastic programs for interior point methods", *Operations Research* 39(1991) 757-770.

[11] J.M. Mulvey and A. Ruszczyński, "A diagonal quadratic approximation method for large scale linear programs," *Operations Research Letters* 12(1992) 205-215.

[12] J.M. Mulvey and A. Ruszczyński, "A new scenario decomposition method for large-scale stochastic optimization", technical report SOR 91-19, Department of Civil Engineering and Operations Research, Princeton University, Princeton 1991 (to appear in *Operations Research*).

[13] B.N. Pshenichnyi, *Convex Analysis and Extremal Problems* (Nauka, Moskva, 1980) (in Russian).

[14] R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, 1973).

[15] R.T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming", *Mathematics of Operations Research* 1(1976) 97-116.

[16] R.T. Rockafellar and R.J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty", *Mathematics of Operations Research* 16(1991) 1-23.

[17] A. Ruszczyński, "A regularized decomposition method for minimizing a sum of polyhedral functions", *Mathematical Programming* 35(1986) 309-333.

[18] A. Ruszczyński, "Parallel decomposition of multistage stochastic programs", *Mathematical Programming* 58(1993) 201-228.

[19] A. Ruszczyński, "An augmented Lagrangian decomposition method for block diagonal linear programming problems", *Operations Research Letters* 8(1989) 287-294.

[20] A. Ruszczyński, "Augmented Lagrangian decomposition for sparse convex optimization", working paper WP-92-75, IIASA, Laxenburg, 1992 (to appear in *Mathematics of Operations Research*).

[21] G. Stephanopoulos and W. Westerberg, "The use of Hestenes' method of multipliers to resolve dual gaps in engineering system optimization", *Journal of Optimization Theory and Applications* 15(1975) 285-309.

[22] R. Van Slyke and R.J.-B. Wets, " L-shaped linear programs with applications to optimal control and stochastic programming", *SIAM Journal on Applied Mathematics* 17(1969) 638-663.

[23] R.J.-B. Wets, "Large scale linear programming techniques", in: Yu. Ermoliev and R.J.-B. Wets, eds, *Numerical Methods in Stochastic Programming* (Springer-Verlag, Berlin, 1988) pp. 65-94.