

# Mean Value Cross Decomposition for Two-Stage Stochastic Linear Programming with Recourse

Han-Suk Sohn<sup>\*,1</sup>, Dennis L. Bricker<sup>2</sup> and Tzu-Liang (Bill) Tseng<sup>3</sup>

<sup>1</sup>Department of Industrial Engineering, New Mexico State University, Las Cruces, NM, USA

<sup>2</sup>Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA, USA

<sup>3</sup>Department of Industrial Engineering, University of Texas, El Paso, TX, USA

**Abstract:** We present the mean value cross decomposition algorithm and its simple enhancement for the two-stage stochastic linear programming problem with complete recourse. The mean value cross decomposition algorithm employs the Benders (primal) subproblems as in the so-called “L-shaped” method but eliminates the Benders master problem for generating the next trial first-stage solution, relying instead upon Lagrangian (dual) subproblems. The Lagrangian multipliers used in defining the dual subproblems are in turn obtained from the primal subproblems. The primal subproblem separates into subproblems, one for each scenario, each containing only the second-stage variables. The dual subproblem also separates into subproblems, one for each scenario which contains both first- and second-stage variables, and additionally a subproblem containing only the first-stage variables. We then show that the substantial computational savings may be obtained by solving at most iterations only the dual subproblem with the first-stage variables and bypassing the termination test. Computational results are highly encouraging.

**Keywords:** Stochastic linear programming; mean value cross decomposition; lagrangian relaxation; benders’ decomposition.

## 1. INTRODUCTION

Uncertainty is the key factor in many decision problems and stochastic programming represents a powerful tool to treat decision problems under uncertainty. In this paper we focus our attention on a specific class of stochastic programming models: two-stage stochastic linear programming problem with recourse (2-SLPwR). This problem has been successfully applied to a wide range of problems such as fleet management (Powell [1]), asset/liability (Kusy and Ziemba [2], Bertocchi *et al.* [3]), air-traffic control (Richetta and Odoni [4]), scheduling (Birge and Dempster [5]), water resource management (Watkins Jr. and McKinney [6]), investment (Dupacova *et al.* [7]), supply chain (Tomasgard and Hoeg [8]), and so on. For real world two-stage stochastic linear programming problems, however, the number of scenarios is generally very large, so a straightforward usage of conventional optimization tools may require an excessive amount of computational effort. Fortunately, two-stage stochastic linear programming problem has a specific structure which can be exploited by decomposition. The key characteristic of decomposition methods is their ability to iteratively split a large, unwieldy problem into a sequence of smaller, more tractable subproblems. As well, the development of efficient solution algorithm captures significant relationships between primal and dual decomposition. In this short paper, therefore,

we present the mean value cross decomposition algorithm (Holmberg [9]) which is a modification of cross decomposition (Van Roy [10]), and its simple enhancement for the 2-SLPwR. The cross decomposition enables us to explicitly couple both primal and dual decomposition into a single decomposition procedure so that tight lower and upper bounds are achieved simultaneously. Our purpose of this paper is not competing with the most advanced method (Ruszczynski [11], Ruszczynski [12]) for the problem class under consideration, but rather to implement a variation of the mean value cross decomposition, which is a simple yet powerful new saving criterion of wide applicability. The rest of this paper is organized as follows. In section 2 the classical two-stage stochastic programming problem with recourse is presented. The applications of the classical cross decomposition and its simple enhancement to solve the addressed problem are illustrated in section 3. In Section 4 a numerical example is described. In Section 5, the computational results based on randomly generated data are reported. Finally, some concluding remarks are presented in Section 6.

## 2. PROBLEM FORMULATION

Let us consider the two-stage stochastic linear programming problem with recourse (2-SLPwR):

$$\text{minimize } cx + \sum_{k=1}^K p_k Q_k(x) \quad (1)$$

subject to  $x \in X$

where, for example, the feasible set of first-stage decisions is defined by

\*Address correspondence to this author at the Department of Industrial Engineering, New Mexico State University, Las Cruces, NM, USA; Tel: +1-575-646-2957; Fax: +1-575-646-2976; E-mail: hsohn@nmsu.edu

$$X = \{x \in R^n : Ax = b, x \geq 0\} \quad (2)$$

and the cost of the second stage is

$$Q_k(x) = \text{Minimum} \{q_k y : W_k y = h_k - T_k x, y \geq 0\} \quad (3)$$

Here  $k$  indexes the finitely-many possible realizations of a random vector  $\xi$ , with  $p_k$  the probability of realization  $k$ . The set of variables  $x$  are the vector of the first-stage variables, i.e., they are to be selected before uncertain parameter value  $\xi$  is observed. Then the set of second-stage decision variables  $y_k$  (the recourses) are to be selected once  $x$  has been selected and the  $k^{\text{th}}$  realization of  $\xi$  is observed. Note that in general, the coefficient matrices  $T$  and  $W$ , the right-hand-side vector  $h$ , and the second-stage cost vector  $q$  are all random. We assume that recourse is complete, i.e., for any choice of  $x$  and realization  $\xi$ , the set  $\{y : W_k y = h_k - T_k x, y \geq 0\}$  is nonempty. This may require the introduction of artificial variables with large costs. The objective is to minimize the expected total costs of first and second stages. The deterministic equivalent form is a large-scale problem  $P$  which simultaneously selects the first-stage variables  $x$  and the second-stage recourse variables  $y_k$  for every realization  $k$ :

$$P: \text{Find } Z = \text{Minimize } cx + \sum_{k=1}^K p_k q_k y_k \quad (4)$$

subject to  $T_k x + W y_k = h_k, k = 1, \dots, K$ ;

$$x \in X$$

$$y_k \geq 0, k = 1, \dots, K$$

### 3. MEAN VALUE CROSS DECOMPOSITION OF THE PROBLEM

The 2-SLPwR has very large dimension and specific structures that can be handled by decomposition technique and it has stimulated a large amount of researches (Abaify and Allevi [13], Beraldi *et al.* [14], Birge [15], Birge *et al.* [16], Birge and Louveaux [17], Chen and Powell [18], Dantzig and Madansky [19], Donohue [20], Frauendorfer [21], Gassmann [22], Glassey [23], Hagle and Sen [24], Infanger and Morton [25], Linderth and Wright [26], Mulvey and Ruszczyński [27], Nielsen and Zenios [28], Ruszczyński [11], Ruszczyński [29], Ruszczyński and Swietanowski [12], Sen [30], Van Slyke and Wets [31], Vladimirov [32], Wets [33]). The technique presented in this paper is based on the mean value cross decomposition algorithm (Holmberg [9]) which is a modification of cross decomposition algorithm (Van Roy [10]). The cross decomposition unifies Lagrangian (dual) decomposition and Benders' (primal) decomposition technique into a single framework that involves successive solutions to Lagrangian (dual) subproblem and Benders' (primal) subproblem. This approach makes it possible to exploit simultaneously the dual and the primal structure of the problem so that the computational effort is significantly reduced. As well, the proposed technique consists in creating a redundant set of the first-stage variables, and adding a number of coupling

constraints, which brings forth a tightening effect on the Lagrangian bounds (Guignard and Kim [34]). Before describing Lagrangian decomposition, we define a vector variable  $x_0$  as well as a vector  $x_k$  for each scenario  $k, k = 1, \dots, K$ . The variable  $x_k$  represents what one wishes in hindsight that he had decided in the first stage after the realization  $k$  has been observed. The variable  $x_0$  represents the decisions to be implemented in the first stage before  $\xi$  has been realized. We will require that  $x_0 = x_k$  for each scenario  $k$  since the first-stage decision should not depend upon the scenario which will be controlled in the second-stage. This condition is known as a nonanticipativity constraint defined by Rockafellar and Wets [35]. An equivalent formulation of the problem  $P$  is then given by:

$$Z = \text{Minimize } cx_0 + \sum_{k=1}^K p_k q_k y_k \quad (5)$$

subject to  $x_k \in X, k = 0, 1, \dots, K$

$$T_k x_k + W y_k = h_k, k = 1, \dots, K$$

$$x_0 = x_k, k = 1, \dots, K$$

$$y_k \geq 0, k = 1, \dots, K$$

Now given a family of Lagrangian multiplier vectors  $\lambda_k, k = 1, \dots, K$ , the Lagrangian relaxation with respect to the nonanticipativity constraints is the problem of finding  $x_0, x_k$ , and  $y_k$  such that:

$$D(\lambda) = \text{Minimize } cx_0 + \sum_{k=1}^K p_k q_k y_k + \sum_{k=1}^K \lambda_k (x_k - x_0) \quad (6)$$

$$= \text{Minimize } \left( c - \sum_{k=1}^K \lambda_k \right) x_0 + \sum_{k=1}^K [\lambda_k x_k + p_k q_k y_k]$$

subject to  $x_k \in X, k = 0, 1, \dots, K$

$$T_k x_k + W y_k = h_k, k = 1, \dots, K$$

$$y_k \geq 0, k = 1, \dots, K$$

This problem then becomes separable in a subproblem with variable  $x_0$  as well as the  $K$  subproblems corresponding to the individual scenarios  $k$ , which is our purpose in variable splitting. It is well known that, for any choice of  $\lambda$ , the value  $D(\lambda)$  provides a lower bound on the optimal value  $Z$ . The Lagrangian dual problem is to find the tightest such lower bound:

$$\widehat{D} = \text{Maximize}_{\lambda} D(\lambda) \quad (7)$$

The function  $D(\lambda)$  is generally not differentiable everywhere. One common procedure that has been used to search for the optimal Lagrangian multipliers  $\lambda$  has been subgradient optimization (Held *et al.* [36]). This technique adjusts  $\lambda_k$  based upon the magnitude of the violation of the constraint  $x_0 = x_k$ . While the subgradient optimization method can be guaranteed to converge, the rate of convergence is often slow.

Benders' partitioning, known also in stochastic programming as the L-Shaped Method, achieves separability of the second stage decisions, but in a different manner. Given a first-stage decision  $x_0$ , if, as before, we introduce the variables  $x_k$  with the nonanticipativity constraints, we obtain the second-stage problem for each scenario  $k$  :

$$Q_k(x_0) = \text{Minimize } q_k y_k \tag{8}$$

subject to  $T_k x_k + W y_k = h_k$ ,

$$x_k = x_0,$$

$$y_k \geq 0$$

It is well known that  $P(x_0) = cx_0 + \sum_{k=1}^K p_k Q_k(x_0)$  provides us with an upper bound on the optimal value  $Z$ . The linear programming dual of the second-stage problem is the linear program:

$$Q_k(x_0) = \text{Maximize } (h_k - T_k x_0) \pi_k \tag{9}$$

subject to  $\pi_k W \leq q_k$

in which the dual variables  $\lambda_k$  for the constraint  $x_0 = x_k$  has been eliminated since  $\lambda_k = -\pi_k T_k$ , and the given assumption that polyhedrons  $\pi_k W \leq q_k$  for  $k=1, \dots, K$  are bounded. Note that the polyhedral feasible region of the second-stage problem for scenario  $k$  is denoted by  $\Pi_k = \{\pi_k : W^T \pi_k \leq q_k\}$ , and the  $i^{\text{th}}$  extreme point of  $\Pi_k$  by  $\hat{\pi}_k^i$ ,  $i=1, 2, \dots, I_k$ . By enumerating the generally large but finite set of extreme points we can write

$$Q_k(x_0) = \text{Maximize}_{i=1, \dots, I_k} \{\hat{\pi}_k^i (h_k - T_k x_0)\} = \text{Maximize}_{i=1, \dots, I_k} \{\hat{\lambda}_k^i x_0 + \hat{\alpha}_k^i\} \tag{10}$$

where  $\lambda_k^i = -\hat{\pi}_k^i T_k$  and  $\hat{\alpha}_k^i = \hat{\pi}_k^i h_k$ . Note that this demonstrates that  $Q_k(x_0)$  is a piecewise-linear convex function. Benders' complete master problem then uses this representation of  $Q_k(x_0)$  to provide an alternate method for evaluating  $Z$ , namely

$$Z = \text{Minimize } cx_0 + \sum_{k=1}^K p_k \theta_k \tag{11}$$

subject to  $x_0 \in X$

$$\theta_k \geq \hat{\lambda}_k^i x_0 + \hat{\alpha}_k^i, \quad i = 1, \dots, I; \quad k = 1, \dots, K$$

While it is possible in principle to solve the problem using Benders' complete master problem, in practice the magnitude of the number of dual extreme points makes it prohibitively expensive. However, if a subset of the dual extreme points of  $\Pi_k$  are available, e.g.,  $\hat{\pi}_k^i$ ,  $i = 1, 2, \dots, M_k$  where  $M_k < I_k$ , then we obtain an underestimate of  $Q_k(x_0)$ , which we denote by

$$\underline{Q}_k(x_0) = \text{Maximize}_{i=1, \dots, M_k} \{\hat{\lambda}_k^i x_0 + \hat{\alpha}_k^i\} \tag{12}$$

Thus, by making use of dual information obtained after  $M$  evaluations of  $Q_k(x_0)$ , we obtain a Partial Master Problem,

$$\Phi_M = \text{Minimize } cx_0 + \sum_{k=1}^K p_k \theta_k \tag{13}$$

subject to  $x_0 \in X$

$$\theta_k \geq \hat{\lambda}_k^i x_0 + \hat{\alpha}_k^i, \quad i = 1, \dots, M; \quad k = 1, \dots, K$$

which provides a lower bound on the solution of  $Z$ . Benders' algorithm solves the current Partial Master Problem, obtaining  $x_0$  (a "trial solution") and an underestimate  $\sum_{k=1}^K p_k \underline{Q}_k(x_0)$  of the associated expected second-stage cost.

The actual expected second-stage cost, i.e.,  $\sum_{k=1}^K p_k Q_k(x_0)$ , is

then evaluated by solving the second-stage problem for each scenario. Additional constraints are added to the Partial Master Problem to complete the iteration. At each iteration,

then, the subproblem solution  $P(x_0) = cx_0 + \sum_{k=1}^K p_k Q_k(x_0)$

provides an upper bound for  $Z$  and the Partial Master Solution  $\Phi_M = \underline{P}(x_0) = cx_0 + \sum_{k=1}^K p_k \underline{Q}_k(x_0)$  provides a lower bound.

The cross decomposition combines the advantages of Benders' (primal) decomposition and Lagrangian (dual) decomposition. It consists mainly of a subproblem phase, where one iterates between primal subproblem and dual subproblem as below:

*Step 0.* Initialization. Let the initial values of the upper bound  $\bar{Z} = +\infty$ , lower bound  $\underline{Z} = -\infty$ . The algorithm can start with either the dual subproblem or the primal subproblem. After solving any of the subproblems, the following tasks are to be performed. Note that, for the purpose of illustration, we start with solving the dual subproblem

*Step 1a.* Solve the dual subproblem to evaluate  $D(\lambda)$  with the given initial values of the dual variables  $\lambda_k = 0$  for  $k = 1, \dots, K$ . This separates into  $K+1$  subproblems, i.e.,  $D_k(\lambda)$ ,  $k = 0, 1, \dots, K$ .

*1b.* From the solution of  $D_0(\lambda)$  determine the next proposed first-stage decision  $x_0$ .

*1c.* Update the lower bound, i.e.,  $\underline{Z} = \max\{\underline{Z}, D(\lambda)\}$ .

*1d.* Check for optimality: if  $\bar{Z} - \underline{Z} \leq \epsilon$ , i.e., the upper and the lower bounds on the objective function are close enough, then STOP; else continue to *Step 2*.

*Step 2a.* Solve the primal subproblem to evaluate  $P(x_0)$  with the given values of the first-stage decisions  $x_0$ . This separates into  $K$  subproblems for each scenario  $k = 1, \dots, K$ .

*2b.* Construct a Lagrangian multiplier vector  $\lambda = [\lambda_1, \dots, \lambda_k]$ .

2c. Generate an optimality cut to be added to the primal (or Benders') master problem.

2d. Update the upper bound, i.e.,  $\bar{Z} = \min\{\bar{Z}, P(x_0)\}$ .

2e. Check for optimality: if  $\bar{Z} - \underline{Z} \leq \epsilon$ , STOP; else continue to Step 1.

At each iteration, upper and lower bounds on the optimal value  $Z$  may be updated. However, the convergence of the bounds cannot be assured, as "stalling" can occur, e.g., Step 1b may generate the same first-stage decisions  $x_0$  in successive iterations. In this case, we must rely on a solution of the standard Benders' master problem to restore convergence before continuing. The need to use master problem is a serious disadvantage of the cross decomposition method, since the master problem usually are much harder to solve than the subproblems. This reliance on the master problem may be avoided when using a variation by K. Holmberg (Holmberg [9], Holmberg [37]) called mean value cross decomposition. The base for mean value cross decomposition algorithm is a subproblem phase in the cross decomposition, where the algorithm iterates between the dual subproblem and the primal subproblem. However, it eliminates completely the need for using master problems so that step 2c may be omitted. This variation of cross decomposition maintains the mean values of all previously-generated first-stage decisions  $x_0$  and Lagrangian multipliers  $\lambda$ . The current mean values of  $x_0$  and  $\lambda$  are then used in the primal and dual subproblems, respectively. At each iteration, then, for both the Lagrangian and Benders' subproblems, one solves a set of linear programming problems, one for each realization  $k$ . The Lagrangian subproblem includes additional variables,  $x_k$ , reflecting the first-stage decisions that, after  $\xi$  has been realized, we wish in hindsight that we had made. In the Lagrangian subproblem, an additional linear programming must be solved at each iteration in order to determine the first-stage decisions,  $x_0$ .

It is worth noting that, in practice, we could omit solution of the dual subproblems  $D_k(\lambda)$  for each realization  $k$  in most iterations. This is because the information to be returned by the dual subproblem consists of both lower bound  $\underline{Z}$  on the optimum, used in the termination test and first-stage decisions,  $x_0$ , and solution of dual subproblem 0, i.e.,  $D_0(\lambda)$ , is sufficient to determine  $x_0$ . However, the termination test must be bypassed, since no new lower bound will have been computed. Hence, we might iterate between  $D_0(\lambda)$  and  $P_k(x_0)$  for each realization  $k$  for several iterations if it is expected that the termination criterion would not be satisfied, and only occasionally test the termination criterion by solving the dual subproblems  $D_k(\lambda)$  for each realization  $k$ . As a result, the computational effort will be substantially reduced by solving only one of the  $K+1$  subproblems, especially since for each realization  $k$ , the subproblems  $D_k(\lambda)$  include both first and second stage variables. It should be noted that the dual subproblem  $D_0(\lambda)$  includes only first-stage variables (see Fig. 1).

4. EXAMPLE

To illustrate the algorithm discussed in the earlier section, we use the farmer's problem in Birge and Louveaux's book (Birge and Louveaux [38]). A farmer raises wheat, corn, and sugar beets on 500 acres of land. Before the planting season he wants to decide how much land to devote to each crop. At least 200 tons of wheat and 240 tons of corn are needed for cattle feed, which can be purchased from a wholesaler if not raised on the farm. Any grain in excess of the cattle feed requirement can be sold at \$170 and \$150 per ton of wheat and corn, respectively. If additional wheat or corn is required to supplement that which is raised on the farm, the cost is 40% more, namely \$238 and \$210 per ton, respectively. Up to 6000 tons of sugar beets can be sold at a subsidized price of \$36 per ton. However, any additional amounts can be sold for only \$10 per ton. Crop yields are uncertain, depending upon weather conditions during the growing season, and three scenarios

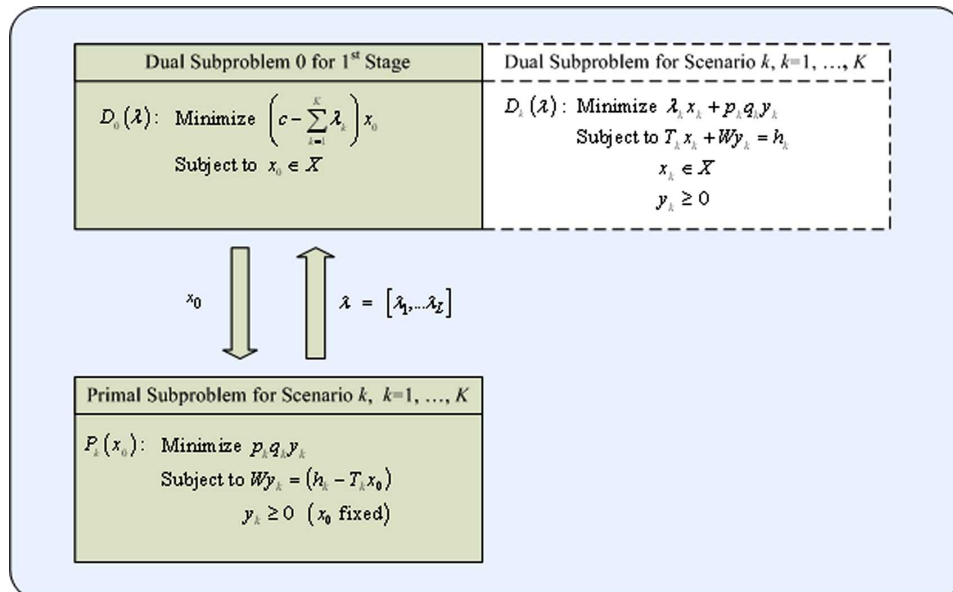


Fig. (1). Information exchanges in the mean value cross decomposition.

have been identified, each equally likely. That is, it is assumed that years are good, fair, or bad for all crops, resulting in above average, average, or below average yields for all crops. The average yield on his land is roughly 2.5, 3, and 20 tons per acre for wheat, corn, and sugar beets, respectively. The above and below average indicate a yield 20% above or below the mean yield.

Decision variables  $x_i$  is the first-stage decision which represents acres of land planted in each crop whereas decision variables  $w_i$  and  $y_j$  are the second-stage decisions that represent tons of each crop sold and purchased, respectively. The stochastic decision problem then becomes as follows:

$$\text{minimize } 150x_1 + 230x_2 + 260x_3 + \frac{1}{3} \sum_{k=1}^3 Q_k(x) \quad (14)$$

$$\text{subject to } x_1 + x_2 + x_3 \leq 500$$

$$x_j \geq 0, j=1,2,3$$

$$\text{where } Q_1(x) = \text{Minimum } 170w_1 + 150w_2 + 36w_3 + 10w_4 - 238y_1 - 210y_2 \quad (15)$$

$$\text{s.t. } y_1 - w_1 \geq 200 - 3x_1$$

$$y_2 - w_2 \geq 240 - 3.6x_2$$

$$w_3 + w_4 \leq 24x_3$$

$$y_1 \geq 0, y_2 \geq 0$$

$$w_1 \geq 0, w_2 \geq 0, 0 \leq w_3 \leq 6000, w_4 \geq 0,$$

$$Q_2(x) = \text{Minimum } 170w_1 + 150w_2 + 36w_3 + 10w_4 - 238y_1 - 210y_2 \quad (16)$$

$$\text{s.t. } y_1 - w_1 \geq 200 - 2.5x_1$$

$$y_2 - w_2 \geq 240 - 3x_2$$

$$w_3 + w_4 \leq 20x_3$$

$$y_1 \geq 0, y_2 \geq 0$$

$$w_1 \geq 0, w_2 \geq 0, 0 \leq w_3 \leq 6000, w_4 \geq 0$$

and

$$Q_3(x) = \text{Minimum } 170w_1 + 150w_2 + 36w_3 + 10w_4 - 238y_1 - 210y_2 \quad (17)$$

$$\text{s.t. } y_1 - w_1 \geq 200 - 2x_1$$

$$y_2 - w_2 \geq 240 - 2.4x_2$$

$$w_3 + w_4 \leq 16x_3$$

$$y_1 \geq 0, y_2 \geq 0$$

$$w_1 \geq 0, w_2 \geq 0, 0 \leq w_3 \leq 6000, w_4 \geq 0$$

The mean value cross decomposition algorithm described in the previous section was implemented in Mosel language for Xpress-MP and was compared to Benders' decomposition algorithm and the ordinary cross decomposition algorithm. The problem was not solved to completion this time, but was terminated after 100 iterations. Table 1 summarizes the results obtained by these four algorithms.

Both Benders' decomposition and the ordinary cross decomposition can find the optimal solution within 10 iterations and the optimal first-stage decisions are to plant 170, 80, and 250 acres of wheat, corn, and beets, respectively, which will yield a minimum expected cost of \$108,390, i.e., a profit of \$108,390. The mean value cross decomposition presented finds an incumbent solution at iteration #85 with a total cost of \$108,177 which yields approximately 0.197% of duality gap. The behaviors of the mean values of  $x$  passed to the primal subproblems as well as the total expected cost computed by the primal subproblems are shown in Figs. (2, 3). From these figures, we may say that the mean value cross decomposition algorithm converges asymptotically without solving master problems. Therefore, the algorithm requires much less memory than both Benders' decomposition and the ordinary cross decomposition, and can solve much larger problems. At the risk of over-generalizing based upon the computational results obtained for this relatively small example, it suffices to demonstrate the use of the mean value cross decomposition for generating solution of the addressed problem and suggests that the results are comparable to those obtained by Benders' original algorithm and the ordinary cross decomposition algorithm.

### 5. COMPUTATIONAL STUDY

In order to compare the performance of the mean value cross decomposition presented with ones of Benders' decomposition, the ordinary cross decomposition, and the implemented algorithms were coded in the APL language using APL+Win 3.6 and run on an Intel Pentium IV CPU 3GHz with 1GB of RAM. The test problems are generated according to the following scheme:

**Table 1. Summary of Results from Four Different Algorithms**

	Benders' Decomposition	Ordinary Cross Decomposition	Mean Value Cross Decomposition
# of iteration	5	7	94
# of Master Problem	5	1	0
CPU time (sec)	1.435	1.420	1.096
Duality Gap (%)	0	0	0.197

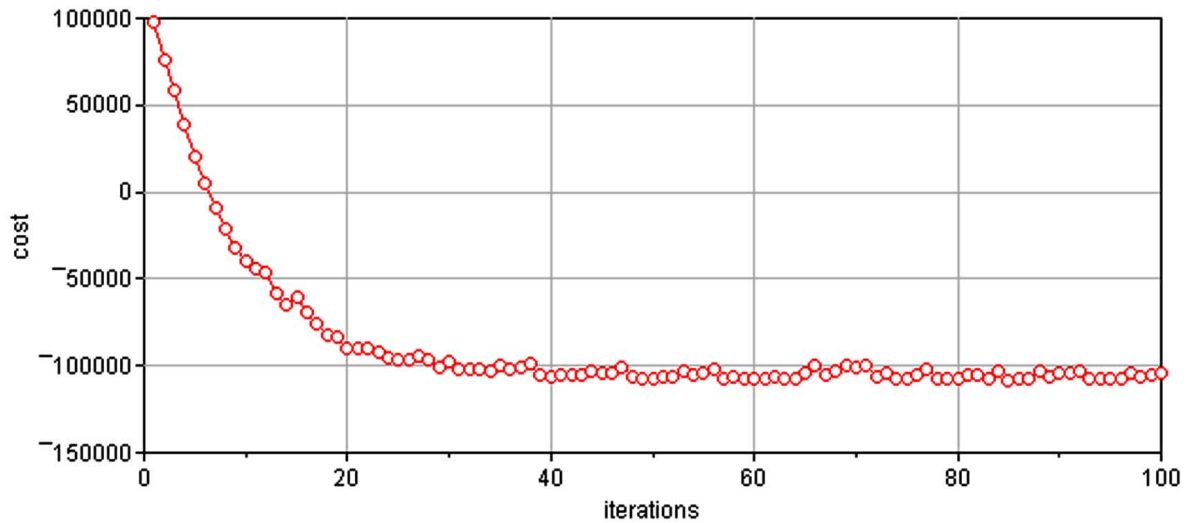


Fig. (2). Convergence of total expected cost vs iteration.

1. A set of five random problems were generated for each of the dimensions:  $K \in n^4$ , for  $n = 2, 3, \dots, 15$ . Therefore, 70 test problems were obtained. Here,  $K$  represents the number of scenario to be considered.
2. The last first-stage constraint is a generalized upper bound constraint in order to bound the feasible region of the first-stage problem so that the Benders' master problem does not have an unbounded solution.
3. The second-stage variables include simple recourse variables with relatively high costs, so that recourse is complete, i.e., the second stage is feasible for every scenario for every first-stage decision.
4. The coefficient matrices  $T$  and  $W$  of both first- and second-stage variables in the second stage constraints are fixed, i.e., only the right-hand-sides and the second-stage costs are random.

The APL program stops when the gap between the upper and lower bound is either less than 0.001%, or after the

computing time exceeds two hours. The outcome of the computational experience is reported in Tables 2-4, and they report on the average results taken over five random problems for each problem size.

First, Benders' decomposition algorithm was implemented and the corresponding results are summarized in Table 2. We list the problem name (ID), the number of scenario ( $K$ ), the number of second-stage constraints ( $m_2$ ), the running time when the optimal solutions were found (CPU time), and the number of Benders master problems solved (NBM). As can be seen in Table 2, the mean computing times increase dramatically in keeping with the dimension of the test instances. Even though only about 13 iterations on the average were needed for the Benders' decomposition procedure, it was unable to optimally solve the largest two instances within a specified time. This is mainly because the Bender's decomposition algorithm uses traditional search methods for the master problem, where the majority of computing time is spent.

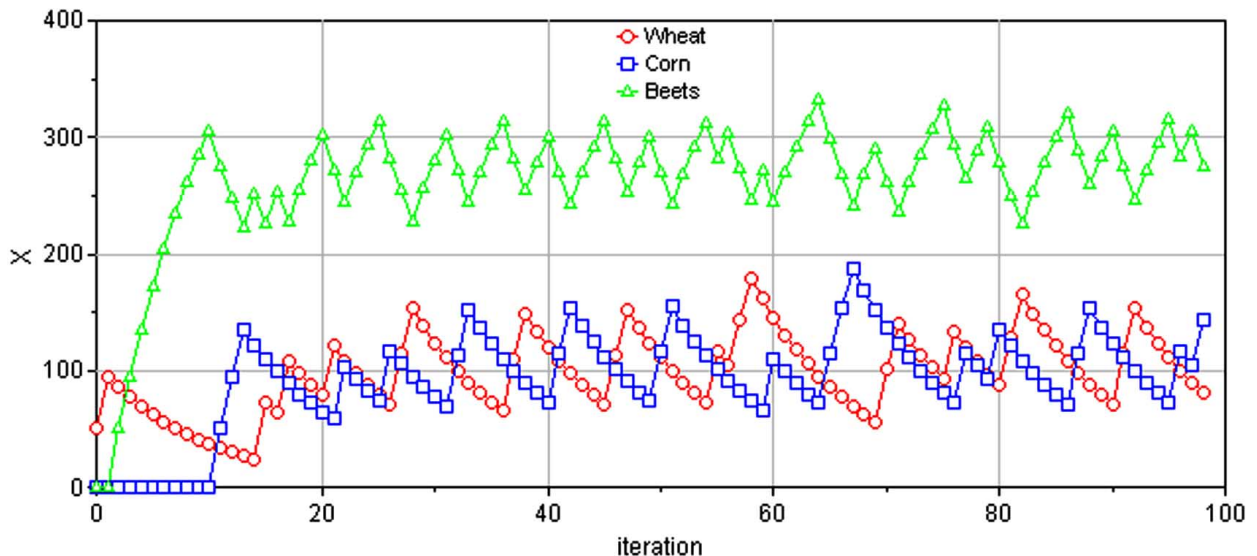


Fig. (3). Convergence of first-stage variables  $x_i$  vs iteration.

**Table 2. Computational Results from Benders' Decomposition**

ID	K	m2	CPU time	NBM
s2p4	16	64	1.4166	7
s3p4	81	324	1.9218	7.6
s4p4	256	1,024	2.4272	12
s5p4	625	2,500	3.2636	8.8
s6p4	1,296	5,184	6.2946	8.4
s7p4	2,401	9,604	21.4294	17
s8p4	4,096	16,384	45.2732	15.8
s9p4	6,561	26,244	73.3376	9.2
s10p4	10,000	40,000	257.3592	16.2
s11p4	14,641	58,564	470.5700	16.6
s12p4	20,736	82,944	1,028.1780	15.6
s13p4	28,561	114,244	1,861.1640	17.4
s14p4	38,416	153,664	--	16.4
s15p4	50,625	202,500	--	17.4

Secondly, the ordinary cross decomposition algorithm was used. That is, the algorithm was restarted but the Benders' cuts were modified by the algorithm of Van Roy. The results presented in Table 3 show the efficiency of this variation of the original Benders' decomposition. For all test instances, less iterations were needed for the cross decomposition for strengthening the Benders' cuts than for the original Benders' decomposition. As well, on the average much less computational times were required and the cross decomposition algorithm was able to solve the second largest

**Table 3. Computational Results from Ordinary Cross Decomposition**

ID	K	m2	CPU time	NBM	RT <sub>BD-CD</sub>
s2p4	16	64	1.263	2.8	10.84
s3p4	81	324	1.6692	3.6	13.14
s4p4	256	1,024	2.1245	5.4	12.47
s5p4	625	2,500	2.7454	4.8	15.88
s6p4	1,296	5,184	4.2791	4.8	32.02
s7p4	2,401	9,604	13.8620	9.2	35.31
s8p4	4,096	16,384	32.3513	6.4	28.54
s9p4	6,561	26,244	58.6054	4.4	20.09
s10p4	10,000	40,000	155.3484	7.2	39.64
s11p4	14,641	58,564	348.9646	10.4	25.84
s12p4	20,736	82,944	704.3740	8.4	31.49
s13p4	28,561	114,244	1,344.6710	11.6	27.75
s14p4	38,416	153,664	3,234.9670	9.2	34.10
s15p4	50,625	202,500	--	7	41.66

instance optimally within two hours. Compare to the Benders' original algorithm, the cross decomposition requires only about 50% of the number of master problem to generate the optimal solution, which results in about 26% of savings in computational time on the average. This is not surprising since the ordinary cross decomposition makes it possible to couple explicitly both primal and dual subproblems into a single decomposition procedure which enables us to achieve tight lower and upper bounds simultaneously. Note that RT<sub>BD-CD</sub> refers the percentage of time's reduction when Benders' decomposition algorithm is replaced by the ordinary cross decomposition algorithm.

Finally, the mean value cross decomposition algorithm was implemented by restarting the subproblem phase in the ordinary cross decomposition, and we take the mean values of all the previously obtained solutions and proceed with the cross decomposition. In most iteration, however, only the dual subproblem with the first-stage variables is solved. Table 4 gives the results, and we list RT<sub>BD-SCD</sub> and RT<sub>OCD-SCD</sub> as the percentage reduction in time if the cross decomposition replaces the Benders' decomposition and the ordinary cross decomposition, respectively. As the problem size becomes larger, the mean value cross decomposition algorithm performs better than both Benders' decomposition and the ordinary cross decomposition algorithms. Compare to these two decomposition schemes, on the average, the mean value cross decomposition algorithm can save about 36% and 16 % of its total consuming time to find the optimal solution. As a result, the algorithm can find the optimal solution for all test instances within a specified time. These results indicate that the mean value cross decomposition presented outperforms both Benders' decomposition and the ordinary cross decomposition in terms of the computing times while no degradation of solution quality. The results

**Table 4. Computational Results from Mean Value Cross Decomposition Presented**

ID	K	m2	CPU Time	RT <sub>BD-SCD</sub>	RT <sub>OCD-SCD</sub>
s2p4	16	64	1.5320	-8.15	-21.30
s3p4	81	324	2.4927	-29.71	-49.34
s4p4	256	1,024	3.2082	-32.18	-51.01
s5p4	625	2,500	2.13495	34.58	22.24
s6p4	1,296	5,184	2.81225	55.32	34.28
s7p4	2,401	9,604	8.07055	62.34	41.78
s8p4	4,096	16,384	22.9067	49.40	29.19
s9p4	6,561	26,244	44.9784	38.67	23.25
s10p4	10,000	40,000	105.9769	58.82	31.78
s11p4	14,641	58,564	242.1565	48.54	30.61
s12p4	20,736	82,944	506.6693	50.72	28.07
s13p4	28,561	114,244	924.5226	50.33	31.25
s14p4	38,416	153,664	2,239.8190	54.37	30.76
s15p4	50,625	202,500	4,217.2160	69.52	47.76

also demonstrate that the mean value cross decomposition algorithm and its simple enhancement presented are very effective for the addressed two-stage stochastic linear programming problem.

## 6. CONCLUSION

In this paper, we have presented mean value cross decomposition algorithm and its simple enhancement for the two-stage stochastic linear programming problem with recourse. Unlike Benders' decomposition algorithm or L-shaped method, in which primal and dual variables are exchanged between a primal master problem and primal subproblem, the cross decomposition algorithm obtains the primal variables from a Lagrangian relaxation, thus unifying Benders' decomposition and Lagrangian relaxation into a single framework which makes it possible to exploit both primal and dual structures of the problem simultaneously. In order to avoid stalling, in which case the algorithm must occasionally rely upon a master problem, mean values of the primal and dual variables may be exchanged. While constructing the Lagrangian dual subproblem for the mean value cross decomposition method, we introduce extra variables for each scenario, and use them, together with the nonanticipativity constraints, to replace the first-stage variable in each of the second-stage constraints. With this, the Lagrangian dual subproblem decomposes into a series of smaller problems. Furthermore, the first-stage variables only appear in one of these smaller problems. The mean value cross decomposition with the saving criterion, therefore, is able to achieve the substantial computational savings by solving at most iterations only the dual subproblem with the first-stage variables and bypassing the termination test. The computational results show that quite large problems can be solved fairly quickly by the proposed algorithm. The results also reveal that the proposed algorithm is quite easily implemented. Our purpose in developing the proposed algorithm was not to compete with the most advanced method for the problem class under consideration, but rather to implement a simple yet powerful new saving criterion of wide applicability. Future work could extend the proposed algorithm to other stochastic programming problems.

## 7. AVAILABILITY

All the tested instances used in this report as well as the optimal solution for each instances are available at the web page [http://ie.nmsu.edu/sohn\\_stage2/2SLPwR.htm](http://ie.nmsu.edu/sohn_stage2/2SLPwR.htm). Further enquiries regarding this test instances may be made to the corresponding author at [hsohn@nmsu.edu](mailto:hsohn@nmsu.edu).

## ACKNOWLEDGEMENTS

This work was partially supported by the US Department of Agriculture (USDA under grant #2011-38422-30803). The authors are grateful to the anonymous referees for their helpful comments and suggestions which led to an improved presentation of the work.

## REFERENCE

- [1] Powell WB. A stochastic model of the dynamic vehicle allocation problem. *Transp Sci* 1986; 20: 117-29.
- [2] Kusy MI, Ziemba WT. A bank asset and liability management model. *Ope Res* 1986; 34(3): 356-76.
- [3] Bertocchi M, Moriggia V and Dupacova J Chapter 7: Bond Portfolio Management *via* Stochastic Programming. *Handbook of Asset and Liability Management: Volume 1*. Zenios SA, Ziemba WT, Elsevier. 2006: 305-36.
- [4] Richetta O, Odoni AR. Dynamic solution to the ground-holding problem in air traffic control. *Transp Res Part A: Policy and Practice*. 1996; 28(3): 167-85.
- [5] Birge JR and Dempster MAH. Stochastic programming approach to stochastic scheduling. *J of Glob Optim* 1996; 9(3-4): 417-51.
- [6] Watkins Jr. DW and McKinney DC. Finding robust solutions to water resources problems. *J of Water Resour Plan and Manag* 1997; 123: 49-58
- [7] Dupacova J, Consigli G and Wallace SW. Scenarios for Multistage Stochastic Programs. *Ann of Ope Res* 2000; 100: 25-53.
- [8] Tomasgard A and Hoeg E. Chapter 14: A supply chain optimization model for the Norwegian meat cooperative. *Applications of Stochastic Programming*. Wallace SW, Ziemba, W.T., Society for Industrial and Applied Mathematic 2005; 253-76.
- [9] Holmberg K. Linear mean value cross decomposition: A generalization of the Kornai-Liptak method. *Eur J of Ope Res* 1992; 62(1): 55-73.
- [10] Van Roy TJ. Cross decomposition for mixed integer programming. *Math Program* 1983; 25: 46-63.
- [11] Ruszczyński A. A regularized decomposition method for minimizing a sum of polyhedral functions. *Math Program* 1986; 35: 309-33.
- [12] Ruszczyński A, Swietanowski A. Accelerating the regularized decomposition method for two stage stochastic linear problems. *Eur J of Ope Res* 1997; 101: 328-42.
- [13] Abaffy J, Allevi E. A modified L-Shaped Method. *J Optim Theory Appl* 2004; 123(2): 255-70.
- [14] Beraldi P, Grandinetti L, Mousmanno R and Triki C. Parallel algorithms to solve two-stage stochastic linear programs with robustness constraints. *Parallel Comput* 2000; 26(13): 1889-908.
- [15] Birge JR. Decomposition and partitioning methods for multistage stochastic linear programs. *Ope Res* 1985; 33: 989-1007.
- [16] Birge JR, Donohue CJ, Holmes DF, Svitsitski OG. A parallel implementation of the nested decomposition method for multistage stochastic linear programs. *Math Program* 1996; Series B 75: 327-52.
- [17] Birge JR, Louveaux FV. A multicut algorithm for two-stage stochastic linear programs. *Eur J of Ope Res* 1988; 34(3): 384-92.
- [18] Chen ZL, Powell WB. Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *J of Optim Theor and Appl* 1999; 102: 497-524.
- [19] Dantzig G, Madansky A. On the solution of two-stage linear programs under uncertainty. *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley: University of California Press 1961.
- [20] Donohue CJ, Birge, J.R. The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorith Ope Res* 2006; 1: 20-30.
- [21] Frauendorfer K. *Stochastic two-stage programming*. Berlin, Springer 1992.
- [22] Gassmann HI. A computer code for the multistage stochastic linear programming problem. *Math Program* 1990; Series B 47: 407-23.
- [23] Glassey CR. Nested decomposition and multistage linear programs. *Manag. Sci* 1973; 20: 282-92.
- [24] Hagle JL, Sen S. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Math of Ope Res* 1991; 16(3): 650-69.
- [25] Infanger G, Morton DP. Cut sharing for multistage stochastic linear programs with interstage dependency. *Math Program* 1996; Series B 75: 241-56.
- [26] Linderoth JT, Wright SJ. Implementing a decomposition algorithm for stochastic programming on a computational grid, Preprint ANL/MCS-P875-0401., Mathematics and Computer Science Division, Argonne National Laboratory 2001.
- [27] Mulvey JM, Ruszczyński. A new scenario decomposition method for large-scale stochastic optimization. *Ope Res* 1995; 43: 477-90.
- [28] Nielsen SS, Zenios SA. Scalable parallel Benders decomposition for stochastic linear programming. *Parallel Comput* 1997; 23(8): 1069-88.
- [29] Ruszczyński A. Parallel decomposition of multistage stochastic programming problem. *Math Program* 1993; 58: 201-28.



- [30] Sen S. Subgradient decomposition and differentiability of the recourse function of a two stage stochastic linear program. *Ope Res Lett* 1993; 13(3): 143-48.
- [31] Van Slyke RM, Wets R. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J of Appl Math* 1969; 17(4): 638-63.
- [32] Vladimirov H. Theory and methodology computational assessment of distributed decomposition methods for stochastic linear programs. *Eur J of Ope Res* 1969; 108: 653-70.
- [33] Wets RJ-B. Large scale linear programming techniques. *Numerical Techniques for Stochastic Optimization* (Springer Series in Computational Mathematics) Ermoliev Y, Wets, R.J.-B. Berlin, Springer 1988; 65-93.
- [34] Guignard M, Kim S. Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Math Program, Series B* 1987; 39: 215-28.
- [35] Rockafellar RT, Wets RJ-B. Scenarios and policy aggregation in optimization under uncertainty. *Math of Ope Res* 1991; 16: 119-47.
- [36] Held M, Wolfe, P., Crowder, H.P. Validation of Subgradient Optimization. *Math Program, 1974; Series B* 6(1): 62-88.
- [37] Holmberg K. A convergence proof for linear mean value cross decomposition. *Zeitschrift fur Ope Res* 1994; 39(2): 157-86.
- [38] Birge JR, Louveaux F. *Introduction to stochastic programming*. New York, Springer-Verlag 1997.

---

Received: October 10, 2011

Revised: November 1, 2011

Accepted: November 1, 2011

© Sohn *et al.*; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.