Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research xxx (2004) xxx–xxx

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Discrete Optimization

A factor $\frac{1}{2}$ approximation algorithm for two-stage stochastic matching problems

Nan Kong, Andrew J. Schaefer [*]

Department of Industrial Engineering, University of Pittsburgh, 1048 Benedum Hall, Pittsburgh, PA 15261, USA

Received 26 November 2003; accepted 22 October 2004

Abstract

We introduce the two-stage stochastic maximum-weight matching problem and demonstrate that this problem is $\mathcal{NP}$-complete. We give a factor $\frac{1}{2}$ approximation algorithm and prove its correctness. We also provide a tight example to show the bound given by the algorithm is exactly $\frac{1}{2}$. Computational results on some two-stage stochastic bipartite matching instances indicate that the performance of the approximation algorithm appears to be substantially better than its worst-case performance.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Stochastic programming; Approximation algorithms; Matching; Combinatorial optimization

1. Introduction

Let $G = (V, E)$ be a graph, and let each edge $e \in E$ have an edge weight c_e. The maximum-weight matching problem (Cook et al., 1998) is

$$\max\left\{\sum_{e\in E} c_e x_e \middle| \sum_{e\in\delta(v)} x_e \leqslant 1, \forall v\in V; x_e\in\{0,1\}, \forall_e\in E\right\}. \tag{1}$$

It is well known that the maximum-weight matching problem is polynomially solvable (Edmonds, 1965). Consider a stochastic programming extension of this problem as follows. Each edge has two weights, a first-stage weight c_e, and a discretely distributed second-stage weight $\tilde{d}_e$. The first-stage decision x is to choose a matching in G. After the decision, a scenario of the second-stage edge weights is realized. That is, each edge weight is assigned to one of the r possible values $d_e^1, \ldots, d_e^r$ with corresponding probabilities $p_1, \ldots, p_r$. For each scenario $s = 1, \ldots, r$, the second-stage decision y^s is to choose a matching over those vertices unmatched by the first-stage matching. Without loss of generality, the edge weights c_e and d_e^s for

[*] Corresponding author. Tel.: +1 412 624 5045; fax: +1 412 624 9831.
E-mail address: schaefer@ie.pitt.edu (A.J. Schaefer).

each scenario $s = 1,\ldots,r$ are nonnegative, since any edge with negative c_e or d_e^s won't be chosen in any optimal solution. The goal is to maximize the total expected edge weight in these matchings. The stochastic programming extension of (1) can then be written as:

$$\max \sum_{e\in E} c_e x_e + \sum_{s=1}^{r} p_s \sum_{e\in E} d_e^s y_e^s \tag{2}$$

subject to

$$\sum_{e\in\delta(v)} x_e + \sum_{e\in\delta(v)} y_e^s \leqslant 1, \quad \forall v \in V,\ s = 1,\ldots,r,$$

$$x_e \in \{0,1\},\ y_e^s \in \{0,1\}, \quad \forall e \in E,\ s = 1,\ldots,r.$$

For an introduction to stochastic programming, we refer to Kall and Wallace (1994) and Birge and Louveaux (1997). Interestingly, unlike the polynomially solvable deterministic maximum-weight matching problem, this stochastic programming extension is $\mathcal{NP}$-complete, as will be shown in Section 2. Therefore, it is natural to develop approximation algorithms that finds solutions with a performance guarantee in a polynomial number of steps for the stochastic programming extension. Hochbaum (1997) and Vazirani (2001) provided surveys of approximation algorithms. There have been very few studies of the computational complexity of stochastic programs and the applications of approximation algorithms to such problems. Dye et al. (2003) studied the computational complexity of the stochastic single-node service provision problem arises from an application of distributed processing in telecommunication networks. They showed the strong $\mathcal{NP}$-completeness of the problem and presented several approximation algorithms.

The remainder of the paper is organized as follows. In Section 2, we show the $\mathcal{NP}$-completeness of the stochastic matching problem. In Section 3, we present a factor $\frac{1}{2}$ approximation algorithm and provide a class of instances for which the bound is tight. Section 4 provides computational results that show the performance of the approximation algorithm on a set of randomly generated two-stage stochastic bipartite matching instances.

2. The complexity of two-stage stochastic matching

We state the two-stage stochastic matching problem formally.

Instance: Graph $G = (V,E)$, for each $e \in E$, first-stage edge weights c_e and second-stage edge weights d_e^s for $s = 1,\ldots,r$, and probability p_s for scenario s, a positive integer number r, and a positive real number k.

Question: Are there matchings $M^0, M^1,\ldots,M^r$ in the graph G such that for $s = 1,\ldots,r$, $M^0 \cap M^s = \phi$, $M^0 \cup M^s$ is a matching, and the total expected edge weight given by

$$\sum_{e\in M^0} c_e + \sum_{s=1}^{r} p_s \sum_{e\in M^s} d_e^s \tag{3}$$

is at least k?

Theorem 1. *Two-stage stochastic matching is $\mathcal{NP}$-complete.*

Aboudi (1986) studied a similar problem, constrained matching, and demonstrated that it is also $\mathcal{NP}$-complete with a somewhat similar proof.

Proof. *Two-stage stochastic matching* is clearly in $\mathcal{NP}$. We assume that for all s, $p_s > 0$, since any scenario with $p_s = 0$ may be eliminated. We will use a reduction from *CNF-satisfiability* to establish the theorem. Let C be an expression in conjunctive normal form with ρ clauses: $C = C_1 \wedge C_2 \wedge \cdots \wedge C_p$ and q literals $x_1, x_2,\ldots,x_q$. We assume that x_i and $\bar{x}_i$ do not appear in the same clause, since each clause is a disjunction and thus any clause containing both x_i and $\bar{x}_i$ is always satisfied. We construct the graph G as follows:

For each x_i, create vertices v_i, w_i, and $\bar{w}_i$. For each v_i, construct edges (v_i, w_i) and $(v_i, \bar{w}_i)$. For each such edge e, let $c_e = 1$, and let $d_e^s = 0$ for $s = 1,\ldots,\rho$. For each C_s, create a vertex u_s. For $i = 1,\ldots,q$, construct edges (w_i, u_s), and $(\bar{w}_i, u_s)$. For each edge $e = (w_i, u_s)$, let $c_e = 0$, and if x_i is in C_s, let $d_e^s = \rho$, otherwise, $d_e^s = 0$. For each edge $e = (\bar{w}_i, u_s)$, let $c_e = 0$, and if x_i is in C_s, let $d_e^s = \rho$, otherwise, $d_e^s = 0$. Define $r \equiv \rho$ and $k \equiv \rho + q$. For $s = 1,\ldots,r$, let $p_s = \frac{1}{r}$. Note that G is a bipartite graph with bipartition $(V + U, W)$, where

V, U, W are the sets containing all vertices v_i, u_s, and w_i and $\bar{w}_i$, respectively.

We now claim that G contains matchings $M^0 \cup M^s$ for $s = 1, \ldots, r$, and the total expected edge weight given in (3) is at least k if and only if the expression C is satisfiable. To see this we demonstrate the correspondence between matchings with which the value of (3) is at least k and a literal assignment which satisfies C.

Suppose that there exists a literal assignment that satisfies C. Construct the matchings $M^0, M^1, \ldots, M^r$ as follows.

1. For all x_i, if x_i is `True`, add (v_i, w_i) to M^0.
2. For all x_i, if x_i is `False`, add $(v_i, \bar{w}_i)$ to M^0.
3. For all clauses C_s, pick any literal that satisfies C_s. If x_i is chosen, add $(u_s, \bar{w}_i)$ to M^s. If $\bar{x}_i$, is chosen, add (u_s, w_i) to M^s.

It is easy to check that for $s = 1, \ldots, r$, $M^0 \cap M^s = \emptyset$ and $M^0 \cup M^s$ is a matching, and the total expected edge weight is k.

Now let us suppose that there exist matchings $M^0, M^1, \ldots, M^r$ such that for $s = 1, \ldots, r$, $M^0 \cap M^s = \emptyset$ and $M^0 \cup M^s$ is a matching, and the total expected edge weight is at least k. Note that no more than q edges with $c_e > 0$ can be in M^0, and $c_e = 1$ for all such edges. Also, note that for each s, no more than one edge with $d_e^s > 0$ can be in M^s, and $d_e^s = \rho$ for this edge. Hence, $\sum_{e \in M^0} c_e \leqslant q$ and $\sum_{e \in M^s} p_s d_e^s \leqslant 1$. The latter inequality implies that $\sum_{s=1}^{r} \sum_{e \in M^s} p_s d_e^s \leqslant \rho$ and thus the value of (3) is at most $q + \rho = k$. Since the total expected edge weight is at least k, it follows that M^0 matches every vertex in V with a weight 1 edge and each M^s matches vertex u_s with a positively weighted edge.

Consider any literal in C, we construct the literal assignment as follows.

1. If (v_i, w_i) is in M^0, x_i is `True`.
2. If $(v_i, \bar{w}_i)$ is in M^0, x_i is `False`.

It is easy to check that this literal assignment satisfies C.

The above transformation is clearly polynomial, so we conclude that *Two-stage stochastic matching* is $\mathcal{NP}$-complete. □

Table 1
Edge weights of the graph constructed from C

e	c_e	d_e^1	d_e^2
(v_1, w_1)	1	0	0
$(v_1, \bar{w}_1)$	1	0	0
(v_2, w_2)	1	0	0
$(v_2, \bar{w}_2)$	1	0	0
(w_1, u_1)	0	2	0
$(\bar{w}_1, u_1)$	0	0	0
(w_2, u_1)	0	0	0
$(\bar{w}_2, u_1)$	0	2	0
(w_1, u_2)	0	0	2
$(\bar{w}_1, u_2)$	0	0	0
(w_2, u_2)	0	0	2
$(\bar{w}_2, u_2)$	0	0	0

2.1. Example of the reduction

Consider the expression

$$C = \{\bar{x}_1 \vee x_2\} \wedge \{\bar{x}_1 \vee \bar{x}_2\}. \tag{4}$$

There are two literals and two clauses, so $r = \rho = 2$, $q = 2$ and $k = 4$. Then G is as in Fig. 1 and the edge weights are as in Table 1. All edge weights are also labeled in Fig. 1. The two scenarios are assigned with equal probability.

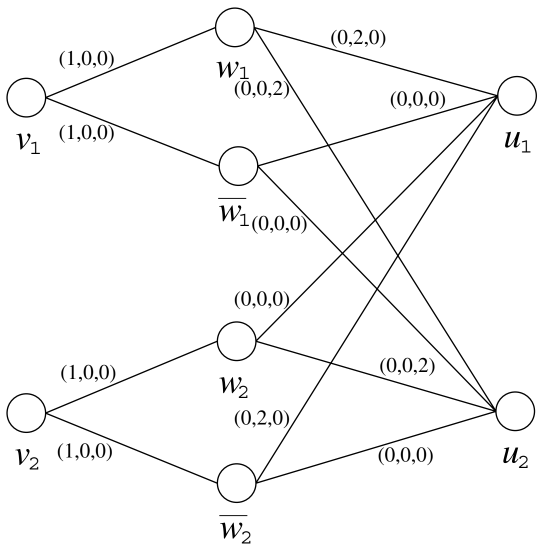

Fig. 1. Example of the graph G constructed from the expression C given in (4). The edge weights are represented by $(c_e, d_e^1, \ldots, d_e^r)$.

From Theorem 1, there exist matchings M^0, M^1 and M^2 such that $M^0 \cap M^1 = \emptyset$, $M^0 \cap M^2 = \emptyset$, and $M^0 \cup M^1$ and $M^0 \cup M^2$ are matchings and

$$\sum_{e\in M^0} c_e + \frac{1}{2}\sum_{e\in M^1} d_e^1 + \frac{1}{2}\sum_{e\in M^2} d_e^2 \geqslant 4$$

if and only if there exists a literal assignment satisfying C.

Matchings $M^0 = \{(v_1, \bar{w}_1), (v_2, \bar{w}_2)\}$, $M^1 = \{(w_1, u_1)\}$ and $M^2 = \{(w_2, u_2)\}$ have a total expected weight of $(1 + 1 + 2 \times \frac{1}{2} + 2 \times \frac{1}{2} = 4)$, and these matchings correspond to the assignment of literal x_1 to `False` and literal x_2 to `False` which satisfies C. Note that $M^0 \cup M^1$ and $M^0 \cup M^2$ are matchings. An alternative is matchings $M^0 = \{(v_1, \bar{w}_1), (v_2, w_2)\}$, $M^1 = \{(w_1, u_1)\}$, and $M^2 = \{(\bar{w}_2, u_2)\}$, which also have a total expected weight of 4, and correspond to the assignment of literal x_1 to `False` and literal x_2 to `True` which satisfies C as well.

3. A factor $\frac{1}{2}$ approximation algorithm

Definition 1. A *first-stage myopic solution* is an optimal solution to:

$$(\text{MYOPIC1}) : \max\left\{\sum_{e\in E} c_e x_e \,\middle|\, \sum_{e\in\delta(v)} x_e \leqslant 1,\ \forall v \in V;\ x_e \in \{0,1\},\ \forall e \in E\right\}.$$

Definition 2. A *second-stage myopic solution* for scenario s is an optimal solution to:

$$(\text{MYOPIC2}) : \max\left\{\sum_{e\in E} d_e^s y_e \,\middle|\, \sum_{e\in\delta(v)} y_e \leqslant 1,\ \forall v \in V;\ y_e \in \{0,1\},\ \forall e \in E\right\}.$$

A first- (second-) stage myopic solution is the solution to a deterministic maximum-weight matching problem with the appropriate choice of objective.

The intuition behind our approximation algorithm is straightforward. We consider $r + 1$ solutions: one first-stage myopic solution, and r second-stage myopic solutions for all scenarios. We compare two objective values: the objective value of the first-stage myopic solution, and the expected objective value of the second-stage myopic solutions over all scenarios. Of these two values, the larger one gives the output of the approximation algorithm.

We state the algorithm formally:

Algorithm 1

```
INPUT: A two-stage stochastic maximum-weight matching problem.
Let $x_1$ be a first-stage myopic solution, and let $z_1 = cx_1$.
For scenario $s$, Let $y_2^s$ be a second-stage myopic solution, and let $z_2^s = d^s y_2^s$.
Let $\hat{z} = \max\{z_1, \sum_{s=1}^r p_s z_2^s\}$.
OUTPUT: If $\hat{z} = z_1$, then return $(x_1, \mathbf{0}, \ldots, \mathbf{0})$ and $z_1$; otherwise, return $(\mathbf{0}, y_2^1, \ldots, y_2^r)$ and $\sum_{s=1}^r p_s z_2^s$.
```

Theorem 2. *Algorithm* 1 *is an approximation algorithm with performance guarantee* $\frac{1}{2}$ *for the two-stage stochastic maximum-weight matching problem given in* (2).

Proof. Solutions $(x_1, \mathbf{0}, \ldots, \mathbf{0})$ and $(\mathbf{0}, y_2^1, \ldots, y_2^r)$ are clearly feasible to (2). Let $x^* = (x_0, y_0^1, \ldots, y_0^r)$ and z^* be an optimal solution and the optimal objective value to (2), respectively. Since solution x_0 is feasible to (MYOPIC1),

$$\hat{z} \geqslant z_1 \geqslant cx_0. \tag{5}$$

Since solution y_0^s is feasible to (MYOPIC2) for scenario s, $s = 1, \ldots, r$,

$$\hat{z} \geqslant \sum_{s=1}^r p_s z_2^s \geqslant \sum_{s=1}^r p_s d^s y_0^s. \tag{6}$$

Summing up inequalities (5) and (6) yields $2\hat{z} \geqslant cx_0 + \sum_{s=1}^r p_s d^s y_0^s = z^*$, and thus the result follows. □

Since both (MYOPIC1) and (MYOPIC2) are polynomially solvable, Algorithm 1 runs in polynomial time.

3.1. A tight example for Algorithm 1

We give a tight example of two-stage stochastic bipartite matching. The problem is defined on the graph $G = (V, E)$ as in Fig. 2 and its objective function is given as in (2).

Let G be a bipartite graph with bipartition $V = (S, T)$ where $|S| = |T|$ and furthermore let $S = (S_1, S_2)$ with $|S_1| = |S_2|$ and let $T = (T_1, T_2)$ with $|T_1| = |T_2|$. Let l be any positive integer. For all edges $e = (u, v)$ with $u \in S_1$ and $v \in T_1$, let $c_e = l$. For all other edges connecting S and T, let $c_e = 0$. For any scenario s, $s = 1, \ldots, r$, for all edges $e = (u, v)$ with $u \in S_2$ and $v \in T_2$ let $d_e^s = l$; for all other edges connecting S and T, let $d_e^s = 0$. Fig. 2 illustrates such an instance with $|S| = |T| = 4$.

The first-stage myopic solution is given by choosing any complete matching from S_1 to T_1, together with any matching from S_2 to T_2. Hence, the output of Algorithm 1 is to use the first-stage myopic solution in the first stage and to choose no edges in the second stage. This output gives the total expected edge weight $l \cdot \frac{|V|}{4}$. For any scenario, the second-stage myopic solution is given by choosing any complete matching from S_2 to T_2, together with any matching from S_1 to T_1. Hence, the output of Algorithm 1 is to choose no edges in the first stage and to use the sth second-stage myopic solution in the second stage if scenarios s is realized. This output gives the total expected edge weight $l \cdot \frac{|V|}{4}$. The maximum of these two solutions is $l \cdot \frac{|V|}{4}$, so the approximation algorithm gives a solution of $l \cdot \frac{|V|}{4}$.

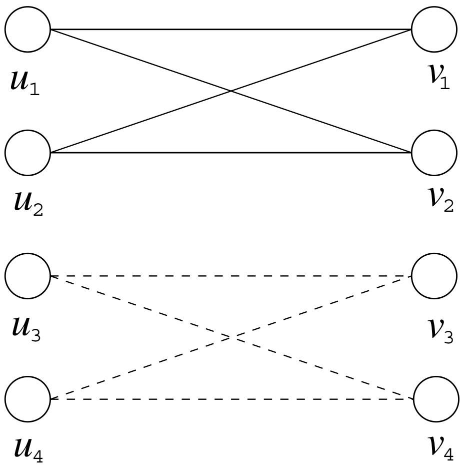

Fig. 2. Example on the complete bipartite graph G when the bound is tight. Edges with presence have edge weights. Solid edges are weighted l in the first stage and dashed edges are weighted l in the second stage. All other edges have zero weight in both stages. $S_1 = \{u_1, u_2\}$, $S_1 = \{u_3, u_4\}$, $T_1 = \{v_1, v_2\}$, $T_1 = \{v_e, v_4\}$.

The optimal solution to the two-stage stochastic bipartite matching problem is to choose any complete matching from S_1 to T_1 as the first-stage decision, and for each scenario, choose any complete matching from S_2 to T_2 as the second-stage decision. The first-stage matching gives edge weight $l \cdot \frac{|V|}{4}$, and the expected second-stage edge weight is $l \cdot \frac{|V|}{4}$, so the total expected edge weight is $l \cdot \frac{|V|}{2}$. Thus the approximation algorithm returns a solution whose total expected edge weight is exactly $\frac{1}{2}$ of the optimal objective value.

4. Computational results

We tested our approximation algorithm on a set of randomly generated two-stage stochastic bipartite matching instances with 10 vertices in each side of the bipartition and 100 scenarios. In our computational experiments, we used CPLEX 7.0 to find the solutions to (MYOPIC1) and (MYOPIC2). To check the performance of our approximation algorithm, we also solved the stochastic programming formulation directly using the L-shaped method (Van Slyke and Wets, 1969), a variant of Benders' decomposition (Benders, 1962) and a standard technique for exactly solving two-stage stochastic linear programs with continuous or integer first stage. For some large instances, the L-shaped method tended to be very time-consuming, so we imposed a 1-hour CPU time limit on it and obtained the solution of the restricted master problem, which is an upper bound on the exact solution.

In all test instances, the first-stage and second-stage edge weights were normally distributed. All scenarios were realized with equal probability. We tested four groups of instance classes, in each of which only one of the four distribution parameters (mean and standard deviation of the first-stage and second-stage edge weights) was varied and other three were fixed. For example, in group 1, we varied the mean of the first-stage edge weights from 5 to 25. We generated 100 instances for each instance class and reported the average CPU time. Table 2 presents the characteristics and computational results of these instance classes. Our computational experiments indicate that the CPU time of the approximation algorithm is insensitive to the distribution parameter settings. On the other hand, when the first-stage and second-stage edge weights are generated from the same or similar distributions, the L-shaped method is relatively less efficient due to the symmetry between the edge weights in the two stages. We also report the average ratio of the approximation solution to the exact solution in the table. When the first-stage and second-stage edge weights are generated from the same or similar distributions, this ratio tends to be the lowest for the same reason. The last column in Table 2 shows the integrality gap of the LP-relaxation for these instances.

Table 2
Characteristics and computational results of small instances

Group	Instance		Average approx. CPU	Average L-shaped CPU	Average # of L-shaped iterations	Average performance ratio	Average integrality gap (%)
	First stage	Second stage					
1	$\sim N(5, 15^2)$	$\sim N(10, 15^2)$	0.08	0.71	13.39	0.976	0.1
	$\sim N(10, 15^2)$	$\sim N(10, 15^2)$	0.08	1.88	25.35	0.958	0.3
	$\sim N(15, 15^2)$	$\sim N(10, 15^2)$	0.08	2.25	23.70	0.984	0.3
	$\sim N(20, 15^2)$	$\sim N(10, 15^2)$	0.08	0.41	9.05	0.998	0.0
	$\sim N(25, 15^2)$	$\sim N(10, 15^2)$	0.08	0.14	4.79	1.000	0.0
2	$\sim N(10, 5^2)$	$\sim N(10, 15^2)$	0.08	0.10	3.01	1.000	0.0
	$\sim N(10, 10^2)$	$\sim N(10, 15^2)$	0.08	0.34	7.20	0.997	0.0
	$\sim N(10, 15^2)$	$\sim N(10, 15^2)$	0.08	1.88	25.35	0.958	0.3
	$\sim N(10, 20^2)$	$\sim N(10, 15^2)$	0.08	1.21	18.93	0.966	0.2
	$\sim N(10, 25^2)$	$\sim N(10, 15^2)$	0.08	0.53	11.54	0.978	0.0
3	$\sim N(10, 15^2)$	$\sim N(5, 15^2)$	0.08	1.81	21.97	0.980	0.1
	$\sim N(10, 15^2)$	$\sim N(10, 15^2)$	0.08	1.88	25.35	0.958	0.3
	$\sim N(10, 15^2)$	$\sim N(15, 15^2)$	0.08	0.79	14.02	0.980	0.1
	$\sim N(10, 15^2)$	$\sim N(20, 15^2)$	0.08	0.32	7.20	0.994	0.0
	$\sim N(10, 15^2)$	$\sim N(25, 15^2)$	0.08	0.15	4.11	0.998	0.0
4	$\sim N(10, 15^2)$	$\sim N(10, 5^2)$	0.08	0.16	5.74	0.989	0.0
	$\sim N(10, 15^2)$	$\sim N(10, 10^2)$	0.08	0.82	15.20	0.976	0.1
	$\sim N(10, 15^2)$	$\sim N(10, 15^2)$	0.08	1.88	25.35	0.958	0.3
	$\sim N(10, 15^2)$	$\sim N(10, 20^2)$	0.09	0.97	12.66	0.991	0.0
	$\sim N(10, 15^2)$	$\sim N(10, 25^2)$	0.09	0.19	4.86	0.999	0.0

Table 3
Characteristics and computational results of larger instances

Instance class	Weights		Average approx. CPU	Average L-shaped CPU	Average # of L-shaped iterations	Average performance ratio
	First stage	Second stage				
1	$\sim N(10, 15^2)$	$\sim N(10, 15^2)$	160.4	$\geqslant 3600$	$\geqslant 3$	0.954
2	$\sim N(10, 15^2)$	$\sim N(10, 20^2)$	144.9	$\geqslant 3600$	$\geqslant 3$	0.998
3	$\sim N(10, 15^2)$	$\sim N(10, 30^2)$	131.8	197.0	2	1.000
4	$\sim N(15, 15^2)$	$\sim N(10, 15^2)$	161.3	$\geqslant 3600$	$\geqslant 3$	0.986
5	$\sim N(20, 15^2)$	$\sim N(10, 15^2)$	159.7	$\geqslant 3600$	$\geqslant 3$	0.997
6	$\sim N(20, 15^2)$	$\sim N(10, 30^2)$	132.3	191.4	2	1.000

We then considered some large stochastic bipartite instances with 500 vertices in each side of the bipartition and 10 scenarios. Table 3 presents the characteristics and computational results of these instances. Each instance class consists of 10 instances. As above, each random parameter was generated according to a normal distribution and each scenario was assigned with equal probability. In the table, we also report the average ratio of the approximate solution to the exact solution or its upper bound if the L-shaped method did not terminate within 1 hour.

5. Conclusions

As we have shown in this paper, the stochastic programming extension of a polynomially solvable combinatorial optimization problem may become $\mathcal{NP}$-complete. However, the line between easy and hard stochastic combinatorial optimization problems has yet to be fully explored. Meanwhile, given difficulty of solving stochastic programs, particularly stochastic integer programs, developing approximation algorithms for such problems is a promising direction for future research.

Acknowledgment

This research was partially supported by National Science Foundation grant DMI-0217190 and a grant from the University of Pittsburgh Central Research Development Fund.

References

Aboudi, R., 1986. A constrained matching program: A polyhedral approach. Ph.D. Thesis, Cornell University, Ithaca, NY.

Benders, J.F., 1962. Partitioning procedures for solving mixed variables programming problems. Numerische Mathematik 4, 238–252.

Birge, J.R., Louveaux, F.V., 1997. Introduction to Stochastic Programming. Springer, New York.

Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A., 1998. Combinatorial Optimization. John Wiley and Sons, New York.

Dye, S., Stougie, L., Tomasgard, A., 2003. Approximation algorithm and relaxations for a service provision problem on a telecommunication network. Discrete Applied Mathematics 129 (1), 63–81.

Edmonds, J., 1965. Maximum matching and a polyhedron with 0–1 vertices. Journal of Research of the National Bureau of Standards 69B, 125–130.

Hochbaum, D.S., 1997. Approximation Algorithms for $\mathcal{NP}$-Hard Problems. PWS Publishing Company, Boston, MA.

Kall, P., Wallace, S.W., 1994. Stochastic Programming. John Wiley and Sons, Chichester, UK.

Van Slyke, R., Wets, R.J.-B., 1969. L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics 17, 638–663.

Vazirani, V., 2001. Approximation Algorithms. Springer-Verlag, New York.