European Journal of Operational Research 198 (2009) 983–986

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Short Communication

A note on Liu–Iwamura's dependent-chance programming

Roberto Rossi [a,d,*], S. Armagan Tarim [b], Brahim Hnich [c], Steven Prestwich [d], Cahit Guran [e]

[a] Centre for Telecommunication, Value-Chain Research, Ireland
[b] Department of Management, Hacettepe University, Ankara, Turkey
[c] Faculty of Computer Science, Izmir University of Economics, Izmir, Turkey
[d] Cork Constraint Computation Centre, University College, 14th Washington Street West, Cork, Ireland
[e] Department of Public Finance, Hacettepe University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 5 February 2008
Accepted 4 November 2008
Available online 14 November 2008

Keywords:
Inventory control
Integer programming
Constraint programming
Demand uncertainty

ABSTRACT

Sometimes a complex stochastic decision system undertakes multiple tasks called events, and the decision-maker wishes to maximize the chance functions which are defined as the probabilities of satisfying these events. Originally introduced by Liu and Iwamura [B. Liu, K. Iwamura, Modelling stochastic decision systems using dependent-chance programming, European Journal of Operational Research 101 (1997) 193–203], dependent-chance programming is aimed at maximizing some chance functions of events in an uncertain environment. In this work, we show that the original dependent chance-programming framework needs to be extended in order to capture an exact reliability measure for a given plan.

1. Introduction

Chance-constrained programming, pioneered by Charnes and Cooper [1], provides a means of handling uncertainty by specifying a confidence level at which it is desired that the stochastic constraint holds. Chance-constrained programming models can be converted into deterministic equivalents only for some special cases, and then solved by using solution methods of deterministic mathematical programming. In order to overcome this difficulty, Liu [4] provided a new stochastic programming framework, called *dependent-chance programming*, in which a complex stochastic decision system undertakes multiple tasks called events, and the decision-maker wishes to maximize the chance functions which are defined as the probabilities of satisfying these events. Liu and Iwamura [6] proposed a stochastic simulation-based genetic algorithm for solving general chance-constrained programming as well as chance-constrained multi-objective programming, and chance-constrained goal programming (for a more detailed discussion see [5]).

Roughly speaking, dependent-chance programming is aimed at maximizing some chance functions of events in an uncertain environment. In deterministic mathematical programming the feasible set is essentially assumed to be deterministic and the optimal solution can always be implemented. However when uncertainty is taken into account the given solution may be infeasible if the realization of uncertain parameters is unfavorable. In other words, the feasible set of dependent chance-programming is described by a so-called *uncertain environment*. Although a deterministic solution is given by the dependent chance-programming model, this solution needs to be as flexible as possible with respect to the uncertain environment. This special feature of dependent chance-programming is very different from other existing stochastic programming frameworks. However, such problems do exist in the real world. Some applications of dependent chance programming have been presented by Liu and Ku [7], Liu [2,3], Liu and Iwamura [6], and more recently by Wu et al. [8].

In this note, we argue that the original dependent chance-programming framework proposed by Liu and Iwamura needs to be extended in order to capture an exact notion of reliability and we show that the way Liu and Iwamura express constraint dependencies, without taking into account the values assigned to decision variables, does not guarantee optimal plans since in certain instances common variables may take values which break the link between dependent constraints.

This paper is organized as follows: In Section 2, we recall the dependent-chance programming framework proposed by Liu and Iwamura. In Section 3, we describe a motivational water supply-allocation problem originally proposed in [4] and we analyze the reliability of different distribution plans according to their framework. In Section 4, we propose an exact notion of reliability obtained by expressing constraint dependencies taking into account

* Corresponding author. Address: Cork Constraint Computation Centre, University College, 14th Washington Street West, Cork, Ireland. Tel.: +353 (0)85 122 3582; fax: +353 (0)21 425 5424.
E-mail addresses: rrossi@4c.ucc.ie (R. Rossi), armagan.tarim@hacettepe.edu.tr (S. Armagan Tarim), brahim.hnich@ieu.edu.tr (B. Hnich), s.prestwich@4c.ucc.ie (S. Prestwich), guran@hacettepe.edu.tr (C. Guran).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.11.004

the values assigned to decision variables. An exact reliability measure is then proposed for the distribution plans being analyzed. In Section 5, we draw conclusions.

2. Formal background

This section presents a summary of dependent-chance programming of Liu [2,3] and underlying concepts.

If Ω is a collection of objects denoted generally by x, then the *stochastic set A* in Ω is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in \Omega\},$$

where $\mu_A(x)$ is called the probability function of x in A. In uncertain environments, the feasible set, represented by a series of stochastic constraints, may be described by a stochastic set. In contrast to the deterministic case, we cannot say a point is feasible or not when our problem is defined on a stochastic set. We have to say a point x^* is feasible with probability α, where α is the value of probability function $\mu_A(x^*)$.

Usually, a solution x is a vector composed of n components, $x_1, x_2, \dots, x_n$. We will suppose that we know the following relationship among the decision components.

Stochastic relationship: there is a known partition of n components of a decision vector into k groups such that these k groups are mutually stochastically independent and in each group any elements are stochastically dependent and have the same chance to appear if they require to be realized simultaneously.

Thus, in stochastic decision systems, the feasible set of decision vectors is represented by a stochastic set, say S, whose probability function is $\mu_S(x)$.

Next we consider the purpose of our system. Usually there are multiple purposes, functions or tasks of a complex system. Liu denotes the actions meeting the purposes or performing the tasks as *events*. Each event is represented by a set E which is composed of all the possible decisions meeting certain conditions. Let $V(E)$ denote the set of all components of x which are necessary to the event E and $D(E)$ be the set of all components which are stochastically dependent of any elements in $V(E)$. It is clear that $V(E) \subset D(E)$.

For each element of an event E, we have to give an evaluation, i.e. criterion function, of a decision vector. In view of the uncertainty of the stochastic decision system, we are not certain whether a decision is feasible before knowing the realization of stochastic parameters, so we employ *chance functions* as objective functions to evaluate some of the events. Generally, the chance function, denoted by $f(x)$, is the probability function on the event E.

Thus, for single event case, the dependent-chance programming (DCP) is given as follows:

$$\max_{x \in S} f(x), \tag{1}$$

where x is an n-dimensional decision vector, S is a stochastic set on $\mathbb{R}^n$ with probability function $\mu_S(x)$, $f(x)$ is a chance function of a certain event, borrowing the symbol $\in$ from classical set theory, $x \in S$ means x is feasible with probability $\mu_S(x)$. A point $x^* \in S$ is called an optimal solution of the problem in Eq. (1) if $f(x^*) \geqslant f(x)$ for any $x \in S$.

As an extension, the dependent-chance multiobjective programming (DCMOP) for multiple events case is given as follows,

$$\max_{x \in S} f(x) = [f_1(x), f_2(x), \dots, f_m(x)], \tag{2}$$

where $f(x)$ is a vector of real-valued functions f_i which are chance or deterministic functions.

In Liu and Iwamura [6] the authors highlight that the key aspect of algorithm for solving DCP, DCMOP and DCGP (i.e. dependent-chance goal programs, for a detailed discussion refer to [6]) consists in constructing the relationship between the decision vectors and chance functions. They consider a set of t objectives $f_i(x), i = 1, 2, \dots, t$. They assume that every $f_i(x)$ is a chance function that represents a probability of a certain event which is represented by E_i. Then they define

$$E = E_1 \cap E_2 \cap \cdots \cap E_t,$$

and

$$V(E) = V(E_1) \cup V(E_2) \cup \cdots \cup V(E_t).$$

In order to realize each event E_i, as far as possible without sacrificing the chances of other events, they treat all elements in the stochastically dependent set $D(E_i)$ of $V(E_i)$ at an equitable level, i.e., these elements would have the same chance to be realized. On the other hand they disregard elements out of $V(E)$ because they do not make any contribution to the events that have to be realized. Thus the authors consider all the elements in and only in $D(E_i) \cap V(E)$ simultaneously for the event E_i. From the stochastic relationship it follows that all the elements in $D(E_i) \cap V(E)$ are independent of any other elements in $V(E)$, therefore we can perform the elements in $D(E_i) \cap V(E)$ as far as possible.

It has to be noted that the relationship between the decision vectors and chance functions is defined by the authors in [6] without taking into account the values assigned to decision variables. For this reason we shall see that their definition does not guarantee optimal plans, since in certain instances common variables may take values which break the link between two dependent constraints. In order to show this, in the following section we recall the water supply-allocation problem presented in Liu and Iwamura [6] to demonstrate the subtleties inherent in dependent-chance programming.

3. A dependent-chance programming example

Fig. 1 depicts a water supply system with three suppliers S_1, S_2, S_3 with their given *probabilistic* supply capacities and three different customers, denoted by C_1, C_2, C_3, with known demands. The scopes of the suppliers are $S_1 \rightsquigarrow \{C_1, C_2\}, S_2 \rightsquigarrow \{C_1, C_2, C_3\}, S_3 \rightsquigarrow \{C_2, C_3\}$.

The deterministic customer demands are [8,7,4]. The suppliers' probabilistic capacities are expressed as discrete probability density functions:

$$\phi_{S_1} = \{3(0.3), 7(0.5), 12(0.2)\},$$
$$\phi_{S_2} = \{6(0.4), 7(0.2), 10(0.4)\},$$
$$\phi_{S_3} = \{3(0.3), 8(0.7)\},$$

where values in parentheses represent probabilities. We must answer the following two types of question.

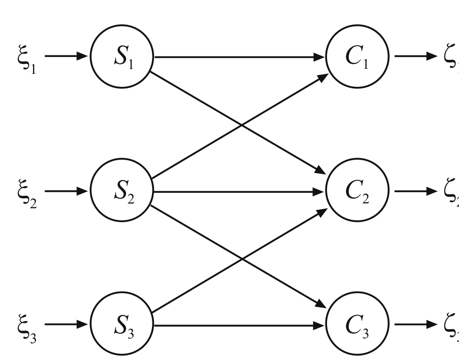

Fig. 1. Water supply-allocation problem.

- Supply problems. In order to achieve certain objectives in the future, decisions must be made concerning present actions to be taken. That is, we must determine the optimal combination of *inputs*, for example to determine the quantities ordered from the 3 inputs.
- Allocation problems. One of the basic allocation problems is the optimal allocation of the resources. Here the task is to determine the *outputs* that result from various combinations of resources such that certain objectives are achieved.

Certainly, in this system supply and allocation decisions should not be separate.

Let $\mathbf{S}$ be the set of suppliers and $\mathbf{C}$ the set of customers. Define decision variables $x_{s,c} \in \mathbb{Z}^+ \bigcup \{0\}$ denoting the planned non-negative supply from supplier s to customer c. Also define random variables ξ_s, with probability density function ϕ_s, denoting the uncertain supply available to supplier s. First we have the following constraints

$$\sum_{c\in\mathbf{C}_s} x_{s,c} \leqslant \xi_s, \quad \forall s \in \mathbf{S},$$

where $\mathbf{C}_s$ is the set of customers for supplier s. A constant ζ_c denotes the deterministic demand of customer c. Event E_c is defined as follows:

$$E_c : \sum_{s\in\mathbf{S}_c} x_{s,c} = \zeta_c,$$

where $\mathbf{S}_c$ is the set of suppliers for customer c. Event E_c means that the decision should satisfy the demand of customer c. In view of the uncertainty of this system, we are not sure whether a decision is feasible before knowing the realization of stochastic variables, so we employ chance functions to evaluate these events. Let

$$f_c(x) = \Pr\left\{E_c : \sum_{s\in\mathbf{S}_c} x_{s,c} = \zeta_c\right\},$$

where Pr denotes the probability of the event in $\{\cdot\}$. Usually we hope to maximize all the chance functions, i.e. increase the reliability levels of all the events as much as possible.

Without loss of generality we will now assume that all the events have the same priority and we will formulate the problem as DCGP. The model is therefore

$$\max \quad \sum_{c\in\mathbf{C}} f_c(x) \tag{3}$$

subject to, (4)

$$\sum_{c\in\mathbf{C}_s} x_{s,c} \leqslant \xi_s \quad s\in\mathbf{S}, \tag{5}$$

$$x_{s,c} \in \mathbb{Z}^+ \bigcup \{0\} \quad s\in\mathbf{S},\ c\in\mathbf{C}. \tag{6}$$

The stochastic feasible set S will be defined by a probability function

$$\mu_S(x) = \Pr\left\{\sum_{c\in\mathbf{C}_s} x_{s,c} \leqslant \xi_s, \quad \forall s\in\mathbf{S}\right\}. \tag{7}$$

The authors in Liu and Iwamura [6] divide the decision components into three groups $\{x_{s,c}|s = S_1\}$, $\{x_{s,c}|s = S_2\}$ and $\{x_{s,c}|s = S_3\}$ which are mutually stochastically independent and in each group any element has the same probability of occurring. From the water supply-allocation problem definition it follows that

$$V(E_1) = \{x_{S_1,C_1}, x_{S_2,C_1}\}, \tag{8}$$

$$V(E_2) = \{x_{S_1,C_2}, x_{S_2,C_2}, x_{S_3,C_2}\}, \tag{9}$$

$$V(E_3) = \{x_{S_2,C_3}, x_{S_3,C_3}\}, \tag{10}$$

and

$$D(E_1) = \{x_{S_1,C_1}, x_{S_1,C_2}, x_{S_2,C_1}, x_{S_2,C_2}, x_{S_2,C_3}\}, \tag{11}$$

$$D(E_2) = \{x_{S_1,C_1}, x_{S_1,C_2}, x_{S_2,C_1}, x_{S_2,C_2}, x_{S_2,C_3}, x_{S_3,C_2}, x_{S_3,C_3}\}, \tag{12}$$

$$D(E_3) = \{x_{S_2,C_1}, x_{S_2,C_2}, x_{S_2,C_3}, x_{S_3,C_2}, x_{S_3,C_3}\}, \tag{13}$$

therefore the induced constraint on $D(E_1) \cap V(E)$ is then, according to Liu and Iwamura, $\{x_{S_1,C_1} + x_{S_1,C_2} \leqslant \xi_{S_1}, x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} \leqslant \xi_{S_2}\}$; on $D(E_2) \cap V(E)$ it is $\{x_{S_1,C_1} + x_{S_1,C_2} \leqslant \xi_{S_1}, x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} \leqslant \xi_{S_2}, x_{S_3,C_2} + x_{S_3,C_3} \leqslant \xi_{S_3}\}$; and finally on $D(E_3) \cap V(E)$ it is $\{x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} \leqslant \xi_{S_2}, x_{S_3,C_2} + x_{S_3,C_3} \leqslant \xi_{S_3}\}$. Hence

$$f_{C_1}(x) = \Pr\{(\xi_{S_1}, \xi_{S_2}) | x_{S_1,C_1} + x_{S_1,C_2} \leqslant \xi_{S_1}, x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} \leqslant \xi_{S_2}\}, \tag{14}$$

$$f_{C_2}(x) = \Pr\{(\xi_{S_1}, \xi_{S_2}, \xi_{S_3}) | x_{S_1,C_1} + x_{S_1,C_2} \leqslant \xi_{S_1}, x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} \leqslant \xi_{S_2}, x_{S_3,C_2} + x_{S_3,C_3} \leqslant \xi_{S_3}\}, \tag{15}$$

$$f_{C_3}(x) = \Pr\{(\xi_{S_2}, \xi_{S_3}) | x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} \leqslant \xi_{S_2}, x_{S_3,C_2} + x_{S_3,C_3} \leqslant \xi_{S_3}\}. \tag{16}$$

Table 1 presents some representative distribution plans (columns 2–8) and their corresponding reliability measures according to Liu and Iwamura (column under heading "Liu–Iwamura").

4. Decision variable value based dependency

Liu–Iwamura's framework ignores the important dependency between constraints and values of decision variables.

Consider a plan in which $x_{S_1,C_1} = 0$ so that C_1 must receive all supplies from S_2. The reliability of the satisfaction of C_1 (event E_1) should now be independent of the ability of S_1 to meet its demand. But the dependent-chance programming, in its current form which does not take variable assignments into account, always relates the demand satisfaction of C_1 to S_1 and S_2 (Eq. (14)), which is not necessarily correct. Therefore, one should refine the objectives (Eqs. (14)–(16)) via further logical connectives between constraints:

Table 1
Representative distribution plans.

Plan no.	Planned delivery $S_i \rightsquigarrow D_j : (i,j)$							Reliability measures	
	(1,1)	(1,2)	(2,1)	(2,2)	(2,3)	(3,2)	(3,3)	Liu–Iwamura	New
1	3	5	5	1	1	1	3	0.624	0.624
2	4	7	4	0	4	0	0	0.560	0.680
3	6	2	2	5	0	0	4	0.624	0.940
4	5	0	3	3	4	4	0	0.756	0.960
5	7	5	1	1	1	1	3	1.040	1.040
6	2	5	6	0	4	2	0	0.960	1.380
7	8	2	0	2	4	3	0	1.400	1.400
8	0	7	8	0	0	0	4	0.756	1.800
9	5	0	3	3	0	4	4	1.890	2.100
10	6	0	2	0	4	7	0	1.890	2.400

$$f_c(x) = \Pr\left\{x_{s,c} \neq 0 \rightarrow \sum_{c' \in \mathbf{C}_s} x_{s,c'} \leqslant \xi_s, \quad \forall s \in \mathbf{S}_c\right\}, \tag{17}$$

where $\rightarrow$ denotes logical implication: $\mathfrak{C} \rightarrow \mathfrak{C}'$ is the sum of the probabilities of the scenarios in which either $\mathfrak{C}$ is violated or $\mathfrak{C}'$ is satisfied, or both. Because of this modification, under a decision in which $x_{S_1,C_1} = 0$ there is no longer a penalty if

$$\sum_{c' \in \mathbf{C}_1} x_{s,c'} \leqslant \xi_1$$

is violated.

The new reliability measures calculated using Eq. (17) are listed in the last column in Table 1.

To gain more insight into this problem class we examine allocation plans given in Table 1 in three categories: Plans {1,5}, Plans {2,3,4,6,8,9,10}, and Plan {7}.

In the first category (Plans 1 and 5) the plans have non-zero values assigned to all decision variables and, therefore, as expected the results of Liu–Iwamura and those produced by the extended model proposed here are the same (in Eq. (17), $x_{s,c} \neq 0$ becomes redundant). In the second group, however, since certain variables have zero assignments now a discrepancy between the Liu–Iwamura model and the extended model proposed here is observed. As explained above, this difference in probabilistic measure values is due to the broken constraint dependencies that arise when decision variables are assigned value zero. In the third group, we have only one plan (Plan 7). In this case, although two decision variables are assigned zero values the two frameworks produce the same result. To understand the reason behind this observation we need to look at the amounts committed by suppliers S_2 and S_3 according to Plan 7. Supplier S_2 (S_3) is expected to provide in total $x_{S_2,C_1} + x_{S_2,C_2} + x_{S_2,C_3} = 6$ ($x_{S_3,C_2} + x_{S_3,C_3} = 3$) units. When we look at the uncertain supply capacities for suppliers S_2 and S_3, it is clear that these units can be provided in full even under the worst-case scenarios. In other words, the zero assignment does not make any difference in Plan 7, because breaking the dependency is important only if there is a chance of failure in complying with supply commitments.

5. Conclusion

We showed how to extend Liu and Iwamura's original dependent-chance programming framework in order to obtain an exact reliability measure. Our experiments show that in most cases expressing constraint dependency without taking into account the values assigned to decision variables does not guarantee optimal plans, in fact in certain instances common variables may take values which break the link between two dependent constraints.

Acknowledgements

Roberto Rossi is supported by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of the Centre for Telecommunications Value-Chain-Driven Research (CTVR) and Grant No. 05/IN/I886. S. Armagan Tarim and Brahim Hnich are supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. SOBAG-108K027.

References

[1] A. Charnes, W.W. Cooper, Chance-constrained programming, Management Science 6 (1) (1959) 73–79.
[2] B. Liu, Dependent-chance goal programming: A class of stochastic programming, Technical report, Institute of System Science, Chinese Academy of Sciences, 1995.
[3] B. Liu, Dependent-chance goal programming and its genetic algorithm approach, Technical report, Institute of System Science, Chinese Academy of Sciences, 1995.
[4] B. Liu, Dependent-chance programming: A class of stochastic optimization, Computers & Mathematics with Applications 34 (1997) 89–104.
[5] B. Liu, Uncertain Programming, John Wiley & Sons, New York, 1999.
[6] B. Liu, K. Iwamura, Modelling stochastic decision systems using dependent-chance programming, European Journal of Operational Research 101 (1997) 193–203.
[7] B. Liu, C. Ku, Dependent-chance goal programming and an application, Journal of Systems Engineering & Electronics 4 (1993) 40–47.
[8] Y. Wu, J. Zhou, J. Yang, Dependent-chance programming model for stochastic network bottleneck capacity expansion based on neural network and genetic algorithm, in: Advances in Natural Computation, Lecture Notes in Computer Science, vol. 3612, Springer-Verlag, 2005, pp. 120–128.