Finding reliable solutions

Citation for published version:
Tarim, SA, Hnich, B, Prestwich, S & Rossi, R 2009, 'Finding reliable solutions: Event-driven probabilistic constraint programming', *Annals of Operations Research*, vol. 171, no. 1, pp. 77-99.
https://doi.org/10.1007/s10479-008-0382-6

Digital Object Identifier (DOI):
10.1007/s10479-008-0382-6

Published In:
Annals of Operations Research

Annals of Operations Research manuscript No.
(will be inserted by the editor)

Finding Reliable Solutions: Event-Driven Probabilistic Constraint Programming

S. Armagan Tarim[1], Brahim Hnich[2], Steven Prestwich[3], Roberto Rossi[3,4]

[1] Department of Management, Hacettepe University, Ankara, Turkey, e-mail: armagan.tarim@hacettepe.edu.tr
[2] Faculty of Computer Science, Izmir University of Economics, Izmir, Turkey, e-mail: brahim.hnich@ieu.edu.tr
[3] Cork Constraint Computation Centre, University College, Cork, Ireland, e-mail: s.prestwich@4c.ucc.ie
[4] Centre for Telecommunication Value-Chain Driven Research, Ireland, e-mail: rrossi@4c.ucc.ie

Received: / Revised version:

Abstract Real-life management decisions are usually made in uncertain environments, and decision support systems that ignore this uncertainty are unlikely to provide realistic guidance. We show that previous approaches fail to provide appropriate support for reasoning about reliability under uncertainty. We propose a new framework that addresses this issue by allowing logical dependencies between constraints. Reliability is then defined in terms of key constraints called "events", which are related to other constraints via these dependencies. We illustrate our approach on three problems, contrast it with existing frameworks, and discuss future developments.

Key words Event-Driven – Probabilistic – Constraint Programming – Uncertainty

1 Introduction

Real-life management decisions are usually made in uncertain environments. Random behavior such as the weather, lack of essential exact information such as the future demand, incorrect data due to errors in measurement,

---

Send offprint requests to: Roberto Rossi
Correspondence to: Roberto Rossi, Cork Constraint Computation Centre, University College, 14 Washington St. West, Cork, Ireland., Tel. +353 (0)85 122 3582, Fax. +353 (0)21 425 5424.

and vague or incomplete definitions, exemplifies the theme of uncertainty in such environments.

It is generally impossible for any set of decisions to satisfy all the constraints under all circumstances. For instance, consider a probabilistic single-item distribution problem in which there are n independent suppliers with their given *probabilistic* supply capacities, and m different customers with known demands. It is realistic to assume that the deliveries are fixed in advance, by consideration of the probabilistic supply capacities. The need to fix the deliveries in advance has been at the heart of many problems such as the buying of raw materials on markets with fluctuating prices [15]. Thus the investigation of modeling approaches and solution algorithms is potentially important not only from a theoretical point of view, but also from the perspective of practical applications. It is quite unrealistic to ask for a plan that satisfies all demand and probabilistic supply constraints, irrespective of the unfolding of uncertainties. In order to deal with the optimization problems with stochastic/fuzzy factors, stochastic programming and fuzzy programming have been greatly developed. The theory of stochastic programming has been summarized by several books such as Sengupta [24], Vajda [27], Kall and Wallace [14] etc.

To address this and related situations, we propose that one should determine in advance a distribution plan that satisfies customer demands as far as possible, under some measure that accurately captures the user's notion of reliability. To address this important class of problems, we take a novel approach and develop a modeling framework that supports more reliable decisions in uncertain environments, yet reduces the cognitive burden on a decision-maker. Our *Event-Driven Probabilistic Constraint Programming* (*EDP-CP*) modeling framework allows users to designate certain probabilistic constraints (such as demand constraints) as *events* whose chance of satisfaction must be maximized, subject to hard constraints (such as a lower bound on profit), and also logical dependencies among constraints (such as the dependency of demand constraints on the satisfaction of the probabilistic supply constraints). We shall show that the EDP-CP framework allows more realistic modeling of some problems than previous approaches.

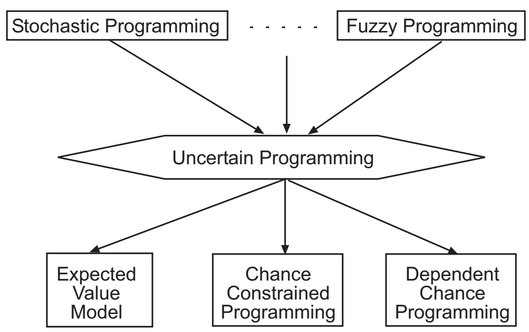

Fig. 1 Techniques for modeling decision problems under uncertainty

Complex decision systems are usually multidimensional, multifaceted, multifunctional and multicriteria, and include stochastic or fuzzy factors. With the requirement of considering randomness, appropriate formulations of stochastic programming have been developed to suit the different purposes of management (Fig. 1). The first method dealing with stochastic parameters in stochastic programming is the so-called *expected value model* [3], which optimizes the expected objective functions subject to some expected constraints. The second, *chance-constrained programming*, was pioneered by Charnes and Cooper [5] as 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 for some special cases, and then solved by some solution methods of deterministic mathematical programming. However it is almost impossible to do this for complex chance-constrained programming models. In order to overcome this dilemma, Liu and Iwamura [20] 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, in which the stochastic simulation is employed to check the feasibility of solutions and to handle the objective functions. 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. In order to model this type of problem, Liu [18] provided a theoretical framework of the third type of stochastic programming, called *dependent-chance programming*. Dependent-chance multiobjective programming and dependent-chance goal programming have also been presented (for a more detailed discussion see [19]).

Roughly speaking, dependent-chance programming is aimed at maximizing some chance functions of events in an uncertain environment. In deterministic mathematical programming as well as expected value models and chance-constrained programming, the feasible set is essentially assumed to be deterministic after the real problem is modeled. That is, an optimal solution is always given regardless of whether is it can be performed in practice. However the given solution may be impossible to perform if the realization of uncertain parameters is unfavorable. Thus, the dependent chance-programming model never assumes that the feasible set is deterministic. In fact, 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 performed as far as possible. This special feature of dependent chance-programming is very different from other existing stochastic programming techniques. However, such problems do exist in the real world. Some real and potential applications of dependent chance programming have been presented by Liu and Ku [21], Liu [16,17], Iwamura and Liu [20], and more recently by Wu et al. [28]. In what follows we will see that the framework we propose, EDP-CP, extends and improves Liu's framework by providing to the user more expressiveness, in order to capture a more realistic and accurate measure of plan reliability, and an exact solution method in contrast to Liu's genetic algorithm.

The rest of this paper is organized as follows. In Section 2 we motivate the work. We define the new modelling framework in Section 3 and show how to compile any EDP-CP model into an equivalent constraint program in Section 4. In Section 5 we survey a scenario reduction technique that may be applied to keep the number of possible scenarios under control, references are given to other works adopting the same strategy to reduce the number of scenarios considered. In Section 6 we illustrate the flexibility and usefulness of our framework by studying three examples: probabilistic supply chain planning, scheduling, and production planning/capital budgeting. In Section 7 we survey related work. Finally, in Section 8 we summarise our work and discuss future directions.

2 Motivation

Our motivation for this work comes from an application in the supply chain management area: more precisely, addressing supply and demand uncertainties. The main inherent difficulty in dealing with this class of probabilistic problems is the fact that certain constraints (such as the ones imposing on complete satisfaction of customer demands) may hinge on the satisfaction of others (such as supply constraints). The problem is particularly interesting when the latter constraints are exposed to uncertainty.

2.1 A motivating example

We provide a concrete example of the distribution problem to motivate the work. Figure 2 depicts a distribution system with three suppliers $S_{1,2,3}$ and three customers $D_{1,2,3}$. The scopes of the suppliers are $S_1 \rightsquigarrow \{D_1, D_2\}$, $S_2 \rightsquigarrow \{D_1, D_2, D_3\}$, $S_3 \rightsquigarrow \{D_2, D_3\}$. The deterministic customer demands are $[8, 7, 4]$. The suppliers' probabilistic capacities are expressed as discrete probability density functions: $f_{S_1} = \{3(0.3), 7(0.5), 12(0.2)\}$, $f_{S_2} = \{6(0.4), 7(0.2), 10(0.4)\}$ and $f_{S_3} = \{3(0.3), 8(0.7)\}$, where values in parentheses represent probabilities. The objective is to obtain the most reliable distribution plan. In the following sections we shall consider a series of models of increasing sophistication. Our running example will emphasize differences between these models.

2.2 Model 1: Naive

Define decision variables $x_{s,c}$ where $s, c \in \{1, 2, 3\}$, denoting the planned supply from supplier s to customer c. Also define random variables ξ_i denoting the uncertain supply available to supplier i. A constant ζ_c denotes

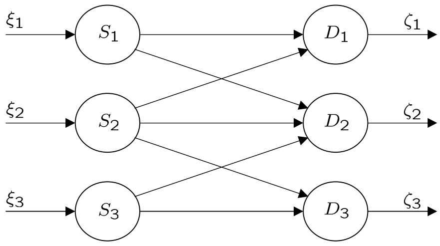

Fig. 2 Distribution Problem

the deterministic demand of customer c. Any plan must satisfy the hard constraints

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

where $\mathbf{S}_c$ is the set of suppliers for customer c. There are also probabilistic constraints between decision and random variables:

$$\sum_{c \in \mathbf{C}_s} x_{s,c} \leq \xi_s$$

where $\mathbf{C}_s$ is the set of customers for supplier s. These probabilistic constraints are "soft": they may be violated in some scenarios. We therefore do not add them to the model (as with the deterministic constraints), but instead use them to define an objective function:

$$\max \sum_s E\left\{\sum_{c \in \mathbf{C}_s} x_{s,c} \leq \xi_s\right\} \tag{1}$$

where $E\{C\}$, the "expectation operator" [13], is the sum of the probabilities of the scenarios in which constraint C is satisfied. This model may be viewed as a *Soft Probabilistic CSP*, that is a Probabilistic CSP [8] where some contraints are hard, plus an optimization criterion that we wish to maximise the probability that other probabilistic soft constraints are satisfied.

A drawback of this model is that the objective function does not measure plan reliability in a realistic way. For example, in any scenario in which supplier 2 cannot meet its demands (so that $x_{2,1} + x_{2,2} + x_{2,3} > \xi_2$) we cannot guarantee that *any* customer is supplied. This is therefore a worst-case plan for the given scenario, yet in the above model only one probabilistic constraint is violated under this scenario. A plan in which two or three probabilistic constraints are violated would be assigned a lower objective function value, but would be no less reliable. Worse still, consider a similar problem in which supplier 1 supplies only customer 1, supplier 3 supplies only customer 3, and supplier 2 again supplies customers 1, 2 and 3. A plan in which suppliers 1 and 3 are unable to meet their demands under some scenario would be classed as less reliable than one in which supplier 2 is

Plan No	Planned Delivery $S_i \rightsquigarrow D_j$: (i,j) $(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$	$(2,3)$	$(3,2)$	$(3,3)$	ERM Model 1	Model 2	Model 3	Model 4
1	3	0	5	6	1	1	3	1.7	0.0	0.0	0.0
2	3	5	5	1	1	1	3	1.5	0.6	0.6	0.6
3	0	2	0	2	4	3	0	0.0	0.0	0.0	2.0
4	5	0	3	3	0	4	4	2.4	1.8	2.1	2.1
5	6	0	2	0	4	7	0	2.4	1.8	2.4	2.4

Table 1 Representative distribution plans and event realization measures (ERM), that is reliability measures, computed by the different models we presented

unable to meet its demand under the same scenario, because more probabilistic constraints are violated. However, the latter plan is less reliable: in the first plan customer 2 is satisfied, but in the second plan no customer is.

In Table 1 we show how this naive model (column "Model 1") classifies reliability of five different plans for our concrete example. In the next sections we will show that other models can give a more accurate and realistic measure of the reliability of these plans.

2.3 Model 2: Dependent-Chance Programming

To improve the naive model we may define a more intelligent objective function: the reliability of a plan is now the sum of the reliabilities of three *events*, where an event is the satisfaction of a customer:

$$\max \sum_{c} E\left\{ \bigwedge_{s\in\mathbf{S}_c} \left(\sum_{c'\in\mathbf{C}_s} x_{s,c'} \le \xi_s \right) \right\} \tag{2}$$

where $\wedge$ denotes logical conjunction: $E\{C \wedge C'\}$ is the sum of the probabilities of the scenarios in which both C and C' are satisfied. For example the reliability of satisfaction of customer 1 is the sum of the probabilities of the scenarios in which suppliers 1 and 2 both meet their demands. Under this objective function, our worst-case plan (in which supplier 2 cannot meet its demands) is assigned reliability 0 in the scenario, because the violated probabilistic constraint $x_{2,1} + x_{2,2} + x_{2,3} \le \xi_2$ affects the reliability of each customer. Allowing logical connectives between constraints allows us to express the problem more accurately. This model is similar to a Dependent-Chance Programming [20] approach to a related problem.

Let us observe in Table 1 how this new notion of reliability affects the plans already considered. Note that the new objective function defines a completely new notion of reliability, therefore results provided by Model 1 and 2 are incomparable since Model 1 measures reliability in terms of expected number of suppliers that meet their demand, while in Model 2 the measure refers to the expected number of unsatisfied customers. We shall see that the second notion of reliability reflects a *higher level of expressiveness* and is closer to what is perceived as *reliable* by common sense.

To gain more insight into the notion of reliability captured by Model 2 we now examine two different distribution plans, 1 and 2. These two plans share common decisions, except at $S_1 \rightsquigarrow D_2$ and $S_2 \rightsquigarrow D_2$. Plan 1 (2) requires a capacity value of 3 units (8 units) at S_1 to be feasible, which is available with probability 1.0 (0.2). However if we consider S_2, Plan 1 (2) requires a capacity value of 12 units (7 units) to be feasible. The corresponding probability is 0.0 (0.6), thus Plan 2 is more reliable than Plan 1. It is now easy to see how logical connectives introduced in Eq. 2 capture a more intuitive and accurate measure for the reliability of a plan that, as seen, is expressed in terms of expected number of satisfied customers, respectively 0.0 and 0.6 for Plan 1 and 2. Note that the reliability measure in Model 1 classifies Plan 1 as more reliable than Plan 2, since the latter violates more probabilistic constraints. Obviously such a measure is flawed since Plan 1 is never able to reliably satisfy any customer as supplier 2 can not provide 12 units of capacity.

2.4 Model 3: EDP-CP

However, even the second model is flawed. Consider a plan in which $x_{1,1} = 0$ so that customer 1 must receive all supplies from supplier 2. The reliability of the satisfaction of customer 1 should now be independent of the ability of supplier 1 to meet its demand, but in the second model it is still dependent; this point was not considered in [20]. We should therefore refine the objective via further logical connectives between constraints:

$$\max \sum_{c} E\left\{\bigwedge_{s\in\mathbf{S}_c}\left(x_{s,c} \neq 0 \Rightarrow \sum_{c'\in\mathbf{C}_s} x_{s,c'} \leq \xi_s\right)\right\} \tag{3}$$

where $\Rightarrow$ denotes logical implication: $E\{C \Rightarrow C'\}$ is the sum of the probabilities of the scenarios in which either C is violated or C' is satisfied, or both. Because of this modification, under a scenario in which $x_{1,1} = 0$ there is no longer a penalty if

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

is violated. In this case, the reliability of a plan is gauged by an event realization measure which gives equal importance (i.e., equal weights) to satisfying demands completely at D_1, D_2, and D_3.

We now consider Plans 4 and 5. By observing differences between these plans it is easy to see how the further logical connectives introduced in Eq. 3 affects reliability of the solutions. In Plan 4, $\{S_1, S_2\} \rightsquigarrow D_1$, $\{S_2, S_3\} \rightsquigarrow D_2$, $S_3 \rightsquigarrow D_3$. In other words, S_3 supplies two customers. In Plan 5 $\{S_1, S_2\} \rightsquigarrow D_1$, but $S_3 \rightsquigarrow D_2$ and $S_2 \rightsquigarrow D_3$. Therefore in both the plans S_1 supplies the same customer D_1, and S_2 supplies two customers (respectively D_1, D_2

and D_1, D_3), while S_3 in Plan 4 supplies two customers (D_2 and D_3) and in Plan 5 only supplies one customer (D_3). If S_1 fails to meet the requirement in both the plans it will affect only D_1 with probability 0.3. S_2 cannot fail to meet the demand in both the plans. But in Plan 4 if S_3 does not provide 8 units as required, with probability 0.3, it will affect both D_2 and D_3, while in Plan 5 if S_3 does not provide 7 units, with the same probability 0.3, it will affect only D_2. Thus Plan 5 is obviously more reliable than plan 4. Such a notion is captured by Model 3, which therefore provides a more accurate reliability measure with respect to the former ones we presented. In fact the reader may observe that in Model 1 and 2 Plan 4 and 5 are classified as equally reliable.

2.5 Model 4: EDP-CP

So far, the decision-maker's objective has been to maximize the plan reliability, defined in such a way that all violated plans are treated equally. In other words, plans in which not all customer demand constraints hold are considered equally unreliable, irrespective of the number of customers that are completely satisfied. This obviously constitutes a limit for the first three models presented, since often we may get unrealistic solutions where we try to satisfy every customer achieving a poor overall reliability. An alternative objective could aim to satisfy as many customers as possible, that is to meet as many demand constraints as possible under probabilistic supply constraints. Clearly, this new objective may have a wider application and may lead to more realistic solutions where some customers may be dropped in order to serve the others with higher reliability.

In the first EDP-CP model any plan must satisfy the hard constraints on demands ζ_c, but a plan that reliably satisfies two customers might be more desirable than one that satisfies all three customers less reliably. We can model such a measure of plan reliability by removing the hard constraints and using them in the objective function instead:

$$\max \sum_{c} E\left\{\left[\bigwedge_{s\in S_c}\left(x_{s,c}\neq 0 \Rightarrow \sum_{c'\in C_s} x_{s,c'} \leq \xi_s\right)\right] \wedge \left(\sum_{s'\in S_c} x_{s',c} = \zeta_c\right)\right\} \tag{4}$$

A direct consequence of this new objective on optimized plans is that solutions may no longer aim for complete satisfaction of all customers, but most likely a subset of it, with higher reliability. Under this new objective, distribution Plan 3 in Model 4 guarantees complete satisfaction of D_2 and D_3 with a reliability score of 2.0, whereas under the previous models it is assigned reliability score 0.

In Table 1 the column for Model 4 depicts an accurate and realistic classification for the reliability measures of the plans considered in our concrete example.

2.6 A meta-constraint

We believe that our final model is of a form that will apply to many problems. The following sections present a formalization of a modeling framework to express such problems naturally and propose a compilation from the given formalization into a standard constraint program. In Section 3.2, we shall introduce a meta-constraint to simplify complex expressions such as those in our final model, so that it can be written in the form

Maximize
$\sum_c E\left\{e_c : \sum_{s' \in \mathbf{S}_c} x_{s',c} = \zeta_c\right\}$
given that
$(\forall c)\,(\forall s \in \mathbf{S}_c)$ $\textsc{Dependency}\left(\, e_c,\ \sum_{c' \in \mathbf{C}_s} x_{s,c'} \leq \xi_s,\ x_{s,c} \neq 0 \,\right).$

This construct is equivalent to Eq. 4, but by expressing the problem in this form, through the new keyword "**given that**", we separate the logical dependencies involving the events from the definition of the events. This way we aim to reduce the cognitive burden on the user. It should be noted that events e_c, although they appear as deterministic constraints[1], are actually probabilistic. In fact their probabilistic nature is induced by the given dependencies. In practice the satisfaction of customer demands, that is constraints e_c, *depends* on the selected suppliers and on the capacity they can provide. More formally, constraints e_c ($\sum_{s' \in \mathbf{S}_c} x_{s',c} = \zeta_c$) are the *events* whose reliability we wish to maximize, and in each scenario these events are subject to certain *pre-requisite* constraints ($\sum_{c' \in \mathbf{C}_s} x_{s,c'} \leq \xi_s$) and certain *conditions* ($x_{s,c} \neq 0$). Intuitively, if a pre-requisite is unsatisfied in a scenario then the event is also classed as unsatisfied in that scenario; and if a condition is unsatisfied in a scenario then the event is classed as satisfied in that scenario.

3 Event-Driven Probabilistic Constraint Programming

In this section we formalise the EDP-CP modeling framework.

3.1 Preliminaries

Recall that a constraint satisfaction problem (CSP) consists of a set of variables $\mathcal{X}$, each with a finite domain of values D_i, and a set of constraints $\mathcal{C}$, each over a subset of $\mathcal{X}$ (denoted by $Scope(C)$) and specifying allowed combinations of values for given subsets of variables. A solution is an assignment of values to the variables satisfying the constraints. A Constraint Optimisation Problem (COP) is a CSP with given objective function over a subset of $\mathcal{X}$ that we wish to maximize or minimize.

[1] Note that in the general case customer demand ζ_c may also be a random variable.

Recall that a probabilistic CSP as introduced in [8] is defined as a 6-tuple $\langle \mathcal{X}, \mathcal{D}, \Lambda, \mathcal{W}, C, \Pr \rangle$ where:

- $\mathcal{X}= \{x_1, \ldots, x_n\}$ is a set of decision variables;
- $\mathcal{D}= D_1 \times \ldots \times D_n$, where D_i is the domain of x_i;
- $\Lambda = \{\lambda_1, \ldots, \lambda_l\}$ is a set of uncertain parameters;
- $\mathcal{W}= W_1 \times \ldots \times W_l$, where W_i the domain of λ_i;
- C is a set of (probabilistic) constraints each involving at least one decision variable (and possibly some uncertain parameters);
- $\Pr : \mathcal{W} \to [0, 1]$ is a probability distribution over uncertain parameters.

In [8] a complete assignment of the uncertain parameters (resp. of the decision variables) is called a *world* (resp. a *decision*). The probability that a decision is a solution is the probability of the set of the worlds in which it is a solution.

In its most general form, event-driven probabilistic CP supports uncertain parameters as well as decision variables. A constraint is said to be *probabilistic*, if it involves both decision variables and uncertain parameters. In the rest of this paper we will sometimes refer to classical constraints as deterministic constraints to distinguish them from the probabilistic ones. We will refer to the possible values of an uncertain parameter λ_i as $W(\lambda_i)$ and to the probability of λ_i taking a given value v in $W(\lambda_i)$ as $\Pr(\lambda_i = v)$. As in [8], we refer to a complete assignment of uncertain parameters as a *possible world* and denote by $\mathcal{W}$ the set of all possible worlds. We also assume that the probability of each possible world w is given by the probability function $\Pr : \mathcal{W} \to [0, 1]$.

Definition 1 ([8]) *Given a probabilistic constraint c over decision variables and some uncertain parameters, the reduction of c by world $w \in \mathcal{W}$, denoted by $c_{\downarrow w}$, is the deterministic constraint obtained by setting all its uncertain parameters as in w.*

3.2 Modeling framework

In EDP-CP some of the constraints can be designated by the user as *event constraints*. The user's objective is to maximize his/her chances of realizing these events. For instance, in our running example the user may consider the customer demand constraints as events. The objective is then to construct a plan satisfying customer demand constraints as far as possible.

The feasibility of certain event constraints depends on the satisfaction of other constraints. For instance, having a plan that meets the customer demands depends on whether or not the supply constraints are met with such a plan. For this purpose we introduce a new meta-constraint (already described in Section 2.6) useful for modeling such situations in our EDP-CP framework, which we refer to as a *dependency meta-constraint*. We first introduce the dependency constraint in the deterministic setting.

Definition 2 DEPENDENCY(e, p, c) iff *Scope*$(e) \cap$*Scope*$(p) \neq \emptyset$ & *Scope*$(c) \subseteq$ *Scope*$(e) \cap$*Scope*(p) & $e \wedge (c \Rightarrow p)$*, where* e*,* p*, and* c *are all deterministic constraints.*

The DEPENDENCY meta-constraint is satisfied *if and only if* e is satisfied and, *if* c holds, p is satisfied. We refer to p as a *pre-requisite* constraint for event constraint e, and c as a *condition* constraint for p. Note that by definition, DEPENDENCY$(e, p, c \vee c')$ is equivalent to DEPENDENCY$(e, p, c) \wedge$ DEPENDENCY(e, p, c'), and similarly DEPENDENCY$(e, p \wedge p', c)$ is equivalent to DEPENDENCY$(e, p, c) \wedge$ DEPENDENCY(e, p', c).

We now introduce a measure for event realization in a deterministic setting, and later generalize it to probabilistic events.

Definition 3 *Given a deterministic event constraint* e *with* $Scope(e) = \{x_1, \ldots, x_k\}$*, an event realization measure* $E\{e\}$ *on* e *is a mapping* M *from* $D(x_1) \times \ldots \times D(x_k)$ *into* $\{0, 1\}$ *such that for all* $t \in D(x_1) \times \ldots \times D(x_k)$*,* $M(t) = 1$ *iff* t *satisfies all the* DEPENDENCY *constraints that have* e *as event constraint argument.*

Example 1 Given the meta-constraint DEPENDENCY$(e, x_1 \leq 4, x_2 \neq 0)$, an event realization measure on event constraint $e : x_1 + x_2 = 8$, denoted by $E\{e\}$, takes value 1 only when the values v_1 and v_2 assigned to decision variables x_1 and x_2 (resp.) sum to 8 and, if x_2 is different than zero, x_1 is less or equal to 4, otherwise it takes value 0.

When the events are probabilistic constraints, the event realization measure is defined on the set of possible worlds as follows.

Definition 4 *Given a probabilistic event constraint* e *with* $Scope(e) = \{x_1, \ldots, x_k\}$ *and uncertain parameters* $\Lambda = \{\lambda_1, \ldots, \lambda_l\}$*, an event realization measure* $E\{e\}$ *on* e *is a mapping* M *from* $D(x_1) \times \ldots \times D(x_k)$ *into interval* $[0, 1]$ *such that*

$$E\{e\} = \sum_{w \in W(\lambda_1) \times \ldots \times W(\lambda_l)} Pr(w) E\{e_{\downarrow w}\}$$

Example 2 An event realization measure on probabilistic constraint $e : x_1 + x_2 \leq \xi)$, where ξ is a discrete random variable assuming $\{6(0.2), 8(0.7), 11(0.1)\}$, is denoted by $E\{e\}$ and takes the value 0.8 when $x_1 = 4$ and $x_2 = 3$, and the value 0.1 when $x_1 = 6$ and $x_2 = 3$.

For convenience we shall only considered the "expectation operator" in defining an event realization measure. However, any other relevant operator, such as the nth moment generator [13], can be used instead.

The following example demonstrates the use of the DEPENDENCY meta-constraint in a probabilistic setting.

Example 3 In Figure 2 the event e_1 is the demand constraint for the first customer $e_1 : x_{1,1}+x_{2,1} = 8$, while the pre-requisite constraints are the probabilistic supply constraints $p_1 : x_{1,1} + x_{1,2} \leq \xi_1$, $p_2 : x_{2,1} + x_{2,2} + x_{2,3} \leq \xi_2$, and $p_3 : x_{3,2} + x_{3,3} \leq \xi_3$. Now consider event e_1. From the constraint scopes we see that $Scope(e_1) \cap Scope(p_1) = \{x_{1,1}\}$, $Scope(e_1) \cap Scope(p_2) = \{x_{2,1}\}$ and $Scope(e_1) \cap Scope(p_3) = \emptyset$, so e_1 depends on p_1 and p_2, not p_3. From the problem semantics we should introduce the condition constraints $c_1 : x_{1,1} \neq 0$ and $c_2 : x_{2,1} \neq 0$, to express the fact that there is no dependency relation between e_1 and p_1 if $x_{1,1} = 0$, and that there is no dependency relation between e_1 and p_2 if $x_{2,1} = 0$. Thus we write the dependency meta-constraints $\textsc{Dependency}(e_1, p_1, x_{1,1} \neq 0)$ and $\textsc{Dependency}(e_1, p_2, x_{2,1} \neq 0)$.

Equipped with these concepts, we now define EDP-CP as follows.

Definition 5 *An EDP-CP is a 9-tuple* $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \Lambda, \mathcal{W}, \mathcal{E}, \mathcal{C}, \mathcal{H}, \Psi, Pr\rangle$ *where:*

- $\mathcal{X} = \{x_1, \ldots, x_n\}$ *is a set of decision variables;*
- $\mathcal{D} = D_1 \times \ldots \times D_n$*, where* D_i *is the domain of* X_i*;*
- $\Lambda = \{\lambda_1, \ldots, \lambda_l\}$ *is a set of uncertain parameters;*
- $\mathcal{W} = W_1 \times \ldots \times W_l$*, where* W_i *the domain of* λ_i*;*
- $\mathcal{E} = \{e_1, \ldots, e_m\}$ *is a set of event constraints. Each* e_i *may either be probabilistic (involving a subset of* $\mathcal{X}$ *and a subset of* Λ*) or deterministic (involving only a subset of* $\mathcal{X}$*);*
- $\mathcal{C} = \{c_1, \ldots, c_o\}$ *is a set of dependency meta-constraints. For each dependency meta-constraint* $c_i : \textsc{Dependency}(e, p, f)$ *we have* $e \in \mathcal{E}$*, where* p *may be either a probabilistic or a deterministic pre-requisite constraint, and* f *is a deterministic condition constraint;*
- $\mathcal{H} = \{h_1, \ldots, h_p\}$ *is a set of hard constraints. Each* h_i *may either be probabilistic (involving a subset of* $\mathcal{X}$ *and a subset of* Λ*) or deterministic (involving only a subset of* $\mathcal{X}$*);*
- Ψ *is any expression involving the event realization measures on the event constraints in* $\mathcal{E}$*;*
- $Pr : \mathcal{W} \rightarrow [0, 1]$ *is a probability distribution over uncertain parameters.*

In Figure 3 we show a modeling template for EDP-CP.

Example 4 The motivational example of Section 2 can be expressed as an EDP-CP $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \Lambda, \mathcal{W}, \mathcal{E}, \mathcal{C}, \mathcal{H}, \Psi, \mathrm{Pr}\rangle$ where:

- $\mathcal{X} = \{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, x_{3,2}, x_{3,3}\}$;
- $\mathcal{D} = [0..99] \times [0..99] \times [0..99]$;
- $\Lambda = \{\xi_1, \xi_2, \xi_3\}$;
- $\mathcal{W} = \{3(0.3), 7(0.5), 12(0.2)\} \times \{6(0.4), 7(0.2), 10(0.4)\} \times \{3(0.3), 8(0.7)\}$;
- $\mathcal{E} = \{e_1 : x_{1,1} + x_{2,1} = 8, e_2 : x_{1,2} + x_{2,2} + x_{3,2} = 7, e_3 : x_{2,3} + x_{3,3} = 4\}$;
- $\mathcal{C} = \{c_1 : \textsc{Dependency}(e_1, p_1 : x_{1,1} + x_{1,2} \leq \xi_1, f_{1,1} : x_{1,1} \neq 0),$
  $c_2 : \textsc{Dependency}(e_1, p_2 : x_{2,1} + x_{2,2} + x_{2,3} \leq \xi_2, f_{2,1} : x_{2,1} \neq 0),$
  $c_3 : \textsc{Dependency}(e_2, p_1, f_{1,2} : x_{1,2} \neq 0),$
  $c_4 : \textsc{Dependency}(e_2, p_2, f_{2,2} : x_{2,2} \neq 0),$

Maximize:
$\Psi(E\{e_1\},\ldots,E\{e_m\})$

Given that:

dependency meta-constraint c_1
...
dependency meta-constraint c_o

Subject to:

hard constraint h_1
...
hard constraint h_p

Fig. 3 An EDP-CP template

c_5 : Dependency$(e_2, p_3 : x_{3,2} + x_{3,3} \leq \xi_3, f_{3,2} : x_{3,2} \neq 0)$,
c_6 : Dependency$(e_3, p_2, f_{2,3} : x_{2,3} \neq 0)$,
c_7 : Dependency$(e_3, p_3, f_{3,3} : x_{3,3} \neq 0)\}$;
– $\mathcal{H} = \{x_{1,1} \geq 0, \ldots, x_{3,3} \geq 0\}$;
– Ψ is $E\{e_1\} + E\{e_2\} + E\{e_3\}$;
– $\Pr(\langle \xi_1{=}3, \xi_2{=}6, \xi_3{=}3\rangle){=}0.036, \ldots, \Pr(\langle \xi_1{=}12, \xi_2{=}10, \xi_3{=}8\rangle){=}0.056$.

Finally, we define optimal solutions to EDP-CPs as follows.

Definition 6 *An optimal solution to an EDP-CP $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \Lambda, \mathcal{W}, \mathcal{E}, \mathcal{C}, \mathcal{H}, \Psi, Pr\rangle$ is any assignment S to the decision variables such that:*

1. *for each $h \in \mathcal{H}$, for each $w \in \mathcal{W}$, $h_{\downarrow w}$ is satisfied; and*
2. *there exists no other assignment satisfying all the hard constraints with a strictly better value for Ψ, according to the* Dependency *constraints introduced in the model.*

Note that when the total number of worlds is 1 with probability 1, the event realization measure on c is the same as in the deterministic case.

4 Solution methods for EDP-CP

We now show how to map an EDP-CP $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \Lambda, \mathcal{W}, \mathcal{E}, \mathcal{C}, \mathcal{H}, \Psi, Pr\rangle$ into an equivalent classical COP $\mathcal{P}' = \langle \mathcal{X}', \mathcal{D}', \mathcal{C}', \Psi'\rangle$.

4.1 Mapping variables and domains

Algorithm 1 shows how to create the decision variables in $\mathcal{P}'$ starting from $\mathcal{P}$, in two steps. The first step (Line 3) duplicates the decision variables in

Algorithm 1: Variable-Mapping($\mathcal{X},\mathcal{D},\Lambda,\mathcal{W},\mathcal{E},\mathcal{C}$):$\langle\ \mathcal{X}',\mathcal{D}'\ \rangle$

1 $\mathcal{X}' \leftarrow \emptyset$;
2 $\mathcal{D}' \leftarrow \emptyset$;
3 **foreach** $x \in \mathcal{X}$ **do**
  create x' with the same domain as x and add it to $\mathcal{X}'$;
4 **foreach** $e \in \mathcal{E}$ **do**
  foreach $w \in \mathcal{W}$ **do**
    create a Boolean b_w^e and add it to $\mathcal{X}'$;

Algorithm 2: Constraint-Mapping($\mathcal{X},\mathcal{D},\Lambda,\mathcal{W},\mathcal{E},\mathcal{C},\mathcal{H}$):$\mathcal{C}'$

1 $\mathcal{C}' \leftarrow \emptyset$;
2 **foreach** $e \in \mathcal{E}$ **do**
  foreach $w \in \mathcal{W}$ **do**
    $k \leftarrow\ e_{\downarrow w} \wedge \left[\bigwedge_{\{\textsc{Dependency}(\epsilon,p,c)\in\mathcal{C}|\epsilon=e\}} (c \Rightarrow p_{\downarrow w}) \right]$;
    add $b_w^e = 1 \leftrightarrow k$ to $\mathcal{C}'$;
3 **foreach** $h \in \mathcal{H}$ **do**
  foreach $w \in \mathcal{W}$ **do**
    add $h_{\downarrow w}$ to $\mathcal{C}'$;

$\mathcal{P}'$ along with their domains. The second step (Line 4) introduces a Boolean variable that is used later to represent the truth value of each event e in each possible world w.

4.2 Mapping constraints

Algorithm 2 shows how to create the constraints in $\mathcal{P}'$, again in two steps. In step one (Line 2) we introduce a reification constraint for each event e in each possible world $w \in W$. This ensures that b_w^e is assigned the value 1 iif $e_{\downarrow w}$ is satisfied and, for each $\textsc{Dependency}(e,p,c)$ constraint involving event e, if the given condition c is met, the respective prerequisite $p_{\downarrow w}$ is satisfied. In the second step (Line 3) each probabilistic constraint is transformed into a set of deterministic constraints in $\mathcal{C}'$.

4.3 Mapping the objective function

Finally, the objective function of $\mathcal{P}'$ is the same function Ψ as in $\mathcal{P}$, except that we replace each occurrence of an event measure $E\{e\}$ with

$$\sum_{w\in\mathcal{W}} Pr(w) b_w^e$$

as shown in Algorithm 3.

Algorithm 3: Objective-Function-Mapping($\mathcal{X}, \mathcal{D}, \Lambda, \mathcal{W}, \mathcal{E}, \mathcal{C}, \mathcal{H}, \Psi$):$\Psi'$

```
1 Ψ' ← Ψ;
2 foreach E{e} ∈ Ψ' do
      replace E{e} with ∑_{w∈W} Pr(w) b_w^e ;
```

5 Scenario Reduction

In the former section we showed how to compile any EDP-CP program in an equivalent ordinary constraint program. Unfortunately the more scenarios we consider the more decision variables need to be introduced in the model. This may easily lead to large intractable problems when the number of scenarios is high. In [26] the authors discuss several scenario sampling techniques to cope with a similar problem arising in a scenario based approach for *stochastic constraint programming*. The purpose of these techniques is to replace a large intractable set of scenarios with a small tractable set so that solving the problem over the small set yields a solution not much different than the solution over the large one. Obviously these technique may also be applied to reduce the number of scenarios considered in EDP-CP. The scenario reduction techniques presented are well known in statistics. Typically they determine a subset of scenarios and a redistribution of probabilities relative to the preserved scenarios. No requirements on the stochastic data process are imposed and therefore the concept is general. However the authors point out that, depending on their sophistication, the reduction algorithms may require different types of data.

The simplest scenario reduction algorithm considers just a single scenario in which stochastic variables take their expected values. This is called the *expected value problem*. In what follows we recall one of the best sampling methods for experimental design, that is *Latin Hypercube Sampling* (LHS) [22]. This method ensures that a range of values for a variable are sampled. Suppose we want the sample size to be n. We divide the unit interval into n intervals, and sample a value for each stochastic variable exactly once. More precisely, let $f_i(a)$ be the cumulative probability that X_i takes the value a or less, $P_i(j)$ be the jth element of a random permutation P_i of the integers $\{0, \ldots, n-1\}$, and r be a random number uniformly drawn from $[0, 1]$. Then the jth latin hypercube sample value for the random variable X_i is:

$$f_i^{-1}\left(\frac{P_i(j) + r}{n}\right).$$

However it should be noted that the sample size n does not guarantee to produce a sample of n scenarios, since a single scenario may be chosen more than once due to, for example, the discreteness of the data.

Techniques like the one illustrated may be applied to reduce the number of scenarios to a reasonable size so that the resulting reduced problem is a tractable one. An example of this will be presented in Section 6.2, where LHS is applied to a probabilistic scheduling problem in order to reduce the set of scenarios considered and preserve the quality of the solution provided by the EDP-CP model described.

6 Illustrative examples

In this section we present three illustrative problems and model them using the EDP-CP framework. The first example is a probabilistic supply chain planning problem, which is an extended version of the example of Section 2. In this extended version, demand uncertainty, as well as supply uncertainty, is considered. The second example is a probabilistic scheduling problem which generalizes the one proposed in [12]. In this example task durations are uncertain. The third example is a production planning problem with an emphasis on capital budgeting, and assumes that production rates, demands, prices and costs are all uncertain parameters.

6.1 An EDP-CP Model for Probabilistic Supply Chain Planning

There is a sizeable literature on supply chain modeling under uncertainty (see, for example, [7] and [23]). Recently, the authors of this work also experienced at first-hand the relevance of modeling supply and demand uncertainties during a research project carried out for a leading international telecommunications company.

Here we adopt a simplified version of the problem, which was presented in Section 2.1 and Figure 2. The objective is to determine the most reliable plan that will meet customers' realised demands at $D_{1,2,3}$ by means of uncertain deliveries from suppliers denoted by $S_{1,2,3}$. It is assumed that (i) the order batch sizes $x_{i,j}$ from supplier i to customer j is not allowed to exceed 6 units, $x_i \leq 6$, (ii) D_3 requires that its order is supplied by only one supplier, $x_{2,3}x_{3,3} = 0$. Scenario parameters are given in Table 2. These parameters can be obtained, for instance, through a sampling method like LHS, which we presented in the former section. Excess supplies from suppliers are stored at customers with a negligible inventory carrying cost until the next order issue.

We consider two possible EDP-CP models for this probabilistic supply chain problem. In the first one we try to find a solution in which all events are realised, while in the second this condition is relaxed. The EDP-CP model in Figure 4 describes the first case. The second case can be simply achieved by dropping e_1–e_3 from the set of hard constraints.

The EDP-CP model is compiled into a standard CP model using the algorithm presented in Section 4. The optimal solution is $x_{1,1} = 6$, $x_{1,2} = 1$, $x_{2,1} = 3$, $x_{2,2} = 4$, $x_{2,3} = 0$, $x_{3,2} = 2$, $x_{3,3} = 6$. In the optimal plan $E\{e_1\} = 0.420$, $E\{e_2\} = 0.294$ and $E\{e_3\} = 0.700$, giving an optimal objective function value of 1.414. In other words, this plan guarantees to meet customer

Maximize:
$E\{e_1 : x_{1,1} + x_{2,1} \geq \zeta_1\}+$
$E\{e_2 : x_{1,2} + x_{2,2} + x_{3,2} \geq \zeta_2\}+$
$E\{e_3 : x_{2,3} + x_{3,3} \geq \zeta_3\}$

Given that:

$\textsc{Dependency}(e_1, p_1 : x_{1,1} + x_{1,2} \leq \xi_1, f_{1,1} : x_{1,1} \neq 0)$
$\textsc{Dependency}(e_1, p_2 : x_{2,1} + x_{2,2} + x_{2,3} \leq \xi_2, f_{2,1} : x_{2,1} \neq 0)$
$\textsc{Dependency}(e_2, p_1, f_{1,2} : x_{1,2} \neq 0)$
$\textsc{Dependency}(e_2, p_2, f_{2,2} : x_{2,2} \neq 0)$
$\textsc{Dependency}(e_2, p_3 : x_{3,1} + x_{3,2} \leq \xi_3, f_{3,2} : x_{3,2} \neq 0)$
$\textsc{Dependency}(e_3, p_2, f_{2,3} : x_{2,3} \neq 0)$
$\textsc{Dependency}(e_3, p_3, f_{3,3} : x_{3,3} \neq 0)$

Subject to:

$0 \leq x_{i,j} \leq 6, \forall i, j \in \{1, 2, 3\}$
$x_{2,3}.x_{3,3} = 0$
$e_i, \forall i \in \{1, 2, 3\}$
$x_{i,j} \in \mathbb{Z}^{0,+}$

Fig. 4 An EDP-CP model for Probabilistic Supply Chain Planning

$\Pr(w)$	0.036	0.084	0.018	0.042	0.036	0.084	0.060	0.140	0.030
w	1	2	3	4	5	6	7	8	9
S_1	3	3	3	3	3	3	7	7	7
S_2	6	6	7	7	10	10	6	6	7
S_3	3	8	3	8	3	8	3	8	3
D_1	8	8	8	7	7	7	8	8	8
D_2	7	7	7	5	5	5	5	7	7
D_3	4	6	6	4	4	6	6	4	4
$\Pr(w)$	0.070	0.060	0.140	0.024	0.056	0.012	0.028	0.024	0.056
w	10	11	12	13	14	15	16	17	18
S_1	7	7	7	12	12	12	12	12	12
S_2	7	10	10	6	6	7	7	10	10
S_3	8	3	8	3	8	3	8	3	8
D_1	9	9	9	8	8	8	7	7	7
D_2	5	5	5	5	3	3	3	5	5
D_3	6	6	4	4	6	6	4	4	6

Table 2 Scenario Data

demands at $D_{1,2,3}$ with probabilities 42.0%, 29.4% and 70.0%, respectively. This plan aims to satisfy customer demands completely.

In most circumstances it would be more realistic to assume that the event constraints e_1, e_2, and e_3 are not hard constraints and the expected plan should not aim for a complete demand satisfaction. When we drop these hard event constraints, the following plan is optimal under such a relaxation: $x_{1,1}=6$, $x_{1,2}=0$, $x_{2,1}=3$, $x_{2,2}=3$, $x_{2,3}=0$, $x_{3,2}=2$, $x_{3,3}=6$. The event constraint satisfaction probabilities are now $E\{e_1\}=0.700$, $E\{e_2\}=0.476$ and $E\{e_3\}=0.700$, giving a total of 1.876.

A comparison of two plans shows that there are differences between them at $x_{1,2}$ and $x_{2,2}$. It may not be immediately obvious why we change $x_{1,2}$ from 1 to 0, as in both plans the probability of acquiring the required capacity at S_1 (7 and 6, respectively) is 0.8. The explanation lies in the probability distribution of the uncertain capacity of S_2. Supplier S_2 can provide 6 units with a probability of 1.0, but not 7 units. The second plan exploits this situation and aims for a partial satisfaction at D_2 by providing only 5 units. Thus there is no need for any delivery from S_1 to D_2. The second plan has higher reliability at the expense of partial satisfaction at D_2. It should be noted that there are alternative optimal solutions to this instance.

6.2 An EDP-CP Model for Scheduling

We consider a specific scheduling problem similar to the one considered by Hooker et. al [11]. This scheduling problem was described in [12] and it involves finding a least-cost schedule to process a set of orders I using a set of dissimilar parallel machines M. Processing an order $i \in I$ can only begin after the release date r_i and must be completed at the latest by the due date d_i. Order i can be processed on any of the machines. The processing cost and the processing time of order $i \in I$ on machine $m \in M$ are c_{im} and p_{im}, respectively.

The model just described is fully deterministic, but we will now consider a generalization of this problem to the case where some inputs are uncertain. For convenience we will just consider uncertain processing times π_{im} for order $i \in I$ on machine $m \in M$. Nevertheless it is easy to see that EDP-CP can be also employed to model more complicated generalizations of this problem where release dates and due dates are uncertain or processing costs are uncertain.

Scheduling with uncertainty is a topic that has been explored in a variety of fields including artificial intelligence, operations research, fault-tolerant computing and systems. For surveys on the literature see Davenport and Beck [6], Herroelen and Leus [10], and Bidot [2]. In Beck and Wilson [1] a classification of possible approaches for scheduling under uncertainty is summarized. They report three techniques that are usually employed to face uncertainty. In redundancy-based techniques extra resources/time are allocated to every task to cushion the impact of unexpected events during execution. Probabilistic techniques instead tend to build a schedule that optimizes a measure of probabilistic performance, such as expected makespan or expected weighted tardiness. Contingent and policy based approaches typically generate a branching or contingent schedule or, in extreme cases,

Maximize:
$\sum_{i \in I} E\{e_i : s_i + \sum_{m \in M} \pi_{im} * \delta_{im} \leq d_i\}$

Given that:

$\textsc{Dependency}(e_i, s_j \geq s_i + \sum_{m \in M} \pi_{im} * \delta_{im}, \sigma_{ij} = 1)$, $\forall i, j \in I, i \neq j$

Subject to:

$s_i \geq r_i, \forall i \in I$
$\sigma_{ij} = 1 \Rightarrow s_i < s_j, \forall i, j \in I, i \neq j$
$\sum_{m \in M} \delta_{im} = 1, \forall i \in I$
$\sigma_{ij} + \sigma_{ji} \geq \delta_{im} + \delta_{jm} - 1, \forall m \in M, \forall i, j \in I, i \neq j$
$\sigma_{ij} + \sigma_{ji} \leq 1, \forall i, j \in I, i \neq j$
$\sum_{i \in I} \left(\sum_{m \in M} c_{im} * \delta_{im}\right) \leq B$

$\sigma_{ij} \in \{0, 1\}, \forall i, j \in I$
$\delta_{im} \in \{0, 1\}, \forall i \in I, \forall m \in M$
$s_i \in [L_s, L_e], \forall i \in I$

Fig. 5 An EDP-CP model for Scheduling

a policy, that specify a set of actions to be taken when a particular set of circumstances arises. Our EDP-CP approach can be classified as probabilistic under a predefined policy.

Since EDP-CP is meant to model and optimize the reliability of a given plan we will assume in our problem that a fixed budget B is given and that our plan has to meet such a constraint on the costs. Therefore we will no longer look for a least-cost plan, rather we will optimize a reliability measure expressed in terms of events, as it is usual in EDP-CP. The specific event whose probability we wish to maximize is the successful completion of each job within the given time frame defined by its release and due date. Since jobs are scheduled in sequence on each machine dependencies will arise between subsequent jobs. We adopt a specific policy that unschedules a job whether this is not processed within the given due date or before the planned start time of the subsequent job on the respective machine. This policy guarantees that every order will always start at the planned start time, since the respective machine will be free and will start processing it. More complicated EDP-CP models may also consider the case where we aim to minimize total tardiness or total completion time of a given plan. In this cases the realized processing time of an order may affect the scheduling time of subsequent orders. We will not analyze these cases in this example. An EDP-CP model for the problem described is given in Fig. 5. Let us analyze

	$w \in W$	1		2		3		4		5		6	
	$\Pr\{w\}$	0.1		0.1		0.05		0.1		0.05		0.2	
	$m \in M$	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
$i \in I$	1	10	14	9	15	11	13	9	14	9	15	11	13
	2	6	8	5	9	7	7	7	8	5	9	7	12
	3	11	16	10	18	12	15	4	16	10	18	14	15
	4	7	9	6	10	8	8	8	9	6	10	8	8
	5	12	17	11	18	13	16	12	17	4	18	13	16

	$w \in W$	7		8		9		10		11		12	
	$\Pr\{w\}$	0.05		0.05		0.1		0.05		0.05		0.1	
	$m \in M$	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
$i \in I$	1	10	14	9	15	11	13	10	14	9	15	11	13
	2	6	8	15	9	7	7	16	8	5	9	7	7
	3	16	16	10	18	10	15	11	16	10	18	12	15
	4	7	9	6	10	8	8	17	9	6	10	8	8
	5	12	17	11	18	13	16	12	17	11	18	13	6

Table 3 Order processing times

the given model. The objective function maximizes the expected number of tasks completed by the respective due dates. Dependency constraint states that, when two jobs i, j are executed in sequence on the same machine (condition $\sigma_{ij} = 1$), job i has to be completed by its due date (event e_i is satisfied) and before the start time of job j (pre-requisite $s_j \geq s_i + \sum_{m \in M} \pi_{im} * \delta_{im}$). The hard constraints respectively state that: the start time of job i, s_i, must be no less than the release time r_i for this job; if two jobs i, j are processed on the same machine m and i is processed before j then the start time of i, s_i, must be less than the start time of j, s_j; each job must be processed on a machine; if two jobs i, j are processed on the same machine m, either i is processed before j, $\sigma_{ij} = 1$, or j is processed before i, $\sigma_{ji} = 1$; the processing costs must be no greater than the given budget B.

We now consider an instance of this problem. We consider 5 orders $\{I_1, \ldots, I_5\}$ on 2 parallel machines $\{M_1, M_2\}$. The uncertain processing times of each order on each machine are shown in Table 3. The release dates for the orders are $[2, 4, 6, 8, 10]$. The due dates are $[16, 13, 30, 41, 35]$. The costs for processing orders on machine M_1 are $[10, 8, 12, 11, 9]$, and on machine M_2 they are $[16, 5, 17, 9, 4]$. The given budget B is 40.

We define

$$L_s = \min_{i \in I} r_i$$

and

$$L_e = L_s + \min_{m \in M} \sum_{i \in I} \lceil \pi_{im} \rceil$$

where $\lceil \pi_{im} \rceil$ is the maximum duration of order $i \in I$ on machine $m \in M$ for every possible world $w \in W$. Therefore

$$\lceil \pi_{im} \rceil = \max_{w \in W} \pi_{im}.$$

In order to solve the proposed scheduling problem we compiled the EDP-CP model into a standard constraint program as described in Section 4. This constraint program was solved using OPL Studio 3.7 on an Intel(R)

	$w \in W$	2		6		7		10	
	$\Pr\{w\}$	0.25		0.25		0.25		0.25	
	$m \in M$	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
$i \in I$	1	9	15	11	13	10	14	10	14
	2	5	9	7	12	6	8	16	8
	3	10	18	14	15	16	16	11	16
	4	6	10	8	8	7	9	17	9
	5	11	18	13	16	12	17	12	17

Table 4 Order processing times, LHS with 4 samples

	$w \in W$	3		11	
	$\Pr\{w\}$	0.5		0.5	
	$m \in M$	M_1	M_2	M_1	M_2
$i \in I$	1	11	13	9	15
	2	7	7	5	9
	3	12	15	10	18
	4	8	8	6	10
	5	13	16	11	18

Table 5 Order processing times, LHS with 2 samples

Centrino(TM) CPU 1.50GHz with 2Gb RAM. We chose the provided *dichotomic* strategy and *depth-bounded discrepancy search* procedure.

An optimal solution for the given instance was found in 6.97 seconds, it has a cost of 40 and an overall reliability measure of 4.8, which means that in our plan 4.8 orders over 5 will be, in the average case, processed within the required due date and before the next order scheduled on the same machine. More specifically, the realization measures for each event constraint are: $E\{e_1\} = 100\%$, $E\{e_2\} = 80\%$, $E\{e_3\} = 100\%$, $E\{e_4\} = 100\%$ and $E\{e_5\} = 100\%$. The optimal plan assigns orders $\{1, 3\}$ to M_1 and orders $\{2, 4, 5\}$ to M_2. The start times for the orders are $[2, 4, 13, 31, 13]$.

In order to reduce the size of the model input we will now perform a LHS on the original problem instance presented in Table 3. The original 12 scenarios are then reduced to only 4 sampled scenarios. The reduced instance is presented in Table 4. The optimal solution for the LHS reduced instance was found in 2.08 seconds, it has a cost of 40 and an overall reliability measure of 4.8. More specifically, the optimal plan assigns orders $\{1, 3\}$ to M_1 and orders $\{2, 4, 5\}$ to M_2. The start times for the orders are $[2, 4, 13, 31, 13]$. This is the same optimal plan found for the original problem with 12 scenarios.

We now reduce the number of scenarios to only 2 samples as shown in Table 5. The optimal solution was found in only 0.72 seconds and also in this case it corresponds to the same optimal plan described above.

We finally solved the *expected value problem* in which the random order processing times are replaced with their expected values. The expected times for processing orders on machine M_1 are $[10, 7, 11, 8, 12]$, and on machine M_2 they are $[14, 9, 16, 9, 16]$. The optimal solution for the expected value problem was found in 0.25 seconds, it has a cost of 40 and an overall reliability measure of 3.55. More specifically, the optimal plan assigns orders $\{1, 3\}$ to M_1 and orders $\{2, 4, 5\}$ to M_2. The start times for the orders are $[2, 4, 12, 32, 16]$. This plan is 26.04% less reliable than the previous ones.

As this simple example demonstrates, the expected value approach to probabilistic problems may produce solutions which are far from being close to the optimal solutions, while a sampling approach usually brings benefits in terms of processing time without sacrificing too much the optimality of the solution produced.

6.3 An EDP-CP Model for Production Planning/Capital Budgeting

In this section, a production planning problem with an emphasis on capital budgeting is used to demonstrate the flexibility of the proposed modeling framework in dealing with uncertainties.

The production planning/capital budgeting problem assumes that there are $n = 7$ types of products to be produced, under uncertain demands d_i, $i = 1, ..., 7$. Each product can be produced on only one type of machine which is assigned to this product only. The existing production floor space is $A = 50$ m^2, in which each machine type requires m_i ($m = [3, 6, 5, 3, 7, 8, 9]$) in m^2 per machine of type i. The cost of operating each machine involves two types of costs: fixed cost f_i ($f = [40, 75, 62, 39, 53, 19, 38]$) and variable production cost c_i. The total production budget is $B = \$670$. The variable production cost components $c_{1,...,7}$ are uncertain, taking different values in each world $w_{1,...,4}$ (see Table 6). The produced amount of each product depends on the number of machines used, x_i, and the uncertain machine production rate, r_i, is also given in Table 6. Table 6 shows two more uncertain problem parameters: demand d_i and selling price p_i.

Under these uncertainties, a realistic objective is to determine the most reliable plan (i.e. how many machines to purchase of each type) that maximizes our chances of meeting our demand constraints as much as possible, while achieving a specified target profit of $T = \$40$, not exceeding our budget B, and meeting all space and production constraints. It is assumed that meeting customer demands and the profit target are equally important events.

In specific solution/plans, depending on unfolding of uncertainties, the budget constraint may hold, as well as demand and target profit objectives. It should be noted that it is not generally possible to find a solution which always satisfies all the constraints. For that reason, the problem addressed here is very different from the well-established techniques dealing with uncertainty.

An EDP-CP model of the production planning/capital budgeting problem is shown in Figure 6, where $r_i x_i$ and $\min(r_i x_i, d_i)$ denote the amount produced and sold, respectively, of product type $i \in \{1, ..., n\}$, and x_i denotes the number of machine used in the production of type i product. There is only one pre-requisite constraint (the budget constraint) and no condition constraint.

The optimal solution found is $x^* = [2, 0, 2, 0, 0, 2, 2]$. This production plan gives $E\{e_1\} = 100\%$, $E\{e_2\} = 0$, $E\{e_3\} = 73\%$, $E\{e_4\} = 0$, $E\{e_5\} =$

		production cost							demand						
w	Pr	c_1	c_2	c_3	c_4	c_5	c_6	c_7	d_1	d_2	d_3	d_4	d_5	d_6	d_7
1	0.16	3	6	1	1	6	10	2	4	7	2	8	3	5	2
2	0.19	4	4	7	2	4	7	7	7	9	9	9	4	7	4
3	0.38	5	3	5	8	7	6	10	9	11	12	10	7	8	7
4	0.27	5	6	8	5	5	3	6	11	13	17	11	13	16	13
		selling price							production rate						
w	Pr	p_1	p_2	p_3	p_4	p_5	p_6	p_7	r_1	r_2	r_3	r_4	r_5	r_6	r_7
1	0.16	8	14	4	16	14	10	4	2	3	2	1	2	1	2
2	0.19	10	16	18	18	10	14	14	4	4	5	2	4	3	6
3	0.38	18	22	14	18	14	16	24	5	5	6	3	5	5	4
4	0.27	22	26	26	22	16	24	18	9	6	8	4	7	7	7

Table 6 Problem Data

Maximize:

$\frac{1}{2n}\sum_{i=1}^{n} E\{e_i : \min(r_i x_i, d_i) = d_i\} +$
$\frac{1}{2}E\{\bar{e} : \sum_{i=1}^{n} p_i \min(r_i x_i, d_i) - f_i x_i - c_i r_i x_i \geq T\}$

Given that:

$\textsc{Dependency}(e_j, \sum_{i=1}^{n}(f_i + c_i r_i)x_i \leq B, \text{True}),\ \forall j \in \{1, \dots, n\}$
$\textsc{Dependency}(\bar{e}, \sum_{i=1}^{n}(f_i + c_i r_i)x_i \leq B, \text{True})$

Subject to:

$\sum_{i=1}^{n} m_i x_i \leq A$
$x_i \in \mathbb{Z}^{0,+}$

Fig. 6 An EDP-CP model for Production Planning/Capital Budgeting

0, $E\{e_6\} = 38\%$, $E\{e_7\} = 100\%$, where event constraint e_i denotes the complete satisfaction of demand for product type i. In this plan the profit target is achieved $E\{\bar{e}\} = 65\%$ of the time.

We also solved the *expected value problem* in which the random variables production cost, demand, selling price and production rate are replaced with their expected values. The expected value data used in the deterministic problem are given in Table 7.

product type	1	2	3	4	5	6	7
production cost	4.49	4.48	5.55	4.93	5.73	6.02	7.07
demand	8.36	10.52	11.18	9.76	7.41	9.49	7.25
selling price	15.96	20.66	16.40	18.76	13.78	16.82	17.28
production rate	5.41	4.76	5.71	2.76	4.87	4.52	4.87

Table 7 Expected Value Problem Data

The solution to this resultant deterministic problem is $x^* = [2, 0, 0, 4, 2, 0, 2]$. We used this plan in the original probabilistic setting to evaluate the quality of the expected value solution. The expected value solution gives $E\{e_1\} = 35\%$, $E\{e_2\} = 0$, $E\{e_3\} = 0\%$, $E\{e_4\} = 0$, $E\{e_5\} = 35$, $E\{e_6\} = 0\%$, $E\{e_7\} = 35\%$, where event constraint e_i denotes the complete satisfaction of demand for product type i. It is not possible to achieve the profit target under any scenario using this plan; in other words, $E\{\bar{e}\} = 0\%$.

Also in this case the expected value approach to probabilistic problems produces a solution which is far from being close to the optimal one.

7 Related works

The EDP-CP framework we present is a generalization of the work of Liu [20] on dependent-chance programming. Firstly, our notion of constraint dependency introduces condition constraints in addition to the event and pre-requisite constraints. It should be noted that constraint dependency without condition constraints does not guarantee optimal plans since in certain instances common variables may take values which break the link between two dependent constraints. Secondly, while a feasible solution in Liu's framework satisfies all event constraints, in our framework such a requirement is relaxed, and this gives the decision-maker more flexibility in modeling. Finally, while Liu's work only considers Monte Carlo-based simulation methods, we propose a complete solution method.

EDP-CP is also related to the probabilistic CSP framework [8]. However, probabilistic CSP treats all probabilistic constraints uniformly, whereas EDP-CP distinguishes between event, pre-requisite, condition, and hard constraints. For instance, in probabilistic CSP, all customer and demand constraints will be treated in the same way. In a given world, either all constraints are satisfied or the problem is over-constrained. While finding a plan that has the highest probability of success is an interesting objective, our approach answers different questions and achieves different objectives.

It should also be noted that, when all the constraints are deterministic, our EDP-CP framework is closely related to Partial CSPs [9]. Partial CSPs can be divided into two main categories: The Minimal Violation Problem and the Maximal Utility Problem. In the first case the goal is to find a solution which satisfies as many constraints as possible (e.g. Soft CSPs [4]) or equivalently to minimise the number of violated constraints. In the Maximal Utility Problem the objective is to find a partial solution, which violates none of the constraints where a partial solution is an assignment in which not all variables are assigned a value. In our approach we also find partial solutions, but instead of treating all constraints equally we have shown that we can obtain partial assignments that satisfy as many event constraints as possible according to the given probability distributions for the random variables and to the dependencies that have been modeled. Partial CSPs do not explicitly model high level concepts such as probability distributions, event, pre-requisite, condition, and hard constraints.

Another technique addressing constraint problems under uncertainty is Stochastic Constraint Programming (SCP) [25]. The SCP approach assumes that the constraints are stochastically independent (i.e., there are no Dependency constraints among them). Thus SCP addresses a completely different class of stochastic problems.

8 Conclusion

In this paper we propose EDP-CP as a novel modeling framework that helps decision makers in uncertain environments to realistically model their problems and find reliable solutions. The characteristic features of our modeling framework can be summarized as follows:

- To better model the uncertainties in real-world problems, we allow the set of constraints to be either deterministic or probabilistic;
- We move away from classical approaches that treat all constraints uniformly to one that distinguishes between event, pre-requisite, condition, and hard constraints;
- We introduce the Dependency meta-constraint that allows the modeler to state a problem by explicitly specifying dependency relationships between event, pre-requisite, and condition constraints;
- In an uncertain environment, it is quite unrealistic to assume that a solution is valid irrespective of the unfolding of the uncertain parameters. In fact, there is a certain degree of fuzziness associated with each candidate solution. Therefore, in our framework, we view the set of feasible solutions as probabilistic due to the inherent uncertainties;
- We introduce an event realization measure, which can be used by the modeler to define solution reliability.

Our future work will extend the proposed framework in various directions, and provide efficient and effective solving methods. Our first steps will be:

- The development of specialized solution methods for EDC-CP. For instance a specialized global constraint for the Dependency meta-constraint can be designed.
- In large-scale uncertain problems, the number of worlds can be prohibitively large. We proposed a well-known scenario reduction technique that may help to reduce the number of scenarios considered. However we will investigate further ways of reducing the number of world as well as employing effective decomposition techniques;
- We will look at ways of extending EDP-CP to deal with recourse actions.

Acknowledgements: S. Armagan Tarim and Brahim Hnich are supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. SOBAG-108K027. 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.

References

1. J. C. Beck and N. Wilson. Proactive algorithms for job shop scheduling with probabilistic durations. *Journal of Artificial Intelligence Research*, 28:183–232, 2007.
2. J. Bidot. *A General Framework Integrating Techniques for Scheduling under Uncertainty.* PhD thesis, Ecole Nationale d'Ingieurs de Tarbes, 2005.
3. J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer Verlag, New York, 1997.
4. S. Bistarelli, U. Montanari, and F. Rossi. Soft constraint logic programming and generalized shortest path problems. *Journal of Heuristics, Kluwer*, 2001.
5. A. Charnes and W. W. Cooper. Chance-constrainted programming. *Management Science*, 6(1):73–79, 1959.
6. A. Davenport and J. C. Beck. A survey of techiniques for scheduling with uncertainty. Technical report, Available at: http://www.tidel.mie.utoronto.ca/publications.php, 2000.
7. A. G. de Kok and S. C. Graves. *Handbooks in Operations Research and Management Science: Supply Chain Management: Design, Coordination and Operation*, volume 11. Elsevier, Amsterdam, 2003.
8. H. Fargier, R. Martin-Clouaire J. Lang, and T. Schiex. A constraint satisfaction framework for decision under uncertainty. In *Proceedings of the Eleventh International Conference on Uncertainty in Artificial Intelligence, Montreal, Canada*, 1995.
9. E. C. Freuder and J. W. Richard. Partial constraint satisfaction. *Artificial Intelligence*, 1992.
10. W. Herroelen and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165:289–306, 2005.
11. J. N. Hooker, G. Ottosson, E. S. Thorsteinsson, and H. J. Kim. On integrating constraint propagation and linear programming for combinatorial optimization. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 136–141. The AAAI Press/MIT Press, Cambridge, Ma., 1999.
12. V. Jain and I. E. Grossmann. Algorithms for hybrid MILP/CP models for a class of optimization problems. *INFORMS Journal on Computing*, 13:258–276, 2001.
13. H. Jeffreys. *Theory of Probability.* Clarendon Press, Oxford, 1961.
14. P. Kall and S. W. Wallace. *Stochastic Programming.* John Wiley & Sons, 1994.
15. B. G. Kingsman. *Raw Materials Purchasing: An Operational Research Approach.* Pergamon Press, New York, 1985.
16. B. Liu. Dependent-chance goal programming: a class of stochastic programming. Technical report, Intitute of System Science, Chinese Academy of Sciences, 1995.
17. B. Liu. Dependent-chance goal programming and its genetic algorithm approach. Technical report, Intitute of System Science, Chinese Academy of Sciences, 1995.
18. B. Liu. Dependent-chance programming: A class of stochastic optimization. *Computers & Mathematics with Applications*, 34:89–104, 1997.
19. B. Liu. *Uncertain Programming.* John Wiley & Sons, New York, 1999.

20. B. Liu and K. Iwamura. Modelling stochastic decision systems using dependent-chance programming. *European Journal of Operational Research*, 101:193–203, 1997.
21. B. Liu and C. Ku. Dependent-chance goal programming and an application. *Journal of Systems Engineering & Electronics*, 4:40–47, 1993.
22. M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.
23. E. L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, Stanford, CA, 2002.
24. J. K. Sengupta. *Stochastic Programming: Methods and Applications*. North-Holland, Amsterdam, 1972.
25. S. A. Tarim, S. Manandhar, and T. Walsh. Scenario-based stochastic constraint programming. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, pages 257–262, 2003.
26. S. A. Tarim, S. Manandhar, and T. Walsh. Stochastic constraint programming: A scenario-based approach. *Constraints*, 11:53–80, 2006.
27. S. Vajda. *Probabilistic Programming*. Academic Press, New York, 1972.
28. Y. Wu, J. Zhou, and J. Yang. Dependent-chance programming model for stochastic network bottleneck capacity expansion based on neural network and genetic algorithm. *Advances in Natural Computation, Springer Verlag, Lecture Notes in Computer Science No. 3612*, pages 120–128, 2005.