Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Finding (α, ϑ)-Solutions via Sampled SCSPs

Roberto Rossi,[1] Brahim Hnich,[2] S. Armagan Tarim[3] and Steven Prestwich[4]
[1]Logistics, Decision and Information Sciences, Wageningen University, the Netherlands, roberto.rossi@wur.nl
[2]Department of Computer Engineering, Izmir University of Economics, Turkey, brahim.hnich@ieu.edu.tr
[3]Department of Management, Hacettepe University, Ankara, Turkey, armagan.tarim@hacettepe.edu.tr
[4]Cork Constraint Computation Centre,University College Cork, Ireland, s.prestwich@4c.ucc.ie

Abstract

We discuss a novel approach for dealing with single-stage stochastic constraint satisfaction problems (SCSPs) that include random variables over a continuous or large discrete support. Our approach is based on two novel tools: sampled SCSPs and (α, ϑ)-solutions. Instead of explicitly enumerating a very large or infinite set of future scenarios, we employ statistical estimation to determine if a given assignment is consistent for a SCSP. As in statistical estimation, the quality of our estimate is determined via confidence interval analysis. In contrast to existing approaches based on sampling, we provide likelihood guarantees for the quality of the solutions found. Our approach can be used in concert with existing strategies for solving SCSPs.

1 Introduction

In a stochastic constraint satisfaction problem (SCSP) one typically requires a number of constraints, involving decision and random variables, to be satisfied at a prescribed probability. These constraints, which we call chance-constraints, take the form $\Pr\{\langle\text{constraint}\rangle\} \geq \beta$ and they enforce the probability $\langle\text{constraint}\rangle$ is satisfied by a given assignment to be greater or equal to a given threshold β. Typically, these constraints do not admit a closed form solution and are complex enough to rule out any chance of obtaining an exact solution via standard scenario-based approaches — for instance because they constrain random variables defined over a continuous support. We argue that a decision maker, instead of looking for an exact solution, may then aim to "estimate" for the chance-constraints in the model, the satisfaction probability guaranteed by a given assignment. In other words, having introduced a confidence level α and a tolerance threshold ϑ, the decision maker may look for a solution that, with confidence level α, guarantees a satisfaction probability that is no lower than $\beta-\vartheta$. By choosing different values for α and ϑ the set of solutions may vary. For this reason, we will introduce a new notion of solution that is parameterized by these two parameters and that we call an (α, ϑ)-solution. Intuitively, as α tends to 1 and ϑ tends to 0 the set of (α, ϑ)-solutions will converge to the actual set of solutions to the original SCSP, which we therefore rename $(1, 0)$-solutions. In this work, we formally introduce the concept of (α, ϑ)-solution and we apply it to SCSPs that include continuous random variables — i.e. an infinite number of scenarios — and that cannot be solved by exact approaches in the stochastic constraint programming literature.

2 Formal Background

An m-stage SCSP [Walsh, 2002; Tarim *et al.*, 2006; Hnich *et al.*, 2009] is defined as a 7-tuple $\langle V, S, D, P, C, \beta, L\rangle$, where V is a set of decision variables and S is a set of random variables, D is a function mapping each element of V (respectively, S) to a domain (respectively, support) of potential values. In classical SCSPs both decision variable domains and random variable supports are assumed to be finite. P is a function mapping each element of S to a probability distribution for its associated support. C is a set of chance-constraints over a non-empty subset of decision variables and a subset of random variables. β is a function mapping each chance-constraint $h \in C$ to β_h which is a threshold value in the interval $(0, 1]$. $L = [\langle V_1, S_1\rangle, \dots, \langle V_i, S_i\rangle, \dots, \langle V_m, S_m\rangle]$ is a list of *decision stages* such that each $V_i \subseteq V$, each $S_i \subseteq S$, the V_i form a partition of V, and the S_i form a partition of S.

To solve an m-stage SCSP an assignment to the variables in V_1 must be found such that, given random values for S_1, assignments can be found for V_2 such that, given random values for S_2, ..., assignments can be found for V_m so that, given random values for S_m, the hard constraints are satisfied and the chance constraints are satisfied in the specified fraction of all possible scenarios. Under the assumption that random variable supports are finite, the solution of an m-stage SCSP is, in general, represented by means of a *policy tree* [Tarim *et al.*, 2006]. The arcs in such a policy tree represent values observed for random variables whereas nodes at each level represent the decisions associated with the different stages. We call the policy tree of an m-stage SCSP that is a solution a *satisfying policy tree*.

In order to simplify the presentation, we assume without loss of generality, that each $V_i = \{x_i\}$ and each $S_i = \{s_i\}$ are singleton sets. All the results can be easily extended in order to consider $|V_i| > 1$ and $|S_i| > 1$ (see [Hnich *et al.*, 2009]). Let $S = \{s_1, s_2, \dots, s_m\}$ be the set of all random variables and $V = \{x_1, x_2, \dots, x_m\}$ be the set of all decision variables. Let p be a path from the root node of the policy tree to a leaf. Let Ψ denote the set of all distinct paths of a policy

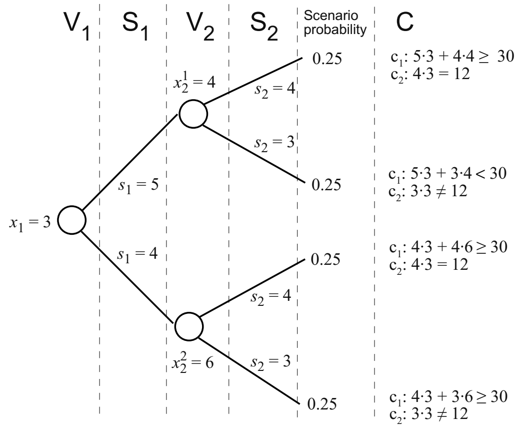

Figure 1: Policy tree for the SCSP in Example 1

tree. For each $p \in \Psi$, we denote by $arcs(p)$ the sequence of all the arcs in p whereas $nodes(p)$ denotes the sequence of all nodes in p. We denote by $\Omega = \{arcs(p) | p \in \Psi\}$ the set of all scenarios of the policy tree. The probability of $\omega \in \Omega$ is given by $\Pr\{\omega\} = \prod_{i=1}^{m} \Pr\{s_i = \bar{s}_i\}$, where $\Pr\{s_i = \bar{s}_i\}$ is the probability that random variable s_i takes value $\bar{s}_i$.

Now consider a chance-constraint $h \in C$ with a specified threshold level β_h. Consider a policy tree $\mathcal{T}$ for the SCSP and a path $p \in \mathcal{T}$. Let $h_{\downarrow p}$ be the deterministic constraint obtained by substituting the random variables in h with the corresponding values ($\bar{s}_i$) assigned to these random variables in $arcs(p)$. Let $\bar{h}_{\downarrow p}$ be the resulting tuple obtained by substituting the decision variables in $h_{\downarrow p}$ by the values ($\bar{x}_i$) assigned to the corresponding decision variables in $nodes(p)$. We say that h is *satisfied wrt to a given policy tree* $\mathcal{T}$ iff

$$\sum_{p \in \Psi : \bar{h}_{\downarrow p} \in h_{\downarrow p}} \Pr\{arcs(p)\} \geq \beta_h.$$

Definition 1 *Given an m-stage SCSP $\mathcal{P}$ and a policy tree $\mathcal{T}$, $\mathcal{T}$ is a satisfying policy tree to $\mathcal{P}$ iff every chance-constraint of $\mathcal{P}$ is satisfied wrt $\mathcal{T}$.*

Example 1 Let us consider a two-stage SCSP in which $V_1 = \{x_1\}$ and $S_1 = \{s_1\}$, $V_2 = \{x_2\}$ and $S_2 = \{s_2\}$. Random variable s_1 may take two possible values, 5 and 4, each with probability 0.5; random variable s_2 may also take two possible values, 3 and 4, each with probability 0.5. The domain of x_1 is $\{1, \ldots, 4\}$, the domain of x_2 is $\{3, \ldots, 6\}$. There are two chance-constraints in C, $c_1 : \Pr\{s_1 x_1 + s_2 x_2 \geq 30\} \geq 0.75$ and $c_2 : \Pr\{s_2 x_1 = 12\} \geq 0.5$. In this case, the decision variable x_1 must be set to a unique value before random variables are observed, while decision variable x_2 takes a value that depends on the observed value of the random variable s_1. A possible solution to this SCSP is the satisfying policy tree shown in Fig. 1 in which $x_1 = 3, x_2^1 = 4$ and $x_2^2 = 6$, where x_2^1 is the value assigned to decision variable x_2, if random variable s_1 takes value 5, and x_2^2 is the value assigned to decision variable x_2, if random variable s_1 takes value 4. As the example shows, a solution to a SCSP is not simply an assignment of the decision variables in V to values, but it is instead a satisfying policy tree.

3 Existing Approaches for Modeling and Solving SCSPs

In [Tarim *et al.*, 2006], the authors discuss an equivalent scenario-based reformulation for SCSPs. This reformulation makes it possible to compile SCSPs down into conventional (non-stochastic) CSPs. The scenario-based reformulation approach allows us to exploit the full power of existing constraint solvers. However, as pointed out in [Hnich *et al.*, 2009], it has a number of serious drawbacks that might prevent it from being applied in practice: weakened constraint propagation and significant space requirements. In [Hnich *et al.*, 2009] an alternative approach based on global chance-constraints was proposed. Global chance-constraints were introduced first in [Rossi *et al.*, 2008] and bring together the reasoning power of global constraints from constraint programming and the expressive power of chance-constraints from stochastic programming. The approach in [Hnich *et al.*, 2009] is able to reuse existing propagators available for the respective deterministic global constraint obtained when all the random variables are replaced by constant parameters.

Unfortunately, both the above approaches operate under the assumption that the number of scenarios must be finite, otherwise a policy tree would comprise an infinite number of paths. This, in turn, means that complete approaches such as the one in [Tarim *et al.*, 2006] and in [Hnich *et al.*, 2009] can only deal with random variables having finite supports. Furthermore, these approaches do not scale well, since even problems having a limited number of random variables with large support immediately produce policy trees whose size makes impractical the use of a complete method.

In practice, it is often the case that random variables either range over continuous supports or have a very large number of possible values in their domain. In [Tarim *et al.*, 2006], the authors therefore proposed to employ a number of sampling strategies in order to reduce a-priori the support of random variables and therefore produce SCSPs that are manageable. Nevertheless, their approach is purely heuristic and does not provide any likelihood guarantee on the quality of the assignments produced. The same holds for other heuristic approaches in the literature, such as the one in [Prestwich *et al.*, 2009], in which a neural network is employed in order to encode a policy function that takes the best possible decision with respect to the past history of decisions taken and values observed for the random variables.

In the rest of this work we will discuss an effective way of dealing with single-stage SCSPs comprising random variables with continuous or very large support. This approach exploits sampling in order to keep under control the amount of scenarios that must be analyzed in order to find a solution. Intuitively, our approach "estimates" if a given assignment is consistent or not with respect to a given set of chance-constraints. As in statistical estimation, the quality of this estimate is determined by confidence interval analysis. In contrast to [Tarim *et al.*, 2006], we provide likelihood guarantees for the solutions found. In fact, we explicitly indicate a confidence probability that bounds the actual probability of making a wrong estimation. Before discussing our approach, we now introduce the concept of "sampled SCSP".

4 Sampled SCSPs

Consider a SCSP $\mathcal{P}$ over a set S of random variables. Assume that random variables are defined on continuous or large discrete supports. Solving the original SCSP clearly poses a hard combinatorial challenge, in fact the policy tree comprises a number of scenarios that is exponential in the size of random variable domains. Furthermore, if the random variable support is continuous, the policy tree comprises an infinite number of scenarios. In this section we discuss how to *sample* a more compact SCSP, which comprises at most N scenarios, out of the original problem. We shall call this new problem $\widehat{\mathcal{P}}_N$ or "sampled SCSP" over N scenarios. Intuitively, a sampled SCSP is a reduced version of the original problem, the solution of which is a policy tree that comprises a bounded number of paths sampled out of the original policy tree. In the following sections we will discuss under which conditions the solution to $\widehat{\mathcal{P}}_N$ is, with a prescribed confidence probability, likely to be also a solution to the original SCSP $\mathcal{P}$.

We now discuss how to employ Simple Random Sampling to obtain a sampled SCSP out of the original problem. Of course, more advanced stratified sampling techniques may be used in order to reduce variance and improve the effectiveness of the approach. Nevertheless, we leave this discussion as future work.

Consider a complete realization, $\bar{s}_1, \ldots, \bar{s}_m$, for the random variables in S obtained by sampling a value from the support $D(s_i)$ of each of the random variables $s_i \in S$ according to its probability distribution $P(s_i)$. From the definition of policy tree it is clear that there always exists a path associated with this realization. In other words, this realization corresponds to one of the scenarios comprised in the policy tree. Consider a policy tree $\mathcal{T}$ for $\mathcal{P}$ and N complete sets of random variable realizations generated independently: $\{\bar{s}^1_1, \ldots, \bar{s}^1_m\}, \{\bar{s}^2_1, \ldots, \bar{s}^2_m\}, \ldots, \{\bar{s}^N_1, \ldots, \bar{s}^N_m\}$, where $\bar{s}^i_j$ is the realized value for random variable j observed in the i-th set of realizations. We remove from $\mathcal{T}$ every path which corresponds to an arc labeling not observed in the former N complete realizations. Let $\widehat{\mathcal{T}}$ be the reduced policy tree. $\widehat{\Psi}$ denotes the reduced set of distinct paths in $\widehat{\mathcal{T}}$. The probability of each of the remaining path $p \in \widehat{\Psi}$, i.e. $\Pr\{arcs(p)\}$, is simply set equal to the frequency of occurrence of such a path in the above N realizations. Of course, $\widehat{\mathcal{T}}$ represents a policy tree for a different SCSP than the one we started with. We call this new problem the sampled SCSP $\widehat{\mathcal{P}}_N$. Now consider a chance-constraint $h \in C$ with a specified threshold level β_h, a policy tree $\widehat{\mathcal{T}}$ for the sampled SCSP $\widehat{\mathcal{P}}_N$ and a path $p \in \mathcal{T}$. We say that h is *satisfied wrt to a given policy tree* $\widehat{\mathcal{T}}$ iff

$$\sum_{p \in \widehat{\Psi}: \bar{h}_{\downarrow p} \in h_{\downarrow p}} \Pr\{arcs(p)\} \geq \beta_h.$$

Example 2 Let us consider the two-stage SCSP $\mathcal{P}$ discussed in Example 1. We set $N = 3$ and we derive a sampled SCSP $\widehat{\mathcal{P}}_N$. By using simple random sampling we draw the following three complete realizations for random variables in $\mathcal{P}$: $\{\bar{s}^1_1 = 5, \bar{s}^1_2 = 4\}, \{\bar{s}^2_1 = 4, \bar{s}^2_2 = 4\}, \{\bar{s}^3_1 = 5, \bar{s}^3_2 = 4\}$. A possible solution to the sampled SCSP $\widehat{\mathcal{P}}_N$ is the satisfying

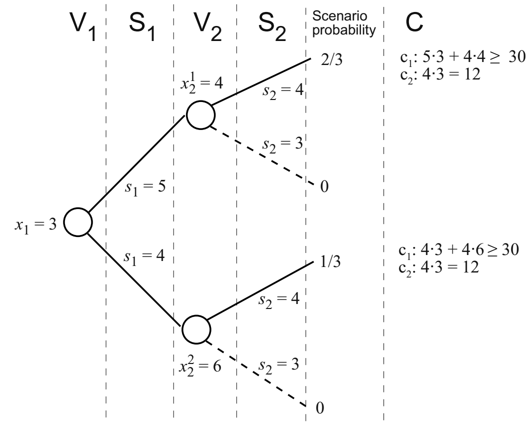

Figure 2: Policy tree for the sampled SCSP in Example 2

policy tree shown in Fig. 2, in which $x_1 = 3, x^1_2 = 4$ and $x^2_2 = 6$, where x^1_2 is the value assigned to decision variable x_2, if random variable s_1 takes value 5, and x^2_2 is the value assigned to decision variable x_2, if random variable s_1 takes value 4. The above policy tree has two paths sampled out of the original tree: p_1 has an associated probability of 2/3, since we observed two occurrences of the scenario associated with this path over the 3 complete realizations sampled for the random variables; p_2 has an associated probability of 1/3, since we observed a single occurrence of the scenario associated with this path over the 3 complete realizations sampled for the random variables. Paths that were not observed in the sampled realizations have an associated probability equal to 0 and are not considered.

It should be noted that every policy tree $\widehat{\mathcal{T}}$ for a sampled SCSP $\widehat{\mathcal{P}}$ can be employed as a (partial) policy tree for the original SCSP $\mathcal{P}$. Nevertheless, by sampling we lose completeness. If at stage i in $\mathcal{P}$ we observe, for a given random variable, a realized value that is not comprised in $\widehat{\mathcal{T}}$, it will be of course impossible to determine the correct decisions for subsequent stages. This means that all paths in the corresponding subtree will never be satisfied. In multi-stage SCSPs, and especially in those including random variables with continuous support, this is a critical issue that prevents the direct use of the approach discussed in this work. It is therefore essential to adopt a "rolling horizon" approach [Sethi and Sorger, 1991] in order to reduce the multi-stage SCSPs to a sequence of single-stage SCSPs. For space reasons, we leave this discussion as future work. In general, however, it is possible that the remaining paths form a (partial) policy tree that is a satisfying policy tree for $\mathcal{P}$. In Example 2, incidentally, the satisfying policy tree for the sampled SCSP is also a satisfying policy tree for the original SCSP. It is relatively intuitive to see that if we repeatedly produce new sampled SCSPs with $N = 3$, with a certain probability a satisfying policy tree for the sampled SCSP will also be a satisfying policy tree for the original SCSP. The rest of this work is mainly concerned with the estimation of this probability for *single-stage* SCSPs. We next introduce the relevant background in confidence interval analysis, the key tool we employ to perform this estimation.

5 Confidence Interval Analysis

Confidence interval analysis is a well established technique in statistics. Informally, confidence intervals are a useful tool for computing, from a given set of experimental results, a range of values that, with a certain confidence level (or confidence probability), will cover the actual value of a parameter that is being estimated.

Consider a discrete random variable that follows a Bernoulli distribution. Accordingly, such a variable may produce only two outcomes, i.e. "yes" and "no", with probability q and $1-q$, respectively.

Let us assume that the value q — the "yes" probability — is unknown. Obviously, if we observe the outcome of a Bernoulli trial once, the data collected will not reveal much about the value of q. Nevertheless, in practice, we may be interested in "estimating" q, by repeatedly observing the behavior of the random variable in a sequence of Bernoulli trials. This problem is well-known in statistics and both exact and approximate techniques are available for performing this estimation [Clopper and Pearson, 1934; Agresti and Coull, 1998]. The estimation produced by the methods available in the literature typically does not come as a point estimate, rather it consists of an interval of values computed from a set of representative samples for the quantity being estimated. This interval is known as "confidence interval" and consists of a range of values that, with a certain confidence probability α, covers the actual value of the parameter that is being estimated.

A method that is commonly classified as the "exact confidence intervals" for the Binomial distribution has been introduced by Clopper and Pearson in [Clopper and Pearson, 1934]. This method uses the Binomial cumulative distribution function (CDF) in order to build the interval from the data observed. The Clopper-Pearson interval can be written as (p_{lb}, p_{ub}), where

$$p_{lb} = \min\{q \mid \Pr\{\mathit{Bin}(N;q) \geq X\} \geq (1-\alpha)/2\},$$
$$p_{ub} = \max\{q \mid \Pr\{\mathit{Bin}(N;q) \leq X\} \geq (1-\alpha)/2\},$$

X is the number of successes (or "yes" events) observed in the sample, $\mathit{Bin}(N;q)$ is a binomial random variable with N trials and probability of success q and α is the confidence probability. Note that we assume $p_{lb} = 0$ when $X = 0$ and that $p_{ub} = N$ when $X = N$.

Because of the close relationship between Binomial distribution and the Beta distribution, the Clopper-Pearson interval is sometimes presented in an alternative format that uses percentiles from the beta distribution [Evans *et al.*, 2000]:

$$p_{lb} = 1 - \mathit{BetaInv}(1-(1-\alpha)/2, N-X+1, X),$$
$$p_{ub} = 1 - \mathit{BetaInv}((1-\alpha)/2, N-X, X+1),$$

where *BetaInv* denotes the inverse Beta distribution. This form can be efficiently evaluated by existing algorithms.

An interesting property of confidence intervals related to the estimation of the "success" probability associated with a Bernoulli trial consists in the fact that, given a confidence probability, it is possible to derive mathematically, by performing a worst case analysis, the minimum number of samples that should be observed in order to produce a confidence interval of a given size.

Therefore, for a given confidence probability α, it is possible to determine the minimum number of samples that should be considered in order to achieve a margin of error of $\pm\vartheta$ in the estimation of the "success" probability of a Bernoulli trial. This computation plays a central role in our novel approach. In fact, intuitively estimating the satisfaction probability of a chance-constraint is equivalent to estimating the "success" probability of the associated Bernoulli trial.

6 Properties of Sampled SCSP Solutions

We will now characterize the probability that the solution of a sampled SCSPs $\widehat{\mathcal{P}}_N$ over N scenarios, which may be computed by using any of the existing approaches discussed in Section 3, is a solution to the original *single-stage* SCSP $\mathcal{P}$.

We will firstly discuss what the minimum value for N is in order to achieve a predefined probability α that a given policy tree $\mathcal{T}$ that satisfies a chance-constraint h in the sampled SCSPs $\widehat{\mathcal{P}}_N$ also satisfies the same chance-constraint in the original SCSP $\mathcal{P}$. Since a policy tree $\mathcal{T}$ in $\widehat{\mathcal{P}}_N$ by definition only comprises a subset $\widehat{\Psi}$ of all the paths that constitute a policy tree for the original SCSP $\mathcal{P}$, this policy tree, in order to satisfy h in the original SCSP $\mathcal{P}$, must clearly provide a sufficient satisfaction probability regardless of the scenarios that have been ignored by the sampling process.

Consider a confidence probability α and a margin of error of $\pm\vartheta$; The number of scenarios N for the sampled SCSP depends on ϑ, α and also β, which we recall is the satisfaction probability we aim for our chance-constraint h.

Definition 2 *N is computed as the minimum value for which*

$$\max(p^{\beta}_{ub} - \beta, \beta - p^{\beta}_{lb}) \leq \vartheta,$$

where $(p^{\beta}_{lb}, p^{\beta}_{ub})$, is the Clopper-Pearson confidence interval for a confidence probability $\widehat{\alpha}$, where $\widehat{\alpha} = 2\alpha - 1$,[1] and round($\beta N$) "successes" in N trials; round() approximates the value to the nearest integer.[2]

Definition 3 *Any policy tree $\mathcal{T}$, which can be proved to satisfy h in $\mathcal{P}$ with probability α, satisfies h in $\mathcal{P}$ with probability α if it satisfies h in $\widehat{\mathcal{P}}_N$. Conversely, any policy tree $\mathcal{T}$, which can be proved to not satisfy h in $\mathcal{P}$ with probability α, does not satisfy h in $\mathcal{P}$ with probability α, if it does not satisfy h in $\widehat{\mathcal{P}}_N$.*

Proposition 1 *A policy tree $\mathcal{T}$ can be proved to satisfy h in $\mathcal{P}$ with probability α if the actual satisfaction probability $\delta > \beta$ provided by $\mathcal{T}$ wrt h is such that $\delta \geq p^{\beta}_{ub}$. Conversely, if the actual satisfaction probability $\delta < \beta$ provided by $\mathcal{T}$ wrt h is such that $\delta \leq p^{\beta}_{lb}$ $\mathcal{T}$ can be proved to not satisfy h in $\mathcal{P}$ with probability α.*

[1] This transformation is required because Clopper-Pearson interval is a symmetric two-sided confidence interval, while when we determine if a policy tree satisfies or not a given chance-constraint we do this on the basis of a single-sided interval.

[2] This is justified by the fact that the Clopper-Pearson interval is, in fact, a step function — see [Clopper and Pearson, 1934], p. 405 — since the Binomial is a discrete probability distribution.

Proof: Let $\delta \geq p_{ub}^{\beta}$. By definition,

$$p_{ub}^{\beta} = \max\{q | \Pr\{\mathit{Bin}(N;q) \leq \mathit{round}(\beta N)\} \geq (1-\widehat{\alpha})/2.$$

Therefore, it is clear that $\Pr\{\mathit{Bin}(N;\delta) \leq \mathit{round}(\beta N)\} < 1-\alpha$. This means that

$$\Pr\left\{\sum_{p\in\widehat{\Psi}:\bar{h}_{\downarrow p}\in h_{\downarrow p}} \Pr\{arcs(p)\} \leq \beta\right\} < 1-\alpha,$$

where we recall that $\widehat{\Psi}$ is the set of paths in the sampled SCSP $\widehat{\mathcal{P}}_N$. This implies

$$\Pr\left\{\sum_{p\in\widehat{\Psi}:\bar{h}_{\downarrow p}\in h_{\downarrow p}} \Pr\{arcs(p)\} \geq \beta\right\} \geq \alpha.$$

Therefore, by using the test

$$\sum_{p\in\widehat{\Psi}:\bar{h}_{\downarrow p}\in h_{\downarrow p}} \Pr\{arcs(p)\} \geq \beta,$$

a policy tree $\mathcal{T}$ can be proved to satisfy h in $\mathcal{P}$ with probability α. Conversely, let $\delta \leq p_{lb}^{\beta}$. By definition,

$$p_{lb}^{\beta} = \min\{q | \Pr\{\mathit{Bin}(N;q) \geq \mathit{round}(\beta N)\} \geq (1-\widehat{\alpha})/2.$$

Therefore, it is clear that

$$\Pr\{\mathit{Bin}(N;\delta) \geq \mathit{round}(\beta N)\} < 1-\alpha.$$

This means that

$$\Pr\left\{\sum_{p\in\widehat{\Psi}:\bar{h}_{\downarrow p}\in h_{\downarrow p}} \Pr\{arcs(p)\} \geq \beta\right\} < 1-\alpha,$$

which implies

$$\Pr\left\{\sum_{p\in\widehat{\Psi}:\bar{h}_{\downarrow p}\in h_{\downarrow p}} \Pr\{arcs(p)\} \leq \beta\right\} \geq \alpha.$$

Therefore, by using the test

$$\sum_{p\in\widehat{\Psi}:\bar{h}_{\downarrow p}\in h_{\downarrow p}} \Pr\{arcs(p)\} \leq \beta,$$

a policy tree $\mathcal{T}$ can be proved to not satisfy h in $\mathcal{P}$ with probability α. □

Proposition 2 *Any policy tree $\mathcal{T}$ which provides a satisfaction probability $\delta \geq \beta+\vartheta$ wrt h in $\mathcal{P}$ can be proved to satisfy h in $\mathcal{P}$ with probability α. Any policy tree $\mathcal{T}$ which provides a satisfaction probability $\delta \leq \beta-\vartheta$ wrt h in $\mathcal{P}$ can be proved to not satisfy h in $\mathcal{P}$ with probability α.*

Proof: this directly follows from Definition 2 and Proposition 1. □

Proposition 3 *Any policy tree $\mathcal{T}$ which can not be proved to satisfy or to not satisfy h in $\mathcal{P}$ with probability α, can be either proved to satisfy h in $\mathcal{P}$ with probability γ, where γ is a probability ranging in $(0.5, \alpha($, if it satisfies h in $\widehat{\mathcal{P}}_N$, or to not satisfy h in $\mathcal{P}$ with probability γ, where γ is a probability ranging in $(0.5, \alpha($, if it does not satisfies h in $\widehat{\mathcal{P}}_N$.*

Proof: Consider the two limiting cases. (i) The actual satisfaction probability δ provided by $\mathcal{T}$ wrt h in $\mathcal{P}$ is exactly equal to β. Since the sample mean, used to estimate the satisfaction probability out of the N samples considered, is an unbiased estimator of δ, it will overestimate β with probability 0.5 and, similarly, it will underestimate β with probability 0.5; this sets the lower bound for γ. (ii) The actual satisfaction probability δ provided by $\mathcal{T}$ wrt h in $\mathcal{P}$ is exactly equal to $\beta+\vartheta$. From the proof of Proposition 1 it immediately follows that, in this case, $\gamma = \alpha$, and also that, if $\delta < \beta+\vartheta$ then $\gamma < \alpha$; this sets the upper bound for γ.□

Definition 4 *An (α,ϑ)-solution to a SCSP $\mathcal{P}$ is a policy tree $\widehat{\mathcal{T}}$ that at least with probability α provides for every chance-constraint h_i in $\mathcal{P}$ with satisfaction threshold β_i a satisfaction probability greater or equal to $\beta_i - \vartheta$.*

It is apparent that ϑ may be interpreted as a parameter that the user can set in order to define a "region of indifference", i.e. $\beta \pm \vartheta$, for the satisfaction probability. In such a region, we assume that assignments can be safely misclassified with probability greater than α and that satisfaction probabilities remain in an acceptable range.

6.1 Example

Consider the single-stage SCSP $\mathcal{P} = \langle V, S, D, P, C, \beta, L\rangle$, where $V = \{X_1, X_2\}$, $S = \{r_1, r_2\}$, $D(X_1) = D(X_2) = \{0, 1\}$, $D(r_1) = (0, 100)$, $P(r_1) = \mathit{Uniform}(0, 100)$, $D(r_2) = (0, 300)$, $P(r_2) = \mathit{Uniform}(0, 300)$, $C = \{c : C_1 \geq X_1 r_1 + X_2 r_2\}$, $\beta_c = 0.5$, and $L = [\langle V, S\rangle]$. $C_1 = 185$ is a constant. This problem comprises random variables defined on a continuous support and it cannot be solved by existing complete approaches to SCSPs. If we set $\alpha = 0.95$ and $\vartheta = 0.05$, from Definition 2 we compute the number of samples $N = 250$ required to guarantee that any solution to the sampled SCSP $\widehat{\mathcal{P}}$ over N samples is an (α,ϑ)-solution for $\mathcal{P}$.

Furthermore, the simple structure of the constraint c considered in $\mathcal{P}$ allows us to perform some further analysis. Consider the assignment $X_1 = 1$ and $X_2 = 1$. A simple reasoning on the convolution of two independently non-identically distributed uniform random variables (see [Sadooghi-Alvandi *et al.*, 2009]) immediately suggests that this assignment is indeed inconsistent. r_1 and r_2 are two independently non-

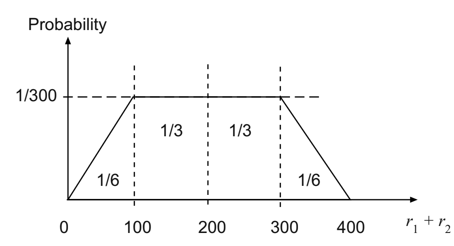

Figure 3: Probability density function of the convolution of two independently non-identically distributed uniform random variables r_1 and r_2.

identically distributed uniform random variables. The distribution that results from their convolution is shown in Fig. 3. This distribution is shaped like a trapezoid. Clearly, since the area for the whole figure must be equal to 1, the area of each of the two rectangle triangles at the side of the trapezoid must be equal to 1/6. Consequently, the area of the internal rectangle must be equal to 2/3. It is easy to see that the cumulative distribution function for value 200 returns a probability of 0.5. Then, since 1/3*(15/100)=0.05, the 0.45 quantile of the inverse cumulative distribution function which results from convoluting r_1 and r_2 is exactly equal to $C_1 = 185$. Therefore, since the satisfaction probability provided by the assignment $X_1 = 1$ and $X_2 = 1$ is equal to $\beta_c - \vartheta = 0.45$, this assignment will be correctly classified as inconsistent with probability α, when the sample size is set to $N = 250$.

6.2 Multiple Chance-constraints

Let $h_1, \ldots, h_k$ be k chance-constraints in a SCSP $\mathcal{P}$. Let $\widehat{\mathcal{P}}$ be a sampled SCSP over N samples, where N is the number of samples required to guarantee a confidence level α and an error tolerance threshold ϑ for each constraint h_i considered independently, according to Definition 2.

Proposition 4 *Let $\widehat{\mathcal{T}}$ be a policy tree that is a solution to $\widehat{\mathcal{P}}$. Then $\widehat{\mathcal{T}}$ is an (α, ϑ)-solution for $\mathcal{P}$.*

Proof: Consider a chance-constraint h_i. Let β_i be the respective satisfaction threshold. By definition, the probability that a solution $\widehat{\mathcal{T}}$ to $\widehat{\mathcal{P}}$ provides a service level less or equal to $\beta_i - \vartheta$ for h_i in $\mathcal{P}$ is less or equal to $1 - \alpha$. Therefore $\widehat{\mathcal{T}}$ is an (α, ϑ)-solution. Now consider a pair of chance-constraints $\langle h_i, h_j \rangle$ with satisfaction thresholds β_i, β_j, respectively. The probability that a solution $\widehat{\mathcal{T}}$ to $\widehat{\mathcal{P}}$ provides a service level less or equal to $\beta_i - \vartheta$ for h_i and to $\beta_j - \vartheta$ for h_j in $\mathcal{P}$ is less or equal to $(1-\alpha)^2$, in fact we must misclassify both the constraints in order to accept such a solution. Even a single constraint correctly classified will make $\widehat{\mathcal{T}}$ inconsistent w.r.t. $\widehat{\mathcal{P}}$. This reasoning can be easily generalized to k chance-constraints, for which the probability becomes $(1-\alpha)^k$. Noting that $(1 - \alpha)^k < \ldots < (1 - \alpha)^2 < (1 - \alpha)$ and that $1 - (1 - \alpha)^k \geq \alpha$ the probability that a solution is misclassified in a model comprising a single constraint, i.e. $(1 - \alpha)$, represents an upper bound for the probability that a solution $\widehat{\mathcal{T}}$ to $\widehat{\mathcal{P}}$ does not provide a satisfaction probability within the required tolerance threshold for one or more constraints in a generic model $\mathcal{P}$. By rephrasing, the probability that a solution $\widehat{\mathcal{T}}$ provides a satisfaction probability greater or equal to $\beta_i - \vartheta$ for each constraint h_i is greater or equal to α. Therefore, by Definition 4, $\widehat{\mathcal{T}}$ is an (α, ϑ)-solution for $\mathcal{P}$. □

Our approach is quite conservative and it will often attain a confidence probability greater than α for a generic model with k chance-constraints, since several assignments will typically violate more chance-constraints. However, when the number of chance-constraints increases, also the probability of misclassifying solutions increases, and we may need a higher confidence α to find one.

7 Conclusions

We proposed a framework for exploiting sampling in order to solve *single-stage* SCSPs that include random variables over a continuous or very large discrete support. Our framework is based on two novel tools: sampled SCSPs and (α, ϑ)-solutions. We employed statistical estimation to determine if a given assignment is consistent with respect to a given set of chance-constraints. As in statistical estimation, the quality of our estimate is determined via confidence interval analysis. In contrast to existing approaches based on sampling, we provide likelihood guarantees for the quality of the solutions found. In fact, we explicitly state a confidence probability α that bounds the probability of exceeding a given error tolerance threshold ϑ in our estimation. By properly choosing the estimation error ϑ and the confidence probability α it is possible to generate compact sampled SCSPs that can be effectively solved by existing solution methods. We demonstrated our approach on a single-stage SCSP that comprises random variables defined on a continuous support and cannot be modeled and solved by complete approaches in the literature.

References

[Agresti and Coull, 1998] A. Agresti and B. A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[Clopper and Pearson, 1934] C. Clopper and E. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[Evans *et al.*, 2000] M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. Wiley, New York, 2000.

[Hnich *et al.*, 2009] B. Hnich, R. Rossi, S. A. Tarim, and S. D. Prestwich. Synthesizing filtering algorithms for global chance-constraints. In Ian P. Gent, editor, *CP 2009, Proceedings*, volume 5732 of *Lecture Notes in Computer Science*, pages 439–453. Springer, 2009.

[Prestwich *et al.*, 2009] S. D. Prestwich, S. A. Tarim, R. Rossi, and B. Hnich. Evolving parameterised policies for stochastic constraint programming. In Ian P. Gent, editor, *CP 2009, Proceedings*, volume 5732 of *Lecture Notes in Computer Science*, pages 684–691. Springer, 2009.

[Rossi *et al.*, 2008] R. Rossi, S. A. Tarim, B. Hnich, and S. D. Prestwich. A global chance-constraint for stochastic inventory systems under service level constraints. *Constraints*, 13(4):490–517, 2008.

[Sadooghi-Alvandi *et al.*, 2009] S. Sadooghi-Alvandi, A. Nematollahi, and R. Habibi. On the distribution of the sum of independent uniform random variables. *Statistical Papers*, 50(1):171–175, January 2009.

[Sethi and Sorger, 1991] S. Sethi and G. Sorger. A theory of rolling horizon decision making. *Ann. Oper. Res.*, 29(1-4):387–416, 1991.

[Tarim *et al.*, 2006] S. A. Tarim, S. Manandhar, and T. Walsh. Stochastic constraint programming: A scenario-based approach. *Constraints*, 11(1):53–80, 2006.

[Walsh, 2002] T. Walsh. Stochastic constraint programming. In Frank van Harmelen, editor, *European Conference on Artificial Intelligence, ECAI'2002, Proceedings*, pages 111–115. IOS Press, 2002.