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 propose 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.

Stochastic Constraint Satisfaction Problems (SCSPs)

An m-stage SCSP [Walsh, 2002; Tarim et al., 2006; Hnich et al., 2009] is defined as a 7-tuple <V, S, D, P, C, β, L> where:

- V is a set of decision variables,
- 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].
- $L = [\langle V_1, S_1\rangle, \ldots, \langle V_i, S_i\rangle, \ldots, \langle V_m, S_m\rangle]$ is a list of decision stages such that each $V_i \in V$, each $S_i \in S$, the V_i form a partition of V, and the S_i form a partition of S.

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, . . . , 4}, the domain of x_2 is {3, . . . , 6}. There are two chance-constraints in C, $c_1 : \Pr\{s_1x_1 + s_2x_2 \geq 30\} \geq 0.75$ and $c_2 : \Pr\{s_2x_1 = 12\} \geq 0.5$.

Policy tree

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 Vm 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.

A possible solution to the SCSP in Example 1 is the satisfying policy tree shown in Fig. 1

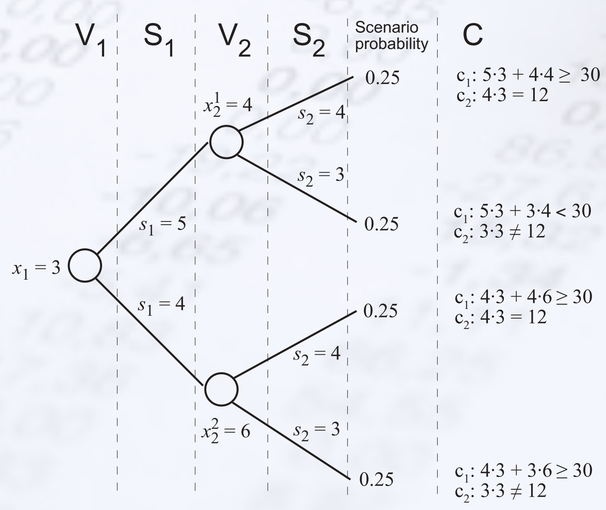

Fig. 1: a satisfying policy tree for the SCSP in Example 1

Sampled SCSPs

Consider a SCSP 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.

We propose to sample a more compact SCSP, which comprises at most N scenarios, out of the original problem and we shall call this new problem "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 Fig. 2 we show a sampled SCSP comprising 3 sampled scenarios for the problem discussed in Example 1.

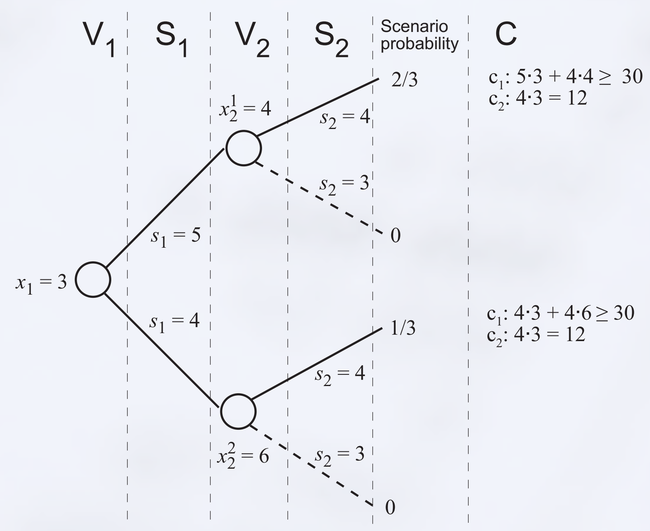

Fig. 1: a satisfying policy tree for the SCSP in Example 1

It should be noted that every satisfying policy tree for a sampled SCSP can be employed as a (partial) policy tree for the original SCSP P. In general, it is possible that this policy tree is a (partial) policy tree that is a satisfying policy tree for 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 over 3 scenarios, 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. In other words, we discuss under which conditions the solution to a sampled SCSP is, with a prescribed confidence probability, likely to be also a solution to the original SCSP.

Our approach exploits sampling in order to keep under control the number 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.

Confidence interval analysis for Binomial proportions

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 p and 1 − p, respectively. Let us assume that the value p — 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, covers the actual value of the parameter that is being estimated.

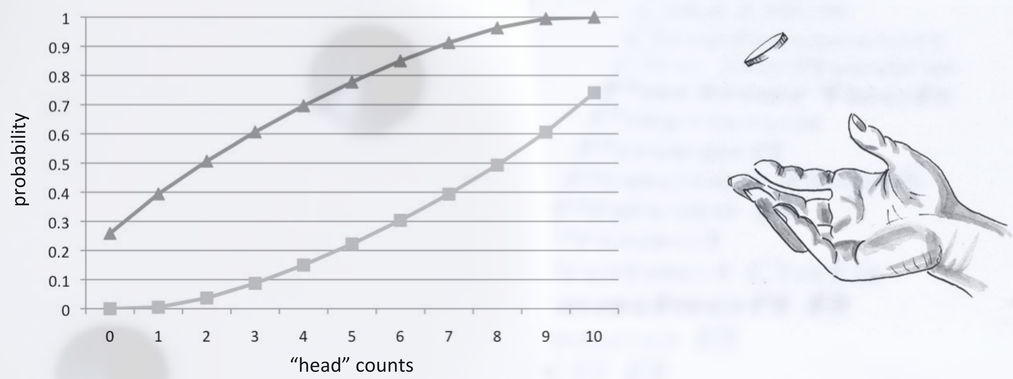

Fig. 2: Clopper-Pearson upper and lower bounds for different "head" counts over 10 tosses

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

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

X is the number of successes (or "yes" events) observed in the sample, 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 $p_{ub} = 1$ when X = N.

Properties of SCSPs solutions

We will now characterize the probability that the solution of a sampled SCSPs over N scenarios is a solution to the original single-stage SCSP P. We firstly discuss what the minimum value for N is, in order to achieve a predefined probability α that a given policy tree that satisfies a chance-constraint h in the sampled SCSPs also satisfies the same chance-constraint in the original SCSP. Since a policy tree for the sampled SCSP by definition only comprises a subset of all the paths that constitute a policy tree for the original SCSP P, this policy tree, in order to satisfy h in the original SCSP, this policy tree 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 ±ϑ. The number of scenarios N for the sampled SCSP depends on α, ϑ 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 $\max(p_{ub} - \beta, \beta - p_{lb}) \leq \vartheta$, where (p_{lb}, p_{ub}), is the Clopper-Pearson confidence interval for a confidence probability 2α − 1, and round(N) "successes" in N trials; round() approximates the value to the nearest integer.

Definition 4: An (α,ϑ)-solution to a SCSP P is a policy tree that at least with probability α provides for every chance-constraint h_i with satisfaction threshold β_i a satisfaction probability greater or equal to β_i-ϑ in the original SCSP.

It is apparent that ϑ may be interpreted as a parameter that the user can set in order to define a "region of indifference" 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.

It can be proved that any solution to a sampled SCSP over N scenarios, where N is computed according to Definition 2, is an (α,ϑ)-solution for the original SCSP.

Example 2 (Newsvendor problem). Consider the following SCSP; P = <V, S,D, P,C, β, L>, where V ={Q}, S = {d}, D(Q) = {0, . . . , 200}, D(d) = (0, 200), P(d) = uniform(0, 200), C = {c : Pr{Q ≥ d} ≥ 0.5} and L = [<V, S>]. A (trivial) analytical solution for the above problem is Q ≥ 100. Consider α = 0.9 and ϑ = ±0.15. This directly implies that N = 24, since this is the minimum N that guarantees $\max(p_{ub} - \beta, \beta - p_{lb}) = \max(0.64 - 0.5, 0.5 - 0.35) \leq 0.15$. By solving a sampled SCSP for N = 24 we obtain an (α,ϑ)-solution for the original SCSP. The reader may easily note that not all the values in the domain of Q can be proved to satisfy or to not satisfy the chance-constraint with the prescribed confidence probability α. More specifically, only values in {0, 70} and {130, 200} can be proved to not satisfy and satisfy the chance-constraint, respectively, with confidence probability α (Fig. 3). However, every value smaller than 70 will be correctly classified with confidence level greater or equal to α. In other words, assignments that provide a satisfaction probability less or equal to β-ϑ in the original SCSP, will be correctly classified as infeasible with probability greater or equal to α. Hence, the solution of the sampled SCSP is is an (α,ϑ)-solution for the original SCSP.

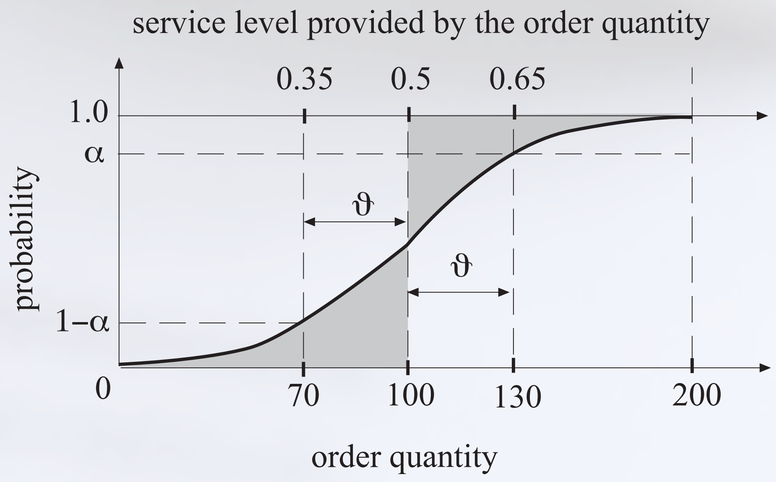

Fig. 3: probability of classifying as feasible a given order quantity Q by solving the sampled SCSP in our example. The error probability is denoted by the area in gray. The shape of the function can be derived by using Chernoff's inequality