Stochastic Constraint Optimisation with Applications in Network Analysis

Anna L. D. Latour[1] *, Behrouz Babaki[2], Daniël Fokkinga[1], Marie Anastacio[1], Holger Hoos[1,3], and Siegfried Nijssen[4]

[1] Leiden University, Leiden, The Netherlands
[2] Polytechnique Montréal, Montreal, Canada
[3] University of British Columbia (UBC), Vancouver, Canada,
[4] UCLouvain, Louvain-la-Neuve, Belgium

Abstract

Stochastic Constraint (optimisation) Problems (SCPs) are problems that combine weighted model counting (WMC) with constraint satisfaction and optimisation. We present an extensive study of methods for exactly solving SCPs in network analysis, where the underlying probability distributions have a monotonic property. These methods use knowledge compilation to address the model counting problem; subsequently, either a constraint programming (CP) solver or mixed integer programming (MIP) solver is used to solve the overall SCP. To configure the space of parameters of these approaches, we propose to use the framework of programming by optimisation. The result shows that a CP-based pipeline obtains the best performance.

Summary

This is an extended abstract of an earlier publication at IJCAI 2019 [7] and work presented at the 2019 Data Science meets Optimisation workshop [3], augmented by work in progress.

Making decisions under uncertainty is an important problem in business, governance and science. We find examples in the fields of planning and scheduling, as well as in fields like data mining and bioinformatics.

Many of these problems can be formulated on *probabilistic networks*. Examples are social networks [1], where we are uncertain about how likely people are to adopt ideas from others, and power grid networks [2], where we are uncertain about the reliability of power lines. The uncertainty can be represented by associating probabilities with edges or nodes in the network. Consider the problem of *spread-of-influence* in more detail [5]. Given a social network of people (vertices) that have stochastic relationships (edges), we want to use word-of-mouth advertisement to turn friends of our customers into new customers. We start this spread-of-influence campaign by distributing at most k free product samples to members of the network. Each node may convince a neighbor with a probability indicated on the edge towards its neighbor. What is the k-sized set S of most influential nodes in this network?

This is an instance of a general class of problems, called *stochastic constraint (optimisation) problems* (SCPs). SCPs have the following characteristics: (1) they involve (Boolean) random variables and (Boolean) decision variables; (2) they require finding assignments to decision variables, such that an optimisation criterion or constraint on the random variables is satisfied; (3) they may involve other constraints on the decisions we can take. The example above has a fourth characteristic: (4) the optimisation criterion requires the calculation of an expectation over the probabilistic variables given an assignment to the decision variables, where the expectations are higher if more nodes or edges are selected; this makes these probability distributions *monotonic* and makes the problem an instance of a *stochastic constraint (optimisation) problem on monotonic distributions* (SCPMD). While (4) seems

*Corresponding author, a.l.d.latour@liacs.leidenuniv.nl.

limiting, problems with this characteristic are plentiful. In the example above, adding nodes to S cannot decrease the expected number of eventual customers. This is an intuition for the meaning of monotonicity in this context.

SCPs are NP-hard, as they generalise NP-hard constraint satisfaction and optimisation problems.

The perspective we take is to treat SCPs as problems that combine *weighted model counting* (WMC) with constraint satisfaction and optimisation: we look for assignments to decision variables, such that a weighted model count is either optimised or satisfies a given constraint. Hence, we treat calculating a probability or expectation as a WMC problem.

We evaluate two methods for solving SCPs exactly. They have in common that they use *knowledge compilation* to address the WMC problem. The challenge is how to incorporate the results of the knowledge compilation in a subsequent step that solves the overall SCP.

The first method converts *decision diagram* encodings of probability distributions into *arithmetic circuits* (ACs) that are used to compute conditional probabilities through WMC. A constraint on a distribution translates to a constraint on the AC. This constraint is then *decomposed* into a multitude of local constraints that are solved by an off-the-shelve CP or MIP solver [6]. This *decomposition method* is solver-agnostic and straightforwardly implemented. However, some relations between variables are lost during decomposition, causing the method to prune the search space inadequately. Specifically, it does not guarantee *generalised arc consistency* (GAC).

The second method preserves GAC by introducing a *global* constraint for SCPMDs (whose underlying probability distributions are *monotonic*). We propose and implement a constraint propagator for this *stochastic constraint on monotonic distributions* (SCMD) that preserves relations between variables. Because the monotonicity of the distribution is crucial to preserving GAC, this *global constraint method* is less general than decomposition.

For both methods, we followed the paradigm of *programming by optimisation* [4] and implemented alternative *design choices*. We automatically configured the resulting solver on problem instances from two different domains and compared the PAR10 values[1] of the resulting optimised methods. We find that the global method performs comparably to a MIP-based decomposition method for smaller instances (a closer look shows that these two methods tend to perform complementarily here), but outperforms this variant of the decomposition method for larger instances. The optimised global method outperforms the CP-based decomposition method by up to two orders of magnitude.

References

[1] Domingos, P.M., Richardson, M.: Mining the network value of customers. In: KDD. pp. 57–66. ACM (2001)

[2] Dueñas-Osorio, L., Meel, K.S., Paredes, R., Vardi, M.Y.: Counting-based reliability estimation for power-transmission grids. In: AAAI. pp. 4488–4494 (2017)

[3] Fokkinga, D., Latour, A.L.D., Anastacio, M., Nijssen, S., Hoos, H.: Programming a stochastic constraint optimisation algorithm, by optimisation. In: Workshop Data Science meets Optimisation (DSO), IJCAI 2019 (aug 2019)

[4] Hoos, H.H.: Programming by optimization. Commun. ACM **55**(2), 70–80 (2012)

[5] Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD. pp. 137–146. ACM (2003)

[6] Latour, A.L.D., Babaki, B., Dries, A., Kimmig, A., Van den Broeck, G., Nijssen, S.: Combining stochastic constraint optimization and probabilistic programming — From knowledge compilation to constraint solving. In: CP. Lecture Notes in Computer Science, vol. 10416, pp. 495–511. Springer (2017)

[7] Latour, A.L.D., Babaki, B., Nijssen, S.: Stochastic constraint propagation for mining probabilistic networks. In: IJCAI. pp. 1137–1145. ijcai.org (2019)

[1] Penalised average running time: every instance that reaches the cutoff time in an experiment is treated as if its running time was ten times that cutoff time.

External sources

This is an extended abstract of the following earlier work:

A.L.D. Latour, B. Babaki, S. Nijssen, *Stochastic constraint propagation for mining probabilistic networks.* IJCAI 2019, which can be found here: www.ijcai.org/Proceedings/2019/159. The corresponding code for this publication is available at github.com/latower/SCMD.

D. Fokkinga, A.L.D. Latour, M. Anastacio, S. Nijssen, H. Hoos, *Programming a stochastic constraint optimisation algorithm, by optimisation.* Workshop Data Science meets Optimisation (DSO), held in conjunction with IJCAI 2019, which can be found here: ada.liacs.nl/papers/FokEtAl19.pdf. The corresponding code for this publication is available at ada.liacs.nl/projects/scop-solver.