Review of the PhD thesis of M. Karpiński "CNF Encodings of Cardinality Constraints Based on Comparator Networks"

June 14, 2019

Background

The PhD thesis of Mr Karpiński deals with constraint programming, an approach to programming that relies on a combination of techniques dealing with reasoning and computing. It has been successfully applied several areas including molecular biology, electrical engineering, operations research, and numerical analysis.

The central notion in this approach to programming is a *constraint*, which in a most general setting is just a relation on the domains of some variables. Best known and most often used are Boolean constraints. The reasoning part of constraint programming is concerned with *constraint solving*. In the case of Boolean constraints these are the SAT solvers. Even though SAT solving is an NP-complete problem, the use of SAT solvers turned out to be spectacularly successful and even new theorems were established using it, the most famous being the *Boolean Pythagorean triples problem* (see M.J.H. Heule, O. Kullmann, and V. Marek, "Solving and Verifying the Boolean Pythagorean Triples problem via Cube-and-Conquer", in Proceedings of Theory and Applications of Satisfiability Testing, SAT 2016, pp. 228–245, 2016).

This explains the intense research that aims at improving the techniques for solving Boolean constraints. These improvements are the driving force behind the implementations of improved SAT solvers. One of the important threads in this research is search for natural constraints that go beyond the customary syntax of Boolean formulas and are convenient for expressing various often occurring problems, for example various types of timetabling. The challenge here is to find an efficient way of encoding such constraints.

One constraint that was identified as especially helpful is the *cardinality constraint* and its Boolean version. It states that at most (or exactly, at least, etc.) k out of n literals in a Boolean formula are true. The subject of this thesis is an in-depth study of this constraint, notably by modelling it by means of comparator networks. In what follows I shall discuss the thesis in some detail.

Contents of the Thesis

The thesis consists of eight chapters. **Chapter 1** is of an introductory character and consists of an account of the history of SAT solving and an exposition of Constraint Programming. The latter is admittedly very brief, which is understandable given that the focus of the thesis are Boolean constraints and not Constraint Programming in general. The author does provide several relatively recent examples of successful applications of Constraint Programming in several new areas, for example in the field of ecology.

The presentation in this chapter is lucid, well organized and informative. It is clear that the author is very well informed about the state of the art and is well familiar with the employed techniques and approaches.

Chapter 2 is devoted to an introduction of the main concepts around which the thesis evolves. These are in particular various natural relations on sequences of elements, notably sequences of 0s and 1s, such as the *weak domination* relation that plays an important role in the subsequent chapters. The crucial notions for the results established in the thesis are that of a *bitonic sequence*, a circular shift of a $\wedge$-shaped sequence of values, and of a *comparator network* acting on a sequence of elements, which is a composition of functions each of which just swaps two elements in a sequence. Its compact, mathematical (i.e., functional), definition is followed by two alternative ones, declarative and procedural, and by an introduction of a modification of it called a *generalized selection network* (GSN). The rationale for introducing comparator networks and generalized selection networks is that they are used to obtain a clausal encoding of the cardinality constraints.

The remainder of this chapter is devoted to an exposition of the notion of arc-consistency that is crucial in constraint programming. My main complaint here is that what the author calls arc-consistency should be rather called hyper-arc consistency (see Chapter 5 of my book *Principles of Constraint Programming*, Cambridge University Press, 2003). Arc consistency deals with binary constraints, while hyper-arc consistency is concerned with n-ary constraints. I should add that the same confusion appears in various cited papers.

Here, for the record, is the definition of hyper-arc consistency for an arbitrary constraint. A constraint C on n variables is *hyper-arc consistent* if every value in every variable domain is used in a solution to C. Also, the author chose to define this notion for the cardinality constraint instead of for an arbitrary constraint. As a result, the definition is verbose and it takes

some time to understand that it coincides with the simple one given above. It would be more natural to introduce the adopted definition in a simple lemma showing that it is equivalent to hyper-arc consistency.

The chapter also provides a first result of the thesis, that (hyper-)arc consistency of the Boolean cardinality constraint is preserved by any standard encoding of the GSNs based on the Boolean formulas. Such an encoding consists of layers, each layer in turn consisting of a sequence of selectors.

As already mentioned, the cardinality constraint was found to be most useful in several important applications of Constraint Programming. This explains a large number of publications devoted to a modeling of it by means of Boolean formulas, so that a SAT solver can be used to solve the original constraints. **Chapter 3** is devoted to a detailed account of the encodings used in the literature, including an account of the size of the obtained encodings. The encodings use a variety of concepts and techniques, including adders, binary trees, counters, sorting networks, and selection networks. It turns out that some of these encoding do not preserve hyper-arc consistency.

The second part of the chapter deals with the at-most-one-constraint and pseudo Boolean constraints, for which additional techniques were used, including binomials, binary decision diagrams (BDDs), and Reduced Order BDDs. The last two also proved to be useful in the area of automated theorem proving.

The author also clarifies that the selection networks became a most promising approach. This explains the focus of this thesis and justifies its importance: any improvement in the efficiency of the encoding has a direct beneficial effect on the size of applications one can tackle using the cardinality constraints.

The chapter does not contain any results, but the presentation is excellent and results in an insightful survey of the literature. The main techniques are explained by means of examples, which makes the reading pleasant.

Chapter 4 is devoted to a construction of a new encoding of the cardinality constraint based on selection networks. It uses a class of pairwise selection networks introduced in 2012 by Codish and Zazon-Ivry. These networks are modified to bitonic selection networks that use as black boxes some networks used to sort bitonic sequences. Subsequently the latter networks are further optimized to pairwise bitonic selection networks. Corresponding results show that the resulting networks remain correct in the sense that the outputs have the k largest elements correctly sorted (are top k sorted). Detailed calculations provide the bounds on the size of these networks which

allows the author to show an improvement w.r.t. the networks used by Codish and Zazon-Ivry.

The results of this chapter hold for any domain, not only $\{0, 1\}$. On the other hand, they assume that n (the number of variables) and k (the constant of the 'at most k' qualification) are the powers of 2.

Chapter 5 is concerned with another class of networks, called *m-wise selection networks*. This class exploits insights from the construction of the pairwise sorting networks by focusing on a concept of an *m-wise sequence* that consists of (appropriately chosen) m sorted sequences that are also componentwise pairwise sorted. The construction of these networks is combined with the construction of the *m-wise merging networks*, or more precisely with the algorithm that generates a permutation of the input sequence that is top k sorted. Given the complexity of the problem the construction is given only for $m = 4$. A detailed analysis of the provided algorithm yields the desired conclusion that the output is indeed as claimed.

I miss here an explanation what difficulties arise for values of m different from 4, though it is clear that the considered algorithm is involved and difficult to analyze.

The chapter ends by providing a favourable comparison in terms of the number of variables with the pairwise selection networks. In contrast to the previous chapter, the results presented here hold for all values of n and k but assume that the domain is $\{0, 1\}$.

Chapter 6 is devoted to a presentation of an odd-even selection network that is an improvement on (an improvement) of the original sorting network proposed by Batcher and Lee in 1995. More precisely, the author introduces 4-odd-even selection networks that are combined with corresponding merging networks and subsequently provides algorithms that represent them. The number 4 corresponds to the number of columns into which the input sequence is arranged. The relevant result is that the output of the proposed algorithm is k sorted. It is also shown that the encoding achieved by this algorithm leads to less variables than the one based on the networks used by Codish and Zazon-Ivry. The final section of the chapter explains how to generalize this idea of a network to other values than 4.

These results are complemented by a comparison of the implementation of the approach proposed in this chapter with three other encodings of the cardinality constraints used in the literature. It is carried out on the inputs coming from a library of benchmarks for the Pseudo-Boolean constraints using a state-of-the-art SAT solver that won numerous SAT competitions. The

favourable outcome of these comparisons provide a much needed justification for a study of the proposed selection networks.

Chapter 7 discusses another implementation of the solver based on the 4-odd-even selection networks of the previous chapter and provides an experimental evaluation of it by means of a comparison of it with the state-of-the-art solvers for the Pseudo-Boolean constraints. The (favourable) comparison is based on more recent library of benchmarks, probably because this work was done later than the one reported in the previous chapter.

Finally, **Chapter 8** provides a short summary of the results of the thesis, combined with some speculations about the effectiveness of the comparator networks for encoding the cardinality constraints.

Conclusions and Recommendation

The thesis is very well and clearly written. The introductory chapters, notably Chapter 3, are most helpful in assessing the contributions of the thesis. I always recommend to my students that each definition should be followed by a useful example. So I am pleased that Mr Karpiński adopted this style in most of the chapters of the thesis. Each chapter starts with an insightful introduction and ends with a clear summary of the results.

Also English is excellent. I noticed only a couple of mistakes (like the reference to a 'fix point' should be replaced by a 'fixpoint' or a 'fixed point' and 'are build' should be 'are built').). A small remark about the references: reference [78] (crucial for the results of the thesis) is incomplete.

The obtained results are original and relevant. They are based on insights from the area of comparator networks and are applied in the area of Constraint Programming. Some of these results were already published in a journal (Theoretical Computer Science) and conferences, including the main conference on Constraint Programming that is very competitive. This confirms my highly positive opinion about this thesis.

The subject of Chapter 6 looks at first sight like a trivial generalization of an existing concept (by using 4 and not 2 columns). However, both the theoretical analysis (Theorem 6.3) and the implementations discussed in Chapters 6 and 7 reveal that simple ideas can have far reaching consequences.

One topic that could have been mentioned in the thesis is the issue of constraint propagation that is central to Constraint Programming. In the context of this thesis hyper-arc consistency is relevant (and is regularly men-

tioned). Therefore the algorithms achieving it, i.e., hyper-arc consistency algorithms, have a natural place here. Possible improvements in this area could be achieved by proposing new propagation algorithms tailored to the considered constraints.

The author is familiar with the rich literature on this subject. The proofs are rigorous and well organized. Some of them are pretty involved and tedious, yet they are clearly presented. In conclusion I find that the manuscript presented by Mr Karpiński qualifies as a PhD thesis and recommend that it be accepted.

Prof. Dr. Krzysztof R. Apt
Centrum Wiskunde en Informatica (CWI), Amsterdam
and MIMUW, University of Warsaw.

K. R. Apt