The Fundamental Conjecture of Reformulation *

Ian P. Gent

School of Computer Science, University of St. Andrews, Fife KY16 9SS, United Kingdom. ipg@dcs.st-and.ac.uk

Abstract. I conjecture that there may always be an optimal encoding from one domain into another in the sense that propagation is achieved optimally in the new encoding with no degradation of time complexity.

A number of pieces of recent work by various members of the APES research group have a similar theme. Stergiou and Walsh have shown that the hidden variable encoding of n-ary CSP's into binary CSP's can achieve generalised arc consistency (GAC) in the same time as the optimal general GAC algorithm [Ste01]. Irving, Manlove, Prosser, Smith and I have shown that there is a boolean encoding of the Stable Marriage problem in which propagation can work in the same time and act as powerfully as the specialist Gale-Shapley algorithm from the Stable Marriage community [GIM$^+$01]. Smith has shown that the occurrence constraint can be reasoned with as efficiently as via Regin's algorithm, again using a boolean encoding [Smi01]. Finally, I have shown that CSP's can be translated into SAT and unit propagation used to establish arc consistency in optimal time [Gen01].

The common theme of these pieces of work is that with a cunning encoding, and standard propagation techniques, we can achieve full generalised arc consistency without specialised propagation algorithms and in the same time as the specialised algorithm. This, combined with discussion with Christian Bessiere, Patrick Prosser and Kostas Stergiou, has suggested to me that there may *always* be an encoding with these properties.

Conjecture 1. **(Fundamental Theorem of Reformulation)** Suppose that a propagation algorithm for a constraint runs in time $O(p(n))$ for input size n. Then a specific instance of the constraint can be reformulated in time at most $O(p(n))$ to a set of constraints for which propagation can be performed by a non specialised arc-consistency algorithm in time $O(p(n))$ establishing the same level of consistency as the given algorithm.

The presentation of Cook's theorem in Garey & Johnson [GJ79] goes some way towards a proof. The idea of Cook's theorem is that a Turing machine is converted to SAT clauses. If the set of SAT clauses was of size $O(p(n))$, it would only be necessary to show that unit propagation (or perhaps arc consistency)

* I thank members of the APES research group, of which I am a member, for helpful discussions: www.dcs.st-and.ac.uk/~apes. I am especially grateful for helpful discussions with Christian Bessiere and David Mitchell.

could make the deductions necessary. The algorithm must at some point remove values, and by identifying these steps within the computation, we could work out the SAT variables to tie to variables in our original formulation, guaranteeing the desired consistency strength. Careful analysis would be necessary to guarantee time optimality, or at worst to revise the conjecture to weaken the time claims, and to guarantee that unit propagation is enough. Unfortunately, the SAT encoding of Cook's theorem is too large for our purposes, because the encoding is of size $O(p(n)^4)$. Even this is positive, because it suggests that we can get 'reasonably' close to optimal time via standard propagation.

It is possible the literature has smaller encodings for Cook's theorem,[1] or some more subtle arguments might be necessary: for example there might be an existence proof of the conjecture without a construction to create the encoding from. It must be said that the conjecture is in danger of being trivial. If we allow 'propagation' to include the act of performing Turing Machine calculations, then clearly the theorem is true: the reformulation of a problem would just be the Turing Machine description of the specialised algorithm. I interpret 'non specialised arc-consistency algorithm' to mean something like AC5, or unit propagation in SAT. Therefore there is scope for the conjecture to be false. The real question is: what richness of language is necessary to make the conjecture provable?

Why would the theorem be fundamental if true? Theoretically, it would make specialised consistency algorithms unnecessary, although in practice that would never be true. More practically, an encoding can be worked on by any solver, and it's usually easier to present an encoding to a problem than to implement propagation methods. For example, purely boolean encodings can be given easily to any SAT solver in a standard format. In general, it would show that reformulation can do as well as it is possible to do, encouraging us in working on that topic. One interesting area would be 'propagation compilation': given a specification of a constraint problem, can we automate the discovery of efficient encodings which would be better than just throwing AC5 at the problem?

References

[Gen01] I.P. Gent. An optimal SAT encoding of arc consistency in binary CSP's. In *Proceedings of Formul '01 Workshop*, November 2001.

[GIM⁺01] I.P. Gent, R.W. Irving, D.F. Manlove, P. Prosser, and B.M. Smith. A constraint programming approach to the stable marriage problem. In T. Walsh, editor, *Proceedings of CP-2001*, LNCS. Springer, 2001.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness.* W H Freeman, 1979.

[Smi01] B.M. Smith. Boolean representation of occurrence constraints. Personal communication, February 2001.

[Ste01] K. Stergiou. *Representation and Reasoning with Non-Binary Constraints.* PhD thesis, University of Strathclyde, 2001.

[1] David Mitchell tells me that the best to date is $O(n \log n)$