Problem Reformulation and Search

Final Report

EPSRC Grant GR/M90641/02

Patrick Prosser, Ian P. Gent and Barbara M. Smith

1 Background

Constraint programming (CP) is a commercially successful technology, but it is also an emerging technology. We are still learning how best to model problems in terms of variables and constraints, and to understand the impact of changes in the model on performance of search algorithms and heuristics. Our research aimed to explore the effects that can be brought about by reformulating specific problems, and to determine what characteristics of a problem impact on performance. A general study of representation of constraint satisfaction problems was considered to be a far too ambitious project. Therefore we proposed to investigate three classes of problems: vehicle routing, scheduling, and car sequencing.

To study the vehicle routing problem and scheduling problem we used commercial toolkits, ILOG Dispatcher and Scheduler, provided free by our industrial partner ILOG. The goal was to determine what features of a problem impinged on performance of the toolkit. For example, real world problems are very rarely *pure*, and are typically messy with side constraints: the pure jobshop scheduling problem has never been spotted in the wild, but only in academic labs. The same holds for the vehicle routing problem with time windows. Benchmark problems exist, but again they are pure and unrealistic. We planned to determine how messy we could make a problem before it couldn't really be treated as a pure problem, and in the extreme, when that problem may best be represented as coming from an entirely different class. This information would obviously be useful when fielding applications or developing tools. This part of the study we refer to below as *reformulation in the large*. This was an expensive study, consuming a large part of our resources. Numerous problem generators were produced, some using real geographic data for vehicle routing problems. The experiments were costly to perform, some taking weeks of cpu time. The results of this study were publications, problem generators, problem encodings, and technology transfer back into ILOG.

Our studies of vehicle routing and scheduling problems also lead us to studies of *reformulation in the small*, i.e. fine grained reformulations of problems. Our first such reformulation was brought about by an attempt to *compress* the traveling element of vehicle routing problems, such that they might look more like scheduling problems. This resulted in a number of new graph transformations. We also looked at the effects that can arise when making small changes to logically equivalent encodings, and to determine if these effects were consistent across a number of constraint programming toolkits. The results of this study had an impact on at least one of the toolkits used, choco.

After unpromising initial studies, we decided not to pursue our research on the car sequencing problem. The introduction of a new sequencing constraint into ILOG Solver meant that the dual models we had intended to investigate no longer looked likely to give any additional benefit. Instead we had an opportunity to study a new problem for constraint programming, the stable marriage problem. We were able to prove useful properties of our encoding of the problem as a constraint satisfaction problem. We then proposed a radically different encoding, not of the problem, but of the behaviour of an optimal algorithm for this problem. This led us to other formulations of the same problem into SAT, and the first practical complete algorithm for a hard variant of the stable marriage problem.

This idea of the *formulation of an algorithm* was a surprise, and turned into a significant theme of our work. We repeated this by modeling a newly reported algorithm for an important class of constraint used in symmetry breaking, and also by modeling propagation algorithms for arc consistency in SAT.

Another unforeseen theme was *formulations of new problems*. As well as variants of stable marriage, we provided the first encoding in constraints of supertree construction, an important biological problem.

Our industrial partners were of great benefit to this project. They provided us with valuable software. They also gave us access to their development team, and arranged project meeting in their Paris offices, bringing together their entire team for two days of meetings (in excess of 24 man days). Many of the publications that resulted from this work are co-authored with J. Chris Beck, an ILOG employee now at the Cork Constraints Computation Centre (4C).

2 Key Advances and Supporting Methodology

We investigated the effects of gross reformulation of vehicle routing problems (VRP) and job shop scheduling problems (JSP), i.e. *reformulation in the large.* We also investigated the effects of *reformulation in the small*, in particular on 0/1 encodings and graph transformations on the VRP. Study of stable marriage problem lead us to a novel reformulation, i.e *reformulation of algorithms.* Finally, we also provided *formulations of new problems* in constraints.

2.1 Reformulation in the large

In an industrial setting routing problems and scheduling problems are typically solved with different toolkits. In our study we aimed to identify the problem characteristics in scheduling and vehicle routing problems that make these specialised tools necessary. We used two commercial toolkits, ILOG Dispatcher for VRPs, and ILOG Scheduler for JSPs. These toolkits were used *out of the box*, that is we made no attempt to modify or specialise the toolkits so that they would better fit the problems we were studying. We started by generating benchmark VRPs with time windows and capacity constraints, i.e. vanilla VRPs. These were solved with the VRP toolkit Dispatcher, then modeled as scheduling problems and solved again with ILOG Scheduler. Symmetrically, benchmark jobshop scheduling problems were solved with ILOG Scheduler, then modeled as VRPs and solved with ILOG Dispatcher. At this initial stage we were confirming that these toolkits were indeed fitted to the different classes of problems [8, 3]. Five problem characteristics were identified that explained the difference in performance between VRP and scheduling technology:

- Alternative Resources: In vehicle routing there are typically many vehicles that can be used to perform a visit. In the factory scheduling problem, resources tend to be specialised so that there are few alternatives for any given activity. By decreasing the number of vehicles that can perform a visit the VRP may become more like a scheduling problem and better suited to a scheduling toolkit. Conversely, as more alternative resources become available on the shop floor, scheduling problems may become more VRP-like and better solved with routing technology.
- Temporal Constraints: In pure VRPs each visit is independent, i.e. there are no constraints requiring a visit to have temporal relations with other visits. However, temporal relations between visits do arise. In a study of workforce management for British Telecom [7] service engineers are routed to customers to install and repair equipment. In this domain, temporal constraints are ubiquitous: certain installation tasks must be done in different locations in sequence, or sometimes simultaneously (such as end to end tests). In contrast, scheduling problems typically have long chains of temporal relationships between activities. As we add temporal constraints to VRPs they may become more like scheduling problems, and as we remove sequencing constraints in scheduling problems they may become more like VRPs.
- Ratio of Operation Duration to Transition Time: In pure VRPs a visit has no duration and in pure JSPs the transition time between operations is zero. In both the literature and the real world there are VRPs with visit durations and scheduling problems with transition times and costs. Even then, the ratio of operation duration to transition time tends to be different: in VRPs the ratio is very small while in scheduling it is often large.
- Optimization Criterion: In a VRP the standard objective is to minimise the total distance traveled by each vehicle (and a secondary objective is to minimise vehicles used). In scheduling, it is common to minimise makespan: the time between the start of the first operation and the end of the last operation. The toolkits tend to specialise on the optimization criterion. However, there are VRP instances where all visits must be performed as quickly as possible (i.e. minimising makespan) and in scheduling problems there are instances where we are to minimise expensive setup costs.
- Temporal Slack: the difference in length between an operations time window and its duration, can affect the performance of the search techniques in both routing and scheduling. Slack can be important while solving both routing and scheduling problems, e.g. there are state-of-the-art global constraint propagation algorithms and efficient search heuristics based on temporal slack.

We varied each of these parameters in turn for both VRPs and scheduling problems, to determine how sensitive the technologies were to problem characteristics. Figure 1 pictorially summarises our study. The pure VRP and pure JSP domains are shown as circles. Problem transformations are depicted by arrows labeled with the corresponding problem parameters. We also show the response to each parameter

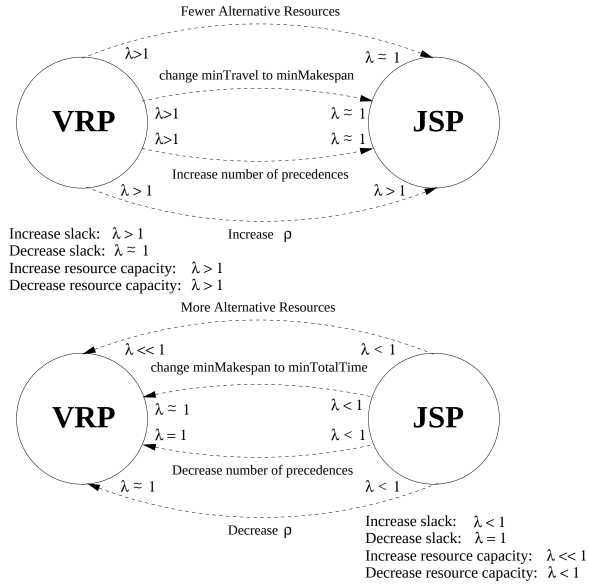

Fig. 1. Influence of various problem characteristics on the performance of VRP and JSP technologies. Arrows represent reformulation direction, arrow labels show the varied parameter. $\lambda > 1$ means routing technology outperforms scheduling technology, when $\lambda < 1$ scheduling technology is a winner, $\lambda \approx 1$ shows that both technologies perform equally. NOTE: the parameter ρ is the ratio of visit duration to travel time for the VRP or the ratio of activity duration to transition time for the JSP. This parameter is used to set the speed of the vehicles or the speed of transitions on machines. Resource capacity in a VRP is the maximum number of visits that a vehicle can perform in any solution, and in a scheduling problem resource capacity is the number of activities that can simultaneously use a resource. This diagram was taken from [4].

change in the relative quality λ of the solving techniques, where $\lambda = C_{sched}/C_{rout}$ where C_{sched} and C_{rout} are the costs of the best solutions found by the scheduling and routing technologies in a set amount of CPU time (10 minutes). The most prominent effects on the performance of the solving techniques were brought about by varying:

the number of alternative resources p**:** The removal of alternative vehicles in the VRPs favours the scheduling technology but the addition of alternative machines to the JSPs proves even less amenable to the routing technique than pure JSPs;

the number of temporal constraints between activities pc**:** An increase in pc in VRPs favours the scheduling technique, and, symmetrically, a decrease in pc in the JSP improves the performance of the routing technique;

optimisation criterion c**:** Changing c from total travel minimisation to makespan minimisation in VRPs notably favours the scheduling technology. Symmetrically, changing c from makespan minimisation to total processing time minimisation in JSPs favours the routing technique.

The results of the experiments are reported in [5] and [4] and the programs, problem generators, benchmarks and results are on the project web site [19].

2.2 Reformulation in the small

As noted above, one difference between VRP and JSP is the ratio of operation duration to transition times: small for VRP and large for JSP. A core component of solving each problem can be modeled as finding a minimum cost Hamiltonian path in a complete weighted graph. The graphs extracted from VRPs and JSPs have different characteristics, however, notably in the ratio of edge weight (travel between visits or setup costs between activities) to node weight (activity durations). We proposed and investigated five transformations for complete weighted graphs that preserve the cost of Hamiltonian paths. These transformations are based on increasing node weights while reducing edge weights, or the inverse. We demonstrated how the transformations affect the ratio of edge to node weight and how they change the relative weights of edges at a node [2]. These transformations can be used to *compress* VRPs such that they might look more like scheduling problems and thus become more suitable for scheduling solvers, and *stretch* scheduling problems with setup costs so that they become more like VRPs.

We also looked at the effect of reformulation of 0/1 problems, involving such relations as implication and the bi-conditional. We used a variety of problems, such as balanced incomplete block designs and maximal independent sets. Two constraint programming toolkits were used, ILOG Solver and choco (a freely available solver [18]). We showed that two logically equivalent encodings, formulated in subtly different ways, could have very different performance. Furthermore, the effect was not consistent across the toolkits. Our results were reported in [20] and at a meeting with ILOG at Paris. choco was modified as a result of this work, and the problem set was added to the choco benchmark suite.

2.3 Reformulation of Algorithms

In the stable marriage problem (SM), we have n men and n women. Each man ranks the women into a preference list, and each woman ranks the men into a preference list. The problem is then to marry men to women so that they are stable, i.e. there is no incentive for individuals to divorce and elope. There is a polynomial time algorithm for this problem, the Gale-Shapley algorithm. We produced a constraint encoding of the problem, where the constraints were expressed extensionally as sets of disallowed tuples, and proved that enforcing arc-consistency on this encoding produced the same end result as an execution of an extended version of the Gale-Shapley algorithm. With Rob Irving and David Manlove we produced a constraint encoding of the algorithm, such that propagating the constraints is equivalent to a proposal/marriage/divorce step in the Gale-Shapely algorithm. We proved that this encoding was optimal [13]. Our study showed that the CSPs resulting from our encoding are tractable, although they did not fall into any of the known tractability classes. Recent work by Green and Cohen [10] has explained this.

We applied a similar encoding to the (patented) algorithm for maintaining lexicographic ordering between 0/1 vectors, proposed in [9]. This resulted in a free and easy encoding of the GAC_{lex} constraint [16]. This encoding can now be freely used to assist in symmetry breaking in problems with 0/1 vectors.

A critical step in solving CSPs is constraint propagation, typically enforcing arc consistency on the current state. Following Kasif, we introduced and studied the *support encoding*, a reformulation of CSP into satisfiability (SAT) [12]. In this reformulation, unit propagation in SAT has the same effect as enforcing arc consistency, in the same worst-case time as arc consistency algorithms. We showed that this reformulation performs better both for complete and local search than the standard encoding into SAT. This result has attracted considerable interest, including follow up papers, from other researchers, e.g. [1]. Overall, we see that the reformulation of propagation algorithms has led to a number of new results, as well as opening up a research area that is attracting interest worldwide.

2.4 Formulation of New Problems

During our work we have provided formulations of new problems. While of course it is common to formulate problems into constraints, our approach has led to interesting new results. First, we have encoded problems not normally viewed as constraint problems, giving the first constraint encodings of these problems. Second, while encoding problems in constraints, we have focused on general aspects of the formulation which therefore teach us more about formulation and reformulation.

An exciting example of our approach was to provide what we believe to be the first complete algorithm for an NP-complete variant of SM, the stable marriage problem with ties and incomplete lists (SMTI). In [14] we presented the first complete algorithm for SMTI (i.e. for the decision problem and the optimisation problem), problem generators, and an empirical study. The stable marriage problem was also encoded into SAT and a study was performed using a state-of-the-art SAT solver [15]. That this was a simple generalisation of our work on SM showed the flexibility of using reformulation to solve such problems, instead of hard-wiring algorithms for each new problem. One outcome of this was an EPSRC research proposal to pursue this work further, but unfortunately it was not funded.

A second example was our recent work providing the first constraint encoding of the problem of construction of supertrees [17]. This problem is of immense significance in biology, as it is the foundation of construction of species trees, indicating the relative dates of divergence of different species. This work is still in its preliminary stages, but holds the promise of allowing much more flexible solving with additional constraints, compared to existing algorithms. A grant proposal in this area is being developed, as is a possible collaboration with a Swedish research group.

The third example is our formulation of a combinatorial optimisation problem, Maximum Density Still Life [6]. While our formulation was not the first constraint encoding of the problem, we were able to solve certain instances for the first time. Furthermore, our approach of seeking general insights paid off. We showed that the dual graph translation could be made more efficient than had previously been realised, by removing many redundant constraints.

3 Project Plan Review

The grant was awarded when the PI was in Strathclyde University. While the proposal was under review Ian Gent left Strathclyde for St. Andrews, where he is now a Reader. In April 2000 Patrick Prosser moved to Glasgow University, joining the Glasgow Algorithms and Complexity Group (with Rob Irving and David Manlove already in place). The grant had to be moved to Glasgow, and recruitment took some time. In April 2001 Barbara Smith moved to Huddersfield to take up a chair. Therefore the start of the project was delayed by more than a year from first being awarded the grant.

The research questions on car sequencing that we outlined in the proposal did not seem to be leading to new insights over and above those obtained from the research on JSP and VRP. The move to Glasgow plus our developing interests led to a body of work on (a) encoding new problems and (b) encoding propagation and levels of consistency. These are both exciting new areas which we admit to pursuing opportunistically, but with a significant body of work developed in the life of the project.

Glasgow presented an opportunity to study a new field, namely the stable marriage problem. Rob Irving [11] and David Manlove are world leaders in matching theory. Therefore, our interest moved from car sequencing to stable marriage. Also, we did not plan to study fine grained reformulation, but this opportunity arose partly out of our study of vehicle routing and also from experience of a new constraint programming language that had just become freely available.

4 Explanation of Expenditure

Expenditure was much as expected. We attended conferences in the USA (IJCAI2001, CP02), CP01 in Cyprus where we organised a workshop on reformulation, ECAI02 in Lyon, ICGT02 in Spain, and CP03 in Ireland. We had two project meetings with ILOG in Paris (this was considerably cheaper than holding the meeting in London), and attended two workshops at 4C in Cork, taking an opportunity to meet with Chris Beck. Due to visa problems Evgeny Selensky was unable to present [5] in Italy, and we had to reclaim the conference fees and travel expenses. Two of our papers, SARA2002 and ICAPS03, were presented by our industrial partner Chris Beck. Together, this resulted in an under-spend in our travel budget. The salary budget was well spent, Evgeny Selensky staying to the end of the project and contributing right to the last day.

5 Research Impact

Our work carried out on this project has had both academic and industrial impact. Due to the working relationship we enjoyed with the ILOG development team, we had had a number of points where technology transfer took place into ILOG. And of course there was transfer into our project, with ILOG guiding some of our research in useful directions. We hope that our research results on reformulation in the large will be useful in the next generation of constraint-based toolkits. Our study on reformulation in the small had an impact on choco; choco was modified as a result of our results in [20]. We are not sure but ILOG Solver might also have been modified.

The studies of stable marriage [13–15] introduced the constraint programming community to a new problem, and an encoding that may be incorporated into future toolkits. This will allow constraint programming to be applied to rich problems where one component involves stable matching. We have also introduced the community to a new concept, i.e. constraint encodings of algorithms, where the application of arc-consistency corresponds to the execution of the original algorithm [13, 16, 12]. The constraint encoding in [16] will also allow constraint programmers to freely and simply encode a level of symmetry breaking in 0/1 models. The empirical study of hard variants of stable marriage resulted directly from our ability to model the problem using constraints. The results of the empirical study will be useful to academics studying such problems. Our results are of theoretical interest too: Green and Cohen studied why our encoding of Stable Marriage led to polynomial solution algorithms in an NP-complete class [10].

As the project progressed we have made our results available from the project web site [19]. As well as all our publications and presentations, Evgeny Selensky has made all our code, problem generators, data sets and results readily available. We hope that our empirical studies can then be reproduced if need be.

6 Conclusions

We believe that this has been a successful project, with valuable results. One is that Dr Evgeny Selensky is now an expert in constraint programming, scheduling, routing, and empirical computer science. Our industrial partners were very valuable, giving us free software, training, access to their development team, and project meetings at their site. We have published many papers on the project in refereed conference proceedings (3 of these with our industrial partners), and have submitted further papers to journals. Several of the papers have already proved influential.

References

1. O. Bailleux and Y. Boufkhad. Efficient CNF encodings of cardinality constraints. In Francesca Rossi, editor, *Proceedings of the Ninth International Conference on Principles and Practice of Constraint Programming — CP 2003*, LNCS 2833, pages 108–122. Springer, 2003.
2. J.C. Beck, **P. Prosser**[1], and **E. Selensky**. Graph transformations for the vehicle routing and job shop scheduling problems. In *1st Int. Conference on Graph Transformation (ICGT 2002), Barcelona*, 2002.
3. J.C. Beck, **P. Prosser**, and **E. Selensky**. On the reformulation of vehicle routing problems and scheduling problems. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation (SARA 2002)*, LNAI 2371, pages 282–289. Springer, 2002.
4. J.C. Beck, **P. Prosser**, and **E. Selensky**. A case study of mutual routing-scheduling reformulations. *Journal of Scheduling*, 2003. under review.
5. J.C. Beck, **P. Prosser**, and **E. Selensky**. Vehicle routing and job shop scheduling: What's the difference? In *ICAPS-03*, 2003.
6. **B.M. Smith**. A dual graph translation of a problem in 'life'. In *Principles and Practice of Constraint Programming - CP 2002*, LNCS 2470, pages 402–414. Springer, 2002.
7. C. Brind, C. Muller, and P. Prosser. Stochastic techniques for resource management. *BT Technology Journal*, 13(1):55–65, 1995.
8. **E. Selensky**. On mutual reformulation of shop scheduling and vehicle routing. In *Proceedings of the 20th UK PLANSIG*, 2001.
9. A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, and T. Walsh. Global constraints for lexicographic orderings. In *Principles and Practice of Constraint Programming - CP 2002*, LNCS 2470. Springer, 2002.
10. M.J. Green and D.A. Cohen. Tractability by approximating constraint languages. In *Proceedings of CP03*, 2003.
11. D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* The MIT Press, 1989.
12. **I.P. Gent**. Arc consistency in SAT. In *Proceedings of ECAI 2002*, pages 121–125. IOS Press, 2002.
13. **I.P. Gent**, R.W. Irvine, D.F. Manlove, **P. Prosser**, and **B.M. Smith**. A constraint programming approach to the stable marriage problem. In *Proceedings CP'01*, pages 225–239, 2001.
14. **I.P. Gent** and **P. Prosser**. An empirical study of the stable marriage problem with ties and incomplete lists. In *Proceedings ECAI'02*, 2002.
15. **I.P. Gent** and **P. Prosser**. SAT encodings of the stable marriage problem with ties and incomplete lists. In *Proceedings of the 5th International Symposium on Theory and Application of Satisfiability Testing (SAT2002)*, 2002.
16. **I.P. Gent**, **P. Prosser**, and **B.M. Smith**. A 0/1 encoding of the GACLex constraint for pairs of vectors. In *Proceedings ECAI'02 workshop W9: Modelling and Solving Problems with Constraints*, 2002.
17. **I.P. Gent**, **P. Prosser**, **B.M. Smith**, and W. Wei. Supertree construction using constraint programming. In Francesca Rossi, editor, *Proceedings of the Ninth International Conference on Principles and Practice of Constraint Programming - CP 2003*, LNCS 2833. Springer, 2003.
18. F. Laburthe. Choco: a constraint programming kernel for solving combinatorial optimization problems. http://www.choco-constraints.net/.
19. **P. Prosser** and **E. Selensky**. PRAS: problem reformulation and search. http://www.dcs.gla.ac.uk/pras/.
20. **P. Prosser** and **E. Selensky**. A study of encodings of constraint satisfaction problems with 0/1 variables. In *Recent Advances in Constraints*, LNCS 2627, pages 121–131. Springer, 2003.

[1] Bold indicates a paper by a member of the project team written during the course of this project.