Solving the VRP with probabilistic algorithms supported by constraint programming

Daniel Riera[38], Angel A. Juan[38]
Studies of Computer Science, Multimedia and Telecommunication
Open University of Catalonia, Barcelona, Spain

Javier Faulin[39]
Department of Statistics and OR, Public University of Navarre, Pamplona, Spain

Abstract

Among the human-being problems, there are those which carry with the exploration of a number of feasible and unfeasible possible solutions. These have been a challenge faced constructing specific strategies. Thus, techniques rising from several areas have been used. They can be complete or incomplete, with mathematical background or a more computational one, single techniques, greedy, constructive, etc. Most of these have proven to work fine when they are tailor-made solutions. Therefore, usually, the solution for a problem is totally useless for a different one or even for the same problem with new/different constraints.

With this position paper, we claim a hybrid methodology combining probabilistic algorithms with a complete technique might be useful in order to separate the optimisation engine and the validation tool. This eases the possibility to have a general optimisation tool, with an acceptable minimal fine tuning, which includes neither the problem model nor validation jobs.

1. Introduction

Along history, people have dealt with problems where decisions to be taken imply certain benefits/costs. Several of these problems, due either size or complexity, are not easily tractable. One of the most studied ones is the Vehicle Routing Problem (VRP) (Golden et al., 2008; Laporte, 2007; Toth and Vigo, 2002).

Optimisation is a field where a wide work has been done in the last few decades. Problems introduced in the previous paragraph have been faced by means of different techniques, all of them with its advantages and weaknesses, assuring or not completeness, generic or tailor-made, based on mathematical foundations, algorithms, graphical models, etc. Most of them have been successful for certain problems, but have fallen short for others.

Some successful current optimisation algorithms are based in the generation of solutions combining statistical distributions and simulation features. In short, a new solution is built by evolving a previous one in the following way: possible improvements are analysed, probabilities are distributed among these alternative improvements, and a simulation is run. The result of the simulations becomes the new solution if is feasible.

While these methods work very well with a small tuning effort, changes in the model – new constraints, change of certain conditions, etc. – may penalise the feasibility checking phase. In order to deal with this, and applying this to the already mentioned VRP, we propose to use Constraint Programming (CP) to build the model. CP has proved to be a powerful and flexible tool for

[38] Email: {drierat,ajuanp}@uoc.edu
[39] [3]Email: javier.faulin@unavarra.es

modelling all kind of constraints. Furthermore, its inclusion frees (or at least lightens) the optimisation part from the model construction/adaptation.

1. Search: Probabilistic algorithms

Although literature provides a very wide range of approaches to combinatorial problems, when facing real scenarios with huge numbers of clients, entities, or whatever to be served/optimised, incomplete algorithms based on evolution of previous complete or incomplete solutions seem to perform better than complete techniques. Thus, there are those which complete a single solution by adding parts of it incrementally.

These often make choices based on certain probabilities, which normally favour greedy (in terms of the cost function) selections. On the other hand, there are methodologies which create a set of solutions and generate new ones by crossing features of these. This evolution is also dependent on certain probabilities.

Although these techniques are not complete, the gap between the solution they (quickly) find and the best known or optimal one is usually very tight. Hence, if the aim is not to get the optimal solution but a good solution in a limited time, these techniques can be extremely useful.

2. Modelling and validation: Constraint programming

Constraints arise in most areas of human endeavour. A constraint is simply a logical relation among several unknowns (or variables), each taking a value in a given domain. The constraint thus restricts the possible values that variables can take. Constraint Programming (CP) is the study of computational systems based on constraints. The main idea is to solve problems by stating constraints (requirements) about the problem area and, consequently, finding a solution satisfying all the constraints.

CP combines ideas from a number of fields including Artificial Intelligence, Combinatorial Algorithms, Computational Logic, Discrete Mathematics, Neural Networks, Operations Research, Programming Languages and Symbolic Computation. The problems solved using CP are called Constraint Satisfaction Problems (CSP) (Tsang, 1993). A CSP is defined as:

- set of *variables*, $X = \{x_1, x_2, ..., x_n\}$,
- for each variable x_i, a finite set D_i of possible values (its *domain*), and
- a set of *constraints* restricting the values that the variables can simultaneously take.

CP is based on exhaustive search supported by consistency algorithms which allow refusing paths in the search tree in advance. This paradigm takes advantage of propagation of decisions in order to avoid exploring unfeasible solutions.

CP can be used as a validation tool since it can be called with a complete solution. The answer will be returned immediately saying if the solution is consistent with the model or not.

3. Methodology - General structure

As said before, the main idea we are working with is the separation of the search engine and the model itself. We must state that the former is not totally independent from the model, since, as seen in the algorithm below, the first and second steps require a constructive heuristic (or perhaps a metaheuristic) containing certain knowledge of the problem.

This is not an issue because the simplest method is useful to generate the solutions.

Algorithm1: General view of the optimisation methodology

1. Generation of an initial solution by means of a well-known constructive greedy technique
2. Evolution of the current solution by means of a probabilistic algorithm
3. Validation of the new solution by means of the CP model
   a. If the solution is verified, it becomes the current solution
   b. Else it is refused
4. If final condition is found, then end, else go to step 2

In a very simple way, algorithm 1 shows how the proposed methodology works. For the first step --- and trying to solve the VRP --- the classical Clarke and Wright (C&W) (Clarke and Wright, 1964) algorithm can be used. This gives an initial solution which feeds the rest of the algorithm.

For the second step we have chosen the main idea from the SR-GCWS-CS algorithm (Juan et al., 2010). In this case, following the basis of the C&W algorithm, probabilistic choices are included in the search in order to evolve the current solution without falling into local minimums.

Finally, the third step of the algorithm, for the VRP, is implemented by using a CP library we have constructed for this family of problems (Riera et al., 2009) in order to quicken the creation of new VRP models.

We are currently making the initial tests of the methodology so we hope to reach results very soon.

4. Conclusions and current/future work

For the solution of combinatorial problems we have proposed to separate as much as possible the modelling part from the search engine. Work done till now shows that tailor-made solutions fall short when there are changes in the problem to solve, and hence, new solutions built from scratch are needed. On the other hand, metaheuristics work fine for any problem, but parameters tuning is not very comprehensive and/or modelling might be quite complicated.

In this paper we claim the separation of modelling and search in order to free as much as possible metaheuristics (or other methods) from the modelling effort. For this we propose CP for the modelling part, because this is a straightforward tool for modelling and validation, and probabilistic algorithms for the search part. They are able to evolve solutions by means of their statistical/probabilistic features in order to avoid local minimums while moving to the optimal.

Thus, we consider the use of CP gives the methodology the following advantages:

- Very powerful modelling tool: accepts a wide range of constraints: linear, non linear, suspensions, etc.
- Models easily modified/incremented
- Fast validation performance

The main drawback we find we could meet is: Since we have split between the search engine and the validation tool, communication penalty between them has been introduced. In the current tests we are working to minimise this time in order to make both tools work as a single one (as far as possible).

Although we are working with the VRP family, the proposal can be applied to any problem. In this case, the main points are: the choice of a simple methodology to generate an initial solution, the modelling of the problem by means of CP and the incorporation of a probabilistic behaviour into a methodology to generate solutions for that problem (it can be the same as the one used for the initial solution). We plan to test the proposed methodology with problems coming from scheduling, planning, timetabling, etc. in order to generalise it as much as possible.

5. Acknowledgements

The research of this paper has been partially sponsored by the Internet Interdisciplinary Institute (IN3) and the Knowledge Community on Hybrid Algorithms for solving Real-life rOuting, Scheduling and Availability problems (HAROSA).

6. References

1. Clarke G, and Wright J (1964). Scheduling of Vehicles from a central Depot to a Number of Delivering Points. Operations Research, 12, 568-581.
2. Golden B, Raghavan S and Wasil EE (eds.) (2008). The Vehicle Routing Problem: Latest Advances and New Challenges. Springer.
3. Juan A, Faulin J, Jorba J, Riera D, Masip D and Barrios B (2010). On the Use of Monte Carlo Simulation, Cache and Splitting Techniques to Improve the Clarke and Wright Savings Heuristics, J Opl Res Soc, doi:10.1057/jors.2010.29.
4. Laporte G (2007). What you should know about the Vehicle Routing Problem. Naval Research Logistics, 54: 811-819.
5. Riera D, Juan A, Guimarans D, Pagans E (2009) A Constraint Programming-Based Library for the Vehicle Routing Problem. In: Proceedings of the 21st European Modeling and Simulation Symposium (EMSS 2009), pp. 261-266. Puerto Cruz, Tenerife, Spain.
6. Toth P and Vigo D (2002). The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM.
7. Tsang, E (1993). Foundations of Constraint Satisfaction. Academic Press.