Reasoning with Tradeoffs in Multi-Objective Optimisation Problems

Nic Wilson,
Cork Constraint Computation Centre
Department of Computer Science
University College Cork

In a multiobjective optimisation problem, we have a (possibly exponentially large) set of possible of decisions, and we want to aid the decision maker (DM) regarding which are the better decisions to choose. Each decision is evaluated on a number of criteria, and can thus be associated with a vector of objective values. For example, with a bi-objective problem the objective vector for a particular decision, X, could be (7, 2), representing seven units for the first objective, and two units for the second. The comparison between decisions is based on a comparison between objective vectors. The optimal decisions are those that are not dominated by any other decision.

A key issue is thus how one orders the objective vectors. The two most common methods are using the Pareto ordering, or using a weighted sum. With a weighted sum, one chooses a different weight for each objective, for example, 0.6 for the first and 0.4 for the second, and hence (7, 2) evaluates to a utility value of 0.6*7 + 0.4*2 = 5. Decision X will then be preferred to Decision Y which has objective vector (3, 3), corresponding to utility value 3. With the weighted sum method, the weights determine the tradeoffs between different objectives, i.e., how much of one objective the decision maker would be happy to trade off against one unit of another objective.

With the Pareto or product ordering, (7, 2) and (3, 3) are incomparable, so neither of Decisions X and Y dominate the other. In general, the Pareto ordering leads to a rather weak pre-order on objective vectors and hence on decisions, especially if there are several objectives. This can lead to a sometimes extremely large number of optimal (undominated) decisions, which is unhelpful for the decision maker.

On the other hand, with the weighted sum method, it can be hard to decide on the appropriate weights. These can be difficult and time-consuming to elicit; this may be, for example, because the decision maker is not clear themselves about the appropriate weighting; or there can be more than one person involved in the decision making process, who might have different ideas about the relative importance of the objectives.

In this talk I consider a formalism that allows reasoning about partial tradeoffs between objectives.
For instance, the decision maker can specify that they regard objective vector (7, 2) to be better than objective vector (3, 3) - implying a constraint on the possible weights vector, but without specifying the weights vector precisely. Linear programming can be used to determine the preference/dominance relation

between objective vectors. I describe how a branch-and-bound algorithm can be used to find optimal solutions with respect to the input preferences, and discuss experimental results considering the effects of varying different parameters in the algorithm. I go on to consider the case where there are random variables (as in a Bayesian network) which together with the decision variables and multi-objective utility functions form a generalised influence diagram. A variable elimination algorithm can be used to find the best decisions, i.e., those with maximal expected multi-objective utility.

The talk is based on work done in collaboration with Radu Marinescu of the IBM Research Lab, Dublin, and Abdul Razak of 4C, University College Cork, especially the following the papers.

Radu Marinescu, Abdul Razak and Nic Wilson
Multi-objective Constraint Optimization with Tradeoffs
To appear in: Proceedings of the 19th International Conference on Principles and Practice of Constraint Programming (CP 2013).

Radu Marinescu, Abdul Razak and Nic Wilson
Multi-objective Influence Diagrams
Proc. 28th Conference on Uncertainty in Artificial Intelligence, (UAI 2012).