A Controller for Online Uncertain Constraint Handling

Alfio Vidotto, Kenneth N. Brown, J. Christopher Beck

Cork Constraint Computation Centre,
Dept. of Computer Science, UCC, Cork, Ireland
av1@student.cs.ucc.ie, k.brown@cs.ucc.ie, c.beck@4c.ucc.ie

An online problem is a problem which grows over time and such that partial solutions are to be generated before the complete problem is known. Moreover, if the problem is an optimization problem, partial solutions must be aimed at optimizing the overall final solution. There may be some uncertain knowledge on how the problems develop. How should we make intermediate decisions? Can we extend existing constraint handling techniques?

In *Online Uncertain Constraint Handling* (OUCH!), we assume that the problem starts with a conditional constraint optimization problem (CCOP). At each time step, an extension to the CCOP may arrive; that is, a set of variables, constraints and utility functions. Each variable will have a decision deadline, and a decision on that variable must be committed to by that deadline. We assume that decisions cannot be revised once they have been committed to, and also that there is no benefit in making an early commitment. The CCOP will allow us to 'reject' variables, will state what that means for each constraint, and will determine the appropriate reward. The objective will be to maximize the total reward over some (possibly infinite) time interval. Specifically, at each time step, we must decide what to do with the variables whose decision deadline has arrived, balancing the immediate reward with the potential for future rewards. If we have a probability distribution for the CCOPs that arrive at each timestep, we can express the future reward in terms of maximum expected utility. The best decision for a set of variables at time step i is:

$$\text{argmax}_{\textit{decision}}\ [\textit{reward for decision} + \textit{max expected future reward}]$$

How should we now reason about those probabilities? [1] attempts to search and propagate constraints over the implied tree of possible futures; [2] samples possible futures, and then selects an action which minimises regret for that sample. In this project, we will investigate instead the use of heuristic estimates of the maximum expected utility, and, similar to [3], we will control the parameters of the heuristic by comparison with the performance relative to the optimum decisions for the observed history. We will use a flexibility measure to estimate the reward obtained from each decision, by examining the domains of the remaining variables, and combine the flexibility with the known reward by a weight parameter ($\alpha/(1-\alpha)$). The decision for a set of variables at time step i will be:

$$\text{argmax}_{\textit{decision}}\ [\textit{reward for decision} + \alpha/(1-\alpha) * \textit{flexibility}]$$

The controller's main task is to tune α to get the best estimate. Our controller (*fig 1*) reacts periodically, i.e. we adjust α depending on the history over the last T time steps. For our initial studies, we will consider optimising the number of variables we accept (and assign consistent values). The extreme cases will be where $\alpha = 0$, where we assign every variable we can without regard to the future, and $\alpha \rightarrow 1$, where we reject every variable, to maintain maximum flexibility. We expect the optimal value of α to be somewhere in between. Our current idea is to maintain an initial α until the loss of reward compared to the maximum we could have achieved exceeds a certain tolerance. Reward may be lost for two reasons: we could have assigned variables but chose to reject them, in which case the flexibility was dominant; or we are forced to reject variables because no consistent value is possible, and thus the immediate reward was dominant. In the first case, we need to decrease α, otherwise, we need to increase it.

The work of the controller can be further extended. If we don’t know anything about the distribution of the future problem extensions, then looking back at the history is the only definite knowledge we have. We can use the controller to try to learn the distribution. For example, we might start with a uniform distribution, and gradually adjust the probabilities by observing the actual sequences.

We are currently applying this general idea to a specific online packing problem [4]. Future work concerns the implementation of the controller, developing general flexibility heuristics, and comparing to existing online constraint solving approaches.

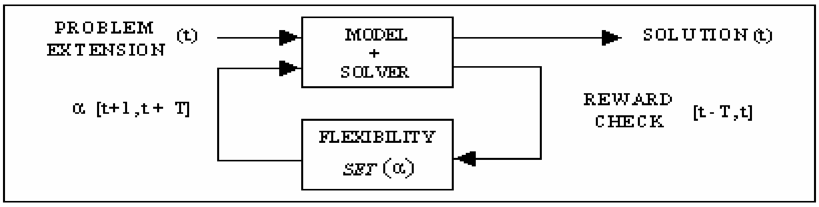

Fig. 1. Flexibility Controller

References

1. D. W. Fowler and K. N. Brown, "Branching constraint satisfaction problems and Markov Decision Problems compared", *Annals of Operations Research*, Volume 118, Issue 1-4, pp85-100, 2003.

2 R. Bent and P. Van Hentenryck. Regrets Only! Online Stochastic Optimization under Time Constraints, Proc. AAAI-04, 2004.

3 L. S. Crawford, M. P. J. Fromherz, C. Guettier, and Y Shang. A Framework for On-line Adaptive Control of Problem Solving. In: *CP'01 Workshop on On-line Combinatorial Problem Solving and Constraint Programming*, Dec. 2001.

4 A. Vidotto, “Online Constraint Solving and Rectangle Packing”, to appear in *Proc CP2004 (Doctoral programme)*, 2004.

Acknowledgements: This work has received support from Enterprise Ireland (SC/2003/81), Science Foundation Ireland (00/PI.1/C075), and ILOG, SA. We thank Christine Wei Wu for many useful discussions on this work.