Randomization for Multi-agent Constraint Optimization

Student: Quang-Huy NGUYEN
Supervisor: Prof. Boi V. FALTINGS

Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
{quanghuy.nguyen,boi.faltings}@epfl.ch

1 Introduction

Many practical applications require solving an optimization problem for multiple agents where each agent has its own preferences in the problem solution. Examples include resource allocation, scheduling, planning, configuration, voting and many others. Two major challenges when solving such problems are the complexity of finding optimal solution and the incentive compatibility for participating agents.

First, these problems are often NP-hard or NP-complete and thus cannot be solved exactly for large problem sizes in real-world settings. *Local search* algorithms are usually used instead of complete search algorithms in such situations: starting from an initial assignment, the local search procedures make a sequence of local changes, called moves, to improve the current solution. Many randomization techniques have been studied to keep such procedures from getting stuck in local minima including simulated annealing ([12]) and a variety of stochastic local search algorithms ([16, 18, 11]). In these techniques, the information from current solution is not much taken into account to guide the search except Walksat algorithms its variants ([16, 13]: it performs random walks on a variable randomly chosen from unsatisfied clauses, thus guides the search more directed towards improving some aspect of the solution.

Second, agents in real-world settings usually have conflicting preferences and they are *self-interested*, i.e., they want to maximize their own benefit in participating in the algorithm. The agents are also *bounded rational*, i.e, they have limited knowledge and limited computation capacity. The strategic behavior should be taken into account in designing algorithms for such agents as it might severally damage the efficiency of the system and prevent it to accomplish the purposes that it was designed for.

This paper discusses a general randomization technique, called *random subset optimization*, for escaping from local minima in local search algorithms. In each local choice step, the local search procedure will randomly choose a part of the optimization function, and optimize for this part only. It turns out that this results in a more focussed optimization strategy and shown to be especially effective in some hard optimization problems.

We show that the uncertainty of our randomization algorithms can make the agents' manipulation hard, thus prevent bounded rational agents from manipulating such algorithms. We propose a new hard-to-manipulate local search algorithm using our randomization scheme ([6]).

2 Random subset optimization

The performance of local search algorithms strongly depends on the choice of the local move in a local choice step. The simplest way to implement this function is hill-climbing: choose the neighbour that most improves the objective function. However, this strategy often gets stuck in local optima, where the objective function can only be improved by making changes beyond the neighbours of the current state. In this case, random moves can help the search to get out of the minimum.

The first process as such was simulated-annealing ([12]), where with some probability *LocalChoice* accepts a move that degrades the objective function. These moves let the search escape from local optima and give it the possibility to restart hill-climbing from a hopefully better starting point. This strategy of *random walks* has become a standard strategy for SAT algorithms, in particular Walksat and its variants ([16, 13]). Other algorithms, such as stochastic local search ([4]), randomly allow moves to neighbouring states that only give some, but not the maximal improvement in the objective function.

Purely random moves such as in the simulated-annealing algorithm have the disadvantage that they are not guided towards actually breaking out of a local optimum; this explains the slow convergence of the algorithm. An exception are the algorithms for SAT problems, where only moves that satisfy at least one currently unsatisfied clause are considered. This means that in fact the move is optimal with respect to a part of the objective function. The underlying realization is that if the current state is not the globally optimal assignment, then some component of the objective function must have a better value elsewhere. Escaping from the local optimum requires refocussing the search on that part of the objective function.

We can generalize this idea to general constraint optimization by having each local choice step randomly choose a subset of the objective function with a probability p and only optimize this part. In this way, we avoid moves that only degrade the solution and rapidly lead back to the original minimum, and focus the search on moves that actually have a chance to break out of the local optimum. We call this *random subset* optimization (RSO).

The behavior of the algorithm is controlled by two parameters:

- d is the number of constraints being left out of the optimization in the current iteration.
- p is the probability of optimizing over a subset only.

In our experiments, it appears that the best performance is reached when d is small and p is close to 1.

In another variant, called *adaptive* random subset optimization (ARSO), we let the parameter d start with a high value and decrease over time, similarly to the cooling schedule used in simulated annealing. At each value of d, the algorithm performs a linear number of local choice steps w.r.t the number of variables. In our implementation, d is reduced by 1 after each iteration. When d decreases below 0, the algorithm terminates.

Completeness

It is clear that our method is not always *probably approximately complete* (PAC) (introduced in [10]), as there are problems where the optimal assignment $\mathbf{v}^*$ is not reachable by the algorithm at all. This is the case when the optimal assignment is a compromise among all relations, i.e. there is no subset of the relations for which $\mathbf{v}^*$ is also optimal.

However, our method can be shown to be PAC by a constructive proof for *decomposable* problems such as max-SAT and graph coloring problems.

3 Achieving bounded-rational incentive compatibility

The local search procedure can only work correctly if agents accurately report their relations to the algorithm. Using side payments, we can create an *incentive-compatible* protocol where agents are motivated to truthfully report these relations. Well-known results in game theory ([8]) have shown that all protocols that are incentive-compatible, individually rational and select the optimal solution must be a kind of VCG mechanism ([17, 2, 9]). Furthermore, Nisan and Ronen [14] have shown that a VCG mechanism requires a provably optimal solution. Thus, there is no mechanism that makes local search incentive-compatible while maintaining individual rationality.

We thus introduce a weaker notion of incentive compatibility, called *bounded-rational incentive-compatibility* that uses computational complexity to rule out manipulation. A mechanism is bounded rational incentive compatible if by varying a parameter, the probability that a bounded rational agent can manipulate the protocol can be made arbitrarily close to 0.

We adopt a similar randomization scheme as in the section the section 2 where we randomly select a set of relations to be left out of the optimization at a local choice step. It turns out that a good way to select these relations is to take all the relations belonging to a randomly selected agent. We shown in [6] that with high probability, the number of states that a manipulator need to be considered for a successful manipulation of the algorithm will grow exponentially with the size of the problem. Thus, for a sufficiently large problem size it will exceed the computational capacity of a bounded rational agent.

This scheme of randomization also allows us to simultaneously guarantee budget-balance of the VCG tax mechanism by simply paying the payment surplus/deficit to the agent that was excluded from the optimization step. This mechanism is similar to the proposal in [3], who proposed giving the surplus to agents that have no interest in the variable being considered. We call such agents *uninterested* agents. The mechanism proposed here applies even when no uninterested agent exists. When there are uninterested agents, optimization can be improved by selecting these to be chosen as excluded agents. More details on the mechanism can be found in [5].

The algorithm is individually rational in expectation as shown in [6] because no agent is systematically disadvantaged by the randomization.

4 Experimental results

We have evaluated the performance of our algorithms both on decomposable and non-decomposable problems. As decomposable problems, we chose network resource allocation (as in [6]). As a non-decomposable problem, we chose the graph coloring with minimized colors on random graphs and some benchmarks from the DIMACS challenge ([1]).

We compare our algorithms (RSO and ARSO) with several known local search techniques: HC-hillclimbing without randomization, RHC-hillclimbing with random restarts (similar to GSAT [15]); SA-simulated annealing ([12]; SLS-stochastic local search [7, 18]. The averaged results are taken over 100 runs.

Figure 1 shows the results for random network resource allocation problems. It can be seen that our results are very promising: RSO, ARSO and SA all eventually reach the same solution quality and slightly outperform the other algorithms. However, RSO converges much faster than all other algorithms, and seems to be a clear winner for this application.

Figure 2 and 3 shows the results for min-coloring problems on random graph with 50 vertices. We start with 10 colors ($x_c = 10$) and set the initial assignment such that all the vertices have different colors. Interestingly, our algorithms, especially ARSO, outperform all other algorithms not only in speed of convergence, but also in the final solution quality achieved. Still, our results are obtained with no tuning of parameters. This is important to us, given that we cannot guarantee any completeness result on such a non-decomposable problem.

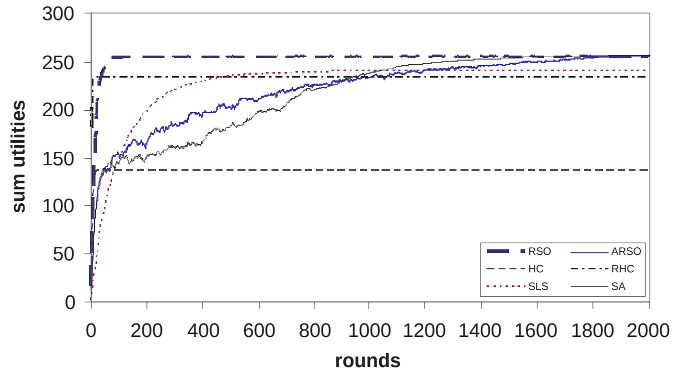

Fig. 1. *Average utility gain of different local search algorithms as a function of the number of steps.*

5 Conclusion

Local search algorithms are often the only possibility to solve large optimization problems. We have introduced a general randomization technique for improving local search algorithms. We presented a set of preliminary experiments on small problems that show that random subset optimization seems to be an excellent local search algorithm. We are currently conducting experiments on benchmark problems from the DIMACS benchmark instances to prove its performance on large and difficult optimization problems.

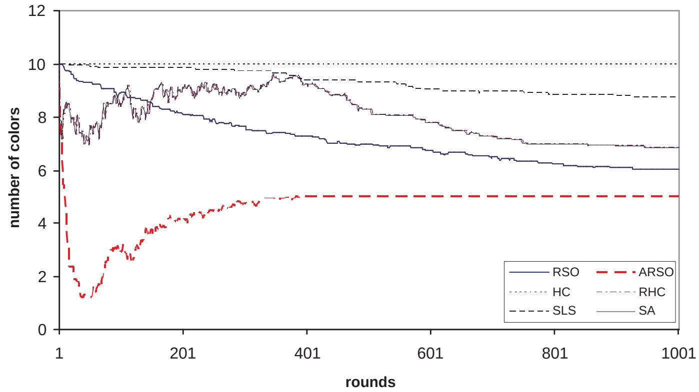

Fig. 2. *Average number of colors used of different local search algorithms as a function of the number of steps.*

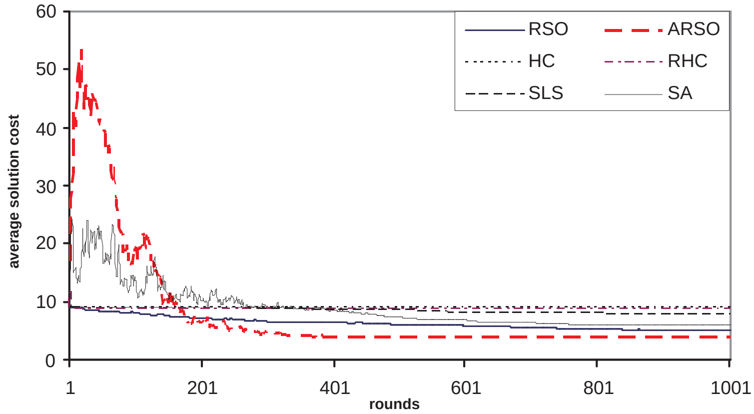

Fig. 3. *Average utility gain of different local search algorithms as a function of the number of steps.*

Our randomization method can be also made to be bounded-rational incentive compatible for bounded-rational agents in multi-agent constraint optimization problems by using randomization to make the manipulation hard.

The success of our approach is based on exploiting the information in the current sub-optimal solution to guide the search more focussed in the right directions. The price to pay is that such a technique is not guaranteed to be probably approximately complete on all problems. It would be interesting to explore further and in more details the method on hard optimization problems.

References

1. The Second DIMACS Challenge, http://mat.gsia.cmu.edu/challenge.html, 1992.
2. E.H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
3. E. Ephrati and J. S. Rosenschein. The clarke tax as a consensus mechanism among automated agents. In *Proceedings of AAAI-91*, pages 173–178, 1991.
4. M. Fabiunke. Parallel distributed constraint satisfaction. In *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 99)*, pages 1585–1591, Las Vegas, 1999.
5. B. Faltings. A budget-balanced, incentive-compatible scheme for social choice. In *Agent-mediated E-commerce (AMEC) VI*, 2004.
6. B. Faltings and Q.-H. Nguyen. Multi-agent coordination using local search. In *Proceedings of IJCAI 05*, 2005.
7. Stephen Fitzpatrick and Lambert Meertens. An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. In Kathleen Steinhoefel, editor, *First Symposium on Stochastic Algorithms: Foundations and Applications*, pages 49–64. Springer-Verlag, 2001.
8. Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):427–38, March 1977.
9. Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–31, July 1973.
10. Holger H. Hoos. On the run-time behaviour of stochastic local search algorithms for sat. In *Proceedings of the AAAI '99/IAAI '99*, pages 661–666. AAAI, 1999.
11. Holger H. Hoos and Thomas Sttzle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco (CA), USA, 2004.
12. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
13. D. McAllester, B. Selman, and H. Kautz. Evidence for invariants in local search. In *In Proceedings of AAAI-97*, pages 321–326, 1997.
14. Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 242–252. ACM Press, 2000.
15. B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the AAAI-92*, pages 440–446, 1992.
16. B. Selman, H. Levesque, and D. Mitchell. Noise strategies for improving local search. In *Proceedings of the AAAI-94*, pages 337–343, 1994.
17. W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, (2):8–37, 1961.
18. Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenburg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 1-2(161):55–88, 2005.