Uncovering Hidden Structure through Parallel Problem Decomposition[*]

Yexiang Xue and **Stefano Ermon** and **Carla P. Gomes** and **Bart Selman**
Computer Science Department, Cornell University, Ithaca, NY, 14850
{yexiang, ermonste, gomes, selman}@cs.cornell.edu

Abstract

A key strategy for speeding up computation is to run in parallel on multiple cores. However, on hard combinatorial problems, exploiting parallelism has been surprisingly challenging. It appears that traditional divide-and-conquer strategies do not work well, due to the intricate non-local nature of the interactions between the problem variables.

In this paper, we introduce a novel way in which parallelism can be used to exploit hidden structure of hard combinatorial problems. We demonstrate the success of this approach on minimal set basis problem, which has a wide range of applications in machine learning and system security, etc. We also show the effectiveness on a related application problem from materials discovery.

In our approach, a large number of smaller sub-problems are identified and solved concurrently. We then aggregate the information from those solutions, and use this to initialize the search of a global, complete solver. We show that this strategy leads to a significant speed-up over a sequential approach. The strategy also greatly outperforms state-of-the-art incomplete solvers in terms of solution quality. Our work opens up a novel angle for using parallelism to solve hard combinatorial problems.

Introduction

Exploiting parallelism and multi-core architectures is a natural way to speed up computations in many domains. Recently, there has been great success in parallel computation in fields such as scientific computing and information retrieval (Dean and Ghemawat 2008). Over the past decade, we have also witnessed tremendous improvements in combinatorial search, especially in fields such as Satisfiability testing (SAT) and Mixed Integer Programming (MIP) (Le Berre and Simon 2005) . These dramatic improvements are largely due to a set of sophisticated heuristics that have been developed, including complex branching rules, fast propagation, clause learning, and rapid restarts. These techniques allow modern solvers to uncover and exploit the inner structure of combinatorial problems, and lead to dramatic speedups in many domains (Williams, Gomes, and Selman 2003). Exploiting parallelism to boost combinatorial search, however, remains a largely open research problem. A natural approach is to use the divide-and-conquer approach, where the search space is divided into sub-spaces, and each sub-space is allocated to a parallel node (Chu, Stuckey, and Harwood 2008). While this strategy looks very natural, it has had limited success in the context of combinatorial search, mainly because of the complex, non-local interactions between variables and constraints. For example, in SAT solving it remains an open problem to define an efficient mechanism for effectively sharing and communicating clauses learned by different processes (Katsirelos et al. 2013), which is one of the key factors in modern day solvers' efficiency. This issue is so problematic that to date, the most successful parallel combinatorial solvers avoid any sort of communication, and are based on the portfolio idea. That is, they run a portfolio of solvers (of different type or with different randomization) in parallel, so that they can terminate as soon as one of the algorithms completes. (Xu et al. 2008; Malitsky et al. 2011).

In this paper, we revisit the problem of exploiting parallelism to boost combinatorial search, taking a novel angle which utilizes parallelism to uncover hidden structure of hard combinatorial problems. We focus on a specific NP-complete problem called the set basis problem, which is an important problem with many applications, ranging from roles-based access control systems (Vaidya, Atluri, and Warner 2006), secure broadcasting (Shu, Lee, and Yannakakis 2006), text and user preference mining (Miettinen et al. 2008) to computational biology (Nau et al. 1978).

We introduce a novel parallel scheme, in which parallelism is used as a prepossessing step to identify a promising portion of search space to be explored by a complete sequential solver. Our approach leverages a nice dual property of the set basis problem. In our new scheme, the original problem is first decomposed into a series of easier sub-problems by relaxing some of the constraints. These sub-problems are then solved concurrently using a set of parallel processes. Next, the solutions to these sub-problems are aggregated to obtain a good initial guess for the solution of the original problem. A global sequential solver then searches for a solution in an iterative deepening manner, starting from the promising portion of the search space identified in the previous phase.

We empirically show that a global solver, when initial-

[*]This work was supported by National Science Foundation Grant No. 0832782, 1344201, 1258330 and 1059284.

Instance		Solution Quality								Run-time for	
		HPe		*Fast-Miner*		*ASSO*		*Complete*		Complete Method	
No.	k	k'	$E\%$	k'	$E\%$	k'	$E\%$	k'	$E\%$	*Parallel*	*Sequential*
A1	8	65	0	8	100	8	26.56	8	0	**37.82**	12029.14
A2	8	86	0	8	100	8	18.18	8	0	**191.5**	3878.4
A3	10	41	0	10	96.67	10	25	10	0	**14.46**	37857.41
A4	10	47	0	10	100	10	18.92	10	0	**2633.06**	15438.26
A5	10	58	0	10	100	10	52	10	0	**199.82**	21678.42
A6	12	51	0	12	96.3	12	25	12	0	**395.49**	65792.00
A7	12	63	0	12	100	12	38.46	12	0	**6389.8**	93111.72
A8	12	93	0	12	100	12	51.06	12	0	**3942.67**	$>$ 48 hours

Table 1: Comparison of different methods on classic set basis problems. k is the optimal number of bases in these instances. In the solution quality block, we show the number of bases k' returned and the error rate $E\%$ for incomplete method *HPe*, *FastMiner* and *ASSO* and the complete method. $k' > k$ means more bases are used than optimal. $E\% > 0$ means the coverage is not perfect. In the run-time block, *Parallel* and *Sequential* show the times (in seconds) to solve the instance using the complete method, with and without the parallel scheme, respectively.

ized with proper information obtained by solving the subproblems, takes much less wall-clock time (typically, by several orders of magnitude) to find the exact solution. For example in table 1, it takes about 400 seconds to solve A6 with the parallel scheme, but over 18 hours sequentially. We also show our strategy greatly outperforms state-of-the-art incomplete solvers in terms of solution quality. We compare our solver with *HPe* from (Ene et al. 2008), *FastMiner* from (Vaidya, Atluri, and Warner 2006) and *ASSO* from (Miettinen et al. 2008). As seen from table 1, *HPe* often requires far more bases than optimal, and the bases found by *FastMiner* and *ASSO* cannot cover the set exactly.

While the set basis problem has many natural applications, our research is motivated by a relatively new application in the field of combinatorial materials discovery (Le Bras et al. 2011). In this domain, the set basis problem is used to find a succinct explanation of a large set of measurements (X-ray diffraction patterns) that are represented in a discrete way as sets. Mathematically, this corresponds to a generalized version of the set basis problem extended with extra constraints. Our parallel solver can be applied to this generalized version of the set-basis problem as well, and we demonstrate significant speed-ups on a set of challenging benchmarks (see table 2).

We believe that our work opens up a novel angle for using parallelism to solve hard combinatorial problems.

System	P	L	*Parallel*	*Sequential*
A1	45	5	**121.26**	902.99
A2	45	5	**158.03**	588.85
A3	45	5	**76.65**	537.55
B1	60	5	**122.07**	972.8
B2	60	5	**180.95**	591.66
B3	60	5	**125.75**	1060.79
B4	60	5	**136.47**	633.52
C1	45	10	**3295.88**	17441.39
C2	45	8	**1190.95**	3948.41
D1	28	7	**209.27**	622.16
D2	28	8	**282.29**	2182.23
D3	28	10	**905.27**	2357.87

Table 2: The time for solving phase identification problems. P is the number of sample points in the system. L is the average number of peaks per basis in the problem. *Parallel* and *Sequential* show the time (in seconds) to solve the problem with and without the parallel scheme, respectively.

References

Chu, G.; Stuckey, P. J.; and Harwood, A. 2008. Pminisat: A parallelization of minisat 2.0. Technical report, SAT race.

Dean, J., and Ghemawat, S. 2008. Mapreduce: simplified data processing on large clusters. *CACM* 51(1):107–113.

Ene, A.; Horne, W. G.; Milosavljevic, N.; Rao, P.; Schreiber, R.; and Tarjan, R. E. 2008. Fast exact and heuristic methods for role minimization problems. In *SACMAT*, 1–10. ACM.

Katsirelos, G.; Sabharwal, A.; Samulowitz, H.; and Simon, L. 2013. Resolution and parallelizability: Barriers to the efficient parallelization of sat solvers. In *AAAI-13*.

Le Berre, D., and Simon, L. 2005. Fifty-five solvers in vancouver: The sat 2004 competition. In *Theory and Applications of Satisfiability Testing*, 321–344. Springer.

Le Bras, R.; Damoulas, T.; Gregoire, J. M.; Sabharwal, A.; Gomes, C. P.; and van Dover, R. B. 2011. Constraint reasoning and kernel clustering for pattern decomposition with scaling. In *CP'11*.

Malitsky, Y.; Sabharwal, A.; Samulowitz, H.; and Sellmann, M. 2011. Non-model-based algorithm portfolios for sat. In *SAT'11*, 369–370. Berlin, Heidelberg: Springer-Verlag.

Miettinen, P.; Mielikainen, T.; Gionis, A.; Das, G.; and Mannila, H. 2008. The discrete basis problem. *TKDE* 20(10):1348–1362.

Nau, D. S.; Markowsky, G.; Woodbury, M. A.; and Amos, D. B. 1978. A mathematical analysis of human leukocyte antigen serology. *Mathematical Biosciences* 40(34):243 – 270.

Shu, G.; Lee, D.; and Yannakakis, M. 2006. A note on broadcast encryption key management with applications to large scale emergency alert systems. In *IPDPS 2006*, 8 pp.–.

Vaidya, J.; Atluri, V.; and Warner, J. 2006. Roleminer: Mining roles using subset enumeration. In *CCS '06*, 144–153. ACM.

Williams, R.; Gomes, C.; and Selman, B. 2003. Backdoors to typical case complexity. In *IJCAI'03*, volume 18, 1173–1178.

Xu, L.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2008. Satzilla: Portfolio-based algorithm selection for sat. *J. Artif. Intell. Res. (JAIR)* 32:565–606.