Heavy-Tailed Behavior and Randomization in Proof Planning

Andreas Meier[1] **Carla Gomes**[2] **Erica Melis**[1]
[1] Fachbereich Informatik, Universität des Saarlandes
66041 Saarbrücken, Germany
{ameier|melis}@ags.uni-sb.de
[2] Computer Science Department, Cornell University
Ithaca, NY 14853, USA
gomes@cs.cornell.edu

Proof Planning Proof planning considers mathematical theorems as planning problems. A proof planning problem is defined by an *initial state* specified by the proof assumptions, the *goal state* given by the theorem to be proved, and a set of planning operators called *methods*. Finding a proof corresponds therefore to searching for a sequence of planning operators that derive the theorem from the assumptions. In the proof planning system ΩMEGA (Benzmueller *et al*. 1997) the traditional proof planning approach is enriched by incorporating mathematical knowledge into the planning process (see (Melis & Siekmann 1999) for details). In particular, methods represent mathematically meaningful inference steps and can be specific for a mathematical domain.

We explore the domain of the residue classes over the integers ((Meier, Pollet, & Sorge 2000)) using a proof planning approach. We apply ΩMEGA to solve large testbeds of algebraic problems of a residue class set RS_n (e.g. $\mathbb{Z}_3, \mathbb{Z}_4, \mathbb{Z}_5 \backslash \{\bar{1}_5\}$) over the integers together with a binary operation $\circ$ (e.g. $\bar{*}, \bar{+}$) such as RS_n is closed with respect to $\circ$, RS_n is associative with respect to $\circ$ etc. The results of these proofs are in turn used to classify a given structure $(RS_n, \circ)$ in terms of the algebraic structure it forms, i.e., whether it is a semi-group, monoid etc. Moreover, another classification process divides given residue class structures into equivalence classes of isomorphic structures. During this classification process we have to prove proof obligations stating that two structures are isomorphic or not. Our experiments show that the hardest problem instances correspond to problems stating that two structures are not isomorphic (*non-isomorphism problems*). For some instances the planner generates long proofs, with long run times, while for other (similar) instances the planner generates short proofs, with short run times. Since we are not able to find a heuristic rule that enables us to control the unpredictability of the planner's performance, we apply randomization and restart techniques to boost the search process and increase the solvability horizon for such non-isomorphism problems.

Heavy-Tailed Problems and Randomization Combinatorial search methods often exhibit a remarkable large variance in performance on problems of the same complexity class. That is, a given search method might solve one problem instance quickly, whereas, on another (similar) problem instance, it may take a long time to solve it. This unpredictability in the run time of the search methods can often be explained by so-called *heavy-tailed* cost distributions (Gomes *et al*. 1998). Heavy-tailed distributions are characterized by a non-negligible probability of runs that take significantly longer than average. A technique to eliminate heavy-tailed behavior and the unpredictability in the run time of the underlying search method is to add randomization to the search procedure combined with a restart strategy to take advantage of short runs.

Recent work demonstrates that several hard combinatorial search methods show heavy-tailed behavior and that randomization and restart techniques can help boost the search (i.e., to decrease the mean solution costs) as well as solve formerly unsolved problem classes. In particular, the technique proved successful for hard scheduling and planning problems in constraint satisfaction and propositional satisfiability formulations (see (Gomes *et al*. 1998) and (Gomes, Selman, & Kautz 1998)).

Experiments with ΩMEGA We formulated a suitable proving technique for non-isomorphism problems using planning methods encoding suitable steps combined with so-called *control rules* that capture domain knowledge. We applied ΩMEGA with this proving technique on a testbed of 160 non-isomorphism problems of identical complexity. We were able to solve about 67.5% of the problems with a time bound of 7200 seconds. Moreover, we found some of the proofs within a few seconds whereas other proofs took much longer. Similarly, we found that some proofs were considerably longer than others, or needed considerably more backtrack steps.

Afterwards we added a stochastic element to this proving technique by randomizing the selection of choices ranked equally good by our search heuristic. We repeatedly applied the randomized version of our proving technique to one problem of our testbed. The randomized version of our proving technique was successful in about 71% of the proving attempts (again with a time bound of 7200). The plot of the run time distribution of the successful runs is given in Fig. 1: the figure shows a curve which is characteristic of heavy-tailed distributions, with a long tail stretching over

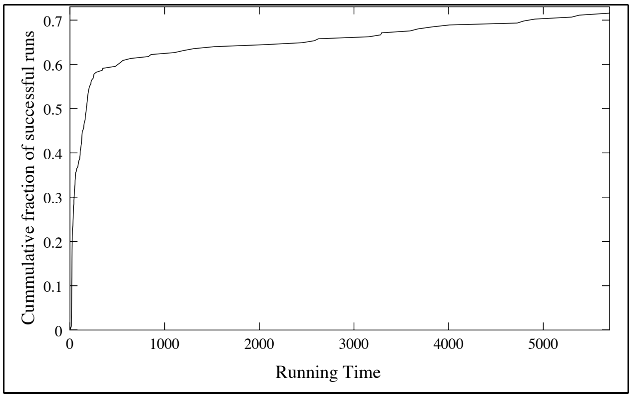

Figure 1: Running Time Cost Distribution

several orders of magnitude.

Based on our our (deterministic) experiments on the full testbed and on the randomized experiments on one problem instance, we found a time bound of 100 seconds to be a suitable cutoff value for a restart strategy of the randomized version of our proving technique. We applied the resulting restart strategy to the full testbed of 160 problems and we were able to solve all but 4 problems (this corresponds to a success ratio of about 97.5%, as opposed to the initial 67.5%) with a mean time considerably smaller than the one produced by the deterministic planner, and also with proofs much shorter in length.

Conclusion Our experiments show the effectiveness of our randomized proof planning approach by showing a significant improvement in performance over a testbed of 160 non-isomorphism problems. In particular, with our approach, a much larger fraction of problem instances is solvable (from 67.5% to 97.5%; 2hr time limit per proof) and a variety of proofs is generated for each problem instance.

Our work demonstrates that randomization and restarts can be successfully applied to deduction systems. We believe these results are very promising and might apply to other problem domains using deduction techniques such as *e.g.*, software verification.

References

Benzmueller, C.; Cheikhrouhou, L.; Fehrer, D.; Fiedler, A.; Huang, X.; Kerber, M.; Kohlhase, M.; Konrad, K.; Meier, A.; Melis, E.; Schaarschmidt, W.; Siekmann, J.; and Sorge, V. 1997. ΩMEGA: Towards a mathematical assistant. In McCune, W., ed., *Proceedings 14th International Conference on Automated Deduction (CADE-14)*, 252–255. Townsville: Springer.

Gomes, C. P.; Selman, B.; McAloon, K.; and Tretkoff, C. 1998. Randomization in backtrack search: Exploiting heavy-tailed profiles for solving hard scheduling problems. In Simmons, R.; Veloso, M.; and Smith, S., eds., *Proceedings of the Fourth International Conference on AI Planning Systems (AIPS'98)*, ???–???

Gomes, C. P.; Selman, B.; and Kautz, H. 1998. Boosting combinatorial search through randomization. In Simmons, R.; Veloso, M.; and Smith, S., eds., *Proceedings of the 15th National Conference on Artifivial Intelligence (AAAI-98)*, 431–437. AAAI Press.

Meier, A.; Pollet, M.; and Sorge, V. 2000. Exploring the domain of residue classes. SEKI-Report SR-00-04 (SFB), Universität des Saarlandes, Saarbruecken, Germany. also submited to the Journal of Symbolic Computation.

Melis, E., and Siekmann, J. 1999. Knowledge-based proof planning. *Artificial Intelligence*.