Partitioning Strategies for Distributed SMT Solving

Author: Clark Barrett

Group Members:
Lailumah Hiba Nadeem (20K-0431)
Fabiha Atique (20K-0369)
Naima Jamal (20K-0404)

Executive Summary

- Explores partitioning strategies for improving performance in parallel Satisfiability Modulo Theories (SMT) solving.
- New partitioning strategies are introduced and evaluated.
- Hybrid portfolios incorporating these strategies outperform traditional portfolios in parallel SMT solving.

Background

- SMT solvers are widely used in verification, model checking, security, synthesis, and optimization.
- Solver performance is a bottleneck for many users.
- Portfolio solving and divide-and-conquer solving are two approaches for improving solver performance.
- Portfolio solving involves running multiple solvers or configurations in parallel.
- Divide-and-conquer solving partitions complex problems into independent subproblems.
- Finding an effective partitioning algorithm for SMT has been challenging.
- The document introduces new partitioning strategies inspired by the cube-and-conquer approach.
- Strategies involve combining sources for collecting variables and creating different partition types.
- Evaluation using CVC5 and diverse benchmarks shows improved performance with partitioning portfolios.
- Hybrid portfolios combining partitioning and traditional strategies yield even better results.
- The strategies offer significant speed-ups in parallel SMT solving.

Methodology - Experimental Setup

- Authors establish a well-defined experimental setup for reliable and reproducible results.
- They choose the CVC5 solver as the base solver for implementing partitioning strategies.
- The experimental environment includes parallel and distributed computing resources.
- Specifications cover hardware, software configurations, core count, memory allocation, network infrastructure, and solver parameters.

Methodology - Selection of Benchmarks

- Authors choose a diverse set of benchmarks covering various domains.
- Benchmarks include challenging SMT problems to comprehensively evaluate partitioning strategies.
- Selection aims for a representative sample of real-world scenarios across different problem types.
- Both synthetic and real-world benchmarks are considered for a comprehensive evaluation.

Methodology - Implementation of Partitioning Strategies

- Authors implement partitioning strategies within the CVC5 solver.
- This involves developing algorithms and heuristics for collecting atoms, creating partition types, and integrating strategies into the solver's framework.
- Implementation ensures seamless application during the solving process, facilitating efficient parallelization and distribution of subproblems.

Methodology - Evaluation Metrics

- Solving time - measures the time taken by the solver to solve a given benchmark problem.
- Dividing the problem into manageable subproblems.
- Other metrics may include the improvement in solving time compared to non-partitioning portfolios and the scalability of the strategies with increased parallelism.

Methodology - Experimental Evaluation

- The implemented strategies are implemented on the selected benchmarks using the established experimental setup.
- The solving time and other metrics are recorded for each strategy and benchmark combination.
- The experiments are repeated multiple times to account for variations and ensure statistical significance.

Methodology - Comparison and Analysis

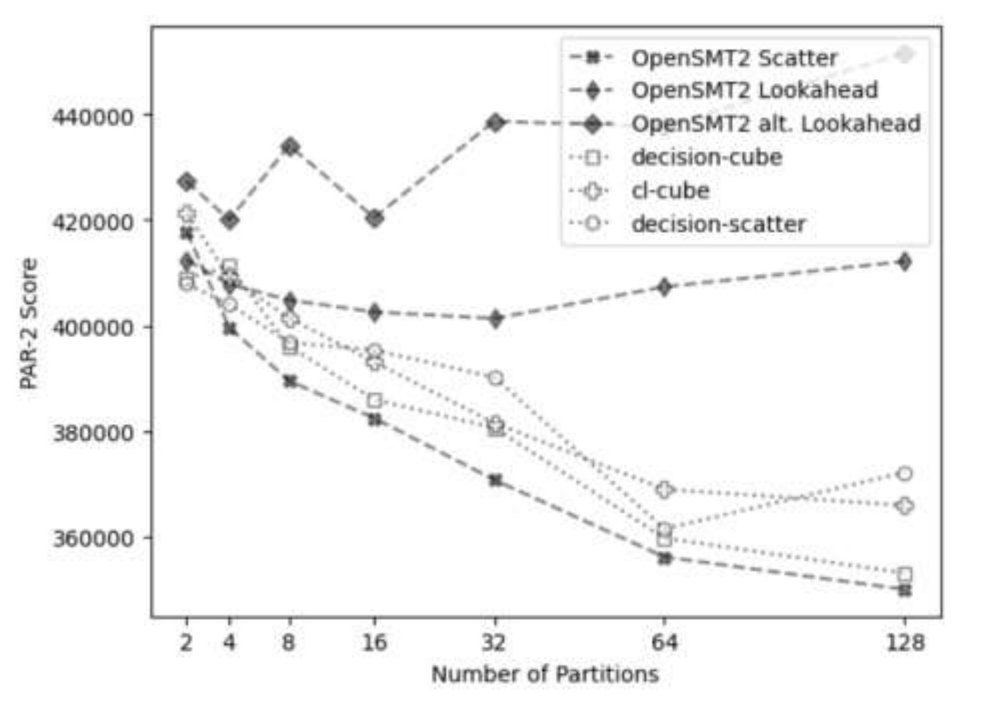
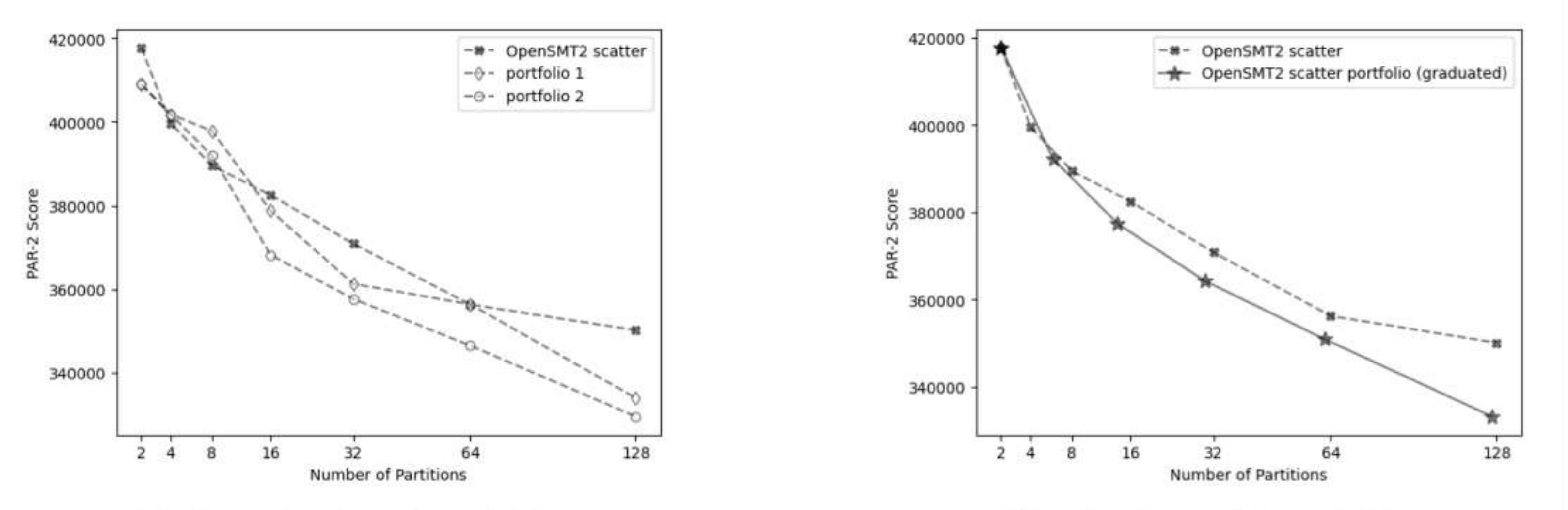


Fig. 1. CVC5 partitioning strategies vs OpenSMT2 partitioning strategies

Methodology - Comparison and Analysis



(a) Partitioning Portfolios

(b) Graduated Portfolio

Fig. 2. Comparison of various partitioning portfolios

Key Discussion Points

- The paper explores ways to make SMT solvers faster using parallel and cloud computing.
- Portfolio solving, running multiple solvers in parallel, can speed up solving problems in SMT.
- Divide-and-conquer approaches, breaking down problems into smaller parts, haven't worked well for SMT solving.
- The paper introduces new strategies to divide problems into smaller parts more effectively.
- The new strategies were implemented in CVC5 solver and tested on various problems.
- The strategies showed good scalability, solving problems faster with more parallelism.
- Contributions of the paper include new partitioning strategies and improved performance in parallel SMT solving.
- The paper explains basic concepts of SMT solving and the framework used by SMT solvers.
- In conclusion, the paper summarizes the main findings and their significance.

Limitations

1. Limited evaluation scope: The paper mentions the evaluation of the proposed partitioning strategies on a large set of benchmarks, but the specific details of the benchmarks and their representativeness are not provided. The limitations of the evaluation in terms of the diversity or complexity of the benchmarks could be a potential limitation.
2. Lack of comparison with other approaches: While the paper discusses the performance of the proposed partitioning strategies, it does not explicitly compare them with other existing approaches for parallel SMT solving. The absence of such comparisons limits the understanding of how the proposed strategies perform in relation to alternative methods.
3. Scalability considerations: The paper briefly mentions that the proposed approaches scale well with more parallelism, but it does not provide in-depth analysis or scalability experiments. The limitations of the proposed strategies in terms of scalability, resource utilization, or performance degradation with a larger number of partitions could be relevant but not addressed.
4. Practical implementation considerations: The paper focuses on the theoretical aspects of the partitioning strategies and their impact on SMT solving performance. However, practical considerations such as implementation complexity, overhead costs, or compatibility with existing SMT solvers are not discussed. These practical limitations may affect the feasibility or adoption of the proposed strategies in real-world scenarios.