Scaling Exact Multi-Objective Combinatorial Optimization by Parallelization

Jianmei Guo, Edward Zulkoski, Rafael Olaechea, Derek Rayside, Krzysztof Czarnecki, Sven Apel, Joanne M. Atlee
Multi-Objective Combinatorial Optimization (MOCO) Problems
Multi-Objective Combinatorial Optimization (MOCO) Problems

• In Software Engineering
  – Architecture design
  – Test data generation
  – Project planning

• In other domains
  – Hybrid vehicle powertrain design
  – Electric vehicle battery design
  – Civil infrastructure repair planning
1. Abstract functions as features

A Running Example

Cross-tree constraints: Video requires Camera

A Running Example

1. Abstract functions as features
2. Infer quality attributes

Cross-tree constraints: Video requires Camera

[Guo et al., Variability-Aware Performance Prediction: A Statistical Learning Approach, ASE 2013]
A Running Example

1. Abstract functions as features
2. Infer quality attributes
3. Find the Pareto front

Cross-tree constraints: Video requires Camera
Objectives: minimizing cost, minimizing latency
Search Space and Pareto Front
Challenges and Trade-offs

• Most MOCO problems are NP-hard.

• Approximate methods
  + Mostly efficient
  - No guarantee for accuracy
  - Parameter sensitivity

• Exact methods
  + Never miss any optimal opportunity
  - Mostly time-consuming
Workflow of Exact Methods using Solvers

- Features +
- Constraints +
- Quality attributes +
- Objectives
Workflow of Exact Methods using Solvers

Features + Constraints + Quality attributes + Objectives

Formal specifications
Workflow of Exact Methods using Solvers

Features + Constraints + Quality attributes + Objectives → Formal specifications → Constraint solvers + Space-exploration algorithms
Workflow of Exact Methods using Solvers

Features + Constraints + Quality attributes + Objectives

Formal specifications

Constraint solvers + Space-exploration algorithms

All, exact Pareto-optimal solutions
Sequential Space Exploration

• Guided Improvement Algorithm (GIA)

Sequential Space Exploration

• Guided Improvement Algorithm (GIA)

Sequential Space Exploration

• Guided Improvement Algorithm (GIA)

Our Research Question

Can parallelization improve the efficiency of exact solving MOCO problems?

To what extent can it improve?
Parallel Space Exploration

- Partition GIA (ParGIA)
  - Collaborative communication
- Objective Split GIA (OS-GIA)
  - Geometric decomposition
- OS-ParGIA
  - A hybrid of OS-GIA and ParGIA
- Feature Split GIA (FS-GIA)
  - Problem division
- FS-ParGIA
  - A hybrid of FS-GIA and ParGIA
Partition GIA (ParGIA): collaborative communication
Objective Split GIA (OS-GIA): geometric decomposition
Objective Split GIA (OS-GIA):
geometric decomposition
OS-ParGIA: geometric decomposition & collaborative communication
Feature Split GIA (FS-GIA): problem division
Feature Split GIA (FS-GIA): problem division

- Recursive division
- Load balance
- #SAT solver
Evaluation

FS-GIA is identified as the fastest and the most scalable algorithm.
Evaluation

FS-GIA gains **super-linear** speedups that **scale** well up to **64 cores**.

A case study
- 44 features
- 2,000,000 variants
- 4 objectives

Time consumption
- 229 minutes using 1 core
  -> 3 minutes using 64 cores!
Threats to Validity

• Generality to other MOCO problems, especially in industry

<table>
<thead>
<tr>
<th></th>
<th>#Features</th>
<th>#Solutions</th>
<th>#Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS</td>
<td>35</td>
<td>5,184</td>
<td>7</td>
</tr>
<tr>
<td>Web Portal</td>
<td>44</td>
<td>2,120,800</td>
<td>4</td>
</tr>
<tr>
<td>E-SHOP</td>
<td>290</td>
<td>5.02E+49</td>
<td>4</td>
</tr>
</tbody>
</table>

[Esfahani et al., GuideArch: Guiding the Exploration of Architectural Solution Space under Uncertainty. ICSE 2013.]

[Sayyad et al., On the Value of User Preferences in Search-Based Software Engineering: A Case Study in Software Product Lines. ICSE 2013.]
Related Work

• According to Talbi et al.’s recent survey, “Parallelization of exact optimization methods” ... “is rarely tackled in the multiobjective context.”

• K-PPM
  – Geometric decomposition into cubes
  – Not scalable

Conclusion

• Five novel parallel MOCO algorithms
  – search for exact optimal solutions using off-the-shelf SAT/SMT/CSP solvers
  – parallelize the search via collaborative communication, divide-and-conquer, or both.

• FS-GIA outperforms all other proposed algorithms
  – Super-linear speedup that scales well up to 64 cores

• A new direction in scaling exact MOCO methods
Future Work

• Industry applications
  – Automotive wire harness optimization

• Hybrid optimization
  – Combine exact and approximate methods

• Theoretical guarantee
  – Performance bounds of exact MOCO algorithms
Thank you!

http://gsd.uwaterloo.ca/epoal