Using Quality of Service Bounds for Effective Multi-objective Software Architecture Optimization

Qais Noorshams, Anne Martens, Ralf Reussner
QUASOSS 2010
Motivation

<table>
<thead>
<tr>
<th>Quality</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>3 s</td>
</tr>
<tr>
<td>$P(\text{failure on demand})$</td>
<td>0.001</td>
</tr>
<tr>
<td>Costs</td>
<td>2000 €</td>
</tr>
</tbody>
</table>
Motivation

<table>
<thead>
<tr>
<th>Quality</th>
<th>Prediction</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>7 s</td>
<td>&lt; 5 s</td>
</tr>
<tr>
<td>P(failure on demand)</td>
<td>0.001</td>
<td>&lt; 0.002</td>
</tr>
<tr>
<td>Costs</td>
<td>1000 €</td>
<td>&lt; 1500 €</td>
</tr>
</tbody>
</table>
Motivation

<table>
<thead>
<tr>
<th>Quality</th>
<th>Prediction</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>4 s</td>
<td>&lt; 5 s</td>
</tr>
<tr>
<td>P(failure on demand)</td>
<td>0.0015</td>
<td>&lt; 0.002</td>
</tr>
<tr>
<td>Costs</td>
<td>1250 €</td>
<td>&lt; 1500 €</td>
</tr>
</tbody>
</table>

Related Work
Motivation

<table>
<thead>
<tr>
<th>Quality</th>
<th>Prediction</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>4 s</td>
<td>&lt; 5 s</td>
</tr>
<tr>
<td>P(failure on demand)</td>
<td>0.0015</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Costs</td>
<td>1250 €</td>
<td>&lt; 1500 €</td>
</tr>
</tbody>
</table>
**Problem**

**Challenges**
- Large search space
- Multiple, conflicting objectives
- Exact required quality may be unclear

**Problems**
- Exploration of search space with multi-objective metaheuristics is time consuming
- Too many uninteresting solutions found

---

Motivation ➤ Problem ➤ Related Work ➤ Approach & Realization ➤ Case Study ➤ Conclusion
Related Work

Canfora2005

- Optimize costs while satisfying constraints

McGregor2007, Menascé2010, Xu2010

- Aim at satisfying constraints

Aleti2009, Martens2010

- Multi-objective optimization using evolutionary algorithms

Stop if requirements fulfilled
No trade-off

Quality requirements not included in the optimization

No other quality criteria
Solution

Combine Multi-objective Architecture Optimization and Quality of Service Bounds
Evolutionary Algorithm

- Approximate
- Non-deterministic
- Efficient for large search spaces

Initial candidate

Reproduction: Generate new candidates

Set of candidates

Evaluation of each candidate

Set of candidates with QoS metrics

Selection: Choose candidates for next generation

Set of Pareto-optimal candidates

Stop?

Motivation  ➔  Problem  ➔  Related Work  ➔  Approach & Realization  ➔  Case Study  ➔  Conclusion

Qais Noorshams - Using Quality of Service Bounds for Effective Multi-objective Software Architecture Optimization
Constrained Evolutionary Algorithm

Response Time

Response Time

Costs

Costs

Architectural Candidate

Solutions

Motivation

Problem

Related Work

Approach & Realization

Case Study

Conclusion
Case Study – Setup

Claim: Our approach finds solutions faster than unconstrained evolutionary algorithms

Minimize:
- Response time
- \( P(\text{failure on demand}) \)
- Costs

Possible Changes:
- Allocate to up to 9 servers
- Different CPU types per server
Case Study – Result

- 800 candidates
- ~10h (@2 x 2GHz, 2GB RAM)

Result – Time savings of bounded search:
- Max. 89.2%
- Avg. 38.8%

Motivation → Problem → Related Work → Approach & Realization → Case Study → Conclusion
Conclusion

Approach

• Combination of metaheuristic multi-objective software architecture optimization and quality of service bounds

Benefits

• Increasing effectiveness in software architecture optimization
• Reducing time to find valuable solutions

Evaluation

• Implementation of a tool for focused software architecture optimization
• Case Study: Focused optimization on average ~40% faster

Future Work

• Compare constraint-handling techniques
• Further analysis and improvements of metaheuristic multi-objective software architecture optimization
Literature


Case Study – Result

coverage

max
mean
min

iteration

17.10.2010
Qais Noorshams - Using Quality of Service Bounds for Effective Multi-objective Software Architecture Optimization
Software Design and Quality Group
Institute for Program Structures and Data Organization
Appendix – QML Declarations

QML Declarations

QMLDeclaration

name : EString

QMLContractType

GenericQMLContract

GenericQMLProfile
Appendix – QML Contract Type

- `enumeration` `EnumNumericDomain`
  - real
  - integer

- `enumeration` `EnumRelationSemantics`
  - decreasing
  - increasing

- `eClass` `QMLContractType`
  - `+ dimensions`
  - `+ unit`

- `eClass` `Dimension`
  - `name : EString`

- `eClass` `Unit`
  - `name : EString`
  - `0..1`

- `eClass` `DimensionType`
  - `+ relationSemantics`
  - `+ type`

- `eClass` `RelationSemantics`
  - `relSem : EnumRelationSemantics`

- `eClass` `DimensionTypeNumeric`
  - `domain : EnumNumericDomain`
  - `0..1`

- `eClass` `DimensionTypeEnum`

- `eClass` `DimensionTypeSet`
  - `+ order`

- `eClass` `NumericRange`
  - `lowerLimit : EDouble`
  - `upperLimit : EDouble`

- `eClass` `Element`
  - `name : EString`
  - `1..*`
  - `+ biggerElement`
  - `+ smallerElement`

- `eClass` `Order`
  - `*`

- `eClass` `Order`
  - `*`
Appendix – QML Contract

Qais Noorshams - Using Quality of Service Bounds for Effective Multi-objective Software Architecture Optimization
Appendix – QML Profile