To Preserve or Not to Preserve Invalid Solutions in Search-Based Software Engineering: A Case Study in Cloud Cost Optimization

Jianmei Guo

Alibaba Group

2018.11.17 @CSBSE

[Guo and Shi, ICSE'18]

Search-based software engineering (SBSE)

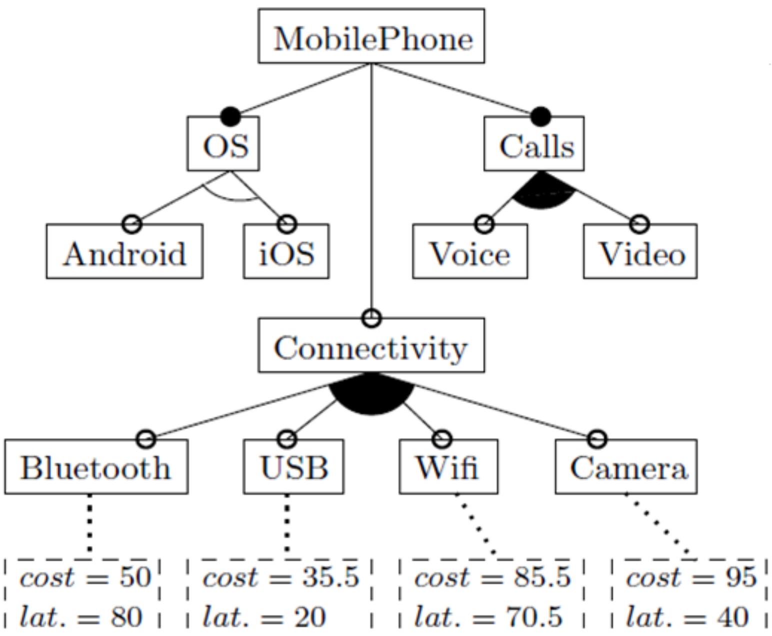
Search-based software engineering (SBSE)

Should we preserve invalid solutions during the search?

Outline

- A case study in software produce lines (SPLs)
  - SPL optimization problem
  - Five algorithm variants based on the state-of-the-art
  - Seven real-world subjects (*a particular one in cloud cost optimization*)
- Three potential limitations of preserving invalid solutions
- Empirical guidance on how to preserve valid solutions
  - How to choose the population for performance evaluation?
  - Should we keep valid solutions in the initial population?
  - Should we fix invalid solutions after mutation?
  - Should we preserve valid solutions all along the way?
  - Should we preserve invalid solutions during the search?

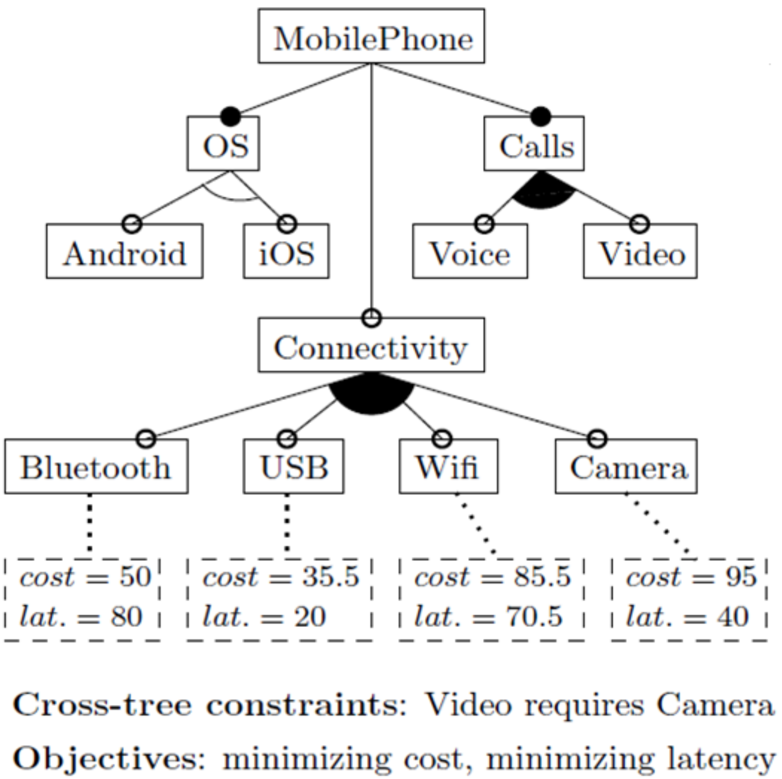
SPL Optimization Problem

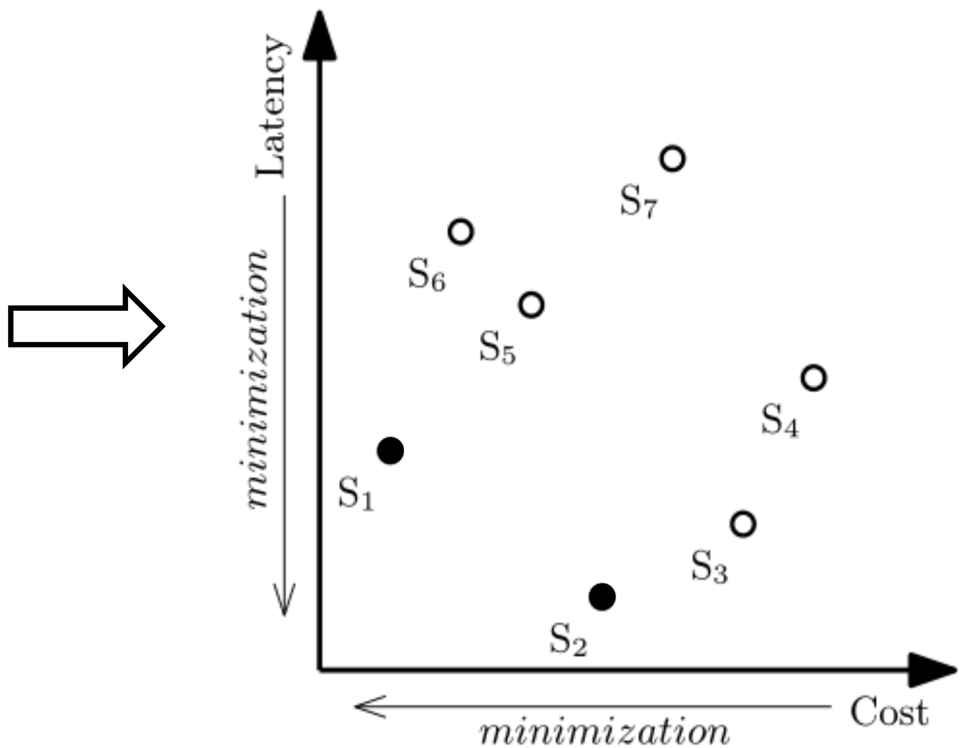


Cross-tree constraints: Video requires Camera

Objectives: minimizing cost, minimizing latency

SPL Optimization Problem





A multi-objective combinatorial optimization problem

Five Algorithm Variants

- Multi-objective evolutionary algorithms (MOEAs)
  - IBEA [Sayyad et al., ICSE'13, ASE'13]
  - SATIBEA [Henard et al., ICSE'15]

Five Algorithm Variants

- Multi-objective evolutionary algorithms (MOEAs)
  - IBEA [Sayyad et al., ICSE'13, ASE'13]
  - SATIBEA [Henard et al., ICSE'15]

Algorithm	λ	δ	TIP
SATIBEA	0.001	0.02	random
SATIBEAv1	0.001	0.02	valid
SATIBEAv2	0.001	0.02	random + rich seed
SATIBEAv3	1	1	random
SATIBEAv4	0.001	1	random
SATIBEAv5	1	1	valid

Mutation rate

Probability of using SAT solving for mutation

Type of initial population

Five Algorithm Variants

- Multi-objective evolutionary algorithms (MOEAs)

Algorithm	λ	δ	TIP
SATIBEA	0.001	0.02	random
SATIBEAv1	0.001	0.02	valid
SATIBEAv2	0.001	0.02	random + rich seed
SATIBEAv3	1	1	random
SATIBEAv4	0.001	1	random
SATIBEAv5	1	1	valid

Preserving validity **"all along the way"** by incorporating a subroutine that resolves **non-Boolean** constraints over integer or real-number variables together with arithmetic or relational operators

Seven Subjects

- Five largest SPLs hitherto reported in the literature
- Two SPLs with realistic values and constraints of attributes

SPL	Version	#Features	#CTC	#Attr	Type of Attr Values	#Attr Constraints
ECos	3.0	1,244	3,146	3	synthetic	N/A
FreeBSD	8.0.0	1,396	62,183	3	synthetic	N/A
Fiasco	2011081207	1,638	5,228	3	synthetic	N/A
uClinux	20100825	1,850	2,468	3	synthetic	N/A
Linux	2.6.28.6	6,888	343,944	3	synthetic	N/A
Drupal	7.23	47	23	22	realistic	N/A
AmazonEC2	12 June 2014	81	3,859	17	realistic	17,049

[Henard et al., ICSE'15]
[Sánchez et al., Software & Systems Modeling, 2017]
[García-Galán et al., Future Generation Comp. Syst., 2016]

Potential Limitations – 1/3

- Ratios of invalid solutions in the final populations produced by state-of-the-art approaches

SPL	IBEA	SATIBEA
eCos	0%	4.94%
FreeBSD	2%	89.02%
Fiasco	0%	63.86%
uClinux	69%	0.02%
Linux	100%	81.83%

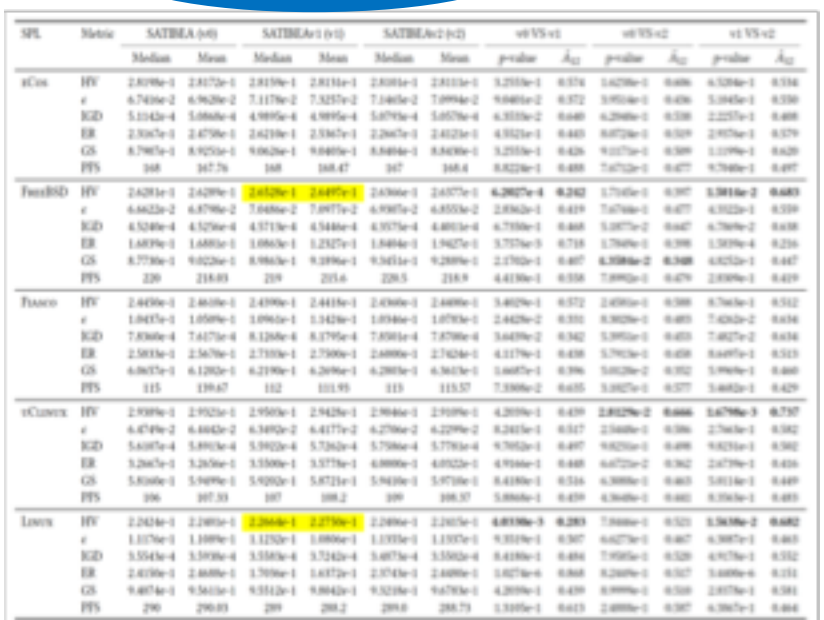
IBEA [Sayyad et al., ICSE’13, ASE’13]
SATIBEA [Henard et al., ICSE’15]

Potential Limitations – 2/3

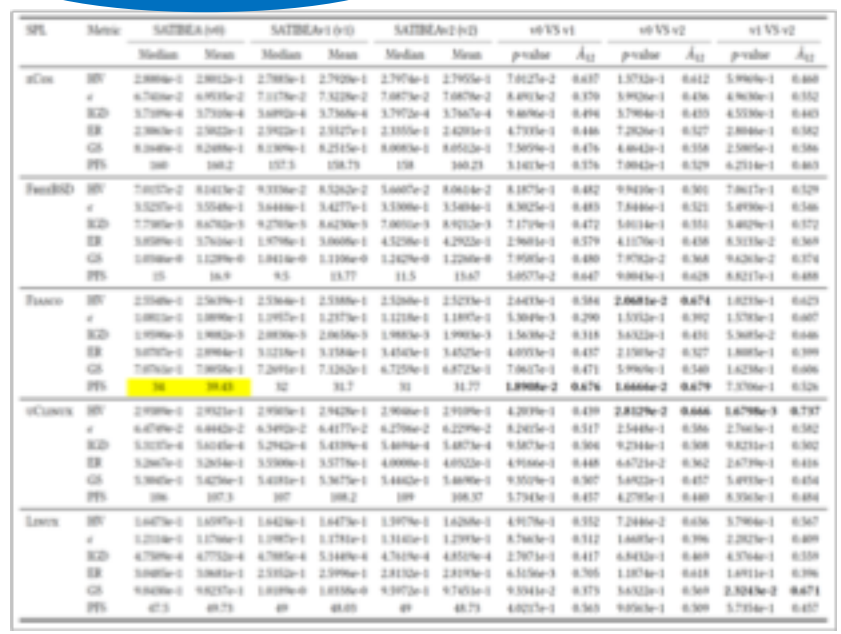
Performance evaluation depends mainly on the metrics calculated on the **population** produced

- Including invalid solutions
- Excluding invalid solutions

SATIBEAv1

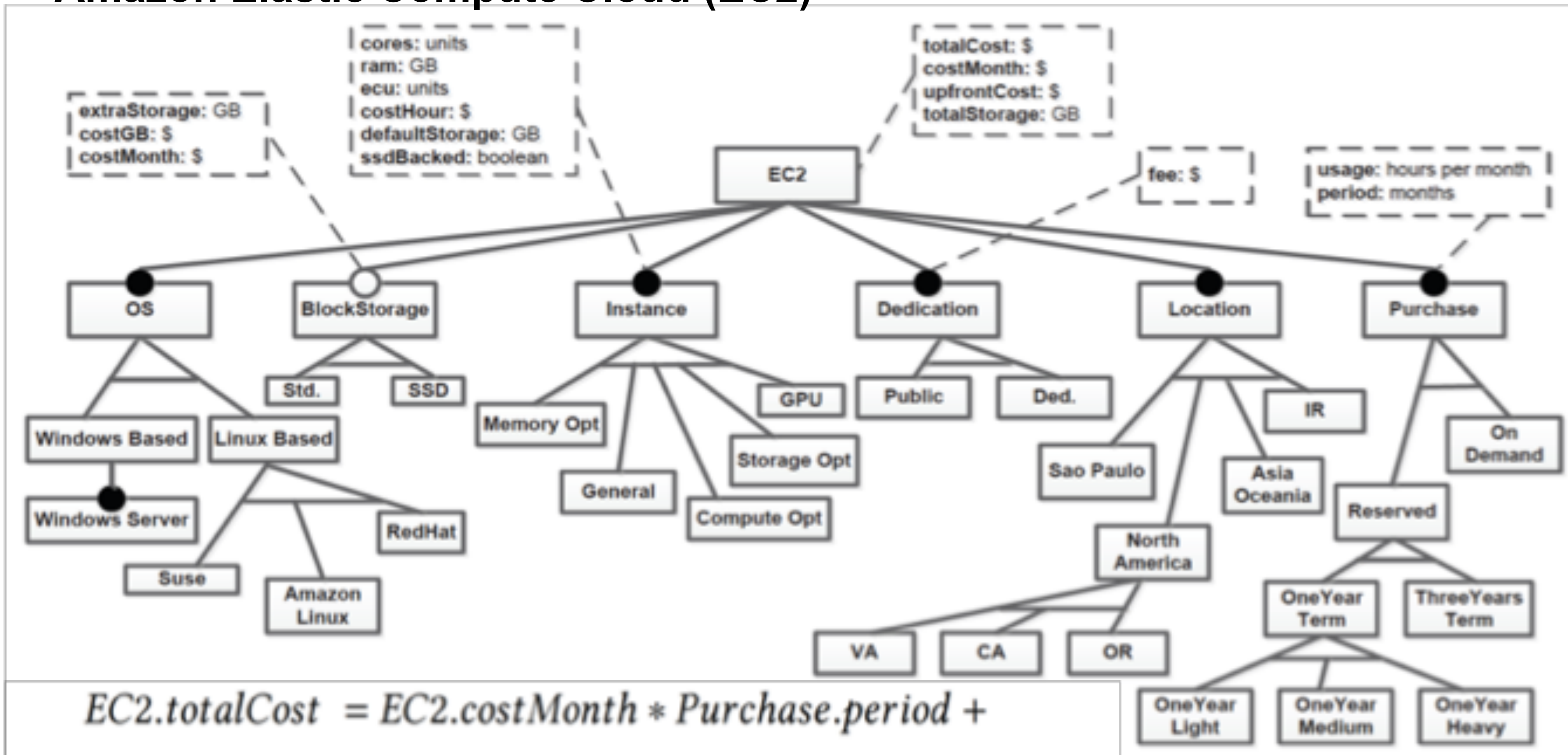


SATIBEA



Potential Limitations – 3/3

Amazon Elastic Compute Cloud (EC2)



$$
\begin{aligned}
EC2.totalCost &= EC2.costMonth * Purchase.period + \\
&\quad EC2.upfrontCost \\
EC2.costMonth &= (Inst.costHour + Inst.dedicatedFee) * \\
&\quad Purchase.signedUsage + BlockS.costMonth \\
BlockS.costMonth &= BlockS.extraSpace * BlockS.costGB
\end{aligned}
$$

Excerpted from
[García-Galán et al., 2016]

Empirical Guidance – 1/5

How to choose the population for performance evaluation?

- Inconsistent results when including and excluding invalid solutions in the population
- Invalid solutions are unbuildable and thus useless in practice

Considering only valid solutions, ***at least***, in the final population

Empirical Guidance – 2/5

Should we keep valid solutions in the initial population?

- Empirical results cannot statistically distinguish three algorithms adopting different initial populations in terms of any quality metric
- SATIBEA is the least and the easiest to build

Using random strategy to generate the initial population

Empirical Guidance – 3/5

Should we fix invalid solutions after mutation?

- Experimental results on six subjects demonstrate the superiority of SATIBEA among three algorithms adopting different mutation operators.

Adopting a lower mutation rate (0.001) and a lower probability (0.02) to fix a solution to be valid after mutation

Empirical Guidance – 4/5

Should we preserve valid solutions all along the way?

- Valid solutions must be preserved all along the way for AmazonEC2
- For the other subjects without attribute constraints, SATIBEA works better for 4 out of 6 subjects.

Preserving valid solutions all along the way might not always be necessary.

Empirical Guidance – 5/5

Should we preserve invalid solutions during the search?

Whether or not to preserve invalid solutions deserves **more attention** in SBSE,
and in some cases, we have to preserve valid solutions all along the way during the search.

Conclusion

- **Open question**: *Should we preserve invalid solutions during the search?*

- Three **potential limitations** of preserving invalid solutions
  - No guarantee to produce valid solutions
  - Evaluation results might be misleading
  - Complex attribute constraints break the fitness calculation based on simple aggregation

- **Empirical guidance** on how to preserve valid solutions
  - Considering only valid solutions, at least, in the final population for evaluation
  - Using random strategy to generate the initial population
  - Adopting a lower mutation rate and a lower probability to fix a solution to be valid
  - Preserving valid solutions all along the way might not always be necessary

- **Take-home message**: *Whether or not to preserve invalid solutions deserves more attention in SBSE, and in some cases, we have to preserve valid solutions all along the way during the search.*

- See paper for:
  - Detailed experimental setup and results
  - Threats to validity & Future work

Thanks for your attention!

https://github.com/jmguo/balanceValidity/