SIP: Optimal Product Selection from Feature Models using Many-Objective Evolutionary Optimisation

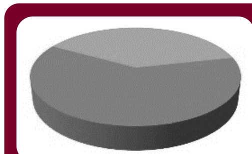
Robert M. Hierons[(1)], Miqing Li[(1)], XiaoHui Liu[(1)], Sergio Segura[(2)], Wei Zheng[(3)]

[(1)] Brunel University London, UK
[(2)] University of Seville, Spain
[(3)] Northwestern Polytechnical University, China


Introduction


Problem


Approach


Results


Conclusions

Introduction

Software product lines

- Family of Products
- Built from a set of reusable assets
- There can be constraints
- Evidence of significant use in industry

Introduction

Feature Models

- Set of features

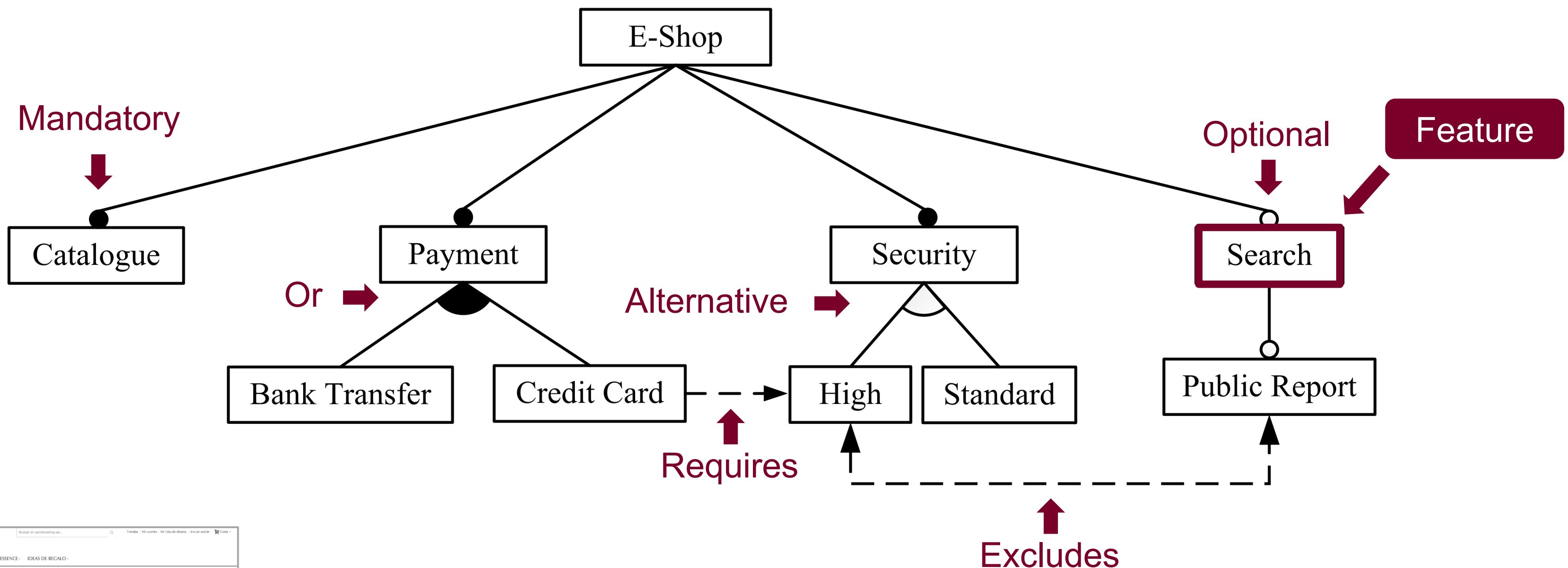
- There can be constraints such as:
    - Or/Alternative
    - Requires/Excludes
    - Mandatory

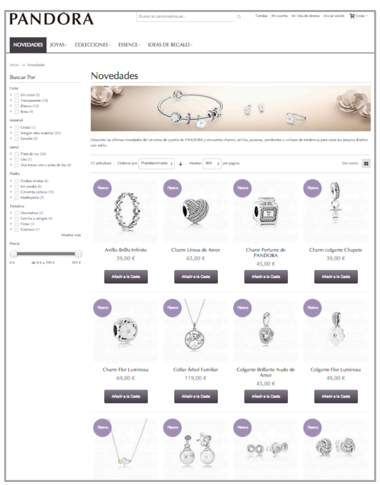
- A valid product is:
    - A set of features that satisfies the constraints

- An FM is typically represented as a tree

Introduction

Feature Models

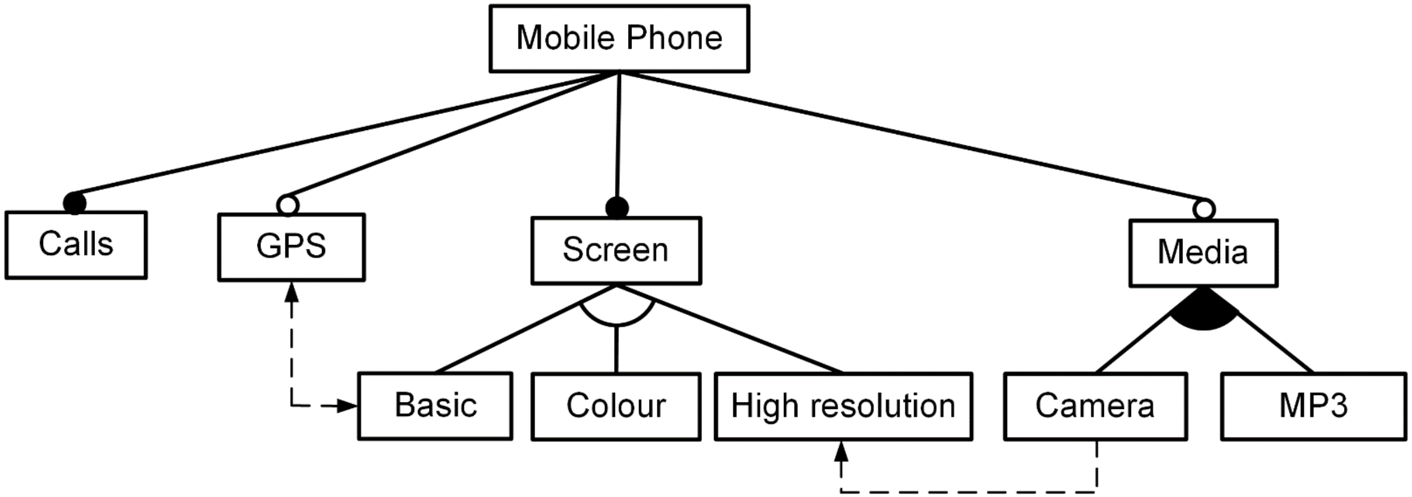


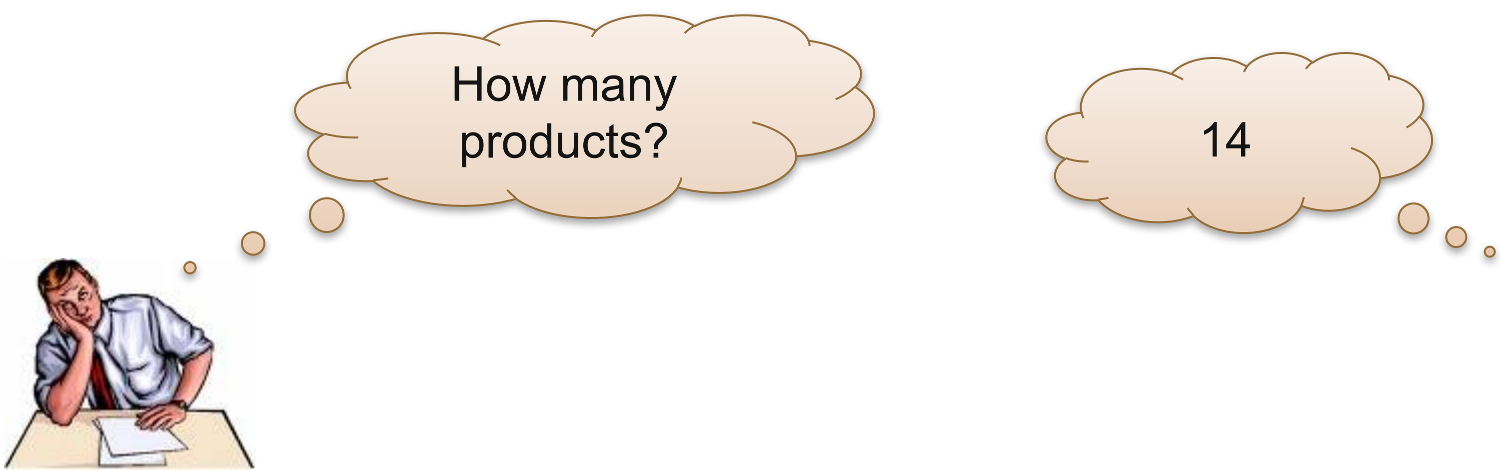


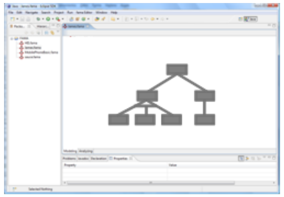
= {E-shop + Catalogue + Payment + Credit Card + Security + High + Search}

Introduction

Automated analysis of feature models:
Computer-aided extraction of information from FMs

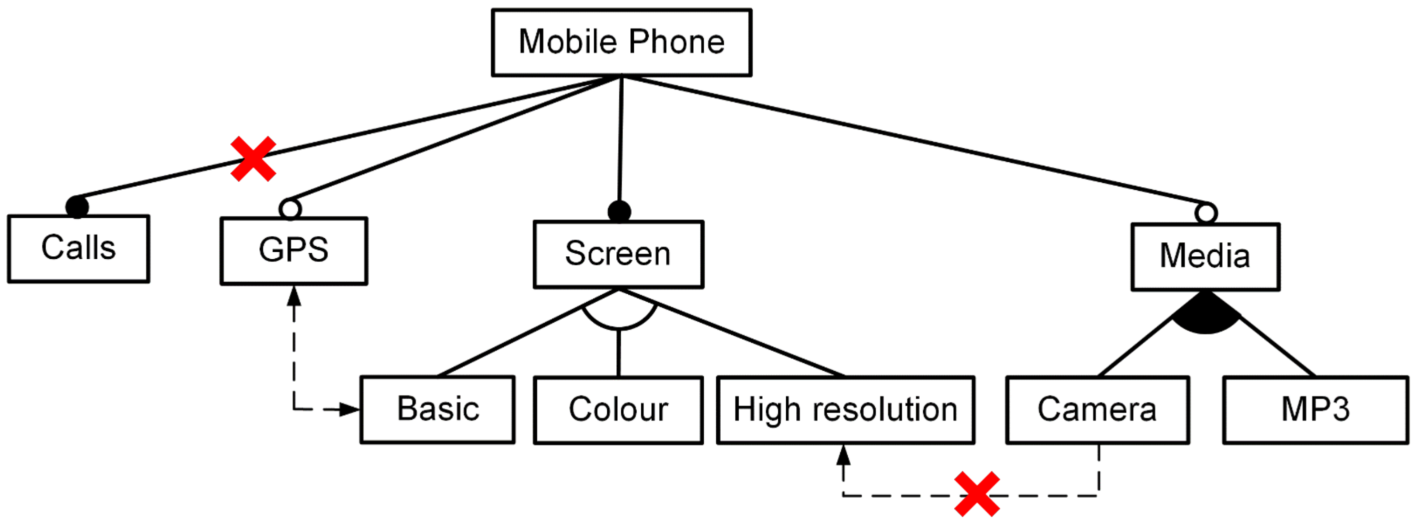




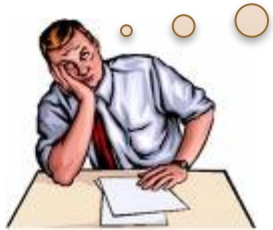


Introduction

Automated analysis of feature models:
Computer-aided extraction of information from FMs

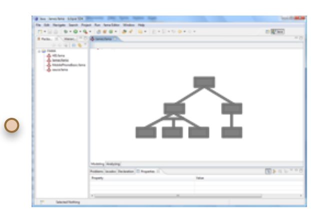


P = {Mobile Phone + Screen + Colour + Media + Camera}



Is P a valid product?

No, two constraints are violated



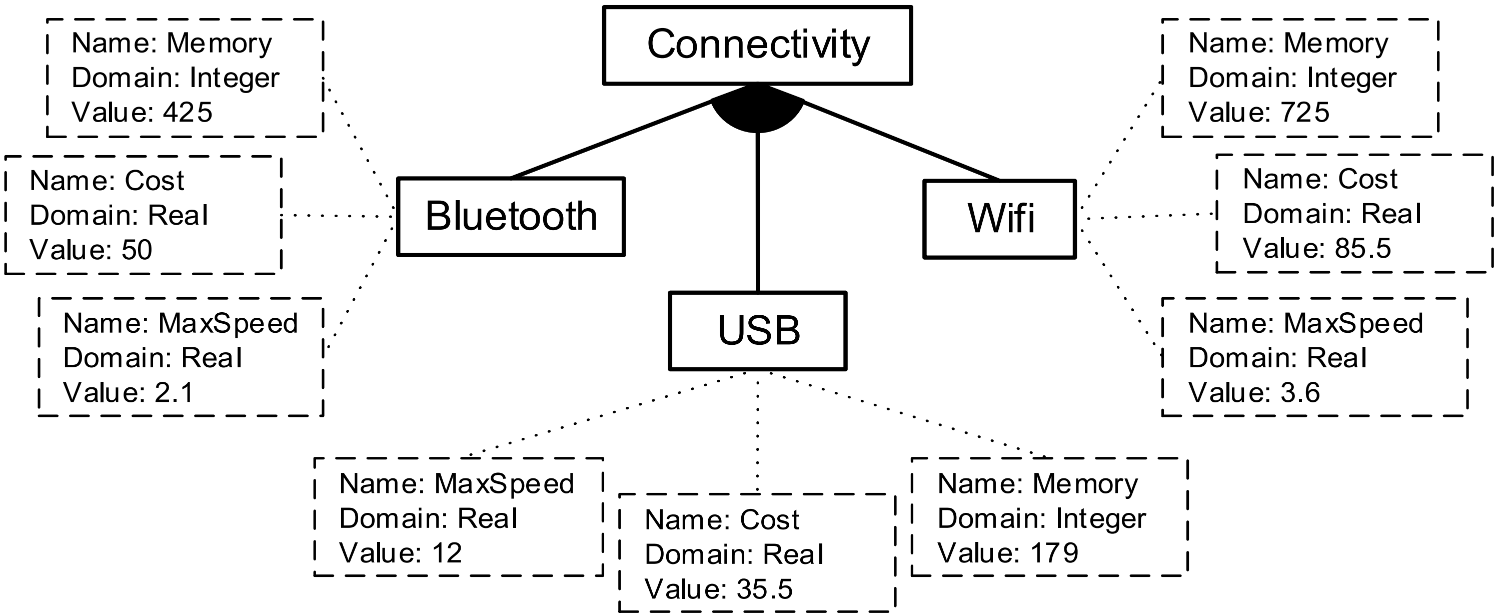
Introduction

Feature models

- Much interest in the automated analysis of feature models.
- Catalogues with up to 30 analysis operations
- Tools that implement these operations
  - e.g. FaMa Framework, SPLOT, …
- Automation is via e.g. SAT solvers.

Introduction

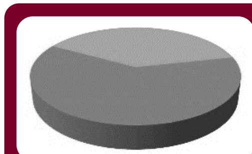
Attributed feature models



Introduction

Problem

Approach



Results

Conclusions

Problem
Motivation

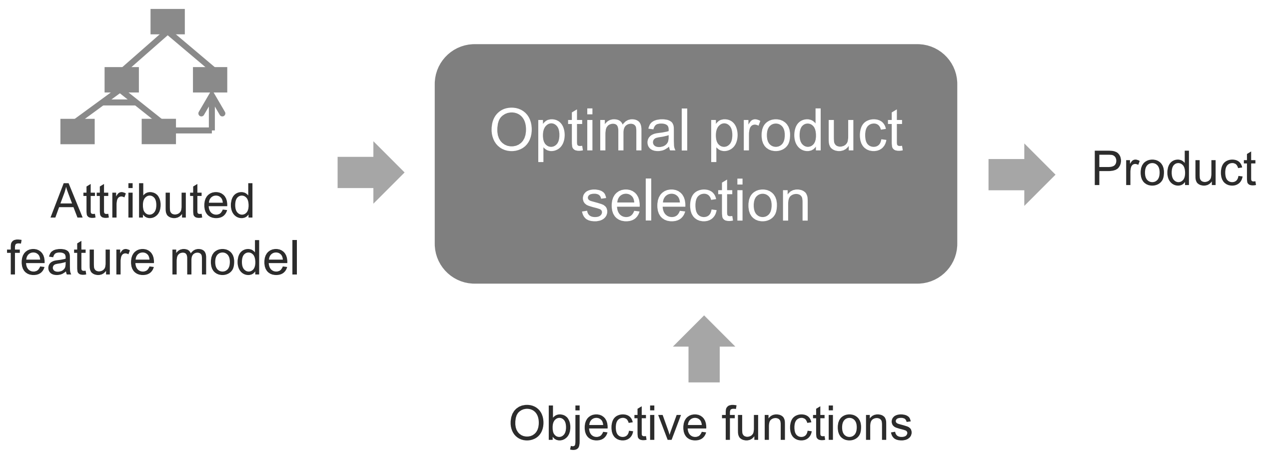
- Usually, it is not feasible to produce all valid products.
- Reasons for choosing a particular product:
  - What product to release first
    - many features, low cost, …
  - What product to test
    - different combinations of features, low cost …

Problem
Motivation

- We require a valid product:
  - We want to minimise the number of constraints failed.

- We want to optimise certain properties eg:
  - Cost (of building product)
  - Number of features
  - Historical faults

Problem



Ex: “Find a product that minimises the number of constraints violated, maximises the number of features, minimises cost, minimises the number of known defects, and minimises the number of changes since the last release.”

Problem

- Several aspects (objective functions) to optimise.

- So: a multi-objective optimisation problem.

- We could weight the objectives and optimise but ...

    - The choice of values for weights would be important.
    - How would we choose the weights?

Pareto Dominance

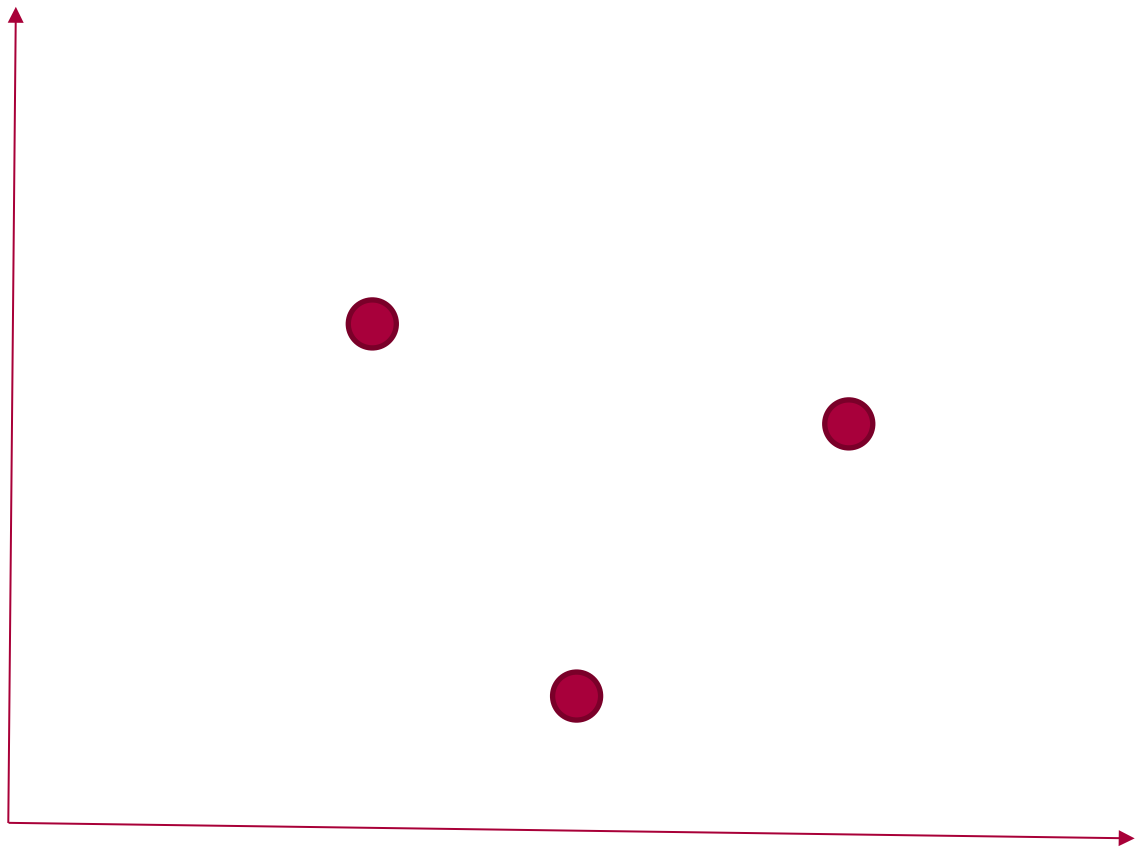
- Candidate (product) C1 dominates candidate C2 if:
  - C1 is at least as good as C2 on all criteria
  - C1 is better than C2 on at least one criterion.

- Used by some evolutionary optimisation algorithms.
- Aim is:
  - get as close as possible to the Pareto front.

- Returns a set of solutions:
  - These are incomparable under Pareto dominance
  - Provide *alternative* compromises.

Pareto Dominance

- Maximising – which dominate others?



Many-objective problems

- Many-objective problems
  - Those with four or more objectives
  - Typically, much more difficult to solve.

- Normally, Pareto dominance becomes ineffective

- There are algorithms that do not use Pareto dominance
  - e.g. IBEA, SPEA2+SDE, MOEA/D

Previous work

Initial work

- Recognised that it is a many-objective problem
- Applied several evolutionary multi/many-objective optimisation algorithms (EOMs) including:
  - NSGA-II, SPEA2, IBEA
- Used two feature models, up to 290 features.

- Difficult to find valid products.
- IBEA performed best.

Abdel Salam Sayyad, Tim Menzies, and Hany Ammar. On the Value of User Preferences in Search-based Software Engineering: A Case Study in Software Product Lines. *2013 International Conference on Software Engineering (ICSE '13)*

Previous work
Enhancements

- Improvements by Sayad et al. included:
  - Removing core features (those that are in all valid products)
  - Removing dead features (those that are in no valid products)
  - Planting an initial seed (a valid product).

Abdel Salam Sayyad, Joseph Ingram, Tim Menzies, and Hany Ammar. Scalable product line configuration: A straw to break the camel's back. *2013 IEEE/ACM International Conference on Automated Software Engineering*. 465–474.

Abdel Salam Sayyad. 2014. Evolutionary Search Techniques with Strong Heuristics for Multi-Objective Feature Selection in Software Product Lines. *Ph.D. Dissertation*. West Virginia University.

- Evaluated on more models:
  - Seven models.
  - Included one large model (6,888 features) – the Linux kernel feature (variability) model.

- Required an initial phase to find a seed
  - took approximately three hours for the large model.

Previous work
Enhancements

- Additional enhancements involved:
  - Adapting mutation (cannot be applied in certain circumstances, avoiding some invalid products).
  - Placing greater emphasis on the number of constraints that fail (a weight)

Abdel Salam Sayyad, Joseph Ingram, Tim Menzies, and Hany Ammar. Scalable product line configuration: A straw to break the camel's back. *2013 IEEE/ACM International Conference on Automated Software Engineering*. 465–474.

- New mutation and replacement operators that used a SAT solver

Christopher Henard, Mike Papdakis, Mark Harman, and Yves Le Traon. Combining Multi-Objective Search and Constraint Solving for Configuring Large Software product Lines. *2015 International Conference on Software Engineering (ICSE '15)*.

Previous work
Summary

- Search as a many-optimisation problem.
- Found to be difficult.
- A number of enhancements help.

- However:
  - Enhancements complicates the approach
  - New operators are more computationally expensive
  - Can be specific to certain optimisation algorithms

- Can we produce a simpler search-based approach?

R. M. Hierons, M. Li, X. Liu, S. Segura, and W. Zheng: SIP: Optimal Product Selection from Feature Models using Many-Objective Evolutionary Optimisation, *ACM Transactions on Software Engineering and Methodology*, 25 2, 2016.

Our initial work

- Initial motivation:
  - Try different evolutionary algorithms – those that have been found to be good at many-objective optimisation.

- We wanted approaches that return a range of solutions
  - Provide alternative products – user can choose between these.

- Initial experiments on two case studies:
  - WebPortal and E-Shop

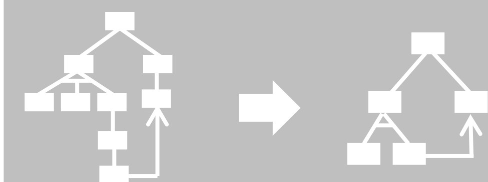
- It was difficult to find valid products (similar to earlier work).

Finding products

- We conjectured that two aspects made this difficult:
  - The presence of many constraints.
  - As the number of objectives increases, there is less evolutionary pressure towards valid products.
- We aimed to address these two points.
- We also aimed to use case studies with realistic attribute values.

Approach
SIP: Shrink and Prioritise

Novel encoding (ShrInk)



Prioritized objectives (Prioritise)

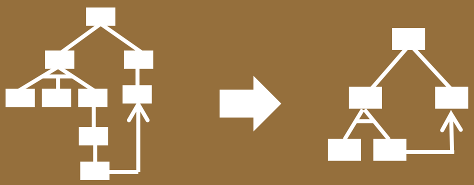
1 + n

Realistic case studies

EMO algorithms comparison

Approach
SIP: Shrink and Prioritise

Novel encoding (ShrInk)



Prioritized objectives (Prioritise)

1 + n

Realistic case studies

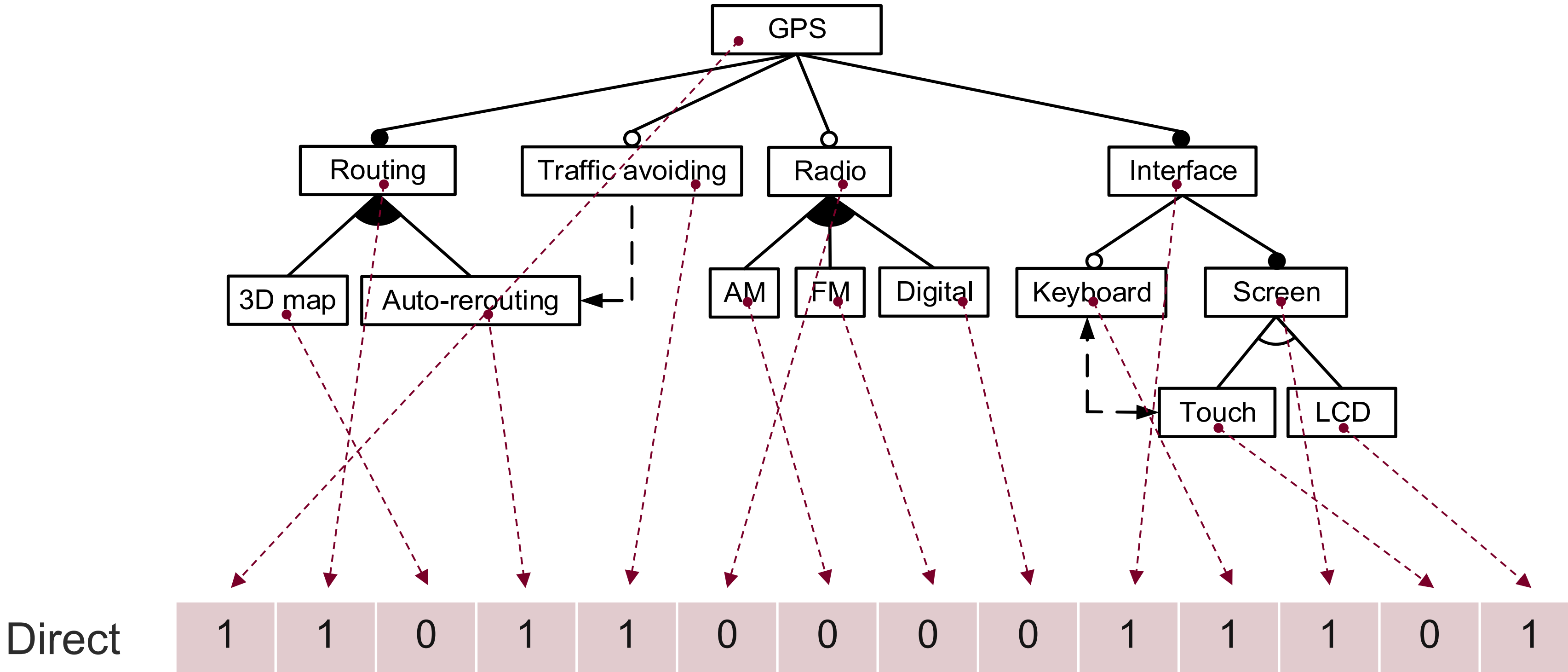
DRUPAL

amazon webservices™ | EC2

EMO algorithms comparison

Approach

Standard encoding



Approach
Novel encoding

- Basic idea:

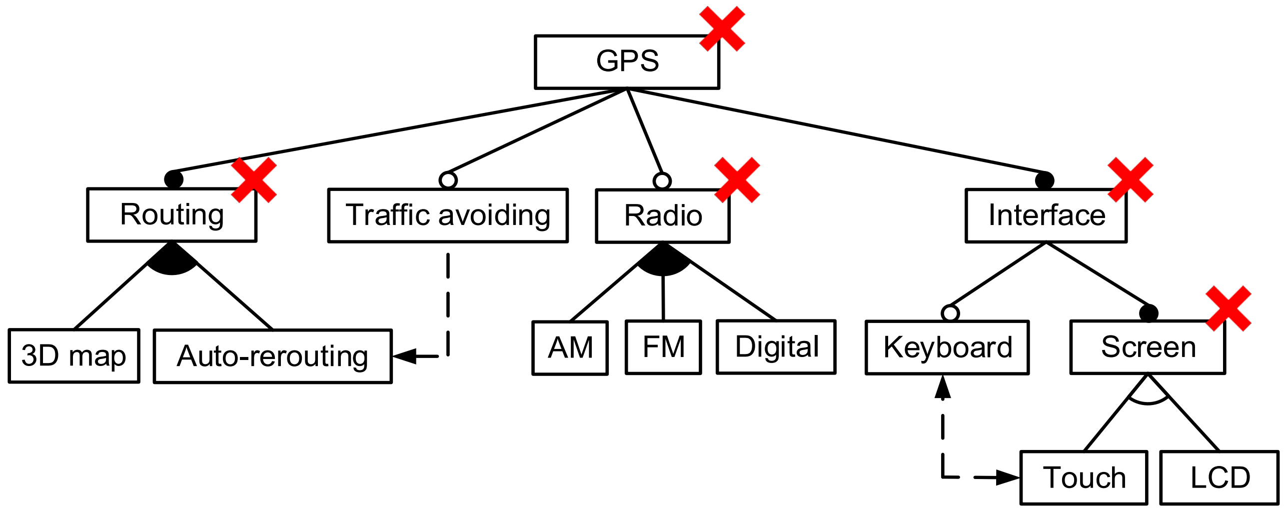
  - No need to include Core features (Sayad at al. and Henard et al. also observed this).

  - Did not remove dead features
    - Feature models should not have these.
    - They should be found and removed in advance.

  - Sometimes *we don't need to include some parents* (e.g. if children are in a OR relationship).

Approach

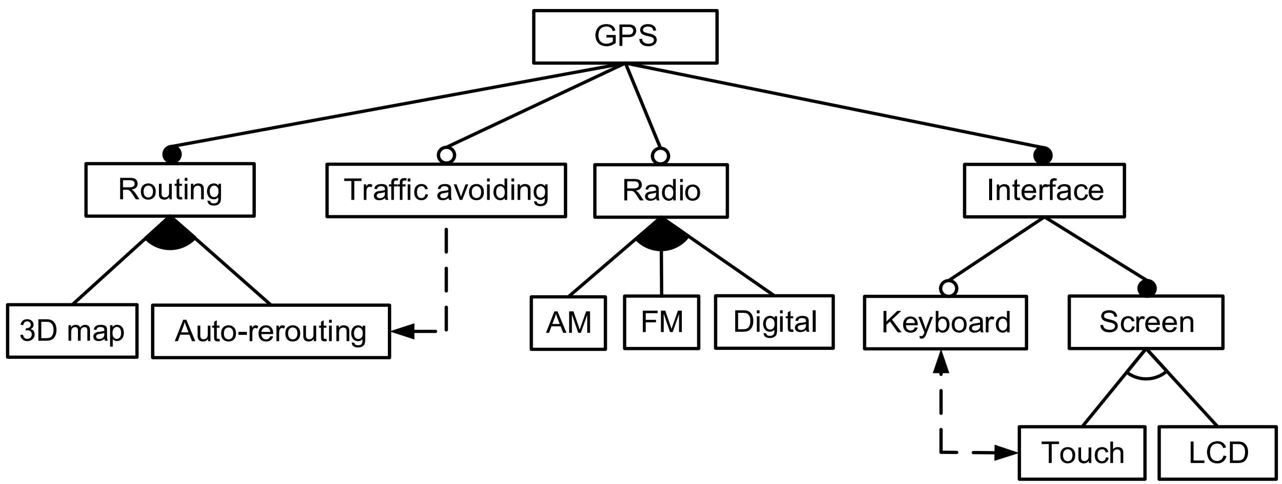
Novel encoding



Direct	1	1	0	1	1	0	0	0	0	1	1	1	0	1
SIP	✗	✗	0	1	1	✗	0	0	0	✗	1	✗	0	1

Approach

Novel encoding



Encoding	GPS	Routing	3D map	Auto-rerouting	Traffic avoiding	Radio	FM	AM	Digital	Interface	Keyboard	Screen	Touch	LCD
Direct	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SIP			✓	✓	✓		✓	✓	✓		✓		✓	✓

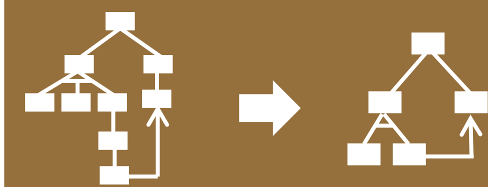
Approach
Novel encoding

- This helps since:
  - We have a smaller search space.
  - Some constraints can be enforced through the encoding.
    - Should make it easier to generate valid products.

Approach

SIP: Shrink and Prioritise

Novel encoding
(ShrInk)



Prioritized objectives
(Prioritise)

1 + n

Realistic case studies

DRUPAL

amazon webservices™ EC2

EMO algorithms comparison

Approach

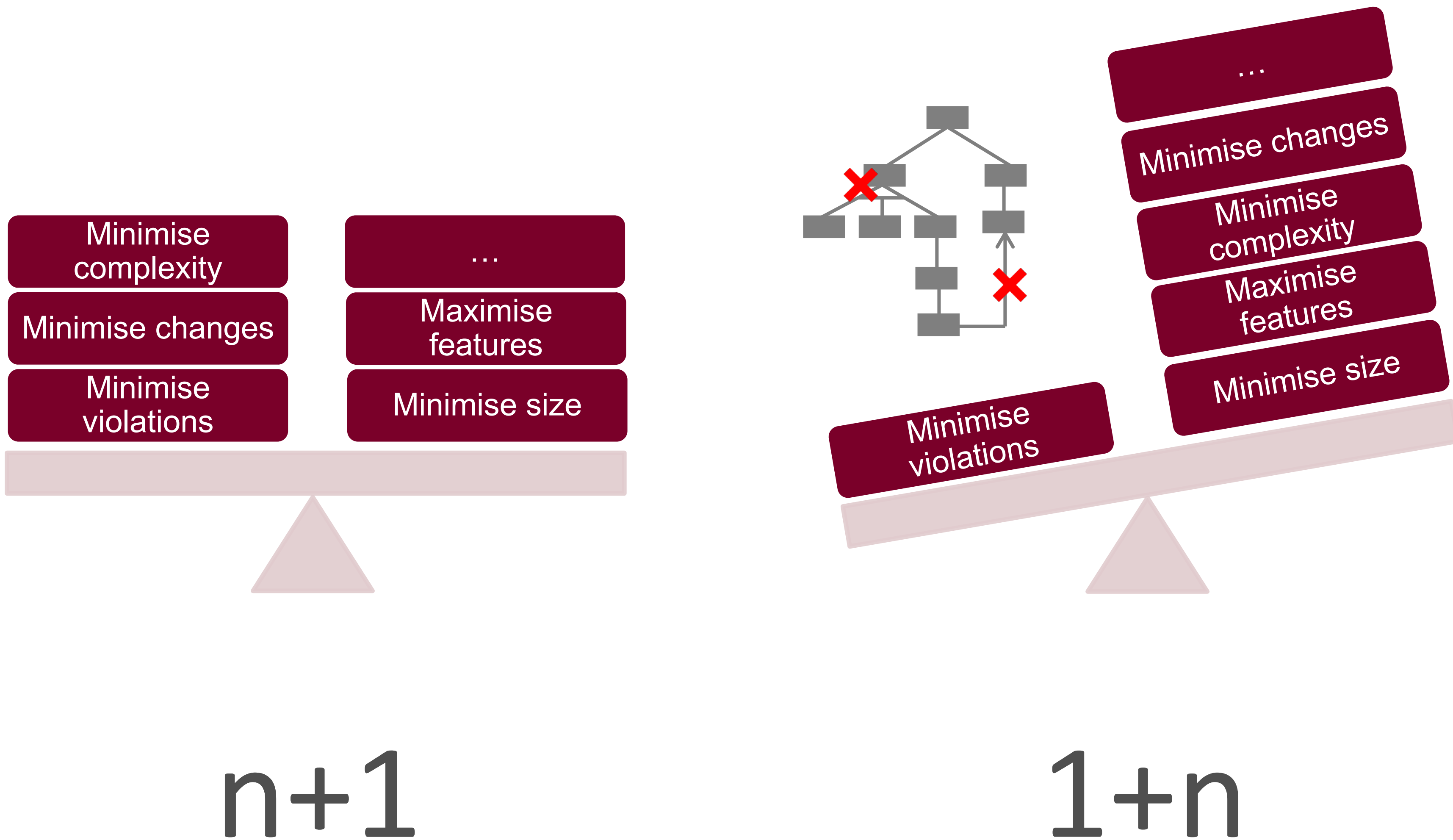
Prioritised objectives: 1+n

- Traditional, (n+1)-approach:
  - All of the objectives are considered together.
- The 1+n-approach:
  - We first consider the number of constraints failed, only then the remaining (n) objectives.
  - Motivation: invalid products have no value.
- Implemented as:
  - First compare individuals using number of constraints failed.
  - If equal on this, then compare as normal (for the given algorithm).

Approach
Implementing 1+n

- Mating selection:
  - Compare individuals on number of constraints failed.
  - If equal on this, then compare as normal (for the given algorithm).
- Environmental selection (determining which survive):
  - Group individuals on number of constraints failed.
  - Keep the best groups.
  - Use normal approach (for EMO) if we need to select a subset from a group.

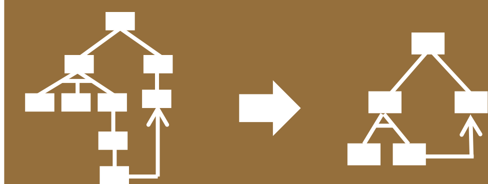
Approach
Prioritised objectives



Approach

SPI: Shrink and Prioritise

Novel encoding (ShrInk)



Prioritized objectives (Prioritise)

1 + n

Realistic case studies

DRUPAL

amazon web services™ | EC2

EMO algorithms comparison

Approach
Realistic case studies

	DRUPAL	amazon web services
Features	48	79
Attributes	22	17
Products	$2.09 \cdot 10^9$	66,528

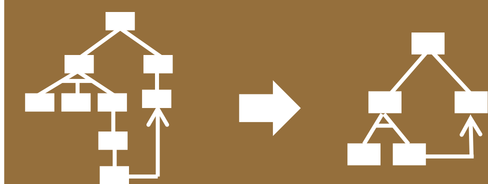
Published FMs with random attributes.

Randomly generated FM with 10,000 features.

Approach

SIP: Shrink and Prioritise

Novel encoding
(ShrInk)



Prioritized objectives
(Prioritise)

1 + n

Realistic case
studies

DRUPAL

amazon web services™ EC2

EMO algorithms
comparison

Evaluation
Encodings

- Four used:
  - *Direct encoding* (one bit for each feature). Used in initial work by Sayyard et al.
  - *Core encoding* – do not include core features. Used by Sayyad et al. and Henard et al.
  - *Hierarchical encoding* – do not include parents unless necessary
  - The (proposed) *novel encoding*
    - Combines the above enhancements
    - (core + hierarchical)

Evaluation
EMO algorithms

NSGA-II
[Deb et al. 2002]

IBEA
[Zitzler et al. 2004]

SPEA2+SDE NEW
[Li et al. 2014]

MOEA/D-WS NEW
[Zhang and Li 2007]

MOEA/D-TCH NEW
[Zhang and Li 2007]

MOEA/D-PBI NEW
[Zhang and Li 2007]

Evaluation

Initial experiments

- We initially used:
  - (n+1)-approach
  - Direct encoding

  - Two feature models (E-shop and WebPortal).

- Found that:
  - All EMOs returned relatively few valid products.

- Motivated the enhancements.

Evaluation

Set of Subjects

SPL	# Features	# CTC	# Attributes Per feature
BerkleyDB	13	0	4
ERS	36	0	7
WebPortal	43	6	4
E-shop	290	21	4
Drupal	48	21	22
AmazonEC2	79	0	17
Random (generated by SPLAR)	10,000	0	4

First four previously used

- All feature models were written in the SXFM format.

Evaluation
Objectives

- For the four previously used models:
  - We used the same five objectives as previous experiments
- For Amazon and Drupal, two experiments for each:
  - We used the same five objectives (synthetic values).
  - We had eight objectives for each (realistic values).
- For the large randomly generated model:
  - The same five objectives as previous work (for comparison).

- All sets included correctness (number of constraints not satisfied).

Subjects with realistic attributes

- Two case studies.
- Drupal:
  - Previously, 22 non-functional attributes extracted from GIT repository.
- Amazon EC2:
  - Configuration space of Amazon Elastic Computing Service.
  - Attributes were not fixed – there are constraints.
  - We randomly assigned values that satisfied the constraints.

Evaluation

Experimental setup

- Each experiment repeated 30 times.
- Termination criterion: 50,000 fitness evaluations
- Population size:
  - 100 for most
  - For MOEA/D – closest possible integer to 100 (depends on number of objectives).

- Parameter values used:
  - Those recommended in the literature.

Evaluation

Performance metrics

- Three metrics:
  - Hypervolume (HV): the volume of the objective space between the solutions and a reference point.
    - Only used the valid solutions returned.
  - The number of executions that returned at least one valid product (VN).
  - Mean rate of valid individuals in the final population (VR).
- Note that HV and VR were averaged over populations that had at least one valid solution.

Results

Results

Results with E-shop

- An example, direct encoding and (n+1)

Algorithm	HV	VN (/30)	VR
NSGA-II	0.0000	0	0%
IBEA	0.0000	0	0%
MOEA/D-WS	0.016184	26	21.42%
MOEA/D-TCH	0.0000	0	0%
MOEA/D-PBI	0.18815	4	10.50%
SPEA2+SDE	0.0000	0	0%

Results

Results with E-shop

- An example, novel encoding and (n+1)

Algorithm	HV	VN (/30)	VR
NSGA-II	0.136545	13	100%
IBEA	0.169191	16	100%
MOEA/D-WS	0.184810	5	100%
MOEA/D-TCH	0.199697	1	100%
MOEA/D-PBI	0.166157	5	100%
SPEA2+SDE	0.144341	15	100%

Results

Results with E-shop

- An example, core encoding and (n+1)

Algorithm	HV	VN (/30)	VR
NSGA-II	0.003343	28	2.07%
IBEA	0.26741	30	33.91%
MOEA/D-WS	0.074223	30	26.74%
MOEA/D-TCH	0.0000	0	0%
MOEA/D-PBI	0.070765	30	30.62%
SPEA2+SDE	0.0000	0	0%

Results

Results with E-shop

- An example, novel encoding and 1+n

Algorithm	HV	VN (/30)	VR
NSGA-II	0.162943	30	100%
IBEA	0.190496	30	100%
MOEA/D-WS	0.222875	30	100%
MOEA/D-TCH	0.226257	30	100%
MOEA/D-PBI	0.192485	30	100%
SPEA2+SDE	0.159930	30	100%

Results

Results with E-shop

- Direct encoding:
  - With, (n+1), only two algorithms returned valid solutions (VR of 4/30 and 26/30).
  - With 1+n, all (sometimes) produced valid solutions.
- Hierarchical and core encodings:
  - More effective, core better than hierarchical.
- Novel encoding:
  - Almost all EMOs had 100% VR.

Results

Results with E-shop

- HV for novel:
  - With 1+n, the best results were with some variants of MOEA/D.
  - With (n+1), IBEA best (similar to previous work).
- We tried E-shop with 500,000 evaluations:
  - Better results.
  - However, only 1+n (with core/novel) gave 100% VN for all algorithms.

Results

Results with Webportal

- Results similar to E-shop but better.
  - Seems to be an easier problem.
- 1+n approach always gave 100% VN and VR.
- (n+1) approach gave 100% VN, VR for core and novel encodings.
- IBEA tended to give the best HV.

Results

Other results (five objectives)

- Similar results.
- 1+n approach and novel always gave 100% VN and VR.
- The only combination that did this.
- HV comparisons were quite variable. Highest values for:
  - SPEA2+SDE (Amazon, Berkley)
  - IBEA (Drupal)
  - NSGA-II (ERS).

Results

Other results (real attributes)

- Drupal:
  - Relatively easy to find valid products.
  - Only (n+1)/direct failed to have 100% VR,VN for all EMOs.
- Amazon:
  - Results were much poorer.
  - Almost no experiments produced valid products with direct/(n+1).
  - Most were effective with novel/1+n.

Results

Amazon (real attributes)

- Results for 1+n with the novel encoding.

Algorithm	HV	VN (/30)	VR
NSGA-II	0.001844	30	100%
IBEA	0.001897	17	100%
MOEA/D-WS	0.001877	30	100%
MOEA/D-TCH	0.001688	30	100%
MOEA/D-PBI	0	0	0%
SPEA2+SDE	0.002001	30	100%

Results

Large, randomly generated model

- Search found this difficult.
- Only novel/1+n returned valid solutions
  - All others had 0% for VN, VR.

Algorithm	HV	VN (/30)	VR
NSGA-II	0.014588	24	100%
IBEA	0.020762	25	100%
MOEA/D-WS	0.042142	15	100%
MOEA/D-TCH	0.025037	19	100%
MOEA/D-PBI	0.042513	18	100%
SPEA2+SDE	0.018173	28	100%

Results

Best performing (novel, 1+n)

- Comparing HV values.
- Not all differences are statistically significant (many were).

Model	Best performing MOEA
E-shop. 50k	MOEA/D-TCH
E-shop, 500k	MOEA/D-WS
WebPortal	IBEA
Amazon	SPEA2+SDE
Berkley	SPEA2+SDE
Drupal	IBEA
ERS	NSGA-II
Drupal, real	SPEA2+SDE
Amazon, real	SPEA2+SDE
Large, random	SPEA2+SDE

Statistical tests used

- First tested the hypothesis that all EMO algorithms perform equally (different techniques, same approach):
  - Kruskal-Wallis test. Always rejected.
- Then used pairwise comparisons:
  - post-hoc Kruskal-Wallis test
  - with Bonferroni adjustment
- Effect size using:
  - Mann-Whitney U test
  - many of the differences had large effect sizes (between 0.8 and 0.9).

Results

Time (large model)

- Time in seconds.
- For novel encoding; results similar (but usually slower) for other encodings.

EMO	(n+1)	1+n
NSGA-II	101.76	98.758
IBEA	154.399	131.465
MOEA/D-WS	135.98	104.44
MOEA/D-TCH	132.9	99.7
MOEA/D-PBI	131.9	103
SPEA2+SDE	247.786	223.777

Results

The choice of the encoding and approach have more impact on performance than the choice of EMO algorithm

There is not clear “best” EMO algorithm.

The 1+n approach, where one objective is given priority over the rest, is more effective that treating all the objectives equally.

The encoding does affect performance, with the proposed encoding proving to be most effective.

Future Work

- Incorporate novel operators from e.g. Henard et al.
- Further enhance the representation:
  - Aim to encode more constraints.
- Consider approaches that use repair.
- Additional studies:
  - Others with realistic attribute values – do the approaches perform less well on these?
  - Varying the number of objectives.

Summary

- The encoding affects performance, with the proposed encoding providing to be most effective.
- The 1+n approach is more effective than the (n+1) approach.
- Both enhancements help.
- There is no clear “best” EMO algorithm.
- SPEA2+SDE performs best most often (and for the most difficult) but takes more time.

SIP: Optimal Product Selection from Feature Models using Many-Objective Evolutionary Optimisation

Robert M. Hierons[(1)], Miqing Li[(1)], XiaoHui Liu[(1)], Sergio Segura[(2)], Wei Zheng[(3)]

[(1)] Brunel University London, UK
[(2)] University of Seville, Spain
[(3)] Northwestern Polytechnical University, China

Thanks!

Hypervolume

- Influenced by:
  - Choice of reference point.
  - Scaling of space (dimensions).

- Reference point used:
  - Nadir point of the problem's range (the point constructed with the worst value on each objective).

- Scaling:
  - we normalised the objective values according to the range of values in the objective space.

Hypervolume

- Computed exact values when fewer than eight objectives.
- Literature suggests this is infeasible with seven or more objectives
- For the large example (eight objectives)
  - We estimated the HV result.
  - We used Monte Carlo sampling with 10,000,000 sampling points.