On the Performance of Multiple Objective Evolutionary Algorithms for Software Architecture Discovery

Aurora Ramírez, José Raúl Romero and Sebastián Ventura

Dept. of Computer Science and Numerical Analysis
University of Córdoba, Spain

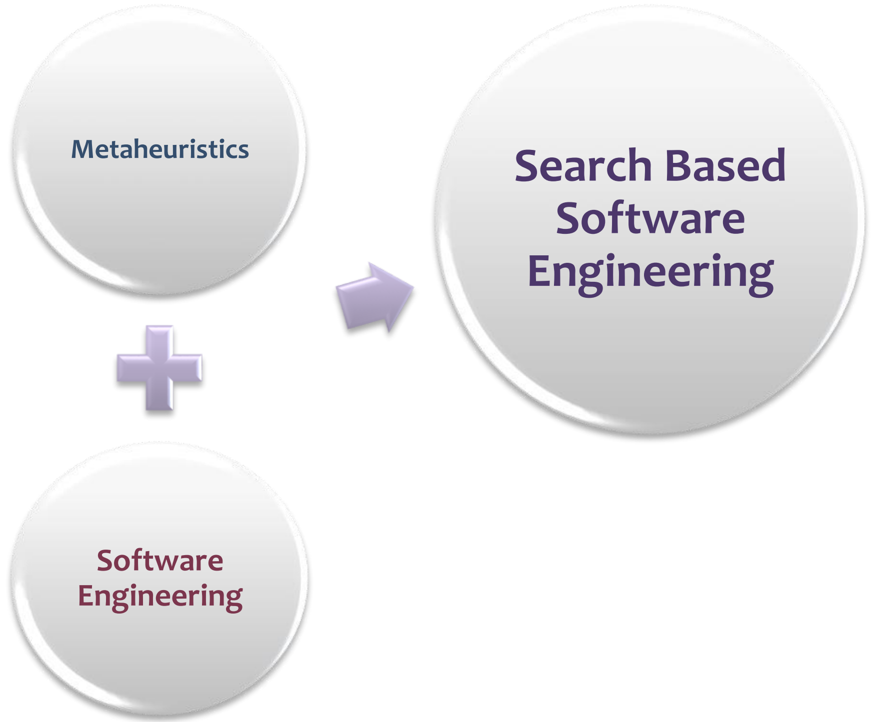
Search Based Software Engineering @ 16th Annual Conference on GECCO
July 12-16, 2014 Vancouver, BC, Canada

Contents

- Introduction
- Evolutionary Discovery of Software Architectures
  - The software design problem
  - The search-based approach
  - Multi- and many-objective evolutionary algorithms
- Experiments and results
  - Problem instances and set-up
  - From the perspective of the evolutionary performance
  - From the perspective of the decision-maker
- Concluding remarks

Introduction

- Search Based Software Engineering (SBSE)
  - Apply metaheuristics to Software Engineering tasks
  - All stages of the software development
- More specifically...
  - Design phase
  - Architectural analysis



Introduction

- Software architectures are important design artefacts in the early software conception

- Software architects face to:
  - Multiple functional and, mainly, non functional requirements
  - A wide set of design decisions
  - Discovery of software structures and their interactions

- SBSE can support in design tasks: efficient search of architectural alternatives

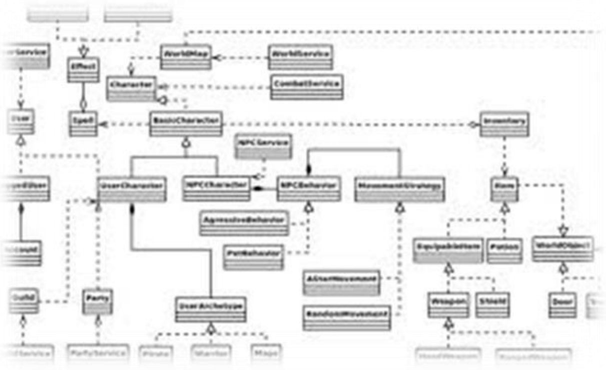
Introduction

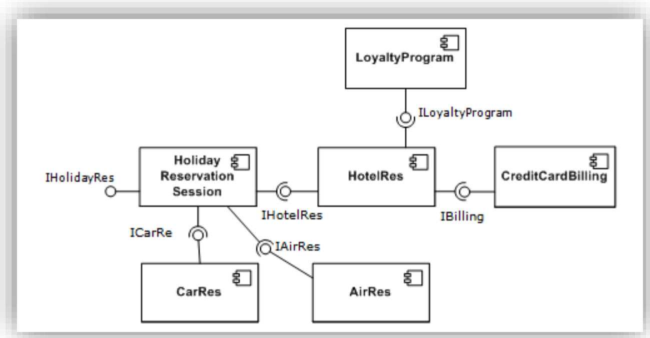
- Multi-objective Evolutionary Algorithms
  - Frequently applied in SBSE
  - Two or three objectives and classical algorithms (SPEA2, NSGA-II)
- Many-objective Evolutionary Algorithms
  - Rarely explored in problem domains like SBSE
  - Interesting alternative for high dimensional search spaces
- Architecture Discovery as a multi/many objective optimization problem
  - Comparative study of multi- and many-objective EAs
  - Scalability analysis: from 2 to 6 objectives
  - Different subsets of objectives related to software design

Evolutionary Discovery of Software Architectures

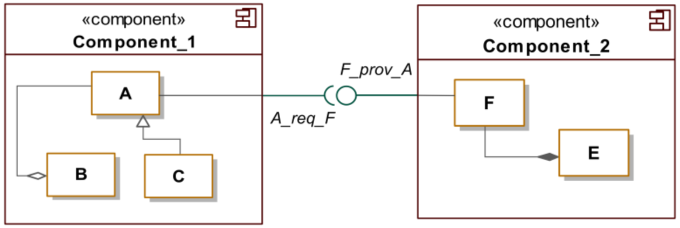
The software design problem

- Component-based software architectures in a nutshell:
  - **Component**: cohesive groups of classes
  - **Interface**: relationships between classes allocated in different components
  - **Connector**: pair of required and provided interfaces
- Focused on non-functional requirements
- Highly combinatorial problem
  - Different architectural styles
  - No prefixed structure

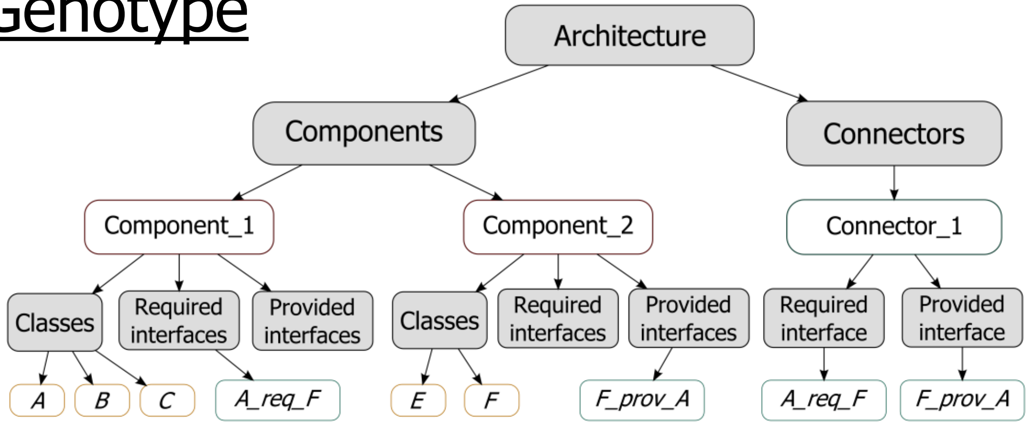




Phenotype



Genotype



Genetic operator

- A roulette-based mutation operator to:
  - Add a component
  - Remove a component
  - Merge two components
  - Split a component
  - Move a class

Initialization and constraints

1. Randomly distribution of classes
   - ✓ No empty components and no replicated classes
2. Set interfaces and connectors
   - ✗ Isolated or mutually dependant components

- The six objectives based on modularity and reusability
  - Intra-modular Coupling Density (ICD)

    $$ICD_i = \frac{CI_i^{in}}{CI_i^{in} + CI_i^{out}} \qquad ICD = \sum_{i=1}^{n} ICD_i$$

  - External Relations Penalty (ERP)

    $$ERP = \sum_{i=1}^{n}\sum_{j=1}^{n}\left[w_{as} \cdot n_{as_{ij}} + w_{ag} \cdot n_{ag_{ij}} + w_{co} \cdot n_{co_{ij}} + w_{ge} \cdot n_{ge_{ij}}\right]$$

  - Encapsulation (Enc)

    $$Enc_i = \frac{\#inner_{classes}}{\#total_{classes}} \qquad Enc = \frac{1}{n} \cdot \sum_{i=1}^{n} Enc_i$$

  - Critical Size (CS)

    $$CC_i = \begin{cases} 1 & if\ \ size(i) > threshold \\ 0 & otherwise \end{cases} \qquad CS = \sum_{i=1}^{n} CC_i$$

  - Instability (Ins)

    $$Ins_i = \frac{EC_i}{EC_i + AC_i} \qquad Ins = \frac{1}{n} \cdot \sum_{i=1}^{n} Ins_i$$

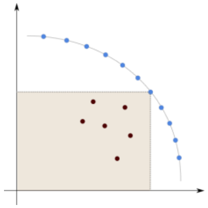
  - Groups/Components Ratio (GCR)

    $$GCR = \frac{\#cgroups}{\#components}$$

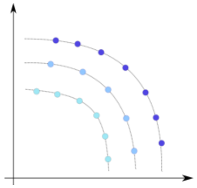
Multi- and many-objective evolutionary algorithms

SPEA2

- Generational algorithm
- Fitness = strength + density
- Binary tournament selection
- Archive with fixed size to store non dominated solutions

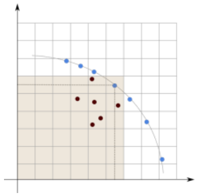


NSGA-II



- Non-dominated sorting
- Selection based on dominance and crowding distance
- Promotes the survival of non dominated solutions

ε-MOEA

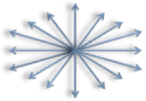


- Steady state algorithm
- Landscape partition in hypercubes
- ε-dominance relation
- Archive of solutions

GrEA

- Inspired by NSGA-II
- Number of divisions as a parameter
- Grid-based metrics for crowding distance and spread of solutions

MOEA/D



- Decomposition approach
- A weight vector for each individual
- Neighborhood information
- Fitness based on a reference point

Experiments and results

Problem instances and set-up

- 6 diverse software designs
- All possible combinations of 2, 4 and 6 objectives per instance
- 30 runs
- Quality indicators:
  - Hypervolume (HV)
  - Spacing (S)
- Friedman and Holm's statistical tests

Problem	*#Class*	*#Relationships*					*#Int*
		As	De	Ag	Co	Ge	
Aqualush	58	69	6	0	0	20	74
Datapro4j	59	3	4	3	2	50	12
Java2HTML	53	20	66	15	0	15	170
JSapar	46	7	33	21	9	19	80
Marvin	32	5	11	22	5	8	28
NekoHTML	47	6	17	15	18	17	46

Common parameters	
Population Size	100, 120, 126
Max. Evaluations	10,000, 15,000, 20,000
Min-Max. Components	2-8
Mutator weights	
	$w_{add} = 0.2, w_{remove} = 0.3, w_{merge} = 0.2$
	$w_{split} = 0.1, w_{move} = 0.2$
ERP metric weights	
	$w_{as} = 1, w_{ag} = 3, w_{co} = 1, w_{ge} = 5$
CS threshold	0.3
SPEA2 parameters	
Parents selector	Binary tournament
External population size	50
k-th neighboor	12
ϵ-MOEA parameters	
ϵ values	
	$\epsilon_{ICD} = 0.25,\ \epsilon_{ERP} = 5,\ \epsilon_{GCR} = 0.1$
	$\epsilon_{CS} = 1,\ \epsilon_{Ins} = 0.05,\ \epsilon_{Enc} = 0.05$
MOEA/D parameters	
Neighboorhood size (τ)	8
Max. Replacements (Nr)	2
H	99,7,4
GrEA parameters	
Number of divisions (div)	12

Experiments and results

From the perspective of the evolutionary performance

2 objectives

- Difficult trade-off between HV and S
- SPEA2 achieves good dispersion of the front
- NSGA-II, ϵ-MOEA and GrEA usually outperform SPEA2 and MOEA/D in HV
- Poor performance of MOEA/D

Objectives	SPEA2		NSGA-II		ϵ-MOEA		MOEA/D		GrEA	
	HV	S	HV	S	HV	S	HV	S	HV	S
ICD-ERP	3.67	2.67	2.58	3.42	**1.50**	**1.33**	*4.67*	*4.67*	2.58	2.92
ICD-GCR	*4.17*	**2.33**	2.75	2.66	2.17	3.67	*4.17*	3.83	**1.75**	2.50
ICD-Ins	*4.17*	**1.50**	*3.25*	2.83	**1.17**	*4.83*	*4.17*	2.33	2.25	3.50
ICD-CS	*4.50*	2.50	*3.25*	3.25	**1.17**	4.08	*3.83*	**2.08**	2.25	3.08
ICD-Enc	*4.17*	**1.33**	2.58	*4.08*	2.17	2.17	*4.17*	2.50	**1.92**	*4.92*
ERP-GCR	**2.83**	**3.00**	3.25	**3.00**	**2.83**	**3.00**	**2.83**	**3.00**	3.25	**3.00**
ERP-Ins	**2.83**	**3.00**	3.25	**3.00**	**2.83**	**3.00**	**2.83**	**3.00**	3.25	**3.00**
ERP-CS	**2.83**	**3.00**	3.25	**3.00**	**2.83**	**3.00**	**2.83**	**3.00**	3.25	**3.00**
ERP-Enc	1.92	**2.00**	**1.67**	2.75	3.50	2.67	3.67	3.50	*4.25*	4.08
GCR-Ins	**2.83**	**3.00**	3.25	**3.00**	**2.83**	**3.00**	**2.83**	**3.00**	3.25	**3.00**
GCR-CS	**2.83**	**3.00**	3.25	**3.00**	2.83	**3.00**	**2.83**	**3.00**	3.25	**3.00**
GCR-Enc	2.75	**1.67**	**1.67**	3.75	3.33	2.00	*4.17*	3.33	3.08	*4.25*
Ins-CS	**2.83**	**3.00**	3.25	**3.00**	**2.83**	**3.00**	**2.83**	**3.00**	3.25	**3.00**
Ins-Enc	3.08	**1.00**	**1.92**	*3.75*	2.92	*3.50*	3.92	2.17	3.17	*4.58*
CS-Enc	2.83	**1.33**	**1.75**	3.17	3.50	*4.08*	*4.33*	2.83	2.58	*3.58*

Algorithms perform similarly for some combinations of objectives (local and global optima)

Experiments and results

From the perspective of the evolutionary performance

4 objectives

Objectives	SPEA2		NSGA-II		ϵ-MOEA		MOEA/D		GrEA	
	HV	S	HV	S	HV	S	HV	S	HV	S
ICD-ERP-GCR-Ins	3.17	*2.33*	1.92	*4.75*	**1.50**	**1.83**	*4.00*	3.67	*4.42*	2.42
ICD-ERP-GCR-CS	*4.00*	*3.00*	2.25	*3.42*	**1.50**	**1.00**	*4.33*	*4.50*	2.92	*3.08*
ICD-ERP-GCR-Enc	3.17	1.83	1.92	*4.08*	**1.50**	**1.17**	*4.00*	*3.67*	*4.42*	*4.25*
ICD-ERP-Ins-CS	*4.33*	2.00	2.58	*4.42*	2.17	**1.00**	4.00	*3.67*	**1.92**	*3.92*
ICD-ERP-Ins-Enc	*4.00*	1.67	2.08	*4.08*	**1.33**	**1.33**	*3.83*	3.33	*3.75*	*4.58*
ICD-ERP-CS-Enc	*4.17*	1.83	2.08	*4.08*	**1.50**	**1.17**	*4.00*	3.00	3.25	*4.92*
ICD-GCR-Ins-CS	*4.17*	2.33	3.08	*4.08*	**1.50**	**1.00**	*4.17*	3.50	2.08	*4.08*
ICD-GCR-Ins-Enc	*4.17*	**1.33**	2.25	*3.75*	**1.17**	1.67	*4.00*	*3.67*	*3.41*	*4.58*
ICD-GCR-CS-Enc	*4.33*	2.00	2.25	*4.08*	**1.67**	**1.33**	*4.00*	3.00	2.75	*4.58*
ICD-Ins-CS-Enc	*4.17*	2.17	2.42	*4.25*	**1.67**	**1.17**	*4.17*	2.67	2.58	*4.75*
ERP-GCR-Ins-CS	**2.83**	**3.00**	3.25	**3.00**	**2.83**	3.00	**2.83**	**3.00**	3.25	**3.00**
ERP-GCR-Ins-Enc	2.33	2.00	**1.67**	*4.58*	3.17	**1.83**	3.08	3.83	*4.75*	2.75
ERP-GCR-CS-Enc	2.00	2.50	**1.75**	*4.58*	3.67	**1.17**	3.00	*3.33*	*4.58*	*3.42*
ERP-Ins-CS-Enc	2.33	2.17	**1.67**	*4.58*	3.33	**1.17**	2.92	*3.50*	*4.75*	*3.58*
GCR-Ins-CS-Enc	3.00	2.67	**1.67**	*3.75*	2.67	**1.00**	2.92	*3.17*	*4.75*	*4.42*

- Multi-objective algorithms decrease their performance
- ϵ-MOEA obtains the best rankings for both indicators

6 objectives

- ϵ-MOEA has significant differences with most of the algorithms (HV) and good spacing values
- SPEA2 maintains a substantial diversity

Objectives	SPEA2		NSGA-II		ϵ-MOEA		MOEA/D		GrEA	
	HV	S	HV	S	HV	S	HV	S	HV	S
ICD-ERP-GCR-Ins-CS-Enc	*3.83*	**1.17**	2.42	*3.58*	**1.33**	1.83	*3.50*	*4.33*	*3.92*	*4.08*

Experiments and results

From the perspective of the decision-maker

- SPEA2
  - 👍 Variety of architectures (types and number)
  - 👎 Low quality solutions
- NSGA-II
  - 👍 Good scalability
  - 👎 Problems with complex instances
- GrEA
  - 👍 Trade-off between metrics
  - 👎 Strong tendency to certain types of solutions
- MOEA/D
  - 👍 Generates more non-dominated solutions
  - 👎 Diversity is not preserved in the external population
- ϵ-MOEA
  - 👍 Good trade-off between high quality and diversity
  - 👍 Low execution time and ability to remove invalid solutions
  - 👎 Some problems with specific combinations of objectives

Experiments and results

From the perspective of the decision-maker

- The selection of metrics has an important influence on the solutions found
  - Number of components comprising the architecture
  - Types of components and interactions

- The trade-off between design criteria
  - Instability and Encapsulation can reach good values in all the problems
  - ERP and GCR tend to complement each other well
  - Critical Size is usually demoted by other metrics
  - ICD is the most difficult metric to optimize

Concluding Remarks

- **Conclusions**
  - A first comparative study of multi- and many-objective evolutionary algorithms in Search-based Software Design
  - Different number and combinations of objectives: close to the reality
  - Strengths and weaknesses of each algorithm from the architect's expectations
- **Future Work**
  - A more in-depth analysis of the most fitting algorithms for dealing with each specific set of architectural requirements
  - To extend the catalogue of metrics and used algorithms

On the Performance of Multiple Objective Evolutionary Algorithms for Software Architecture Discovery

Aurora Ramírez, José Raúl Romero and Sebastián Ventura

Contact at:

jrromero@uco.es

Thank you!