On the Effect of Local Search in the Multi-objective Evolutionary Discovery of Software Architectures

Aurora Ramírez, José Raúl Romero, Sebastián Ventura

Dept. Computer Science and Numerical Analysis

University of Córdoba, Spain

Evolutionary Methods and Machine Learning in Software Engineering, Testing and Software Engineering Repositories

IEEE Congress on Evolutionary Computation (IEEE CEC)

June 5-8, 2017 – San Sebastián (Spain)

Contents

1. **Introduction**
2. **Proposed multi-objective memetic algorithms**
   - Local search as genetic operator
   - Local search as post-processing
3. **Experiments and results**
   - Analysis of quality indicators
   - Influence on the Pareto front
4. **Concluding remarks**

Introduction

Search Based Software Engineering

Search Based Software Engineering ➤ Search Based Software Design ➤ Architecture Discovery

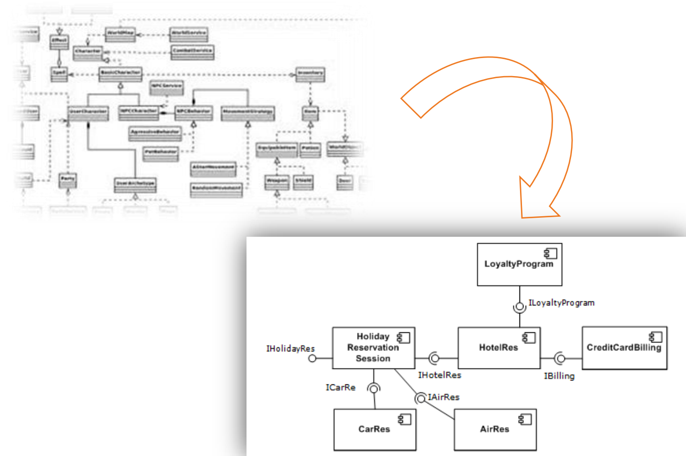
- **Evolutionary methods in software design**
  - ➢ Efficient exploration of design alternatives
  - ➢ Multi-objective search

- **Software architectures**
  - ➢ Important design artefacts in the early software conception
  - ➢ Identification of functional blocks and their interactions

Introduction

Multi-objective evolutionary discovery of architectures

We want to automatically identify the component-based architecture of a system from its analysis model (represented as a class diagram)



NSGA-II

- Component diagrams encoded as trees
- Four objective functions: design metrics
- The mutation performs 5 architectural transformations
- Closer to human-designed architectures

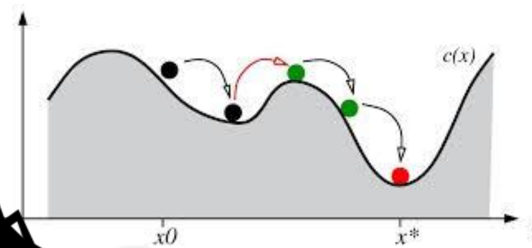
Metric	Formula
ICD: Intra-modular Coupling Density	$ICD_i = ((\#cl_t - \#cl_i)/\#cl_t) \cdot (CI_i^{in}/(CI_i^{in} + CI_i^{out}))$ $ICD = \sum_{i=1}^{n} ICD_i/n$
ERP: External Relations Penalty	$ERP = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_{as} \cdot n_{as_{ij}} + w_{ag} \cdot n_{ag_{ij}}$ $+ w_{co} \cdot n_{co_{ij}} + w_{ge} \cdot n_{ge_{ij}})$
CS: Critical size	$CS = \sum_{i=1}^{n} CC_i,\ CC_i = 1\ if\ \#cl_i > threshold,\ 0\ otherwise$
CB: Component Balance	$SB(n) = \frac{n-\gamma}{\mu-\gamma}\ if\ n < \mu\ ,\ = 1 - \frac{n-\mu}{\omega-\mu}\ if\ \mu < n < \omega\ ,\ = 0\ if\ n \geq \omega$ $CSU(n) = 1 - Gini(\{\#cl_i \forall i \in [1, n]\})\ ,\ CB = SB(n) \cdot CSU$

RQ: How can local search be effectively integrated in the multi-objective evolutionary discovery of software architectures?

Introduction
Multi-objective memetic algorithms

Evolutionary algorithm (MOEA) + **Local search (LS)** = **Memetic algorithm (MOMA)**

- **Local search** techniques
  a. Hill climbing (HC)
  b. Simulated annealing (SA)
  c. Tabu search (TS)



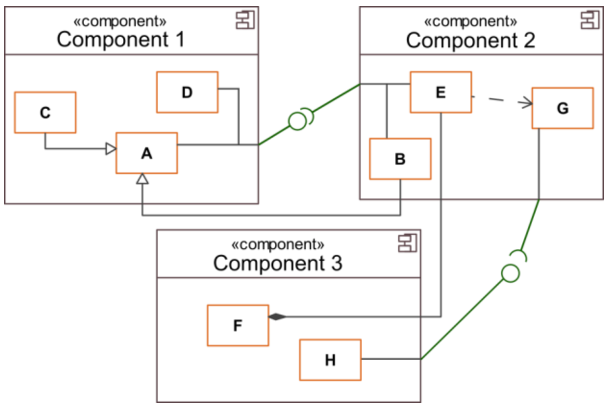
Local search
- Neighbourhood
- Algorithm
- Comparison criterion

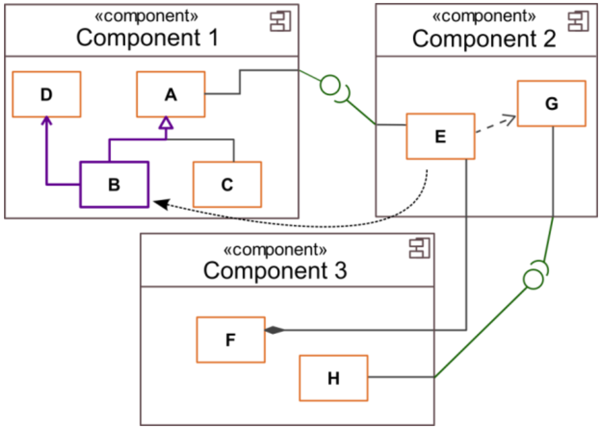
Evolutionary search
- Selection of solutions
- Step of the search
- Number of evaluations

Proposed MOMAs
Local search procedure

- Exploration of the **neighbourhood**:
  - ➢ A random class is reallocated
  - ➢ HC/SA: 1 neighbour/iteration
  - ➢ TS: 5 neighbours/iteration
- **Comparison criterion**:
  - ➢ Dominance (D)
  - ➢ Weights (We)
  - ➢ Best objective (B)
  - ➢ Worst objective (Wo)

Initial solution



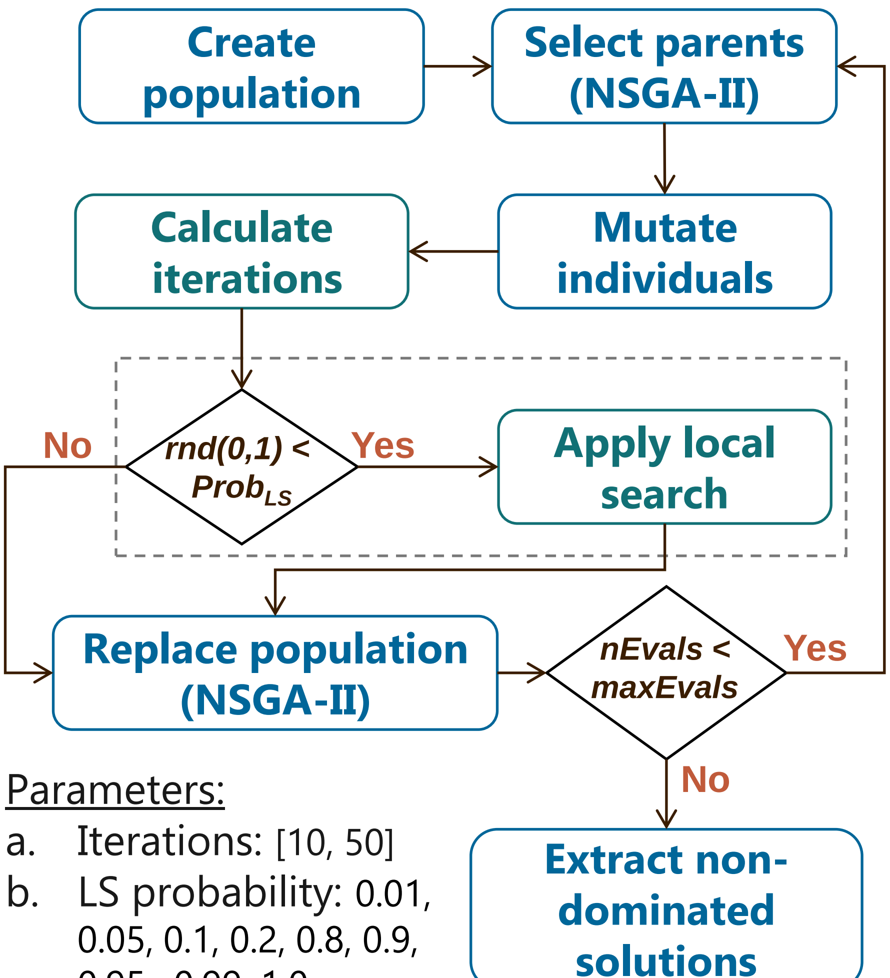


Neighbour solution

Proposed MOMAs

MOEA(LS) and MOEA+LS

MOEA(LS) : LS as a genetic operator

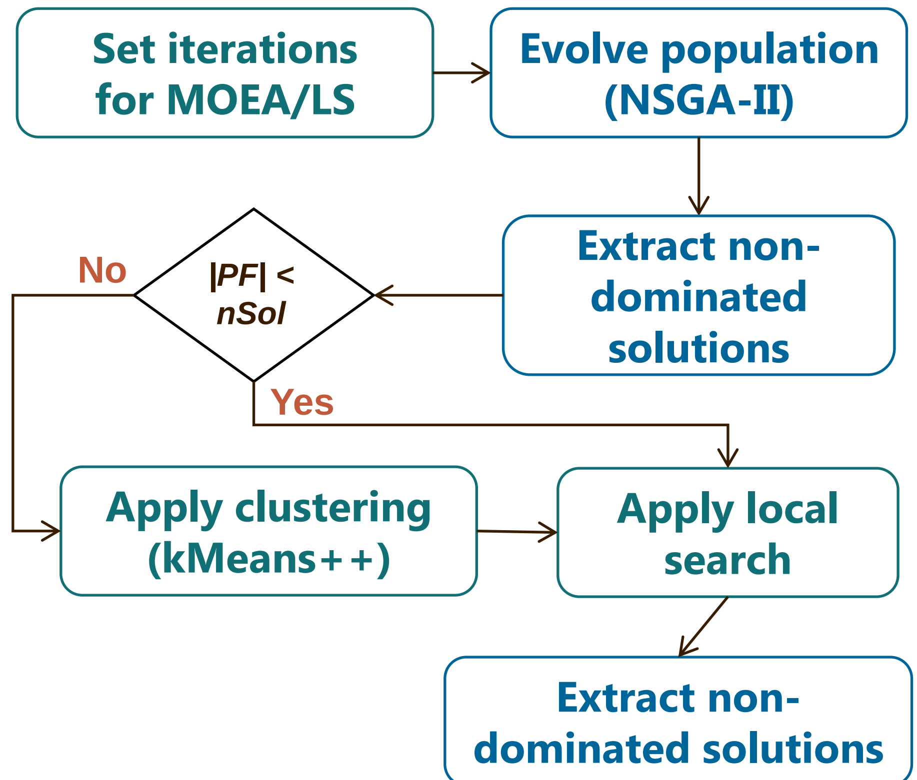


Parameters:

a. Iterations: [10, 50]

b. LS probability: 0.01, 0.05, 0.1, 0.2, 0.8, 0.9, 0.95, 0.99, 1.0

MOEA+LS: LS as post-processing



Parameters:

a. Iterations: 50, 100, 200

b. Solutions (%): 10, 15, 20

Experiments and results
Analysis of quality indicators

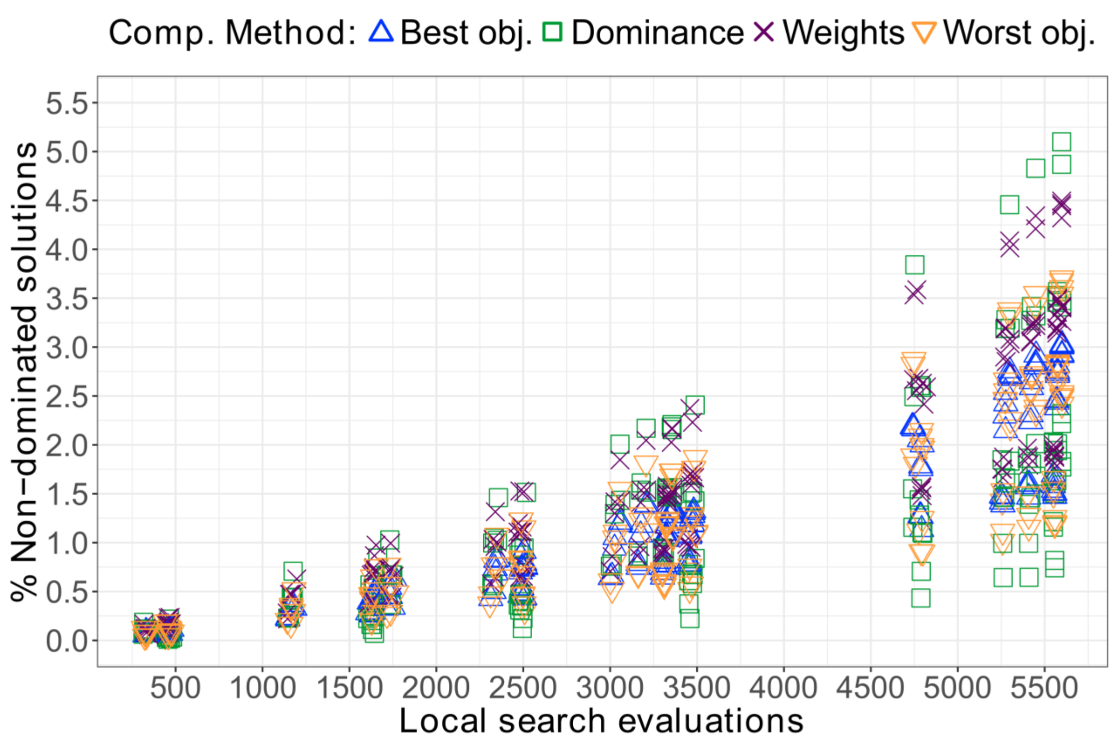
- **Two Set Coverage**
  - TSC = 0 for all comparisons between NSGA-II and MOMA
  - Local search **does not decrease** the efficiency of NSGA-II
- **Spacing**
  - Percentage of **improvement**
    - Few iterations of LS
    - HC/SA vs. TS
    - Different behaviours in MOEA+LS
  - **Effect size** (Cliff's Delta test)
    - Influence of the problem instance
    - MOEA+LS with HC/SA and weights

24 (+1) algorithms:
MOEA(LS) | MOEA+LS
HC | SA | TS
D | We | B | Wo
Common configuration:
150 individuals
24000 evaluations
6 design problems
30 random seeds

		HC	SA	TS
MOEA(LS)	D	[-2.75, 6.66]	[-0.51, 4.02]	[0.07, 4.15]
	We	[0.32, 8.72]	[0.14, 5.74]	[-0.58, 4.80]
	B	[-0.12, 6.49]	[-0.62, 8.53]	[-0.53, 3.55]
	Wo	[-2.76, 6.58]	[-0.56, 5.54]	[-0.13, 3.16]
MOEA+LS	D	[-3.61, 6.68]	[-4.30, 1.95]	[-3.84, -1.79]
	We	[-1.62, 29.47]	[-2.89, 27.42]	[-3.39, 2.11]
	B	[-3.48, 6.32]	[-3.06, 5.50]	[-4.07, -2.04]
	Wo	[-1.77, 14.23]	[-3.92, 11.41]	[-3.55, 0.08]

Experiments and results
Influence on the Pareto Front

- **The number of solutions is similar**
  - Only a small decrease in two problem instances with MOEA+LS
- **Generation of new non-dominated solutions**
  - MOEA(LS):
    - HC > 3%
    - Methods *D* and *We*
  - MOEA+LS:
    - Low percentages
    - Weights are effective



Concluding remarks

- From the **experimental outcomes**:
  - ✓ Local search can enhance the **diversity of solutions**
  - ✓ Influence of the **comparison criterion and algorithm**
  - ✓ Differences among **problem instances**

- **Future work**
  - ➢ **Domain knowledge** to guide the generation of neighbours
  - ➢ Scalability in a **many-objective** space

On the Effect of Local Search in the Multi-objective Evolutionary Discovery of Software Architectures

Thanks!

UNIVERSIDAD DE CÓRDOBA

Aurora Ramírez

Email. aramirez@uco.es
Web. http://www.uco.es/users/aramirez/en