Integration Test of Classes and Aspects with a Multi-Evolutionary and Coupling-Based Approach

Thelma Elita Colanzi, Wesley Klewerton Guez Assunção, Silvia Regina Vergilio, Aurora Trinidad Ramirez Pozo

UFPR- Federal University of Parana, Curitiba, Brazil

Introduction

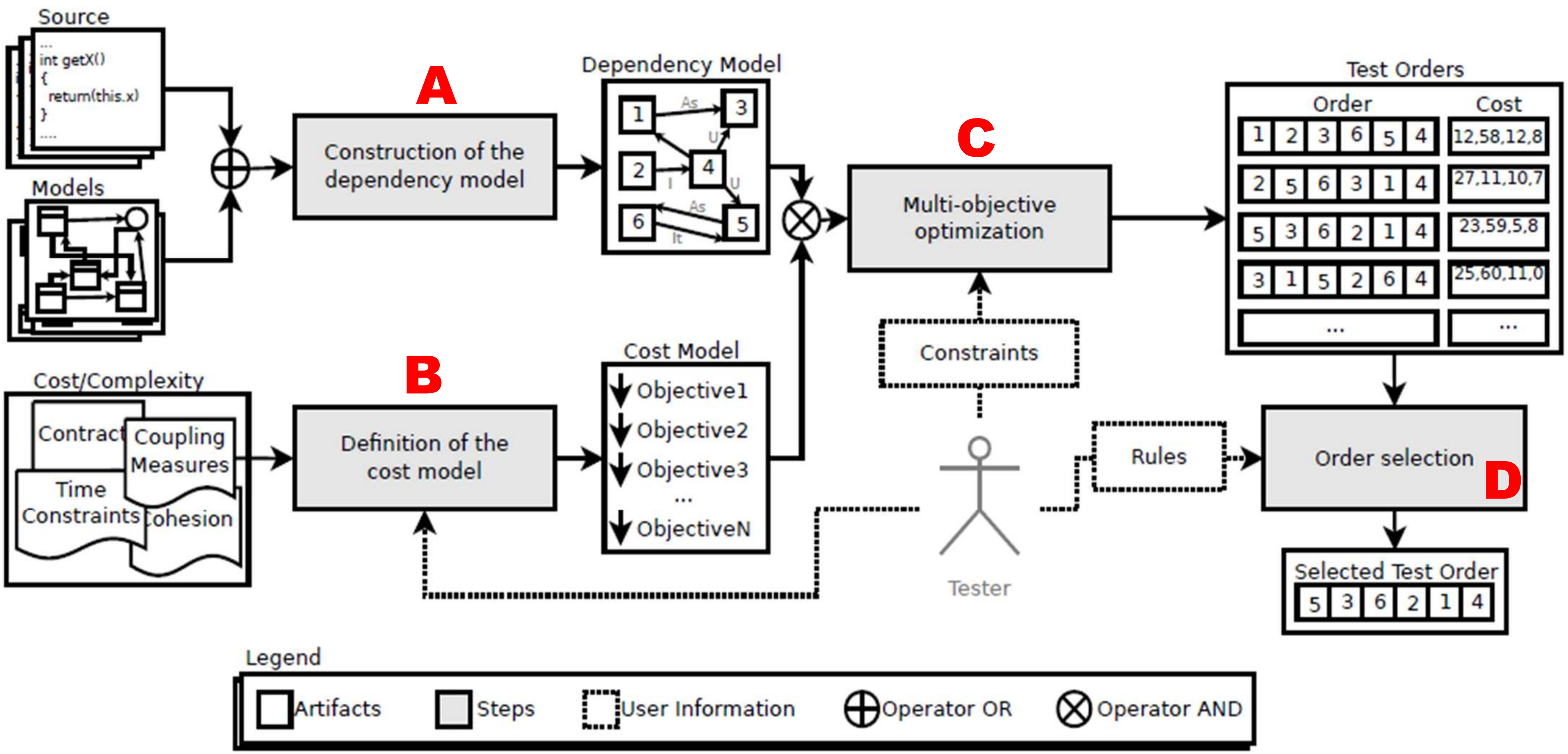
- To determine a sequence for integration and test of classes and aspects that minimizes stubbing efforts
  - CAITO (Class and Aspect Integration and Test Order) problem

- Multi-objective evolutionary algorithms (MOEAs) have achieved better results than approaches based on graphs and genetic algorithms in CAITO/CITO contexts
  - They obtain a set of non-dominated solutions to approximate the Pareto front in a single run of the algorithm

Objectives

- To introduce MECBA (Multi-Evolutionary and Coupling-Based Approach) to solve the integration and test order problem
  - Generic steps for:
    - Definition of both dependency and cost models
    - Optimization through multi-objective algorithms
  - Output: set of solutions to integrate and test modules of a software
- MECBA was instantiated and evaluated in aspect-oriented context, with four AspectJ programs and four coupling measures
- *Do MOEAs deteriorate their performance to the CAITO problem with more than two objectives?*
  - The results of MOEAs were evaluated using four quality indicators and statistical test

MECBA (Multi-Evolutionary and Coupling-Based Approach)



MECBA
A – Construction of the dependency model

- Representation of the dependency relations to be considered
- Different restrictions to some kind of dependency can also be represented
- The dependency model adopted in our evaluation is the extended ORD [24] with the Combined strategy, in which classes and aspects are tested together

[24] Re, R., Lemos, O.A.L., Masiero, P.C.: Minimizing stub creation during integration test of aspect-oriented programs. In: 3rd Workshop on Testing Aspect-Oriented Programs. pp. 1-6. Vancouver, British Columbia, Canada (March 2007)

MECBA
B – Definition of the cost model

- Coupling, cohesion and time constraints can be used
- Objectives to be minimized = **4 coupling measures**
- m_i and m_j are two coupled modules and m_i depends on m_j
- Operation = class methods, aspect methods and aspect advices
- **Coupling measures**:
  - Attribute Coupling (A) = The number of attributes locally declared in m_j when references or pointers to instances of m_j appear in the argument list of some operations in m_i [3]
  - Operation Coupling (O) = The number of operations locally declared in m_j, which are invoked by operations of m_i [3]

[3] L. C. Briand, J. Feng, and Y. Labiche. Using genetic algorithms and coupling measures to devise optimal integration test orders. In 14th SEKE, Ischia, Italy, July 2002.

MECBA
B – Definition of the cost model

- Coupling Measures:
  - Number of distinct return types (R) = Number of distinct return types of the operations locally declared in m_j that are called by operations of m_i [1]
  - Number of distinct parameter types (P) = Number of distinct parameter types of the operations locally declared in m_j that are called by operations of m_i [1]

[1] A. Abdurazik and J. Outt. Coupling-based Class Integration and Test Order. In International Workshop on Automation of Software Test, pages 50-56, Shanghai, China, May 2006. ACM.

MECBA
C - Multi-Objective Optimization

1. **Problem Representation**
   - Permutation of modules which form testing orders

2	4	3	1	5

module 2

number of modules

2. **Fitness Function**
   - 1 matrix with dependencies between modules
   - 4 coupling matrices (one for each coupling measure)
   - Constraints: Inheritance and Inter-types dependencies cannot be broken
   - <u>Fitness of each solution</u>: the sum of dependencies between modules for each coupling measure corresponds to an objective

MECBA
C - Multi-Objective Optimization

3. **Selection of a MOEA**
   - **NSGA-II** (Non-dominated Sorting Genetic Algorithm)
   - **SPEA2** (Strength Pareto Evolutionary Algorithm)

MECBA
D – Order Selection

- The tester selects an order from the Pareto front of non-dominated solutions produced by the algorithms.
- This selection should be based on restrictions and priorities related to the software development, such as test goals, available resources, contractual restrictions, etc.

MECBA Empirical Evaluation

- AspectJ systems

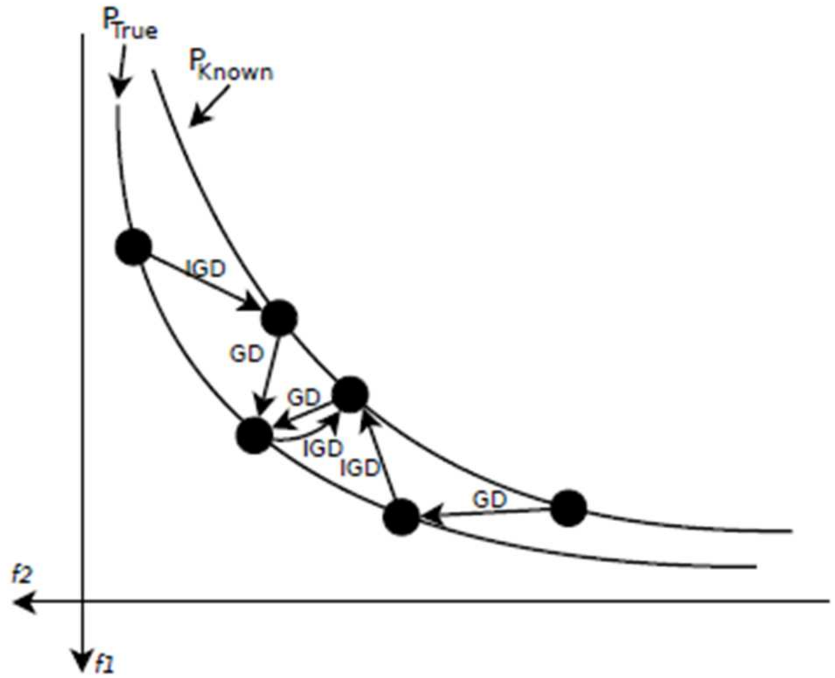
Software	Classes	Aspects	Dependencies	LOC
AJHotDraw	290	31	1592	18586
AJHSQLDB	276	25	1338	68550
Health Watcher	95	22	399	5479
Toll System	53	24	188	2496

- Parameters

- 30 runs

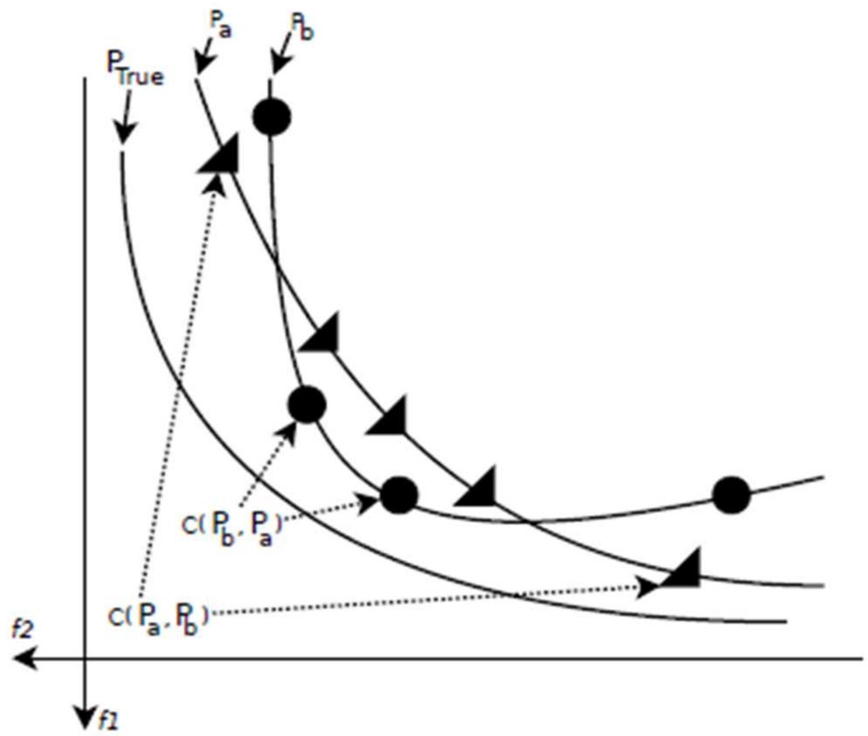
MECBA Empirical Evaluation

- **Quality Indicators**
  - Generational Distance (GD) : calculates the distance from a PFApprox (Pknown) found to the Pareto Front (PFtrue)
  - Inverted Generational Distance (IGD): calculates the distance from PFtrue to a PFApprox found



MECBA Empirical Evaluation

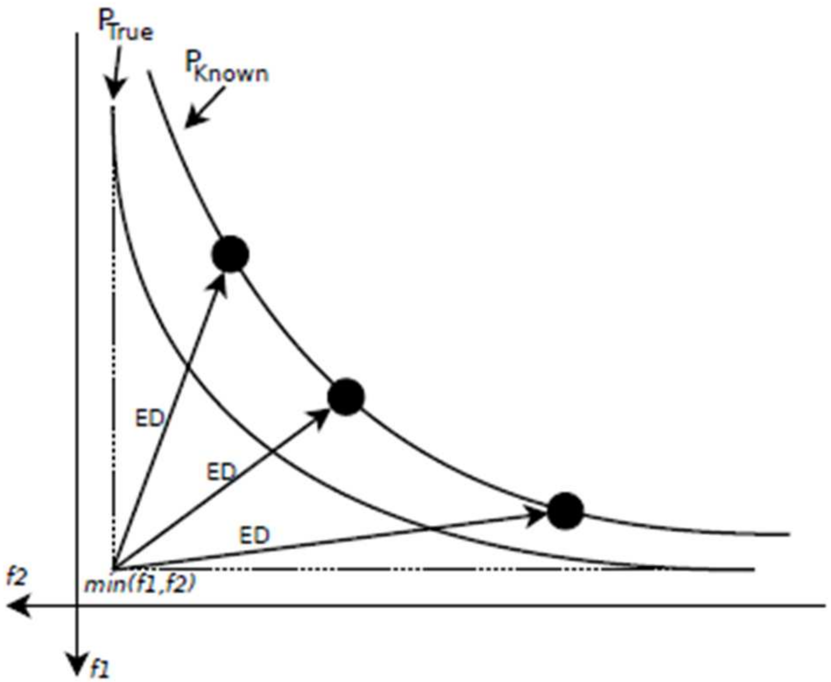
- **Quality Indicators**
  - Coverage (C): measures the dominance between two sets of solutions



- The results of GD, IGD and C were analyzed through Wilcoxon test, in order to verify if NSGA-II and SPEA2 are considered statistically equivalent.

MECBA Empirical Evaluation

- **Quality Indicators**
  - Euclidean Distance from an ideal solution (ED): is used to find the closest solution to the best objectives



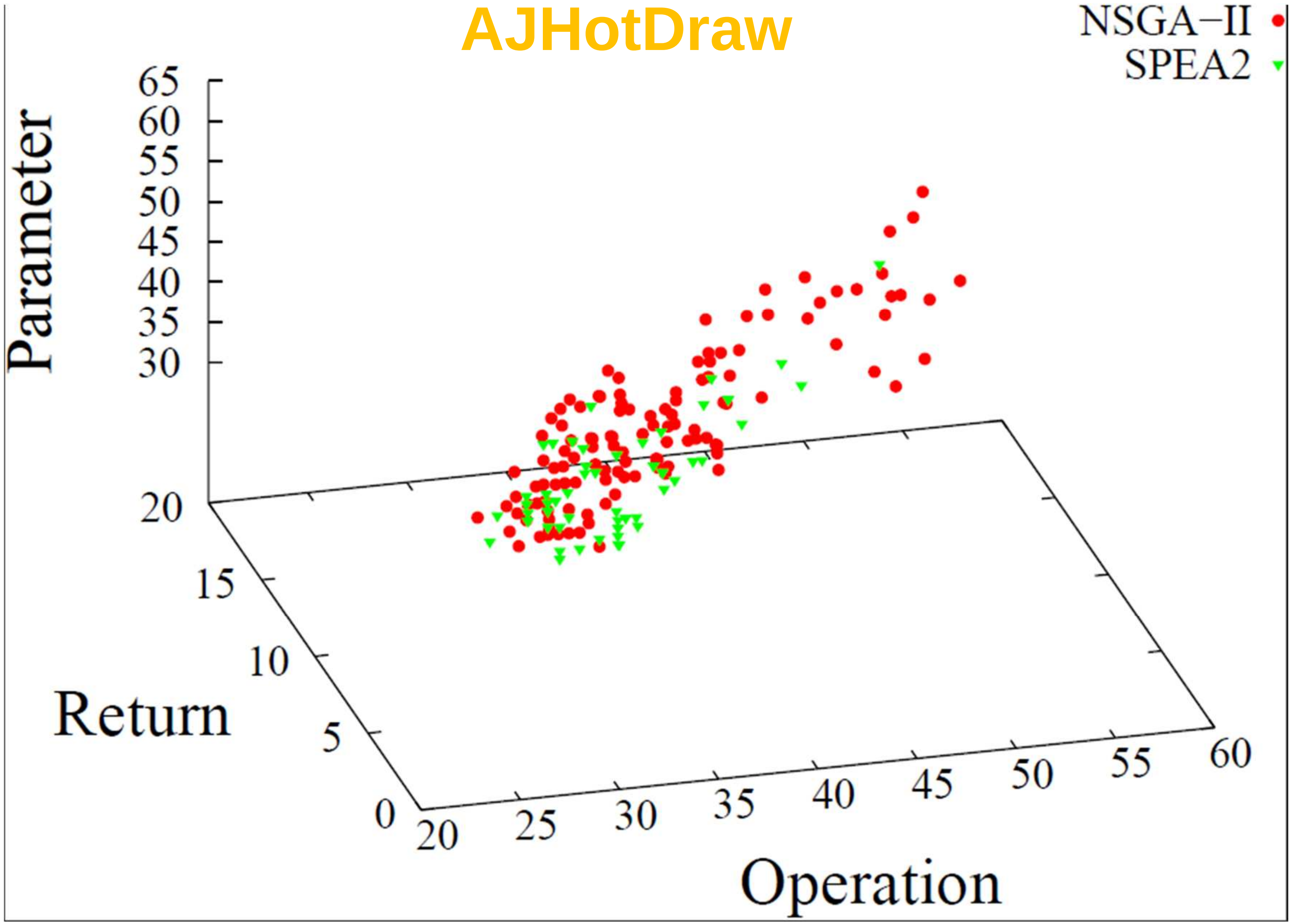
- Some of these quality indicators need the PFtrue, however, in real problems it is not known.
- It is common to use the non-dominated solutions found by all algorithms in all runs.

Results and Analysis

System	Dependencies	PFtrue Cardinality	MOEA	Total of Different Solutions of PFApprox
AJHotDraw	1592	95	**NSGA-II**	**120**
			SPEA2	51
AJHSQLDB	1338	105	**NSGA-II**	**153**
			SPEA2	40
Health Watcher	399	1	NSGA-II	1
			SPEA2	1
TollSystem	188	1	NSGA-II	1
			SPEA2	1

- Health Watcher: (A = 0, O =0, R = 0, P = 0) → 8 cycles
- Toll System: (A = 12, O =2, R = 0, P = 1) → 1 cycle

Results and Analysis



Results and Analysis

- GD and IGD

Indicator	System	Average of NSGA-II	Average of SPEA2
GD	AJHotDraw	**0.0435**	0.0560
	AJHSQLDB	**0.0422**	0.1075
IGD	AJHotDraw	**0.0493**	0.0380
	AJHSQLDB	**0.0357**	0.0641

Wilcoxon test points out that there is statistical difference between them for GD and IGD.

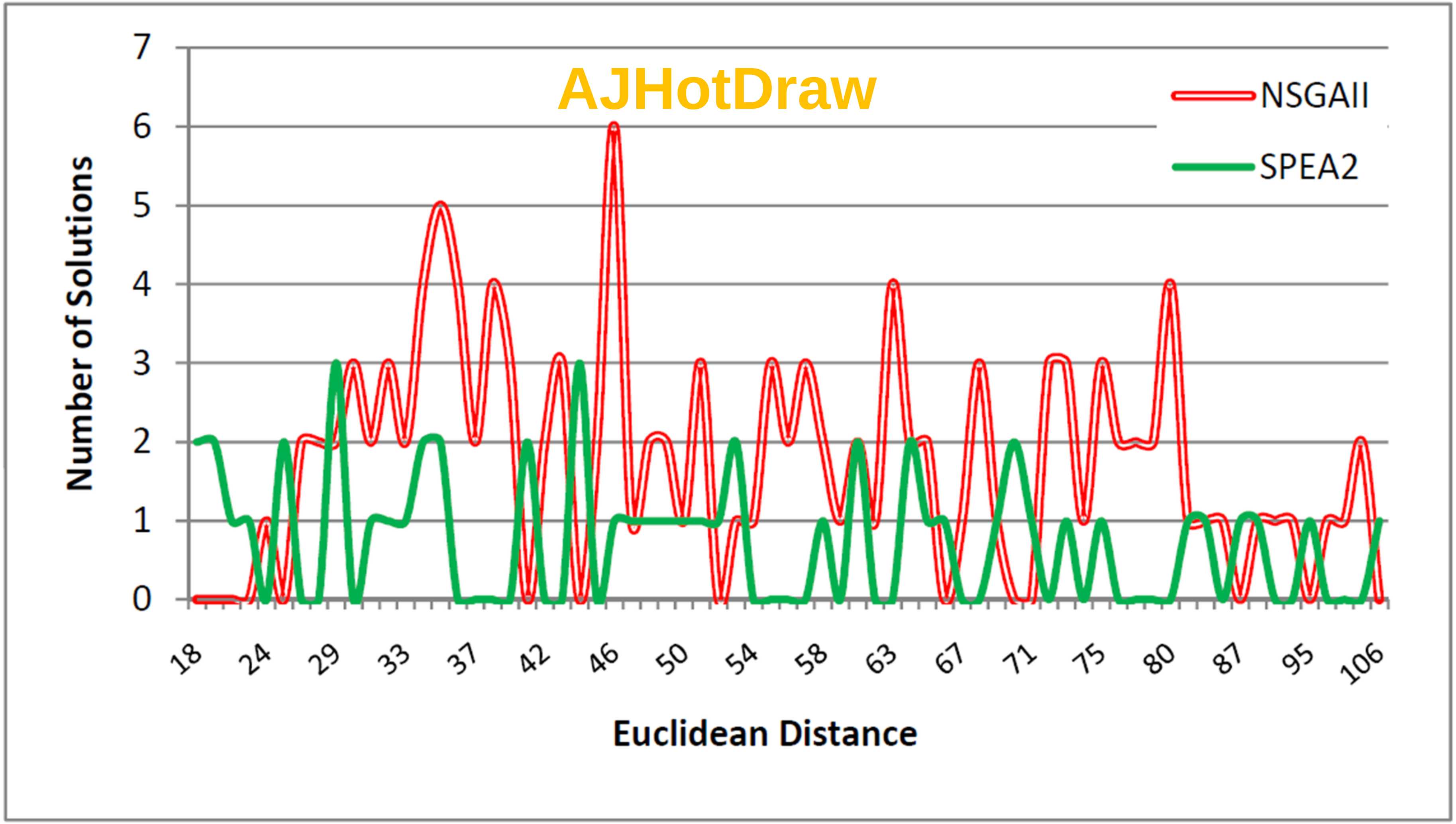
- Coverage: SPEA2 covers NSGA-II for AJHotDraw and AJHSQLDB, although without statistical significancy.

Results and Analysis

- ED: SPEA2 achieved the lowest EDs .

System	Cost of the Ideal Solution	MOEA	Lowest ED	Fitness of the lowest ED
AJHotDraw	80, 24, 0, 31	NSGA-II	24.617	94, 37, 4, 46
		SPEA2	**18.385**	93, 28, 3, 43
AJHSQLDB	1877, 446, 189, 308	NSGA-II	205.842	2008, 569, 273, 363
		SPEA2	**189.365**	1960, 562, 26, 413
Health Watcher	0, 0, 0, 0	NSGA-II	0	0, 0, 0, 0
		SPEA2	0	0, 0, 0, 0
Toll System	12, 2, 0, 1	NSGA-II	0	12, 2, 0, 1
		SPEA2	0	12, 2, 0, 1

Results and Analysis



Discussion about the results

- NSGA-II and SPEA2 achieve feasible solutions despite exploring the solution space in different ways
- NSGA-II has the best distribution of solutions in the search space (great diversity of solutions) → better performance for GD and IGD
- SPEA2 has a good concentration of solutions near to the ideal solution → solutions of lower ED
- These solutions with lower ED cover some NSGA-II solutions improving the coverage rate of SPEA2 on NSGA-II
- Decision makers often prefer solutions near to the ideal solution → SPEA2 should be chosen

Selecting Orders

- *How the tester should select a solution to integrate and test the modules of the system?*

- Costs of solutions achieved by SPEA2 for AJHotDraw

	A	O	R	P	Ranking	Ideal Costs
a	**87**	49	11	52	18º	
b	111	**24**	1	43	14º	
c	102	29	**0**	44	7º	(80, 24,0, 31)
d	184	43	14	**31**	51º	
e	93	28	3	43	1º	

Concluding Remarks

- MECBA was proposed and used for integration and test of classes and aspects
  - The dependency model considers specific characteristics of aspect-oriented programs
  - The cost model considers four coupling measures
  - NSGA-II and SPEA2 were evaluated

- It seems that SPEA2 is more appropriated to generate solutions that are closer to the ideal solution .

- MECBA can be efficiently used to solve the CAITO problem with four objectives
  - MOEAs found a set of different solutions containing different alternatives of compromise among the four objectives.
  - The tester can select the best solution according to the test priorities.

Future Works

- Perform other empirical studies:
  - to use a different strategy for integrate classes and aspects
  - with other aspect-oriented systems, and
  - to evaluate other MOEA and to analyze its behavior in the same context

Questions?

Thelma E. Colanzi
thelmae@inf.ufpr.br

Multi-objective Evolutionary Algorithms (MOEAs)

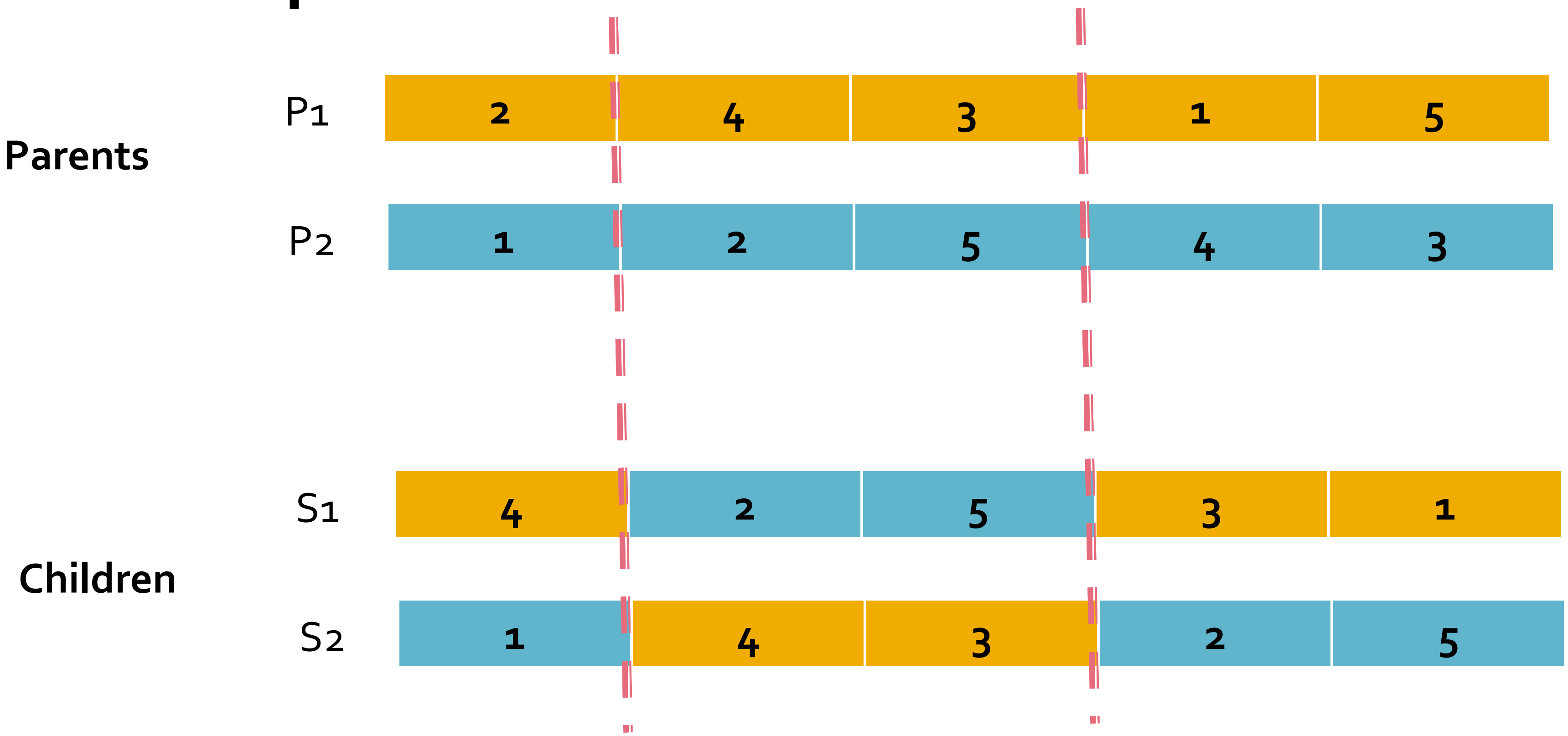
- **NSGA-II** (Non-dominated Sorting Genetic Algorithm)
  - creates several fronts of individuals based on non-dominance relation and discards solutions with lower dominance.
  - crowding distance operator ensures greater spread of solutions.
- **SPEA2** (Strength Pareto Evolutionary Algorithm)
  - an external archive stores non-dominated solutions besides its regular population.
  - each solution has a strength value (fitness), which consists on the number of individuals dominated by this solution.

MECBA Empirical Evaluation

Parameter	NSGA-II	SPEA2
Population Size	300	300
Fitness Evaluation	20000	20000
Mutation Rate	0.02	0.02
Crossover Rate	0.95	0.95
Archive Size	-	250

Crossover operator

- **Two-point crossover**



Matrices

Dependency Matrix

Class	Class | Type of dependency	Class | Type of dependency
1	2 It	3 I
2	4 As	
3	2 As	4 As
4	1 Us	3 As

Coupling Matrix – Measure A

Class	Class | Measure A	Class | Measure A
1	2 2	3 1
2	4 2	
3	2 4	4 2
4	1 2	3 5

Test Order	3	4	2	1	
Costs of Measure A	4	2	-		= 6