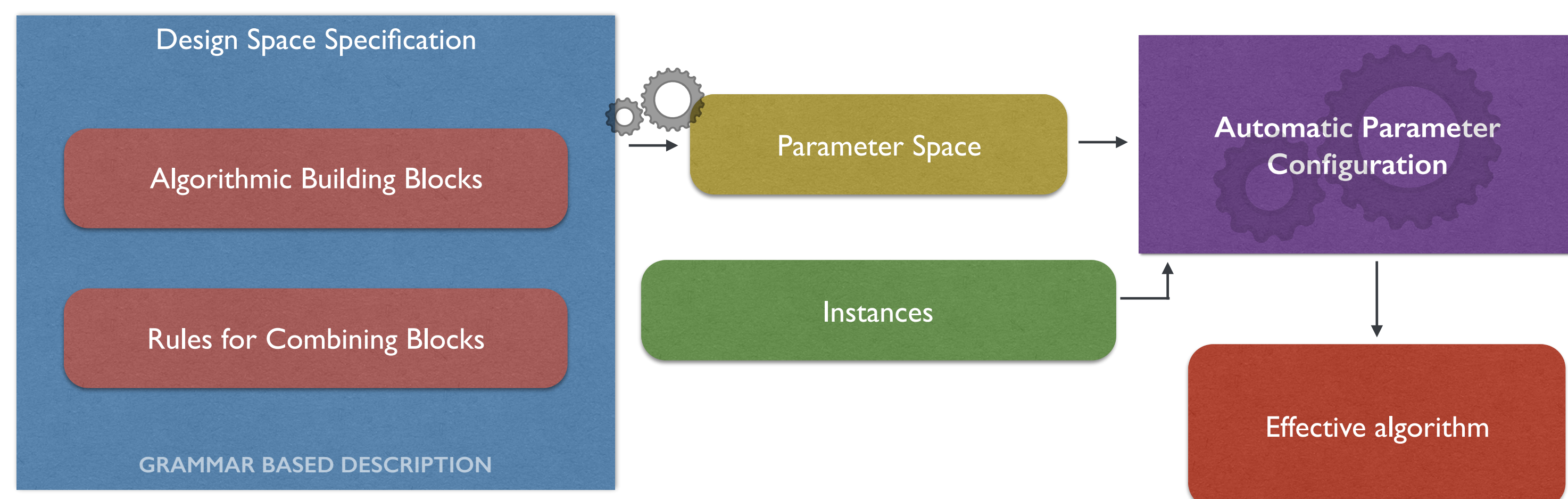


Federico Pagnozzi and Thomas Stützle

Automatic design of Hybrid Stochastic Local Search Algorithms

A configurable metaheuristic framework



Automatic Design

- Design Space Specification
 - Divide the HSLs in components
 - Expose the design choices as parameters
 - The components
 - The rules to combine them
- Use automatic configuration tools to select the best combination

1

Dividing the HSLs in its components

HSLs code outline

```

s0 := initialSolution()
s* := ls(s0)
repeat
  s' := perturbation(s*)
  s" := ls(s')
  s* := acceptanceCriterion(s", s*)
until termination criterion (stop) is satisfied
return s*
    
```

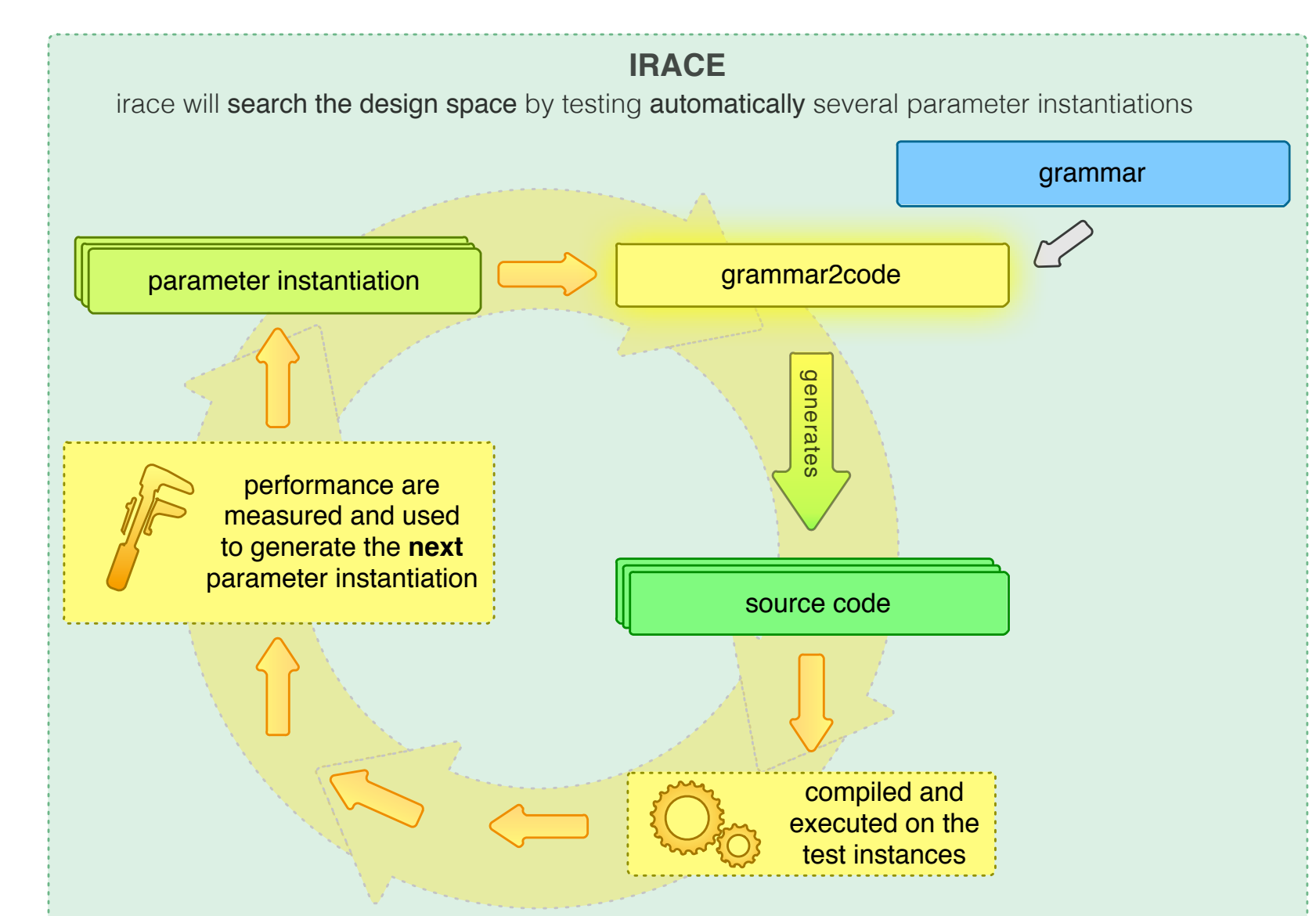
- Problem Independent Components
 - All the components that need only the objective function value to work
 - General structure of the HSLs algorithms
- Problem Dependent Components
 - Neighborhoods
 - Heuristics for the generation of the initial solution
 - Perturbations

Introducing the EMILI framework

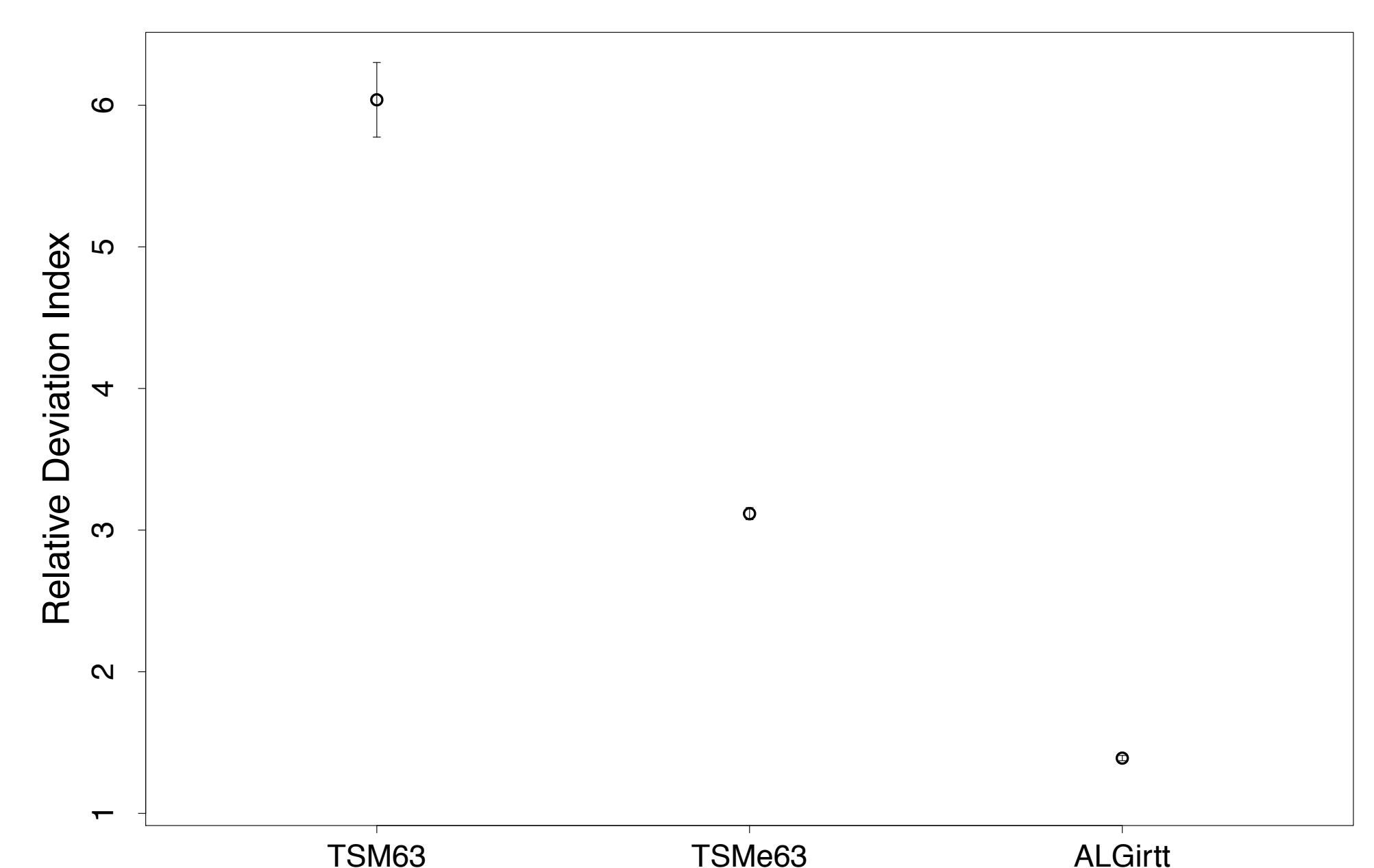
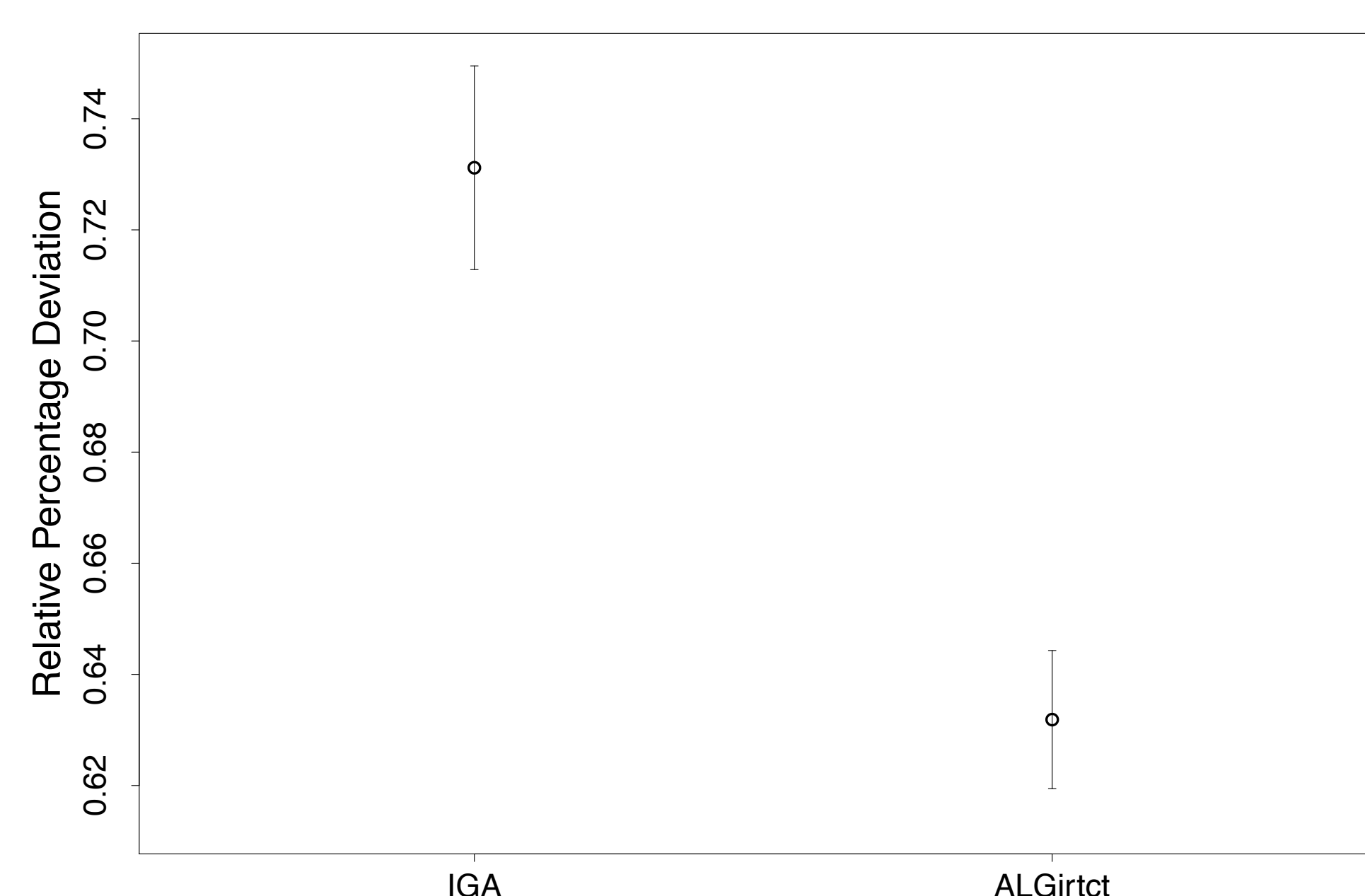
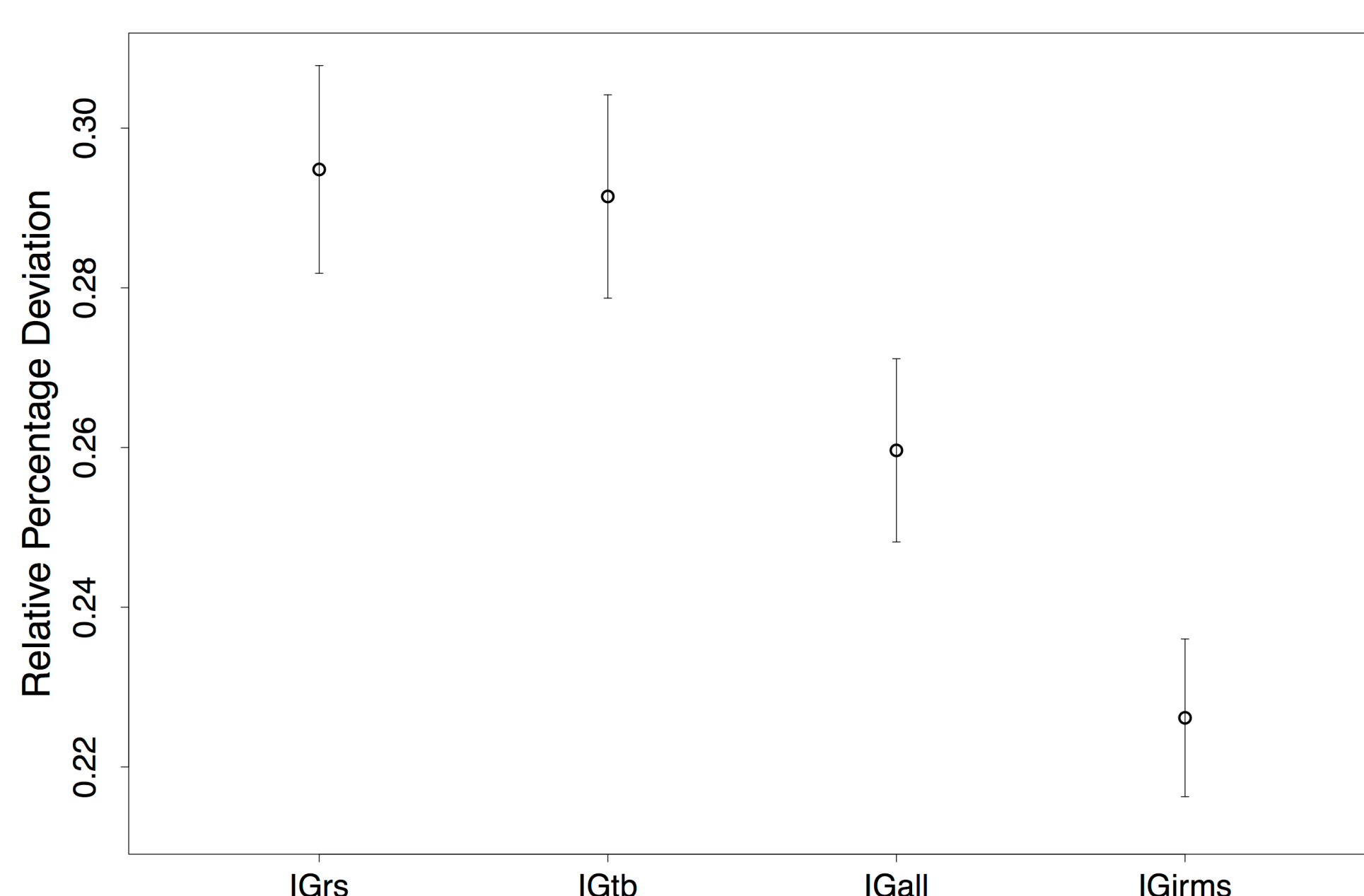
- Easily Modifiable Iterated Local search Implementation
- Designed to be modular and flexible
 - C++
 - Sharing components over several problems
- Already Supports several problems
 - Permutation Flowshop Problem
 - QAP
 - Hybrid Flowshop Problem
 - Inventory Routing Problem
 - Exam Timetabling Problem
 - Vehicle Routing

Describing how to combine the components

- We need to define rules like:
 - An ILS cannot have a tabu tenure
 - A VND has multiple neighborhoods
 - An SA has a perturbation but no LS
- We can represent these rules as the grammar of a simple language



Permutation Flow shop Scheduling Problem



Makespan Taillard benchmark

Total Completion Time Taillard benchmark+

Total Tardiness Vallada benchmark