Revisiting Simulated Annealing: a Component-Based Analysis

Alberto Franzin, Thomas Stützle
IRIDIA, Université Libre de Bruxelles, Bruxelles, Belgium,
{afranzin,stuetzle}@ulb.ac.be

Input: initial solution s_0, neighbourhood, instance
Output: best solution found during search

best sol $s^* =$ incumbent $s = s_0$
$T_0 =$ initial temperature
while stopping criterion is not met do:
  choose solution s' from neighbourhood of s
  according to search space exploration criterion
  if s' meets acceptance criterion then
    $s = s'$
  if s' improves over s^* then
    $s^* = s'$
  update temperature according to cooling rate, temperature length
  and temperature restart scheme
return s^*

Simulated Annealing can be divided into seven basic components. It takes in input two additional problem-specific components.

In the literature we can find thousands of works about/using SA; in many of them we find several Ideas/variants/adaptations that we can collect Into algorithmic frameworks and reuse.

Initial temperature (8 options)
- Fixed value
- Proportional to init.sol. cost
- maximum gap in random walk
- average gap in random walk
- initial probability
- Connolly
- Misevicius
- Simplified Misevicius

Stopping criterion (9 options)
- max time
- max # moves
- minimum temperature
- max # cooling steps
- max # temperature restarts
- max # moves with no accepted solutions
- global acceptance rate
- most recent acceptance rate
- no new best solution recently

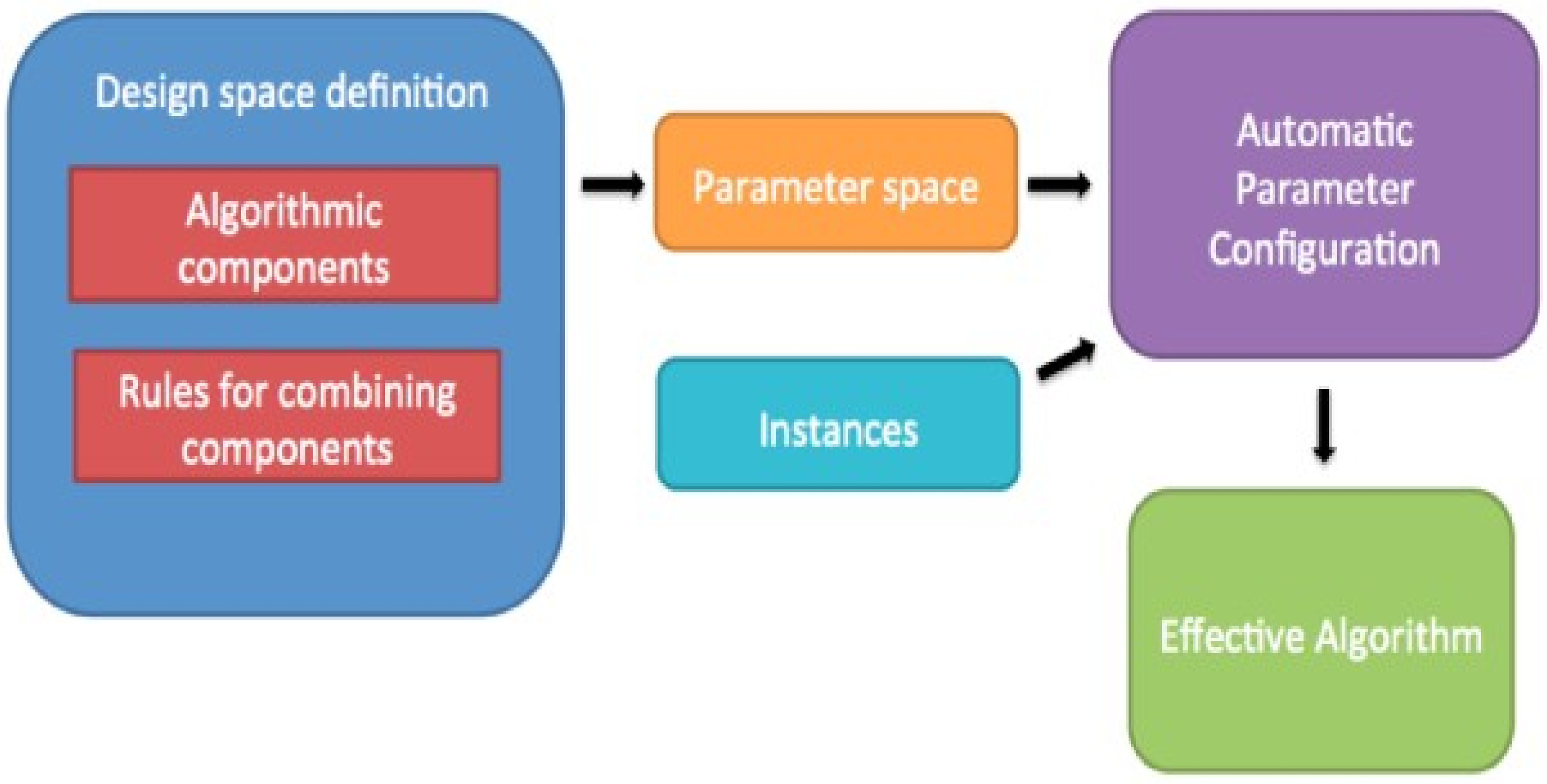
Temperature restart (16 options in total, between restart and reheating)
- never
- # moves
- minimum temperature
- % of initial value
- # cooling steps
- global acceptance rate
- acceptance rate last k moves
- no recently accepted moves

Exploration criterion (4 options)
- random - sequential - Ishibuchi-Misaki-Tanaka 1 - Ishibuchi-Misaki-Tanaka 2

Acceptance criterion (9 options)
- Metropolis
- Bounded Metropolis
- Precomputed Metropolis
- Generalized SA
- Geometric
- Threshold Acceptance
- GDA
- RTR
- LAHC
- HC

Cooling scheme (11 options)
- Geometric 1 ; 2
- Logarithmic 1 ; 2
- Lundy-Mees ; variant
- Q8-7
- Quadratic
- Artithmetic
- Constant temperature
- Temperature band

Temperature length (9 options)
- Fixed # moves
- # modev proportional to problem size
- # moves proportional to size of neighbourhood
- # accepted moved
- bounded (# accepted moves, max # moves)
- arithmetic increase
- geometric increase
- logarithmic increase
- exponential increase



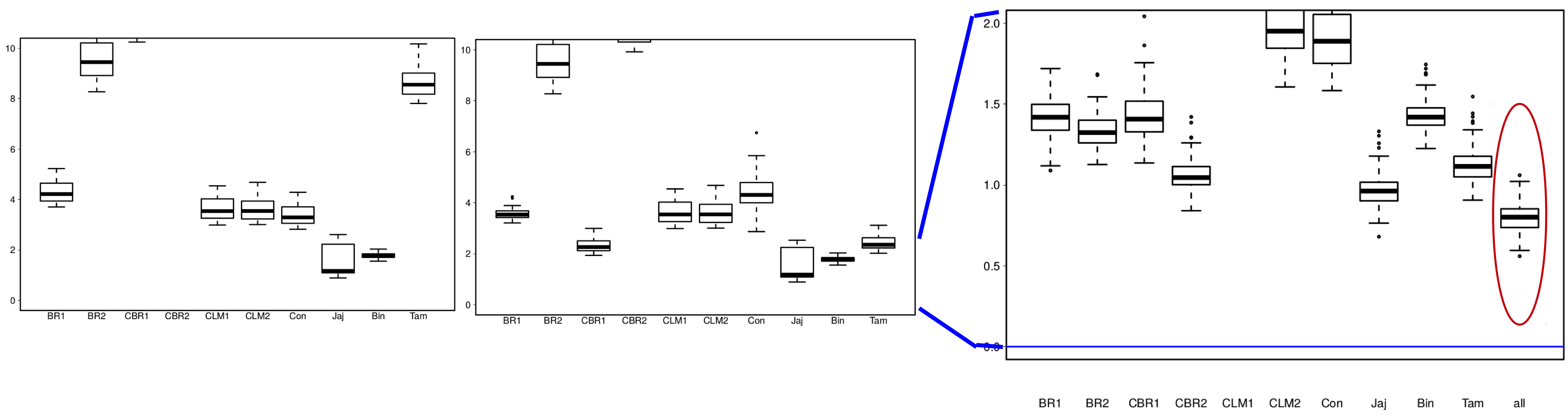
Automatic algorithm configuration approach can be naturally expanded into automatic algorithm design.

Starting from the componenets we have implemented In the framework, we can use AAD to:
- improve existing algorithms
- generate new, more powerful algorithms
- study the algorithms in a more scientific way

We use irace as configurator and EMILI as framework.
We constrain ourselves to SA by fixing the rules to follow the structure of SA (top-down algorithm generation).

We instantiate ten SA algorithms for the QAP from the literature, tune their numerical parameters and generate new SAs.
We report the results obtained on 150 random instances, with default settings, with 10s of runtime, and after the tuning and the generation of new algorithms.

Exp. Setup: 2k budget for tuning the numerical parameters (3-5 for each algoritm), 60k for generation (97 parameters in total), 10s runtime, 15 tunings each exp.



We can perform some analysis. Numerical parameters can differ from the original works.
We do observe the impact of some components/choices after the tuning.
Acceptance, Neighbourhood exploration are the most important components.

We can also improve the anytime behaviour of the algorithms.