TRAIL TRUSTED AI LABS
BY DIGITALWALLONIA / SPW-RECHERCHE

Algorithmic-specific features to understand stochastic local search algorithms

Alberto Franzin

Laboratory of Image Synthesis and Analysis

LISA, ULB

Supported by Service public de Wallonie – Recherche under grant n°2010235 « ARIAC BY DIGITALWALLONIA4.AI »

Context

Metaheuristics and Stochastic Local Search (SLS) algorithms are routinely used in many practical applications.

But: how do they actually work?

Context

Metaheuristics and Stochastic Local Search (SLS) algorithms are routinely used in many practical applications.

But: how do they actually work?

Several study try to understand how a SLS work for some problem.

One possible approach: connect the results obtained by an algorithm on a particular problem by analyzing some **features**.

What kind of features?

In combinatorial optimization, problem-specific features are usually considered. Main issue: they cannot be used to compare findings across different problems.

But SLS algoriths work for different problems.

Landscape features have been also considered. Main issue: they are global features, that measure the state of the problem across all the solution space.

But SLS algorithms use only local information.

Algorithmic-specific features

We investigate the use of features that are problem-independent, and can therefore be used to compare the results of a SLS algorithm across different problems.

In particular, if we focus on one family of methods, we can use features that measure the conditions encountered by the algorithm during its search.

Case study: simulated annealing

Input: a problem instance Π, a **neighbourhood** $\mathcal{N}$ for the solutions, an **initial solution** s_0, control parameters
Output: the best solution s^* found during the search

```
best solution s* = incumbent solution ŝ = s_0;
i = 1;
t_1 = initialize temperature;
while stopping criterion is not met do
    choose a solution s_{i+1} in the neighbourhood of ŝ according
    to search space exploration criterion;
    if s_{i+1} meets acceptance criterion then
        ŝ = s_{i+1};
    end
    if ŝ improves over s* then
        s* = ŝ;
    end
    if temperature length is reached then
        update temperature according to cooling scheme;
    end
    if temperature has to be reset then
        reset temperature according to temperature restart
        scheme;
    end
    i = i + 1;
end
return s*;
```

Old and popular SLS algorithm, with many variants. We compare:

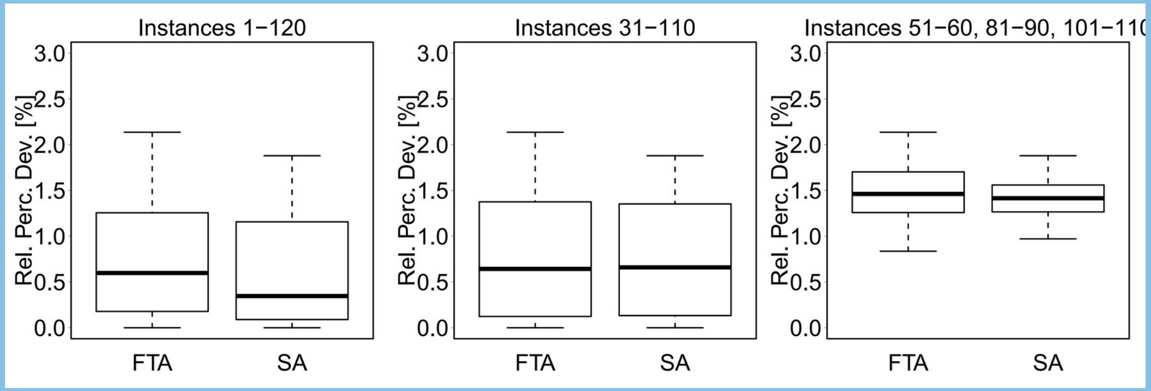
1) traditional SA with geometric cooling scheme (SA)

2) fixed temperature variant (FTA)

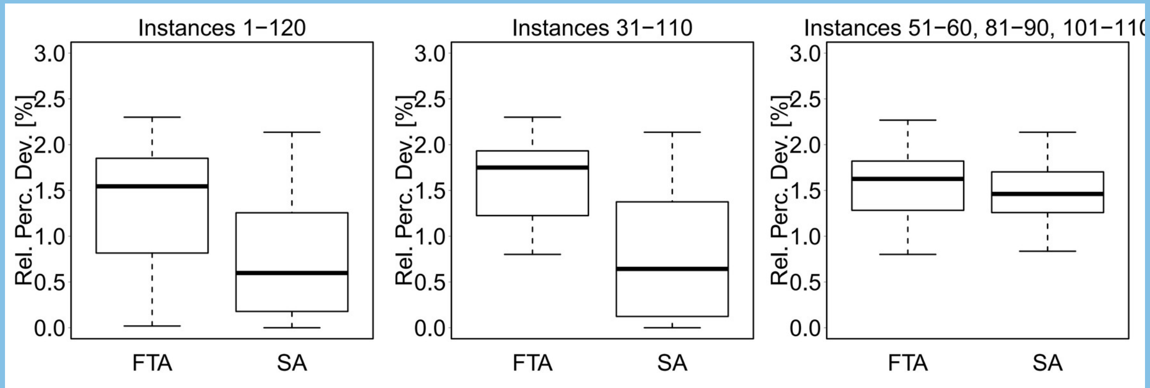
We use AutoAI methods to generate “optimal” SAs and FTAs for three problems and different instance classes.

Results (the lower the boxplots, the better)

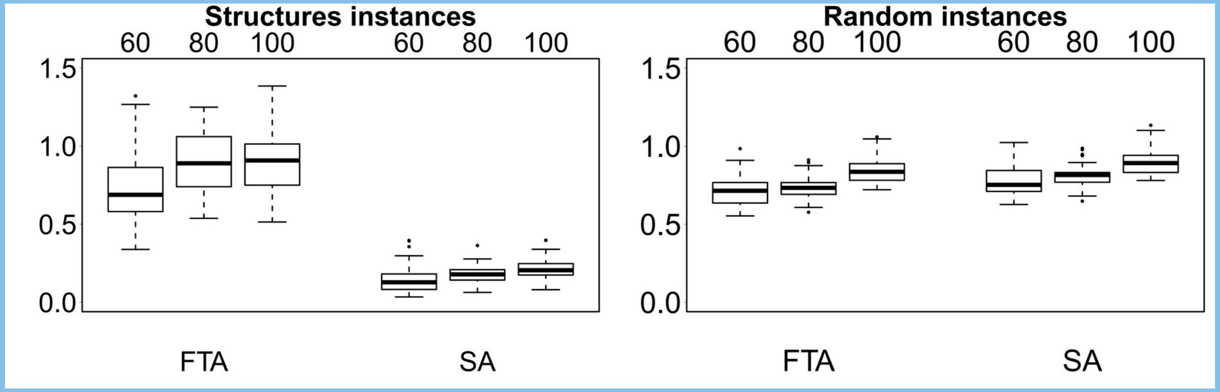
PFSP-MS



PFSP-TCT



QAP



Which features for SA/FTA?

We look for features that represent what the algorithm sees during the search. But the space of possible algorithms is huge (# parameters).

SA can be considered a hill climbing with a diversification mechanism, so we use a First- and a Best-Improvement to approximate what a SA/FTA sees during the search.

Which features for SA/FTA?

1. Average number of moves to local optimum
2. Standard Deviation of number of moves to local optimum
3. Average number of moves to local optimum, rescaled by instance size
4. Standard Deviation of number of moves to local optimum, rescaled by instance size
5. Average number of moves to local optimum, rescaled by neighbourhood size
6. Standard Deviation of number of moves to local optimum, rescaled by neighbourhood size
7. Average R^2 of a linear model fit on the solution values
8. Average R^2 of a linear model fit on the solution values normalized in $[0, 1]$
9. Average R^2 of an exponential model fit on the solution values
10. Average R^2 of an exponential model fit on the solution values normalized in $[0, 1]$
11. Average R^2 of a linear model fit on the number of improving moves in the sequence of neighbourhoods traversed
12. Average proportion of neutral moves in the neighbourhoods traversed
13. Number of neutral last moves
14. Difference of features 7 and 9
15. Difference of features 8 and 10
16. Average slope of the sequence of differences between best and average solution in a neighbourhood
17. Standard deviation of the slopes of the sequence of differences between best and average solution in a neighbourhood
18. Average of the standard deviation of the sequence of differences between best and average solution in a neighbourhood
19. Standard deviation of the standard deviation of the sequence of differences between best and average solution in a neighbourhood

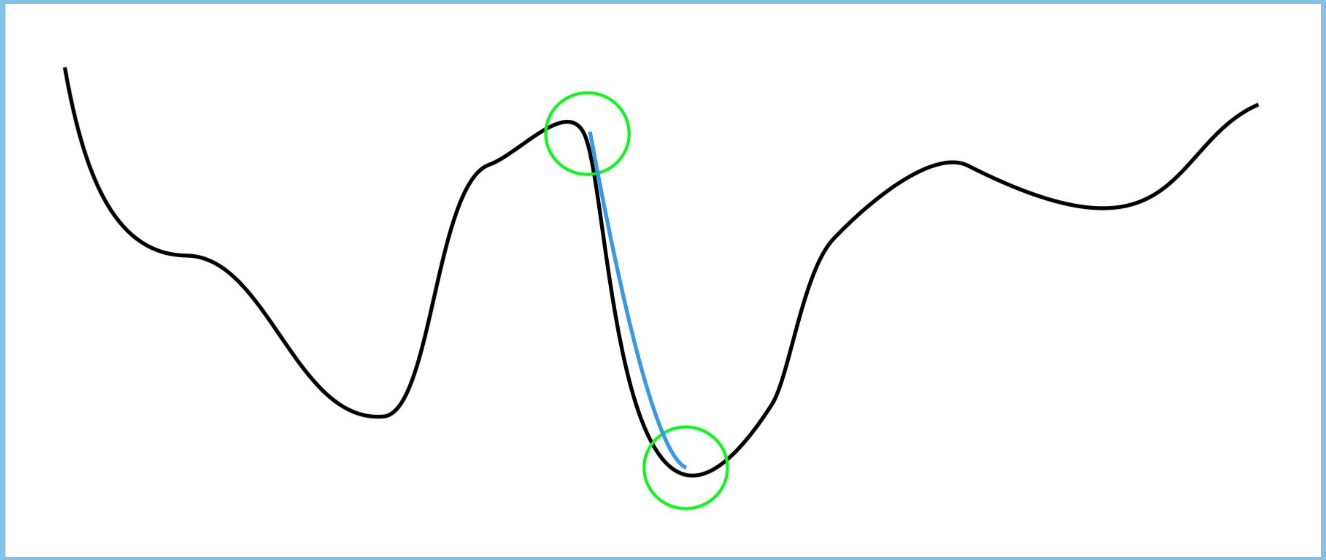
Features importance analysis

We compute the features with 30 independent runs for each instance, analyze their importance.

Results consistent across problems: the most important features measure the change in the landscape observed:

- error when fitting a linear or exponential model with solution values

- sequence of differences between best and average solutions in the neighbourhoods traversed



Conclusions: SA vs FTA

Observations consistent across problems:

If the neighbourhood structures traversed by the algorithm are similar across the search space, FT works well.

SA is better if the landscape changes shape as the search progresses.

Conclusions: algorithmic features

Problem-independent, algorithmic-specific features that capture the inteeraction between a SLS algorithm and a solution space enable a further step in the analysis and the understanding of the behaviour of optimization algorithms.

Work from A. Franzin, T. Stützle "A Landscape-based Analysis of Fixed Temperature and Simulated Annealing", under review. https://iridia.ulb.ac.be/IridiaTrSeries/link/IridiaTr2021-005.pdf