

Parallel Tempering Simulated Annealing

Implementation for asymmetric travelling
salesman problem

Łukasz Grabarski,
Magdalena Jeczeń,
Karolina Mączka,
Mateusz Nizwantowski,
Marta Szuwarska

TABLE OF CONTENTS

O1

ATSP

What does this
problem entail?

O2

ALGORITHM

How does it
work?

O3

MODIFICATIONS

How have we
improved it?

O4

PARAMETERS

What variables
influence it?

O5

RESULTS

How good are our
solutions?

O6

CYTHON

Has Cython
helped?

O7

LONG-TERM RESULTS

Has longer execution
time helped?

O8

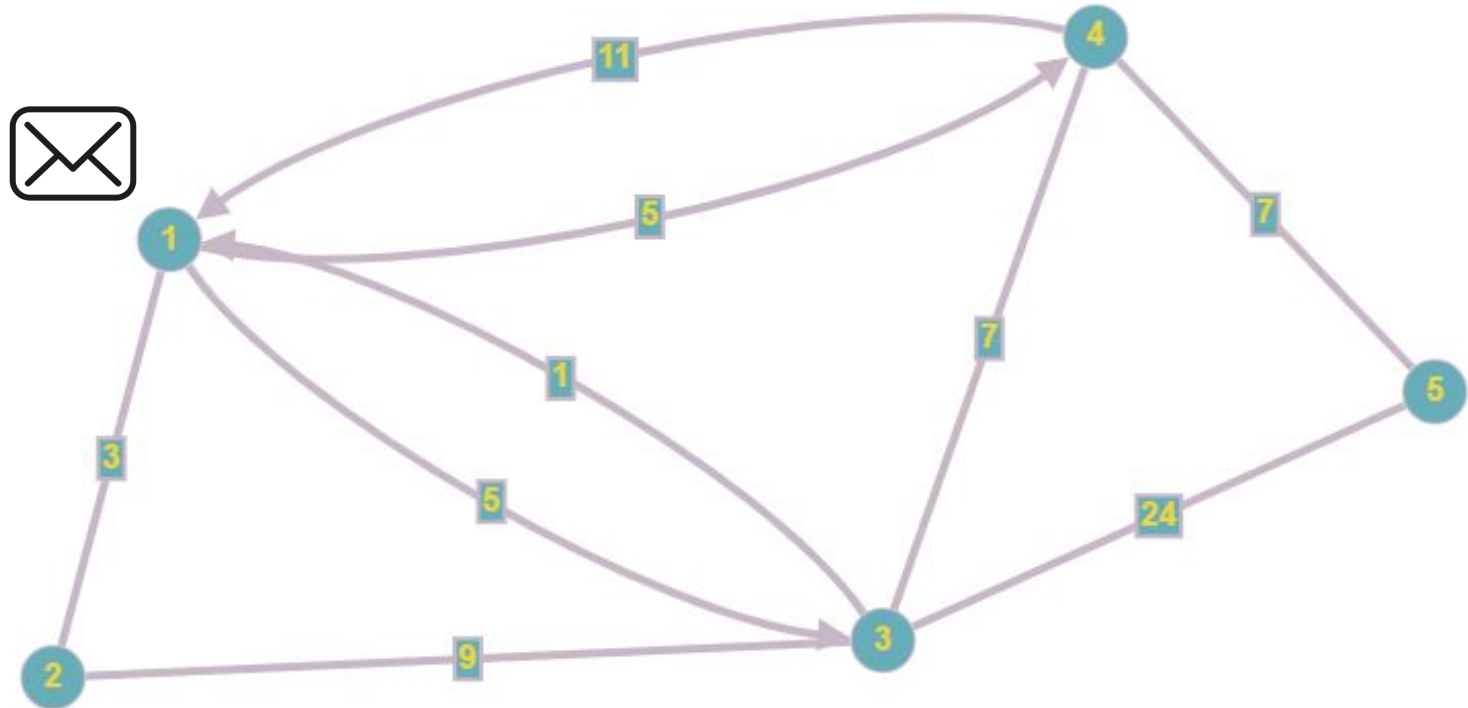
SUMMARY

Was it all worth
it?



01 ASYMMETRIC TRAVELLING SALESMAN PROBLEM

Minimal Hamiltonian cycle = ?

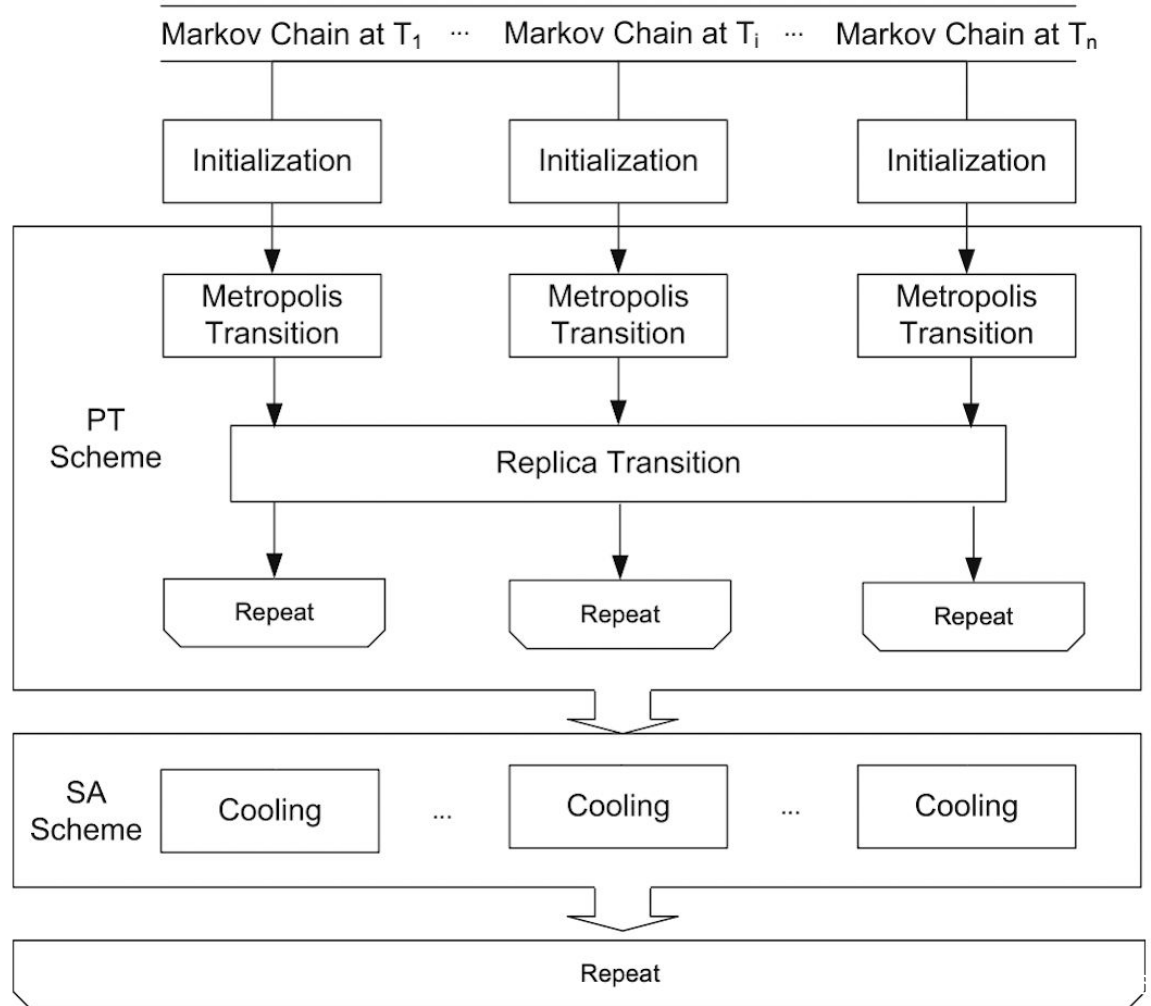




O2

ALGORITHM

Parallel Tempering Simulated Annealing



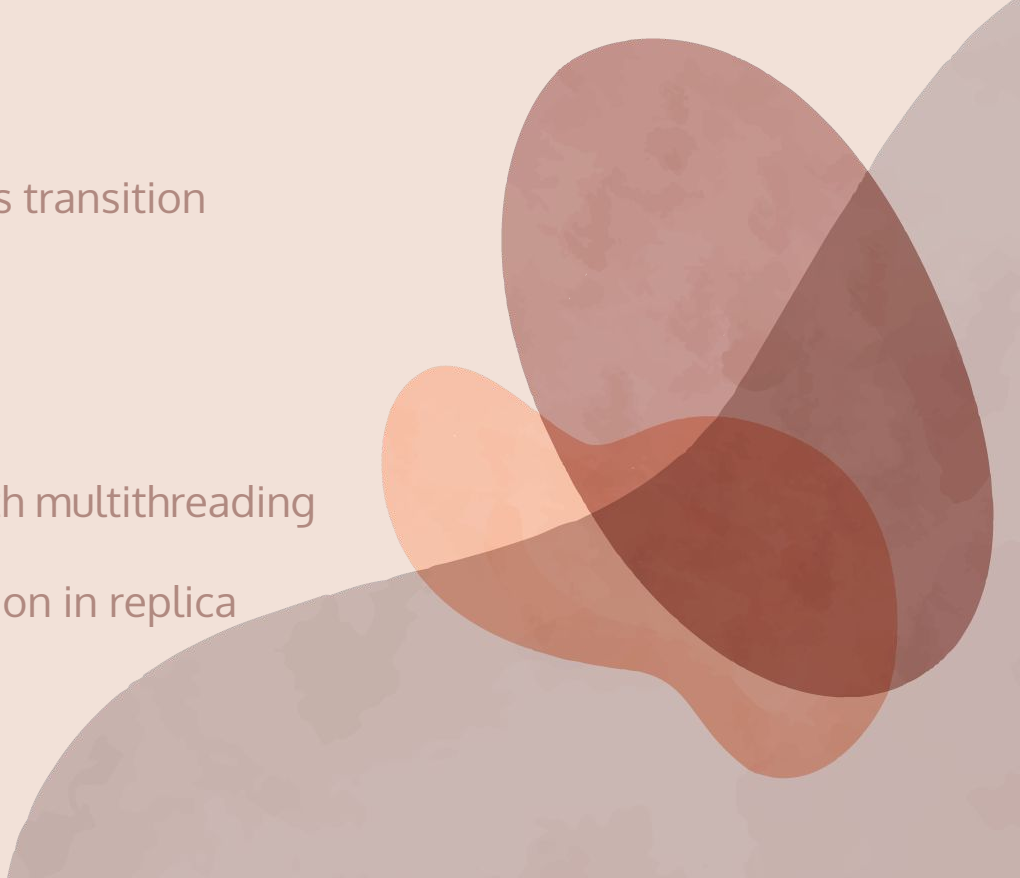


O3

MODIFICATIONS

Modifications

- Starting with heuristic solution
- Implementing two types of metropolis transition
 - Shuffle transition
 - Swap transition
- Speeding metropolis transition up with multithreading
- Skipping solutions close to best solution in replica transition
- Restarting algorithm





O4

PARAMETERS



10

Different parameters

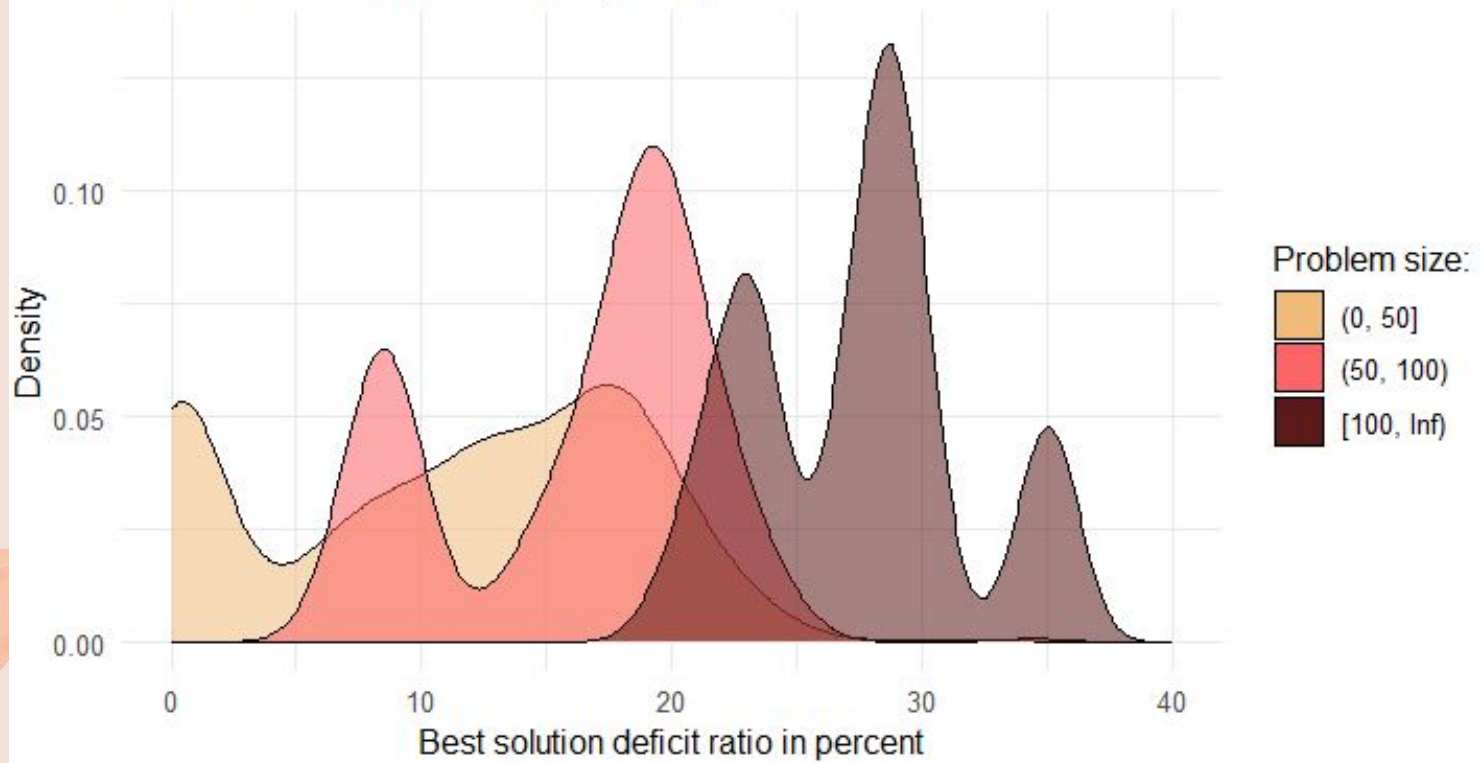
Over 1600

Algorithm invokes for parameters testing

Stability

Parameters have minor impact on results

Best solution deficit ratio in percent





O5

RESULTS



~10%

Mean best solution deficit ratio

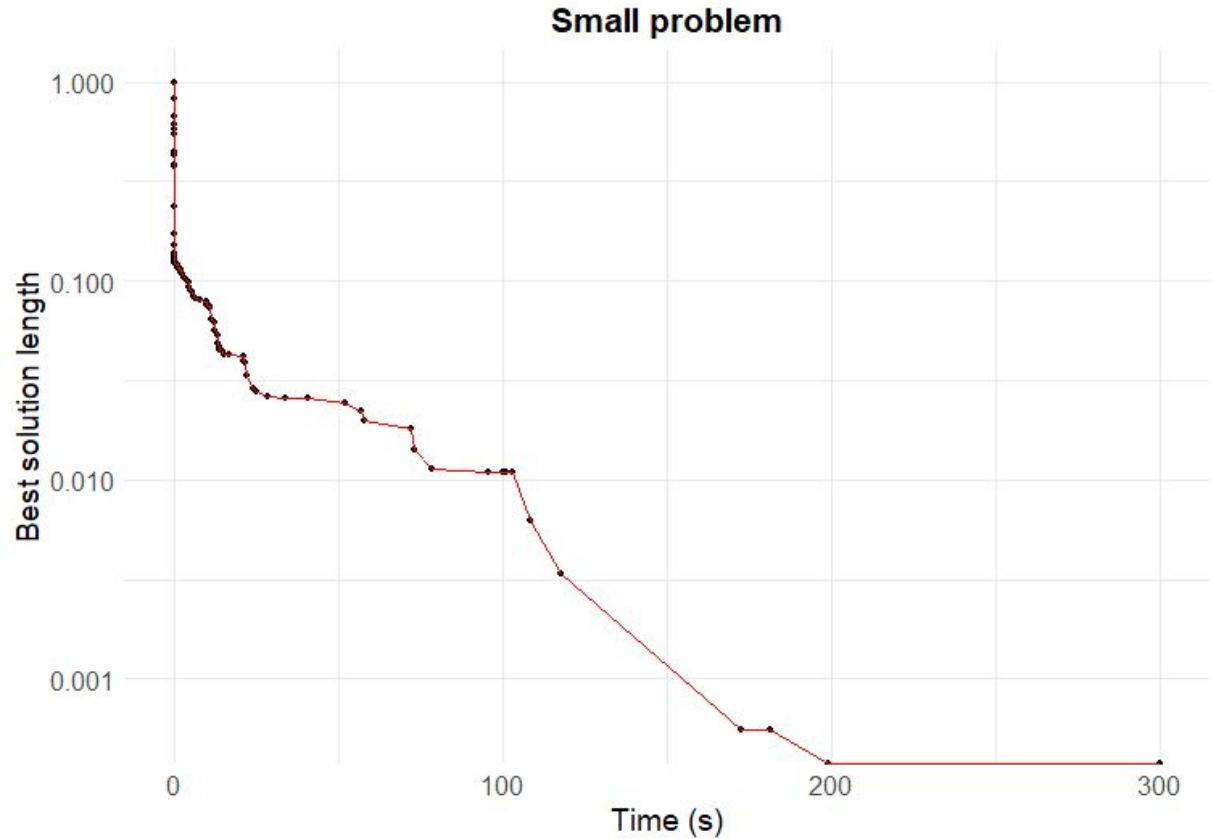
0%

Minimum best solution deficit ratio

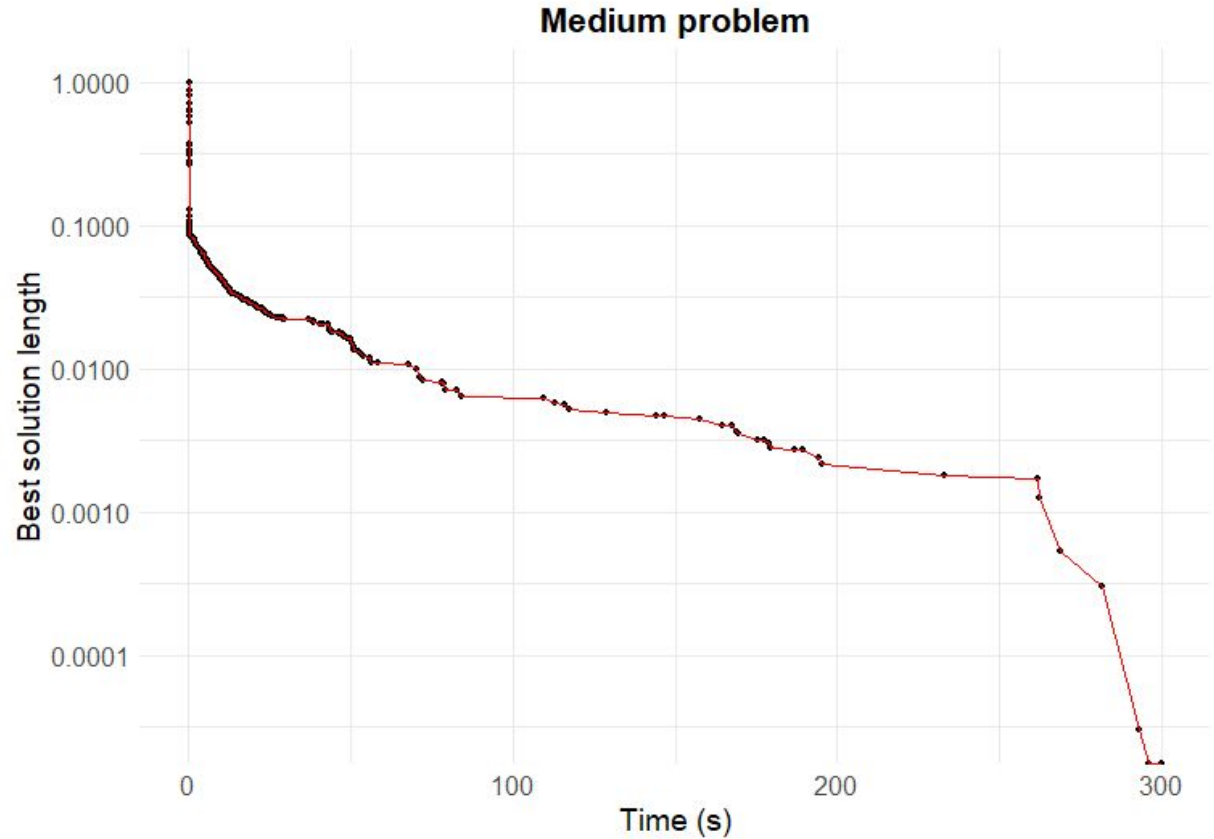
~24%

Maximum best solution deficit ratio

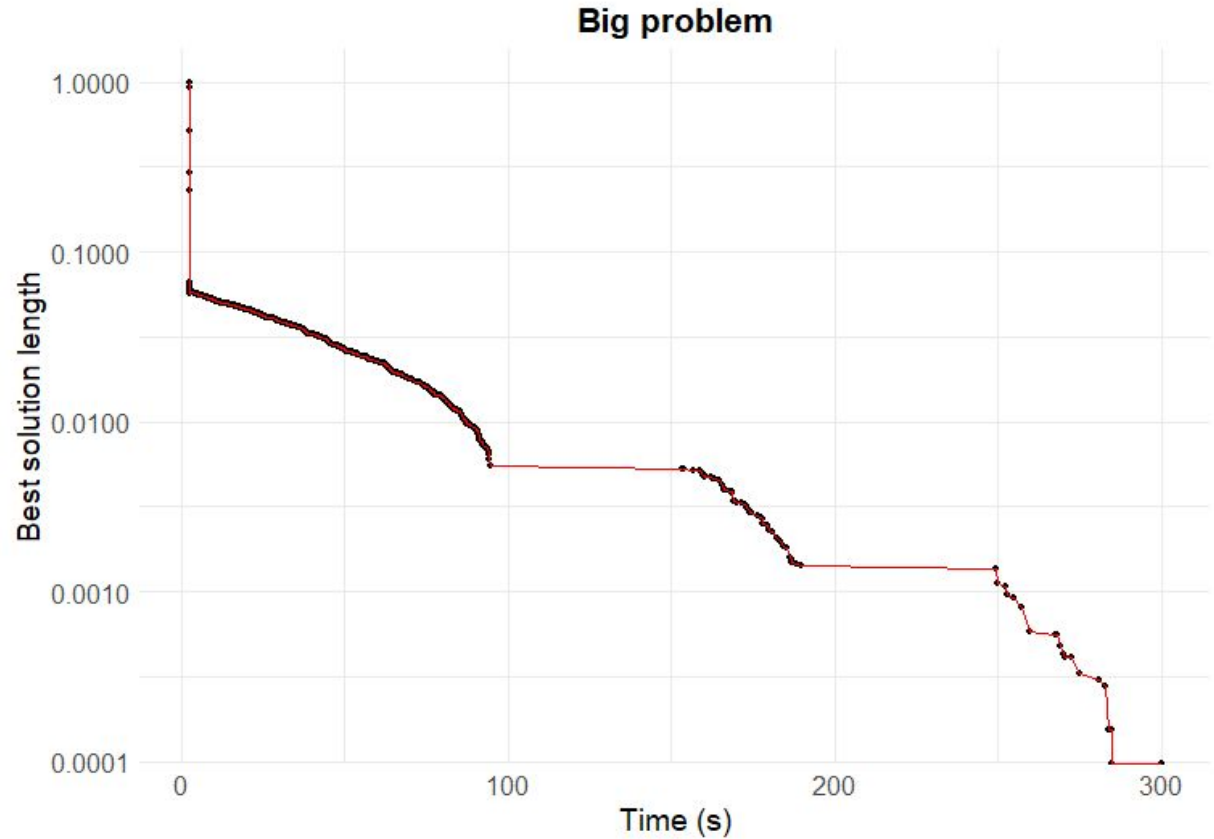
Best
solution
length
depending
on time
- SMALL
PROBLEM



Best
solution
length
depending
on time
- MEDIUM
PROBLEM



Best
solution
length
depending
on time
- BIG
PROBLEM



Our best
solution
lengths
compared with
best known
solution
lengths

Name	Best known solution length	Our best known solution length	Best solution deficit ratio (in percent)
br17	39	39	0.0000000
ftv33	1286	1382	7.4650078
ftv35	1473	1534	4.1412084
ftv38	1530	1614	5.4901961
p43	5620	5627	0.1245552
ftv44	1613	1713	6.1996280
ftv47	1776	1931	8.7274775
ry48p	14422	15105	4.7358203
ft53	6905	8019	16.1332368
ftv55	1608	1782	10.8208955
ftv64	1839	1976	7.4497009
ft70	38673	40597	4.9750472
ftv70	1950	2174	11.4871795
kro124p	36230	40276	11.1675407
ftv170	2755	3379	22.6497278
rbg323	1326	1527	15.1583710
rbg358	1163	1441	23.9036973
rbg403	2465	2742	11.2373225
rbg443	2720	3067	12.7573529



O6

CYTHON



~11%

Mean best solution deficit ratio

0%


Minimum best solution deficit ratio

~31%

Maximum best solution deficit ratio

Comparison of best solution deficit ratio between Python and Cython solutions





O7

LONG-TERM RESULTS



~9%

Mean best solution deficit ratio

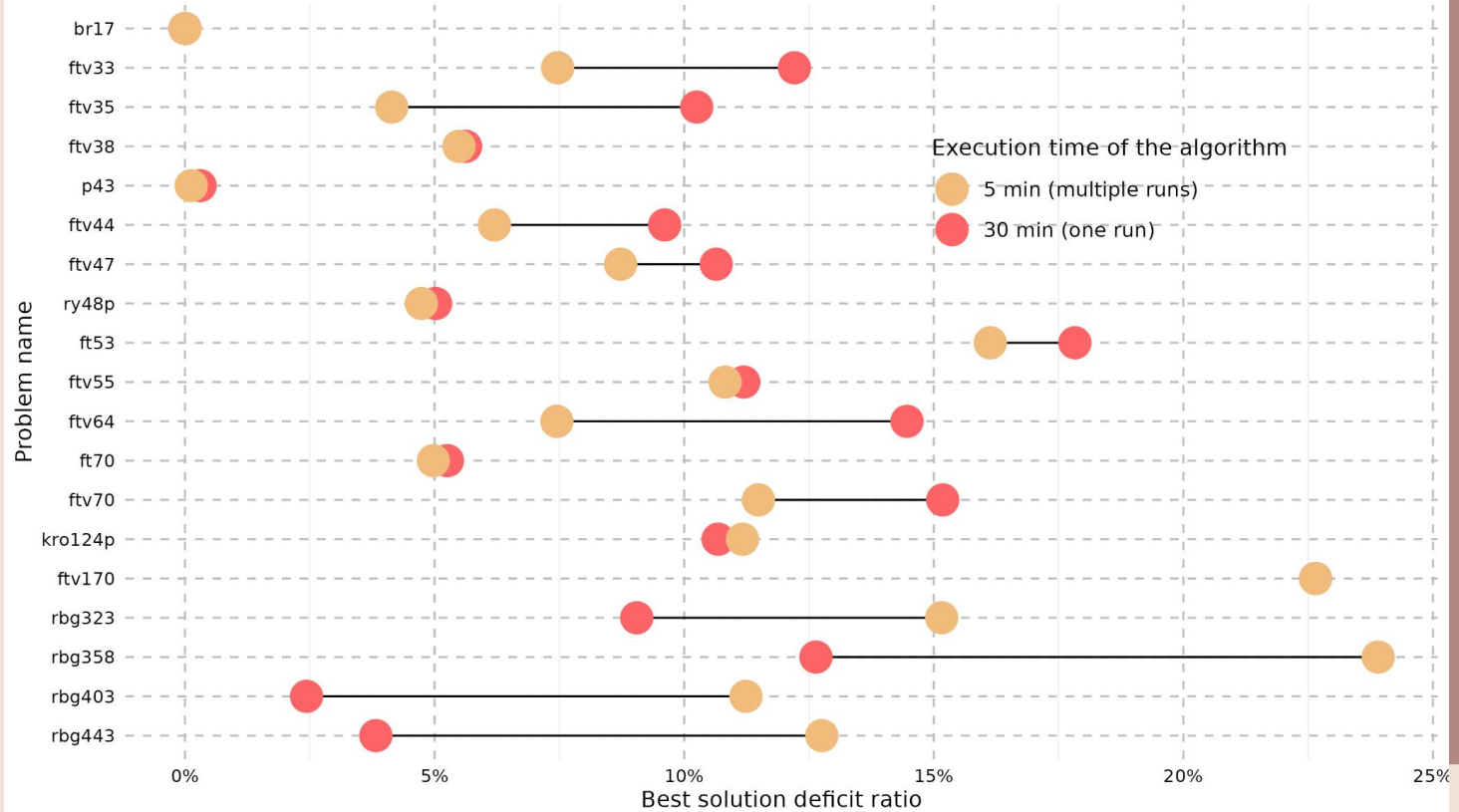
0%

Minimum best solution deficit ratio

~23%

Maximum best solution deficit ratio

Comparison of best solution deficit ratio after 5 and 30 minutes





O8

SUMMARY

Possibilities for further development

- Another method for generating heuristic solutions
- Testing parameters more thoroughly
- Rewriting the algorithm in a faster programming language
- Introduction of new parameters
- Tinkering with a different cooling functions





THANK YOU

Bibliography

https://www.researchgate.net/publication/229004865_Multithreaded_Simulated_Annealing

<https://www.peterbaumgartner.com/blog/intro-to-just-enough-cython-to-be-useful/>

<https://github.com/PrzeChoj/2023Lato-WarsztatyBadawcze>

<https://chat.openai.com/>

<https://www.scirp.org/journal/paperinformation.aspx?paperid=109634>

<https://www.jetbrains.com/help/pycharm/cython.html>

<https://www.scirp.org/journal/paperinformation.aspx?paperid=109634>