Unsupervised Machine Learning for the Quadratic Assignment Problem

Éric Taillard, T.-V. Luong

University of Applied Sciences of Western Switzerland, HEIG-VD

eric(point)taillard(arrobase)heig-vd.ch

July 6, 2022

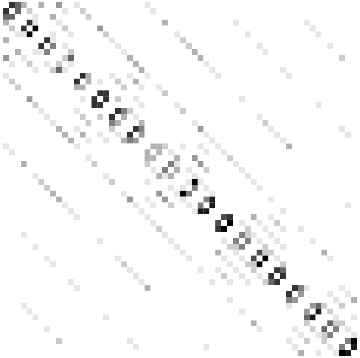
Table of contents

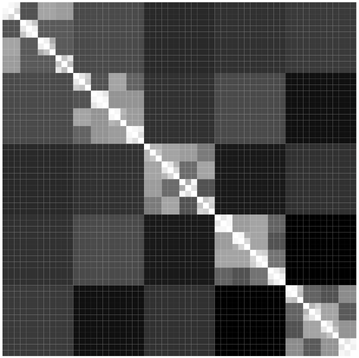
The Quadratic Assignment Problem

First description: [Koopmans and Beckmann(1957)], Monograph: [Çela(1998)]

Definition

- **Data**: Matrices $A = (a_{ij})$ (flows) and $B = (b_{ij})$ (distances), $n \times n$
- **Solution:** A permutation π of $\{1, 2, \ldots, n\}$ (position) of the n items
- **Objective:** $\min_\pi z(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)}$





“Matrix dot product”

Algorithms for the QAP

- Exact algorithms: $n < 35, 49$
- Constructive algorithms (heuristics)
  - Greedy algorithms not efficient: placing the first item costs nothing; the cost of placing the last item can be arbitrarily large
  - Learning good positions for each item (Ant Systems)
  - Goal of this work: can we learn good positions with locally optimum solutions?
- Local searches
  - Taboo searches (Robust, Reactive)
  - Late Acceptance Hill Climbing
  - With Ant Systems: FANT
  - With Population Management: GA, Path Relinking

Machine Learning : Possible approaches

- Ideally: $2n^2$ inputs $\rightarrow$ n outputs (a good permutation)
  - Far from supervised ML capabilities (even for simpler problems like TSP)
- Unsupervised learning
  - Clustering, k-means
  - Neural networks (various topologies, self-organizing (Kohonen) maps)
  - Principal component analysis
  - Frequent features (itemset) appearing in local optima

Extraction and Combination of Frequent Itemsets

Suppose we have 4 local optimum solutions

position	a	b	c	d	e	f	g	$\dots$
	10	8	7	14	1	13	21	$\dots$
item	2	8	7	5	1	17	11	$\dots$
	10	11	7	13	1	14	24	$\dots$
	9	8	4	7	1	14	24	$\dots$

Supported itemsets (frequency $>$ 25% for this portion):

1-itemset		2-itemset			3-itemset				4-itemset
$1 - e$	(100%)	$1 - e$	$7 - c$	(75%)	$1 - e$	$7 - c$	$8 - b$	(50%)	none
$7 - c$	(75%)	$1 - e$	$8 - b$	(75%)	$1 - e$	$7 - c$	$10 - a$	(50%)	
$8 - b$	(75%)	$1 - e$	$10 - a$	(50%)	$1 - e$	$14 - f$	$24 - g$	(50%)	
$10 - a$	(50%)	$1 - e$	$14 - f$	(50%)					
$14 - f$	(50%)	$1 - e$	$24 - g$	(50%)					
$24 - g$	(50%)	$7 - c$	$8 - b$	(50%)					
		$7 - c$	$10 - a$	(50%)					
		$14 - f$	$24 - g$	(50%)					

Apriori algorithm for the extraction of frequent itemsets

Data: $solutions$, min_sup and $itemsets\_limit$

Result: $all\_itemsets$

$k \leftarrow 1$

$C_k \leftarrow generate\_itemsets(solutions, \emptyset, \emptyset)$

$L_k \leftarrow filter\_itemsets(C_k, min\_sup, \emptyset)$

$all\_itemsets \leftarrow L_k$

while $L_k \neq \emptyset$ **do**

  $C_{k+1} \leftarrow generate\_itemsets(solutions, L_k, L_1)$

  $L_{k+1} \leftarrow filter\_itemsets(C_{k+1}, min\_sup, itemsets\_limit)$

  $all\_itemsets \leftarrow all\_itemsets \cup L_{k+1}$

  $k \leftarrow k + 1$

end

Global algorithm for extraction and combination of frequent itemsets

```
Data: instance_data (Flow and distance matrices), nb_solutions (10,000),
nb_generations (8), min_sup (0.1%) and itemsets_limit (1,000,000 for each k)
Result: solutions
forall i ← 1, . . . , nb_solutions do
    solutions[i] ← random_initialization()
    solutions[i] ← local_search(instance_data, solutions[i])
end
forall generation ← 1, . . . , nb_generations do
    all_itemsets ← extract_itemsets(solutions, min_sup, itemsets_limit) (Apriori algorithm)
    forall i ← 1, . . . , nb_solutions do
        solutions[i] ← combine_itemsets(all_itemsets)
        solutions[i] ← local_search(instance_data, solutions[i])
    end
end
```

Policies for combining itemsets

1. **REFI** Random exploration of all frequent itemsets
   - All itemset that can validly complete a partial solution have the same probability of being chosen

2. **ESFI**
   - An itemset has a probability proportional to its occurrence frequency of being chosen
     - Ant System use a similar construction mechanism: Pheromone trails corresponds to 1-itemsets
     - Tabu search exploits itemsets in the reverse way: 2-itemsets to forbid solutions characteristics, 1-itemsets to diversify the search
     - The cross-over operator in genetic algorithms extracts the largest itemset, then add 1-itemsets

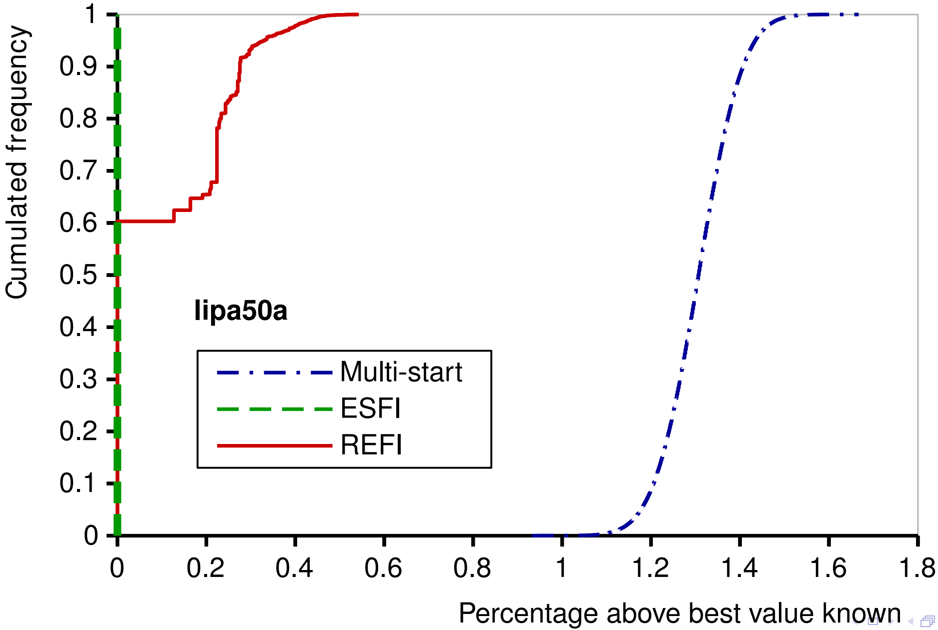
In case a complete solution is not obtained: unassigned items are randomly added

Computational experiments

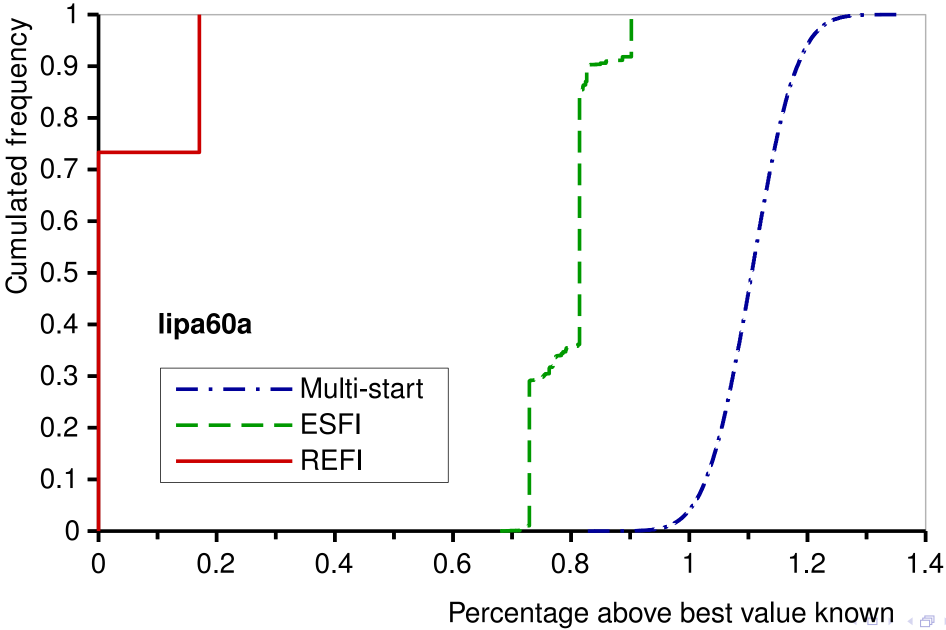
- Computational time
  - Huge
  - About 1 day/generation for $n = 50$ on a CPU with 16 cores (32 threads)
  - The goal was not to design a horse race method

- Instances selection
  - Li and Pardalos : Special generation, known optimum (lipa..a, lipa..b) [Li and Pardalos(1992)]
  - Nugent-Skorin-Kapov : Grid distances, uniform flows (sko..) [Skorin-Kapov(1990)]
  - Symmetric, uniform distances and flows (tai..a) [Taillard(1991)]
  - Asymmetrical, non-uniformly generated (tai..b) [Taillard(1995)]
  - Grey patterns, high number of optimal solutions (tai..c) [Taillard(1995)]
  - Symmetrical and structured (tai..e) [Drezner et al.(2005)Drezner, Hahn, and Taillard]

lipa50a

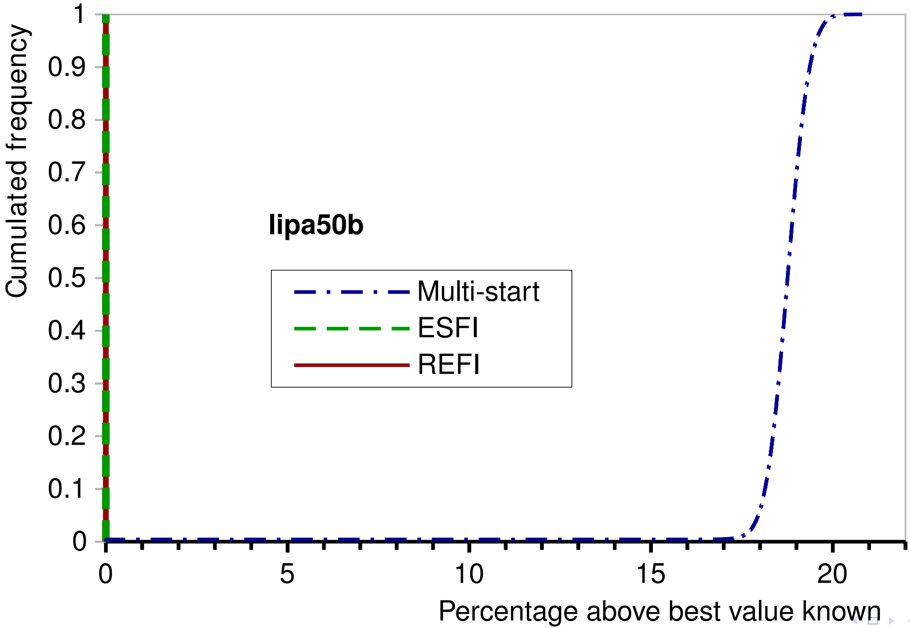

lipa50a
Cumulated frequency
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
Percentage above best value known
Multi-start
ESFI
REFI

lipa60a

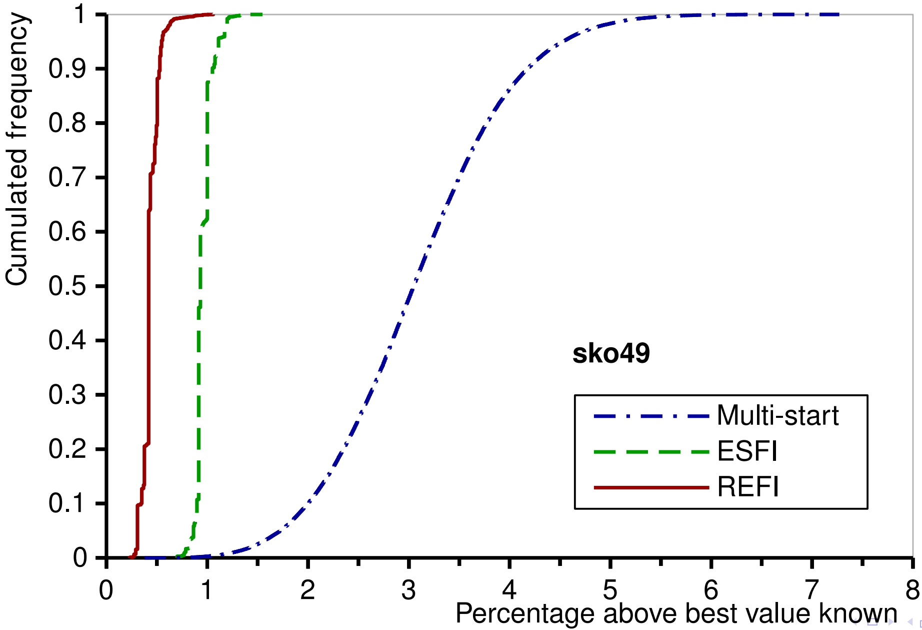


Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
lipa60a
Multi-start
ESFI
REFI
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Percentage above best value known

lipa50b

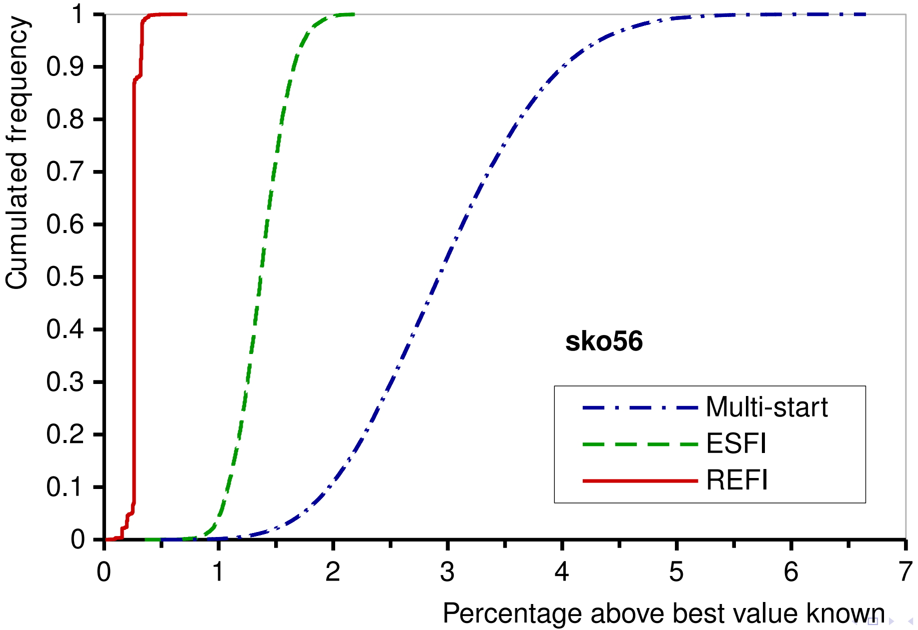

lipa50b
Multi-start
ESFI
REFI
Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
Percentage above best value known

sko49

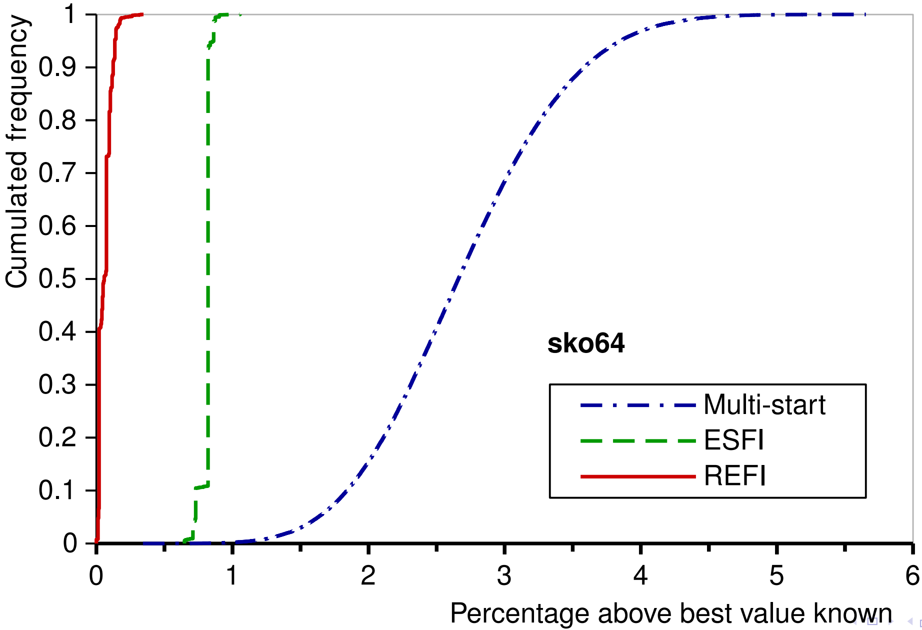


Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
Percentage above best value known
sko49
Multi-start
ESFI
REFI

sko56

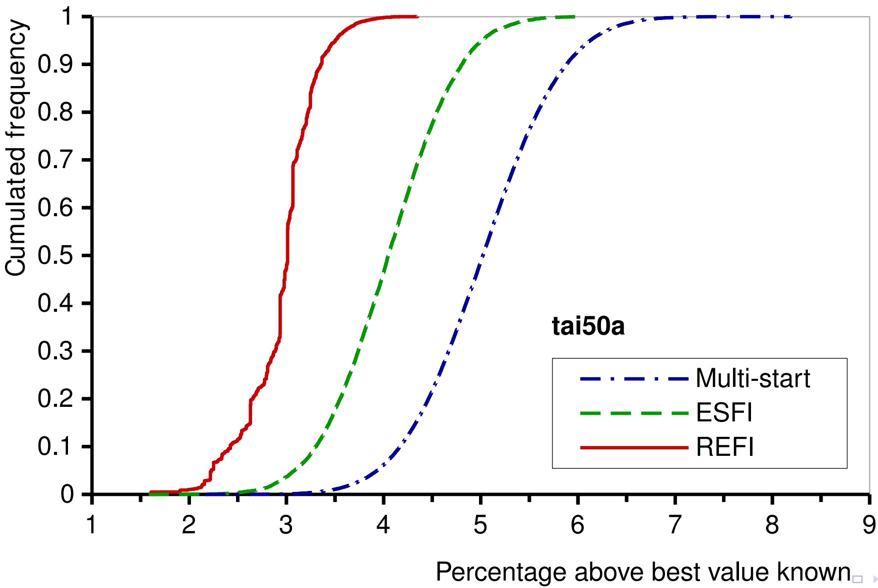

sko56
Cumulated frequency
Percentage above best value known
Multi-start
ESFI
REFI
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1
2
3
4
5
6
7

sko64



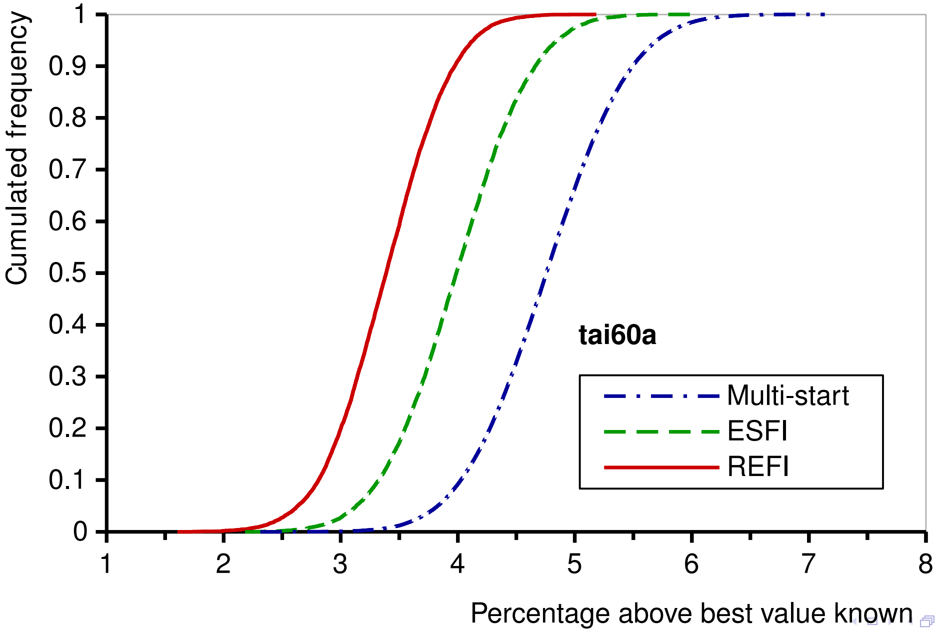
Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
Percentage above best value known
sko64
Multi-start
ESFI
REFI

tai50a

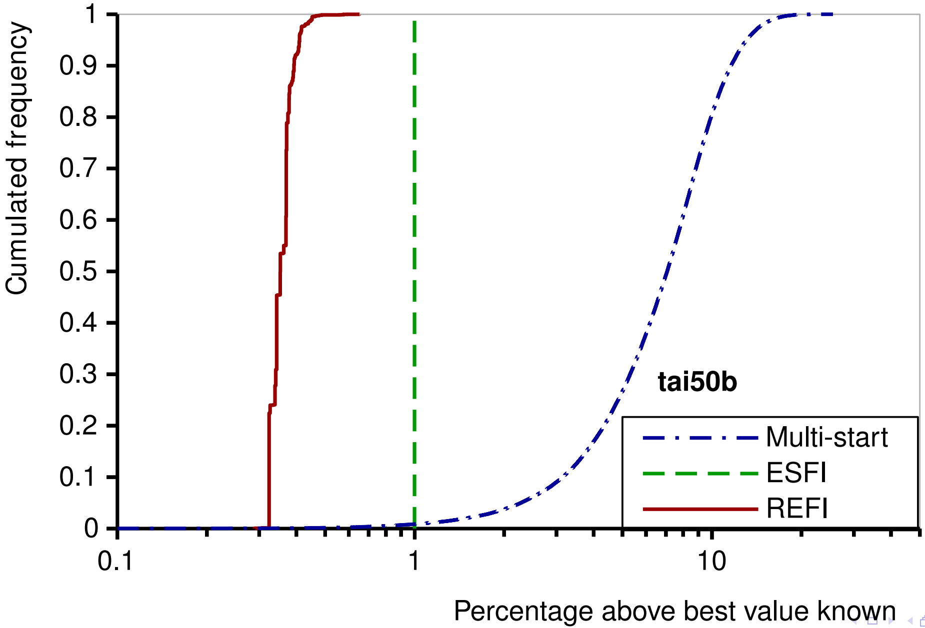


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Cumulated frequency
1
2
3
4
5
6
7
8
9
Percentage above best value known
tai50a
Multi-start
ESFI
REFI

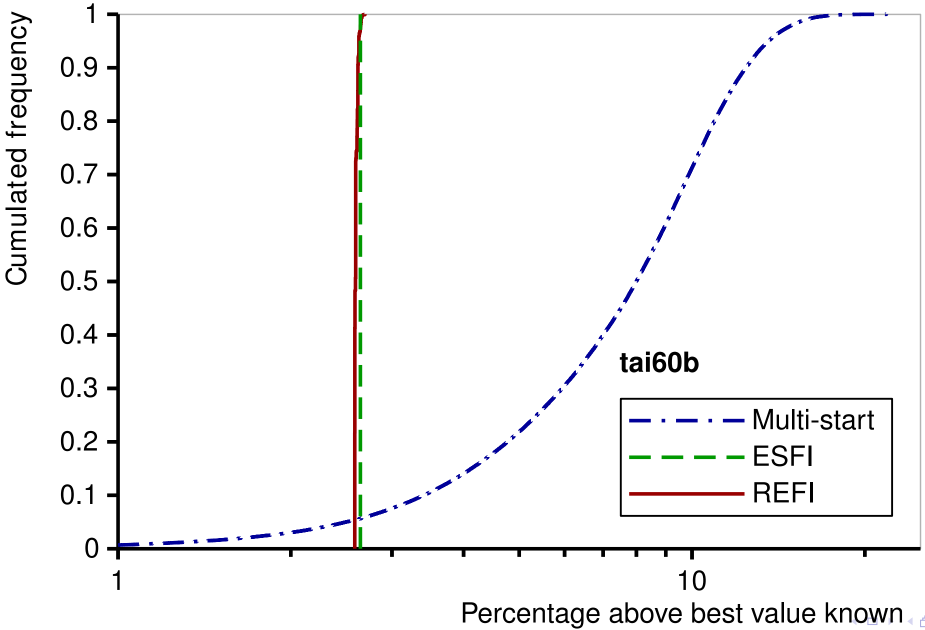
tai60a


Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
Percentage above best value known
tai60a
Multi-start
ESFI
REFI

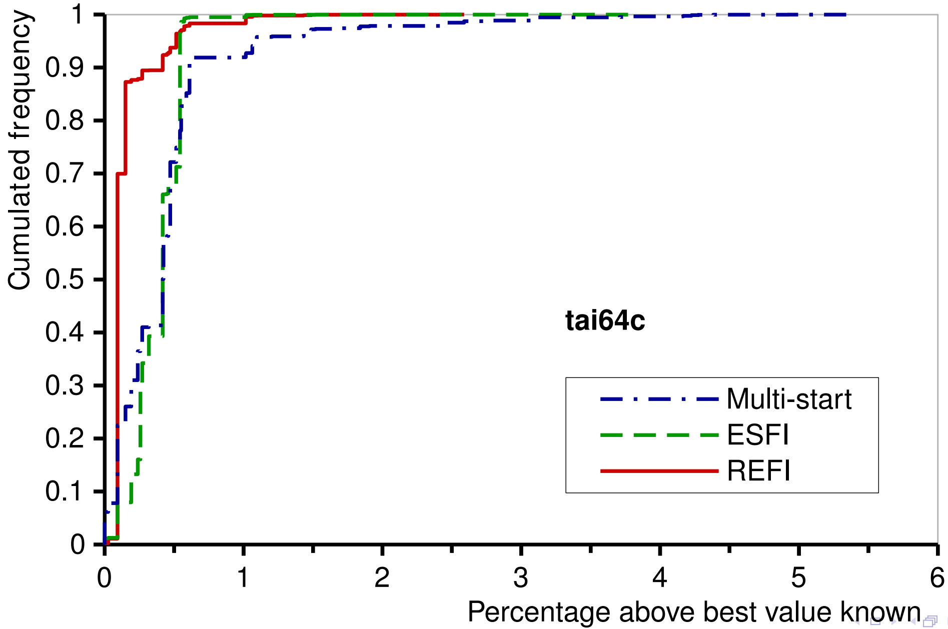
tai50b


Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
1
10
tai50b
Multi-start
ESFI
REFI
Percentage above best value known

tai60b

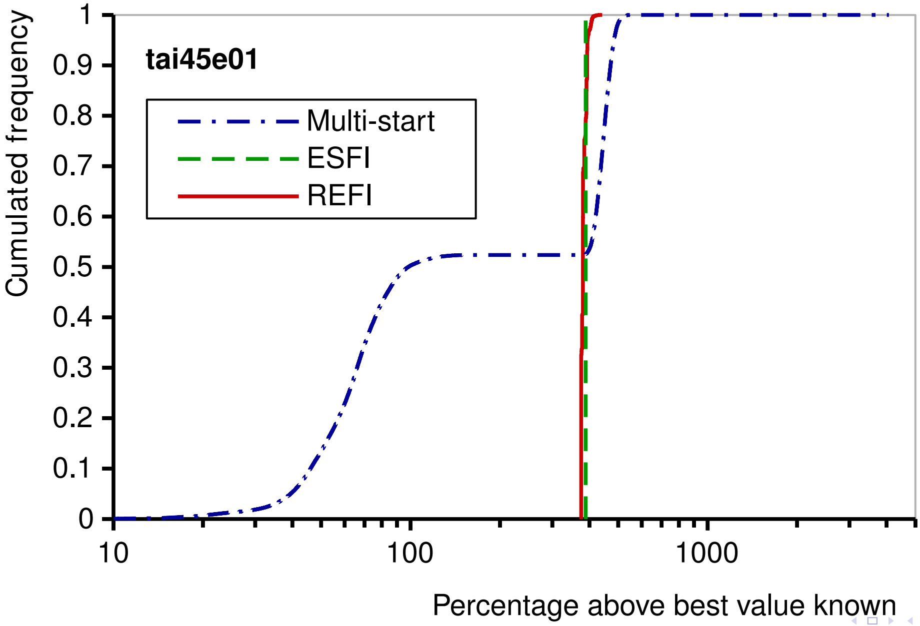

Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
10
Percentage above best value known
tai60b
Multi-start
ESFI
REFI

tai64c



Cumulated frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
Percentage above best value known
tai64c
Multi-start
ESFI
REFI

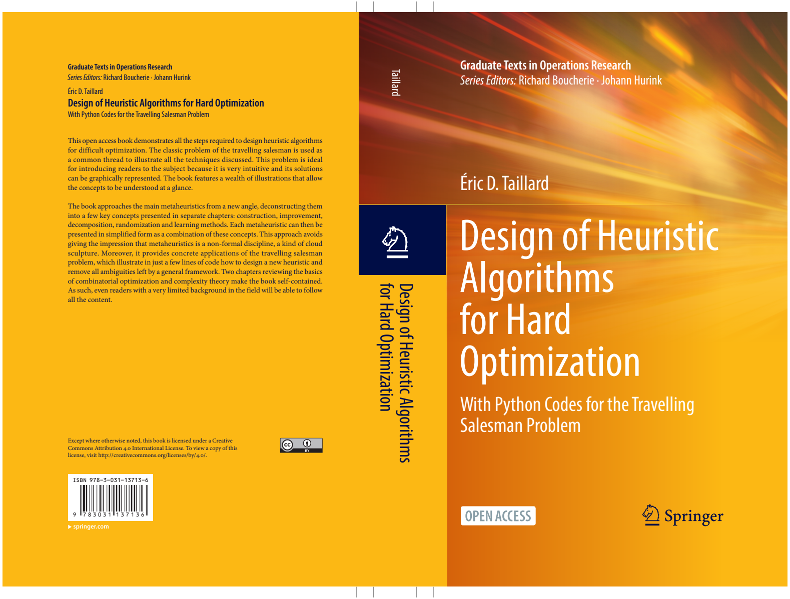
tai45e01


tai45e01
Multi-start
ESFI
REFI
Cumulated frequency
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10
100
1000
Percentage above best value known

Conclusions

- Independent local searches can hardly find solutions less than 1% above optimum (excepted for tai..c instances)
- REFI is generally better than ESFI
- Building solutions with itemsets frequently found in local optima is efficient for limited instance types (lipa.., sko)
- Unsupervised learning can be highly inefficient for the QAP, even for structured instances
- This contrasts with other problems (for the TSP, a very high fraction of the edges of an optimal solution can be found by few dozen of moderately good solutions [Taillard(2022), Taillard and Heslgaun(2019)])
- A learning mechanism with the supervision of the objective function (as usually done in metaheuristics) is much more efficient. See, e.g. FANT [Taillard(1998)], Ro-TS [Taillard(1991)]

Spam



Graduate Texts in Operations Research
Series Editors: Richard Boucherie · Johann Hurink

Éric D. Taillard
Design of Heuristic Algorithms for Hard Optimization
With Python Codes for the Travelling Salesman Problem

This open access book demonstrates all the steps required to design heuristic algorithms for difficult optimization. The classic problem of the travelling salesman is used as a common thread to illustrate all the techniques discussed. This problem is ideal for introducing readers to the subject because it is very intuitive and its solutions can be graphically represented. The book features a wealth of illustrations that allow the concepts to be understood at a glance.

The book approaches the main metaheuristics from a new angle, deconstructing them into a few key concepts presented in separate chapters: construction, improvement, decomposition, randomization and learning methods. Each metaheuristic can then be presented in simplified form as a combination of these concepts. This approach avoids giving the impression that metaheuristics is a non-formal discipline, a kind of cloud sculpture. Moreover, it provides concrete applications of the travelling salesman problem, which illustrate in just a few lines of code how to design a new heuristic and remove all ambiguities left by a general framework. Two chapters reviewing the basics of combinatorial optimization and complexity theory make the book self-contained. As such, even readers with a very limited background in the field will be able to follow all the content.

ISBN 978-3-031-13713-6
springer.com

Taillard
Design of Heuristic Algorithms for Hard Optimization

Graduate Texts in Operations Research
Series Editors: Richard Boucherie · Johann Hurink

Éric D. Taillard

Design of Heuristic Algorithms for Hard Optimization

With Python Codes for the Travelling Salesman Problem

OPEN ACCESS

Springer

- Çela, E.: The quadratic assignment problem — Theory and applications.
  Springer (1998)
- Drezner, Z., Hahn, P.M., Taillard, É.D.: Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods.
  Annals OR 139(1), 65–94 (2005),
  http://mistic.heig-vd.ch/taillard/articles.dir/DreznerHT2005.pdf
- Koopmans, T.C., Beckmann, M.: Assignment Problems and the Location of Economic Activities.
  Econometrica 25(1), 53–76 (1957), http://www.jstor.org/stable/1907742
- Li, Y., Pardalos, P.M.: Generating quadratic assignment test problems with known optimal permutations.
  Computational Optimization and Applications 1(2), 163–184 (1992)
- Skorin-Kapov, J.: Tabu search applied to the quadratic assignment problem.
  ORSA Journal on computing 2(1), 33–45 (1990)
- Taillard, É.D.: Robust taboo search for the quadratic assignment problem.

Parallel Computing 17(4-5), 443–455 (1991)

- Taillard, É.D.: Comparison of iterative searches for the quadratic assignment problem.
  Location Science 3(2), 87 – 105 (1995), http://www.sciencedirect.com/science/article/pii/0966834995000086
- Taillard, É.D.: Fant: Fast ant system.
  Tech. rep., Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale (1998), iDSIA Technical Report IDSIA-46-98
- Taillard, É.D.: A Linearithmic Heuristic for the Travelling Salesman Problem.
  EURO Journal of Operational Research 297(2), 442–450 (2022), http://mistic.heig-vd.ch/taillard/articles.dir/Taillard2021.pdf, available online June 2021
- Taillard, É.D., Heslgaun, K.: POPMUSIC for the travelling salesman problem.
  EURO Journal of Operational Research 272(2), 420–429 (2019), http://mistic.heig-vd.ch/taillard/articles.dir/TaillardHelsgaun2018.pdf