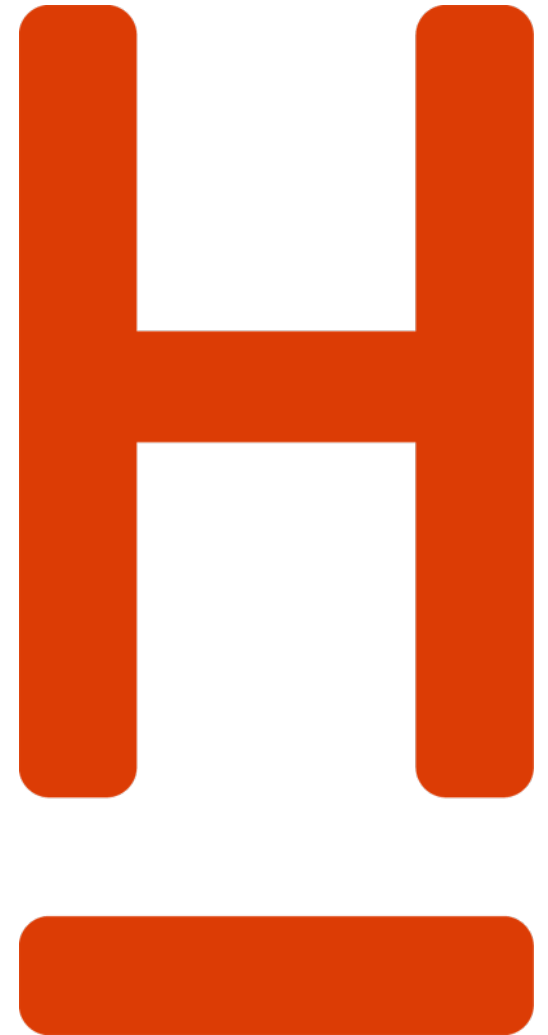


**HOCHSCHULE
HANNOVER**
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

–
*Fakultät IV
Wirtschaft und
Informatik*

Improvements on AFB

Advancing the Metaheuristic for TSP



Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



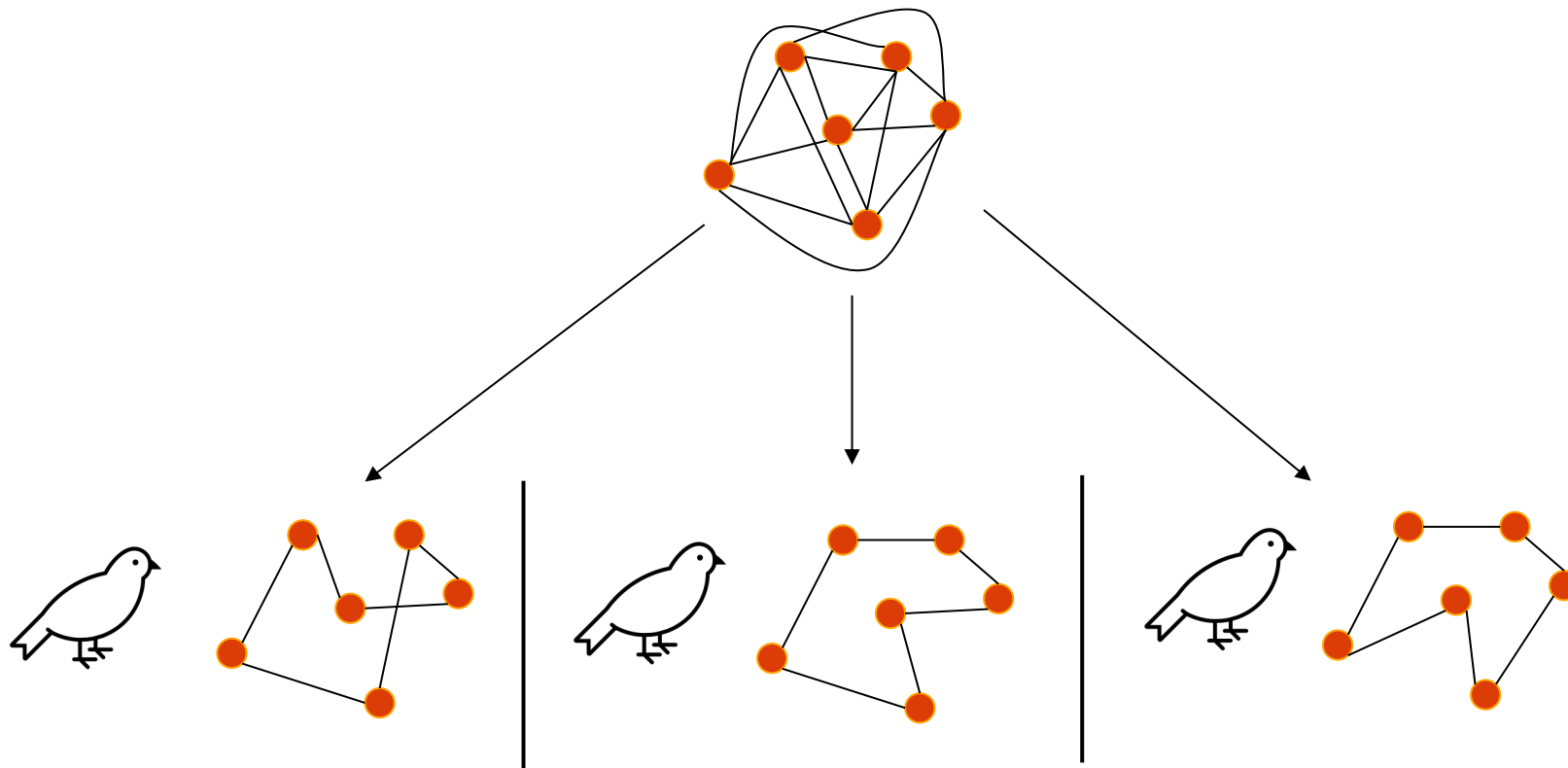
Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>

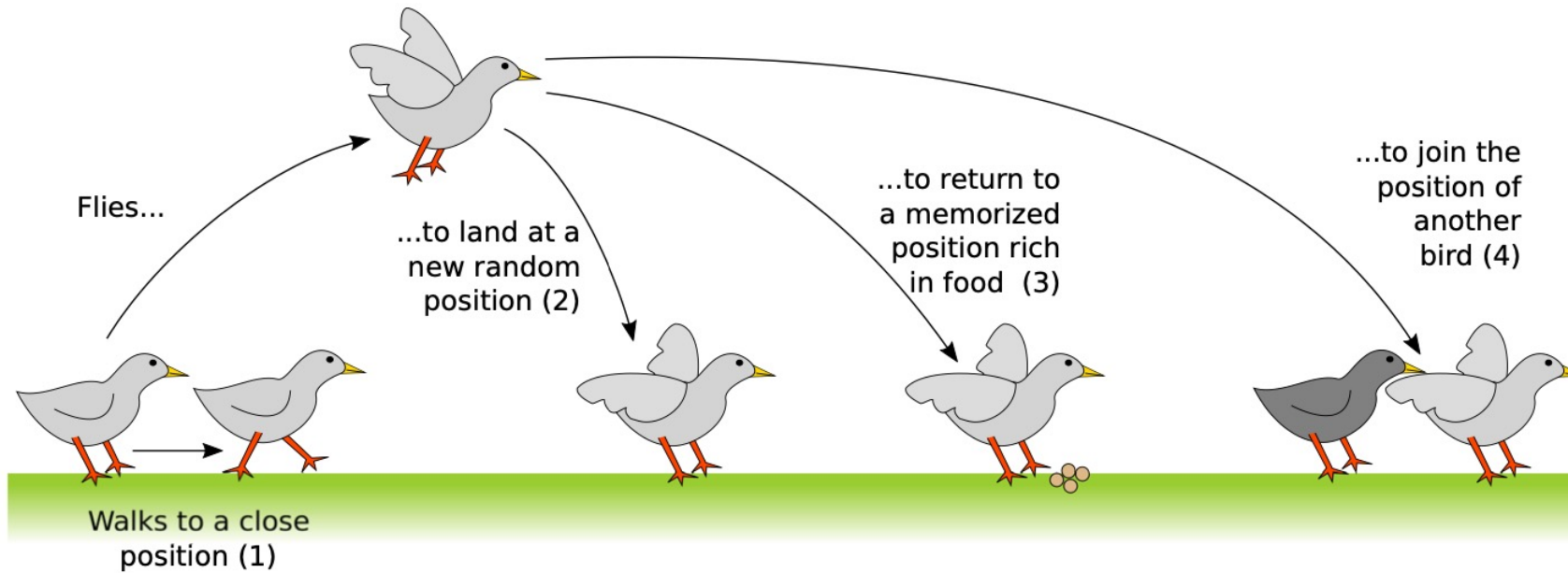


Recap

- Each Bird represents one possible solution (one tour)
- Each operation performed by a bird, alters its respective solutions



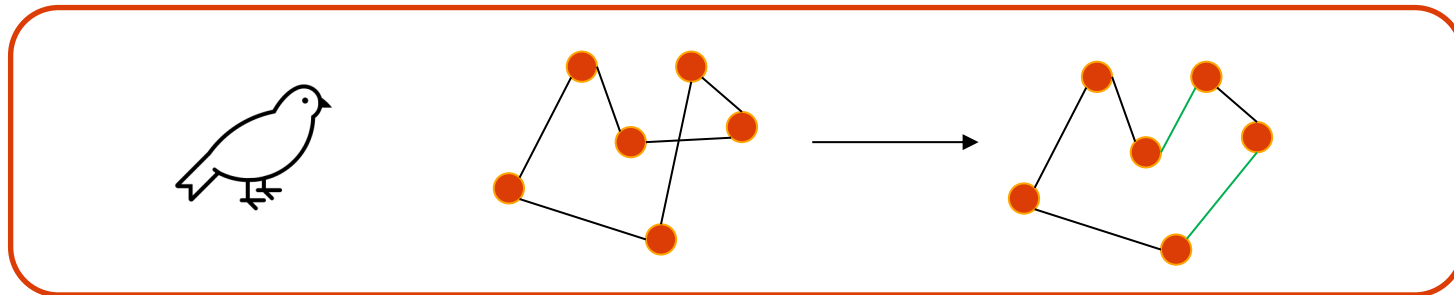
Recap



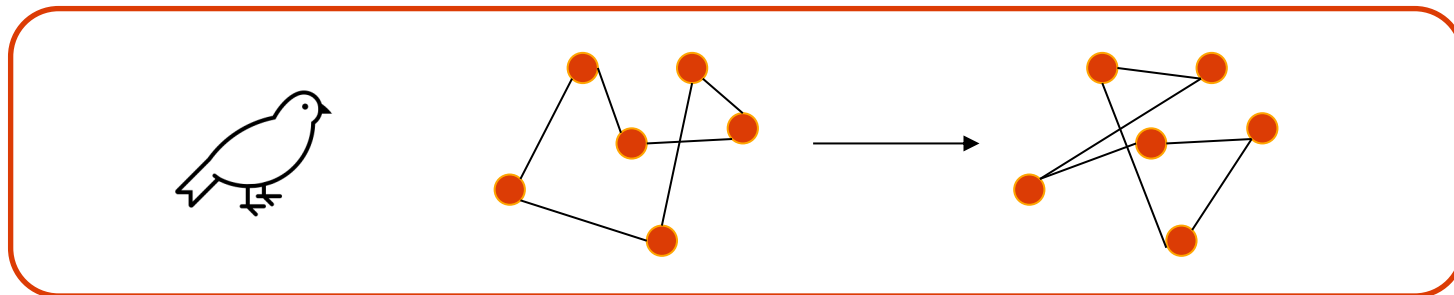
Recap

- Each action of a bird corresponds to a change of its own solution
- Each solution is valid
- The number of candidate solutions (or agents respectively) does not change (currently)

(1) Walk

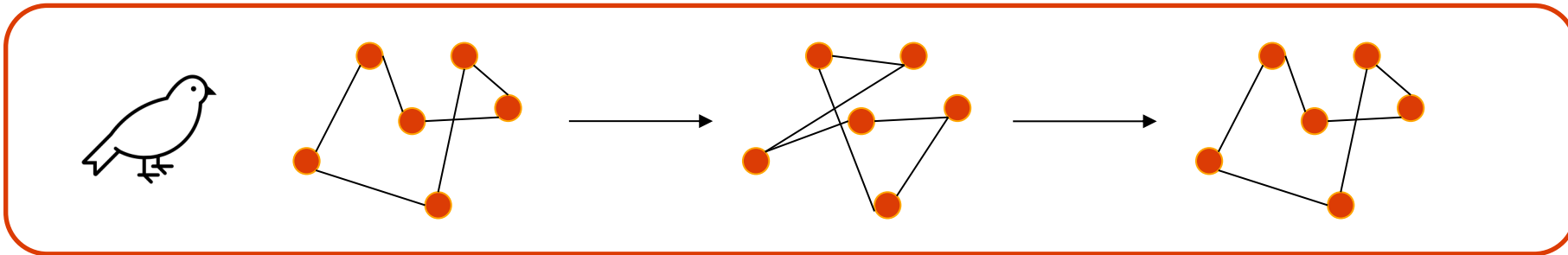


(2) Fly

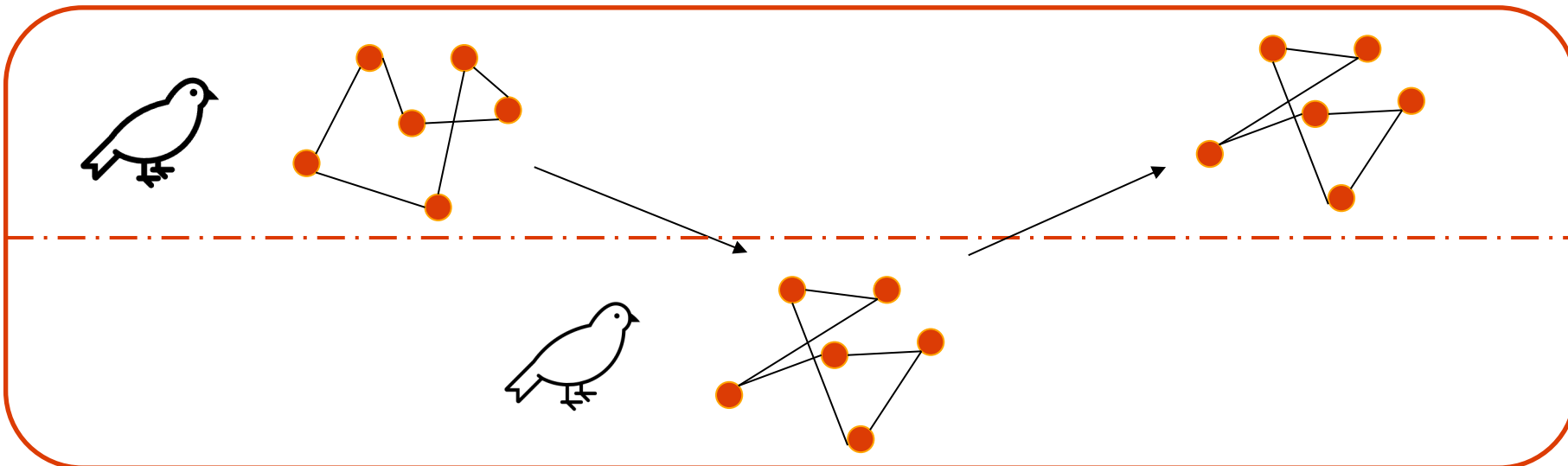


Recap

(3) Return



(4) Join



Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



Methodology

- To benchmark our improvements, we select five feasible solutions from TSPLIB
 - Each problem has a different order of magnitude to account for the variety of different configurations possible
- Each problem is run 10x, to account for the randomness (50 test in total)
- We record the mean percentage error, and the mean time in seconds
- Problems: eil101, pa561, pr1002, u2156, pr2392

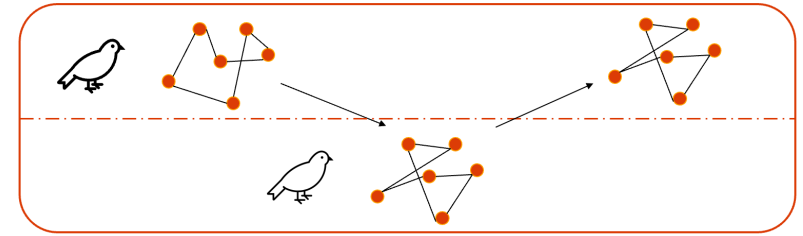


Inhaltsverzeichnis

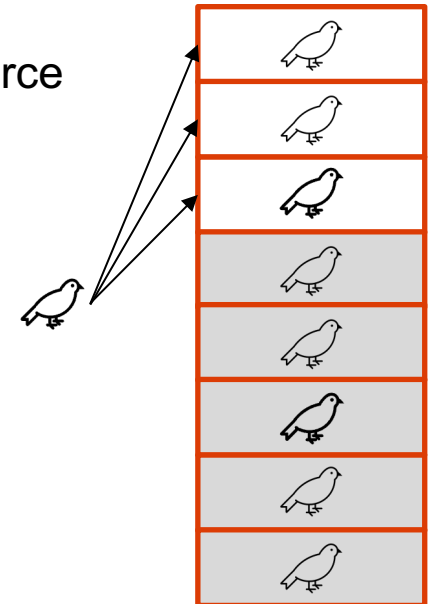
Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



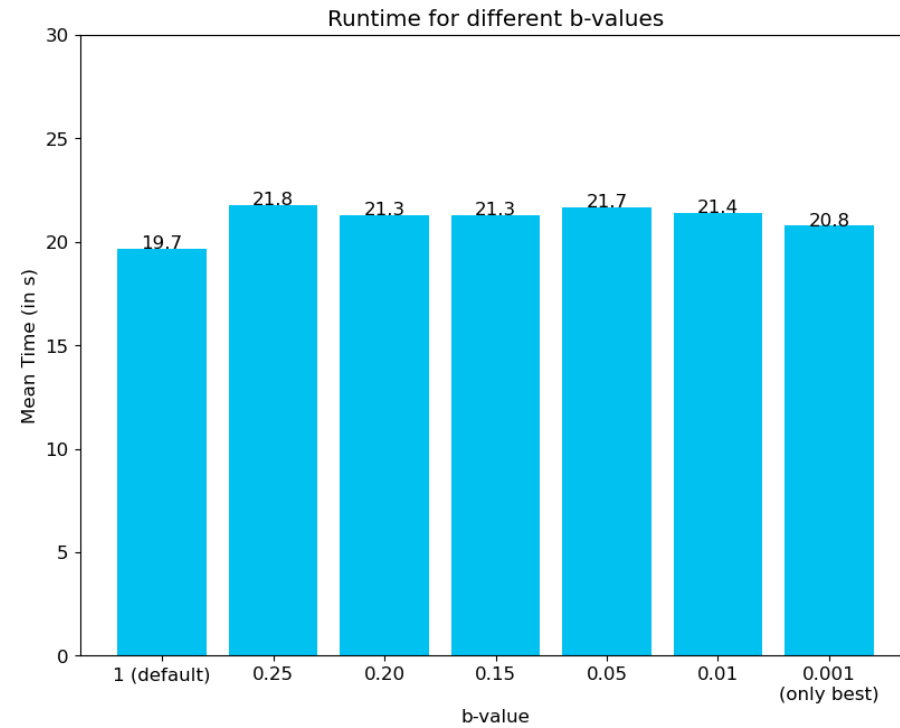
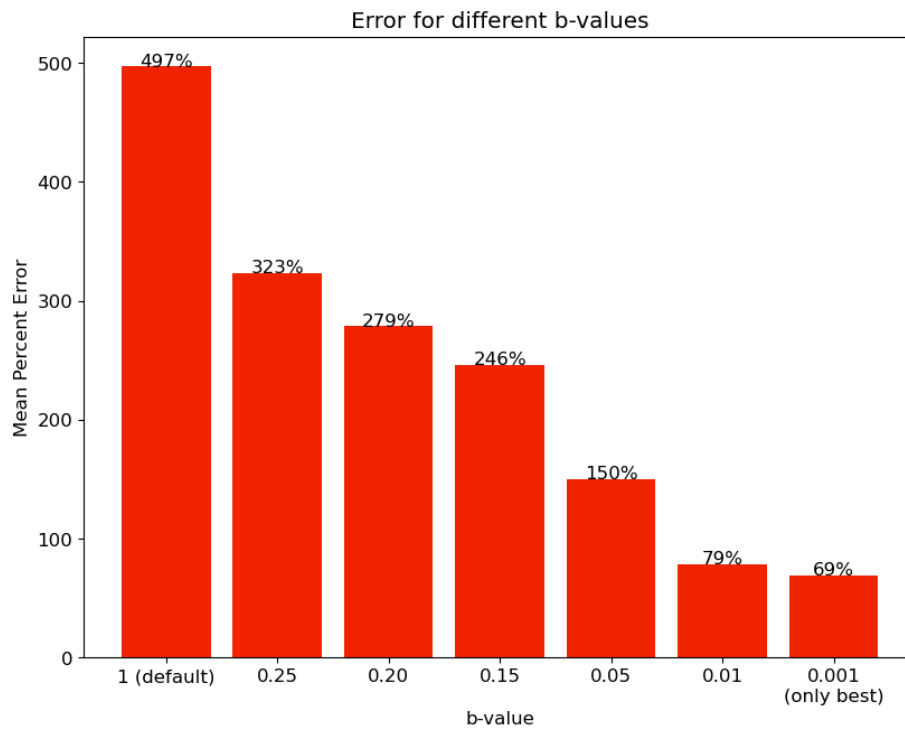
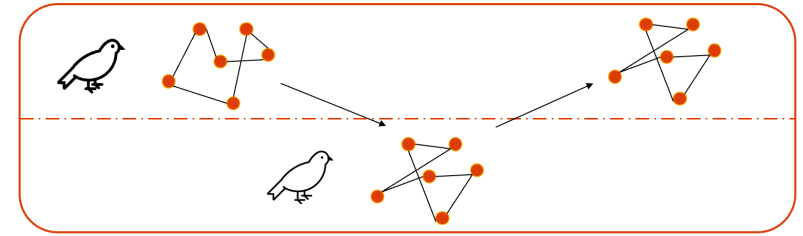
Top-b Join



- Default behavior: If a big bird joins another, he chooses one randomly
- Contradicts the idea that birds tend to join others, if they found a good food source
 - Good food source translates to a good solution
- That is why we decide to allow a big bird to only join the top-b percent
 - Pick one of the top-b birds randomly
- Means ordering the birds by their tour length after each iteration/phase
 - Increases runtime due to sorting complexity



Top-b Join

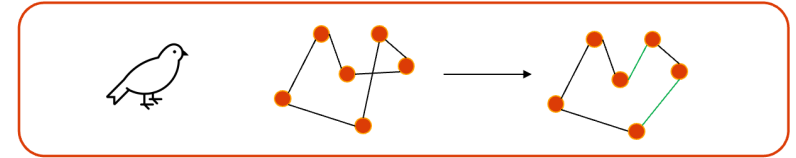


Inhaltsverzeichnis

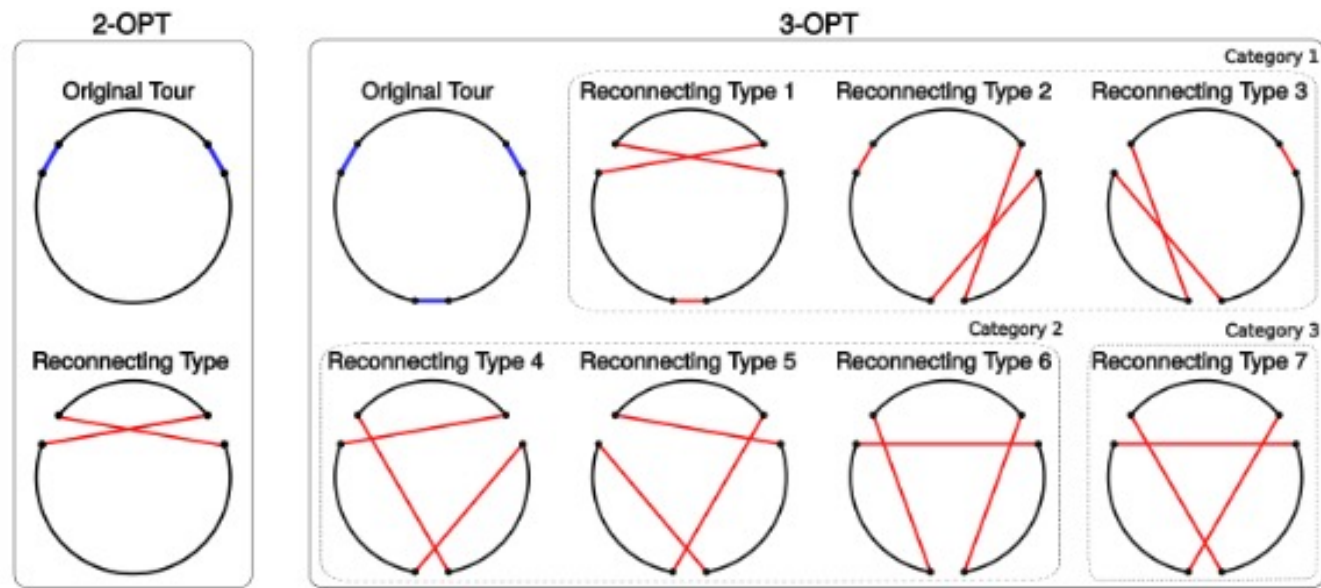
Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



3-Opt



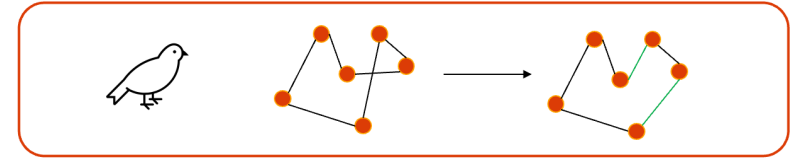
- When performing the walk-operation, so the local search, a bird uses 2-opt to search for a potential better solution
- Naturally, we also tested 3-opt as a more powerful alternative



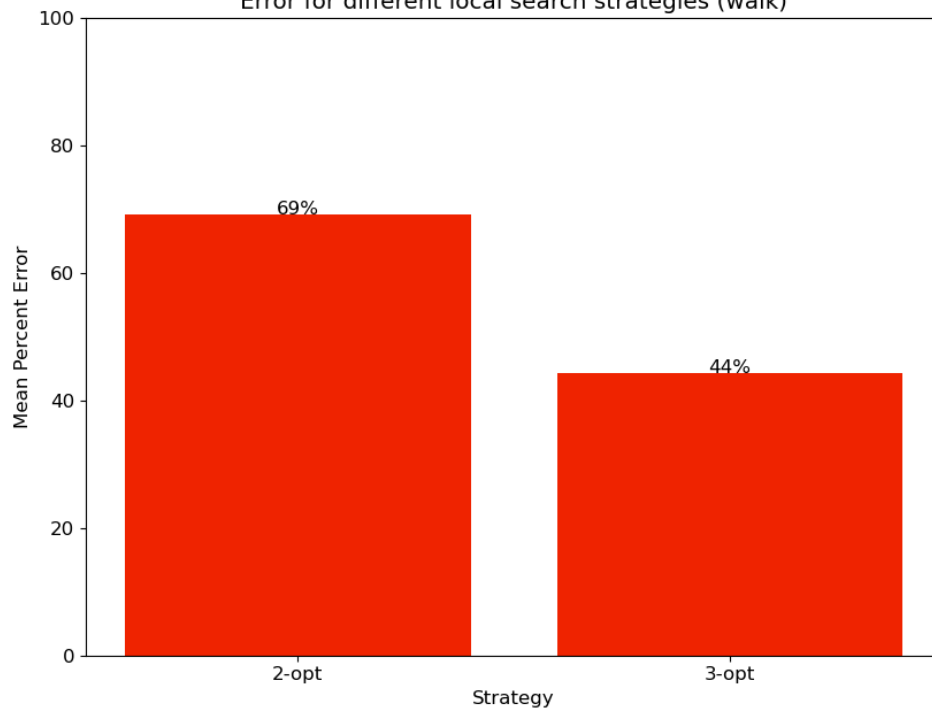
Source: Jingyan Sui, Shizhe Ding, Ruizhi Liu, Liming Xu, Dongbo Bu. Learning 3-opt heuristics for traveling salesman problem via deep reinforcement learning. Proceedings of The 13th Asian Conference on Machine Learning, PMLR 157:1301-1316, 2021.



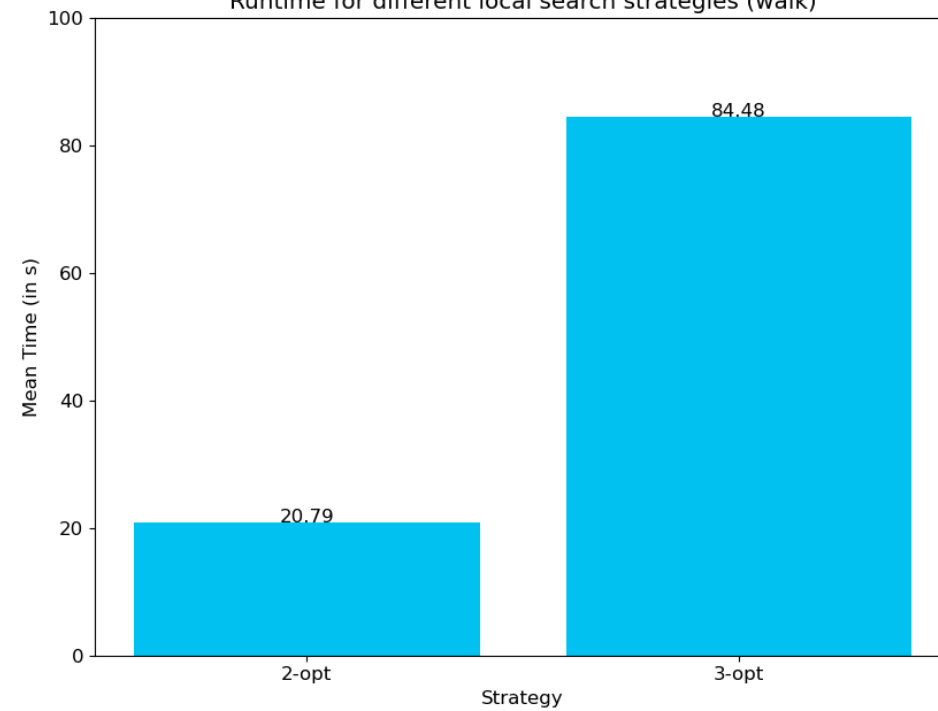
3-Opt



Error for different local search strategies (walk)



Runtime for different local search strategies (walk)

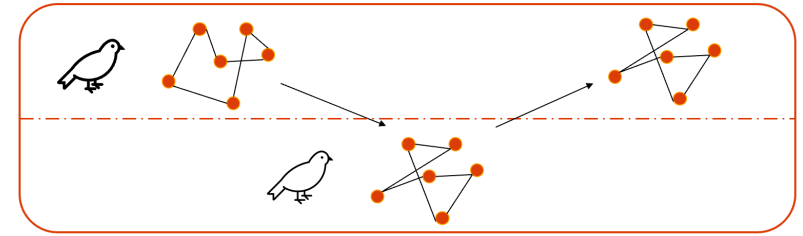


Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



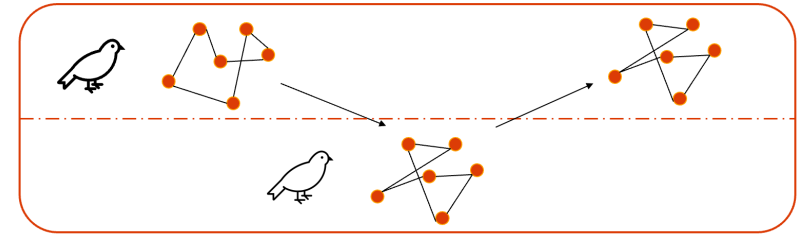
Delegating Responsibility



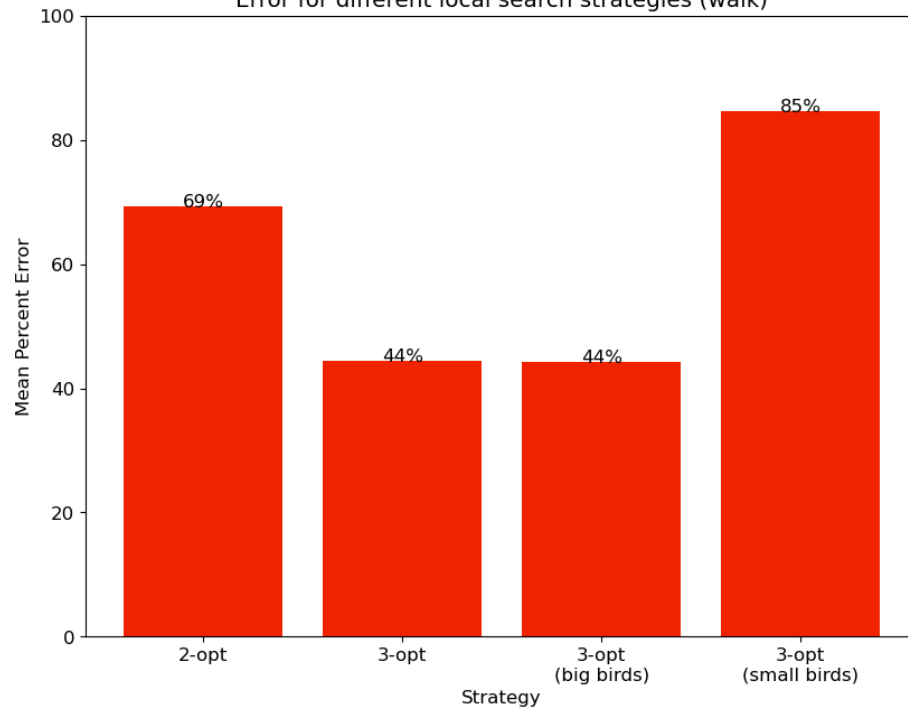
- Seen before: 3-opt (+ sorting for top-b join) yield very high computation effort
- How can one make the algorithm faster while keeping the performance close to before?
- Answer: Allow only big/small birds to perform 3-opt, the other 2-opt
 - Both were tested, but big birds make more sense regarding their “superiority”



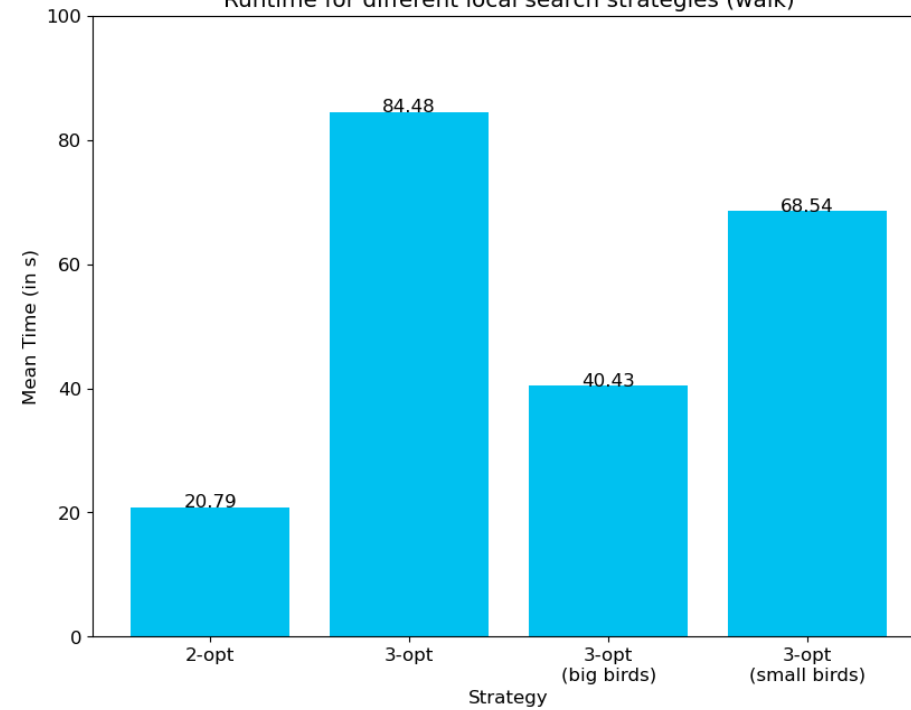
Delegating Responsibility



Error for different local search strategies (walk)



Runtime for different local search strategies (walk)

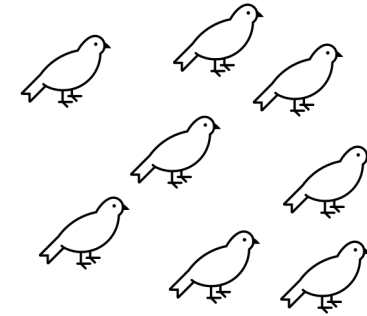


Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



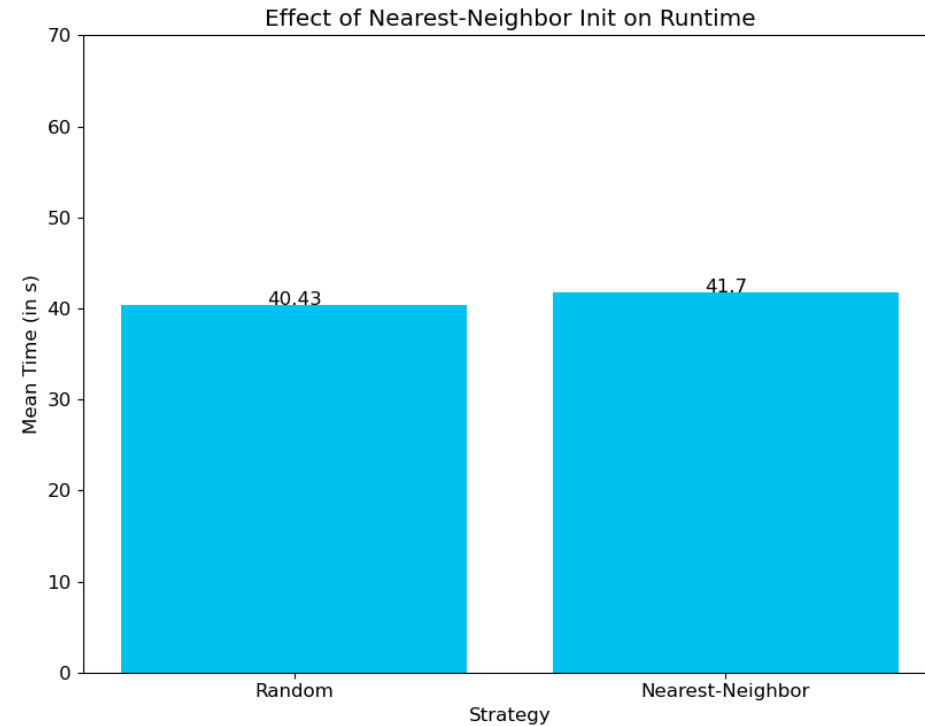
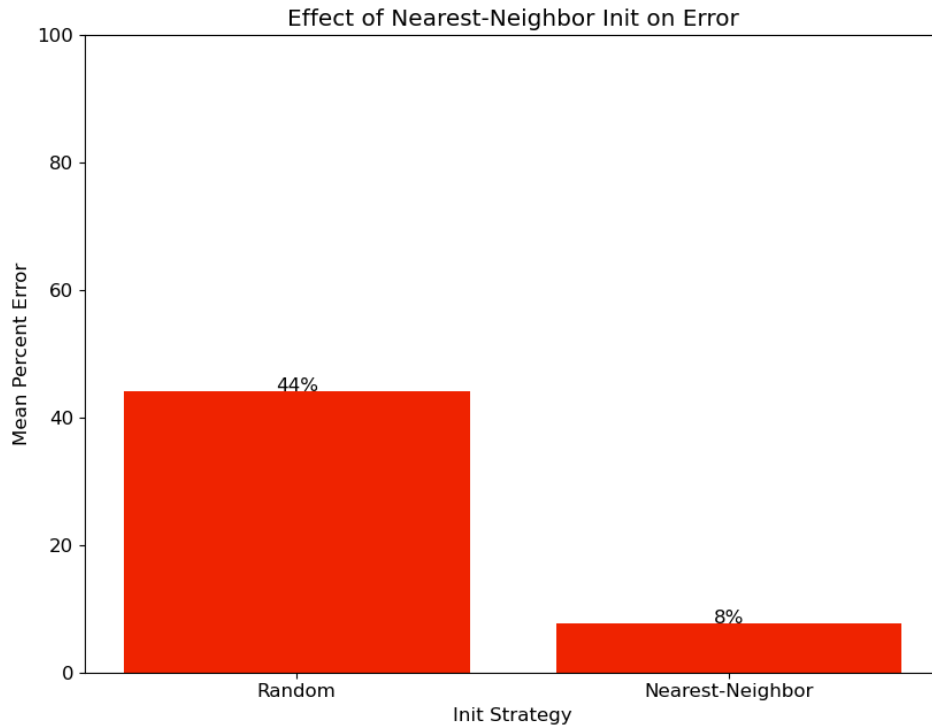
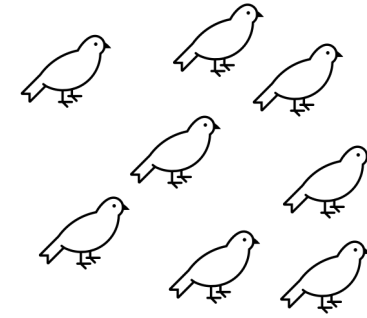
Nearest-Neighbor Initialization



- Instead of starting with random tours, use a simple heuristic for initialization
- Nearest neighbor algorithm
 - Runtime: $O(n^2)$
- Algorithm
 - Start with a random vertex, mark it as visited
 - Add the nearest neighbour of the current vertex, mark it as visited and make it the new current vertex
 - If all vertices are visited the tour is complete



Nearest-Neighbor Initialization

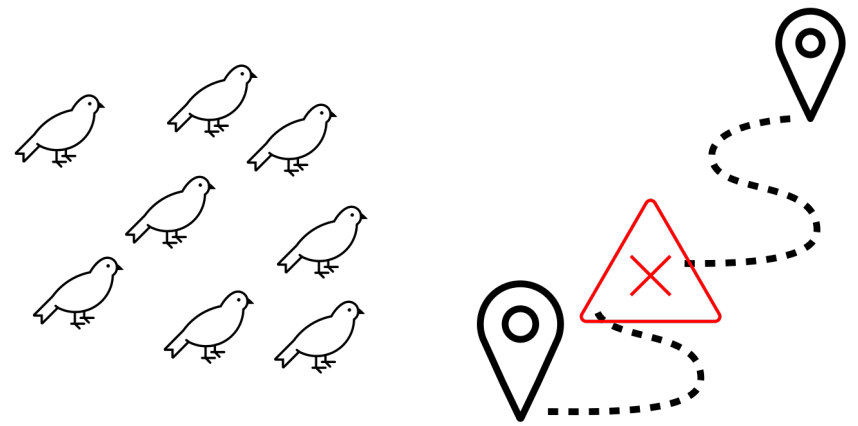


Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



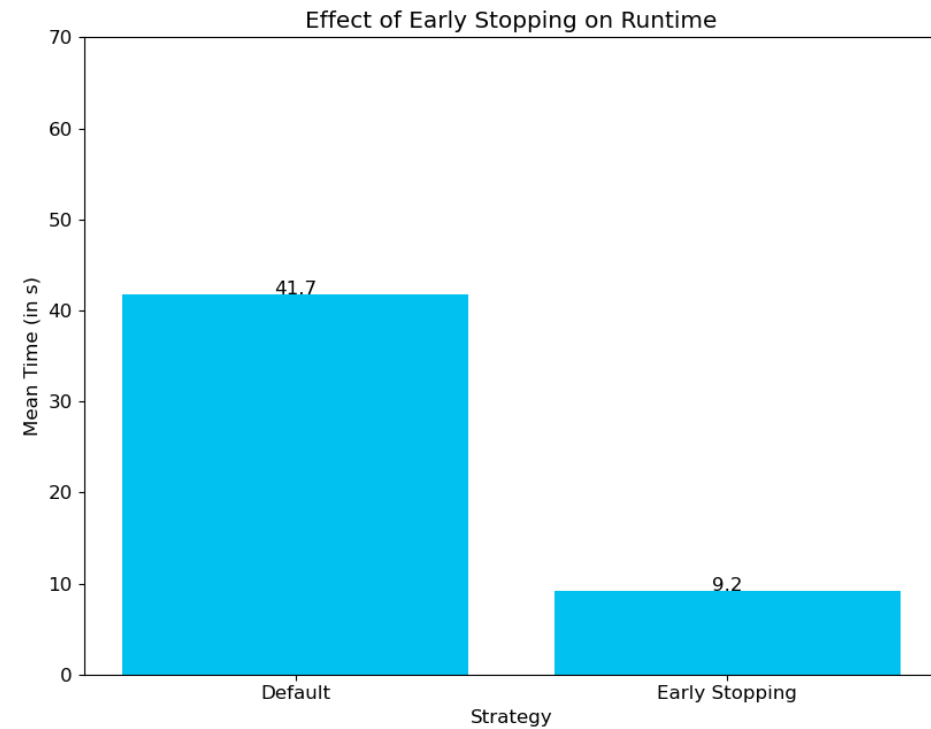
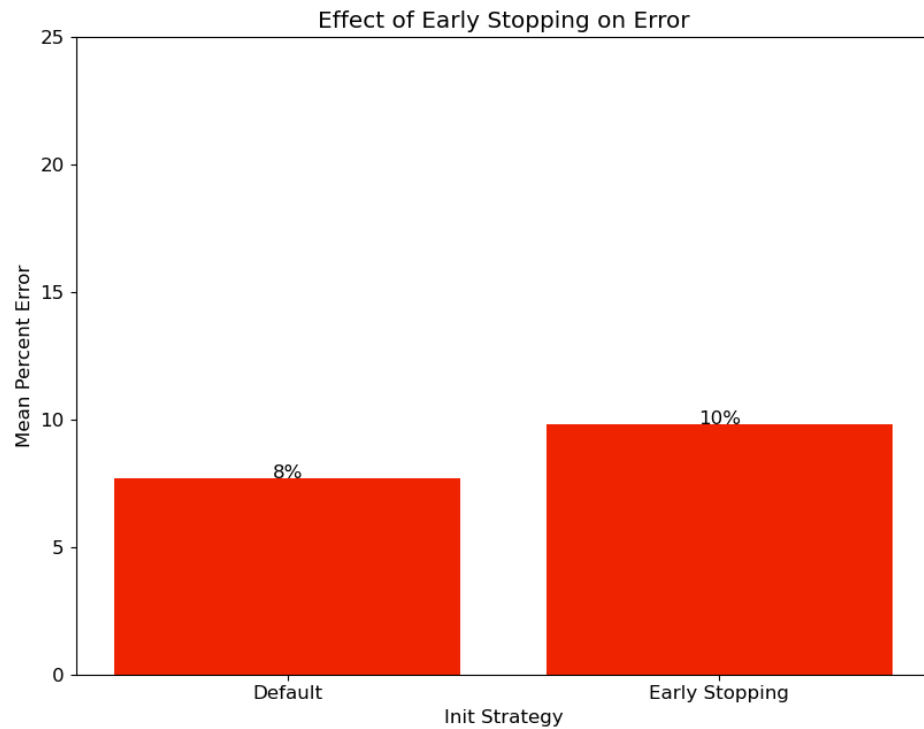
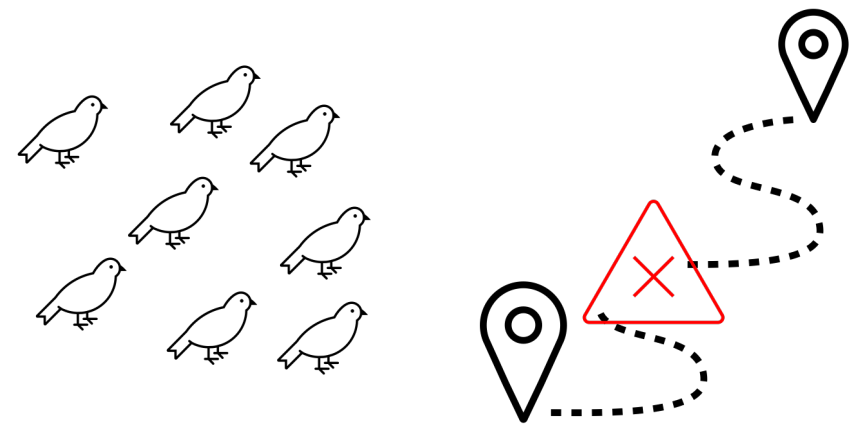
Early Stopping



- Algorithm shows very fast convergence behavior
 - Especially for problems with a rather low number of cities
- How many iterations are needed to achieve a good result for a given problem is difficult to estimate
- Therefore, a predefined number of iterations yields unnecessary long computation times that do not improve the results
- One solution is to stop the algorithm, if the current solution(s) do not improve



Early Stopping

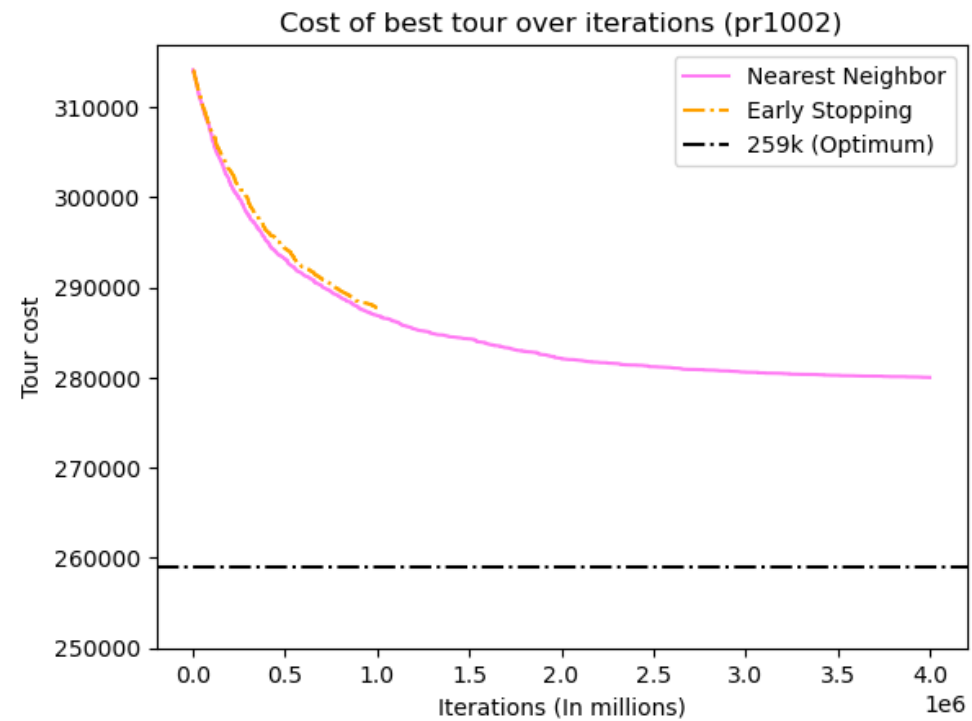
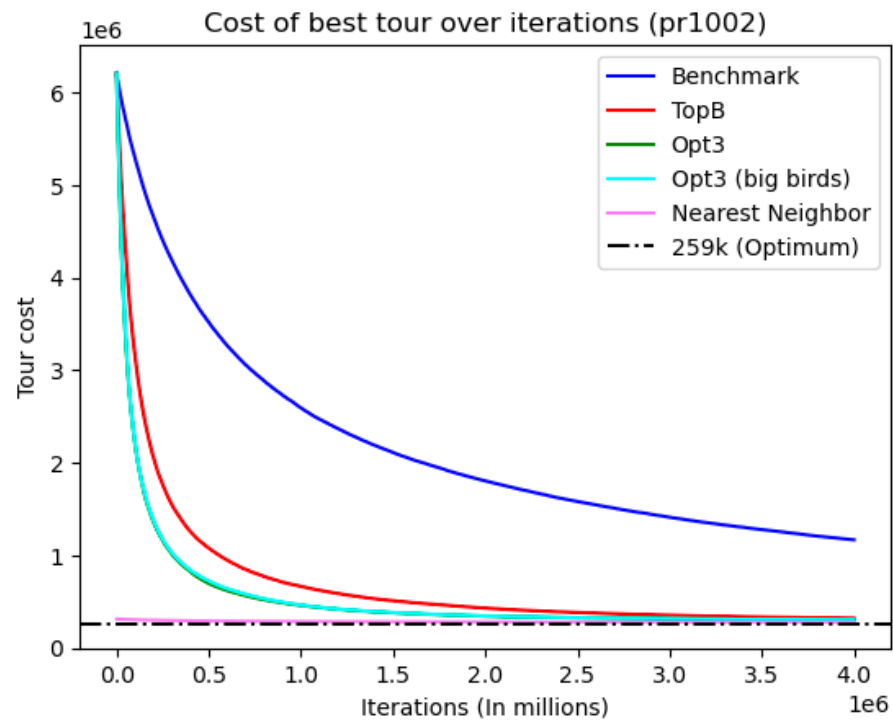
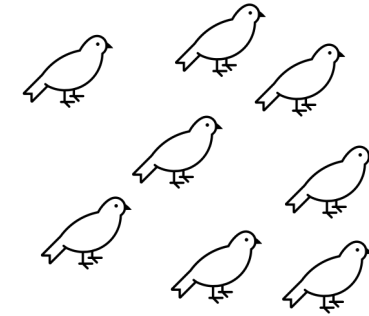


Inhaltsverzeichnis

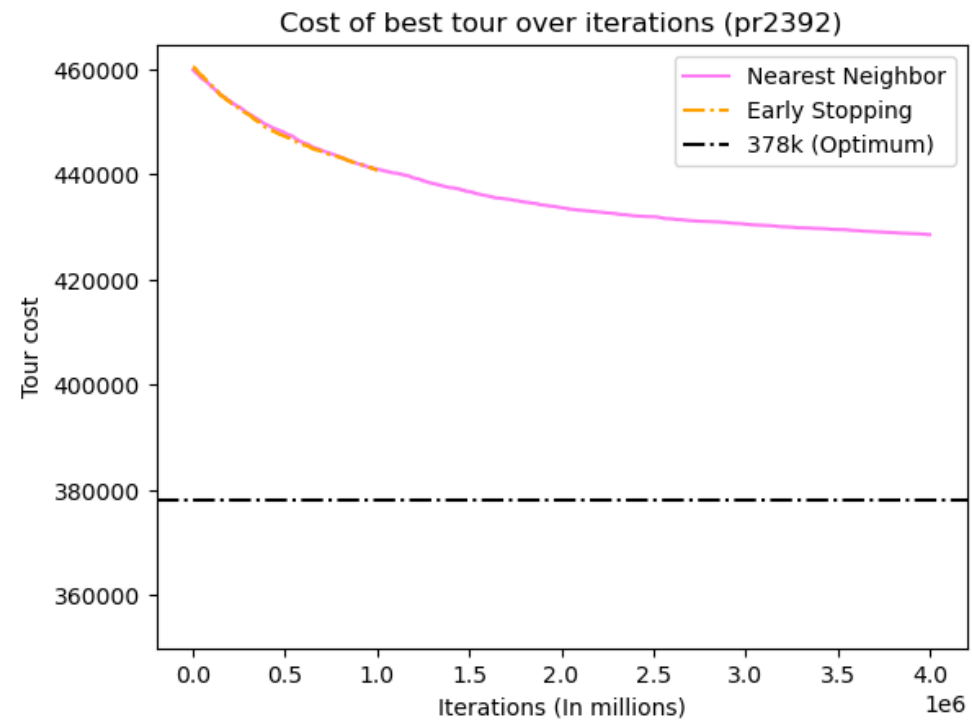
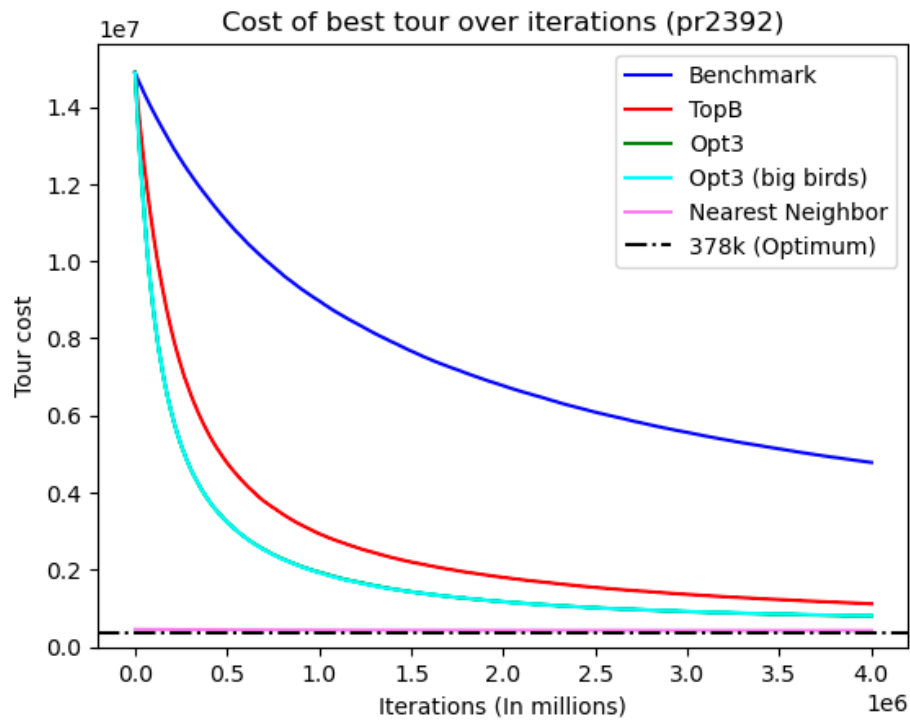
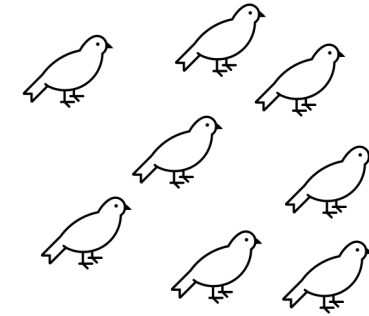
Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



Optimization Behavior



Optimization Behavior



Inhaltsverzeichnis

Chapter 1	Recap	<i>Page 4</i>
Chapter 2	Methodology	<i>Page 9</i>
Chapter 3	Top-b Join	<i>Page 11</i>
Chapter 4	3-Opt	<i>Page 14</i>
Chapter 5	Delegating Responsibility	<i>Page 17</i>
Chapter 5	Nearest-neighbor Initialization	<i>Page 20</i>
Chapter 6	Early Stopping	<i>Page 23</i>
Chapter 7	Optimization Behavior	<i>Page 26</i>
Chapter 8	Metabirds	<i>Page 29</i>



Metabirds



- How do you choose hyperparameters like move probabilities or small bird ratio?
- Apply an optimization algorithm to find optimal values
- What optimization algorithm? Artificial Feeding Birds!
- A Metabird's position is a value for all probabilities, ratios, etc.
 - Flying generates a random position in the hyperparameter space
 - Walking adds or subtracts a random delta from each parameter
 - Calculating the fitness of a Metabird
 - A TSP solver is instantiated with the hyperparameters of the Metabird
 - Multiple runs solving a TSP are averaged to assess the performance with the current parameters.



Vielen Dank für Ihre Aufmerksamkeit!



Literature

- Jean-Baptiste Lamy. Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons. Advances in nature-inspired computing and applications, 2019, 10.1007/978-3-319-96451-5_3 . hal-02264232

