Zeszyty Naukowe WWSI, No 19, Vol. 12, 2018, s. 7-24
DOI: 10.26348/znwwsi.19.7

Dynamics of Stochastic vs. Greedy Heuristics in Traveling Salesman Problem

Maciej Białogłowski*, Mateusz Staniaszek, Wojciech Laskowski, Marcin Grudniak

Warsaw School of Computer Science

Abstract

We studied the relative performance of stochastic heuristics in order to establish the relations between the fundamental elements of their mechanisms. The insights on their dynamics, abstracted from the implementation details, may contribute to the development of an efficient framework for design of new probabilistic methods. For that, we applied four general optimization heuristics with varying number of hyperparameters to traveling salesman problem. A problem-specific greedy approach (Nearest Neighbor) served as a reference for the results of: Monte Carlo, Simulated Annealing, Genetic Algorithm, and Particle Swarm Optimization. The more robust heuristics – with higher configuration potential, i.e. with more hyperparameters – outperformed the smart ones, being surpassed only by the method specifically designed for the task.

Keywords **–** Traveling Salesman Problem, Nearest Neighbor, Monte Carlo, Simulated Annealing, Genetic Algorithm, Particle Swarm Optimization

* E-mail: m_bialoglowski@poczta.wwsi.edu.pl

Manuscript received October 24, 2018.

1. Introduction

Heuristics bypass the complexity of NP-hard problems, by offering approximate solutions (with no guarantee of success). An expert can yield repeatedly good results with a problem-specific or a fine-tuned general method. More sophisticated heuristics, which originate from natural analogies, often reduce the number of knobs (hyperparameters), but do not necessarily outperform the crude ones. The mechanisms which simplify their usage can also limit their flexibility. Performance requirements determine the trade-off between the ease of application and the efficiency of chosen algorithms/heuristics.

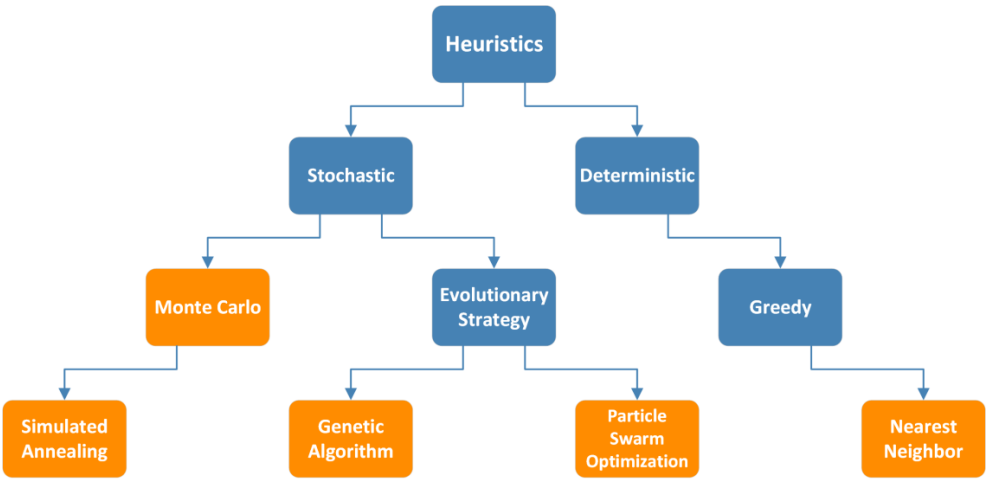

Figure 1. Selected taxonomy of heuristics. The top partition (based on the use of randomization) distinguishes between general (left) and problem-specific (right) methods. The abstract ones (too ambiguous to implement) are marked in blue.

Figure 1 shows a simplified classification of heuristics. Each of them contains a form of solution space representation and a cost function – a way to quantitatively compare competing solutions. Most of the general optimization methods are stochastic, i.e. exploit the search space with a certain degree of randomness. Heuristics related to Monte Carlo evaluate one solution at a time, whereas evolutionary strategies

diversify their scope, considering multiple directions in parallel. On contrary – greedy heuristics, such as Nearest Neighbor, arrive at the solution in a predetermined manner.

Random Sampling – the simplest version of Monte Carlo – generates a large number of entirely independent solution candidates and selects the one with the highest score. The more solutions it checks, the higher the chance of a satisfactory result. Hill Climbing modifies this approach: each next solution is a slight modification of the best so far [1]. It speeds up the search, but creates the vulnerability that the process will get stuck in a local optimum. Simulated Annealing addresses this issue with a decaying probability to temporarily accept an inferior solution [2]. Tabu Search goes even further to prevent repeated evaluations of the same states [3].

Survival of the fittest inspired the evolutionary approach to optimization: multiple solutions compete between one another to reproduce. The relatively best ones become parents of slightly modified offspring, whereas the weaker ones do not get this opportunity. Genetic Algorithms introduce gene-like encoding of features and gamic reproduction through genome crossover (some variants) [4]. Similar mechanism, but with a different narrative, treats the solutions as a swarm of particles or animals, which move through solution space either following the strongest one (Pack Leader, Queen Bee, Alpha Pigeon) or slightly deviating from that route [5].

Differences in nomenclature of hyperparameters originate from the specific narratives, often describing the same features (Table 1.; the Nearest Neighbor can be used *ad hoc*). The exact number of these knobs depends on the implementation: build-in mechanisms reduce it, while deep analogies may increase it. We can distinguish between hyperparameters that dictate the number of evaluated solutions (e.g. Iterations, Generations × Individuals) and the dynamics of simulation (e.g. Mutation Probability, K Constant). The latter must be set within a certain range, outside of which the optimization becomes highly unstable and useless. Fine-tuning within these (problem-specific) boundaries impacts the final result.

Table 1. Hyperparameters of the selected heuristics

Nearest Neighbor	Monte Carlo	Simulated Annealing	Genetic Algorithm	Particle Swarm Optimization
	Iterations	Cooling Steps Steps Per Temp Cooling Fraction Initial Temperature K Constant	Generations Individuals Mutation Probability	Iterations Size

Travelling Salesman Problem (TSP) closely resembles finding a Hamiltonian Cycle, but focuses on weighted graphs: it seeks the shortest Hamiltonian Cycle in a densely connected system. Many real-life optimization problems (e.g. transport logistics, robotic assembly) revolve around NP combinatorial search, thus can be reduced to TSP [1]. Such a standardized solution framework allows to fairly compare heuristics, which claim superiority in seemingly distant applications.

Smart implementations of data structures or parallel computing can boost the performance of certain heuristics. Therefore, instead of measuring their score per time, we measured it in relation to the number of evaluated solutions – calling them steps. The behavior of methods in each step abstracts their search dynamics from the implementation details. Furthermore, carefully designed datasets may favor one method over another. To minimize that risk, we analyzed the heuristics on real geographical datasets. Surprisingly, the robust methods outperformed the more sophisticated ones.

2. Experimental Section

Figure 2 illustrates TSP datasets used for evaluation of the following heuristics [7]:

Nearest Neighbor – used *ad hoc*, tested for every possible starting point

Monte Carlo (Random Sampling)

- Iterations in the range [10 000, 1 000 000]

Simulated Annealing

- Cooling Steps in the range [100, 1000]
- Steps Per Temp in the range [100, 1000]
- Cooling Fraction set to 0.97

- Initial Temperature set to 1.0
- K Constant set to 0.1

Genetic Algorithm (Agamic)

- Generations in the range [200, 20 000]
- Individuals set to 50
- Mutation Probability set to 0.1

Particle Swarm Optimization

- Iterations in the range [50, 5000]
- Size set to 200

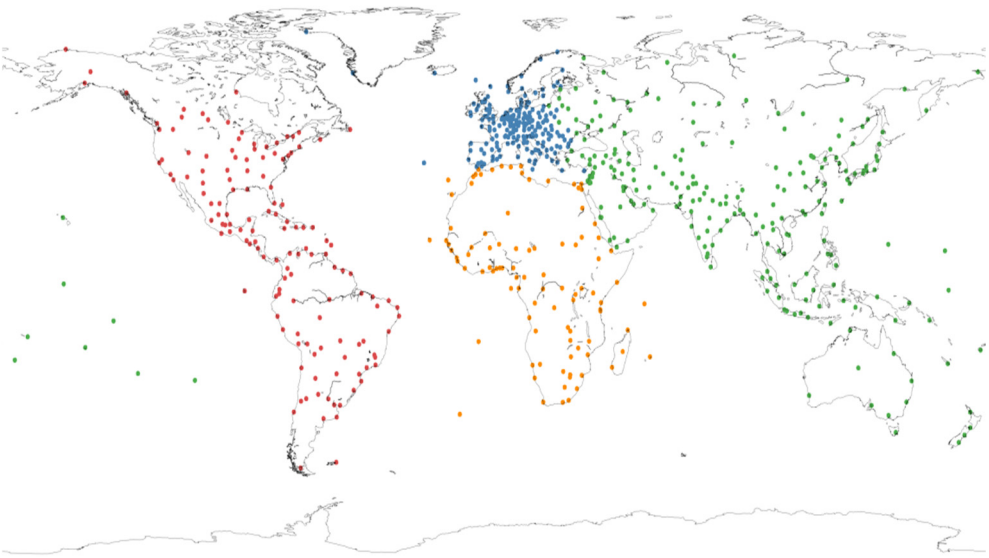

Figure 2. WGS84 projection of the TSP datasets used for evaluation of heuristics: gr96 (yellow), gr137 (red), gr202 (blue), gr229 (green), gr431 (blue + green), and gr666 (all + North and South Poles) [6a-f]. The original data consist of (longitude, latitude)-pairs and the applied distance metric respects the consequences of nodes distribution on a sphere.

3. Results and discussion

Table 2. Comparison of the heuristics' performance

		Africa	America	Europe	Asia--Australia	Europe--Asia--Australia	World
Nodes #		96	137	202	229	431	666
Optimal [km] [8]		55 209	69 853	40 160	134 602	171 414	294 358
Heuristic	**Mode**	**Difference from optimal solution in %***					
Nearest Neighbor	Best	7.7	6.4	11	15	17	16
	Worst	29	40	22	25	30	26
Simulated Annealing	10k	95	160	150	250	420	630
	100k	94	130	73	160	220	340
	1000k	66	110	66	100	140	200
Genetic Algorithm	10k	290	490	400	640	1000	1400
	100k	150	200	190	310	530	870
	1000k	71	100	90	140	260	420
Particle Swarm Optimization	10k	300	480	410	650	900	1300
	100k	260	380	330	550	810	1200
Monte Carlo	10k	440	680	500	810	1200	1500
	100k	430	640	490	790	1200	1500
	1000k	420	630	480	770	1200	1500

*Rounded to 2 significant figures.

Table 2 presents the results in terms of the percentage excess over the optimal solutions. The Nearest Neighbor outclassed the other methods, both in its best and the worst variant (determined by the starting node). With greedy approach – which is problem-specific – the exhaustive search requires evaluation of as many solutions as there are nodes. The stochastic heuristics need to check orders of magnitude more permutations to arrive with an acceptable one.

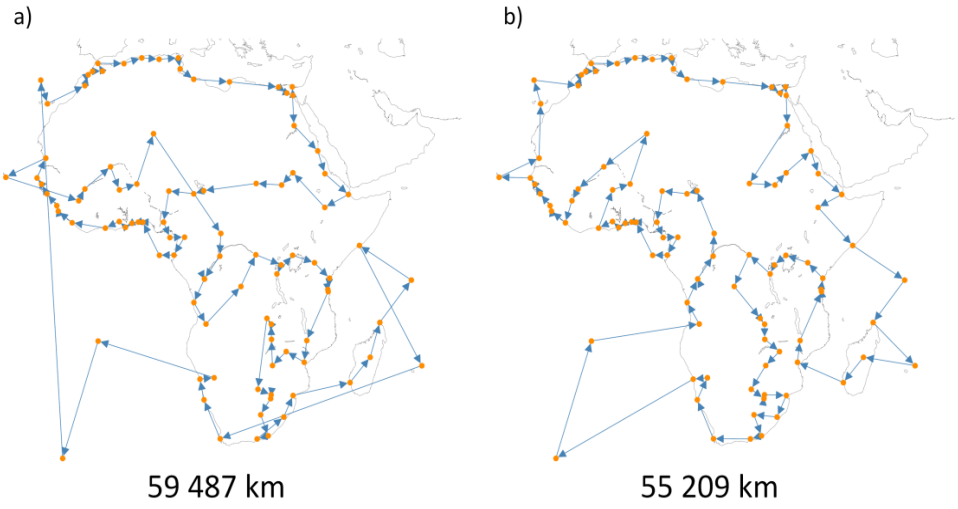

Figure 3. The Nearest Neighbor (a) and optimal (b) tours through 96 African nodes

Figure 3 presents the graphical zoom-in on the best solution of the Nearest Neighbor ("Africa" dataset), contrasted with the optimal one. Both share multiple trails, yet differ by 7.7% in overall score – the former intersects its previous edges, which results in a longer path. The optimal solution delays gratification with temporarily less rewarding choices. The Nearest Neighbor pursues the immediate gain (by design) and regularly backs itself in a corner – the intersections of edges manifest redundant jumps between larger regions. On contrary, the optimal solution systematically exploits clusters of nodes with no returns. The relation between the worst-case Nearest Neighbor and the optimal tours depends on the number of nodes [9]. In practice, with unbiased datasets, the greedy heuristic exceeds the optimal one by 25% [1].

A detailed study of the distribution of the Nearest Neighbor performance, based on the starting point, suggests a relation between anisotropy of the dataset and the plot (Figure 4). The more coherent the search space, the straighter the corresponding curve. Larger datasets tend to smooth the bumps from their subsets, possibly due to their geographical span, which allows for shorter connections from the other side of the sphere. The median instances perform with approximately the mean of the best and the worst cases. Thus, random selection of the starting point – in case

of datasets too big for exhaustive search of the best one – should yield efficiency closer to the mean of the extremes rather than to those extremes.

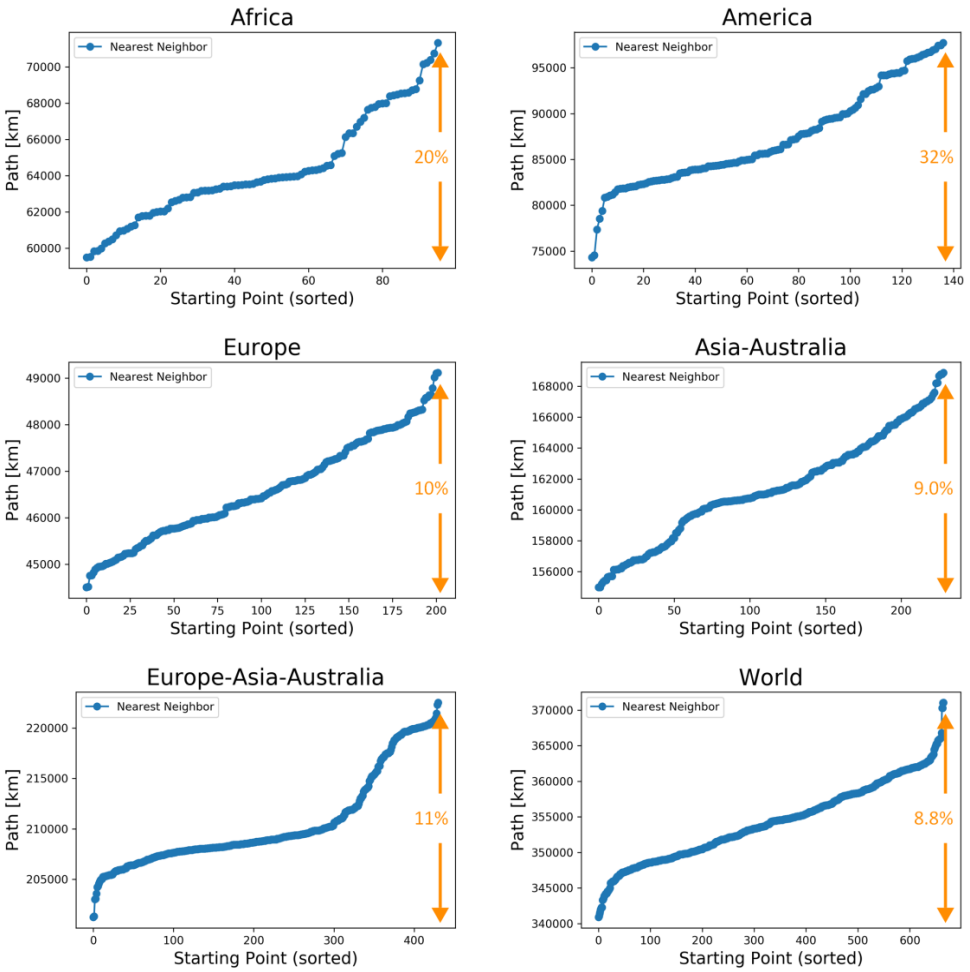

Figure 4. Scores of the Nearest Neighbor by the starting point. Density distribution of nodes affects the shape of plots, whereas their number impacts the relative spread. America punishes nonoptimal choices the most (the steepest slope at the beginning) and Europe the least (the straightest plot).

Figure 5 shows the logarithmic time-lapse of the stochastic heuristics' performance. Unlike the greedy heuristic, Simulated Annealing allows for delayed gratification, but gradually "loses its patience" (the decay-of-probability mechanism).

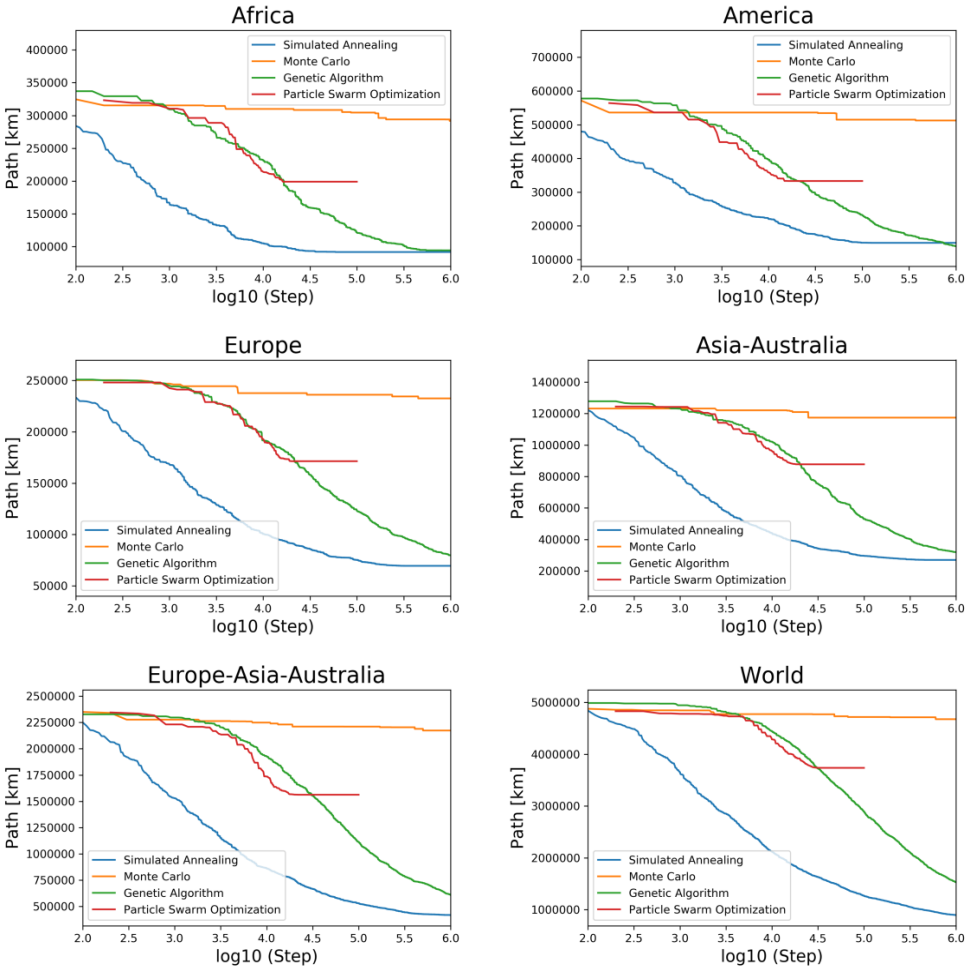

Figure 5. The dynamics of stochastic heuristics in each dataset. Simulated Annealing constantly outran the rest by at least one order of magnitude in terms of speed, but Genetic Algorithm performed similar by score when given time. Particle Swarm Optimization trapped themselves in local optima between 10k and 100k steps, making further 900k steps redundant.

Genetic Algorithm keeps a more stable threshold for temporarily accepting worse solutions; biased reproduction strategies that promote diversity of the gene pool protect against local optima. All but one of the stochastic methods asymptotically approach their performance limit. Only random sampling (Monte Carlo) remains capable of reaching the global optimum – as each solution it evaluates is independent of others and, given enough time, it could find the very best one. However, the practical efficiency of random sampling resembles that of the brute force search.

Appendix contains the graphical representations of the results from Table 2 for stochastic heuristics. We wanted to compare the dynamics of general optimization methods with one another and with reference to a problem-specific technique. We did not aim to solve the traveling salesman problem nor did we pursue a refinement of their implementations. Table 2 and Figure 5 show the uselessness of Monte Carlo in larger datasets as it records no progress within practical period. Our data suggests that a smart approach for solving NP-hard problems, apart from using problem-specific heuristics (when such exist), is to repeatedly simulate annealing and afterwards select the best result. The performance limit of stochastic methods, to which they converge, depends on the instance of simulation. Thus, although Genetic Algorithm will eventually catch up with Simulated Annealing – in fact – the latter will have done multiple laps around the track by then.

4. Conclusions

Problem-specific methods perform the best, however a sufficient number of hyperparameters allows to tune a general technique for a particular task. Heuristics that reduce the number of these knobs, indirectly hide the rest *via* the mechanism of their implementation. This may make them efficient within the problems they were designed for, but restrains the setup for other applications – diminishing their advertised universality. When no apt algorithm exists, the combination of a greedy heuristic and Simulated Annealing – used in parallel to select the better result – can suffice. Most of the time the former should be enough, but in case of biased datasets the stochastic method shall come to rescue. Further studies should examine the relative dynamics of heuristics on other NP-hard problems, preferably the ones with incoherent solution space.

Acknowledgements

This work was not funded by any grant, nor financially supported by any Institution or Company. The authors declare no competing financial interest. All of the authors are students of the Warsaw School of Computer Science and contributed equally to this work.

References

[1] S.S. Skiena, *The Algorithm Design Manual (2nd ed.)*, Springer, 2008.

[2] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by Simulated Annealing, *Science*, 220, 671-680, 1983.

[3] F. Glover, Future paths for integer programming and links to artificial intelligence, *Comput. Oper. Res.*, 13, 533-549, 1986.

[4] D. Whitley, A genetic algorithm tutorial, *Stat. Comput.*, 4, 65-85, 1994.

[5] M.R. Bonyadi, Z. Michalewicz, Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review, *Evol. Comput.*, 25, 1-54, 2017.

[6a] gr96 [Online]. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/gr96.tsp [access 9/11/18]

[6b] gr137 [Online]. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/gr137.tsp [access 9/11/18]

[6c] gr202 [Online]. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/gr202.tsp [access 9/11/18]

[6d] gr229 [Online]. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/gr229.tsp [access 9/11/18]

[6e] gr431 [Online]. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/gr431.tsp [access 9/11/18]

[6f] gr666 [Online]. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/gr666.tsp [access 9/11/18]

[7] Implementations of heuristics [Online]. https://github.com/WojtaX/TSPheuristics [access 26/11/18]

[8] Discrete and Combinatorial Optimization Group, Heidelberg University [Online]. https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/STSP.html [access 9/11/18]

[9] D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis, An Analysis of Several Heuristics for the Traveling Salesman Problem, *SIAM J. Comput.*, 6, 563-581, 1977.

Appendix

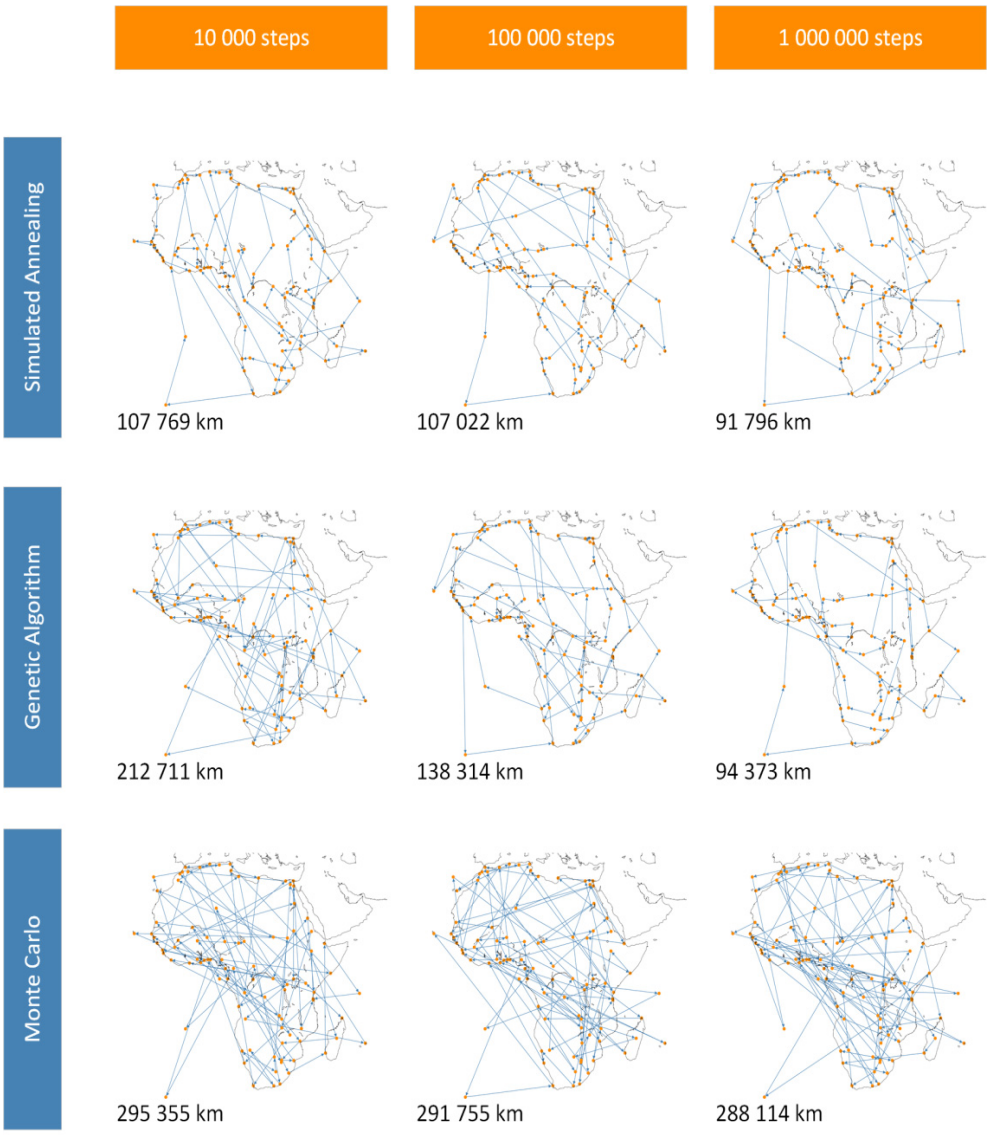

Figure A-1. The tours through 96 nodes of the "Africa" dataset by selected heuristics and the number of evaluated solutions (steps)

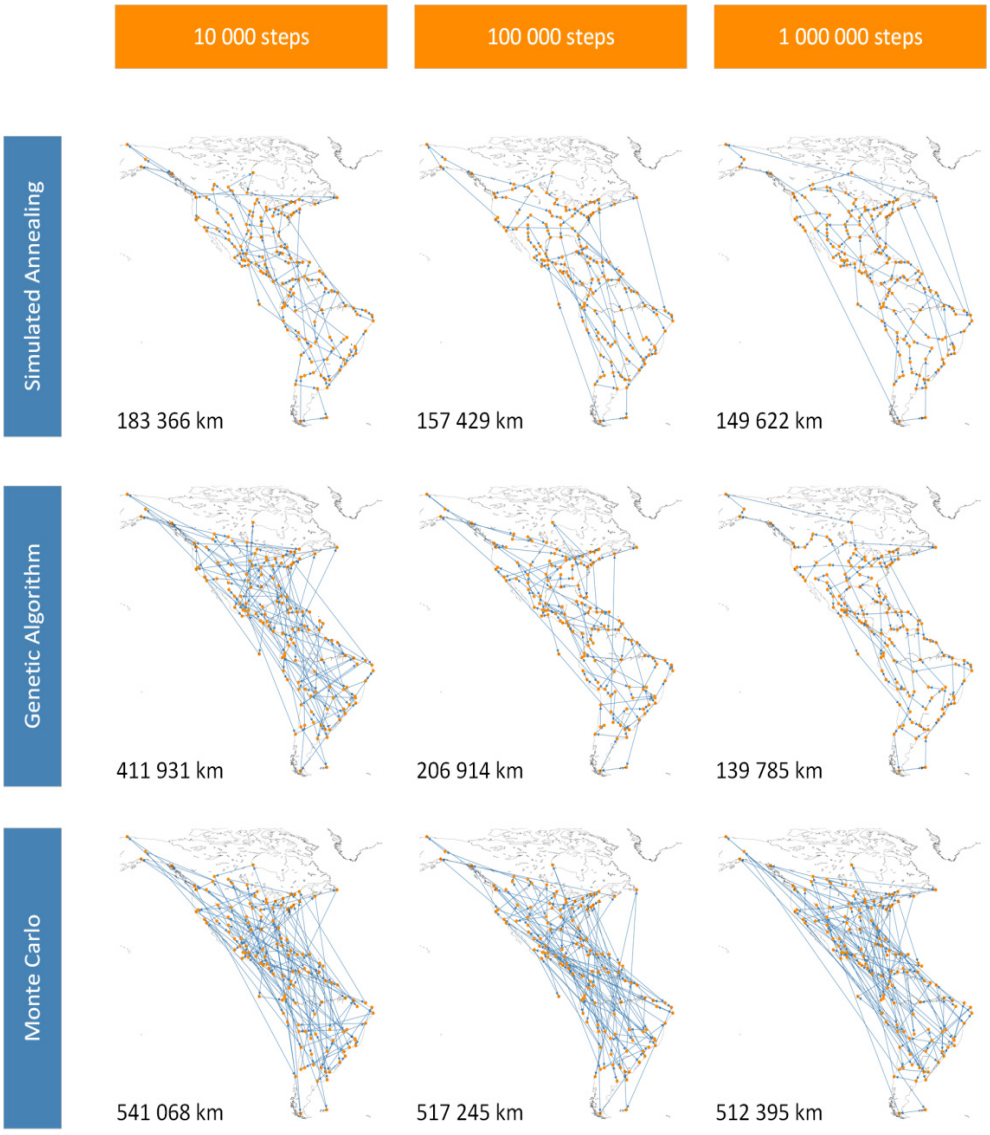

Figure A-2. The tours through 137 nodes of the "America" dataset by selected heuristics and the number of evaluated solutions (steps)

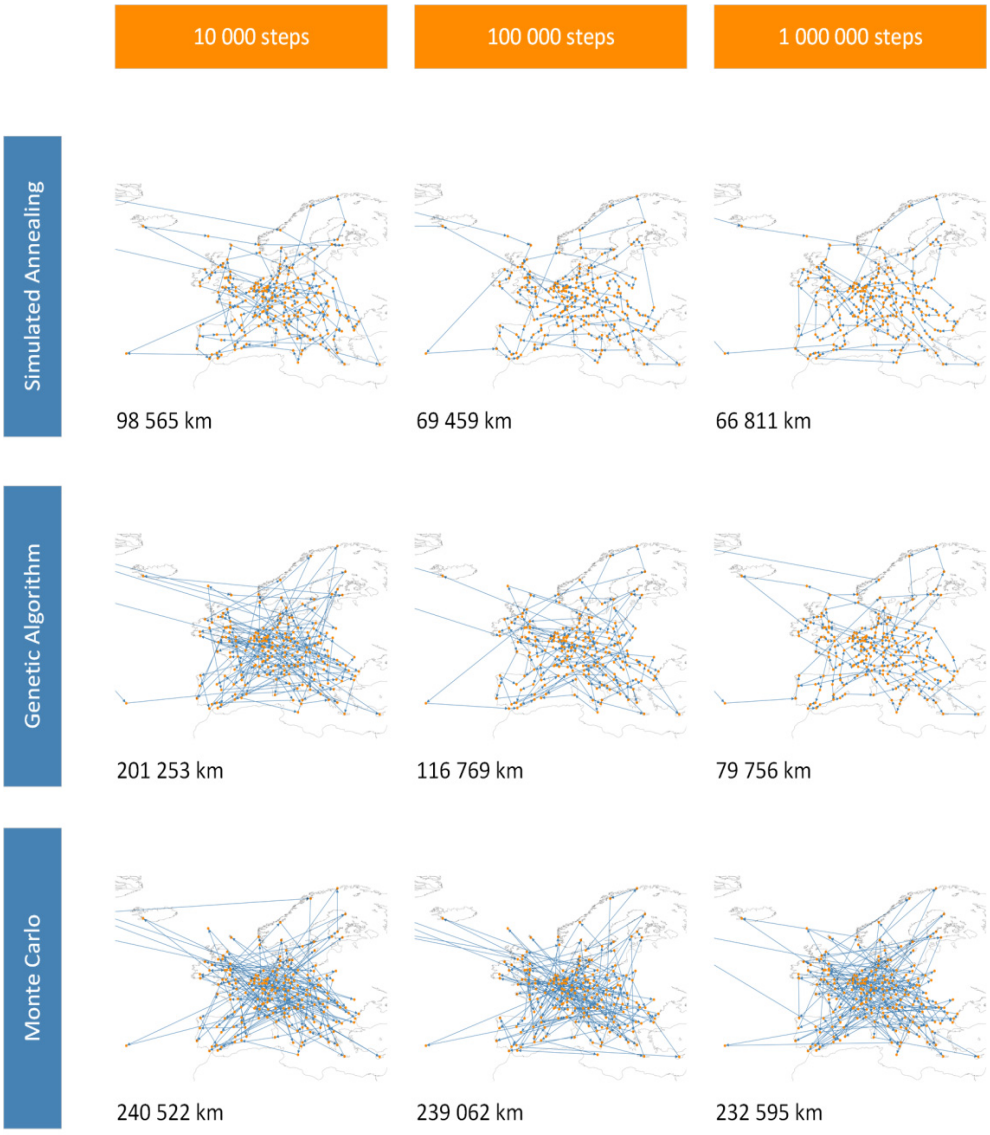

Figure A-3. The tours through 202 nodes of the "Europe" dataset by selected heuristics and the number of evaluated solutions (steps). The upper-left edges connect to Greenland.

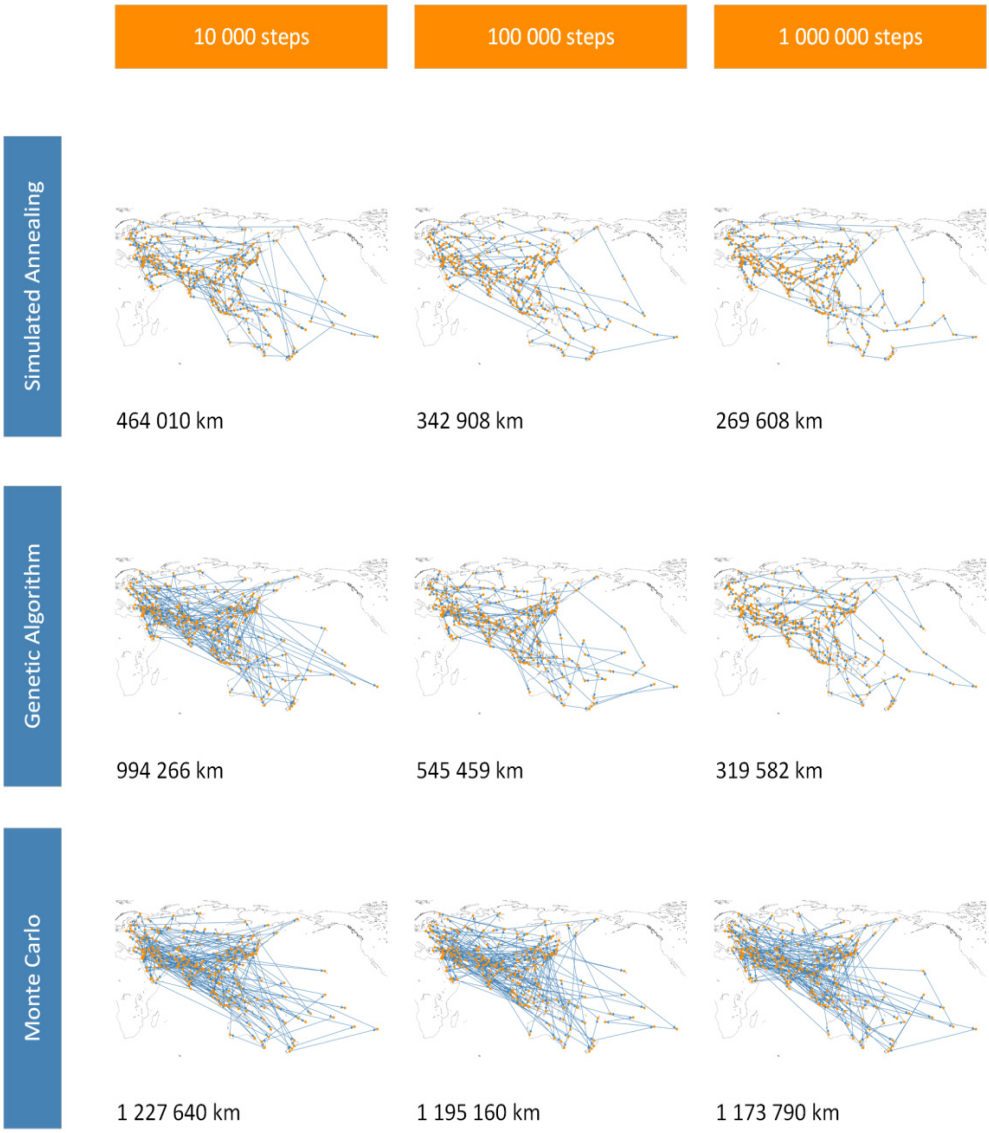

Figure A-4. The tours through 229 nodes of the "Asia-Australia" dataset by selected heuristics and the number of evaluated solutions (steps)

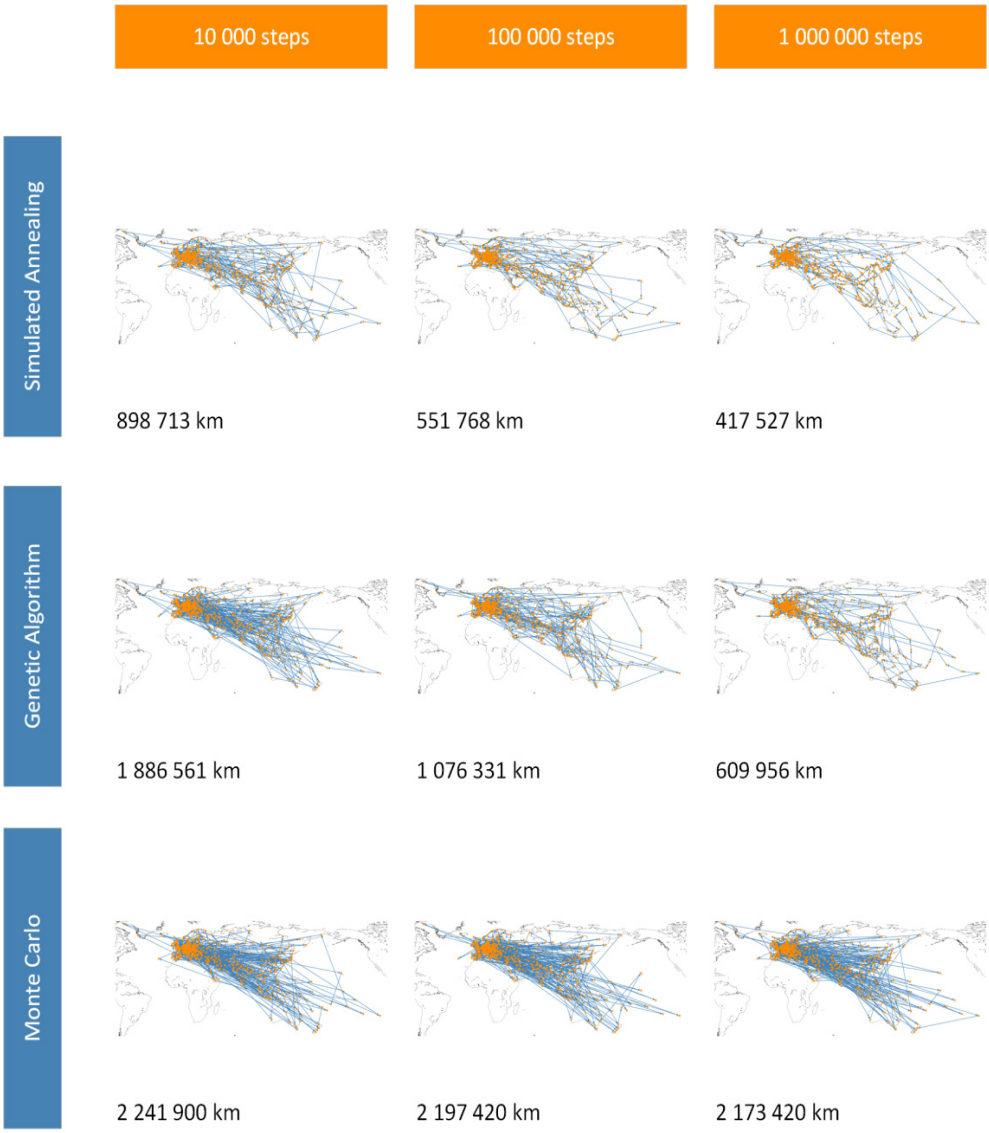

Figure A-5. The tours through 431 nodes of the "Europe-Asia-Australia" dataset by selected heuristics and the number of evaluated solutions (steps). The planar projection of the Globe misrepresents the edges that go through the Pacific.

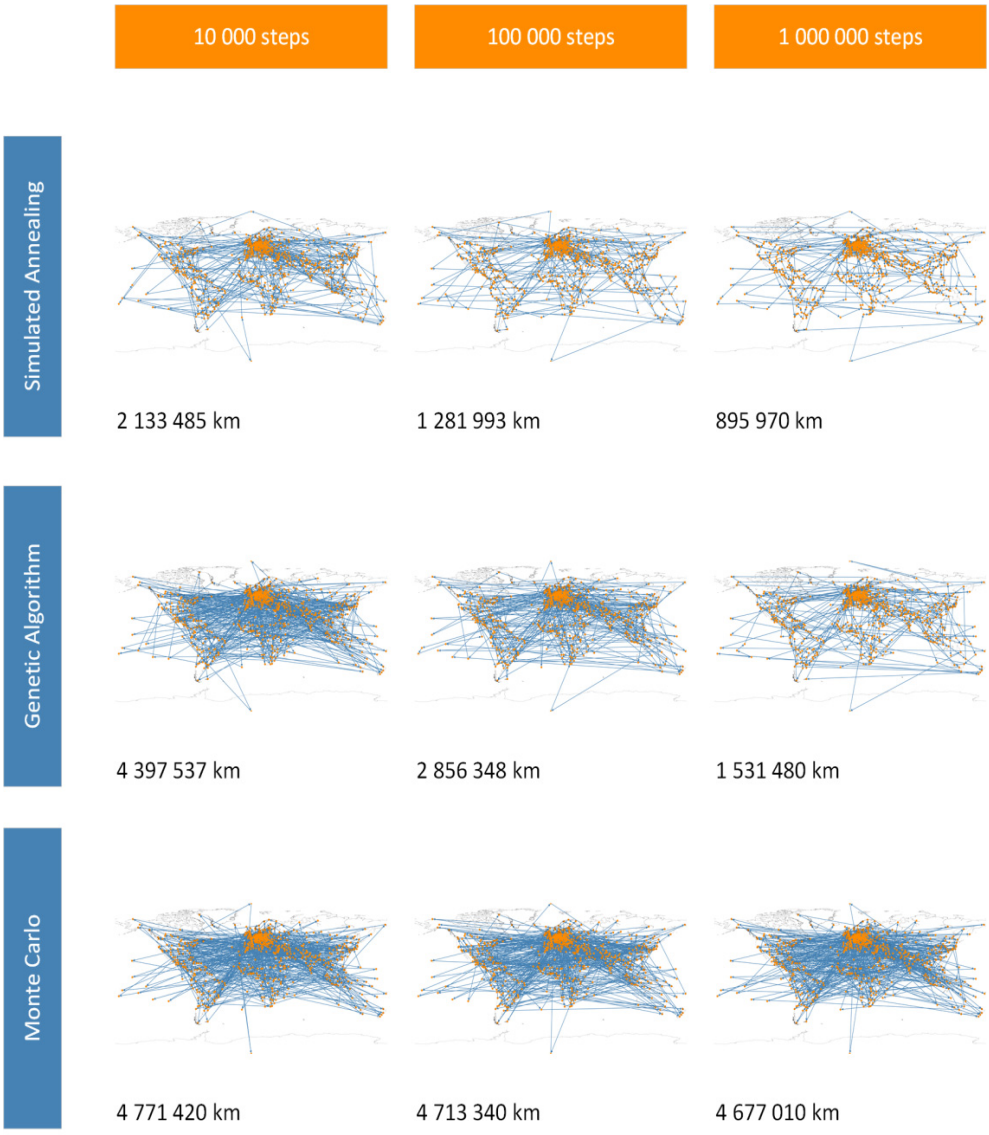

Figure A-6. The tours through 666 nodes of the "World" dataset by selected heuristics and the number of evaluated solutions (steps). The planar projection of the Globe misrepresents the edges that go through the Pacific.