Discrete Tree-seed Algorithm for Solving Symmetric Traveling Salesman Problem

Felix Beese, Jens Lohmann
Universität Hamburg
Dept. Informatik – Knowledge Technology, WTM

http://www.informatik.uni-hamburg.de/WTM/

Outline

1. Motivation and Question
2. Basics and Definition
3. Approach
4. Results
5. Conclusion

Outline

1. Motivation and Question
2. Basics and Definition
3. Approach
4. Results
5. Conclusion

Motivation and Question

- Discrete Tree-Seed Algorithm uses concept of trees and seeds
- Genetic Algorithms use concept of evolution
- Seems like they work in a quite similar way
  - Is the DTSA a Genetic algorithm in disguise?
  - What are the main differences?
- Investigated by the Traveling Salesman Problem

Outline

1. Motivation and Question
2. Basics and Definition
3. Approach
4. Results
5. Conclusion

Traveling Salesman Problem

- Salesman has to visit many cities
- Wants to know the shortest tour
- **Goal:** find shortest Hamiltonian circle
- NP-hard problem
- Heuristic algorithms but also exact ones

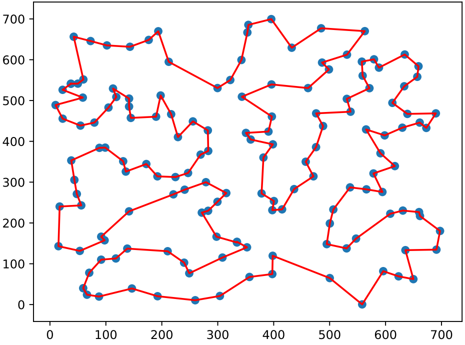


Figure: Example for a solved TSP.

Discrete Tree-Seed Algorithm

- Iterative, heuristic algorithm
- Uses concept of trees and seeds (=possible solutions)
  - get modified by operations
- One tree as nearest neighbor tour
- Optimization of best result by 2-opt algorithm

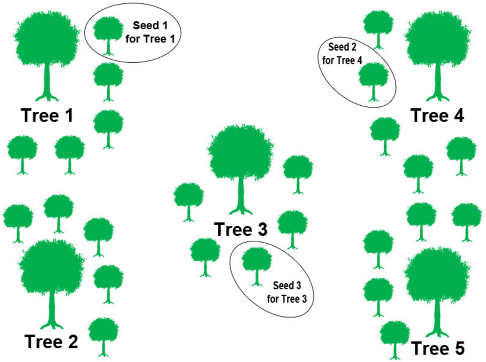


Figure: Figure by Kiran 2015

Operations

Operations used in algorithms:

- swap(i,j) = change cities at position i and j
- shift(i,j) = move path[i+1:j] one to the left and place overwritten city at j
- symmetry(i,j) = change direction of (sub)path[i:j]

Tree=	1	2	3	4	5	6
	swap(2,5)					
Seed=	1	5	3	4	2	6

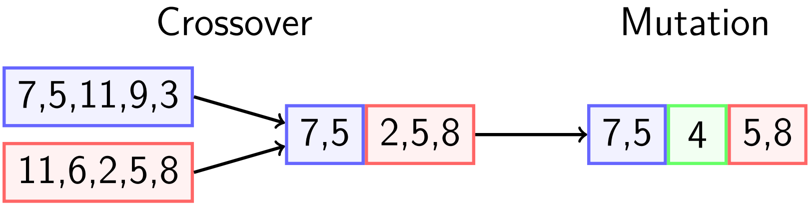
Tree=	1	2	3	4	5	6
	shift(3,5)					
Seed=	1	2	4	5	3	6

Tree=	1	2	3	4	5	6
	symmetry((2,3),(4,5))					
Seed=	1	5	4	3	2	6

Genetic Algorithms

Genetic Algorithms are inspired by the concept of evolution

- Solutions represented as chromosomes (=linear representation)
- Working with generations (batches)
- Mutation and Crossover operations to create new solutions
- Selection step to choose the strongest/best solutions



Outline

1. Motivation and Question
2. Basics and Definition
3. Approach
4. Results
5. Conclusion

Proceeding

1. Implemented two algorithms
   - Discrete Tree-Seed Algorithm
   - Genetic Algorithm
2. Each algorithm run 90 times
   - Six problem instances were taken from tsplib95
     berlin52, st70, kroA100, eil101, ch150, tsp225
3. Compared the results
   - By computing statistical metrics
   - And using different visualization methods
   - **Goal:** Finding similarities and differences of DTSA and GA

DTSA - Main Procedure

- create $k-1$ random trees, one by NN-tour
- repeat until stopping criterion is met
  - loop over all trees/paths (current tree)
    - choose random to create 3 seeds by operations on best tree or current tree
    - create 3 seeds by operations on random tree
    - get best tree out of those 6+1 trees for next iteration
  - k new trees were chosen
  - check if stopping criterion is met
- return overall best tree optimized by 2-opt

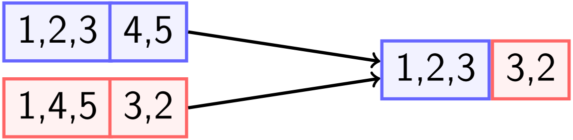
Genetic Algorithm - Main Procedure

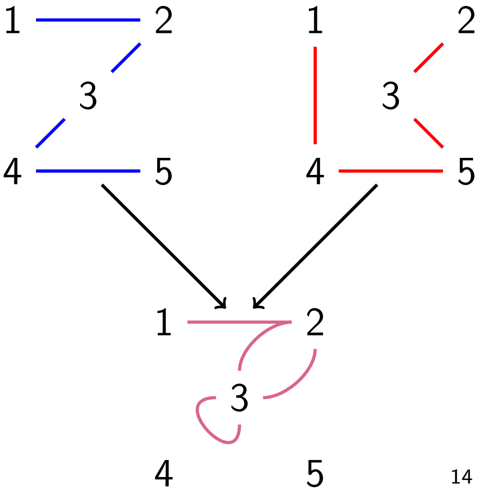
GTSPA = *Genetic Traveling Salesman Problem Algorithm*

- create $k-1$ random trees, one by NN-tour
- repeat until stopping criterion is met
  - loop over all trees/paths (current tree)
    - create 3 seeds by operations on current tree
  - get best k trees out of those $4 \cdot k$ trees for next iteration
  - k new trees were chosen
  - check if stopping criterion is met
- return overall best tree optimized by 2-opt

Crossover for Trees

- Not working with paths every time
- Often gives a useless result
- Not needed for asexual reproduction





Genetic Algorithm - why three versions?

- Three different versions were tested
- They differ in their order of mutation and selection
  - GTSPA-SM
  - GTSPA-MS
  - GTSPA-SMS
- The same effects could be achieved by using different population sizes instead

Outline

1. Motivation and Question
2. Basics and Definition
3. Approach
4. Results
5. Conclusion

Path length over steps

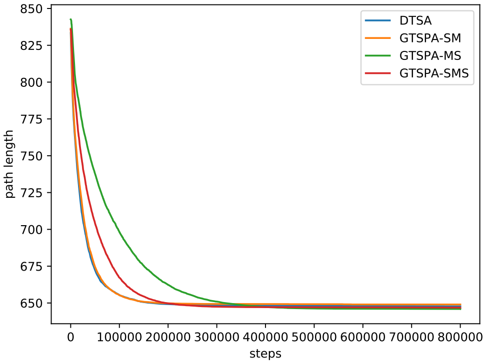


Figure: eil101

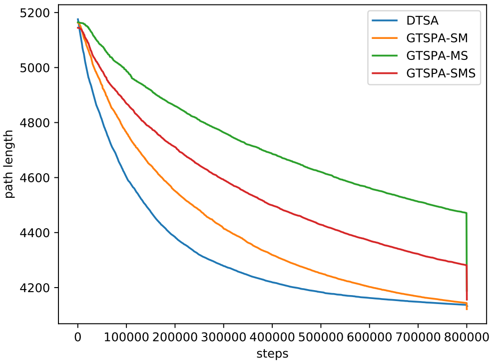


Figure: tsp225

Final path lengths (berlin52 & tsp225)

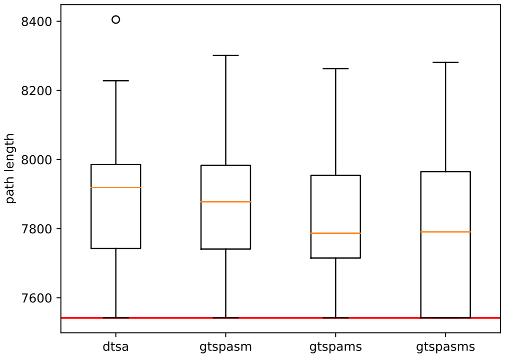


Figure: berlin52

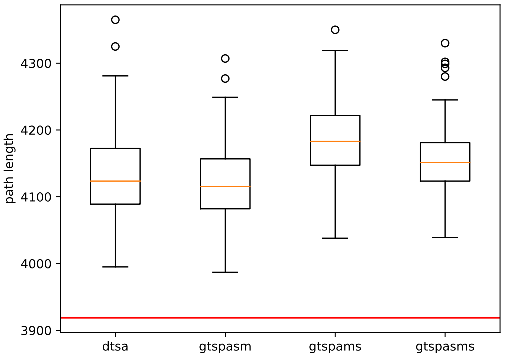
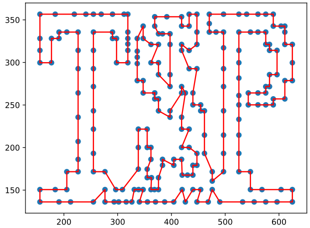
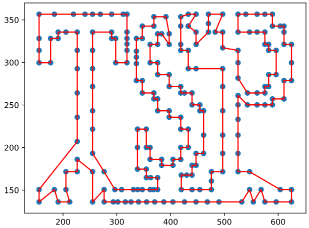


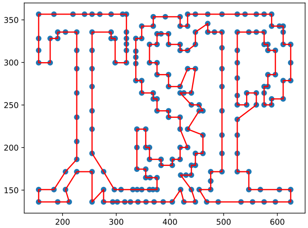
Figure: tsp225



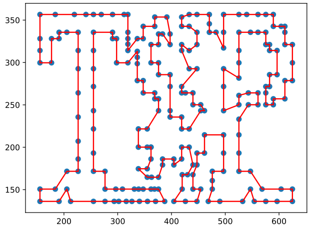
DTSA (3995)



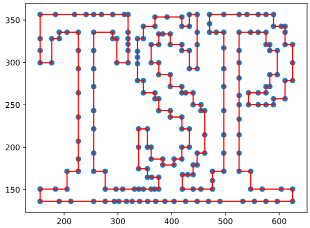
GTSPA-MS (4038)



GTSPA-SMS (4039)



GTSPA-SM (3987)



Optimal (3919)

Outline

1. Motivation and Question
2. Basics and Definition
3. Approach
4. Results
5. Conclusion

Comparison of DTSA and GA

Similarities:

- Concept of evolution with generations, mutations and, selection
- Same operations (swap, shift, symmetry)
- Give results of same magnitude

Differences:

- DTSA has different groups which evolve almost independently from each other
- DTSA converges a little bit faster
- DTSA gives a little worser overall results

Open questions

What can be done now to explore further:

- Test statistical parameters for significance (t-Test, ...)
- Try different parameters and investigate performance
- Implement a classical crossover operation
- Have a look at asymmetric TSP

The End

"The DTSA is a Genetic Algorithm with small adjustments."

Thank you for your attention.

Any question?

Literature:

- Ahmet Cevahir Cinar , Sedat Korkmaz, and Mustafa Servet Kiran. A discrete tree-seed algorithm for solving traveling salesman problem. *In: Engineering Science and Technology*, 2020

Final path lengths

dataset	algorithm	mean	std. dev.	best	worst	RE(%)
berlin52	DTSA	7868.18	185.5	**7542**	**8405**	4.32
	GTSPA-SM	7867.68	193.33	**7542**	8301	4.32
	GTSPA-MS	**7790.32**	175.78	**7542**	8263	3.29
	GTSPA-SMS	7798.98	192.88	**7542**	8281	3.41
st70	DTSA	700.57	16.09	683	738	3.79
	GTSPA-SM	698.26	15.3	**680**	**747**	3.45
	GTSPA-MS	**695.41**	13.09	682	737	3.02
	GTSPA-SMS	695.47	12.2	682	741	3.03

Final path lengths

dataset	algorithm	mean	std. dev.	best	worst	RE(%)
kroA100	DTSA	21831.69	382.83	**21282**	23009	2.58
	GTSPA-SM	21754.38	355.07	**21282**	23136	2.22
	GTSPA-MS	**21588.57**	348.82	**21282**	23096	1.44
	GTSPA-SMS	21719.31	351.1	**21282**	**23224**	2.05
eil101	DTSA	647.76	8.45	**629**	**671**	2.98
	GTSPA-SM	648.99	8.66	630	**671**	3.18
	GTSPA-MS	**645.92**	7.22	631	665	2.69
	GTSPA-SMS	646.9	8.26	630	669	2.85

Final path lengths

dataset	algorithm	mean	std. dev.	best	worst	RE(%)
ch150	DTSA	6688.03	85.32	6549	**6922**	2.45
	GTSPA-SM	6683.87	88.31	6549	6898	2.39
	GTSPA-MS	6665.89	86.8	6552	6913	2.11
	GTSPA-SMS	**6661.59**	77.5	**6544**	6839	2.05
tsp225	DTSA	4132.73	64.11	3995	**4365**	5.45
	GTSPA-SM	**4121.48**	57.02	**3987**	4307	5.17
	GTSPA-MS	4188.68	55.87	4038	4350	6.88
	GTSPA-SMS	4156.21	54.31	4039	4330	6.05

Final path lengths (berlin52 & st70)

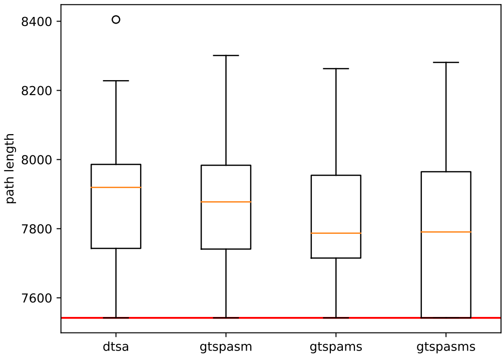


Figure: berlin52

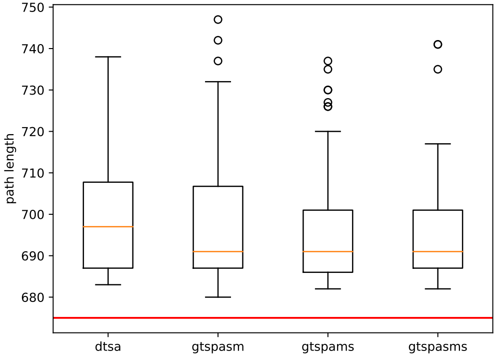


Figure: st70

Final path lengths (kroA100 & eil101)

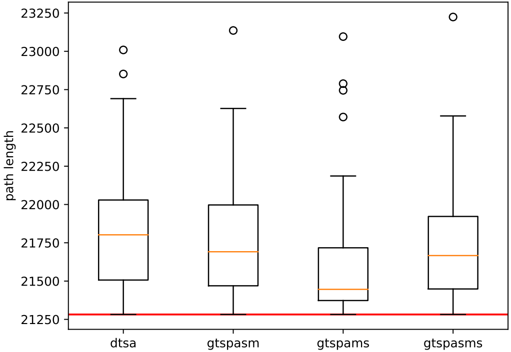


Figure: kroA100

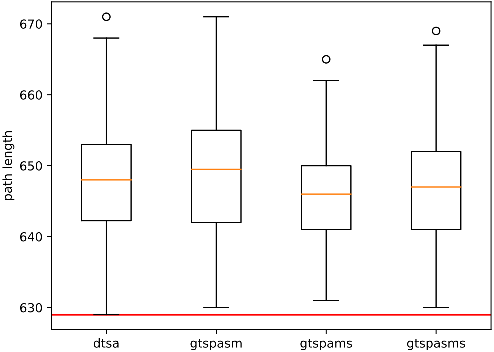


Figure: eil101

Final path lengths (ch150 & tsp225)

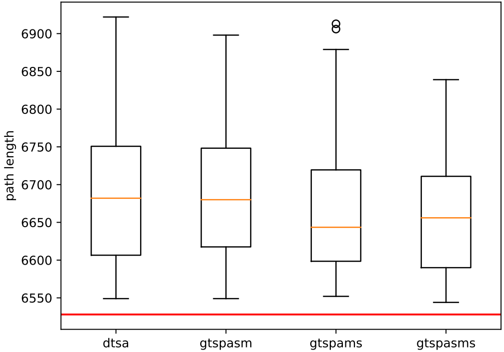


Figure: ch150

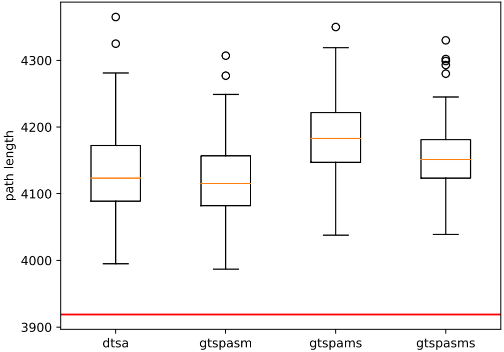
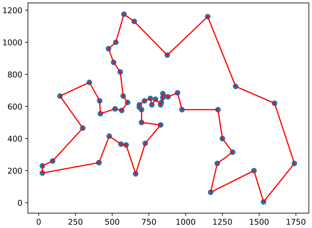
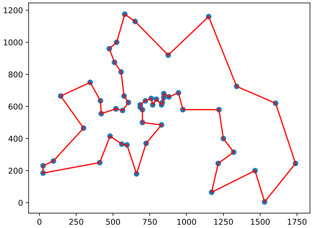


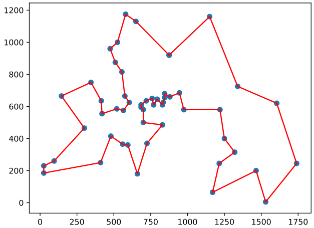
Figure: tsp225



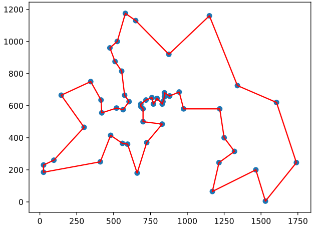
DTSA



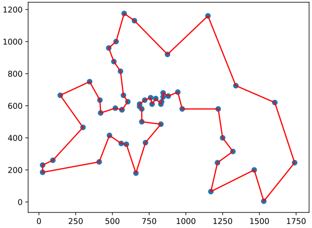
GTSPA-MS



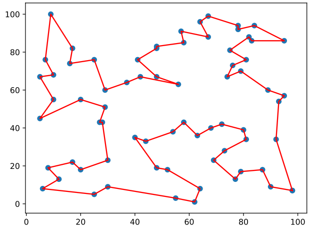
GTSPA-SMS



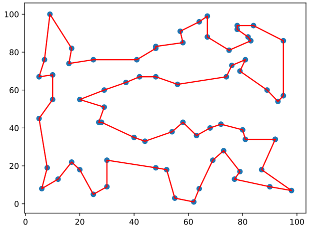
GTSPA-SM



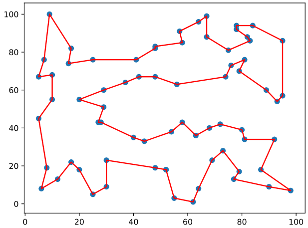
Optimal



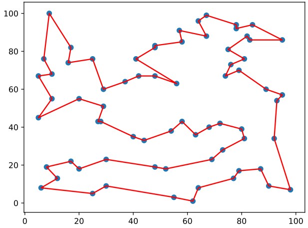
DTSA



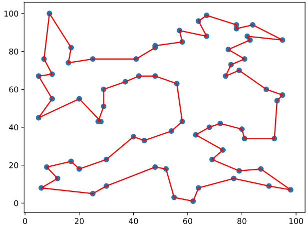
GTSPA-MS



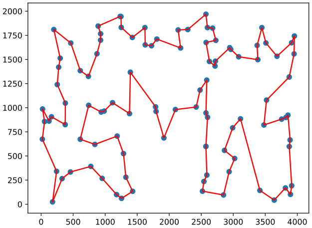
GTSPA-SMS



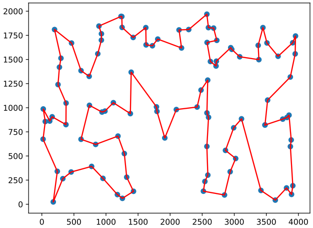
GTSPA-SM



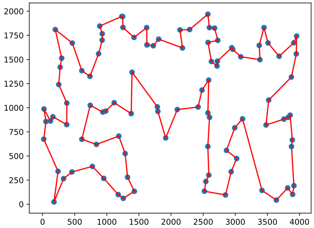
Optimal



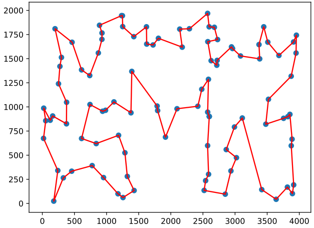
DTSA



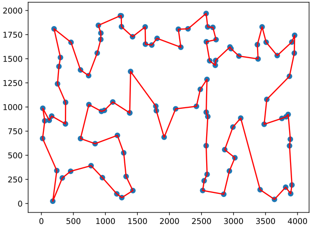
GTSPA-MS



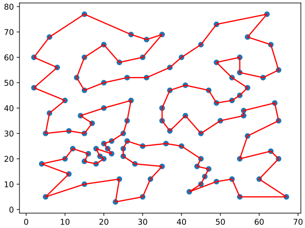
GTSPA-SMS



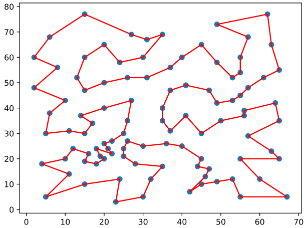
GTSPA-SM



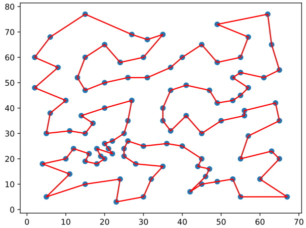
Optimal



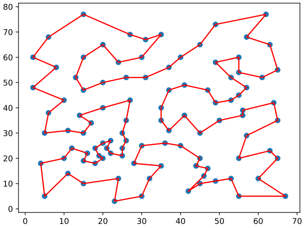
DTSA



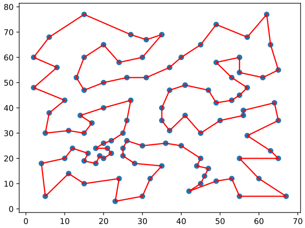
GTSPA-MS



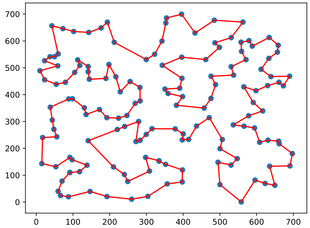
GTSPA-SMS



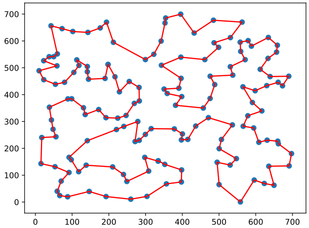
GTSPA-SM



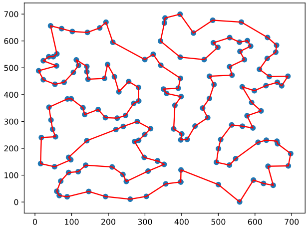
Optimal



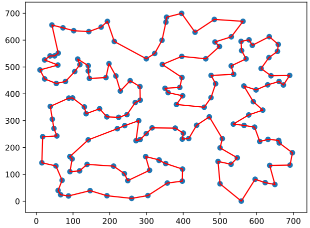
DTSA



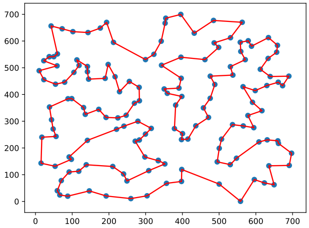
GTSPA-MS



GTSPA-SMS



GTSPA-SM



Optimal