The Generalized Traveling Salesman Problem: A New Genetic Algorithm Approach

by

John Silberholz, University of Maryland

Bruce Golden, University of Maryland

Presented at INFORMS 2007

Coral Gables, January 2007

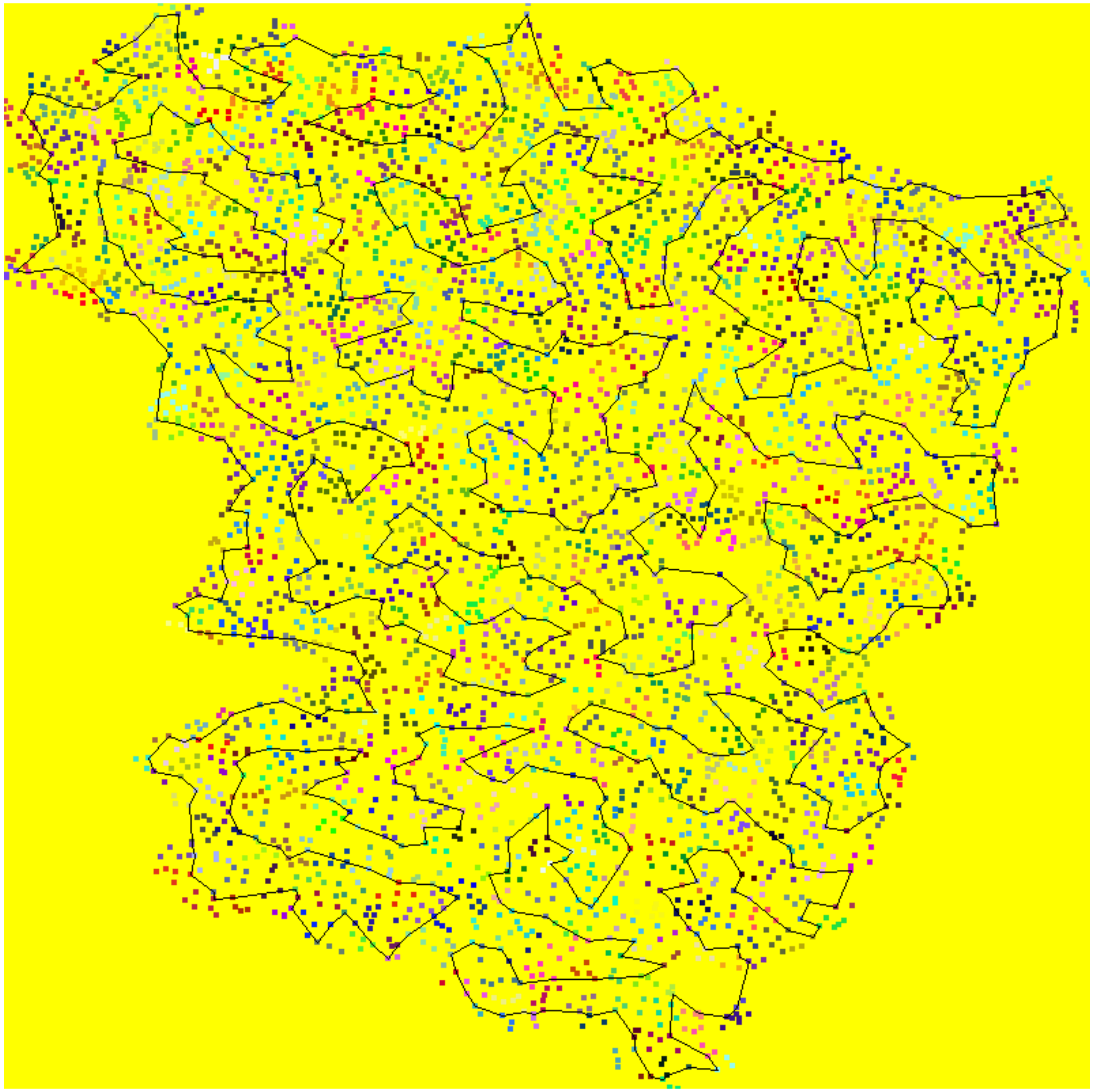
Outline of Lecture

- The generalized traveling salesman problem (GTSP)
  - Formulation
  - Previously published heuristics
- The mrOX genetic algorithm
  - Rotational component
  - Population isolation mechanism
- Computational Results
- Conclusions

The GTSP – Formulation

- Traveling salesman problem visits every city in tour
- Instead of visiting every node, generalized traveling salesman problem (GTSP) breaks nodes into clusters and a tour visits exactly one node in each cluster
  - Other formulations visit ≥ 1 nodes (allowing shorter pathways in datasets that don't conform to Δ inequality)
- Many applications
  - Airplane and vehicle routing
  - Rural mail delivery
  - Warehouse order picking
  - Computer file sequencing

The GTSP – Example



- Different colors indicate different clusters

The mrOX Genetic Algorithm

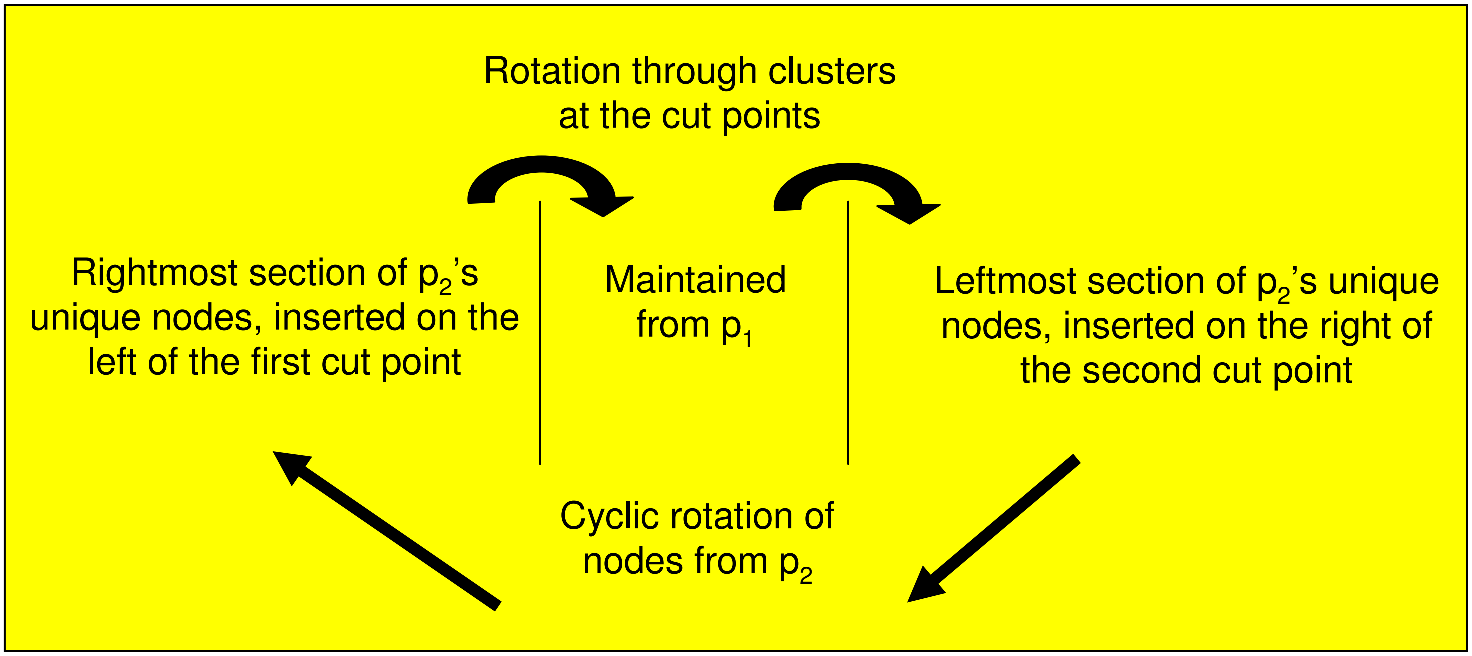
- Heuristics needed to make solutions to datasets like the previous one possible with standard computational resources
- The mrOX GA was developed to produce better solution qualities than previously published heuristics
- Heuristic must still produce reasonable runtimes to be effective

The mrOX GA implementation

- Path representation
- Standard genetic algorithm
  - 50 chromosomes in population
  - 30 new chromosomes each generation through crossover
  - 20 replicated chromosomes each generation
  - Local improvement through *2-opt* and *swap*
  - Isolated population model

Crossover

- A novel crossover (modified rotational ordered crossover, or mrOX) was developed and implemented, using the idea of rotation through adjacent nodes and rotation through node clusters



mrOX crossover – An example

- 12-city dataset for this example found in paper

Node	1	2	3	4	5	6	7	8	9	10	11	12
Cluster	1		2		3		4		5		6	

- Consider crossover between two parents –
  - p_1 = { 12 1 | 3 10 | 6 8 }
  - p_2 = { 2 4 | 6 8 | 10 12 }
- Retain part between cutpoints from p_1
  - O = { x x | 3 10 | x x }

mrOX crossover – An example

- A reminder – parents are
  - p_1 = { 12 1 | 3 10 | 6 8 }
  - p_2 = { 2 4 | 6 8 | 10 12 }
- Retained nodes 3 and 10 are from clusters 2 and 5
- Thus, nodes from other clusters are taken in order from p_2 –
  - 2, 6, 8, 12 (clusters 1, 2, 4, and 6)
- From rotational aspect, nodes will be entered in orders:
  - (2, 6, 8, 12), (6, 8, 12, 2), (8, 12, 2, 6), (12, 2, 6, 8)
- If we allow reversals, we'll also consider orderings:
  - (12, 8, 6, 2), (8, 6, 2, 12), (6, 2, 12, 8), (2, 12, 8, 6)

mrOX crossover – An example

- Last modification involves rotation through clusters adjoining cutpoint
  - Thus, if we insert (2, 6, 8, 12), we need to consider –
    - { 8 12 | 3 10 | 2 6 }
    - { 8 12 | 3 10 | 1 6 }
    - { 8 11 | 3 10 | 2 6 }
    - { 8 11 | 3 10 | 1 6 }
- This increases survival rate of new tour orientation

mrOX crossover – An example

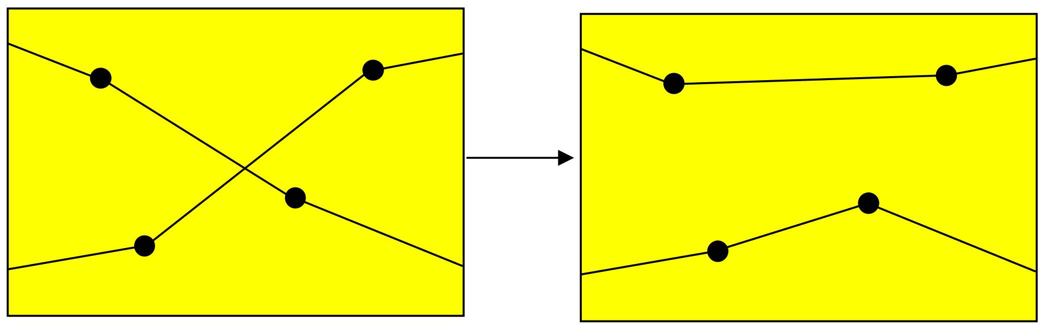
- The final result of this example crossover, with associated costs –

Rotational Crossover							Reverse Rotational Crossover						
pos1	pos2	pos3	pos4	pos5	pos6	cost	pos1	pos2	pos3	pos4	pos5	pos6	cost
6	8	3	10	12	2	317	2	12	3	10	8	6	379
6	8	3	10	11	2	293	2	12	3	10	7	6	362
6	7	3	10	12	2	306	2	11	3	10	8	6	296
6	7	3	10	11	2	282	2	11	3	10	7	6	279
8	12	3	10	2	6	346	12	8	3	10	6	2	408
8	12	3	10	1	6	276	12	8	3	10	5	2	362
8	11	3	10	2	6	315	12	7	3	10	6	2	308
8	11	3	10	1	6	245	12	7	3	10	5	2	262
12	2	3	10	6	8	326	8	6	3	10	2	12	266
12	2	3	10	5	8	318	**8**	**6**	**3**	**10**	**1**	**12**	**215**
12	1	3	10	6	8	297	8	5	3	10	2	12	331
12	1	3	10	5	8	289	8	5	3	10	1	12	280
2	6	3	10	8	12	350	6	2	3	10	12	8	286
2	6	3	10	7	12	244	6	2	3	10	11	8	314
2	5	3	10	8	12	377	6	1	3	10	12	8	238
2	5	3	10	7	12	271	6	1	3	10	11	8	266

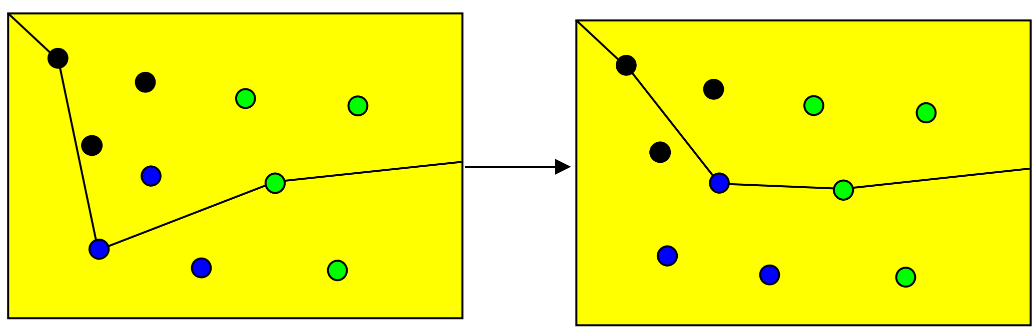
- { 8 6 3 10 1 12 } is final child

Local Optimization

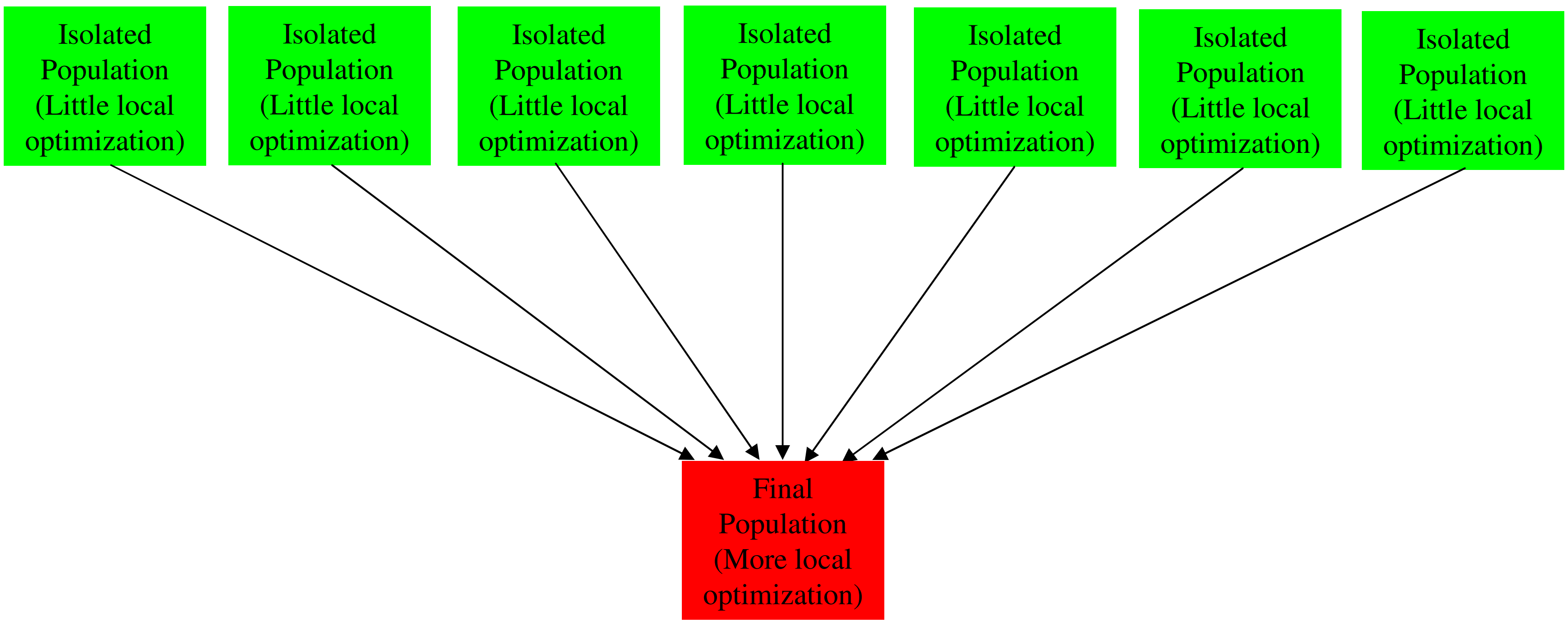
- One method was the *2-opt*:



- Another method was the *swap*:



Isolated population method



- Combination done with greedy algorithm
- Combination after 10 static generations
- Termination after 150 static generations

Computational Experiments

- Computational experiments run on TSPLib datasets with Fishetti et al.'s clusterization method
- Heuristic programmed in Java and run on a computer with a 3.0 GHz processor and 1 GB RAM
- Compared to Snyder and Daskin's GA model for same problem
  - Easy to code
  - Also a GA
  - Best results to date
- S+D's GA was coded so comparisons could be made and larger datasets could be tested

Computational Experiments

- mrOX GA compared with other published heuristics

Heuristic	Average pct. above optimal	Average runtime (sec.)
mrOX GA	0.03	2.69
S+D GA	0.11	1.77
GI^3	0.98	83.09
NN	1.48	171.56
FST-Lagr	0.46	16.44
FST-Root	0.11	964.79
B&C	0.00	3356.47

Computational Experiments

- On larger datasets (from 493 to 1084 nodes), comparison could only be made with S+D's GA
- No optimum values available
- Computational tests featured 150-gen. mrOX GA, 50-gen. mrOX GA, and S+D GA in 13 large datasets
  - Best solution quality: 11 for 150-gen., 2 for 50-gen., 0 for S+D GA
  - Best runtime: 9 for 50-gen., 4 for S+D GA, 0 for 150-gen.
- For 150-gen. model, S+D GA has runtime advantage and mrOX GA has solution quality advantage
- With 50-gen. model, mrOX GA shows a 47.52% decrease in runtime and a 0.56% decrease in solution quality, making it comparably superior

Computational Experiments

- Isolated population method tested with different numbers of isolated populations
- 7-population model has 0.04% better solution quality than 1-population model, with a 3.2% longer runtime
- 20-population model has 0.006% better solution quality than 7-population model, with a 10.35% longer runtime
- Diminishing returns clearly seen; a small number of isolated populations seems appropriate

Computational Experiments

- mrOX crossover tested against the OX crossover
- A 0.18% increase in solution quality was measured
- A 2.59% faster runtime was also measured
- mrOX appears to be superior to the OX crossover, and should be used in its place

Conclusions

- We have developed an effective heuristic to approach the GTSP
- The mrOX can be applied to other transportation problems
- The isolated population mechanism with varying location optimization levels can be applied to many GAs in a variety of areas