Solving the Traveling Salesman Problem with Demon Algorithms and Variants

by

Joshua Pepper, Systems Planning and Analysis

Bruce Golden, University of Maryland

Edward Wasil, American University

Presented at INFORMS Salt Lake City, May 2000

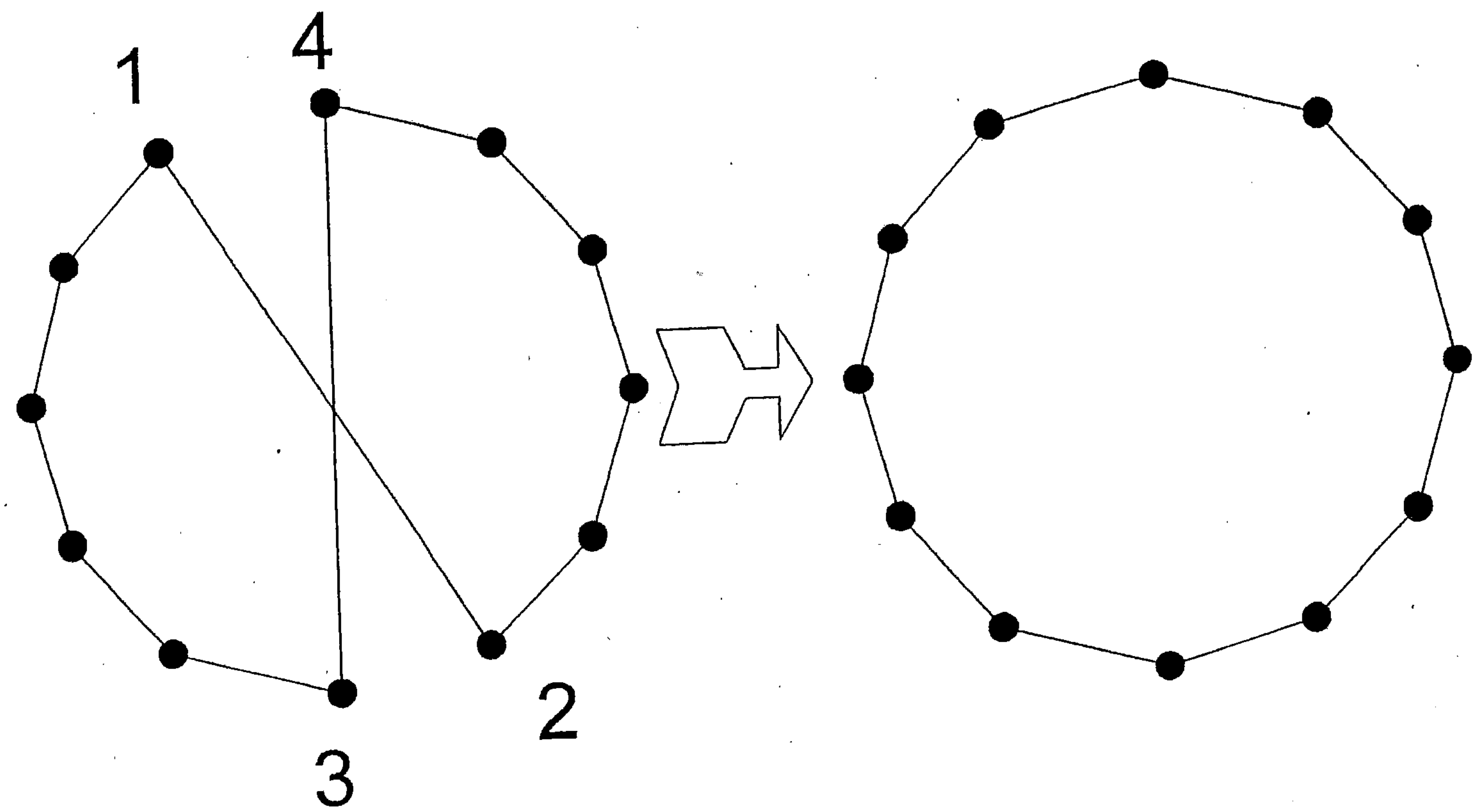
Focus of Paper

- Review several "old" metaheuristics for solving the TSP
  - simulated annealing (SA)
  - threshold accepting (TA)
  - record-to-record travel (RRT)
- Introduce the demon algorithm (DA) and a number of variants
- Perform a computational study
- Present conclusions

Metaheuristics

- Work in this direction began with simulated annealing (1983)
- Metaheuristics for the TSP are often based on the concept of arc exchanges (dating back to the two-opt)
- The starting point was to realize that occasionally **accepting** arc exchanges that increased tour length was okay

Two-opt Exchange



Simulated Annealing

- Generate an initial tour and set T (temperature)
- Repeat until stopping condition:
  - Generate a new tour and calculate ΔE (change in energy)
  - If $\Delta E \leq 0$, accept new tour
  - Else, if $rand(0,1) < exp(-\Delta E/T)$, accept new tour
  - Else, reject new tour
  - Implement annealing schedule ($T=a*T$)
- The choice of T and a are essential

Deterministic Annealing

- In both TA and RRT, arc exchanges are accepted if ΔE is less than some bound

- In TA, we set an initial threshold, T, and anneal the threshold as the algorithm progresses ($T=a*T$)
    - If $\Delta E < T$, accept new solution

- In RRT, we set the record, R, as the length of the best tour found and we set the deviation, D
    - If $E < R + D$, accept new solution

Demon Algorithms

- Wood and Downs developed several demon algorithms for solving the TSP
- In DA, the demon acts as a creditor
- The demon begins with credit = $D > 0$
- Consider an arc exchange
- If $\Delta E < D$, accept new tour and $D = D - \Delta E$
- Arc exchanges with $\Delta E < 0$ build credit
- Arc exchanges with $\Delta E > 0$ reduce credit

Demon Algorithms (continued)

- To encourage minimization, Wood and Downs propose two techniques
  - Impose an upper bound on the demon value, restricting the demon value after energy decreasing moves
  - Anneal the demon value

- Wood and Downs also propose a random component
  - The demon value is a normal random variable centered around the demon mean value
  - All changes in tour length impact the demon mean value

Demon Algorithms (continued)

- This leads to four algorithms (Wood and Downs)
  - Bounded demon algorithm (BD)
  - Randomized bounded demon algorithm (RBD)
  - Annealed demon algorithm (AD)
  - Randomized annealed demon algorithm (RAD)

New Demon Algorithms

- Two new techniques come to mind (Pepper et al.)
  - Annealed bounded demon algorithm (ABD)
  - Randomized annealed bounded demon algorithm (RABD)
- The idea is to impose a bound on the demon value (or demon mean value) and anneal that bound in ABD and RABD
- For RAD and RABD, anneal both the bound on the demon mean and the standard deviation. This leads to two additional algorithms, ADH and ABDH

Computational Study

- Eleven algorithms in all
- We selected 29 instances from TSPLIB
- The instances range in size from 105 to 1,432 nodes
- The instances have different structures
- Each algorithm was applied 25 times to each instance from a randomized greedy start
- Best and average performance and running time statistics were gathered

Parameter Settings

- We selected three test instances
    - Stage 1 GA determines a set of parameter values (parameter vector) for each test instance
    - Stage 2 GA combines the parameter vectors, minimizing tour lengths across all test instances simultaneously
- Resulting parameter vector is applied to all 29 instances

Final Values of Parameters

Algorithm	*if*	*sdf*	α	β
SA	0.0035		0.9646	
TA	0.0722		0.9515	
RRT	0.0055			
BD	0.0027			
RBD	0.0030	0.0072		
AD	0.0745		0.5268	
RAD	0.0585	0.0005	0.4698	
ABD	0.0457		0.9716	
RABD	0.0407	0.0008	0.9418	
ADH	0.0489	0.0282	0.4977	0.8460
ABDH	0.0375	0.0303	0.9720	0.8585

Results

Algorithm	NBS	Avg. Perf.	ET
SA	5	3.09	33.3
TA	1	5.75	41.9
RRT	0	6.78	33.8
BD	0	5.40	53.8
RBD	2	7.66	24.7
AD	5	4.49	58.1
RAD	3	4.76	47.8
ABD	6	3.81	35.8
RABD	6	3.24	29.5
ADH	5	4.03	61.8
ABDH	3	3.76	36.6

NBS = # of instances where algorithm generates the best tour
Avg. Performance = average % above optimal solution over 29 instances
ET = total execution time in hours

Conclusions

- First extensive computational study of demon algorithms and variants
- The computational results are mixed
    - ABD and RABD are competitive with SA
    - The other variants are not
- Future work: Do smarter DA variants exist?

Note: ABD = annealed bounded demon algorithm
RABD = randomized annealed bounded demon algorithm