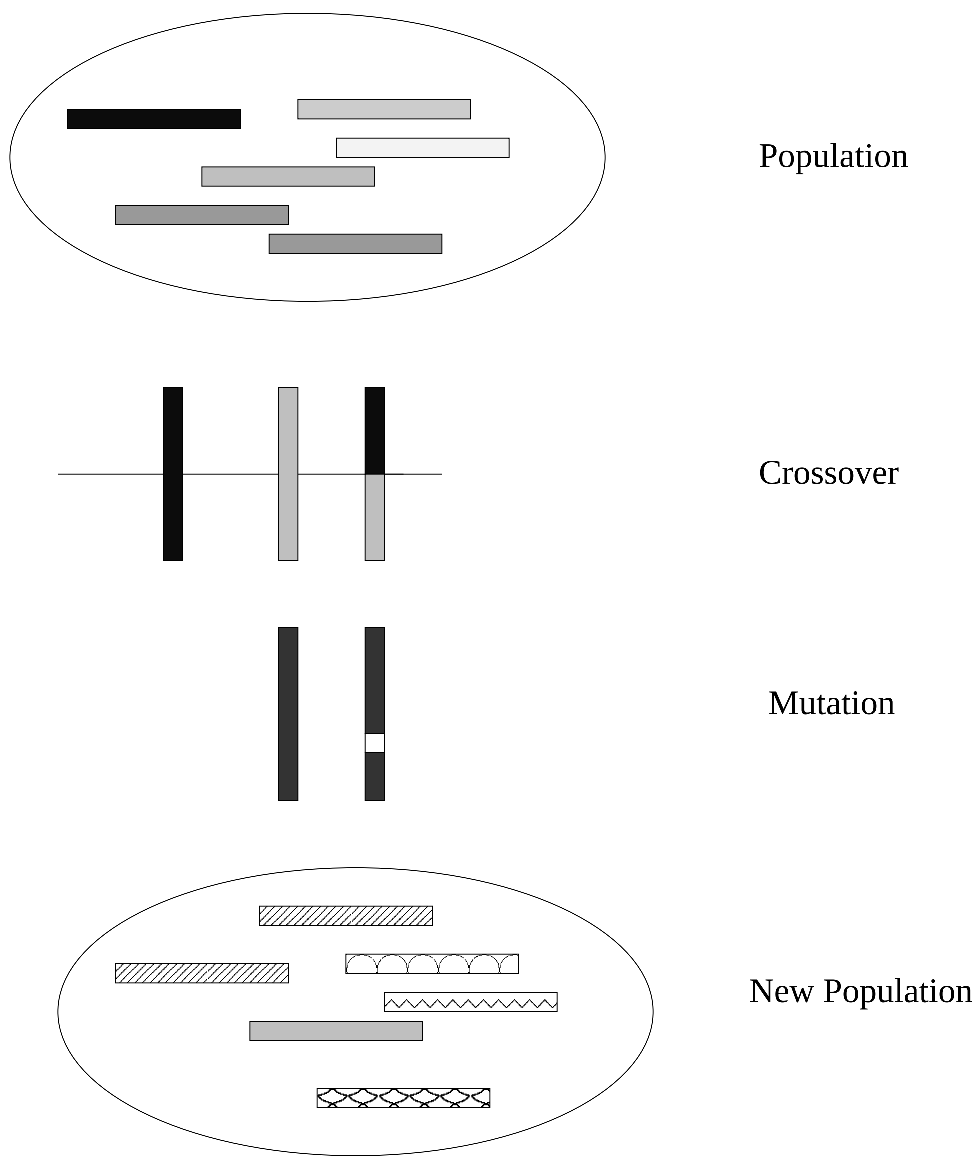
Pitfalls with Adaptive Methods for Combinatorial Optimization: The Traveling Salesman Problem A Case Study

Wolfgang W. Bein
Bradley J. Hendricks
University of Nevada, Las Vegas
Department of Computer Science

June 2002

UNIVERSITY OF NEVADA LAS VEGAS

Genetic Algorithms



Combinatorial Optimization

- Looking for an object from a *finite* set

  permutation

  graph

  set of integers

- $\mathcal{S}$ finite, $c : \mathcal{S} \to \mathbf{R}$,
  find $x^* \in \mathcal{S}$ such that $c(x^*) \leq c(x)$ for all $x \in \mathcal{S}$

- Graph Problems, Sequencing Problems, Matching Problems, Scheduling Problems, Packing Problems...

- Many problems are $\mathcal{NP}$-complete

Traveling Salesman Problem

- Given are n cities $\{$1,2, . . . n $\}$
- Given is a distance matrix $D = (d_{ij})$
- We are interested in finding a permutation π (“the tour”), such that

$$f(t) = \text{dist}(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} \text{dist}(\pi(i), \pi(i+1))$$

is minimized.

Pitfalls

- TSP is PLS-complete [Kentel, 89]
- There is an abundance of local minima [Boese, 96]
- It is not easy to move from one local minimum to another

Crossover does not come naturally

- Crossover should preserve good traits of both parent tours
- Crossover should maintain high diversity in the population
- Crossover must produce two valid tours!

$\rightarrow$ Problematic for sequencing problems

The Cycle Crossover

$\underline{1}$ $\underline{2}$ 5 $\underline{3}$ 4

$\underline{3}$ $\underline{1}$ 4 $\underline{2}$ 5

The first, second, and fourth location contain exactly the same set of cities $\{1, 2, 3\}$.

$\underline{1}$ $\underline{2}$ 4 $\underline{3}$ 5

$\underline{3}$ $\underline{1}$ 5 $\underline{2}$ 4

The Ordered Crossover

Parent 1 Parent 2	184 637 |25 352718|64	random slice 637 add underlined
Child	218637|45	child is a valid tour

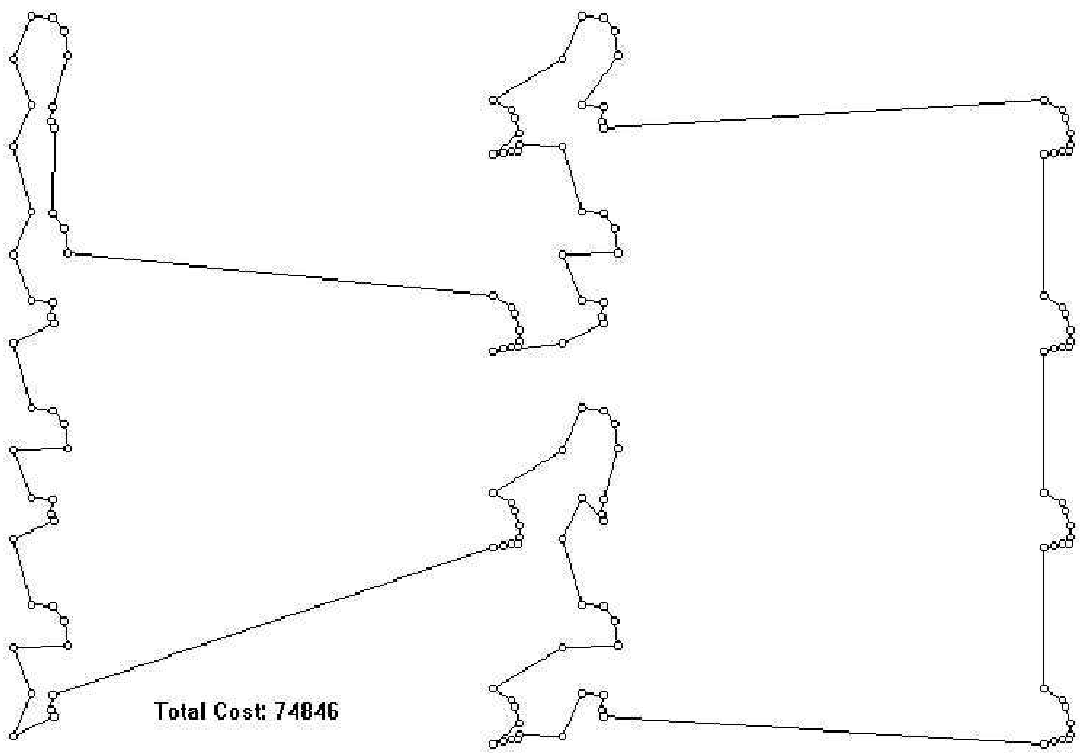
The TSPView Package

- Implementation under MIT's GALib [Wall, 1996]

- Interface under Visual C++

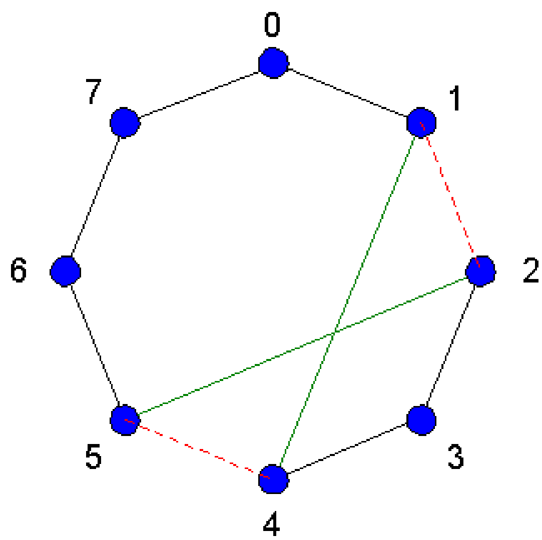
- Available at
  `www.cs.unlv.edu/~bein/TSPView.html`

A Close to Optimal Solution for a 152 City Problem



Local Search

Idea: Define an appropriate neighborhood



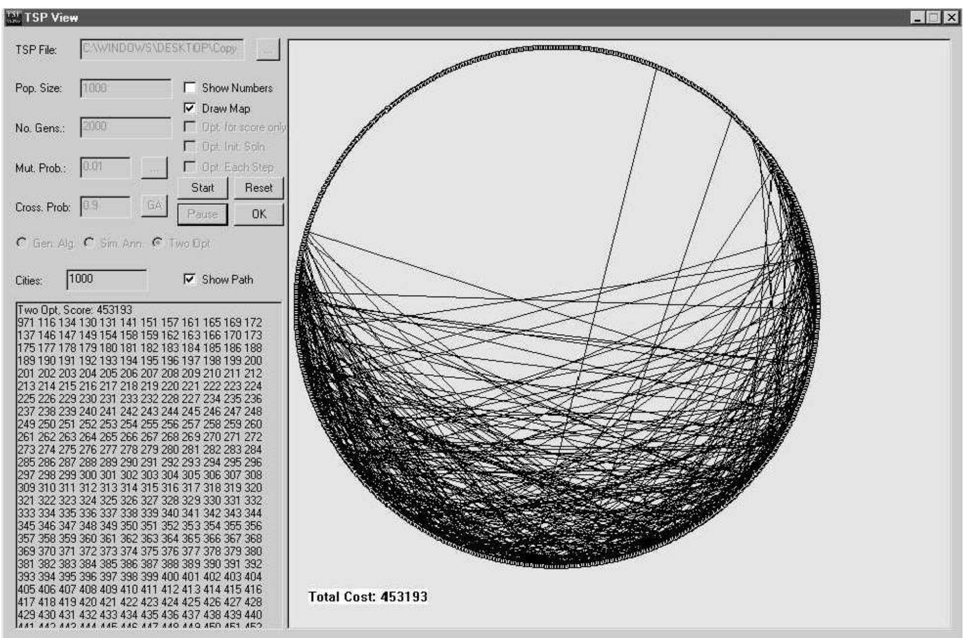
2-exchange:

1. Delete two edges

2. Break tour into two path

3. Reconnect those path in the other possible way

Local Search works well for the Circle



The Simulated Annealing Algorithm

1) Generate a random starting solution S;
   Set initial temperature c;

2) Set $S' =$
   a random two-exchange neighbor of S;

3) $\Delta = Length(S') - Length(S)$;

4) If $\Delta \leq 0$, set $S = S'$;
   Else with probability $e^{-\Delta/c}$, set $S = S'$;

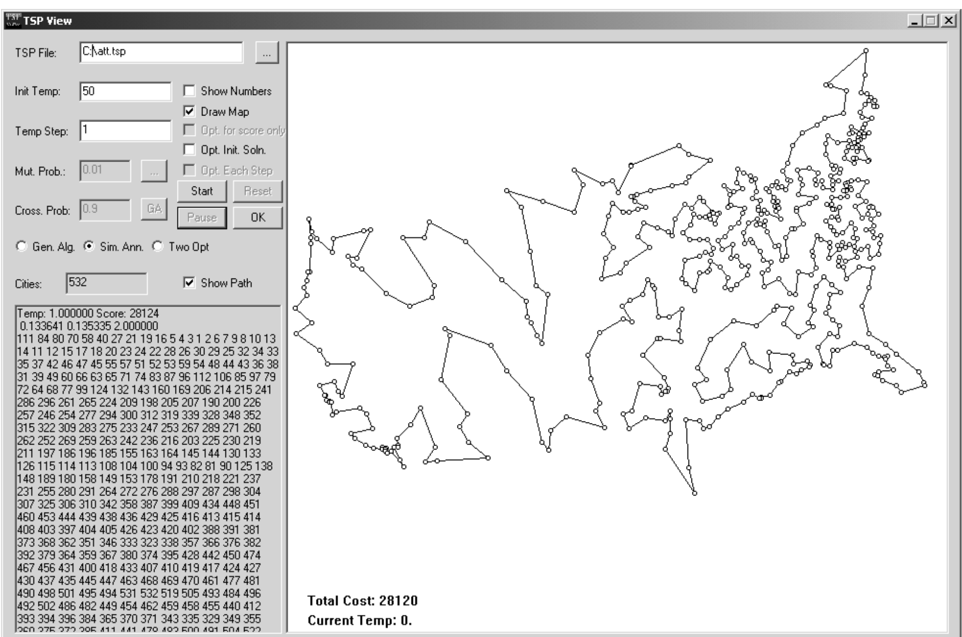
5) If equilibrium reached, $n(n-1)$ iterations completed, reduce c,
   Else goto 2;

6) If $c > 0$ goto 2;

7) Return S as final solution;

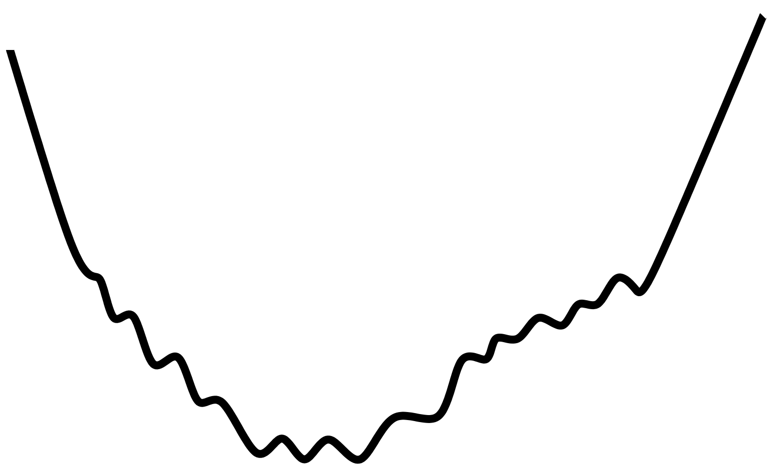
<u>Simulated Annealing:</u>

<u>A Close to Optimal Solution for a 532-City Problem</u>



Big Valley Minima

Under annealing all tours within 5% of optimum



[Boese, 96]

Future directions

- Try different crossovers: Edge Assembly Crossover
- Genotype vs. Phenotype
- Apply lessons learned to other combinatorial optimization problems
- The Resource Constrained Project Scheduling Problem