Heuristic algorithms

• These algorithms come with no guarantee of producing the correct answer, but in practice tend to work well.
• Often used for difficult optimization problems
  – These are problems where we seek the “best” solution among a set of possible solutions
    • Shortest path between two vertices in a graph
    • Shortest Traveling Salesperson tour
    • Maximum value you can put in your backpack
  – Heuristics will give solution that might not be optimal, but that will hopefully be close
• You can imagine all kinds of heuristics, some better than others. Use your imagination!

Traveling Salesperson Problem

• Given:
  – A set of n cities to be visited
  – The distance matrix D, where D(i,j) is the distance between cities i and j
• Find:
  – A way to visit each city exactly once and to return to your starting point, to minimize the total distance traveled
• Example: Solution: ADCBE = 23

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>8</td>
<td>-</td>
<td>7</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>7</td>
<td>-</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>-</td>
</tr>
</tbody>
</table>

Heuristics for TSP

• TSP (in its decision problem version) is NP-Complete
  – No known algorithm can give the optimal solution in polynomial time
  – Still, getting some suboptimal solution is better than getting no solution at all
  – For 50 cities, there are 49! = 6 * 10^{62} possible solutions
• People have come up with all kinds of approaches to get “good solutions”… We’ll look at a few

TSP heuristics - Greedy algorithm

• Greedy algorithm:
  – Start at some randomly chosen city
  – Always move to the closest unvisited city
• Example: A C D E B -> 3 + 2 + 7 + 5 + 8 = 25

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>8</td>
<td>-</td>
<td>7</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>7</td>
<td>-</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>-</td>
</tr>
</tbody>
</table>
Avoiding local minima

Randomization

- Many many approaches have been proposed to avoid falling into local minima

Randomized fastest descent:

Similar to fastest descent, but instead of choosing the “best” neighbor, add some randomization to it, so that

- Good neighbors have higher probability of being chosen than bad ones
- Even bad neighbors have a small chance of being selected

Randomization allows to escape from local minima, but slows down convergence

Fastest descent heuristics

Start with some random solution $S$

Repeat

Consider a set of solutions $T_1…T_k$ in the “neighborhood” of $S$

Replace $S$ by the solution $T_i$ with the best score

Until no improvement is possible

What does “neighborhood” mean? Many possibilities…

$T_1…T_k$ could be the solutions obtained from $S$ by

- Changing the position of one city in $S$, or
- Exchanging the position of two cities in $S$, or
- Reverse the order in which a set of consecutive cities are visited

Avoiding local minima

Genetic algorithm

Idea: Mimic species evolution in biology

Start with a random “population” of random solutions $S_1…S_n$

Repeat for many generations:

For $k = 1…n$

1) Randomly pick two parent solutions $S_i$ and $S_j$ with probability that depends on their score
2) Create a hybrid offspring $T_k$ from $S_i$ and $S_j$ which inherits some of the properties of the parents
3) Insert a few random “mutations” in $T_k$

Replace solutions $S_1…S_n$ by solutions $T_1…T_k$

Fastest descent heuristics

Start with random solution $S$

Repeat

Consider a set of solutions $T_1…T_k$ in the “neighborhood” of $S$

Replace $S$ by the solution $T_i$ with the best score

Until no improvement is possible

What does “neighborhood” mean? Many possibilities…

$T_1…T_k$ could be the solutions obtained from $S$ by

- Changing the position of one city in $S$, or
- Exchanging the position of two cities in $S$, or
- Reverse the order in which a set of consecutive cities are visited

Fastest descent - Trade-offs

- The larger the neighborhood, the higher the chance of converging to a good solution, but the more time it takes to evaluate each neighbor
- Fastest descent heuristics will often get stuck in a local optima solution:

  - Solution that is not optimal but that doesn’t have a neighbor that is better than itself

Avoiding local minima

Randomization

- Many many approaches have been proposed to avoid falling into local minima

Randomized fastest descent:

Similar to fastest descent, but instead of choosing the “best” neighbor, add some randomization to it, so that

- Good neighbors have higher probability of being chosen than bad ones
- Even bad neighbors have a small chance of being selected

Randomization allows to escape from local minima, but slows down convergence

Fastest descent: Example

We consider the method variations are obtained by reversing the order of a consecutive set of cities.

Start with random solution:

$ABDEC: 32$

All possible “reversals”:

$BADEC: 32$
$ADBEC: 26$
$ABEDC: 25$
$ABDCE: 33$
$DBAEC: 29$
$AEDBC: 30$
$ABCED: 32$

Fastest descent - Trade-offs

- The larger the neighborhood, the higher the chance of converging to a good solution, but the more time it takes to evaluate each neighbor
- Fastest descent heuristics will often get stuck in a local optima solution:

  - Solution that is not optimal but that doesn’t have a neighbor that is better than itself

Avoiding local minima

Randomization

- Many many approaches have been proposed to avoid falling into local minima

Randomized fastest descent:

Similar to fastest descent, but instead of choosing the “best” neighbor, add some randomization to it, so that

- Good neighbors have higher probability of being chosen than bad ones
- Even bad neighbors have a small chance of being selected

Randomization allows to escape from local minima, but slows down convergence

Fastest descent heuristics

Start with some random solution $S$

Repeat

Consider a set of solutions $T_1…T_k$ in the “neighborhood” of $S$

Replace $S$ by the solution $T_i$ with the best score

Until no improvement is possible

What does “neighborhood” mean? Many possibilities…

$T_1…T_k$ could be the solutions obtained from $S$ by

- Changing the position of one city in $S$, or
- Exchanging the position of two cities in $S$, or
- Reverse the order in which a set of consecutive cities are visited

Fastest descent: Example

We consider the method variations are obtained by reversing the order of a consecutive set of cities.

Start with random solution:

$ABDEC: 32$

All possible “reversals”:

$BADEC: 32$
$ADBEC: 26$
$ABEDC: 25$
$ABDCE: 33$
$DBAEC: 29$
$AEDBC: 30$
$ABCED: 32$

Fastest descent - Trade-offs

- The larger the neighborhood, the higher the chance of converging to a good solution, but the more time it takes to evaluate each neighbor
- Fastest descent heuristics will often get stuck in a local optima solution:

  - Solution that is not optimal but that doesn’t have a neighbor that is better than itself

Avoiding local minima

Randomization

- Many many approaches have been proposed to avoid falling into local minima

Randomized fastest descent:

Similar to fastest descent, but instead of choosing the “best” neighbor, add some randomization to it, so that

- Good neighbors have higher probability of being chosen than bad ones
- Even bad neighbors have a small chance of being selected

Randomization allows to escape from local minima, but slows down convergence

Fastest descent heuristics

Start with some random solution $S$

Repeat

Consider a set of solutions $T_1…T_k$ in the “neighborhood” of $S$

Replace $S$ by the solution $T_i$ with the best score

Until no improvement is possible

What does “neighborhood” mean? Many possibilities…

$T_1…T_k$ could be the solutions obtained from $S$ by

- Changing the position of one city in $S$, or
- Exchanging the position of two cities in $S$, or
- Reverse the order in which a set of consecutive cities are visited