A fitness landscape analysis of the Travelling Thief Problem

Mohamed El Yafrani, Marcella Martins, Mehdi El Krari, Markus Wagner, Myriam Delgado, Belaïd Ahiod, Ricardo Lüders

Genetic and Evolutionary Computation Conference - GECCO ’18
July 15 - 19, 2018, Kyoto, Japan
Outline

- Introduction
- Background
  - The Traveling Thief Problem (TTP)
  - Fitness Landscapes & Local Optima Networks
- Environment Settings
  - Local Search Heuristics
  - Instance Classifications & Generation
- Results & Analysis
  - Topological properties of LON
  - Degree Distributions
  - Basins of Attraction
- Conclusion
Introduction
Introduction

Objectives:

● Understand the search space structure of the TTP using basic local search heuristics with Fitness Landscape Analysis;
● Distinguish the most impactful non-trivial problem features (exploring the Local Optimal Network representation);
Introduction

Motivation:

- The TTP -> important aspects found in real-world optimisation problems (composite structure, interdependencies,...);
- Only few studies have been conducted to understand the TTP complexity;
- LONs -> useful representation of the search space of combinatorial (graph theory);
- LONs -> characteristics correlate with the performance of algorithms.
Background
Background

The Traveling Thief Problem:

<<Given a set of items dispersed among a set of cities, a thief with his rented knapsack should visit all of them*, only once for each, and pick up some items. What is the best path and picking plan to adopt to achieve the best benefits?>>
The Traveling Thief Problem:

A TTP solution is represented with two components:
1. The path (eg. $x$={A, E, C, F, B, D, A})
2. The picking plan (eg. $y$={15, 16, 5, 17, 20, 9, 11, 12})
Background

The Traveling Thief Problem parameters:

- $W$: The Knapsack capacity
- $R$: The renting rate
- $\frac{v_{\text{max}}}{v_{\text{min}}}$: Maximum/Minimum Velocity

Maximize the total gain:

$$G(x ; y) = \text{total_items_value}(y) - R \times \text{travel_time}(x ; y)$$

The more the knapsack gets heavier, the more the thief becomes slower:

$$\text{current_velocity} = v_{\text{max}} - \text{current_weight} \times \frac{(v_{\text{max}} - v_{\text{min}})}{W}$$
Background

Fitness Landscapes:

A graph $G=(N,E)$ where nodes represent solutions, and edges represent the existence of a neighbourhood relation given a move operator.

⚠ Defining the neighbourhood matrix for $N$ can be a very expensive.
⚠ Hard to extract useful information about the search landscape from $G$. 
Background

Local Optima Networks:

A simplified landscape representation...

✓ Nodes: Local optima / Basins of attraction
✓ Edges: Connectivities between the local optima.

Two basins of attraction are connected if at least one solution within a basin has a neighbour solution within the other given a defined move operator.
Background

Local Optima Networks:

- A simplified landscape representation…
- Provides a very useful representation of the search space
- Exploit data by using metrics and indices from graph theory
Environment Settings
Environment Settings

Local Search Heuristics:

● Embedded neighbourhood structure
  ○ Generates a problem specific neighbourhood function
  ○ Maintains homogeneity of the TTP solutions

Algorithm 1 A basic local search heuristic framework for the TTP

1: \( s \leftarrow \) initial solution
2: \( \textbf{while} \) there is an improvement \( \textbf{do} \)
3: \( \text{for each} \ s^* \ N_{TSP}(s) \ \textbf{do} \)
4: \( \text{for each} \ s^{**} \ N_{KP}(s^*) \ \textbf{do} \)
5: \( \text{if} \ F(s^{**}) > F(s) \ \text{then} \)
6: \( s \leftarrow s^{**} \)
7: \( \text{end if} \)
8: \( \text{end for} \)
9: \( \text{end for} \)
10: \( \text{end while} \)
Environment Settings

**Local Search Heuristics:**

Two local search variants:

1. **J2B:** 2-OPT move
2. **JIB:** Insertion move

} + One-bit-flip operator

---

**Algorithm 1** A basic local search heuristic framework for the TTP

1: \( s \leftarrow \text{initial solution} \)
2: \( \textbf{while} \ \text{there is an improvement do} \)
3: \( \quad \textbf{for each} \ s^* N_{TSP}(s) \ \textbf{do} \)
4: \( \quad \quad \textbf{for each} \ s^{**} N_{KP}(s^*) \ \textbf{do} \)
5: \( \quad \quad \quad \textbf{if} \ F(s^{**}) > F(s) \ \textbf{then} \)
6: \( \quad \quad \quad \quad s \leftarrow s^{**} \)
7: \( \quad \quad \textbf{end if} \)
8: \( \quad \textbf{end for} \)
9: \( \textbf{end for} \)
10: \( \textbf{end while} \)
Environment Settings

● **TTP classification and parameters**
  ○ Number of cities ($n$);
  ○ Item Factor ($F$);
  ○ Profit-value correlation ($T$);
  ○ Knapsack capacity class ($C$);

● **Instance Generation**
  ○ 27 classes of the TTP are considered;
  ○ For each class, 100 samples are generated;
Environment Settings

How we conduct our experiments to achieve the objectives?

1 - Propose a problem classification based on knapsack capacity and the profit-weight correlation;

2 - Create a large set of enumerable TTP instances;

3 - Generate a LON for each instance using two hill climbing variants;

4 - Explore/exploit LONs using specific measures.
Results & Analysis
Topological properties of LONs

Mean number of vertices \(\langle n_v \rangle\) & edges \(\langle n_e \rangle\):

- \(\langle n_v \rangle\) & \(\langle n_e \rangle\) decrease by increasing the knapsack capacity.
- \(\rightarrow\) hardness of search decreases when the knapsack capacity increases
Topological properties of LONs

Mean average degree $\overline{z}$:

- $\overline{z}$ increases with the capacity class
  - Decreases when the capacity class reaches 6
Topological properties of LONs

Mean average clustering coefficients:

- $\bar{C}$: Average clustering coefficients of generated LONs
- $\bar{C}_r$: Average clustering coefficients of corresponding random graphs
  - Random graphs with the same number of vertices and mean degree

- Local optima are connected in two ways
  - Dense local clusters and sparse interconnections
    - Difficult to find and exploit
Topological properties of LONs

Mean path lengths $\bar{l}$

- All the LONs have a small mean path length
  - Any pair of local optima can be connected by traversing only few other local optima.
- $\bar{l}$ is proportional to $\log(n_v)$
- A sophisticated local search-based metaheuristics with exploration abilities can move from a local optima to another only in few iterations
Topological properties of LONs

Connectivity rate $\pi / \text{number of subgraphs} : \overline{S}$

- The connectivity rate shows that all the LONs generated using J2B are fully connected.
- Some of the LONs generated using JIB are disconnected graphs with a significantly high number of non-connected components.
Figure 2: Cumulative degree distribution of J2B (top) and JIB (bottom) for $T = \text{unc}$, $C = 5$ (left), $T = \text{usw}$, $C = 5$ (middle), and $T = \text{bsc}$, $C = 5$ (right). All curves are shown in a log-log scale.
Degree Distributions

Degree distributions decay slowly for small degrees, while their dropping rate is significantly faster for high degrees.

Majority of LO have a small number of connections, while a few have a significantly higher number of connection.

Figure 2: Cumulative degree distribution of J2B (top) and JIB (bottom) for $T = unc, C = 5$ (left), $T = usw, C = 5$ (middle), and $T = bsc, C = 5$ (right). All curves are shown in a log-log scale.
Degree Distributions

Do the distributions fit a power-law as most of the real world networks?

J2B -> A power law cannot be generalised as a plausible model to describe the degree distribution for all the landscape.

Kolmogorov-Smirnov always fails to reject the exponential distribution as a plausible model for all the samples considered.

Table 3: The rates at which the Kolmogorov-Smirnov test fails to reject power-law and exponential as plausible distribution models, with a significance level of 0.1

<table>
<thead>
<tr>
<th></th>
<th>J2B</th>
<th>T=unc, C=5</th>
<th>T=usw, C=5</th>
<th>T=bsc, C=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power-law</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2B</td>
<td>0.22</td>
<td>1</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>JIB</td>
<td>0.39</td>
<td>0.26</td>
<td>0.46</td>
</tr>
<tr>
<td>Exponential</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2B</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>JIB</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Basins of attraction

Average of the relative size of the basin corresponding to the global maximum for each capacity $C$ over the 100 TTP instances for J2B (left) and JIB (right).
In all cases: as the capacity $C$ gets larger, the global optima's basins get larger. (search space size per instance: 46080)
Basins of attraction

Correlation of fitness (x-axis) and basin size (y-axis); J2B (top) and JIB (bottom).

Good correlation can be exploited: get a rough idea (on-the-fly) about achievable performance, and based on this restart dynamically.

[our conjecture, to be implemented]
Conclusions
Conclusions and Future Directions

- Enumerable TTP instances: local area networks created for two heuristics

- Identified characteristics for hardness:
  - Disconnected components
  - Sometimes low correlation of fitness and basin size
    - allows for fitness-based restarts?
  - Easier: large knapsack capacities (larger basins of attraction and overall smaller networks)

- Future work
  - There are (sometimes) many local optima with very small basins
    - Tabu search based on tracked paths and distances to local optima?

- Source code: https://bitbucket.org/elkrari/ttp-fla/
Thank you!

Source code: https://bitbucket.org/elkrari/ttp-fla/