Research Paper Summary: The Orienteering Problem

Brandyn Tucknott

Last Updated: 25 October 2025

1 Introduction to GTSP

Orienteering is a sport in which control points are established in an evironment, and competitors using nothing but a compass and a map must navigate to as many control points as possible within the allotted time limit. To formalize this, given n nodes in a Euclidean plane with score $s_i \geq 0$ $[s_1 = s_n = 0]$, we want to find a route through the nodes to maximize the score beginning at 1 and ending at n, taking no more than TMAX time. This is referred to as the generalized traveling salesman problem (GTSP). GTSP is NP hard, and the traveling salesman is considered a subset of this problem.

2 Common Heuristics

There are two general approaches to solve the GTSP, stochastic and deterministic.

2.1 Stochastic Algorithm

Stochastic algorithms generally rely on Monte Carlo techniques to build a large number of routes, and choosing the best one from this collection. The thought is this: if A_j is a measure of "desirability" for nodes j currently not on the route, then we say

$$A_j = \frac{s_j}{t(\text{last}, j)},$$

where s_j is the score associated with node j and $t(\text{last}, j)$ is the travel time from the last node to j. After choosing at most four values for A_j, we normalize them, and a random number from 0 to 1 is generated to determine which j node is included. This is repeated until no additional nodes can be included in the route.

Deterministic ALgorithm This approach creates routes using a variant of Wren-Holliday vehicle routing procedure. The environment is divided into sectors using concentric circles, and routes are built up from within sectors to save travel time.

3 Center of Gravity Heuristic

This new proposed heuristic has three core steps:

1. Route Construction Step
2. Route Improvement Step
3. Center of Gravity Step

Route Construction The goal of this step is to find a route starting at 1 and ending at n which has a high score while remaining within the time contraints. We form a route using an insertion heuristic, with the best condidates being those with high score that don't add too much to the duration.

3.1 Route Improvement

This step improves upon the route generated in the previous step using an interchange procedure such as 2-OPT to find a shorter route passing through the same set of nodes. This is followed by a cheap insertion, inserting as many nodes as possible without going over TMAX. Call this new route L.

Center of Gravity Suppose that node i has coordinates (x_i, y_i). We can then calculate the center of gravity of L as $g = (\overline{x}, \overline{y})$, where

$$\overline{x} = \frac{\sum_{i \in L} s_i x_i}{\sum_{i \in L} s_i}$$
$$\overline{y} = \frac{\sum_{i \in L} s_i y_i}{\sum_{i \in L} s_i}$$

Now for $i = 1, \ldots, n$, let $a_i = t(i, g)$. We form a route from 1 to n in the following way:

(a) Calculate the ratio s_i/a_i for all i (reward / cost).

(b) Add nodes to the route in decreasing order of this ratio (add nodes with best reward / cost ratio).

(c) Use the route improvement step to make adjustments to the resulting route.

At the end of all this, we end up with a route L'_1. Finding the center of gravity for this route repeats (a) through (c), resulting in another route L'_2. repeat this cycle until routes L'_p and L'_q are identical for $q > p$. Finally, we select the route with the highest score from the collection of routes $\{L'_1, \text{Ł}'_2, \ldots, L'_q\}$. This requires $q \leq 10$, but in practice it has been observed to be $q \leq 5$.