IW - Problem Statement

Oluwapelumi Odimayo

May 7, 2018

Problem Statement:

The Traveling Salesman Problem (TSP) is a well-known and well-studied NP-Hard problem in combinatorial optimization which has been shown to be applicable to various practical and theoretical situations. It state, given a number of cities and distances between each pair of them, find the shortest tour that visits each city exactly once. In this problem, there is a single self-interested agent.

The Competitive Traveling Salesmen Problem (CTSP) is a variation of the standard TSP with multiple self-interested agents (players) compete against each other to reach as many cities as possible. CTSP itself has many possible variations for how it can be defined. This research will focus on a 2-Player game version. Each agent tries to visit as many unvisited cities as they can. For each unvisited city v that an agent reaches before their opponent, they will receive some benefit positive $b(v)$. Furthermore, for each path between cities (u, v) each agent travels along, they pay some cost $c(u, v)$ which is equal to the distance between the two cities.

The game ends when there are no more unvisited cities **and** when both agents have returned to their starting positions. The payoff for each agent *i* is defined as the total benefit they collected minus ther total cost accumulated during their tour. Thus, the ultimate goal for each competing agent is to finish the game with a higher payoff value (total accumulated benefit - total accumulated cost) than their opponent. CTSP is an interesting problem in competitive game theory and has applications like competitve scheduling.

Solutions to CTSP would be a Nash Equilibrium which would be very difficult to find even in small instances of the game.[1] It is for this reason that much of the research on this problem seeks to approximate using a heuristic approach, much like we've seen in the research surrounding the standard TSP. A very interesting area of this heuristic research is the introduction of meta or ”hyer-heuristics” that learn/adapt to game situations and select which low-level heuristic they believe is the best to perform at a point in the game. However, these hyper-heuristic approaches are designed and implemented assuming ”perfect knowledge” i.e. all information about the current state of the game is known to every player. It is shown that the hyper-heuristics outperform the low-level heuristics given perfect knowledge. I want to study how this approach works in a game of incomplete knowledge. The following is a description of how the game is setup, the definition of a ”strategy” for a player, and what information about the state of the game is known to each player.

[1] I'll add in my sources for this and then next part once this is finalized.

The game setup:

- Let $G = (V, E)$ be a **complete** undirected graph.
- b is the benefit function for any vertex $v \in V$
- c is the cost function for any edge $e = (u, v) \in E$
- v_0 is the starting position of Player 1.
- w_0 is the starting position of Player 2.

Each player plays the game described above using their selected strategy. At the start of the game each player's strategy outputs their first move (the node they will travel to). Each player is restricted to moving only to nodes **directly adjacent** to their current position in the graph. To travel from u to v in G, a player must pay cost $c(u, v)$. Furthermore, a player must wait time $t = c(u, v)$ before collecting the benefit $b(v)$ and selecting the next city they will travel to at which point they're strategy will output their next move. If both players try to collect the benefit from a node at the same time, they will each receive $\frac{b(v)}{2}$. Also note that $\forall v \in V$, $b(v) > c(u, v)$. This ensures that each player has incentive to play the game to completion.

A strategy is defined as some function that takes the current state of the game (or some subset of information derived from the state) and outputs their next move (the node they will travel to next). In a game of perfect information, the global state of the game is the same as the information each player knows at each state of the game. This would formally be defined as:

The global state/position of the game:

- S_1 is the current state of Player 1.
- S_2 is the current state of Player 2.
- U is the set of unvisited nodes in the graph G.

We can see that the global state is simply the current state of each player along with the current set of unvisited nodes in the graph G. Strategies in the game of perfect information would take all of this information as input to determine what moves a player would make. In this game of imperfect information, a player is limited by what they know for sure about themselves and the remaining cities left unvisited. Therefore, their strategies only take in the single player's state and the set of unvisited nodes in the graph G to determine how they will play the game. Formally, the state of the game from a single player's perspective is:

Information known to a player i:

- v_i - Player i's starting position.
- v - Player i's current position.
- X - The set of consecutive moves made by Player i.
- T_i - The time Player i must wait before making their next move.
- P_i - Player i's current payoff.

Thus, a strategy for a player i in this game of imperfect knowledge is some function $F(S_i, b, c, U)$ which takes as input player i's state $S_i =< v_i, v, X, T_i, P_i >$ and the current set of unvisited nodes in the graph G and outputs a vertex v' that will be the next move for the given player. This strategy function effectively takes in a state of a player S and outputs a new state S'. This definition of a strategy gives us a relationship between consecutive states of the game.

We say two states S and S' are consecutive if it is possible for the game, subject to the rules previously described, to progress directly from S to S'. The key constraint here is the parameter T which dictates when either player can make a move to another node in the graph. A player i cannot make a move if $T_i \geq 0$. We separate what state changes are allowed into the following cases:

$T_i = 0$ - Player i is allowed to move:

- $v \to v'$ iff $v' \in U$
- $X' = X + v'$
- $T_i' = c(v, v')$
- $P_i' = P_i + (b(v) - c(u, v))$ where u is the node Player i visited before v.

$T_i \neq 0$ - Player i is not allowed to move:

- $v' = v$
- $X' = X$
- $T_i' = T_i - t$ where t is a single in-game time step.
- $P_i' = P_i$

Thus, two Player states S_i and S_i' are consecutive iff they satisfy the above constraints. Furthermore, we say two global states $Y =< S_1, S_2, b, c, U >$ and Y' are consecutive iff their composing substates are consecutive. Let $Z =$ the number of players allowed to make a move at some poin during the game. We define the state changes of U as follows:

$Z = 0$ - No players are allowed to make a move:

- $|U'| = |U|$

$Z = 1$ - A single player is allowed to make a move:

- $|U'| = |U| - 1$
- $\forall v \notin U \rightarrow v \notin U'$

$Z = 2$ - All players are allowed to make a move:

- $|U'| = |U| - 2$
- $\forall v \notin U \rightarrow v \notin U'$

Thus, we now have a complete definition of how the global state of the game can change as it progresses.

The approach of this research is to adapt the perfect knowledge hyper-heuristics developed in other studies to this game of imperfect knowledge. These studies showed that, given perfect information, these heuristics could outperform a restricted set of low-level heuristics. The constraint of imperfect knowledge presents an interesting and novel challenge to CTSP. The low-level strategies I will be developing my heuristic with are:

- Highest Benefit - A pure strategy that takes the state of the game as input and selects the vertex with the highest benefit from the set of unvisited vertices U.
- Nearest Unvisited Neighbor - A pure strategy that takes the state of the game as inpute and selects the vertext that is the shortest distance away from the the player's current position from the set of unvisited vertices U.
- Random - A mixed strategy which plays the Highest Benefit strategy with probability p and Nearest Unvisited Neighbor with probability $1 - p$.

The final goal is to develop an AI that outperforms this subset of strategies by predicting what strategy they are employing and picking moves that will maximize it's payoff accordingly.