Travelling Salesman Problem (TSP)

Josue Daniel Bustamante

R Language implementation

https://github.com/josuedanielbust/tsp-r

Helper functions

```r
# Load nodes and distances from file, Execute Algorithm and Print results
main <- function() {...}

# Swap two nodes
# #Params: route: actual route
#          (i, j): nodes to change
swapNodes <- function(route, i, j) {...}

# Calculate the total distance of the route
# @Params: dist: Distances array list
#          route: Actual route
routeDistance <- function(dist, route) {...}

# Print the results data to a file, export plots to png images
# @Params: result: results from algorithm
#          startTime: start execution time
#          endTime: end execution time
printResult <- function(result, startTime, endTime) {...}
```

Find Route by Nearest Nodes

```r
# @Params: (nodes, dist): Nodes and Distances array lists
findRouteByNearest <- function(nodes, dist) {
  # Start route on first node, adding first node to route
  currentNode <- as.numeric(rownames(nodes)[1])
  route <- c(currentNode)

  # Add nearest neighbor
  # Find the nearest neighbor and add it to the route
  for(i in 1:(nrow(nodes)-2)) {
    distances <- dist[currentNode, -which(rownames(nodes) %in% route)]
    currentNode <- as.numeric(gsub('V', '', names(which(distances == min(distances)))))
    route <- c(route, currentNode)
  }

  # End route on first node
  # Add the final node again to close the path
  last <- which(rownames(nodes) %!in% route)
  route <- c(route, as.numeric(rownames(nodes)[last]))
  route <- c(route, as.numeric(rownames(nodes)[1]))

  # Get the route distance
  # Adding the distance from each node to the next node of the path, using a for loop
  distanceValue <- routeDistance(dist, route)

  # Append to plotsRecords the route (Function removed from slide)

  # Return values for the optimization step
  return(list(distance=distanceValue, route=route, plot=plot))
}
```

Optimize the previous path (Part 1)

```r
# @Params: (nodes, dist): Nodes and Distances array lists
#          actualRoute: findRouteByNearest result
#          startTime: execution start time
optimizeRoute <- function(nodes, dist, actualRoute, startTime) {
  # Control Variables, used on the next While loop
  bestRoute <- actualRoute$route
  minDistance <- actualRoute$distance

  # Array for tracking each change of the distance on the route
  trackDistance <- c()

  # While loop performing swaps if any new route (based on distance) is found
  # (While loop placed on next slide)
  (...)

  # Returning each route found, so we can plot it and export it
  return(list(distance=minDistance, route=route, trackDistance=trackDistance, plot=plot))
}
```

Optimize the previous path (Part 2)

```
while(TRUE) {
  previousDistance <- minDistance
  breakLoop <- FALSE
  # For each node after the second find the distance with their next nodes
  for(i in 2:(nrow(nodes)-1)) {
    for(j in (i+1):(nrow(nodes)-1)) {
      # Try swaping each node and compute their route distance
      route <- swapNodes(route=bestRoute, i=i, j=j)
      newDistance <- routeDistance(dist, route)
      # If the new distance is better than the previous one, update the minimum distance, the best route and plot the route
      if(newDistance < minDistance) {
        minDistance <- newDistance
        bestRoute <- route
        # Append to plotsRecords the route (Function removed from slide)
        # Break the loop for this node and follow with the next node
        breakLoop <- TRUE
        break()
      }
    }
    if(breakLoop) break()
  }
  # Add the new minimum distance to the track distance array
  trackDistance <- c(trackDistance, minDistance)

  # Validate optimization (break while condition)
  # Using probability and the Secretary Problem, 1/e ~= 37%
  if ((i/nrow(nodes)) > (1/exp(1))) { break() }
  # Stop the search if the time is greater than 1 min 40 secs
  actualTime <- Sys.time()
  elapsedTime <- difftime(actualTime, startTime, units="secs")
  if (elapsedTime > 100) { break() }
}
```

Thanks!

Travelling Salesman Problem (TSP)

Josue Daniel Bustamante

> Based on: https://rpubs.com/mstefan-rpubs/salesman