Large Scale Robust Energy Constrained Routing Problem (ECRP) Solved with Clustering Heuristics

Mark Pustilnik

I. ABSTRACT

The Routing problem is a well known and hard to solve problem in practice. The recent developments in car technology, specifically the introduction of Electric Vehicles (EV) introduces a new type of complexity. The limited battery capacity of EV makes the Energy Constrained Routing Problem (ECRP) - a finite number of nodes has to be serviced by a electric fleet trucks while ensuring that none of the truck would run out of energy. This paper suggests to solve the ECRP exactly with a MIP and by clustering Heuristics to speed up the solution. Using the fast solutions achieved by the clustering heuristics, a closed loop planing can be done by re-planning the tours on every time step.

II. INTRODUCTION

The Routing problem (RP) is a well-established combinatorial optimization challenge with significant practical implications in transportation and logistics management - Visiting N nodes by M cars such that every car starts and finishes at a central node (Depot) and each node is visited a single time while minimizing a cost function. The cost function can be a simple total tour time, distance travelled by the cars or can be something more complex like the total financial cost (which is the function of time and energy consumption). The ECRP is an extension of the RP - in addition to the constraints of the RP problem, the energy constraints are added. Every Transition between nodes takes some amount of energy that is subtracted from the vehicle's state-of-charge(SOC). The vehicle SOC must always be positive. Some nodes contain chargers which can charge the car's battery up to a full SOC (or less) by a known charge rate. The cost function to minimize is a function of the total time travelled by all cars and total charging time. Additional constraint may be added such as - limit the maximal number of nodes visited by a single car. The ECRP as well as the classic RP problem is NP-hard and is impractical to solve to optimally for large scale problems [**article2**]. There are many approximated method for solving this problem ([**article1**]). In this paper we develop a method that first clusters the data into M groups (as the number of vehicles) and than solves the problem for each vehicle by clustering each group to subgroups each with a reasonable number of nodes which a simple solver can quickly solve to optimality and connecting them in a simple way. Choosing the clustering method is non trivial [**article3**]. This method can be used to produces sub-optimal solutions of the ECRP (or RP) that, in practice, come close to the optimal solution for the test cases examined and the solution time is a small fraction of the time it takes to get an optimal solution from the best MIP solvers. Another extension of ECRP is the Robust ECRP. Considering the case where the energy consumption of a EV is non deterministic, the solution of the routing problem should take into account the probability properties of the each tour. Meaning, if the energy consumption of travelling between some 2 nodes has a small mean but high variance, we should take it into account when calculating the optimal tour. By assuming that the energy consumption of travelling on an edge between 2 nodes has some distribution (for simplicity we assume normally distributed) we can make sure, that the probability of any vehicle's SOC to reach zero is less than a desired probability. Figure 1 shows an example of a ECRP solution. It shows the nodes on a graph and the tour of each vehicle. Each edge has the mean and standard deviation of the Time of travel in red, and the mean and standard deviation of the energy consumption in blue. It also shows the nominal (solid) and robust (dashed) SOC of each vehicle during the tour. The problem parameters are the number of nodes $N \in \mathbf{N}$, including the Depot (Node 0), and the number of vehicles $m \in \mathbf{N}$. The matrices $T \in \mathbf{R}^{NxN}$ and $E \in \mathbf{R}^{NxN}$ is the time of travel and energy consumption (in term of SOC) between any 2 nodes, respectively. The final parameter is the rate of charge in a charging station $c \in \mathbf{R}$. These parameters are deterministic. Later in the paper a stochastic problem will be examined.

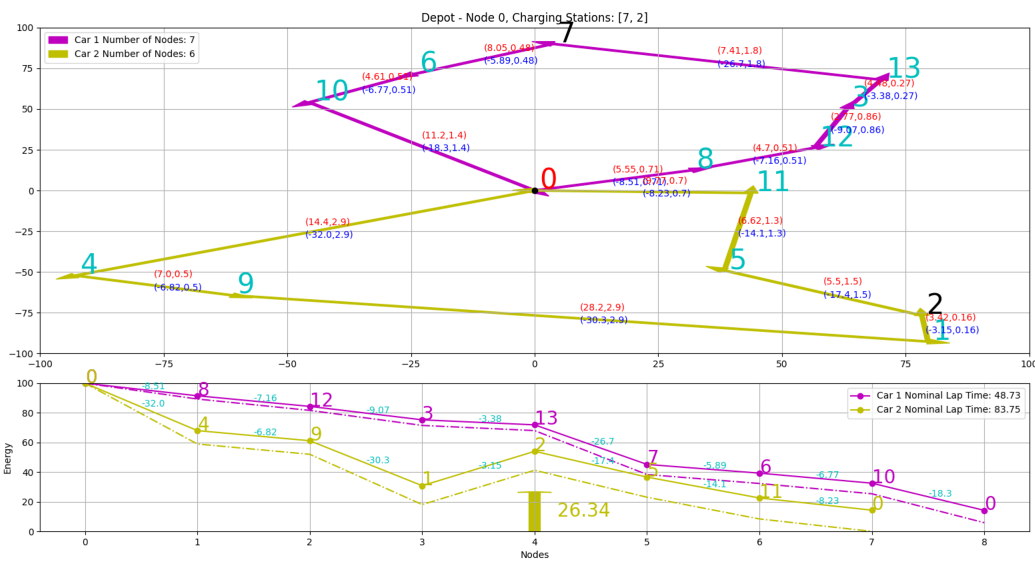


Fig. 1: Example of a Energy Constrained Routing Problem with 20 Nodes and 2 Vehicles

III. Mixed Integer Program Formulation

The ECRP can be described as a Mixed Integer Program (MIP) and solved using any Mixed integer solver. To formulate the MIP - First, define the decision variable matrix $X \in \mathbf{B}^{NxN}$. Every term in the matrix, $x_{i,j}$ is a Boolean variable which is 1 if a tour goes from node i to j, and 0 otherwise. The first constraint of the MIP is that the number of exits leaving node 0 (Depot) is as the number of trucks:

$$\sum_{j=0}^{N-1} x_{0,j} = m \tag{1}$$

The second constraint is the number of cars returning to node 0 (depot) is as the number of trucks:

$$\sum_{i=0}^{N-1} x_{i,0} = m \tag{2}$$

The third set of constrains ensures that every node, other than the depot, has single entrance and exit and no self loops:

$$\begin{aligned} & x_{i,i} = 0, \quad i = 0,1,...,N-1 \\ & \sum_{j=1}^{N-1} x_{i,j} = 1, \quad i = 1,2,...,N-1 \\ & \sum_{i=1}^{N-1} x_{i,j} = 1, \quad j = 1,2,...,N-1 \end{aligned} \tag{3}$$

Constrains 1 to 3 are not enough to define a valid tour. Sub-tour elimination constraints are needed. There are 2 main types of sub-tour elimination types. The first one is known as the Miller–Tucker–Zemlin (MTZ) formulation, and the second one is the Dantzig–Fulkerson–Johnson (DFJ) formulation. Both formulations are valid but each has it's own pros and cons. The DFJ formulation is much more flexible and can be used to constraint many parameters (such as the maximal number of nodes visited per vehicle) but it's not a strong formulation of the problem, in terms of the relaxed MIP solution. This formulation is defined by $\mathcal{O}(N^2)$ constraints on a N auxiliary variable $u_i \in \mathbf{N}$:

$$\begin{aligned} & u_0 = 0 \\ & u_j \geq u_i + 1 - (N_{max} - 1) \cdot x_{i,j}, \quad i \neq j, \quad i,j = 1,2,...,N-1 \end{aligned} \tag{4}$$

Where N_{max} is the maximal number of nodes each vehicle can visit before returning to the depot. The concept of this formulation is that u_i is the order node i was visited. When $x_{i,j} = 0$ the constrains become trivial, when $x_{i,j} = 1$ the constrains makes sure the tour is Hamiltonian. The MTZ formulation is a practical formulation to use for large scale problem. This formulation is essentially a integrator that icreases by 1 every time a node is visited along the tour. This technique will be used later. The Second formulation (DFJ), makes the sub-tour eliminations in more direct approach. For any unwanted sub-tour, there is a constraint that eliminates it. There are 2 types of tours eliminations. The first, any tour that doesn't start from the depot is eliminated. For example, the tour $(4)-(5)-(6)$ should be eliminations and it is done by:

$$x_{4,5} + x_{5,6} + x_{6,4} < 3 \tag{5}$$

These set of constraints are in general written as:

$$\sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{ij} < |Q| - 1, \quad \forall Q \subsetneq \{1, 2, \ldots, n\}, \quad 2 \leq |Q| \leq N/2 \tag{6}$$

The second type is any tour that is longer than allowed:

$$\sum_{i\in Q}\sum_{j\neq i, j\in Q} x_{ij} < |Q|-1, \quad \forall Q\subseteq\{0,1,\ldots,n\}, \quad N_{max}+1<|Q| \tag{7}$$

The DFJ formulation that is defined by (6)-(7) is a stronger formulation than the MTZ for this problem, but it's also exponential in size and therefore impractical to used for large scale problem. The next constraints set meant to deal with the energy constraints of each vehicle. As with the sub-tour elimination constraints, the energy constrains can also be dealt with both the DFJ and MTZ formulations. First, we start with the DFJ formulation. The DFJ formulation main idea is to cancel out any infeasible tour. By going through every possible tour and calculating it's feasibility we can eliminate any infeasible tour. For example, any tour that the sum of energy consumption is greater than the battery capacity and not going through a charging station should be eliminated. In general, any tour that the initial SOC and the charging potential along the tour is greater than the sum of energy consumption along the tour:

$$\sum_{i\in Q}\sum_{j\neq i, j\in Q} x_{ij} \; < \; |Q|-1, \;\; if \;\; SOC_0 + \sum_{j\in Q_{CS}} CP_j + E_Q \; < \; 0, \;\; |Q| \; \leq \; N_{max}+1 \tag{8}$$

Where, Q is the set of all nodes, Q_{CS} is the set of all nodes that contain charging stations, CP_j is the charging potential at charging station j, and SOC_0 is the initial state of charge. The term E_Q is the energy consumption for the tour. This formulation is exponential in size and not practical to use for large problems. The second formulation is the MTZ formulation. Defining The auxiliary variables $\{\epsilon_i\}_{i=0}^{N}$ and $\{e_i\}_{i=0}^{N}$ as the battery SOC when entering and exiting node i, respectively. $\epsilon_i = e_i$ for any node without a charging station and $\epsilon_i \leq e_i$ for $i \in Q_{CS}$. The constrains are:

$$\begin{aligned}
&\epsilon_0 = SOC_0\\
&e_0 = SOC_0\\
&\epsilon_j \geq e_i + E_{i.j} - E_{max}\cdot(1-x_{i,j}), \quad i\neq j, \quad i,j = 1,2,...,N-1\\
&e_i = \epsilon_i, \quad i\in\{Q-Q_{CS}\}
\end{aligned} \tag{9}$$

Where E_{max} is a large number that makes the constraint trivial if $x_{i,j}=0$. The connection between the entering SOC and exiting SOC at each node depends on the charging model. For a linear charging model:

$$\begin{aligned}
&e_i = \epsilon_i + \tau_i\cdot c, \quad i\in Q_{CS}\\
&\tau_i \geq 0, \quad i \in Q
\end{aligned} \tag{10}$$

The charging model can be written as a more realistic non linear charging model where the charging rate is SOC dependent, but this could make the relaxed problem nonlinear. Where $\{\tau_i\}_{i=0}^{n}$ is the charging time at each node. This formulation is essentially an integrator on the SOC state of each truck. Finally, the cost function to be minimized is the tour total time (including the travel and charging time):

$$\min_{X,\epsilon,e,\tau} \sum_{i\in Q} \sum_{j\neq i, j\in Q} x_{ij} \cdot T_{i,j} + \sum_{i\in Q} \tau_i \tag{11}$$

The final MIP formulation is:

$$\begin{aligned}
&\min_{X,\epsilon,e,\tau} \sum_{i\in Q} \sum_{j\neq i, j\in Q} x_{ij} \cdot T_{i,j} + \sum_{i\in Q} \tau_i \\
&s.t. \\
&\sum_{j=0}^{N-1} x_{0,j} = m \\
&\sum_{i=0}^{N-1} x_{i,0} = m \\
&x_{i,i} = 0, \quad i = 0,1,...,N-1 \\
&\sum_{j=1}^{N-1} x_{i,j} = 1, \quad i = 1,2,...,N-1 \\
&\sum_{i=1}^{N-1} x_{i,j} = 1, \quad j = 1,2,...,N-1 \\
&u_0 = 0 \\
&u_j \geq u_i + 1 - (N_{max} - 1) \cdot x_{i,j}, \quad i \neq j, \quad i,j = 1,2,...,N-1 \\
&\epsilon_0 = SOC_0 \\
&e_0 = SOC_0 \\
&e_i \geq 0, \quad i = 1,2,...,N-1 \\
&\epsilon_j \geq e_i + E_{i.j} - E_{max} \cdot (1 - x_{i,j}), \quad i \neq j, \quad i,j = 1,2,...,N-1 \\
&e_i = \epsilon_i, \quad i \in \{Q - Q_{CS}\} \\
&e_i = \epsilon_i + \tau_i \cdot c, \quad i \in Q_{CS} \\
&\tau_i \geq 0, \quad i \in Q \\
&X_{(i,j)} \in \{0,1\}
\end{aligned} \tag{12}$$

IV. Stochastic Problem Formulation

As mentioned, so far we dealt with the deterministic case. To deal with the stochastic case, a normal distribution is assumed for the time of travel and energy consumption between nodes. The time of travel on any edge, as the energy consumption, is assumed independent between ant 2 edges. We introduce the following parameters: $T_\mu \in \mathbf{R}^{NxN}$ and $T_\sigma \in \mathbf{R}^{NxN}$ are the mean and standard deviation of the time of travel between nodes. $E_\mu \in \mathbf{R}^{NxN}$ and $E_\sigma \in \mathbf{R}^{NxN}$ are the mean and standard deviation of the energy consumption between nodes (in terms of SOC). Energy consumption is defined negative if the SOC decreases along the path. The Result tour should have the best cost function with a probability of P_T, and the should met energy constrains with probability greater than P_E. Since the time of travel between nodes is a normally distributed variable and independent on other nodes then the total time of is the time of travel of the tour is also a normally distributed with the following properties:

$$T_{tour} \sim \mathcal{N}(\mu = \sum_{i=0}^{N} \tau_i + \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} x_{(i,j)} \cdot T_{\mu(i,j)}, \quad \sigma^2 = \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} x_{(i,j)} \cdot T^2_{\sigma(i,j)}) \tag{13}$$

Meaning, The total tour time is a normally distributed variable with a mean that is the sum of the means of all the edges of the tour, and the sum of the charging time (deterministic). The variance of the tour time is the sum of the variances of all the edges along the tour. In order to select the tour that minimizes the following cost function:

$$\min_{X,\epsilon,e,\tau} \sum_{i=0}^{N} \tau_i + \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} x_{(i,j)} \cdot T_{\mu(i,j)} + \Phi(P_T) \cdot \sqrt{\sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} x_{(i,j)} \cdot T^2_{\sigma(i,j)}} \tag{14}$$

Where $\Phi(P_T)$ is the value of the CDF function of a normal distribution. For example, if $P_T = 0.5$ then the minimization function depends only on the mean values, and as P_R increases the weight of the variance in the cost function increases. Using this cost function is of course makes the program non-convex and nonlinear. One can approximate the nonlinear non-convex term by a taylor series around the an approximated value of the variance:

$$\sqrt{\alpha} = \sqrt{\alpha - \hat{\alpha} + \hat{\alpha}} = \hat{\alpha}\sqrt{1 + \frac{\alpha - \hat{\alpha}}{\hat{\alpha}}} \approx \hat{\alpha}(1 + 0.5 \cdot \frac{\alpha - \hat{\alpha}}{\hat{\alpha}}) \tag{15}$$

Where $\hat{\alpha}$ is the best estimation for the variance. For example can be the average of all the terms in T_σ multiplied by the number of edges in the tour: $2m + N - 2$. To take into account the stochastic nature of the energy consumption, a new auxiliary variables are introduced - $\{e_i^\sigma\}_{i=0}^{N}$. This variable contains the sum of variances up to node i. To make sure that the energy constrains are met in a probability of a least P_E, we shall demand:

$$\begin{aligned} &e_i \geq \Phi(P_E) \cdot e_i^\sigma, \quad i = 1, 2, ..., N-1 \\ &e_i^\sigma = \sum_{(j,k)\in E_i} {E_{\sigma(j,k)}}^2, \quad i = 1, 2, ..., N-1 \end{aligned} \tag{16}$$

Where E_i is all edges in the tour up to node i. Using this constrains makes the problem nonlinear and non-convex (convexifying the problem with (15) is not possible).

V. Stochastic Mixed Integer Problem Formulation

To formulate the mixed integer problem of the robust ECRP, we combine the deterministic ECRP with the constrains presented in the previous section:

$$
\begin{aligned}
&\min_{X,\epsilon,e,\tau} \sum_{i=0}^{N} \tau_i + \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} x_{(i,j)} \cdot T_{\mu(i,j)} + \Phi(P_T) \cdot \sqrt{\sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} x_{(i,j)} \cdot T^2_{\sigma(i,j)}} \\
&s.t. \\
&\quad \sum_{j=0}^{N-1} x_{0,j} = m \\
&\quad \sum_{i=0}^{N-1} x_{i,0} = m \\
&\quad x_{i,i} = 0, \quad i = 0, 1, ..., N-1 \\
&\quad \sum_{j=1}^{N-1} x_{i,j} = 1, \quad i = 1, 2, ..., N-1 \\
&\quad \sum_{i=1}^{N-1} x_{i,j} = 1, \quad j = 1, 2, ..., N-1 \\
&\quad u_0 = 0 \\
&\quad u_j \geq u_i + 1 - (N_{max} - 1) \cdot x_{i,j}, \quad i \neq j, \quad i, j = 1, 2, ..., N-1 \\
&\quad \epsilon_0 = SOC_0 \\
&\quad e_0 = SOC_0 \\
&\quad e_i \geq \Phi(P_E) \cdot e_i^{\sigma}, \quad i = 1, 2, ..., N-1 \\
&\quad (e_j^{\sigma})^2 \geq (e_i^{\sigma})^2 + (T_{\sigma(i,j)},)^2 - (x_{i,j} - 1) \cdot E_{max} \quad i \neq j, \quad i, j = 1, 2, ..., N-1 \\
&\quad (ef_i^{\sigma})^2 \geq (e_i^{\sigma})^2 + (T_{\sigma(i,0)},)^2 - (x_{i,0} - 1) \cdot E_{max}, \quad i = 1, 2, ..., N-1 \\
&\quad \epsilon_j \geq e_i + E_{i.j} - E_{max} \cdot (1 - x_{i,j}), \quad i \neq j, \quad i, j = 1, 2, ..., N-1 \\
&\quad e_i = \epsilon_i, \quad i \in \{Q - Q_{CS}\} \\
&\quad e_i = \epsilon_i + \tau_i \cdot c, \quad i \in Q_{CS} \\
&\quad \tau_i \geq 0, \quad i \in Q \\
&\quad X_{(i,j)} \in \{0, 1\}
\end{aligned}
\tag{17}
$$

Where $\{ef_i^{\sigma}\}_{i=0}^{N}$ are an auxiliary variables that represent the standard deviation of a tour of a vehicle ending in node i before returning to the depot. This program is hard to solve since even the relaxed problem is non convex (There are quadratic non convex inequalities). This MIP can be solved with any numerical solver that can solve non-convex MIP (Gurobi, for example).

VI. Clustering Heuristics

Solving the robust ECRP is NP hard and there no polynominal time algorithm that solves it, even with the best commercial solvers. For faster solution times, heuristic solving methods can be introduced. In this paper a clustering heuristic is developed to divide the nodes to M groups. Each group is served by a single truck and an optimal Robust Energy Constrained Travelling Salesman Problem (ECTSP) is solved. Since the ECTSP is smaller in size compared to the original ECRP, it's optimal solution can be found faster. The main idea in cluttering the nodes is the understanding that a single truck should serve nodes that are close to each other. When dividing the nodes into groups, it basically turns the big matrices that define the problem (with size $\mathbf{R}^{NxN}$) into smaller matrices, and if the clustering is done wisely many large numbers would disappear (for example, the 2 most distant nodes would probably end up in different groups and the edge between them will not be considered). Each group should include the Depot node. It is also possible to limit the number of nodes in each group to the maximal number of nodes a single truck can visit in a tour. Defining the measure of what is close and how to divide the graph into groups so that the resultant tours would come close to the optimal solution is not trivial. In this paper we would examine a few techniques and analyze the pros and cons of each method. First, lets formulate the clustering problem. The N nodes should me divides into M groups such that some cost function will be minimized. The cost matrix between each node in the graph is given. Since the cost between nodes is not always linear with distance, it's not always possible to calculate a new cost matrix or add new auxiliary points - K-mean method would no be possible to implement with available data. The most obvious example is the edge that connecting 2 nodes with an obstacle between them (for example, a river). The distance is small but the time of travel is long and not trivial. The cost function to be minimized needs to represent the distance between the points in the group. The cost function should be invariant to the order the nodes are arranged in the group. Possible cost function can be: 1) The sum of the Frobenius norm of the m cost matrices - this cost function is easy to calculate but is dominated by the large numbers in the cost matrix that are usually not part of the solution, 2) sum of all the eigenvalues of all M matrices - this is a good measure of the size of the matrix but this also can create outlier points and is hard to practically cluster. 3) sum of the max term in a row of each of the m cost matrices. The cost function that was found to give the best results is: the sum of the maximal eigenvalue of each matrix squared multiplied by the number of nodes in that group:

$$\begin{aligned}
&\min \sum_{i=0}^{m} (\lambda_{max}(C_i))^2 \cdot N_i \\
&s.t. \\
&N_i <= N_{max}
\end{aligned} \tag{18}$$

Where, $C_i \in \mathbf{R}^{N_i x N_i}$ is the i-th group cost matrix that represent the cost of travelling from any 2 nodes in the group. The intuition behind this formulation is to try and lower the size of the matrices while taking into account the group size. The maximal eigenvalue of a matrix is a representation of the maximal size of the matrix in some direction. Trying to lower the maximal cost in worst direction would cause the clustered group to be as square as possible (try to bring all eigenvalues toward the maximum). The cost matrix C_i should be chosen such that the clustered groups would have a good chance to produce as close solution as possible to the optimal one. This cost matrix type is a good measure for choosing the cluster in a relatively simple clustering search algorithm. Optimizing over the maximal eigenvalue gives a sense of the points that should move to other groups. The cost matrix that was found to be the most effective is the following:

$$C = T_\mu + \Phi(P_T) \cdot T_\sigma \tag{19}$$

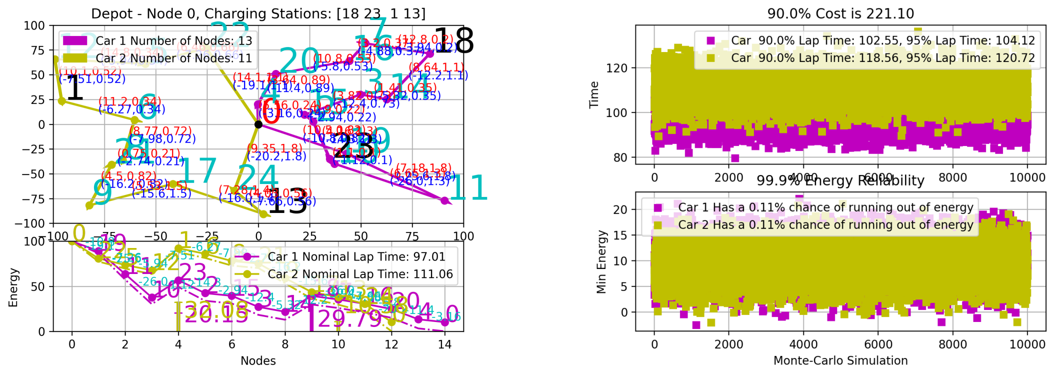
Where, $C \in \mathbf{R}^{NxN}$. This cost function takes into account the time minimization problem, but doesn't take into account the energy problem. This assumes that the energy constraints of the problem are not dominant in the ECTSP (even though this is not always the case). The diagonal terms of C are zeros. The clustering problem is NP hard. The main algorithm used in this paper to cluster the groups is as follows: At first, the nodes are divided randomly between the M groups. Secondly, loop over all nodes and check if any of them should move to a different group. Thirdly, loop over all pair of node in different groups and check if the should switch groups. This search can continue on (loop over all pair of node in the same group and node in a different group to check if they should switch places, and so on...) but it was found that searching up until this point gives good results.

VII. Solving The Robust ECTSP

After the large Robust ECRP has been divided into smaller Robust ECTSP problem, a numerical solver can be used to solve for the optimal tour for each truck. It can be done by solving a MIP or by simple search algorithm. A numerical recursive solver that is described in appendix A is used. Due to the additional energy constraints, many trajectories can be eliminated early in the search, thus a recursive search makes sense in this case. The main idea of the solver is to search recursively all possible routes with breaking points whenever the current best potential route is worst than the best route found so far or it's infeasible due to energy constraints. It is essentially a dynamic programming search algorithm. The search is done using the greedy method to sort the order of search to speed up finding good feasible solution and eliminate many potential tours as early as possible. This recursive algorithm can be paralleled in a multi core processor with a shared memory. If the Robust ECTSP is still to large to solve, there are some techniques that can be used. Stop the search algorithm of the MIP solver or the Recursive solver after a fixed time and save the best solution achieved, or use any other heuristic method. One method that was found to work well is a clustering heuristics that divides the TSP to groups, solves the robust ECTSP problem for each group and then connect all groups by some rule.

VIII. Results

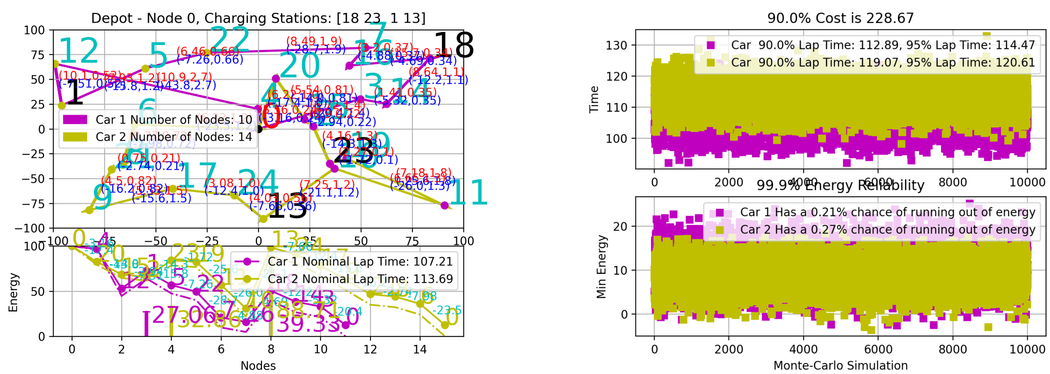
This section presents the results of the Robust ECRP problem introduced in this paper and compares it solution to the optimal global solution (a MIP solved using Gurobi). 2 test cases are examined - 1) A randomly generated stochastic ECRP and 2) a real world data used by Budweiser and it's electric trucks fleet (in this case we can compare their current solution to solution achieved by the Robust ECRP solver). For the first case, we use a randomly generated scenario of 25 nodes over a 2D square map with 2 Trucks fully charged staring from the depot (Node 0). The Solution has to be optimal at 90% of the times ($P_T = 0.9$), and the trucks has to complete the tour at more than 99.9% of the times ($P_E = 0.999$). The solution using the Clustering method and the Recursive ECTSP solver is given in fig 2. Figure 2(a) shows the tour of each of the trucks and the SOC along the tour (solid line is the mean SOC and the dotted line is the minimal SOC at P_E probability). The plot also shows the charging



(a) Tour and SOC along the tour

(b) Monte-Carlo results of the tour

Fig. 2: Routing Problem Solution Solved with Clustering and Dynamic Programming



(a) Tour and SOC along the tour

(b) Monte-Carlo results of the tour

Fig. 3: Routing Problem Solution Solved with MIP formulation and Gurobi Solver

times and charging stations along the tour (vertical arrows). 2(b) shows a Monte-Carlo (MC) simulation results of the tour. The simulation was done by randomizing the time and energy of each edge of the tour with it's respective random properties and summing them to get the tour time and the minimal SOC along the tour. 10000 MC runs were produced. The MC results show that 90% of the tours were less than 102.5 and 118.6 time units for truck 1 and truck 2, respectively. The total simulated 90% tour time is 221.1 time units which is as the optimization predicted. Further more, the probability of not completing the tour is about 0.11% for both trucks as was designed in the problem formulation ($P_E \geq 0.999$). The global optimal solution computed by the solving the robust ECRP MIP using Gurobi is given in fig 3. This program was run for about 10 minutes and the best solution is presented (for comparison, the optimization with clustering and dynamics programming run for less than a minute on the same computer). It can be seen that the solution for the global optimization has larger cost than the cost for the clustered problem. This means that the global problem did not reach the optimal solution.

IX. Conclusions

Solving the Robust ECRP can be done quickly with clustering heuristics that produce sub optimal solution with short computation times. The clustering heuristics is a natural and intuitive way of solving the vehicle routing problem. The added complexity of the energy constraints makes the corresponding MIP extremely hard to solve, but on other hand makes it possible for the introduction of a simple iterative search algorithm that utilized the many tours eliminated by the energy constraints.

X. Appendix A

Solving the Robust ECTSP is done using a recursive program. The inputs to the program is the parameters of the problem and the initial charging state of the truck. The pseudo-code of the program is described below:

Algorithm 1 Robust ECTSP Solver

function RobustECTSP(i, $T_\mu, T_\sigma, E_\mu, E_\sigma, P_T, P_E, CurCost, CurEnergy$) ▷ Where $T_\mu, T_\sigma, E_\mu, E_\sigma \in \mathbf{R}^{nxn} and Pr_T, Pr_E \in \mathbf{R}$
    NodesVisited.append(i)
    $Nodes2Go = set(AllNodes) - set(NodesVisited)$
    if $Nodes2Go == \emptyset$ **then** $CurCost = CalcCost(NodesVisited)$
        if $CurCost < BestCost$ **then** $BestTraj = NodesVisited$ $BestCost = CurCost$
        end if
    else $MinRobustCost2Go = CalcMinCost2Go(Nodes2Go)$ $MinRobustEnergy2Go = CalcMinEnergy2Go(Nodes2Go)$
        if $MinRobustCost2Go >= BestCost or MinRobustEnergy2Go <= 0$ **then**
            $return$
        end if
    end if
    $SortedNodesToGo$ = argsort($T_\mu[i, Nodes2Go] + \Phi(P_T) * T_\sigma[i, Nodes2Go]$)
    for $iNode$ in $SortedNodes2Go$ **do** do
        $BestTraj, BestCost = RobustECTSP(iNode, T_\mu, T_\sigma, E_\mu, E_\sigma, P_T, P_E, CurCost + T_\mu(i, iNode), CurEnergy + E_\mu(i, iNode))$
    end for
end function

Where the function $CalcMinCost2Go(Nodes2Go)$ calculates the minimal possible robust cost as a function of the nodes that still has to be visited. The function $CalcMinEnergy2Go(Nodes2Go$ calculates the maximal possible SOC as a function of the nodes that still has to be visited. This Code is called recursively to solve for the optimal Robust ECTSP. This algoritm utilized the best solution it got to quickly cancel out many other trajectories. The recursive algorithm can be paralleled easily by giving different CPUs a different initial nodes and then choosing the best options.

XI. Appendix B

Connecting small ECTSP groups to one large solution