Separation Routines for the Mixed Fleet Green Vehicle Routing Problem

Jesper Bang Mikkelsen[a]

[a]*Department of Economics and Business Economics, Fuglesangs Alle 4, Aarhus V, DK-8210, , Denmark*

Contents

1. Inequalities

In this report, we describe how we separate inequalities for the Mixed Fleet Vehicle Routing Problem (MFGVRP), that concerns finding the cheap-

*Corresponding author

Email address: jbmikkelsen@econ.au.dk (Jesper Bang Mikkelsen)

est route for a fleet of electric vehicles (EVs) and internal combustion engine vehicles (ICEVs), such that no EV is left stranded and the capacity and maximum allowed duration is respected for all vehicles. For an overview of the notation used and detailed descriptions of the inequalities explain the separation procedures for, we refer to our companion paper.

In this report we will describe the separation procedures for the following inequalities:

No-charge path inequalities:

$$x^E(\mathcal{P}(T)) \leq m - 2 \tag{1}$$

No-charge set inequalities:

$$x^E(A(S)) \leq m - 2 \tag{2}$$

Lifted fixed charging arc inequalities:

$$x^E(A(T)) \leq m - \sum_{i \in W}(\sum_{r \in R_1} y_{iv_1r} + \sum_{r \in R_m} y_{v_mri}) - \sum_{i \in C_1} x^E_{iv_1} - \sum_{j \in C_m} x^E_{v_mj} \tag{3}$$

Fixed charger inequalities:

$$3\sum_{k \in K} x^k(A(S)) + \sum_{i \in V}\sum_{j \in S} y_{ir_1j} + \sum_{i \in S}\sum_{j \in V} y_{ir_2j} \leq 3m - 2 \tag{4}$$

Time infeasible path inequalities:

$$\sum_{k \in K} x^k(\mathcal{P}(T)) + 2y(P^R(\mathcal{P}(T))) \leq m - 2 \tag{5}$$

$$x^E(\mathcal{P}(T)) + 2y(P^R(\mathcal{P}(T))) \leq m - 2 \tag{6}$$

Time infeasible set inequalities:

$$\sum_{k \in K} x^k(A(S_C)) + 2\sum_{i \in S_C}\sum_{r \in S_R}\sum_{j \in S_C} y_{irj} + \sum_{i \in S_C}\sum_{r \in S_R}\sum_{j \in W}(y_{irj} + y_{jri}) \leq m - 2 \tag{7}$$

$$x^E(A(S_C)) + 2\sum_{i \in S_C}\sum_{r \in S_R}\sum_{j \in S_C} y_{irj} + \sum_{i \in S_C}\sum_{r \in S_R}\sum_{j \in W}(y_{irj} + y_{jri}) \leq m - 2 \tag{8}$$

The report is outlined as follows: first, we describe how we identify violated paths and sets in the graph. Next, we describe (if relevant) the procedures we use to check if node sets and sequences are valid for the MFGVRP.

2. Violated paths and sets

Throughout this section, we let (x^*, y^*) be an optimal solution to the LP for which we seek to separate violated inequalities.

2.1. Path inequalities

The separation procedures for inequalities (1), (5), and (6) all relies on identifying paths that cannot be extended to certain nodes (either the depot or a charging option). We begin by describing the separation of no-charge path inequalities (1), after which we proceed with describing the alterations needed for separating time infeasible path inequalities (5) and (6).

2.1.1. No-charge path inequalities

For separating no-charge path inequalities (1) we enumerate paths connecting customers in x^*. Every time a path is extended with a new customer, we check the total slack of the path and the minimal energy required for traversing the path. The procedure is presented as pseudocode in Algorithm 1.

As shown in Algorithm 1, we continue path enumeration, as long as the slack of a path allows us to identify a violation. When a violated path is identified, we check if the minimal energy required for reaching the path from a charging option, traversing the path, and reaching a charging option afterwards exceeds the battery capacity. If this is not the case, we continue extending the path to check if a longer path might violate the battery capacity. The algorithm is terminated, when we have tried building paths with all customers as the first customer on the path.

2.1.2. Time infeasible path inequalities

For separating time infeasible path inequalities (5) and (6) we enumerate paths connecting charger and customers nodes in (x^*, y^*). Every time a path is extended with a new node, we check the total slack of the path and its minimal duration. The procedure is presented as pseudocode in Algorithm 2.

As shown in Algorithm 2, we continue path enumeration, as long as the slack of a path allows us to identify a violation. When a violated path is identified, we check if the total service time and travelling time for reaching the path from the depot, traversing the path, and reaching the depot afterwards exceeds the maximum allowed duration. If this is not the case, we check if this is exceeded for EVs, when we add the minimal required charging time

Algorithm 1: Separating no-charge path inequalities (1)

Let x^* be the direct arcs used in the LP solution (x^*, y^*)
Let B be the battery capacity of the vehicle
Let e_{ij} be the energy required for travelling from customer i to j
Let e_i^+ be the minimal energy required for reaching customer i from a charging option
Let e_j^- be the minimal energy required for reaching a charging option from customer j
Initialize an empty path p: $p \leftarrow \emptyset$
Let E be the energy consumption of p
Let s be the total slack of p
for each customer i connected in x^* **do**
    Set $s = 0$, $E = e_i^+$, and $p = \{i\}$
    for each customer j visited immediately after i in x^* **do**
        Set $s = s + (1 - x_{ij}^*)$, $E = E + e_{ij}$
        if $s < 1$ **then**
            Inequality (1) is violated
            if $E + e_j^- > B$ **then**
                $p \cup j$ is valid for (1). Add (1) for $p \cup \{j\}$
            else
                Add j to p
                Set $i = j$
            end if
        else
            Set $s = s - (1 - x_{ij}^*)$, $E = E - e_{ij}$
        end if
    end for
end for

Algorithm 2: Separating time infeasible set inequalities (5)–(6)

Let (x^*, y^*) be the LP solution
Let T_{MAX} be the maximum allowed duration of route
Let t_{ij} be the travelling time from node i to j
Initialize an empty path p: $p \leftarrow \emptyset$
Let RT be the minimal recharging time for an EV when traversing p
Let ST be the total service time of p
Let T_p be the total time of p
Let s be the total slack of p
for each non-depot node i connected in (x^*, y^*) **do**
 Set $s = 0$, $T_p = t_{0i}$, and $p = \{i\}$
 for each non-depot node j visited immediately after i in (x^*, y^*) **do**
  Set $s = s + (1 - x^*_{ij} - y^*_{ij})$, $T_p = T_p + t_{ij}$
  if $s < 1$ **then**
   Inequalities and (5)–(6) are violated
   if $T_p + ST + t_{j0} > T_{MAX}$ **then**
    $p \cup \{j\}$ is valid for (5). Add (5) for $p \cup \{j\}$
   else if $T_p + ST + t_{j0} + RT > T_{MAX}$ **then**
    p is valid for (6). Add (6) for p
   else
    Add j to p
    Set $i = j$
   end if
  else
   Set $s = s - (1 - x^*_{ij} - y^*_{ij})$, $T_p = T_p - t_{ij}$
  end if
 end for
end for

on top this. Otherwise, we continue extending the path to check if a longer path might violate the allowed duration. The algorithm is terminated, when we have tried building paths with all customers and charging stations as the first node on the path.

2.2. Set inequalities

The separation procedures for inequalities (2), (3), (4), (7), and (8) all relies on identifying paths that cannot be extended to certain nodes (either the depot or charging options). However, compared to the separation procedures of (1), (5) and (6), these inequalities require an additional step of validation before they can be added to the LP.

2.2.1. No-charge set inequalities

For separating no-charge set inequalities (2) we rely on the same procedure as in subsubsection 2.1.1. However, we include an additional step for solving an optimization problem, to determine whether no Hamiltonian path exist through the customers on the path that respects the battery capacity. The procedure is shown in pseudocode in Algorithm 3.

As shown in Algorithm 3, we add the additional validation step to check if it is impossible to service all customers on the path with an EV. Else the procedure is exactly the same as in subsubsection 2.1.1. For further information about the optimization problem we solve to conclude whether the set of customers on the path is valid for the MFGVRP, we refer to subsection 3.1.

2.2.2. Lifted fixed charging arc inequalities

For separating inequality (3) we enumerate paths that starts in a charging option, visit two or more customers and ends in a charging option. Whenever such path is found, we check if the charging options and customers both violate (3) and is valid for the problem. This separation procedure is shown in pseudocode in Algorithm 4.

As shown in Algorithm 4, whenever we find a path that both starts and ends at a charging option, we check if inequality (3) is violated. If we find a combination that violates inequality (3), we validate the combination through the optimization problem described in subsection 3.2. If it is valid, we add it to the LP. The procedure is terminated, whenever we have tried to build paths starting from all charging options.

Algorithm 3: Separating no-charge set inequalities (2)

Let x^* be the direct arcs used in the LP solution (x^*, y^*)
Let B be the battery capacity of the vehicle
Let e_{ij} be the energy required for travelling from customer i to j
Let e_i^+ be the minimal energy required for reaching customer i from a charging option
Let e_j^- be the minimal energy required for reaching a charging option from customer j
Initialize an empty path p: $p \leftarrow \emptyset$
Let TSP_{OBJ} be the minimal energy consuming path for visiting all customers on p.
Let E be the energy consumption of p
Let s be the total slack of p
for each customer i connected in x^* **do**
  Set $s = 0$, $E = e_i^+$, and $p = \{i\}$
  for each customer j visited immediately after i in x^* **do**
    Set $s = s + 1 - \sum_{u \in p \cup}(x^*_{uj} + x^*_{ju})$, $E = E + e_{ij}$
    if $s < 1$ **then**
      Inequality (2) is violated
      if $E + e_j^- > B$ **then**
        Solve TSP described in subsection 3.1 for $p \cup \{j\}$
        if $TSP_{OBJ} > B$ **then**
          $p \cup \{j\}$ is valid for (2). Add (2) for $p \cup \{j\}$
        else
          Add j to p
          Set $i = j$
        end if
      else
        Add j to p
        Set $i = j$
      end if
    else
      Set $s = s - 1 + \sum_{u \in p \cup}(x^*_{uj} + x^*_{ju})$, $E = E - e_{ij}$
    end if
  end for
end for

Algorithm 4: Separating lifted fixed charging arc inequalities (3)

Let (x^*, y^*) be the LP solution
Let B be the battery capacity of the vehicle
Let e_{ij} be the energy required for travelling from node i to j
Initialize an empty path p: $p \leftarrow \emptyset$
Let TSP_{OBJ} be objective value of the optimal solution for p to the problem described in subsection 3.2.
Let E be the energy consumption of p
for each charging option r_1 used in (x^*, y^*) **do**
    for each customer i reached from charging station r_1 in y^* **do**
        Set $E = e_{r_1 i}$ and $p = \{i\}$
        for each node j visited immediately after i in (x^*, y^*) **do**
            Set $E = E + e_{ij}$
            if j is a charging station or the depot and $E > B$ **then**
                $r_2 = j$
                Check if r_1, p, r_2 violates (3)
                if (3) is violated **then**
                    Solve TSP described in subsection 3.2
                    if $TSP_{OBJ} > B$ **then**
                        p is valid for (3). Add (3) for r_1, p, and r_2
                    end if
                end if
            else
                Set $p = p \cup \{j\}$
            end if
        end for
    end for
end for

2.2.3. Fixed charger inequalities

The procedure for separating inequalities (4) is equivalent to Algorithm 4. Hence, the procedure only differs in how we validate the combination of the two charging options and the customers on the path found through the path enumeration. The optimization problem we use to determine whether a combination of charging options and customers is valid for the MFGVRP is described in subsection 3.3.

2.2.4. Time infeasible set inequalities

For separating time infeasible set inequalities (7)–(8) we rely on the same procedure as in subsubsection 2.1.2. However, we include an additional step for solving an optimization problem, to determine whether no Hamiltonian path exist through the customers on the path that respects the maximum allowed duration for different types of vehicles. The procedure is shown in pseudocode in Algorithm 5.

As shown in Algorithm 5, we add the additional validation step to check if it is impossible to service all customers on the path. We first try to validate inequality (7). If we are not able to validate inequality (7), we try adding the minimal required recharging time to check if the set is valid for EVs according to inequality (8). Else the procedure is exactly the same as in subsubsection 2.1.2.

3. Validation

All path inequalities (1), (5) and (6) are validated during the path enumeration, and hence we do not describe these in this section. We refer to subsection 2.1 for how these are validated. In contrast, all set inequalities (2), (3), (4), (7), and (8) require an additional step of validation. In the following, we show how these are validated.

For validating inequalities (2), (3), (4), (7), and (8), we solve different Travelling Sales Man problem (TSP) that are tailored to the inequality we aim to validate. The TSP is a classical optimization problem, that concerns finding the Hamiltonian cycle in a graph that minimizes the objective. Therefore, we use the solution to these problems to access whether it is impossible to service a number of customers in a certain way.

We define the TSP on a complete directed graph $G = (T, A)$, where $T = \{a_0, v_1, \ldots, v_m\}$ is a set of an artificial node a_0 and an ordered set of natural nodes $T_C = v_1, \ldots, v_n$, and $A = \{(i, j) : i, j \in T\}$ is the set of arcs connecting

Algorithm 5: Separating time infeasible set inequalities (7)–(8)

Let (x^*, y^*) be the LP solution
Let T_{MAX} be the maximum allowed duration of route
Let t_{ij} be the travelling time from node i to j
Initialize an empty path p: $p \leftarrow \emptyset$
Let RT be the minimal recharging time for an EV when traversing p
Let ST be the total service time of p
Let T_p be the total time of p
Let TSP_{OBJ} be the minimal travelling time for visiting the nodes on p
Let s be the total slack of p
for each non-depot node i connected in (x^*, y^*) **do**
Set $s = 0$, $T_p = t_{0i}$, and $p = \{i\}$
for each non-depot node j visited immediately after i in (x^*, y^*) **do**
Set $s = s + 1 - \sum_{u \in p}(x^*_{uj} + y^*_{uj} + x^*_{ju} + y^*_{ju})$, $T_p = T_p + t_{ij}$
if $s < 1$ **then**
Inequalities and (5)–(6) are violated
if $T_p + ST + t_{j0} > T_{MAX}$ **then**
Solve TSP described in subsection 3.4
if $TSP_{OBJ} + ST > T_{MAX}$ **then**
$p \cup \{j\}$ is valid for (7). Add (7) for $p \cup \{j\}$
else if $TSP_{OBJ} + ST + RT > T_{MAX}$ **then**
$p \cup \{j\}$ is valid for (8). Add (8) for $p \cup \{j\}$
else
Add j to p
Set $i = j$
end if
else
Add j to p
Set $i = j$
end if
else
Set $s = s - 1 + \sum_{u \in p}(x^*_{uj} + y^*_{uj} + x^*_{ju} + y^*_{ju})$, $T_p = T_p - t_{ij}$
end if
end for
end for

all nodes in T. Furthermore, we define an arc set, $A_C = \{(i,j) : i,j \in T_C\}$, for all arcs connecting natural nodes in T. We use the artificial node a_0 to represent the different path structures considered in (2), (3), (4), (7), and (8).

The asymmetric resource consumption of traversing arc $(i,j) \in A$ is defined as c_{ij}. We assume that these are all non-negative.

The general mathematical model of the TSP is then defined as:

$$\min \sum_{i\in T}\sum_{jinT} c_{ij}x_{ij} \tag{9}$$

$$\text{s.t.} \sum_{i\in T} x_{ij} = 1 \qquad \forall j \in T \tag{10}$$

$$\sum_{j\in T} x_{ij} = 1 \qquad \forall i \in T \tag{11}$$

$$\sum_{i\in Q}\sum_{j\in Q} x_{ij} \leq |Q| - 1 \qquad \forall Q \nsubseteq T, |Q| \geq 2 \tag{12}$$

$$x_{ij} \in \{0,1\} \qquad \forall (ij) \in A \tag{13}$$

The objective (9) minimize the resource consumption. Constraints (10) and (11) ensure that exactly one arc is used to reach all nodes and exactly one arc is used to leave all nodes, respectively. Constraints (12) eliminates subtours. Finally, constraints (13) define variable domains.

In the following we describe how we update the artificial node a_0 and the resource consumption c_{ij} to validate inequalities (2), (3), (4), (7), and (8).

3.1. No-charge set inequalities

For validating (2), we set the resource consumption c_{ij} to the energy consumption required for traversing arc $(ij) \in A_C$, c_{a_0j} to the minimum energy consumption required for reaching node $j \in T_C$ from a charging option, and c_{ia_0} to the minimum energy consumption required for reaching a charging option from node $i \in T_C$.

We then solve the TSP problem with the original objective function (9) and constraints (10)–(13). If the energy consumption of the optimal solution to this problem exceeds the battery capacity of the EV, we conclude that no Hamiltonian path exists that includes all nodes in T_C without violating the battery capacity. Hence, T_C valid for (2), as it is impossible to service T_C with an EV without involving a charging option.

3.2. Lifted fixed charging arc inequalities

For validating (3), we set the resource consumption c_{ij} to the energy consumption required for traversing arc $(ij) \in A_C$, c_{a_0j} to the minimum energy consumption required for reaching node $j \in T_C$ from charging option r_1, and c_{ia_0} to the minimum energy consumption required for reaching charging option r_2 from node $i \in T_C$. We then add the following constraint to the TSP model:

$$x_{a_0v_1} + x_{v_ma_0} = 2 \tag{14}$$

where (14) ensure that arc (a_0v_1) is used to enter T_C and arc (v_ma_0) is used to leave T_C.

We then solve the TSP problem with the original objective function (9) and constraints (10)–(14). If the energy consumption of the optimal solution to this problem exceeds the battery capacity of the EV, we conclude that no Hamiltonian path exists that includes all nodes in T_C, if an EV enters T_C through v_1 from charging option r_1, and that same vehicle leaves T_C from v_m to charging option r_2. Hence, r_1, T_C, and r_2 are valid for (3), as it is impossible to service T_C when (r_1v_1) and (v_mr_2) are used to enter and exit T_C, respectively.

3.3. Fixed charger inequalities

For validating (4), we set the resource consumption c_{ij} to the energy consumption required for traversing arc $(ij) \in A_C$, c_{a_0j} to the minimum energy consumption required for reaching node $j \in T_C$ from charging option r_1, and c_{ia_0} to the minimum energy consumption required for reaching charging option r_2 from node $i \in T_C$.

We then solve the TSP problem with the original objective function (9) and constraints (10)–(13). If the energy consumption of the optimal solution to this problem exceeds the battery capacity of the EV, we conclude that no Hamiltonian path exists through T_C that respects the battery capacity, if T_C is entered from charging option r_1 and charging option r_2 is used to leave T_C. Hence, r_1, T_C, and r_2 are valid for (4).

3.4. Time infeasible set inequalities

For validating (7), we set the resource consumption c_{ij} to the travelling time for traversing arc $(ij) \in A_C$, c_{a_0j} to the travelling time require for reaching customer $j \in T_C$ from the depot, and c_{ia_0} to the travelling time required for reaching the depot from customer $i \in T_C$.

We then solve the TSP problem with the original objective function (9) and constraints (10)–(13). If the total travel in the optimal solution and the total service time of T_C exceeds the maximum allowed duration, we conclude that no feasible Hamiltonian path exists for T_C. Hence, T_C is valid for (2), as it is impossible to service T_C within the maximum allowed duration.

For validating (8) we solve the same problem. However, we add the minimum required charging time on top of the total travelling and charging time. If this exceeds the maximum allowed duration, then T_C is valid for time infeasible set inequality, and no EV can service all nodes in T_C on the same route