PLATON -

Planning Process and Tool for Step-by-Step Conversion of the Conventional or Mixed Bus Fleet to a 100% Electric Bus Fleet

Deliverable 4.4:	Efficient Planning Process Part of Milestone M3 Main work package: 4 Efficient Models and Methods Report work package: 4.5 Economic Models Report work package: 4.7 Vehicle Scheduling Optimization Component
Due Date:	December 31, 2019
Report term:	July 1, 2018 - December 31, 2019
Funding code:	EME/03/PLATON/2018
Project term:	January 1, 2018 - June 30, 2020

Editor: Mikhail Y. Kovalyov (UIIP-NASB)
Contributors: Nikolai Guschinsky (UIIP-NASB)
Mikhail Y. Kovalyov (UIIP-NASB)
Boris Rozin (UIIP-NASB)
Sergey Chistov (BKM)
Krzysztof Krawiec (SUT)
Olaf Czogalla (ifak)

Grant beneficiary of WP leader:
The United Institute of Informatics Problems of the National Academy of Sciences of Belarus (UIIP-NASB)

Funding organization of WP leader:
National Academy of Sciences of Belarus (NASB),
Independence Ave. 66, 220072, Republic of Belarus, Minsk

Abstract

In Deliverable 4.3 “Efficient charging structure” of the project PLATON, the planning process of conversion of the conventional or mixed bus fleet to a 100% electric bus fleet is modeled by the optimization problems Opt, DepOpt and OptSched, which take into account cost and resource factors of the planning process. Solutions of these problems need efficient algorithms and data structures. This Deliverable contains description of such algorithms and data structures.

This deliverable is amended by the chapter 6 of an advanced and efficient model of the Total Cost of Ownership (TCO). The developed TCO model is a socially-oriented, dynamic model of TCO. By design, this model is to serve an ex-ante assessment of the costs of the planned investment. The developed dynamic TCO model provides different ways of financing of the investment, as well as its implementation in parts over various periods of time.

Contents

1 Introduction **5**

2 Efficient algorithm and data structure for OPT **6**

2.1 Computational complexity of OPT 7

2.2 Randomized heuristic algorithm for OPT 9

2.3 Computer implementation 15

2.4 Formats of the input files 17

2.4.1 JSON format of the input file 17

2.4.2 Text format of the input files 23

2.5 Formats of the output files 27

2.5.1 JSON format of the output file 27

2.5.2 Text format of the output files 28

2.6 Minsk case of OPT 29

3 Efficient algorithm and data structure for DEPOPT **33**

3.1 Computational complexity of DEPOPT 35

3.2 Solution method for DEPOPT 36

3.3 Efficient structure of input data for DEPOPT 40

3.4 Format JSON of input files for DEPOPT 43

3.5 Text format of input files for DEPOPT 47

3.6 Efficient structure of output data for DEPOPT 49

3.7 Minsk case of DEPOPT 50

4 Efficient algorithm and data structure for OPTSCHED **52**

4.1 Algorithm for OPTSCHED 53

4.2 Computer implementation, efficient structure of input and output data and Minsk case of OPTSCHED 54

5 Conclusion **55**

6 Total Cost of Ownership model **56**

6.1 Static TCO model . . . 56

6.2 Socially-oriented dynamic TCO model . . . 56

6.3 Dynamic vs static model . . . 58

6.4 Electric Buses Acquisition Costs . . . 60

6.5 Credit with equal installments (annuities) . . . 62

6.6 Credit with decreasing installments (installments with fixed capital part) . . . 62

6.7 Operating Costs . . . 64

6.8 Infrastructure costs . . . 66

6.9 Credit with equal installments (annuities) . . . 67

6.10 Credit with decreasing installments (installments with fixed capital part) . . . 67

6.11 External costs . . . 69

6.12 Liquidation value . . . 70

6.13 The application of TCO model – case study based on realistic operational scenario 71

7 References **75**

1 Introduction

This report describes results of the project PLATON on the determination of efficient algorithms and data structures for the problems Opt, DepOpt and OptSched which are formulated in the previous Deliverable. As in the previous Deliverable, we call an electric bus as an e-bus.

Problem Opt is to determine a fleet of e-buses with fast-charging batteries, places for charging stations and transformers, assignment of charging stations to the selected places, assignment of charging stations to the transformers and assignment of charging stations to the routes such that all e-buses can feasibly drive, the required traffic interval is maintained, and the output power of any transformer is not exceeded. The objective is to maximize the total value (positive ecological and social effect expressed quantitatively), provided that the total capital cost and the total operating, depreciation and energy cost do not exceed their upper bounds. It is assumed that Opt will be solved repeatedly for several successive planning periods (years). Decisions made in the past periods will be used as a part of the input for the future period.

Problem DepOpt considers a given fleet of e-buses with slow-charging batteries and fixed timetables, which charge at the same depot. The problem is to determine dynamic quantities of the required electric power supplied to the depot by the city power grid, the type and the number of charging stations of this type in the depot, types of e-bus batteries and charging times of each e-bus while it is in the depot such that the total daily cost of the charging equipment and the consumed energy is minimized, provided that the arrival and departure times of e-buses to/from the depot, the dynamic upper bound on the supplied power and functions of charge and discharge of the batteries are addressed.

Problem OptSched is to determine a route timetable such that the same average traffic interval of all public vehicles of the same route is maintained and departures of public vehicles of the same type assigned to the same route are distributed as smoothly as possible over departures of all public vehicles in the most representative time period.

Solutions of the problems Opt, DepOpt and OptSched need efficient algorithms and data structures, which are described in Sections 2, 3 and 4, respectively. Since the efficiency of a data structure depends on the algorithm which employs this data structure, the algorithms are described first followed by the description of the data structures. Section 5 contains concluding remarks.

2 Efficient algorithm and data structure for Opt

A solution of the problem Opt is denoted as X. Other notations used in this section can be found in the previous Deliverable. Problem Opt is formulated as follows.

$$\max_X V(X), \text{ subject to}$$

$$CC(X) \leq ucc, \tag{1}$$

$$OC(X) \leq uoc, \tag{2}$$

$$Z_r(X) \leq pas_r + \min_{b \in B_r(X)} \{cap_b\} - 1,\ r \in R(X), \tag{3}$$

$$Z_r(X) \geq \min_{b \in VC_r} \{cap_b\},\ r \in R(X), \tag{4}$$

$$TP_q(X) + oo_q \leq o_q,\ q \in T(X), \tag{5}$$

$$AT_r(X) \leq ut_r,\ r \in R(X), \tag{6}$$

$$c_{jrb}(X) \in C_{p(j)} \cap C_b,\ j \in SR_{rb}(X),\ r \in R_b(X),\ b \in B(X), \tag{7}$$

$$ei_{r(j_k^{rb},j_{k+1}^{rb})b}=1, j_k^{rb}\in SR_{rb}(Q), k=0,\dots,n(r,b)-1, ei_{r(j_{n(r,b)}^{rb},j_1^{rb})b}=1, r\in R_b(X), b\in B(X), \tag{8}$$

$$nc_{pc} + NC_{pc}(X) \leq uc_{pc},\ p \in S_c(X),\ c \in C(X), \tag{9}$$

$$\sum_{c \in C_b} (nc_{p(j)c} + NC_{p(j)c}(X)) \geq 1,\ j \in NM_b,\ p(j) \in S(X),\ b \in B(X), \tag{10}$$

$$nc_{pc} + NC_{pc}(X) \geq BN_{pc}(X),\ p \in S(X) \cup NO,\ c \in C, \tag{11}$$

$$|L_p(X)| = m,\ p \in S(X) \backslash NO. \tag{12}$$

Constraints (1) and (2) bound the total capital cost and the total operating, depreciation and energy cost from above. Constraints (3) limit the total passenger capacity of new e-buses on each route $r \in R(X)$ by the total capacity of conventional vehicles on this route plus the capacity of the largest e-bus selected for this route. Constraints (4) state that the total passenger capacity of new e-buses on each route $r \in R(X)$ should be at least the minimal capacity of a single conventional vehicle on this route. Constraints (5) ensure that the total instant power demand of new and old charging stations linked to the same transformer does not exceed the output power of this transformer. Constraints (6) specify upper bound on the length of the average traffic interval of all e-buses assigned to the same route. Constraints (7) ensure that an appropriate charging station is opened at each node associated with the vertices from the sequence $SR_{rb}(X)$. Constraints (8) guarantee that any new e-bus can feasibly run over the route to which it is assigned if appropriate charging stations are opened at the nodes associated with the vertices from the sequence $SR_{rb}(X)$. Constraints (9) limit the total number of old and new charging stations of any type at any node from above. Constraints (10) state that at least one old or new charging station of a type $c \in C_b$ must be opened at a node associated with the vertex from the set NM_b if this vertex belongs to a route served by at least one new e-bus. At least, depot is such a vertex. Constraints (11) guarantee that the number of old and new charging stations of type c opened at node p is sufficient to serve e-buses of all types assigned to this charging station type and node. Constraints (12) guarantee that the number of new links of a non-transformer node, at which at least one new charging station is open and no old charging station was open, with transformer nodes is equal to m.

In the next section, problem Opt is proved NP-hard in the strong sense for two important special cases, which implies that the development of an algorithm with a reasonable running time for this problem is highly unlikely. Development of such an algorithm would assume solution of the *"P versus NP"* problem, which is among seven "Millennium Prize Problems" of the Clay Mathematics Institute.

2.1 Computational complexity of Opt

In this section, we prove that Opt is NP-hard in the strong sense, which means that the development of an algorithm which will be able to solve this problem to optimality in a reasonable time is highly unlikely, and the development of heuristic (approximation) algorithms is justified.

Theorem 1 *Problem* Opt *is NP-hard in the strong sense even if there is a single charging station type, a single transformer with unlimited output power and eligible for linking with any charging station location, and all costs are zero.*

Proof: We will show that Opt is algorithmically equivalent to the NP-complete in the strong sense problem 3-Partition, see Garey and Johnson [2].

3-Partition: Given $3k+1$ positive integer numbers $h_1, \ldots, h_{3k}$ and H satisfying $\sum_{i=1}^{3k} h_i = kH$ and $H/4 < h_i < H/2$, $i = 1, \ldots, 3k$, does there exist a partition of the set $\{1, \ldots, 3k\}$ into subsets $X_1, \ldots, X_k$ such that $\sum_{i \in X_j} h_i = H$ for $j = 1, \ldots, k$?

Given an instance of 3-Partition, we construct an instance of the problem Opt, in which there are $3k$ e-bus types with $cap_b = 1$, and $3k$ routes with $pas_r = 1$ and $v_r(Z) = Z$, $b = 1, \ldots, 3k$, $r = 1, \ldots, 3k$. Only e-bus type b is eligible for route b, $b = 1, \ldots, 3k$. Upper bound on the average length of the traffic interval of all e-buses on the same route are equal to 1 for all routes. There is a single charging station type c. All routes go via the same stops $j = 1, \ldots, k$, at each of which at most $uc_{jc} = H$ charging stations can be opened. Any e-bus must be charged at least once at one of the k stops and one such charging is sufficient to complete its route cycle. The charging times are $ct_{jbc} = h_b$ for $j = 1, \ldots, k$, $b = 1, \ldots, 3k$. We will show that there exists a feasible solution X for this instance with value $V(X) \geq 3k$ if and only if the original instance of 3-Partition has a solution.

"Only if". Assume that there exists a feasible solution X of the constructed instance of the problem Opt with value $V(X) \geq 3k$. From $V(X) \leq \sum_{r=1}^{3k} pas_r = 3k$, it follows that there exists a solution, in which at least one e-bus of type r is assigned to each route r for $r = 1, \ldots, 3k$. Consider the set of routes $R_j(X)$ whose e-buses charge at the stop j, $j = 1, \ldots, k$. Since $ATE_r(X) = AT_r(X) \leq 1$, it follows from the definition of the function $BN_{jc}(X)$ that $\sum_{r \in R_j(X)} h_r = \sum_{r \in R_j(X)} ct_{jrc} \leq \sum_{r \in R_j(X)} \frac{ct_{jrc}}{ATE_r(X)} = BN_{jc}(X) \leq H$ is satisfied for $j = 1, \ldots, k$. Since $\sum_{i=1}^{3k} h_i = kH$, the latter relations are satisfied only if $\sum_{r \in R_j(X)} h_r = H$ for $r = 1, \ldots, k$. Hence, sets $X_j = R_j(X)$, $j = 1, \ldots, k$, constitute a solution of 3-Partition.

"If". Let $X_1, \ldots, X_k$ be a solution of 3-Partition. Construct a solution X of Opt, in which $NV_{rb}(X) = d_r + h_b$ e-buses of type $b = r$ serve route r for $r = 1, \ldots, 3k$, $R_j(X) = X_j$ and e-buses of types from X_j are charged at stop j and only at this stop, $j = 1, \ldots, k$. For this solution, $V(X) = 3k$, $AT_r(X) = \frac{d_r + h_b}{NV_{rb}(X)} = 1$, and $BN_{jc}(X) = H$, $j = 1, \ldots, k$, as it is required for part "if". ■

Theorem 2 *Problem* Opt *is NP-hard in the strong sense even if there is a single e-bus type, a single route, and all costs are zero.*

Proof: A reduction from the NP-complete problem 3-Partition formulated in the previous theorem is used. Given an instance of 3-PARTITION, we construct an instance of the problem OPT, in which there is a single route r with $pas_r = 1$, $v_r(Z) = Z$ and stops $j = 1, \ldots, 3k$, a single e-bus type b with $cap_b = 1$, which can serve r, and k transformers with output power $o_i = H$, $i = 1, \ldots, k$, which can be linked with any of the $3k$ stops. At stop j, a single charging station of type j can be opened such that $C_{jb} = \{j\}$, and $po_j = h_j$ $j = 1, \ldots, 3k$. Each charging station must be linked with one transformer. Any e-bus must be charged at each of the $3k$ stops in order to complete its route cycle. We will show that there exists a feasible solution X for this instance with value $V(X) \geq 1$ if and only if the original instance of 3-PARTITION has a solution.

"Only if". Assume that there exists a feasible solution X of the constructed instance of the problem OPT with value $V(X) \geq 1$. Then, there exists a solution, in which a single e-bus of type b is assigned to route r. Consider the set of stops $M_i(X)$ linked with transformer i, $i = 1, \ldots, k$. For feasibility, relations $\sum_{j \in M_i(X)} h_j \leq H$ must be satisfied for $i = 1, \ldots, k$. Since $\sum_{j=1}^{3k} h_j = kH$, the latter relations are satisfied only if $\sum_{j \in M_i(X)} h_j = H$ for $i = 1, \ldots, k$. Hence, sets $X_i = M_i(X)$, $i = 1, \ldots, k$, constitute a solution of 3-PARTITION.

"If". Let $X_1, \ldots, X_k$ be a solution of 3-PARTITION. Construct a solution X of OPT, in which a single e-bus of type b serves route r, and $M_i(X) = X_i$, $j = 1, \ldots, k$. For this solution, $V(X) = 1$ and $TP_i(X) = H = o_i$, $i = 1, \ldots, k$, as it is required for part "if". ■

2.2 Randomized heuristic algorithm for OPT

It has been recognized by the Computer Science community that metaheuristic methods such as Simulated Annealing, Tabu Search, Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization and other local search methods are the most appropriate procedures for solving NP-hard problems. We have chosen Particle Swarm Optimization for the problem OPT, because it works well for the problems with a complex solution structure such as the solution structure of the problem OPT, it is easy for implementation because it operates with only three parameters of the candidate solution called position, velocity and value, and it convergence rate is often faster than that of the other metaheuristic methods (Sengupta et al. [6]).

Denote by $\mathcal{X}$ the set of feasible solutions of the problem OPT. By combining a randomized choice of feasible or infeasible partial solutions with a Particle Swarm Optimization technique, we construct a set of feasible complete solutions $\mathcal{Q} \in \mathcal{X}$, which we expect to contain solutions close to the optimal solution. A formal description of our algorithm, denoted as RH, is given below. A feasible solution, in which no new e-bus is selected, is denoted as Q_0. Steps of the algorithm are performed sequentially, unless it is stated differently. Algorithm employs probabilities which are used to determine characteristics of a solution. These probabilities are control parameters of the algorithm. They can be defined by the decision maker, or set to be the same for all possible values of the same solution characteristic, in which case uniform distribution of the values is assumed. Probabilities can also be adjusted in a computer experiment.

Algorithm RH**.**

Step 1. (Initialization) Set $\mathcal{Q} = \{Q_0\}$. In Steps 2-6, a partial solution Q is generated. It can be extended to feasible or infeasible complete solution.

Step 2. (Generation of a set of routes $R(Q)$ served by at least one new e-bus) Define probability pr_r, $0 \leq pr_r \leq 1$, of including $r \in R$ into $R(Q)$. Generate set $R(Q)$ by using these probabilities such that $|R(Q)| \geq 1$. Define the set of vertices $N(Q) = \{j \mid j \in R(Q)\}$.

Step 3. (Generation of a set $B_r(Q)$ of e-bus types of new e-buses to serve route $r \in R(Q)$) For each route $r \in R(Q)$, define probability pr_{rb} of employing new e-buses of e-bus type b of new e-buses on route r, $b \in B_r$. Generate sets $B_r(Q)$, $r \in R(Q)$, by using these probabilities such that $|B_r(Q)| \geq 1$ for each $r \in R(Q)$. Generate set $B(Q) = \cup_{r \in R(Q)} B_r(Q)$ and sets $R_b(Q)$ of routes served by at least one e-bus of type $b \in B(Q)$.

Step 4. (Generation of locations for charging stations and determination of charging station types $c_{jrb}(Q)$ to charge old or new e-buses of type b assigned to route r at node $p(j)$) Values $c_{jrb}(Q)$ are initiated as $c_{jrb}(Q) = False$. Let $SR(Q) = \emptyset$. For each e-bus type $b \in B(Q)$ and each route $r \in R_b(Q)$, include into $SR(Q)$ "obligatory" vertices of the set $NM_b \cap r$ and vertices corresponding to nodes with old charging stations $c \in C_b$ assigned to b and r. It is assumed that charging station types $C_{p(j)} \cap C_b$ are given or they are randomly generated for nodes associated with these "obligatory" vertices. For given route r and e-bus type b, the generation process starts by including in $S_{rb}(Q)$ all vertices from $SR(Q)$ which belong to π_r. Include in $SR_{rb}(Q)$ vertex $j_0 \in \pi_r$ if it has not been included in $SR(Q)$. If $|SR_{rb}(Q)| = 1$ and $ei_{r(j_0,j_0)b} = 1$ then randomly choose j from $(j_1, \ldots, j_r)$ and include j in $SR_{rb}(Q)$. If $|SR_{rb}(Q)| = 1$ and there is vertex j_k from $(j_1, \ldots, j_r)$ such that $ei_{r(j_0,j_k)b} = 0$ then choose with certain probability vertex v_l, $l = 1, \ldots, k-1$ and include v_l in $SR_{rb}(Q)$. This probability can be higher for larger l. Let $SR_{rb}(Q) = (j_0^{rb}, \ldots, j_{n(r,b)}^{rb})$ and $n(r,b) \geq 1$. If $ei_{r(j_k^{rb},j_{k+1}^{rb})b} = 1$ for $k = 0, \ldots, n(r,b) - 1$ and $ei_{r(j_{n(r,b)}^{rb},j_1^{rb})b} = 1$ then stop generation process for current r and b. If $ei_{r(j_k^{rb},j_{k+1}^{rb})b} = 0$ for some $k = 0, \ldots, n(r,b) - 1$ then there is vertex j_h in segment $(r(j_k^{rb}, j_{k+1}^{rb})$ of route r such that $ei_{r(j_k^{rb},j_h)b} = 0$. In this case choose with certain probability vertex v_l from π_r between j_k^{rb} and j_{h-1} and include v_l in $SR_{rb}(Q)$. This probability can be higher if the distance from j_k^{rb} to j_h (respectively, from j_h to ${}_{k+1}^{rb}$) is larger. If $ei_{r(j_{n(r,b)}^{rb},j_1^{rb})b} = 0$ then there is vertex j_h in segment $(j_{n(r,b)}^{rb}, j_r)$ or in segment (j_r, j_1^{rb}) of route r such that $ei_{r(j_{n(r,b)}^{rb},j_h)b} = 0$. In this case choose with certain probability vertex v_l from π_r between $j_{n(r,b)}^{rb}$ and j_r or between j_r and j_1^{rb} and include v_l in $SR_{rb}(Q)$. Repeat this process while $ei_{r(j_k^{rb},j_{k+1}^{rb})b} = 0$ or $ei_{r(j_{n(r,b)}^{rb},j_1^{rb})b} = 0$. For each included vertex j define with a certain probability charging station type $c^* = c_{jrb}(Q) \in C_b \cup C_{p(j}$ for node associated with j. Note that feasibility of such a process can be checked at the stage of data input.

Generate sets $S(Q)$, $S_c(Q)$, $C(Q)$ and $B^{pc}(Q)$, which are analogs of the same concepts defined for a feasible solution X. Note that the following relations are guaranteed to be satisfied at the end of this stage:

$$c_{jrb}(Q) \in C_{p(j)} \cap C_b, \; j \in SR_{rb}(Q), \; r \in R_b(Q), \; b \in B(Q),$$

$$ei_{r(j_k^{rb},j_{k+1}^{rb})b} = 1, \; j_k^{rb} \in SR_{rb}(Q), \; k = 0, \ldots, n(r,b) - 1, \; r \in R_b(Q), \; b \in B(Q),$$

$$\sum_{c \in C_b} (nc_{p(j)c} + NC_{p(j)c}(Q)) \geq 1, \; j \in NM_b, p(j) \in S(Q), \; b \in B(Q),$$

which are analogs of the constraints (7), (8) and (10). Note that $NC_{p(j)c}(Q)$ in the last relation is not determined explicitly, but the relation is satisfied by the definition of Step 4.

Step 5. (Determination of new edges linking locations of charging stations with transformers, numbers $NC_{pc}(Q)$ of new charging stations and numbers $NV_{rb}(Q)$

of new e-buses) For each node $p \in S(Q)\backslash NO$, denote the set of transformer nodes connected to node p according to the partial solution Q as $L_p(Q)$. Denote by $M_q(Q) = \{p \in S(Q) \mid q \in L_p(Q)\}$ the set of new non-transformer nodes linked with transformer node q.

Sets $L_p(Q)$, $p \in S(Q)\backslash NO$, sets $M_q(Q)$, $q \in T$, set $T(Q)$ of new transformers, numbers $NC_{pc}(Q)$ of new charging stations and numbers $NV_{rb}(Q)$ of new e-buses are determined as a solution of the following problem.

$$\max V(Q), \tag{13}$$

subject to $Q \in \mathcal{D}$, where

$$V(Q) = \sum_{r \in R(Q)} v_r(Z_r(Q)),$$

$$CC(Q) = \sum_{p \in S(Q)\backslash NO} \sum_{q \in L_p(Q)} cl_{qp} + \sum_{q \in T(Q)\backslash TO} cb_q + \sum_{c \in C(Q)} \sum_{p \in S_c(Q)} cc_c^{cap} NC_{pc}(Q) + \sum_{r \in R(Q)} \sum_{b \in B_r(Q)} cv_b^{cap} NV_{rb}(Q),$$

$$OC(Q) = \sum_{c \in C(Q)} \sum_{p \in S_c(Q)} cc_c^{ope} NC_{pc}(Q) + \sum_{r \in R(Q)} \sum_{b \in B_r(Q)} cv_{rb}^{ope} NV_{rb}(Q),$$

$$Z_r(Q) = \sum_{b \in B_r(Q)} cap_b NV_{rb}(Q),$$

$$BN_{pc}(Q) = \left\lceil \sum_{r \in R^p(Q)} \sum_{b \in B^{pc}(Q)} \frac{ct_{pbc}}{ATE_r(Q)} \right\rceil, \ p \in NS \cap (S(Q) \cup NO),$$

$$ATE_r(Q) = 1 \Big/ \left(\sum_{b \in B_r(Q)} \frac{NV_{rb}(Q)}{d_r + \sum_{j \in SR_{rb}(Q)\backslash ND} ct_{p(j)bc^*}} + \frac{nbo_r}{do_r} \right), \ c^* = c_{jrb}(Q),$$

$$AT_r(Q) = 1 \Big/ \left(\sum_{b \in B_r(Q)} \frac{NV_{rb}(Q)}{d_r + \sum_{j \in SR_{rb}(Q)\backslash ND} ct_{p(j)bc^*}} + \frac{nbo_r}{do_r} + \frac{nvc_r(X)}{dc_r} \right), \ c^* = c_{jrb}(Q),$$

$$nvc_r(Q) = \sum_{b \in VC_r(Q)} nvc_{rb}(Q),$$

$$VC_r(Q) = \{b \mid b \in VC_r, nvc_{rb}(Q) > 0\},$$

$$nvc_{rb}(Q) = \left\lceil \frac{\max\{0, pas_r - Z_r(Q)\} nvc_{rb}}{pas_r} \right\rceil, b = 1, \ldots, b^* - 1,$$

$$nvc_{rb^*}(Q)=\Big\lceil\Big\lceil\frac{\max\{0,pas_r-Z_r(Q)\}-\sum_{b=1}^{b^*-1} nvc_{rb}(Q)cap_b}{cap_{b^*}}\Big\rceil\Big\rceil,\ nvc_{rb}=0, b=b^*+1,\dots,|VC_r|,$$

$$\text{where } cap_1 \le \dots \le cap_{|VC_r|}, VC_r = \{1,\dots,|VC_r|\},$$

$$\sum_{b=1}^{b^*-1} nvc_{rb}(Q)cap_b < \max\{0,pas_r-Z_r(Q)\} \text{ and } \sum_{b=1}^{b^*} nvc_{rb}(Q)cap_b \ge \max\{0,pas_r-Z_r(Q)\}.$$

$$BN_{pc}(Q) = \Big\lceil\Big\lceil \sum_{r\in R^p(Q)} \sum_{b\in B^{pc}(Q)} \frac{(nv_{rb}+NV_{rb}(Q))ct_{pbc}}{t_p}\Big\rceil\Big\rceil,\ j \in ND,\ p(j) \in NS \cap S(Q).$$

The feasible domain $\mathcal{D}$ is defined by the following constraints.

$$CC(X) \le ucc, \tag{14}$$

$$OC(X) \le uoc, \tag{15}$$

$$Z_r(Q) \le pas_r + \min_{b\in B_r(Q)}\{cap_b\} - 1,\ r \in R(Q), \tag{16}$$

$$Z_r(Q) \ge \min_{b\in VC_r}\{cap_b\},\ r \in R(Q), \tag{17}$$

$$TP_q(Q) + oo_q \le o_q, q \in T(Q), \tag{18}$$

$$AT_r(Q) \le ut_r,\ r \in R(Q), \tag{19}$$

$$nc_{pc} + NC_{pc}(Q) \le uc_{pc},\ p \in S_c(Q),\ c \in C(Q), \tag{20}$$

$$nc_{pc} + NC_{pc}(Q) = BN_{pc}(Q),\ p \in S(Q) \cup NO,\ c \in C(Q). \tag{21}$$

A Particle Swarm Optimization (PSO) technique is used to solve the above problem, see Clerc [1], Kennedy and Eberhart [4] and Pedersen and Chipperfield [5]. PSO is a metaheuristic, which searches among candidate solutions of an optimization problem. Each solution is considered as a *particle* in a *swarm*. For the problem (13)-(21), the solution process starts with a group of random particles (Q, Q_i), $i = 1,\dots,k$, where Q_i is an extension of the partial solution Q determined in Steps 1-4 by a solution of the problem (13)-(21). Thus, (Q, Q_i) is a complete solution of Opt.

Each particle is associated with its *position*, *velocity* and *value*. Velocity determines the increment of the position when passing from one iteration of the PSO algorithm to the next. Let $NV_{rb}^{\max} = \lceil pas_r/cap_b\rceil$. In our algorithm, position of a particle (Q, Q_i) is determined by a set $NV(Q, Q_i) := \{NV_{rb}(Q, Q_i) \mid r \in R_b(Q),\ b \in B(Q)\}$, where the number of e-buses $NV_{rb}(Q, Q_i)$ is selected randomly from the set $\{0, 1,\dots, NV_{rb}^{\max}\}$. Velocity of a particle (Q, Q_i) is determined as a set of numbers $W := \{W_{rb}(Q, Q_i) \mid r \in R_b(Q),\ b \in B(Q)\}$, where $W_{rb}(Q, Q_i)$ is selected randomly from the set $\{-NV_{rb}^{\max}, -NV_{rb}^{\max}+1, \ldots, NV_{rb}^{\max}\}$.

Value of a particle (Q, Q_i) is determined as $V(Q, Q_i)$. It is initialized as $V(Q, Q_i) := -\infty$.

PSO operates with the *best known positions* of each particle and the swarm as a whole over all past iterations of the algorithm. In our implementation, the best known position of a particle (Q, Q_i) is denoted as $NVB(Q, Q_i) = \{NVB_{rb}(Q, Q_i) \mid r \in R_b(Q),\ b \in B(Q)\}$ and it is initialized as the value $NV(Q, Q_i)$ in the first iteration. The best known position of the swarm is the position of its "best particle", denoted as (Q, Q_{i^*}). The best particle has the largest value $V(Q, Q_{i^*})$. The best known position is denoted as $NVG = \{NVG_{rb} \mid r \in R_b(Q),\ b \in B(Q)\}$.

Define the number of iterations of the PSO algorithm. In each iteration, perform the following steps (a)-(e) for each particle (Q, Q_i), $i \in \{1, \ldots, k\}$.

(a) Pick control parameters u_p and u_g as random rational numbers from the interval $(0, 1)$. Determine control parameters ω, φ_p and φ_g. At present, values $\omega = 0.729$ and $\varphi_p = \varphi_g = 1.49445$ are used.

(b) For each pair (r, b), $r \in R_b(Q)$, $b \in B(Q)$, update velocity such that $W_{rb}(Q, Q_i) := \omega W_{rb}(Q, Q_i) + \varphi_p u_p (NVB_{rb}(Q, Q_i) - NV_{rb}(Q, Q_i)) + \varphi_g u_g (NVG_{rb} - NV_{rb}(Q, Q_i))$.

(c) Pick random rational numbers λ_{rb}, $r \in R_b(Q)$, $b \in B(Q)$, from the interval $(0, 1)$. If $\frac{\omega}{\omega + \varphi_p C_p + \varphi_g u_g} \le \lambda_{rb} < \frac{\omega + \varphi_p u_p}{\omega + \varphi_p u_p + \varphi_g u_g}$, then update $NV_{rb}(Q, Q_i) := NV_{rb}(Q, Q_i) + \lfloor \omega W_{rb}(Q, Q_i) \rfloor$. If $\lambda_{rb} \ge \frac{\omega + \varphi_p u_p}{\omega + \varphi_p u_p + \varphi_g u_g}$, then update $NV_{rb}(Q, Q_i) := NV_{rb}(Q, Q_i) + \lceil \omega W_{rb}(Q, Q_i) \rceil$.

(d) For each node $p \in S(Q) \backslash NO$, denote the set of transformer nodes connected to node p according to the partial solution Q as $L_p(Q)$. Denote by $M_q(Q) = \{p \in S(Q) \mid q \in L_q(Q)\}$ the set of new non-transformer nodes linked with transformer node q. Sets $L_p(Q)$, $p \in S(Q) \backslash NO$, sets $M_q(Q)$, $q \in T$, and set $T(Q)$ of new transformers are randomly determined as follows. For each node $p \in S(Q) \backslash NO$, define probability pr_{qp} of linking transformer node $q \in TE_p$ with p. This probability can be higher for larger output power

o_q and it can be higher for smaller cost $cl_{qp} + cb_q y_q$, where $y_q = 1$ if $q \notin TO$ and $y_q = 0$ if $q \in TO$. Link each node $p \in S(Q) \backslash NO$ with m nodes $q \in T$.

(e) *Calculate $CC(Q, Q_i)$ and $OC(Q, Q_i)$. If $CC(Q, Q_i) \leq ucc$ and $OC(Q, Q_i) \leq uoc$, then perform the following computations. Calculate new value $V(Q, Q_i)$ of the particle (Q, Q_i). If it is increased, then update $NVB(Q, Q_i) := NV(Q, Q_i)$. If the new value $V(Q, Q_i)$ exceeds $V(Q, Q_{i^*})$, then re-set $NVG := NV(Q, Q_i)$ and $(Q, Q_{i^*}) = (Q, Q_i)$.*

Let (Q, Q_{i^*}) be the best particle at the end of the last iteration of the PSO algorithm. If $V(Q, Q_{i^*}) > -\infty$, then re-set $\mathcal{Q} := \mathcal{Q} \cup \{(Q, Q_{i^*})\}$. If computational time permits, then perform Step 2, else perform Step 6.

Step 6. Output set $\mathcal{Q}$, several appropriate solutions of this set or a single solution Q^* such that $V(Q^*) = \max_{Q \in \mathcal{Q}} V(Q)$. ■

2.3 Computer implementation

Algorithm RH is implemented in C++ for Windows. It can be used as an executable file *mobopt.exe* or as a DLL-file *mobopt dll.dll*. These files can be used on a PC of a standard configuration. Parameters of the command line for *mobopt.exe* are:

- Full name of directory with input data.
- Full name of directory with configuration file *probl.ini*.

For example: *d:/gn/soft/bat_dll/mobopt.exe d:/gn/soft/mobility/mobopt/Minsk/4 d:/gn/soft/mobility/mobopt*, where *d:/gn/soft/bat_dll* is the directory with *mobopt.exe*, *d:/gn/soft/mobility/mobopt/Minsk/4* is the directory with the input data, and *d:/gn/soft/mobility/mobopt* is the directory with the configuration file *probl.ini*.

From Python *mobopt.exe* can be executed in the following way:

import subprocess

argexe='d:/gn/soft/mobility/mobopt/bc/mobopt.exe'

arg1=' d:/gn/soft/mobility/mobopt/Minsk/4'

arg2=' d:/gn/soft/mobility/mobopt/bc'

args = argexe + arg1+ arg2

```
p=subprocess.Popen(args, shell = False)
p.wait()
ret=p.poll()
```

File *moboptdll.dll* contains function *MOBOPT*, whose prototype is *int MOBOPT(char * dir,char * dir_ini)*, where *dir* is the full name of the directory with the input data and *dir_ini* is the full name of the directory with the configuration file *probl.ini*. The return code of the function *MOBOPT* is equal to 0 if the optimization was successful. In this case, all the output information is placed into the file *solution.out* in the text format and in the file *solution.json* in the JSON format in the directory *dir*. If the return code is not 0, then the corresponding error information is placed into the file *errors.out* in the directory *dir*. An example of calling the function *MOBOPT* from Python (32-bit) is given below.

```
import ctypes
mobDll=ctypes.WinDLL(”d:/gn/soft/bat_dll/moboptdll.dll”)
from ctypes import *
p1=create_string_buffer(b”d:/gn/soft/mobility/mobopt/Minsk/4”)
p2=create_string_buffer(b”d:/gn/soft/mobility/mobopt/bc”)
ret=mobDll.MOBOPT(p1,p2)
```

File *probl.ini* is used for setting the following parameters:

- $json$ – format of the input data, $json \in \{0, 1, 2\}$, where

  $json = 0$ if the input data are in the text format,

  $json = 1$ if the input data are in the JSON format in separate files,

  $json = 2$ if the input data are in the JSON format in one file.

  $json = 2$ is the default value.

- nit – maximum number of iterations of RH, $nit = 10000$ is the default value.

- max_time – maximum calculation time in seconds, $max\_time = 600$ is the default value.

- m – number of links of any charging station location with the transformers, $m = 1$ is the default value.

- w – control parameter ω of the PSO method, $w = 0.729$ is the default value.
- fiP – control parameter φ_p of the PSO method, $fiP = 1.49445$ is the default value.
- fiG – control parameter φ_g of the PSO method, $fiP = 1.49445$ is the default value.

2.4 Formats of the input files

Two formats of the input files are implemented. One of them is the JSON format, see *http://www.json.org/index.html* for a description, and the other is the simple text format. If the input parameter $json = 2$, then the file *problem.json* is transformed into the following files: *probl.json, stations.json, buses.json, cbuses.json, nodes_st.json, graph.json, transf.json, routes.json, nodes_nm.json, nodes_ch_time.json, croutes.json, tdepots.json, buses1.json* and *buses2.json*. Then, each of these files is converted into the corresponding text file. Finally, the data from the text files are imported and analyzed for errors. If there are errors, then the information about them is placed into the file *errors.out* in the directory specified by the parameter *dir*.

2.4.1 JSON format of the input file

If the input data are prepared in the JSON format in separated files, then their names must be the following: *probl.json, stations.json, buses.json, cbuses.json, nodes_st.json, graph.json, transf.json, routes.json, nodes_ch_time.json,* and *tdepots.json*. File *croutes.json* is created only if there are routes already served by e-buses. File *nodes_nm.json* is prepared only if the sets NM_b are non-empty. Files *buses1.json* and *buses2.json* are prepared only if the sets B_1 and B_2, respectively, are non-empty.

File *probl.json* includes values of the following parameters: m – number of links of any location with a new charging station with the transformer nodes, *ucc* – upper bound on the total capital cost, *uoc* – upper bound on the total operating, depreciation and energy cost, and *dtp* – duration of the decisive time period. For example:

```
{
"m": 2,
"ucc": 10000000,
```

"uoc": 5000000,

"dtp": 180

}

File *stations.json* describes the set C of charging stations and defines values of the following parameters: *fn_c* – full name of the charging station type c, *sn_c* – short name, *po_c* – nominal power of any station, *cc_cap_c* – capital cost of any station, *cc_ope_c* – operating and depreciation cost of any station. For example:

{ *"C":* [{

"fn_c": "Charging station 1",

"sn_c": "CS1",

"po_c": 200,

"cc_cap_c": 250000,

"cc_ope_c": 5000

}

]}

File *buses.json* describes the set B and defines values of the following parameters: *fn_b* – full name of the e-bus type b, *sn_c* – short name, *cap_b* – passenger capacity of any e-bus, *cv_cap_b* – capital cost of any e-bus, and *C_b* – array of the short names of the eligible charging stations. For example:

{ *"B":* [{

"fn_b": "Vitovt Max Electro E433",

"sn_b": "E433",

"cap_b": 153,

"cv_cap_v": 475000,

"C_b": [*"CS1"*]

}

]}

File *cbuses.json* describes the set VC of the conventional vehicle types and defines values of the following parameters: *fn_b* – full name of the conventional vehicle type b, *sn_c* – short name, and *cap_b* – passenger capacity of any vehicle. For example:

```
{ ”VC”: [{
”fn_b”: ”Diesel bus MAZ-103”,
”sn_b”: ”M103”,
”cap_b”: 100
}
]}
```

File *nodes_st.json* describes the set NP of the parent nodes and defines values of the following parameters: *fn_p* – full name of the parent node p, *sn_p* – short name, *C_p* – array of the charging station types eligible for opening at p, *nc_pc* – array of the numbers nc_{pc} of old charging stations at p, *uc_pc* – array of the upper bounds uc_{pc} on the number of charging stations of type c to be opened at p. For example:

```
{ ”NP”: [{
”fn_p”: ”Vaneeva”,
”sn_p”: ”V”,
”C_p”: [”CS1”],
”nc_pc”: [1],
”uc_pc”: [4]
}
]}
```

File *graph.json* describes the set NN of the network G and defines values of the following parameters: *fn_j* – full name of the node j, *sn_j* – short name, *sn_j_p* – short name of the parent node $p(j)$, and *type_j* – type of the node (1 for depots, 2 for terminal stops and 3 for regular stops). For example:

```
{ ”NN”: [{
”fn_j”: ”Vaneeva-Depot”,
”sn_j”: ”Vaneeva-D”,
”sn_j_p”: ”V”,
”type_j”: 1
}
]}
```

File *transf.json* describes the set T of transformer nodes and defines values of the following parameters: *fn_q* – full name of the transformer node q, *sn_q* – short name, *o_q* – transformer electrical output power, *cb_q* – transformer capital (building) cost, *oo_q* – transformer electrical power that is used to supply old charging stations, *sn_qp* – – array of the short names of eligible non-transformer nodes p, and *cl_qp* – array of the costs of connection of eligible non-transformer nodes p with the transformer node q. For example:

{ *"EE": [{*

"fn_q": "Vaneeva-Transformer 1",

"sn_q": "Vaneeva-T1",

"o_q": 800,

"cb_q": 0,

"oo_q": 200,

"sn_qp": ["V"],

"cl_qp": [0]

}

]}

File *routes.json* describes the set R of the routes and defines values of the following parameters: *fn_r* – full name of the route r, *sn_r* – short name, *w_r* – preference coefficient, *ut_r* – upper bound on the average length of the traffic interval of all e-buses and conventional buses of any type, *s_r* – array of the short names of the nodes in r, *l_r* – array of the distances between the nodes in r, *B_r* – array of the short names of the e-bus types eligible for the route, *nbo_rb* – array of the numbers nbo_{rb} of old e-buses of type b serving the route, *dm_rb* – array of the single-charge ranges (maximal single-charge travel distances) of e-buses of type b, *ce_rb* – array of the operating, depreciation and energy costs cv_{rb}^{ope}, *d_rb* – array of the durations d_{rb} of any single cycle of any e-bus eligible for the route, *do_rb* – array of the durations do_{rb} of any single cycle of any old e-bus eligible for the route, *VC_r* – array of short names of the types of conventional vehicles serving the route, *nvc_rb* – array of the numbers nvc_{rb} of conventional vehicles serving the route, and *doc_rb* – array of the durations dc_{rb} of any single cycle of conventional vehicles serving the route. For example:

{ *"R": [{*

”fn_r”: ”Railway Station - DS Viasnjanka”,
”sn_r”: ”A1”,
”w_r”: 1,
”ut_r”: 18,
”s_r”: [”Vaneeva-D”,”Kira-T”,”Vias-T”,”Kira-T”],
”l_r”: [7,9,9],
”B_r”: [”E433”,”E420”,”E321”,”E490”,”T32100D”,”T42003D”],
”nbo_rb”: [4,0,0,0,0,0],
”dm_rb”: [15,20,30,25,16,15],
”ce_rb”: [268880,209520,209520,168400,209520,209520],
”d_rb”: [60,60,60,60,60,60],
”do_rb”: [72,72,72,72,72,72],
”VC_r”: [”M103”,”M1035”],
”nvc_rb”: [3,2],
”doc_rb”: [60,60]
}
]}

File *nodes_ch_time.json* describes charging times for the set NP of the parent nodes and defines: *sn_p* – short name of the parent node p, and *ct_pbc* – array of the charging times ct_{pbc}, $c \in C_p$, $b \in B_c$. For example:

{ ”CT_NP”: [{
”sn_p”: ”V”,
”ct_pb_1”: [6,6,10,6,40,30]
}
]}

File *croutes.json* is formed if there are routes that are already served by e-buses. It describes charging stations that are already opened at the nodes belonging to a route $r \in R$ and defines: *sn_r* – short name of the route, and *c_r* – array of the short names of the existing charging stations for each node from $\pi_r = (j_0, j_1, \ldots, j_r)$. If no station is opened at a node j_k, then string ”-1” is used as the corresponding name. For example:

{ *"CO": [{*

"sn_r": "A1",

"c_r": ["CS1","CS1","CS1"]

}

]}

File *tdepots.json* describes the set ND of depot nodes and defines: *sn_j* – short name of the depot node j, and *t_depot_j* – duration t_j^{tdepot} of a time interval of maximum length, in which all e-buses assigned to the depot at node $j \in ND$ are in this depot. For example:

{ *"ND": [{*

"sn_j": "Vaneeva-D",

"t_depot_j": 240

}

]}

File *nodes_nm.json* describes sets NM_b of "obligatory" nodes for an e-bus of type b and defines: *sn_b* – short name of the e-bus type b, and *sn_b_j* – array of the short names of the nodes from NN such that if j belongs to r to be served by an e-bus of type b then at least one charging station of type $c \in C_b$ must be opened at node $p(j)$. For example:

{ *"NM_b": [{*

"sn_b": "E433",

"sn_b_j": ["Kira-D","Vias-S"]

}]}

File *buses1.json* describes the set B_1 of e-buses with batteries which have enough capacity to drive with a single charge at the corresponding depot during the day and defines *sn_b* – array of the short names of the e-bus types from B_1. For example:

{ *"B_1": {*

"sn_b": ["E420"]

}

}

File *buses2.json* describes the set B_2 of e-buses with one charge at the corresponding depot and one charge at a non-depot node during the day and defines *sn_b* – array of the short names of the e-bus types from B_2. For example:

{ "B_2": {

"sn_b": ["E333"]

}

}

2.4.2 Text format of the input files

If data are prepared in the text format, then the following files must be prepared: *probl.txt, stations.txt, buses.txt, cbuses.txt, nodes_st.txt, graph.txt, transf.txt, routes.txt, nodes_ch_time.txt,* and *tdepots.txt.* File *croutes.txt* is created only if there are routes already served by e-buses. File *nodes_nm.txt* is prepared only if the sets NM_b are non-empty. Files *buses1.txt* and *buses2.txt* are prepared only if the sets B_1 and B_2 are non-empty, respectively. Each file can include comments. The comments must start with the symbols // and be placed in the beginning of the file. The main body of the file starts with a new line immediately after the comments. Values of the different input parameters are separated by comma.

File *probl.txt* consists of one row with the following values: number m of links of any new charging station with the transformer nodes, upper bound *ucc* on the total capital cost, upper bound *uoc* on the total operating, depreciation and energy cost, and duration *dtp* of the decisive time period. For example:

2,10000000,5000000,180

File *stations.txt* consists of one row for each element of the set C. Each row contains: full name of the charging station type, short name of the charging station type, nominal power po_c, capital cost cc_c^{cap} and operating and depreciation cost cc_c^{ope}. For example:

Charging station 1,CS1,200,250000,5000

File *buses.txt* consists of two rows for each element of the set B. The first row contains: full name of the e-bus type, short name of the e-bus type, passenger capacity cap_b, capital cost cv_b^{cap}. The second row contains the short names of the eligible charging stations for the e-bus type. For example:

Vitovt Max Electro E433,E433,153,475000

CS1

File *cbuses.txt* consists of one row for each element of the set VC of conventional vehicle types. Each row contains: full name of the conventional vehicle type, short name of the conventional vehicle type, and passenger capacity cap_b. For example:

Diesel bus MAZ-103,M103,100

File *nodes_st.txt* consists of 4 rows for each element of the set NP of parent nodes. The first row contains full name of parent node p and short name of parent node p. Next rows contain short names of the eligible charging stations, numbers nc_{pc} of the old charging stations, and the upper bounds uc_{pc}. For example:

Vaneeva,V

CS1

1

4

File *graph.txt* consists of one row for each element of the set NN. Each row contains: full name of the node, short name of node, short name of the parent node and the type of the node (1 for the depot node, 2 for the terminal node and 3 for the regular en route node. For example:

Vaneeva-Depot,Vaneeva-D,V,1

Kira-Terminal,Kira-T,Kira,2

Ch.Ri-Stop,Ch.Ri-S,Ch.Ri,3

File *transf.txt* consists of 3 rows for each element of the set T. The first row contains: full name of the transformer, short name of the transformer, transformer electrical output power o_q, transformer capital (building) cost cb_q (0 if it has already been built), and already used transformer electrical power oo_q to supply old charging stations. The second row contains short names of the non-transformer nodes eligible for linking with the transformer node. The third row contains costs of connection of nodes in the second row with the transformer node (0 if the corresponding connection exists). For example:

Vias-Transformer 1,Vias-T1,800,0,200

Vias

0

File *routes.txt* consists of 12 rows for each route of the set R. The rows contain:

- row 1: full name of the route, short name of the route, preference coefficient w_r, and upper bound ut_r.

- row 2: sequence $\pi_r = (j_0, j_1, \ldots, j_r)$ of short names of the nodes.
- row 3: distances between stops (nodes).
- row 4: short names of e-buses eligible for the route.
- row 5: numbers nbo_{rb} of old e-buses serving the route.
- row 6: maximal distance of e-buses eligible for the route, without recharging.
- row 7: operating, depreciation and energy cost cv_{rb}^{ope}.
- row 8: durations d_{rb}.
- row 9: durations do_{rb}.
- row 10: short names of conventional vehicles serving the route.
- row 11: numbers nvc_{rb} of conventional vehicles.
- row 12: durations dc_{rb}.

For example:

Railway Station - DS Viasnjanka,A1,1,18

Vaneeva-D, Kira-T, Vias-T, Kira-T

7,9,9

E433,E420,E321,E490,T32100D,T42003D

4,0,0,0,0,0

15,20,30,25,16,15

268880,209520,209520,168400,209520,209520

60,60,60,60,60,60

72,72,72,72,72,72

M103,M1035

3,2

60,60

File *nodes_ch_time.txt* consists of $1 + |C_p|$ rows for $p \in NP$. The first row contains short name of parent node p. Each row from $|C_p|$ rows contain charging times ct_{pbc} at charging station of type $c \in C_p$ for each $b \in C_b$. For example:

V

6,6,10,6,40,30

File *croutes.txt* is created only if there are routes that are already served by the e-buses. It describes charging stations that have already been opened at the nodes belonging to the routes from R. It consists of two rows for each route. The first row contains short name of the route r. The second row contains short names of the charging stations for each node from $\pi_r = (j_0, j_1, \ldots, j_r)$. If no station is opened for some node j_k, then symbols *"-1"* are used as the short name. For example:

A1

CS1,CS1,-1

File *tdepots.txt* consists of one row for each element from the set ND of depot nodes. The row for node j contains short name of the depot node and the duration (min) t_j^{depot}. For example:

Vaneeva-D,240

File *stops obl.txt* consists of two rows for each set NM_b of "obligatory" nodes for e-bus of type b. The first row contains only the short name of the e-bus type b. The second row contains short names of the nodes $j \in NN$. If node j belongs to a route to be served by an e-bus of type b, then at least one charging station of type $c \in C_b$ must be opened at node $p(j)$. For example:

E433

Kira-D, Vias-S

File *buses1* consists of one row with short names of e-bus types from the set B_1 with batteries which have enough capacity to drive with a single charge at a depot during the day. For example:

E420

File *buses2.txt* consists of one row with short names of e-bus types from the set B_2 with batteries which have enough capacity to drive with one charge at a depot and one charge at a non-depot node during the day. For example:

E333

2.5 Formats of the output files

Two formats of the output files are implemented. One of them is the JSON format, and the second is the simple text format.

2.5.1 JSON format of the output file

Object *Solutions* defines output objects *CR* and *X* for each of the obtained solutions. The object *CR* defines values of the following parameters: *V* – total value $V(X)$, *CC* – total capital cost $CC(X)$, and *OC* – total operating, depreciation and energy cost $OC(X)$. For example:

{ *"CR" : { "V": 1260, "CC": 7.88e+006, "OC": 3.01224e+006* } }

Object *X* defines values of the following parameters: *R_X* – routes selected for replacement of conventional vehicles by e-buses and their parameters, *S* – charging stations to be opened and transformers to be built, and *T* – power requirements for the transformers.

Object *R_X* defines values of the following parameters for each route *r*: *r* – full name of the route, *ATE_r* – length of the traffic interval, *Z_r* – total passenger capacity of the new e-buses, *NV_r* – total number of the new buses, *B_r* – full names of the e-bus types, *NV_rb* – numbers of new e-buses of each type *b*, *CH* – nodes for recharging each e-bus and types of the charging stations to be opened, *t* – recommended departure order for all the vehicles. For example:

{ *"R_X":* [{ *"r": "Slavinskogo - Old Airport", "ATE_r": 7, "Z_r": 1260, "NV_r": 16, "B_r":* [*"Vitovt Electro E420","Model E321","Vitovt Mini Electro E490"*], *"NV_rb":* [*1,6,9*], *"CH":* [{*"b": "Vitovt Electro E420", "Nbc_r":*[{*"c": "Charging station 1","j": Kazlova-Depot(Kazlova)*}, {*"c": "Charging station 1","j": "KalSlav-Terminal(KalSlav)"*}, {*"c": "Charging station 1","j": "Aera1-Terminal(Aera1)"*}]}], *"t":* [*"E321", "E490", "E490", "E321", "E490", "E490", "E321", "E490", "E321", "E420", "E490", "E490", "E321", "E490", "E490", "E321"*] }] }

Object *S* defines values of the following parameters: *p* – full name of the parent node *p*, *c* – full type names of the opened charging stations, *NC_pc* – numbers of the opened charging stations, *L_p* – full names of transformer types to be connected with the parent node. For example:

{ *"S": [{"p": "Kazlova","c": ["Charging station 1"],"NC_pc": [1], "L_p": ["Kazlova-Transformer 1","Kazlova-Transformer 2"]}] }*

Object T defines the list of full names of the types of the new transformers. For example:

{ *"T": ["Vaneeva-Transformer 1", "Vaneeva-Transformer 2"] }*

Object TP defines values of the following parameters: q – full name of the transformer type, and TP_q – total transformer power requirement. For example, or the Minsk case:

{ *"TP": [{"q": "Vaneeva-Transformer 1", "TP_q": 200}, {"q": "Vaneeva-Transformer 2", "TP_q": 200}] }*

2.5.2 Text format of the output files

All the obtained solutions are placed into the unique file *solution.out*. The output for each solution includes: values $V(x)$, $CC(x)$ and $OC(x)$; selected routes for the replacement of the conventional vehicles by the e-buses; charging stations to be opened and transformers to be built; power requirement for each transformer.

For each selected route, the output is:

- Selected e-bus types.
- Numbers of new e-buses of each type.
- Total passenger capacity of new e-buses.
- Capital cost of new e-buses.
- Operating and energy cost of new e-buses.
- Conventional vehicle types remained in operation.
- Numbers of the remained conventional vehicles of each type.
- Average Length of the traffic interval for all vehicles serving the route.
- Parent nodes at which new charging stations have to be opened.
- Recommended order of departure of all the vehicles.

2.6 Minsk case of Opt

The experimental software implementing algorithm RH was used to solve instances of the problem Opt for a set of public transport routes in the city of Minsk. In these instances, super-capacitor fast-charging technology is used in batteries of all e-buses. Therefore, there are no e-buses with slow-charging high-capacity batteries. One of these instances is described below.

There are two e-bus depots: $Vaneeva$ (V) and $Kazlova$ (K). Time intervals for e-buses to stay in the depots are the same: $t_j^{depot} = 240$ minutes for $j \in \{V, K\}$. It is the night time period. One old charging station is opened at depot $Vaneeva$ and each of the e-bus stops $Kira$, $Vias$, $Druz$, $Siar$ and $Daug$. New and old charging stations are of the same type and they can charge an e-bus of any type. Upper bounds on charging times, ct_{jbc}, depend on the e-bus type and they do not depend on the location of the charging station. Upper bounds on the number of charging stations are the same for all locations: $uc_{jc} = 4$. Capital cost of one charging station is $cc_c^{cap} = 125000$ € and operating and depreciation cost of one charging station per year is $cc_c^{ope} = 5000$ €. The power of one charging station is $po_c = 260$ Kw. Parameters of the used old vehicles are the following. Diesel bus MAZ-103 (M103): 100 passengers, 25 litres/100 km. Diesel bus MAZ-105 (M105): 160 passengers, 33 litres/100 km. Trolleybus model 420 (T420): 115 passengers. Trolleybus model 333 (T333): 170 passengers.

Any e-bus can feasibly drive from any of the two depots to the closest stop of any route, eligible for opening a charging station. All eligible stops are terminal stops of some routes, and they are visited by e-buses in both direct and backward directions of their routes intersecting at these stops. The value function is defined as $V_r(Z_r) = Z_r$ for any route. The total distance travelled by one e-bus of any type in a year is assumed to be 80000 km. The operating and depreciation costs of e-buses per 100 km range 200-320 €. The energy consumption of e-buses per 100 km range 150-230 kWh. The cost of energy is 7 € per 100 kWh. Upper bound ut_r on the average length of the traffic interval of new and old e-buses of all types on route r is equal to the length of the traffic interval of conventional passenger vehicles on route r.

Locations for charging stations and their names are 1($Kira$), 2($Vias$), 3($Akva$), 4($RKMC$), 5($Ch.Ri$), 6($Losh2$), 7($Masu$), 8($Vaneeva$), 9($DS\ Sera$), 10($AV\ Cant$), 11($YZap$), 12($Daug$), 13($Siar$), 14($Druz$), 15($Kara$), 16($Suh5$), 17($Mal4$), 18($KalSlav$), 19($Aera1$), 20($Vsnin$), 21($Liab$), 22($Pl.Y.Kol$), 23($Zdan$), 24 ($Kazlova$). Two transformers $(i,1)$ and $(i,2)$ are associated with each charging station location i, $i = 1,\dots,24$. Charging station location i can only be linked with transformers $(i,1)$ and $(i,2)$ for all i. The costs of building a transformer are $cb_{i,1} = cb_{i,2} = 20000$€ for $i = 3,4,20,21,22,23$. The existing (old) transformers are from the set $TO = \{(i,1),(i,2) \mid i = 1,2,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,24\}$. The cost of linking charging station location i with transformer $(i,1)$ or $(i,2)$ is the same and it is equal to $cl_{(i,1)i} = cl_{(i,2)i} = 5000$€. The output power of any transformer is 800 kW. The upper bounds on the capital and operating, depreciation and energy costs are $ucc = 10000000$€ and $uoc = 5000000$€, respectively. The remaining input data are given in Tables 1, 2 and 3. Passenger capacity, range of one charge drive, cost and time are measured in persons, kilometres, euros (€) and minutes, respectively.

Table 1: E-bus types. Input data.

b	Name	Capacity cv_b^{cap}	Range of 1 charge	Char.time ct_{jbc}	Cap.cost cv_{rb}^{cap}	Oper.& Ener.cost cv_{rb}^{ope}, $\forall r$
1	Vitovt Max Electro E433	153	15	6	475000	268880
2	Vitovt Electro E420	87	20	6	350000	209520
3	Model E321	83	30	10	400000	205520
4	Vitovt Mini Electro E490	75	25	6	400000	168400
5	Trolleybus 32100D (E321)	85	15	40	370000	209520
6	Trolleybus 42003D (E420)	85	15	30	400000	209520

Table 2: Routes. Conventional service. Locations to be visited twice are marked with $*$.

ID (depot)	Name	One way route	Length between stops	Duration between stops	Traff. inter. ($= ut_r$)
$1(V)$	$A1$	$(Kira, Vias)$	9	30	5
$2(V)$	$A119c$	$(Kira, Vias^*, Akva)$	(9,4)	(15,10)	20
$3(V)$	$A190e$	$(Kira, Vias^*, RKMC)$	(9,6)	(20,10)	20
$4(V)$	$T5$	$(Kira, Ch.Ri)$	5	20	5
$5(V)$	$T6$	$(Kira, Ch.Ri, Losh2)$	(5,4)	(18,12)	5
$6(V)$	$A69$	$(Kira, Masu)$	11	40	15
$7(V)$	$T20$	$(Kira, V, DS\ Sera)$	(5,4)	(25,15)	10
$8(V)$	$T36$	$(DS\ Sera, AV\ Ca, YZap)$	(5,13)	65	10
$9(V)$	$A46$	$(AV\ Ca, Masu)$	9	35	40
$10(V)$	$T58$	$(AV\ Ca, Masu)$	9	30	15
$11(V)$	$T59$	$(Daug, Siar)$	13	40	10
$12(V)$	$A38$	$(Druz, Kara)$	8	30	20
$13(V)$	$T43$	$(Druz, Siar)$	7	25	15
$14(V)$	$T40$	$(Kara, Druz^*, YZap)$	(7,7)	50	10
$15(K)$	$T63$	$(Druz, YZap)$	9	30	20
$16(K)$	$A50c$	$(Suh5, Druz)$	15	50	20
$17(K)$	$T7$	$(Suh5, AV\ Cant)$	14	45	10
$18(K)$	$T9$	$(Suh5, Druz)$	13	40	15
$19(K)$	$A32c$	$(Mal4, Druz)$	14	45	10
$20(K)$	$A100$	$(KalSlav, Aera1)$	12	45	5
$21(K)$	$A91$	$(Vsnin, Vias^*, KalSlav)$	(2,18)	70	20
$22(K)$	$A73$	$(Siar, Druz^*, Vias^*, Liab)$	(7,9,2)	60	10
$23(K)$	$T22$	$(Kara, Pl.Y.Kol)$	6	20	10
$24(K)$	$A44$	$(Kara, Vias^*, Zdan)$	(10,6)	50	10
$25(K)$	$A136$	$(Kara, Vias^*, Zdan)$	(10,7)	55	30
$26(K)$	$T38$	$(Suh5, KalSlav)$	21	70	10

Table 3: Routes. Input data.

r	$d_r = dc_r$	do_r	nbo_{rb}, b=$E433$	nbo_{rb}, b=$E420$	nvc_{rb}, b=$M103$	nvc_{rb}, b=$M105$	nvc_{rb}, b=$T420$	nvc_{rb}, b=$T333$
1	60	72	4	0	3	2	0	0
2	50	-	0	0	0	2	0	0
3	60	-	0	0	1	2	0	0
4	40	-	0	0	0	0	2	5
5	60	-	0	0	0	0	4	6
6	80	-	0	0	5	0	0	0
7	80	-	0	0	0	0	2	5
8	130	-	0	0	0	0	3	6
9	70	-	0	0	0	2	0	0
10	60	-	0	0	0	0	2	2
11	80	92	6	0	0	0	2	0
12	60	-	0	0	1	2	0	0
13	50	56	4	0	0	0	2	0
14	100	-	0	0	0	0	2	5
15	60	-	0	0	2	2	0	0
16	100	-	0	0	3	3	0	0
17	90	-	0	0	0	0	2	4
18	80	-	0	0	0	0	1	3
19	90	-	0	0	3	6	0	0
20	90	-	0	0	3	6	0	0
21	140	-	0	0	2	5	0	0
22	120	-	0	0	5	7	0	0
23	40	-	0	0	0	0	3	0
24	100	-	0	0	2	5	0	0
25	110	-	0	0	1	2	0	0
26	140	-	0	0	0	0	6	8

Solution. One solution is found. Information about this solution is given in Tables 4-6.

Table 4: Minsk case. Solution value and costs

Routes	Value	Capital cost	Operating, depreciation & energy cost
A1,T59,T43,T9,T7,A46	2684	9915000	4800480

Table 5: Minsk case. Infrastructure

Charging stations	Routes
Vaneeva. Old: 1	A1,T59,T43,A46
Kira. Old: 1	A1
Vias. Old: 1	A1
Masu. New: 1	A46
AV Ca. New: 1	T7,A46
Daug. Old: 1	T59
Druz. Old: 1	T43,T9
Kazlova. New: 1	T9,T7
Suh5. New: 1	T9,T7

Table 6: Minsk case. Fleet

Routes	Old e-buses	New e-buses	Old buses
A1	E433 - 4	E433 - 4	M103 - 1
T59	E433 - 7	E433 - 1	M103 - 1
T43	E433 - 4	E433 - 1	M103 - 1
T9		E433 - 4	M103 - 1
T7		E433 - 6	
A46		E433 - 1, E420 - 1	M103 - 1

3 Efficient algorithm and data structure for DepOpt

Problem DepOpt studied in this section deals with a single depot and a set of e-buses with slow charging batteries, repetitively charging at this depot and serving a given subset of trips each. The problem is to determine the required electric power supplied to the depot by the city power grid, the type and the number of charging stations of this type in the depot, options of e-bus batteries and charging times of each e-bus while it is in the depot such that the total daily cost of charging equipment of the depot, of e-bus batteries and the consumed energy is minimized, provided that the arrival and departure times of e-buses to/from the depot, the dynamic upper bound on the supplied power and functions of charge and discharge of the batteries are addressed.

In the previous Deliverable, problem DepOpt was formulated as

$$\min F(p_D, c, K, \mathbf{b}, \mathbf{u}^{\mathrm{c}}, \mathbf{s}^{\mathrm{a}}) =$$

$$= \frac{cost(p_D)}{365} + \frac{K}{365}(cost^{cap}(c) + cost^{ope}(c)) + \sum_{j\in J} \frac{cost_{b_j}\overline{N}_j \chi_j(s_j^{low})}{365} + \sum_{i=1}^{m-1} c_{ei} \sum_{j\in J} f'_{jc}\delta_j^i, \tag{22}$$

subject to

$$s_j^{ap} = \varphi_{jp}(s_j^{dp}), p = 1, \ldots, n_j, j \in J. \tag{23}$$

$$s_j^{ap} = f_{jc}(u_j^{ap}), p = 1, \ldots, n_j, j \in J, \tag{24}$$

$$s_j^{dp+1} = f_{jc}(u_j^{ap} + u_j^{cp}), p = 1, \ldots, n_j - 1, j \in J, \tag{25}$$

$$s_j^{d1} = f_{jc}(u_j^{dn_j} + u_j^{cn_j}), j \in J, \tag{26}$$

$$u_j^{cp} = \sum_{i\in I_j^p} \delta_j^i, p = 1, \ldots, n_j, j \in J, \tag{27}$$

$$0 \le \delta_j^i \le (t_{i+1} - t_i), i \in I_j^p, p = 1, \ldots, n_j, j \in J, \tag{28}$$

$$\delta_j^i = 0, i \in \{1, \ldots, m-1\} \setminus \cup_{p=1}^{n_j} I_j^p, j \in J, \tag{29}$$

$$\underline{s}_j(b_j) \le s_j^{ap} \le \bar{s}_j(b_j), p = 1, \ldots, n_j, j \in J, \tag{30}$$

$$\underline{s}_j(b_j) \le s_j^{dp} \le \bar{s}_j(b_j), p = 1, \ldots, n_j, j \in J, \tag{31}$$

$$0 \le u_j^{dp} \le \tau_{jc}^{max}, p = 1, \ldots, n_j, j \in J, \tag{32}$$

$$0 \le u_j^{cp} \le \tau_{jc}^{max}, p = 1, \ldots, n_j, j \in J, \tag{33}$$

$$\sum_{j\in J} \delta_j^i \le \min\{k_{ci}(P_i), K\}(t_{i+1} - t_i), i = 1, \ldots, m-1, \tag{34}$$

$$p_D \in \Theta_D, \tag{35}$$

$$c \in C, \tag{36}$$

$$K \le \lfloor p_D/P_c \rfloor, \tag{37}$$

$$b_j \in B_{cj}, j \in J, \tag{38}$$

where the given parameters and the variables are determined in the previous Deliverable. Constraints (23) define SOC level s_j^{ap} of e-bus j at the time of its p-th arrival to the depot if at the time of its p-th departure from the depot it was equal to s_j^{dp}. Constraints (24) define charging time u_j^{ap} required to restore the SOC level s_j^{ap} of the e-bus j from its minimal SOC level s_j^{min}. Relations (25) specify SOC level s_j^{dp+1} of e-bus j at the time of $p+1$-th departure from the depot after its charging over time u_j^{cp} in the interval between p-th arrival to and $p+1$-th departure from the depot. Constraints (26) require that the initial SOC levels s_j^{d1} of all e-buses must be restored prior to their first departure from the depot on the next day. Constraints (27) represent the total charging time u_j^{cp} of e-bus j between its p-th arrival to the depot and $p+1$-th departure from it. Constraints (28) indicate that charging time δ_j^i of the e-bus j in time interval i is positive and does not exceed its duration. Constraints (29) set charging time δ_j^i to zero for e-bus j in the time interval i, when the e-bus is outside the depot. Constraints (30) specify the range of the SOC level s_j^{ap} of e-bus j at its p-th arrival to the depot. Similarly, constraints (31) specify the range of the SOC level s_j^{dp} of e-bus j at its p-th departure from the depot. Constraints (32) require that the charging time u_j^{dp} of e-bus j at its p-th departure is positive and it does not exceed the upper bound. Similarly, constraints (33) require that the charging time u_j^{cp} of e-bus j in the interval between its p-th arrival and $p+1$-th departure is positive and it does not exceed the upper bound. Constraints (34) limit the total charging time of all e-buses J in the time interval i by the available charging time derived for the parallel charging stations from the given supplied power. Constraint (35) restricts the power supplied to the depot by a given range. Constraint (36) indicates that type c of charging station can be selected from given set C. Constraint (37) limits the number of charging stations K by an upper bound derived from the supplied power. Constraints (38) state that the battery type b_j of e-bus j can be selected from a set B_{cj}.

3.1 Computational complexity of DepOpt

In this section, we prove that DepOpt is NP-hard in the strong sense.

Theorem 3 *Problem* DepOpt *is NP-hard in the strong sense even if all e-buses arrive to the depot at the same time, they depart from the depot at the same time, their charging times are* *given constants, the supplied power is unlimited, and the only non-zero cost is the capital cost of the charging stations.*

Proof: Let there be $3k$ e-buses, their charging times be equal to h_i, $i = 1, \dots, 3k$, and satisfy $\sum_{i=1}^{3k} h_i = kH$ and $H/4 < h_i < H/2$, $i = 1, \dots, 3k$. E-buses arrive to the depot at time zero and depart at time H. It is easy to see that the NP-complete in the strong sense problem 3-Partition formulated in Section 2.1 is equivalent to the problem DepOpt which asks for a charging plan whose cost does not exceed kC, where C is the cost of one charging station. ■

3.2 Solution method for DepOpt

We suggest to employ the following four-level decomposition scheme for solving the problem DepOpt, which is illustrated in Fig. 1.

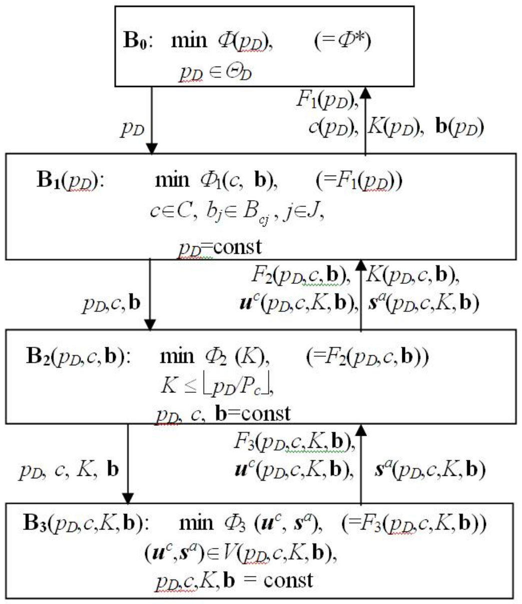
$\mathbf{B_0}$: min $\Phi(p_D)$, $(=\Phi^*)$
$p_D \in \Theta_D$
p_D
$F_1(p_D)$,
$c(p_D)$, $K(p_D)$, $\mathbf{b}(p_D)$
$\mathbf{B_1}(p_D)$: min $\Phi_1(c, \mathbf{b})$, $(=F_1(p_D))$
$c \in C$, $b_j \in B_{cj}$, $j \in J$,
p_D=const
$p_D, c, \mathbf{b}$
$F_2(p_D, c, \mathbf{b})$, $K(p_D, c, \mathbf{b})$,
$\boldsymbol{u}^c(p_D, c, K, \mathbf{b})$, $\boldsymbol{s}^a(p_D, c, K, \mathbf{b})$
$\mathbf{B_2}(p_D, c, \mathbf{b})$: min $\Phi_2(K)$, $(=F_2(p_D, c, \mathbf{b}))$
$K \le \lfloor p_D/P_c \rfloor$,
p_D, c, $\mathbf{b}$=const
p_D, c, K, $\mathbf{b}$
$F_3(p_D, c, K, \mathbf{b})$,
$\boldsymbol{u}^c(p_D, c, K, \mathbf{b})$, $\boldsymbol{s}^a(p_D, c, K, \mathbf{b})$
$\mathbf{B_3}(p_D, c, K, \mathbf{b})$: min $\Phi_3(\boldsymbol{u}^c, \boldsymbol{s}^a)$, $(=F_3(p_D, c, K, \mathbf{b}))$
$(\boldsymbol{u}^c, \boldsymbol{s}^a) \in V(p_D, c, K, \mathbf{b})$,
$p_D, c, K, \mathbf{b}$ = const

Figure 1: Decomposition scheme.

At the lower level, a sub-problem $\mathbf{B}_3(p_D, c, K, \mathbf{b})$ is solved for various fixed sets $(p_D, c, K, \mathbf{b})$. For each such set the optimal charging times $\mathbf{u}^c(p_D, c, K, \mathbf{b})$ of the e-buses and respective SOC levels $\mathbf{s}^a(p_D, c, K, \mathbf{b})$ on their arrivals to the depot are selected from the set $V(p_D, c, K, \mathbf{b})$ to minimize the total daily cost of the e-bus batteries and the consumed energy. The set $V(p_D, c, K, \mathbf{b})$ is determined by the constraints (23)-(34).

At the third level, for a given triple $(p_D, c, \mathbf{b})$ the number $K(p_D, c, \mathbf{b})$ of charging stations c and respective $\mathbf{u}^{\mathrm{c}}(p_D, c, K(p_D, c, \mathbf{b}), \mathbf{b})$, $\mathbf{s}^{\mathrm{a}}(p_D, c, K(p_D, c, \mathbf{b}), \mathbf{b})$ are sought that minimize the total daily cost of these charging stations, given options of e-bus batteries and consumed energy. This sub-problem is referred to as $\mathbf{B}_2(p_D, c, \mathbf{b})$.

At the second level, for a given power p_D supplied to the depot, a sub-problem $\mathbf{B}_1(p_D)$ of selection the type $c(p_D) \in C$ of charging stations and battery options $\mathbf{b}(p_D)= (b_j(p_D)|j \in J)$, $b_j(p_D) \in B_{cj}$, $j \in J$ is solved. The objective function in this sub-problem is the total daily cost of charging stations in the depot, e-bus batteries and consumed energy.

At the upper level, a sub-problem $\mathbf{B}_0$ of one-dimensional search in the discrete interval $[p_D^1, p_D^{\bar{k}}]$ of value p_D is performed to determine the optimal value p_D^* of the electric power supplied to the depot that minimizes the total daily cost of charging infrastructure of the depot (charging stations, transformers, etc), e-bus batteries and consumed energy. The sub-problems are solved as follows.

1. Sub-problem $\mathbf{B}_3(p_D, c, K, \mathbf{b})$(to be solved for a fixed set $(p_D, c, K, \mathbf{b})$):

$$\min \Phi_3(\mathbf{u}^{\mathrm{c}}, \mathbf{s}^{\mathrm{a}}) = \Big\{ \sum_{\mathrm{j}\in\mathrm{J}} \frac{\mathrm{cost}_{\mathrm{b_j}} \overline{\mathrm{N}}_{\mathrm{j}} \chi_{\mathrm{j}}(\mathrm{s}_{\mathrm{j}}^{\mathrm{low}})}{365} + \sum_{\mathrm{i}=1}^{\mathrm{m}-1} \mathrm{c}_{\mathrm{ei}} \sum_{\mathrm{j}\in\mathrm{J}} \mathrm{f}'_{\mathrm{jc}} \delta_{\mathrm{j}}^{\mathrm{i}} \Big\},$$

subject to (23)-(34). The variables to be determined are s_j^{dp}, s_j^{ap}, u_j^{cp}, u_j^{ap} and δ_j^i, $p = 1, \ldots, n_j$, $i = 1, \ldots, m-1$, $j \in J$.

In the sub-problem $\mathbf{B}_3(p_D, c, K, \mathbf{b})$ the distribution of charging time at K charging stations of type c among partially discharged e-buses equipped with the batteries b_j is optimized while these e-buses are in the depot. The objective function of the sub-problem is the daily cost of all e-bus batteries and consumed energy.

We will now show that for a sufficiently general case of piecewise linear functions $\varphi(\cdot)$ and $f(\cdot)$ this sub-problem reduces to a *Mixed Integer Linear Programming* problem. Assume that

$\varphi_{jp}(s) = a_{jp}^k s + b_{jp}^k$ if $\underline{s}_{jp}^k \le s \le \bar{s}_{jp}^k$, $k = 1, \ldots, l_{jp}$,

$\underline{s}_{jp}^{k+1} = \bar{s}_{jp}^k$, $k = 1, \ldots, l_{jp} - 1$, $\underline{s}_{jp}^1 = \underline{s}_j$, $\bar{s}_{jp}^{l_{jp}} = \bar{s}_j$,

and

$f_{jc}(u) = c_{jc}^k u + d_{jc}^k$ if $\underline{u}_{jc}^k \le u \le \bar{u}_{jc}^k$, $k = 1, \ldots, l_{jc}$,

$\underline{u}_{jc}^{k+1} = \bar{u}_{jc}^k$, $k = 1, \ldots, l_{jc} - 1$, $\underline{u}_{jc}^1 = 0$, $\bar{u}_{jp}^{l_{jc}} = \bar{\tau}_{jc}$.

Introduce binary variables w_{jp}^k and continuous variables x_{jp}^k that satisfy the following constraints.

$$\varphi_{jp}(s) = \sum_{k=1}^{l_{jp}} a_{jp}^k x_{jp}^k + \sum_{k=1}^{l_{jp}} b_{jp}^k w_{jp}^k, \tag{39}$$

$$s = \sum_{k=1}^{l_{jp}} x_{jp}^k, \tag{40}$$

$$\sum_{k=1}^{l_{jp}} w_{jp}^k = 1, \tag{41}$$

$$\underline{s}_{jp}^k w_{jp}^k \le x_{jp}^k \le \bar{s}_{jp}^k w_{jp}^k, \tag{42}$$

$$w_{jp}^k \in \{0, 1\}. \tag{43}$$

In a similar way, introduce binary variables v_{jp}^k and continuous variables y_{jp}^k that satisfy the following constraints.

$$f_{jc}(u) = \sum_{k=1}^{l_{jc}} c_{jc}^k y_{jc}^k + \sum_{k=1}^{l_{jc}} d_{jc}^k v_{jc}^k, \tag{44}$$

$$u = \sum_{k=1}^{l_{jc}} y_{jc}^k, \tag{45}$$

$$\sum_{k=1}^{l_{jc}} v_{jc}^k = 1, \tag{46}$$

$$\underline{u}_{jc}^k v_{jc}^k \le y_{jc}^k \le \bar{u}_{jc}^k v_{jc}^k, \tag{47}$$

$$v_{jc}^k \in \{0, 1\}. \tag{48}$$

After (39)-(43) have been incorporated into (23) and (44)-(48) into (24)-(26), the sub-problem $\mathbf{B}_3(p_D, c, K, \mathbf{b})$ becomes a MILP problem.

Denote solution of the sub-problem $\mathbf{B}_3(p_D, c, K, \mathbf{b})$ as $\mathbf{u}^{\mathrm{c}}(p_D, c, K, \mathbf{b})$, $\mathbf{s}^{\mathrm{a}}(p_D, c, K, \mathbf{b})$, and its value as $F_3(p_D, c, K, \mathbf{b})$. Set $F_3(p_D, c, K, \mathbf{b}) = \infty$ if this sub-problem is unsolvable.

2. Sub-problem $\mathbf{B}_2(p_D, c, \mathbf{b})$(to be solved for a fixed upper bound p_D of the supplied power, type c of charging stations and battery options $\mathbf{b}$):

$$\min \Phi_2(K) = \frac{K(cost^{cap}(c) + cost^{ope}(c))}{365} + F_3(p_D, c, K, \mathbf{b}),$$

subject to (37) under fixed $p_D, c, \mathbf{b}$. Objective function of the sub-problem is a sum of an increasing linear function and a decreasing function $F_3(p_D, c, K, \mathbf{b})$ of the parameter K on the given segment. The function $\Phi_2(K)$ may have several minima in general. Therefore, in order to solve the sub-problem one can use directed search of local search methods on a discrete segment. Denote solution of the sub-problem as $K(p_D, c, \mathbf{b})$ and its value as $F_2(p_D, c, \mathbf{b})$. Set $F_2(p_D, c, \mathbf{b}) = \infty$ if the sub-problem has no solution.

3. Sub-problem $\mathbf{B}_1(p_D)$(to be solved for a fixed upper bound p_D of the supplied power):

$$\min \Phi_1(c, \mathbf{b}) = \mathrm{F_2(p_D, c}, \mathbf{b}),$$

subject to (36) and (38). The sub-problem is to determine the type $c \in C$ of the charging stations and battery options $\mathbf{b}$ =$(b_j | j \in J)$, $b_j \in B_{cj}$, $j \in J$. The sub-problem $\mathbf{B}_1(p_D)$ is a discrete programming problem. This sub-problem can be solved by directed search among feasible pairs (c,$\mathbf{b}$) and solving a sub-problem $\mathbf{B}_2(p_D, c, \mathbf{b})$ for each such pair. For deleting infeasible pairs in the directed search, the following necessary conditions are used.

$$\varphi_{jp}(\overline{s}_j) \geq \underline{s}_j, p = 1, \ldots, n_j, j \in J, \tag{49}$$

Denote an exact solution of sub-problem $\mathbf{B}_1(p_D)$ as $(c(p_D), \mathbf{b}\ (p_D))$ and its value as $F_1(p_D)$. Set $F_1(p_D) = \infty$ if the sub-problem is unsolvable.

4. Sub-problem $\mathbf{B}_0$:

$$\min \Phi = \frac{cost(p_D)}{365} + F_1(p_D),$$

subject to (35). Denote an exact solution of this sub-problem as p_D^*.

Properties of sub-problem $\mathbf{B}_0$

The function Φ is a sum of two functions of the variable p_D defined on a given discrete interval $[p_D^1, p_D^{\bar{k}}]$. It is natural to assume that $cost(p_D)$ increases (or does not decrease) with growth of p_D. The function $F_1(p_D)$ decreases (does not increase) with the growth of p_D. This follows from the fact that the solution $(c(p_D), K(p_D), \mathbf{b}\ (p_D))$ of the sub-problem $\mathbf{B}_1(p_D)$ remains feasible for any $p'_D \geq p_D$.

The function Φ may have multiple extrema in general. Therefore, the sub-problem $\mathbf{B}_0$ can be solved by a directed search or local search methods on the discrete segment $[p_D^1, p_D^{\bar{k}}]$.

It is easy to verify that if all sub-problems of the decomposition scheme are solved exactly then p_D^*, $c(p_D^*), \mathbf{b}(p_D^*)$, $K(c(p_D^*), \mathbf{b}(p_D^*))$, $\mathbf{u}^c(c(p_D^*), K(c(p_D^*), \mathbf{b}(p_D^*)), \mathbf{b}(p_D^*))$, $\mathbf{s}^a(c(p_D^*), K(c(p_D^*), \mathbf{b}(p_D^*)), \mathbf{b}(p_D^*))$ constitute an exact solution $p_D^*, c^*, K^*, \mathbf{b}^*, \mathbf{u}^{c*}, \mathbf{s}^{a*}$ of the initial problem DepOpt.

3.3 Efficient structure of input data for DepOpt

The following classes are created in C++: *TLinearPieceWiseFunction*, *TELem*, *TStation*, *TBattery*, *TBus*, *TTrip*, *TEBus*, and *TProblem*.

Class *TLinearPieceWiseFunction* provides representation, input and calculation of piecewise linear functions. It consists of the following items:

- Number of pieces (segments) n.
- Array a of left endpoints of the segments.
- Array b of right endpoints of the segments.
- Arrays of coefficients $k1$ and $k2$ of the linear functions for segments.

Class *TELem* is the base class for the classes *TStation*, *TBattery*, *TBus*, *TTrip*, and *TEBus*. Class *TELem* includes the following items: full name, short name, and element index. Full name used for generating output information of optimization. Short name is used as the reference to the object of classes *TStation*, *TBattery*, *TBus*, *TTrip*, and *TEBus*. Element index is used for indexing of the objects of these classes.

Additional items of the class *TStation* for representing charging station $c \in C$ are:

- Nominal power P_c.
- Capital cost $cost^{cap}(c)$.
- Operating and depreciation cost $cost^{ope}(c)$.

Additional items of the class *TBattery* for representing battery $b \in B$ are:

- Capital cost $cost_b$.

- Minimal SOC level $\underline{s}_b$.
- Maximal SOC level $\bar{s}_b$.
- Array of appropriate charging station indices C_b.
- Dimension of C_b.
- Array of pointers to the objects of the class *TLinearPieceWiseFunction* for $f_{bc}(\tau)$.
- Pointer to the object of the class *TLinearPieceWiseFunction* for $N_j(s_j^{low})$.

Additional items of the class *TBus* for representing elements of set EB are:

- Array of eligible battery indices B^e.
- Dimension of B^e.

Additional items of the class *TTrip* for representing elements of set TR are:

- Departure time $\underline{t}$.
- Arrival time $\bar{t}$.

Additional items of the class *TEBus* for representing $j \in J$ are:

- Pointer to the object of class *TBus*.
- Array of eligible battery indices B_j.
- Dimension of B_j.
- Array of eligible trip indices TR_j.
- Dimension of TR_j.
- Array of pointers to the objects of the class *TLinearPieceWiseFunction* for functions $\varphi_{jbp}(s)$ for each pair (battery, trip), $b \in B_j$, $p \in TR_j$.
- Annual number $\overline{N}_j$ of charge/discharge cycles.

Items of the class *TProblem* are:

- Array of pointers to the objects of the class *TStation* for representing the set C.
- Dimension of C.
- Array of pointers to the objects of the class *TBattery* for representing set B.
- Dimension of B.
- Array of pointers to the objects of the class *TBus* for representing the set EB.
- Dimension of EB.
- Array of pointers to the objects of the class *TTrip* for representing the set TR.
- Dimension of TR.
- Array of pointers to the objects of the class *TEBus* for representing the set J.
- Dimension of J.
- Arrays of values for representing the function $P(t)$ (ends of periods and shares of power at the ends of the periods).
- Arrays of values for representing the function $c_e(t)$ (ends of the periods and tariffs at the ends of the periods).
- Arrays of values for representing the function $cost(p_D)$ (powers supplied and costs of the power).

It should be noted that all the dimensions are defined dynamically at the stage of the data input. Elements of the sets (arrays) are indexed starting with 0 in the order of their input. An access to the element of a set is performed by using its index.

Two formats of the input files are implemented. One of them is the JSON format and another is the simple text.

3.4 Format JSON of input files for DepOpt

If the JSON format is used for the input data, then the following files must be prepared: *stations.json, batteries.json, fbatteries.json, fcbatteries.json, buses.json, routes.json, trips.json, fleet.json, febuses.json, fpower.json.*

File *stations.json* describes the set C and defines values of the following parameters: *fn_c* (full name of the charging station), *sn_c* (short name of the charging station), *po_c* (nominal power po_c),*cc_cap_c* (capital cost cc_c^{cap}) and *cc_ope_c* (operating and depreciation cost cc_c^{ope}).

```
{ "C": [{
"fn_c": "Charging station",
"sn_c": "CS",
"po_c": 200,
"cc_cap_c": 8590,
"cc_ope_c": 5000
}
]}
```

File *batteries.json* describes the set B and defines values of the following parameters: *fn_bat* (full name of the battery), *sn_bat* (short name of battery), *smin_bat* (minimal SOC level),*smax_bat* (maximal SOC level), *costb_bat* (cost), *C_bat* (short names of the eligible charging stations), and *chr_bat* (charging rates of the eligible charging stations). For example:

```
{ "B": [{
"fn_bat": "Battery 1",
"sn_bat": "B1",
"smin_bat": 94.05,
"smax_bat": 470.25,
"costb_bat": 202206,
"C_bat": ["CS"],
"chr_bat": [188.1]
}
]}
```

The following parameters are defined in the file *fbatteries.json*: *sn_bat* (short name of the battery b) and functions $f_{bc}(\tau)$ for each eligible charging station c from C_b. Functions are defined by *a_1, b_1, k1_1, k2_1, . . . , a_n, b_n,k1_n,k2_n* where n is the number of eligible charging stations ($|C_b|$). Here *a_i* is the array of left endpoints of the segments of the function for ith charging station from C_b, *b_i* is the array of the right endpoints of the segments of the function for ith charging station from C_b, *k1_i* is the array of the coefficients $k1$ of the function for ith charging station from C_b, and *k2_i* is the array of the coefficients $k2$ of the function for ith charging station from C_b. For example:

{ "BF": [{

"sn_bat": "B1",

"a_1": [0],

"b_1": [2],

"k1_1": [188.1],

"k2_1": [94.05]

}

]}

The following parameters are defined in the file *fcatteries.json*: *sn_bat* (short name of the battery) and function $N_b(b_b^{low})$. The function is defined by *soc_bat* (discharge levels) and *ncycl_bat* (maximal numbers of charge/discharge cycles). For example:

{ "CBF": [{

"sn_bat": "B1",

"soc_bat": [47.03,94.1,141.1,188.1,235.1,282.2,329.2,376.2,423.2,470.25],

"ncycl_bat": [450000,150000,50000,24000,14000,7000,4400,3000,2300,1900]

}

]}

File *buses.json* describes the set EB and defines: *fn_b* (full name of the e-bus type), *sn_c* (short name of the e-bus type), and *B_b* (short names of the eligible batteries). For example:

{ "EB": [{

"fn_b": "Vitovt Max Electro E433",

"sn_b": "E433",

```
”B_b”: [”B1”]
}
]}
```

File *routes.json* describes the set R and defines values of the parameters *fn_r* (full name of the route), *sn_r* (short name of the route), and *l_r* (the total length of the route). For example:

```
{ ”R”: [{
”fn_r”: ”Railway Station - DS Viasnjanka”,
”sn_r”: ”A1”,
”l_r”: 9
}
]}
```

File *trips.json* describes the set TR and defines values of the parameters *fn_t* (full name of the trip), *sn_t* (short name of the trip), *dep_t* (departure time), *arv_t* (arrival time) and *R_t* (short names of the routes). For example:

```
{ ”T”: [{
”fn_t”: ”Trip 1”,
”sn_t”: ”T1”,
”dep_t”: 5.13,
”arv_t”: 22.44,
”R_t”: [”T43”]
}
]}
```

File *fleet.json* describes the set J and defines values of the parameters *fn_j* (full name of the e-bus), *sn_j* (short name of the e-bus), *sn_b_j* (short name of the e-bus type), *Nc_j* (annual number of charge/discharge cycles), *B_j* (short names of the eligible batteries) and *TR_j* (short names of the trips). For example:

```
{ ”J”: [{
”fn_j”: ”2816”,
”sn_j”: ”2816”,
”sn_b_j”: ”E433”,
```

"Nc_j": 350,

"B_j": ["B1"],

"TR_j": ["T1"]

}

]}

The following parameters are defined in the file *febuses.json*: *sn_j* (short name of the e-bus) and functions $\varphi_{\mu bp}(s)$ for each pair (battery $b \in B_j$, trip $p \in TR_j$). The functions are defined by *a_1_1, b_1_1, k1_1_1, k2_1_1, . . . , a_1_n2, b_1_n2, k1_1_n2, k2_1_n2,*a_2_1, b_2_1, k1_2_1, k2_2_1, . . . , a_2_n2, b_2_n2, k1_2_n2, k2_2_n2,. . . ,a_n1_n2, b_n1_n2, k1_n1_n2, k2_n1_n2 where *n1* is the number of eligible batteries in B_j and *n1* is the number of served trips in TR_j. Here *a_i_l* is the array of the left endpoints of the segments of the function for ith battery from B_j and lth trip from TR_j, *b_i_l* is the array of the right endpoints of the segments of the function for ith battery from B_j and lth trip from TR_j, *k1_i_l* is the array of the coefficients $k1$ of the function for ith battery from B_j and lth trip from TR_j, and *k2_i_l* is the array of the coefficients $k2$ oof the function for ith battery from B_j and lth trip from TR_j. For example:

{ *"FJ": [{*

"sn_eb": "2816",

"a_1_1": [94.05],

"b_1_1": [470.25],

"k1_1_1": [1],

"k2_1_1": [-376.2]

}

]}

File *fpower.json* includes definitions of functions $P(t)$, $c_e(t)$ and $cost(p_D)$: *T_p* (endpoints of the periods for $P(t)$), *F_p* (shares of the power at the endpoints of the periods), *T_c* (endpoints of the periods for $c_e(t)$), *F_c* (tariffs at the endpoints of the periods), *Pd* (powers supplied), *Cpd* (costs of the power). For example:

{

"T_p": [5,12,20,24],

"F_p": [1,1,1,1],

"T_c": [5,12,20,24],

"F_c": [0.065,0.065,0.065,0.065],

"Pd": [200,400,600,800],

"Cpd": [2000,4000,6000,8000]

}

3.5 Text format of input files for DepOpt

If the input data are prepared in the text format, then the following files must be created: *stations.txt, batteries.txt, fbatteries.txt, fcbatteries.txt, buses.txt, trips.txt, fleet.txt, febuses.txt, fpower.txt*. Each file can include comments. They must start with the symbols // and be placed at the top of file. The main body of the file starts with a new line immediately after the comments. Values in the rows are separated by commas.

File *stations.txt* consists of one row for each element of the set C. Each row contains: full name of the charging station, short name of the charging station, nominal power, capital cost, operating and depreciation cost. For example: *Charging station 1, CS1, 200, 250000, 5000.*

File *batteries.txt* consists of three rows for each element of the set B. The first row contains: full name of the battery, short name of the battery, minimal SOC level, maximal SOC level, cost of the battery. The second and third rows contain short names and charging rates of the eligible charging stations for the battery, respectively. For example, row 1: *Battery 1, B1, 94.05, 470.25, 202206*, row 2: *CS1*, row 3: *188.1.*

File *fbatteries.txt* consists of the data for each element of the set B. The first row contains short name of the battery. The next $4 \cdot |C_b|$ rows consist of the data for the functions $f_{bc}(\tau)$ for each eligible charging station c (first two rows specify left a and right b endpoints respectively of charging time segments, the last two rows define coefficients $k1$ and $k2$ of the linear functions for each of these segments). For example, row 1: *B1*, row 2: *0*, row 3: *2*, row 4: *188.1*, row 5: *94.05.*

File *fcbatteries.txt* consists of three rows for each element of the set B. The first row contains short name of the battery. The next two rows consist of the data for the function $N_b(b_b^{low})$ (first row specifies the discharge levels, second row defines maximal number of charge/discharge cycles). For example, for *Battery 1*, row 1: *B1*, row 2: *47.03, 94.1, 141.1, 188.1, 235.1, 282.2,* *329.2, 376.2, 423.2, 470.25*, row 3: *450000, 150000, 50000, 24000, 14000, 7000, 4400, 3000, 2300, 1900*.

File *buses.txt* consists of two rows for each element of the set EB. The first row consists of full and short names of the e-bus type. The second row contains short names of the eligible batteries. For example, for e-bus type *Vitovt Max Electro E433*, row 1: *Vitovt Max Electro E433, E433*, row 2: *B1*.

File *routes.txt* consists of one row for each element of the set R in the format *full name of the route, short name of the route, total length of the route*. For example, for the route *DS Drugnaja - DS Siarova*: *DS Drugnaja - DS Siarova, T43, 7*.

File *trips.txt* consists of two rows for each element of the set TR. The first row contains full name of the trip, short name of the trip, departure time and arrival time. The second row consists of the short names of the served routes. For example, row 1: *Trip 1, T1, 5.13, 22.44*, row 2: *T43*.

File *fleet.txt* consists of three rows for each element of the set J. The first row contains full name of the e-bus, short name of the e-bus, short name of the e-bus type, annual number of its battery charge/discharge cycles. The second row includes short names of the eligible batteries. The third row contains short names of the eligible trips. For example, row 1: *2816, 2816, E433, 350*, row 2: *B1*, row 3: *T1*.

File *febuses.txt* consists of the data for each element of the set J. The first row contains short name of the e-bus. The next rows include data for the functions $\varphi_{jbp}(s)$ (4 rows) for each pair (battery $b \in B_j$, trip $p \in TR_j$). It is assumed that trip is changed first. First two rows specify arrays a and b of left and right endpoints of its battery SOC level segments. The next two rows define arrays $k1$ and $k2$ of function coefficients). For example, row 1: *2816*, row 2: *94.05*, row 3: *470.25*, row 4: *1*, row 5: *-376.2*.

File *fpower.txt* consists of 6 rows and defines functions $P(t)$, $c_e(t)$ and $cost(p_D)$. The first two rows contain endpoints of the periods and shares of supplied power at the endpoints of these periods. The next two rows include endpoints of the periods and fixed power rates in these periods. The last two rows include powers supplied and costs of the power. For example, row 1: *5, 12, 20, 24*, row 2: *1, 0.85, 0.85, 1*, row 3: *5, 12, 20, 24*, row 4: *0.065, 0.065, 0.065, 0.065*, row 5: *200, 400, 600, 800*, row 6: *2000, 4000, 6000, 8000*.

3.6 Efficient structure of output data for DepOpt

The following items of the class *TProblem* are used for representing the results of the optimization.

- Maximal power supplied to the depot p_D^*.
- Type of charging station c^*.
- Number of charging stations K^*.
- Array of types of battery for each e-bus b_j^*, $j \in J$.
- Array of variables x obtained from the Open Source MIP solver LpSolve.

The following data is derived from this output.

- The optimal cost Φ^*.
- The total unused daily charging time resource in the depot $\bar{t}^{ch}$.
- The share of the used daily resource of charging time in the depot Ψ^{ch}.
- Arrays of SOC levels s_j^{dp} for each e-bus.
- Arrays of SOC levels s_j^{ap} for each e-bus.
- Arrays of times u_j^{ap} for each e-bus.
- Arrays of charging time u_j^{cp} for each e-bus.
- Arrays of vectors of charging times u_j^c for each e-bus.
- Arrays of charging times δ_j^i for each e-bus.
- Arrays of unused charging time resources t_i^{ch} for each time interval $[t_i, t_{i+1}]$.

3.7 Minsk case of DepOpt

The experimental software implementing decomposition algorithms was used to solve instances of the problem DepOpt for a set of public transport routes in the city of Minsk. In these instances, slow-charging infrastructure of a single depot was considered. One of these instances is described below.

The name of the depot is $Vaneeva$ (V). The power p_D supplied to the depot V can take the following values: 200, 400, 600, 800 kW. The annual cost $cost(p_D)$ of the electric power supplied to the depot V is 2000, 4000, 6000, and 8000 €, respectively.

The remaining input data on e-buses and served routes are given in Tables 7 and 8. Passenger capacity, range of the single-charge drive, cost and time are measured in persons, kilometers, euros (€) and hours, respectively.

Table 7: E-bus types. Input data.

No	Name	Battery cap.(kWh)	Max. char. time τ_j^{max}(h)	Capacity (pass.)	Route	Single-charge range (km)
1	Vitovt Max Electro E433	470.25	2	123	1-3	220

Table 8: Routes.

ID (depot)	Name	Return route	Total length	Duration	Number of stops
$3(V)$	$T43$	$(Druz, Siar)$	14	50	26

Depot V is equipped with the slow-charging stations of the same type c with output power P_c = 200 kW. Thus, C=$\{c\}$. Capital cost of one charging station c is 171806 €. Given the estimated period of operation of the station as 20 years, its annual capital cost is $cost^{cap}(c) =$ 8590 €. Annual operating and depreciation cost of one charging station c is $cost^{ope}(c) = 5000$ €. The set of routes is $R = \{3\}$.

The set of e-buses serving the three routes is $J = \{1, 2, 3, 4\}$. All the e-buses are of the same type $E433$, so $EB = \{E433\}$. Maximal range of any e-bus is 220 km.

All the sets $B_j = \{b\}$, $j = 1, 2, 3, 4$, where b is the only battery option with 188 LFP cells and total capacity of $470.25kWh$. Cost $cost_b$ of the battery option b used in all four e-buses is 202206 €. Minimal and maximal SOC levels of the battery b for e-bus j are the same: $\underline{s}_j = 94.05$ and $\bar{s}_j = 470.25$, respectively, $j = 1, 2, 3, 4$.

The time moments of departure from/ arrival to the depot for the e-buses are:

1: $[5.217, 22.73]$; 2: $[5.45, 12.8]$, $[15.93, 22.27]$; 3: $[5.75, 23.13]$; 4: .$[6.00, 22.45]$. Here j : specifies e-bus j, $j = 1, 2, 3, 4$.

Functions of charge loss of the e-buses on the routes are as follows:

$\varphi_{11}(s) = s - 376.2$; $\varphi_{21}(s) = s - 160.74$; $\varphi_{22}(s) = s - 136.8$; $\varphi_{31}(s) = s - 376.2$; $\varphi_{41}(s) = s - 352.26$.

The battery charge recovery functions are:

$f_{jc}(\tau) = 94.05 + 188.1\tau$ for $\tau \in [0, 2]$, $j = 1, 2, 3, 4$. $f'_{jc} = 188.1$.

The number of charge/discharge cycles of the battery b over its entire life, depending of its depth of discharge, is presented in Table 9.

Table 9: Number of battery charge/discharge cycles as a function of depth of its discharge

No	Depth of discharge $\%/s_b^{low}$	Number of cycles	Reverse value
1	10/423.2	450000	2.222E-6
2	15/400	280000	3.57E-6
3	20/376.2	150000	6.6666E-6
4	25/352.7	80000	1.255E-5
5	30/329.2	50000	2E-5
6	35/305.7	35000	2.857E-5
7	40/282.2	24000	4.16666E-5
8	45/258.6	18000	5.55555E-5
9	50/235.1	14000	7.142857E-5
10	55/211.6	9200	0.0001087
11	60/188.1	7000	0.0001428
12	65/164.6	5600	0.00017857
13	70/141.1	4400	0.00022727
14	75/117.6	3500	0.00028571
15	80/94.1	3000	0.00033333
16	85/70.5	2600	0.0003846
17	90/47.03	2300	0.00043478
18	95/23.51	2100	0.00047619
19	100/0	1900	0.0005263

The number of annual charge/discharge cycles of the e-buses are: $\overline{N}_j = 350, j = 1, 3, 4,$

$\overline{N}_2 = 700$.

The electric power rate is 6.5 € per 100 kWh.

Function $P(t, p_D) = p_D$ for $t \in [0, 24]$.

Solution.

One solution is found. It is:

$p_D^* = 200$; $c^* = c$; $K^* = 1$; $b_j^* = b$, $j = 1, 2, 3, 4$; $s_1^{a1*} = 94.05$; $s_2^{a1*} = 309.51$; $s_2^{a2*} = 333.45$; $s_3^{a*} = 94.05$; $s_4^{a*} = 117.99$; $u_1^{c1*} = 2.0$; $u_2^{c1*} = 0.85454$; $u_2^{c2*} = 0.72727$; $u_3^{c1*} = 2.0$; $u_4^{c1*} = 1.87273$.

The total charging time of all e-buses of the route T43 is 7.45454 hours.

The optimal daily cost $\Phi^* = 329.08$ €.

The share of total unused daily charging time resource in the depot $\bar{t}^{ch} = 0.689$.

The share of the used daily resource of charging time in the depot $\Psi^{ch} = 0.311$.

4 Efficient algorithm and data structure for OptSched

Recall that the problem OptSched is to distribute departures of buses of the same type assigned to the same route as uniform as possible over the departures of all buses serving this route in the decisive time period. The timetables which address this objective are called balanced. It is assumed that buses of different types have different passenger capacities. Therefore, a balanced timetable ensures a uniform allocation of bus capacities over time in the decisive time period of the same route.

Denote $V = \sum_{b=1}^{n} v_b$ and $s_b = v_b/V$, $b = 1, \ldots, n$. The value of V is equal to the number of buses operating on the same route in the decisive time period. According to the balanced timetable objective, the number of buses of type b departed in the first k traffic intervals must be kept as close to $s_b k$ as possible for $b = 1, \ldots, n$. Introduce non-negative integer variables x_{bk} representing the number of buses of type b departed in the first k traffic intervals, $b = 1, \ldots, n$, $k = 1, \ldots, V$. Define $x_{b0} = 0$, $b = 1, \ldots, n$. Denote by x matrix with entries x_{bk}. In the previous Deliverable, we have selected the following mathematical model for the problem OptSched.

$$\textbf{Problem } \textsc{OptSched-Max}: \quad \min_{x} \max_{1 \le k \le V, 1 \le b \le n} |x_{bk} - s_b k|, \text{ subject to}$$

$$\sum_{b=1}^{n} x_{bk} = k,\ \ k = 1, \ldots, V, \tag{50}$$

$$0 \le x_{bk} - x_{b(k-1)},\ \ b = 1, \ldots, n,\ \ k = 1, \ldots, V, \tag{51}$$

$$x_{b0} = 0,\ \ b = 1, \ldots, n, \tag{52}$$

$$x_{bV} = v_b,\ \ b = 1, \ldots, n, \tag{53}$$

$$x_{bk} \in Z_0,\ \ b = 1, \ldots, n,\ \ k = 1, \ldots, V. \tag{54}$$

4.1 Algorithm for OptSched

We employ optimal $O(V \log V)$ time algorithm of Steiner and Yeomans [7], denoted as SY, to solve the problem OptSched. This algorithm can be described as follows. Consider the following auxiliary decision problem.

Problem OptSched-Max(M) : $\max_{1 \le k \le V, 1 \le b \le n} |x_{bk} - s_b k| \le M,$ subject to (50)-(54),

where M satisfies $\frac{V - v_{\max}}{V} \le M \le \frac{V-1}{V}$, $v_{\max} = \max_b\{v_b\}$. This problem is equivalent to the scheduling problem $1|r_j, \bar{d}_j, p_j = 1|\cdot$, in which there is a single machine (route), unit-time ($p_j = 1$) jobs (buses with unit-time traffic interval) of a set N, each job j has a release date r_j and a deadline $\bar{d}_j$, no two jobs can be processed concurrently, and each job j has to be processed between r_j and $\bar{d}_j$. For any instance of OptSched-Max(M), an equivalent instance of $1|r_j, \bar{d}_j, p_j = 1|\cdot$ is constructed by setting $N = \{(b, h) \mid b = 1, \ldots, n,\ h = 1, \ldots, v_b\}$, $r_{(b,h)} = \lceil \frac{h-M}{s_b} \rceil$ and $\bar{d}_{(b,h)} = \lfloor \frac{h-1+M}{s_b} + 1 \rfloor$, $(b, h) \in N$, where (b, h) is a bus of type b numbered h. The problem $1|r_j, \bar{d}_j, p_j = 1|\cdot$ is solved by applying the following rule formulated by Horn [3]: In any time interval k, $k = 1, \ldots, V$, schedule an available unassigned job j ($r_j \le k \le \bar{d}_j$) with the smallest deadline. If no job is available for $k \in \{1, \ldots, V\}$, then $1|r_j, \bar{d}_j, p_j = 1|\cdot$, and hence, corresponding OptSched-Max(M), has no solution.

Algorithm SY**.**

Step 1. (Initialization) Set $L = \frac{V - v_{\max}}{V}$ and $U = \frac{V-1}{V}$. We have $L \le M^* \le U$. Solve the problem OptSched-Max(L). If a feasible solution of this problem is found, then it is optimal for OptSched-Max, stop. Assume that it is not found. Solve the problem OptSched2-Max(U). Observe that $|M_1 - M_2| \ge \frac{1}{V^2}$ for any two distinct objective functions values M_1 and M_2 of OptSched-Max.

Step 2. (Bisection search) If $U - L < \frac{1}{V^2}$, then stop: a feasible solution for the problem OPTSCHED-MAX(U) is optimal for OPTSCHED-MAX. Else, solve the problem OPTSCHED-MAX($(L+U)/2$). If a feasible solution of this problem is found, then re-set $U := (L+U)/2$ and repeat Step 2. Else, re-set $L := (L+U)/2$ and repeat Step 2. ■

The number of iterations of the bisection search in Step 2 is $O(\log V)$.

4.2 Computer implementation, efficient structure of input and output data and Minsk case of OPTSCHED

Algorithm SY is implemented in C++ in two ways: as function *Schedule* of class *TRoute* and as a standalone program. In the first case, the input and output data of the problem OPTSCHED are part of the input and output data of the problem OPT, which includes OPTSCHED as a subproblem.

In the second case the input consists of the short names of vehicle types (e-bus and conventional vehicle types) and the numbers v_b of vehicles of each type b, $b = 1, \ldots, n$, serving the considered route. Recall that $V = \sum_{b=1}^{n} v_b$.

Two formats of the input data are implemented: JSON format and simple text. Type of the format is specified in the configuration file *sched.ini* by the parameter *json*, where $json = 1$ if input data are in the JSON format and $json = 0$ if they are in the text format. In the first case, the file *schedin.json* should be prepared in which values for names *sn_b* (short names of the e-bus and conventional vehicle types) and *nv_b* (number of vehicles of type b) are to be defined. For example:

```
{
sn_b: [MAZ103, E433],
nv_b: [3, 8]
}
```

In the second case, the file *schedin.txt* should be prepared which consists of two rows. The first row includes short names of the e-bus and conventional vehicle types. The second row contains number of vehicles of each type. For example:

```
MAZ103, E433
3, 8
```

The output data is a sequence $(t_1, \ldots, t_V)$, where t_i is the type of the vehicle departed i-th among all V vehicles in the decisive time period of the same route. Two formats of the output data are implemented: JSON format and simple text.

Object t is defined in the file *sched.json*. It consists of the short names of e-bus and conventional vehicle types. For example:

{*t:* [

E433, MAZ103, E433, E433, E433, MAZ103, E433, E433, E433, MAZ103, E433]

}

The obtained solution, which is the sequence of vehicle types in the departure order, is placed into the file *sched.out*. It consists of the sequence of short names of e-bus and conventional vehicle types. For example:

E433 MAZ103 E433 E433 E433 MAZ103 E433 E433 E433 MAZ103 E433

The experimental software implementing algorithm SY was used to solve a real-life instance of the city of Minsk described in Section 2.6. The solution obtained is given in Table 10.

Table 10: Minsk case. Sequences of vehicle types

Routes	Sequence of bus types
A1	(E433,E433,E433,E433,E433,E433,E433,E433)
T59	(E433,E433,E433,E433,T420,E433,E433,E433,E433)
T43	(E433,E433,E420,E433,E433,E433)
T20	(E433,E433,T420,E433,E433,E420,E433,E433)
T40	(E433,T333,E433,T420,E433,E433,T333,E433)

5 Conclusion

In this report, efficient algorithms and data structures are described for the problems Opt, DepOpt and OptSched, which were analyzed and formulated in the previous Deliverable. Computer experiments with the algorithms demonstrated that they are able to solve real-sized instances of these problems in few minutes on a standard PC. The proposed methods can be used as decision support tools for planning process of conversion of the conventional bus fleet to a fully electric bus fleet.

6 Total Cost of Ownership model

Total Cost of Ownership (TCO) analysis helps to investigate all costs of the owner-ship of a product during its useful life [9] (Bickert and Kuckshinrichs, 2011). TCO is considered also as a tool and philosophy, which is aimed at the identification of the true costs of buying goods or services [14] (Hagman et al., 2016). Two ways of TCO analysis can be identified: a consumer-oriented approach and society-oriented approach. Consumer-oriented TCO analysis is focused on the consumers point of view, so include only such costs which are perceived and borne by consumers [10], [11] (Ellram, 1999, 1995; [15] Lebeau et al., 2014; [16] Letmathe and Suares, 2017). Society-oriented TCO analysis has much broader scope and considers external costs [15] (Lebeau et al., 2014). The developed TCO model is an socially-oriented, dynamic model of TCO. By design, this model is to serve an ex-ante assessment of the costs of the planned investment. The developed dynamic TCO model provides different ways of financing of the investment, as well as its implementation in parts over various periods of time.

6.1 Static TCO model

In the static TCO model (see 2) the following assumptions were made regarding the calculation and analysis of the bus fleet conversion process: a) One-time purchase of the necessary number of buses immediately and construc-tion of the infrastructure in an amount ensuring full service of the bus battery charging needs, b) Different financing models for the purchase of buses and battery charging infra-structure: i) Self-financing, ii) Self-financing and subsidy, iii) Loan for own funds, c) fixed values of all characteristics/variables in subsequent years of the analysis period, including annual bus operation, prices of electricity and its supply, bus prices and infrastructure construction costs, d) Cost allocation for only one stakeholder/beneficiary of the investment.

6.2 Socially-oriented dynamic TCO model

The presented TCO model is a dynamic model and the development of the static economic model. The dynamics of the TCO model is primarily to take into account the following phenomena overtime associated with the period of calculations and analyzes of bus fleet conversion

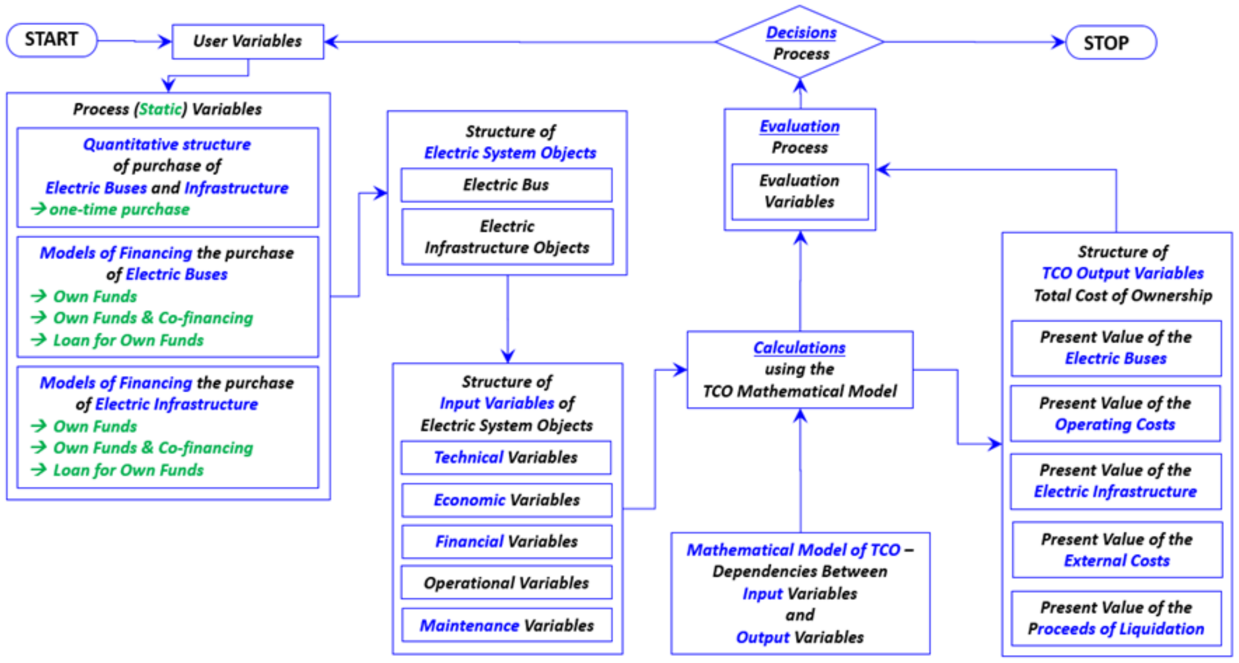
START
User Variables
Decisions Process
STOP
Process (Static) Variables
Quantitative structure of purchase of Electric Buses and Infrastructure
→ one-time purchase
Models of Financing the purchase of Electric Buses
→ Own Funds
→ Own Funds & Co-financing
→ Loan for Own Funds
Models of Financing the purchase of Electric Infrastructure
→ Own Funds
→ Own Funds & Co-financing
→ Loan for Own Funds
Structure of Electric System Objects
Electric Bus
Electric Infrastructure Objects
Structure of Input Variables of Electric System Objects
Technical Variables
Economic Variables
Financial Variables
Operational Variables
Maintenance Variables
Evaluation Process
Evaluation Variables
Calculations using the TCO Mathematical Model
Mathematical Model of TCO – Dependencies Between Input Variables and Output Variables
Structure of TCO Output Variables Total Cost of Ownership
Present Value of the Electric Buses
Present Value of the Operating Costs
Present Value of the Electric Infrastructure
Present Value of the External Costs
Present Value of the Proceeds of Liquidation

Figure 2: General algorithm of socially-oriented model of TCO (static approach).

processes: a) Purchase of the necessary number of buses immediately, b) Purchase of buses in tranches from time to time, c) Construction of infrastructure in an amount to provide full service to the needs of bus battery charging, d) Expansion of infrastructure as the growing needs arising from the purchase of buses in tranches from time to time, e) Different financing models for the purchase of buses for each of options a) and b): i) Self-financing, ii) Self-financing and subsidy, iii) Loan for own funds, iv) Leasing, f) Different financing models for bus battery charging infrastructure for each of op-tions c) and d): i) Self-financing, ii) Self-financing and subsidy, iii) Loan for own funds, iv) Leasing, g) Change in the values of the following characteristics/variables in subsequent years of the analysis period: i) Annual bus operation, ii) The price of electricity and the price of the supply of this energy, iii) Bus price in subsequent tranches, iv) Price of infrastructure in subsequent tranches, h) Different cost-sharing structure for individual stakeholders/investment beneficiar-ies: i) Operator, ii) Transport organizer, iii) Public transport authority / local government.

The general algorithm of socially-oriented dynamic model of TCO shows 3.

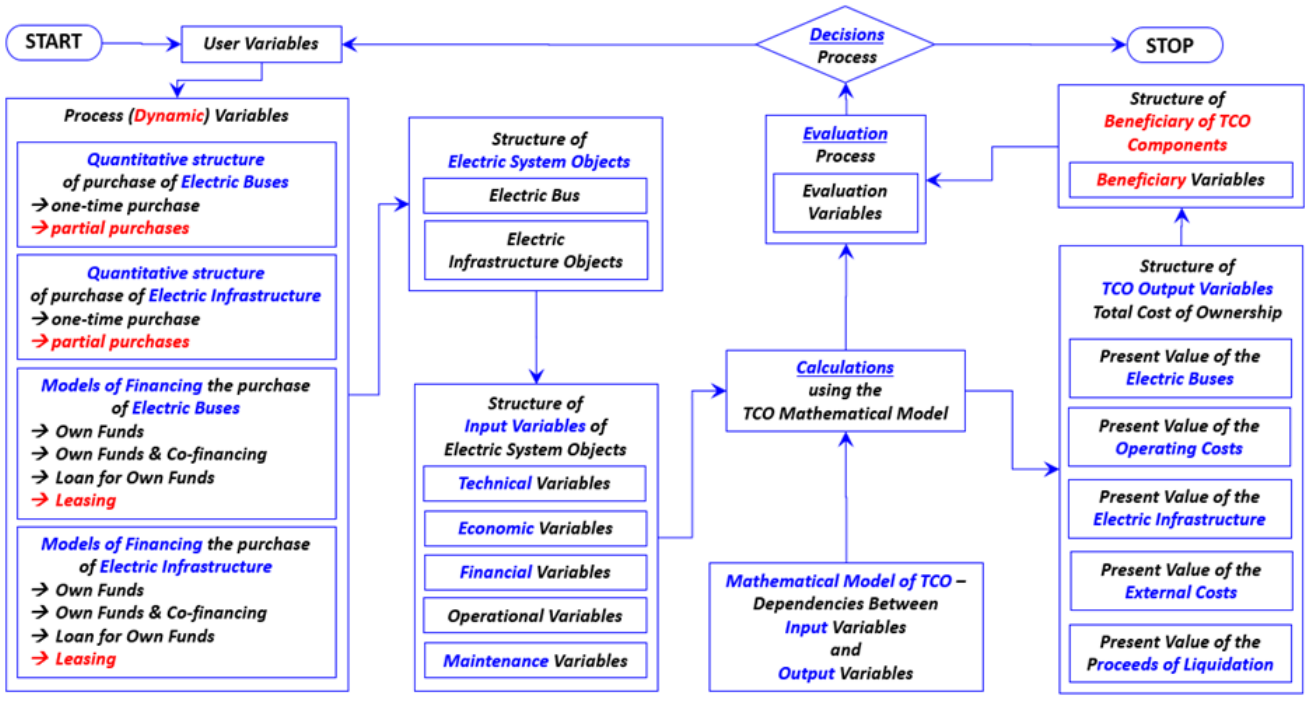
START
User Variables
Decisions Process
STOP
Process (Dynamic) Variables
Quantitative structure of purchase of Electric Buses
→ one-time purchase
→ partial purchases
Quantitative structure of purchase of Electric Infrastructure
→ one-time purchase
→ partial purchases
Models of Financing the purchase of Electric Buses
→ Own Funds
→ Own Funds & Co-financing
→ Loan for Own Funds
→ Leasing
Models of Financing the purchase of Electric Infrastructure
→ Own Funds
→ Own Funds & Co-financing
→ Loan for Own Funds
→ Leasing
Structure of Electric System Objects
Electric Bus
Electric Infrastructure Objects
Structure of Input Variables of Electric System Objects
Technical Variables
Economic Variables
Financial Variables
Operational Variables
Maintenance Variables
Evaluation Process
Evaluation Variables
Calculations using the TCO Mathematical Model
Mathematical Model of TCO – Dependencies Between Input Variables and Output Variables
Structure of Beneficiary of TCO Components
Beneficiary Variables
Structure of TCO Output Variables Total Cost of Ownership
Present Value of the Electric Buses
Present Value of the Operating Costs
Present Value of the Electric Infrastructure
Present Value of the External Costs
Present Value of the Proceeds of Liquidation

Figure 3: General algorithm of socially-oriented model of TCO (dynamic approach).

6.3 Dynamic vs static model

The differences between the current dynamic TCO model and its static version are primarily that: I) The purchase of the required number of buses in the static model was carried out in full, i.e. without tranches. Whereas in the dynamic model, variants a) and b) were additionally included, II) The purchase and construction of the needed bus battery charging infrastructure in the static model were carried out in its entirety, i.e. without tranches. Whereas in the dynamic model, variants c) and d) were additionally included, III) In the dynamic model, combinations of variants for the purchase of buses and variants for the construction of the infrastructure are additionally possible, e.g.: i. a) & c); ii. a) & d), however, there may be limited possibilities of bus coverage due to the phased construction of the infrastructure (in tranches), iii. b) & c), however, it is recommended to adapt the needs of servicing a specific number of buses in subsequent tranches to the resources of the battery charging infrastructure - to optimally use the resources of the battery charging infra-structure to the changing needs of bus service related to their number in subsequent tranches, iv. b) & d), however, it is recommended to adapt the required amount of infrastructure in subsequent tranches to the assumed number of buses in subsequent tranches - to optimally use the resources of the battery charging infrastructure for the needs of bus service, IV) In the dynamic model, combinations of variants for financing bus purchase mod-els are additionally possible e) with models for financing bus battery charging infrastructure f) for specific combinations of variants regarding bus purchase and infrastructure construction - point III. V) In the dynamic model, additional values of specific characteristics/variables can be changed in subsequent years of the analysis period, including: i. The annual work operating the bus, ii. The price of electricity and the price of the supply of this energy, iii. The price of the bus in subsequent tranches, iv. Price of infrastructure in subsequent tranches, VI) In the dynamic model, additional variants of the distribution of TCO costs into various stakeholders/beneficiaries of the investment related to the purchase of buses and the infrastructure of electric battery charging are possible. VII) In the dynamic model, in addition to the static model, all of the above-mentioned data change possibilities are possible at any time during the analysis - the variables are time-dependent.

The developed TCO model covers all major cost categories associated with the acquisition of an electric bus fleet and relevant infrastructure. It also includes the cost of depreciation, which is the cost of market value loss of buses over time as the difference between their bus purchase price and the resell price/liquidation value. Depreciation is one of the largest costs in the structure of the costs of transport. The estimation of present worth of costs that will take place in future years is made with the present value (PV), as it is a method widely used in business and economics to analyze cash flows at different time periods. PV reflects the current value of future cash flows. To receive the present value of money the future value of money needs to be discounted. The present worth is usually less or equal to future worth ([15] Lebeau et al., 2014; [17] Nurhadi et al., 2014).

The Total Cost of Ownership was calculated using the following equation:

$$TCO = (PV_{bus} - \ PVL_{liq\_bus}) + PVOC_{bus} + PV_{infr} + PVE_{exter} \qquad (1)$$

where: TCO – total cost of ownership [PLN, EUR], PV_{bus} – present value of the acquisition costs of electric buses [PLN, EUR], $PVOC_{bus}$ – present value of buses operating costs [PLN, EUR], PV_{infr} – present value of the infrastructure [PLN, EUR], PVE_{exter} – present value of the external costs [PLN, EUR], PVL_{liq_bus} – present value of the proceeds of liquidation [PLN, EUR]

6.4 Electric Buses Acquisition Costs

Present value of the acquisition costs of electric buses is a sum of discounted nominal acquisition costs of each purchased bus:

$$PV_{bus} = \sum_{i=1}^{n} DAC_{bus\_nom_i} \quad (2)$$

where: $DAC_{bus\_nom_i}$ – discounted nominal acquisition costs of i-th bus [EUR], i – number of consecutive bus (i-th bus number), n – number of purchased buses.

Discounted nominal acquisition costs of i-th bus is carried out by the equation:

$$DAC_{bus\_nom_i} = \sum_{m_{ac}}^{M_{ac}} (AC_{bus\_nom_i} \frac{1}{(1+i_{bus})^{m_{ac}}}) \quad (3)$$

where: $AC_{bus\_nom_i}$ – nominal acquisi-tion costs of bus [EUR], i_{bus} – market interest rate [-], M_{ac} – investment term [years], i – number of consecutive bus (i-th bus number) , m_{ac} – time with bus acquisition.

Nominal acquisition costs of bus are calculated as sum of battery nominal costs, bus costs and costs of double-layer capacitors:

$$AC_{bus\_nom_i} = (Bat_{cap} Bat_{unit_{Cost}}) + Bus_{cost} + Capac_{cost} \quad (4)$$

where: $AC_{bus\_nom_i}$ – nominal acquisi-tion costs of bus [EUR], Bat_{cap} – battery capacity [kWh], $Bat_{unit_{Cost}}$ – cost of battery unit ca-pacity [EUR/kWh], Bus_{cost} – bus costs [EUR], $Capac_{cost}$ – costs of double-layer capacitors [EUR].

The economic, dynamic model of TCO provides several options of financing of in-vestment, taking into account different possibilities of combining own resources (self-financing), bank credit, subsidies and leasing:

$$AC_{bus\_nom_i} = AC_{bus\_cred\_i} + Bus_{self_i} + Bus_{sub_i} + Vbus\_init_i \quad (5)$$

where: $AC_{bus\_nom_i}$ – nominal acquisition costs of bus [EUR], $AC_{bus\_cred_i}$ – amount of credit for the bus purchase [EUR], Bus_{self_i} – costs of bus acquisition (self-financing) [EUR], Bus_{sub_i} – subsidies for bus [EUR], $Vbus\_init_i$ – value of the leased bus without initial fees.

Thus, various combinations of purchase financing sources are possible. In addition, according to the developed model different batches of purchased buses can be financed in different ways. In addition, the model takes into account that cash flows resulting from the purchase of a bus fleet can take place in different time periods, i.e. that the fleet is purchased in batches in subsequent years / periods (which is not a rare situation taking into account the specific procedures for this type of investment resulting from public procurement law). The subsidy can also take place in different time periods and can be paid in tranches (which happens quite often, especially for projects financed from European Union funds, when the payment is delayed after presentation of relevant accounting documents, etc.).

Each of these cash flows (CF) is discounted in time with a discount rate adequate to the period in which a given cash flow occurs, and Present Value is the sum of these discounted flows in time. Taking into consideration diversified funding structure of bus fleet acquisition in time the present value of the acquisition costs of electric buses is carried out using the following equation:

$$PV_{bus} = \sum_{i=1}^{n} (DAC_{bus\_cred_i} + DAC_{bus\_self_i} - DBus_{sub_i} + DBus_{leas_i} + DAC_{bat2_i} + DBatD_i) \quad (6)$$

where: DAC_{bus_cred} – discounted annual instalments for bus acquisition with credit [EUR], $DAC_{bus\_self_i}$– discounted costs of bus acquisition (self-financing) [EUR], $DBus_{sub_i}$ – discounted annual subsidies for bus acquisition [EUR], $DBus_{leas_i}$ – discounted annual bus lease instalments [EUR], DAC_{bat2_i} – discounted costs of spare battery for i-th bus[EUR], $DBatD_i$ – discounted costs of battery disposal for i-th bus [EUR], i – number of consecutive bus (i-th bus number), n – number of purchased buses.

The developed model of TCO includes two types of bank credit: i) Credit with equal installments (annuities), ii) Credit with decreasing installments (installments with fixed capital part).

6.5 Credit with equal installments (annuities)

The structure of installments is not the same for each period. In the initial periods, the share of interest is larger than the share of capital. Over time, the share of interest decreases and the share of capital increases. The calculation of nominal value of annual annuity is carried out by the equation:

$$AC_{bus\_cred\_ann_i} = AC_{bus\_cred} \frac{s_{bus}(1+s_{bus})^{n_{bus}}}{(1+s_{bus})^{n_{bus}}-1} \quad (7)$$

where: $AC_{bus\_cred_i}$ – amount of credit for the bus purchase [EUR], n_{bus} – number of payments/annuities, s_{bus} – credit interest rate (bus) [-].

6.6 Credit with decreasing installments (installments with fixed capital part)

In case of credit with decreasing installments each installment is with fixed capital part. Nominal value of annual credit installment for the installment number n_{inst} can be calculated using the following equation:

$$AC_{bus_{credinst\,i}}(n_{inst}) = \frac{AC_{bus\_cred_i}}{n_{bus}} \left[1 + (n_{bus} - n_{inst} + 1)\, s_{bus}\right] \quad (8)$$

where: $AC_{bus_{credinst\,i}}$ – amount of credit for the bus purchase [EUR], n_{bus} – number of payments/annuities, s_{bus} – credit interest rate (bus) [-], n_{inst}- number of credit installment.
TCO analysis is carried out annually, so if installments payments are monthly, then in order to receive the nominal annual value, the monthly installments of a given year should be added together. There is need to decide for which type of installments the analysis will be carried out. Usually bank credit with decreasing installments is cheaper than the credit with decreasing installments. In the next step to calculate the PV, installments are discount-ed with a discount rate adequate for the year in which such cash flows take place (credit repayments) and then Present Value is the sum of these discounted flows in time:

$$DAC_{bus_{cred i}} = \sum_{m_{bus}}^{M_{bus}} (AC_{bus\_cred\_ann\_i} \frac{1}{(1+i_{bus})^{m_{bus}}}) \quad (9)$$

or

$$DAC_{bus_{cred_i}} = \sum_{m_{bus}}^{M_{bus}} (AC_{bus\_cred\_inst\_i} \frac{1}{(1+i_{bus})^{m_{bus}}}) \quad (10)$$

where: DAC_{bus_cred} – discounted annual annuities for bus acquisition with credit [EUR], $AC_{bus\_cred\_ann\_i}$ – nominal value of annual credit installment (annuity) [EUR], $AC_{bus\_cred\_inst_i}$- nominal value of annual credit installment for the rate number n_{inst}, i_{bus} – market interest rate [-], M_{bus} – repayment term [years], i – number of consecutive bus (i-th bus number) , m_{bus} – time with credit installment for bus acquisition.

The investment can be financed from own resources, leasing or co-financed by a subsidy as well. In such case each cash flow are discounted with a discount rate adequate for the year in which such cash flows take place:

$$DAC_{bus\_self_i} = \sum_{m_{self}}^{M_{self}} (Bus_{self_i} \frac{1}{(1+i_{bus})^{m_{self}}}) \quad (11)$$

where: $DAC_{bus\_self_i}$– discounted costs of bus acquisition (self-financing) [EUR], Bus_{self_i} – costs of bus acquisition (self-financing) [EUR], m_{self} – time with payment (self-financing), M_{self} – investment term (self-financing).

$$DBus_{sub_i} = \sum_{m_{sub}}^{M_{sub}} (Bus_{sub_i} \frac{1}{(1+i_{bus})^{m_{sub}}}) \quad (12)$$

where: $DBus_{sub_i}$ – discounted subsidies for bus acquisition [EUR], m_{sub} – time with subsidy, M_{sub} – subsidized investment term, Bus_{sub_i} – subsidies for bus [EUR].

$$DBus_{leas_i} = \sum_{m_{leas}}^{M_{leas}} (AC_{bus_{leas_i}} \frac{1}{(1+i_{bus})^{m_{leas}}}) \quad (13)$$

where: $DBus_{leas_i}$– discounted annual annuities for bus acquisition with credit [EUR], $AC_{bus\_leas\_i}$ – nominal value of annual leas installment [EUR], i_{bus} – market interest rate [-], M_{leas} – bus leasing term [years], i – number of consecutive bus (i-th bus number), m_{leas} – time with bus lease installments.

The nominal value of annual leas installment ($AC_{bus\_leas\_i}$) may be the value entered by the user for example on the basis of offers received or calculated on the basis of the equation for an equal lease installment:

$$AC_{bus\_leas\_i} = \frac{Vbus\_init_i \, i_{bus_{leas}} (1 + \, i_{bus_{leas}})^{n_{lease}} - \, Vbus\_pur_i \, i_{bus_{leas}}}{(1 + \, i_{bus_{leas}})^{n_{lease}} - 1} \tag{14}$$

where: $AC_{bus\_leas\_i}$ – nominal value of annual leas installment [EUR], $i_{bus_{leas}}$ – lease interest rate [-], n_{lease} – number of lease installments, $Vbus\_init_i$ – value of the leased bus without initial fees, $Vbus\_pur_i$ – value of the leased bus purchase.

TCO analysis considers also costs of spare battery and costs of used battery disposal. Such costs are discounted with a discount rate adequate for the year in which such cash flows take place as follows:

$$DAC_{bat2_i} = AC_{bat2_{self}} \, (t = Bat_{life}) \, \frac{1}{(1 + i_{bat2} \,)^{Bat_{life}}} \tag{15}$$

where: DAC_{bat2} – discounted costs of spare battery [EUR], $AC_{bat2\_self} \;\; (t = Bat_{life})$ – acquisition costs of spare battery [EUR] after period t equal lifetime of the bus battery bat_life, i_{bat2} – market interest rate for bat2 [-], Bat_{life} – lifetime of the bus battery [years].

$$DBatD_i = \; BatD_i \, (t = Bat_{life}) \, \frac{1}{(1 + i_{bat} \,)^{Bat_{life}}} \tag{16}$$

where: $DBatD_i$ – discounted costs of the battery disposal for i-th bus [EUR], $BatD_i \;\; (t = Bat_{life})$ – costs of battery disposal for i-th bus [EUR] after period t equal lifetime of the bus battery bat_life, i_{bat} – market interest rate for bat1 [-], Bat_{life} – lifetime of the bus battery [years].

6.7 Operating Costs

The analysis of operating costs is conducted globally, i.e. for the entire fleet operating in a given year. This model takes into account not only the costs of future maintenance of buses planned to be bought. As part of the cash flow in the year "0" – so the year of conducting analysis and making decisions, it is possible to include operating costs of the fleet that the investor already has and forecast the value of these costs in the future as future cash flows, so discounting them with a discount rate appropriate for a given year using the following equation:

$$PVOC_{bus} = \sum_{m_{OC}}^{Bus_{life}} (OC_{bus} \frac{1}{(1 + i_{bus})^{m_{oc}}}) \quad (17)$$

where: OC_{bus}– annual buses operating costs [EUR], i_{bus} – market interest rate for bus [-], m_{OC} – time with operating costs, Bus_{life} – bus lifetime [years].

Nominal value of annual operating costs of electric buses is calculated as a sum of annual energy costs, maintenance costs, insurance, costs, costs of daily energy supply and other costs:

$$OC_{bus} = OC_{ener} + OC_{maint} + OC_{insur} + OC_{ener\_supp} + \mathbf{OC}_{other} \quad (18)$$

where: OC_{bus} – annual operating costs of the bus fleet [EUR], OC_{ener} – annual energy costs [EUR], OC_{maint} – annual maintenance costs [EUR], OC_{insur} – annual insurance cost [EUR], OC_{ener_supp} – annual costs of daily energy supply [EUR], OC_{other}- other annual costs (for example vehicle tax) [EUR].

Annual costs of the daily energy supply are calculated using the following equation:

$$OC_{ener\_supp} = Bus_{oper\_ann} Ener_{supp\_cost\_r} \quad (19)$$

where: OC_{ener_supp} – annual costs of the daily energy supply [EUR], Bus_{oper_ann} – annual transport work [vkm/year], $Ener_{supp\_cost\_r}$– energy supply cost rate [EUR/vkm].

The calculation of annual energy costs is carried out by the equation:

$$OC_{ener} = Bus_{oper\_ann} \left[Ener_{cons} \left(Ener_{cost} - Tax_{relief} \right) \right] \quad (20)$$

where: OC_{ener} – annual energy costs [EUR], Bus_{oper_ann} – annual transport work [vkm/year], $Ener_{cons}$ – energy consumption [kWh/vkm], $Ener_{cost}$ – cost rate of energy [EUR/kWh], Tax_{relief} – tax relief [EUR/kWh].

Annual maintenance costs depend on the number of staff service hours and staff service cost rate as follows:

$$OC_{maint} = Work_{staff\_service} Staff_{cost\_r} \quad (21)$$

where: OC_{maint} – annual maintenance costs [EUR], $Work_{staff\_service}$ – staff service hours [staff-service-hours/bus], $Staff_{cost\_r}$– staff service cost rate [EUR/staff-service-hours].
Other annual costs and annual insurance costs are user input values entered in years such cash flows take place.

6.8 Infrastructure costs

Infrastructure nominal acquisition costs in a given year include all types of charging infrastructure and are calculated using the following equation:

$$AC_{infra\_nom_y} = AC_{infra\_dep} + AC_{infra\_swap} + AC_{infra\_panto} + AC_{infra\_stop} + AC_{infra\_induct} \quad (22)$$

where AC_{infra_nom} – infrastructure nominal acquisition costs [EUR], AC_{infra_dep} – acquisition costs of depot conductive plug-in charging [EUR], AC_{infra_swap} – acquisition costs of battery swapping-charging [EUR], AC_{infra_panto} – acquisition costs of pantograph charging [EUR], AC_{infra_stop} – acquisition costs of on bus-stop charging [EUR], AC_{infra_induct} – acquisition costs of in-motion inductive charging [EUR],

The calculation of present value of infrastructure is carried out the same way as in previous cases, so as a sum of discounted nominal infrastructure acquisition costs by the equation:

$$PV_{infra} = \sum_{m_{infra\_ac}}^{M_{infra\_ac}} (AC_{infra\_nom_y} \frac{1}{(1 + i_{infra})^{m_{infra\_ac}}}) \quad (23)$$

where $AC_{infra\_nom_y}$- infrastructure nominal acquisition costs (in a given year), i_{infra} – market interest rate of infrastructure [-], M_{infra_ac} – repayment term of infrastructure [years], m_{infra_ac}- time with infrastructure acquisition.
Furthermore, the structure of financing the purchase or construction of an appropriate infrastructure may be analogous to that for a bus fleet, i.e. various combinations of own funds, credit, subsidies and leasing as follows:

$$AC_{infra\_nom_y} = AC_{infra\_cred} + AC_{infra\_self} + Infra_{sub} + V_{infra\_init} \quad (24)$$

where $AC_{infra\_nom_y}$ – infra-structure nominal acquisition costs (in a given year), AC_{infra_cred} – acquisition costs of infrastructure with credit [EUR], AC_{infra_self}- costs of infrastructure acquisition (self-financing) [EUR], $Infra_{sub}$ –subsidies for infrastructure [EUR], V_{infra_init} – value of the leased infrastructure without initial fees [EUR],

Including diversified investment funding structure and annual maintenance costs of infrastructure the present value of infrastructure is calculated using the following equation:

$$PV_{infra} = DAC_{infra\_cred} + DAC_{infra\_self} - DAC_{infra\_sub} + DAC_{infra\_lease} + DMC_{infra} \tag{25}$$

where: DAC_{infra_cred} – discounted annual installments for infrastructure acquisition with credit [EUR], DAC_{infra_self} – discounted acquisition costs of infrastructure (self-financing) [EUR], DMC_{infra} – discounted annual Maintenance Costs of infrastructure [EUR], DAC_{infra_sub} – discounted annual subsidies for infrastructure acquisition [EUR], DAC_{infra_lease}- discounted annual infrastructure lease installments.

Including investment funding from bank credit there is need to decide for which type of installments the analysis will be carried out (annuities or decreasing installments) as follows:

6.9 Credit with equal installments (annuities)

$$AC_{infra\_cred\_ann} = AC_{infra\_cred}\frac{s_{infra}(1+s_{infra})^{n_{infra}}}{(1+s_{infra})^{n_{infra}}-1} \tag{26}$$

where $AC_{infra\_cred\_ann}$ – annual annuity acquisition costs of infrastructure with credit [EUR], AC_{infra_cred} – acquisition costs of infrastructure with credit [EUR], n_{infra} – number of payments/annuities [-], s_{infra} –credit interest rate (infrastructure) [-],

6.10 Credit with decreasing installments (installments with fixed capital part)

$$AC_{infra_{cred(n_{inst})}} = \frac{AC_{infra\_cred}}{n_{infra}}\left[1+(n_{infra}-n_{inst}+1)\,s_{infra}\right] \tag{27}$$

where AC_{infra_cred} – acquisition costs of infrastructure with credit [EUR], n_{infra} – number of payments/annuities [-], s_{infra} –credit interest rate (infrastructure) [-], n_{inst}- number of credit installment.

In the next step to calculate the PV, installments are discounted with a discount rate adequate for the year in which such cash flows take place (credit repayments) and then Present Value is the sum of these discounted flows in time:

$$DAC_{infra\_cred} = \sum_{m_{infra}}^{M_{infra}} (AC_{infra\_cred\_ann} \frac{1}{(1+i_{infra})^{m_{infra}}}) \quad (28)$$

or

$$DAC_{infra\_cred} = \sum_{m_{infra}}^{M_{infra}} (AC_{infra\_cred(n_{inst})} \frac{1}{(1+i_{infra})^{m_{infra}}}) \quad (29)$$

where $AC_{infra\_cred\_ann}$ – nominal value of annual credit installments [EUR], i_{infra} – market interest rate of infrastructure [-], M_{infra} – repayment term of infrastructure [years], m_{infra}- time with annuity for infrastructure acquisition,

Infrastructure acquisition can be financed from own resources, leasing or co-financed by a subsidy as well. In such case each cash flow are discounted with a discount rate adequate for the year in which such cash flows take place:

$$DAC_{infra\_self} = \sum_{m_{infra_{self}}}^{M_{infra\_self}} (AC_{infra\_self} \frac{1}{(1+i_{infra})^{m_{infra\_self}}}) \quad (30)$$

where: i_{infra} – market interest rate of infrastructure [-], M_{infra_self} – investment term (self-financing) [years], m_{infra_self} - time with payment from own resources (infrastructure), AC_{infra_self}- costs of infrastructure acquisition (self-financing) [EUR].

$$DAC_{infra\_sub} = \sum_{m_{infra\_sub}}^{M_{infra\_sub}} (Infra_{sub} \frac{1}{(1+i_{infra})^{m_{infra\_sub}}}) \quad (31)$$

where: $Infra_{sub}$ –subsidies for infrastructure [EUR], i_{infra} – market interest rate of infrastructure [-], M_{infra_sub} – subsidized investment term (infrastructure) [years], m_{infra_sub} – time with subsidy.

$$DAC_{infra\_lease} = \sum_{m_{leas}}^{M_{leas}} (AC_{infra\_lease} \frac{1}{(1+i_{infra})^{m_{leas}}}) \quad (32)$$

where: AC_{infra_lease} – nominal value of annual leas installment (infrastructure) [EUR], i_{infra} – market interest rate of infrastructure [-], M_{leas} – infrastructure lease term [years], m_{leas} – time with infrastructure lease installments.

The nominal value of annual leas installment (AC_{infra_lease}) may be the value entered by the user for example on the basis of offers received or calculated on the basis of the equation for an equal lease installment:

$$AC_{infra\_lease} = \frac{V_{infra\_init}\ i_{infra\_leas}(1+\ i_{infra\_leas})^{n_{lease}} -\ V_{infra\_purch}\ i_{infra\_leas}}{(1+i_{infra\_leas})^{n_{lease}}-1} \qquad (33)$$

where: i_{infra_lease}– lease interest rate [-], n_{lease} – number of lease installments, V_{infra_init} – value of the leased infrastructure without initial fees, V_{infra_purch}– value of the leased infrastructure purchase.

Furthermore, annual maintenance costs of infrastructure are discounted with a dis-count rate adequate for the year in which such cash flows take place (credit repayments) and then Present Value is the sum of these discounted flows in time:

$$MC_{total} = MC_{infra\_dep} + MC_{infra\_swap} + MC_{infra\_panto} + MC_{infra\_stop} + MC_{infra\_induct} + MC_{other}$$

$$DMC_{infra} = \sum_{m_{infra_{life}}}^{Infra_{life}} MC_{total}\frac{1}{(1+i_{infra})^{m_{infra}}} \qquad (34)$$

where: DMC_{infra} – discounted annual Maintenance Costs of infrastructure [EUR], MC_{infra_dep} – annual maintenance costs of depot conductive plug-in charging [EUR], MC_{infra_swap} – annual maintenance costs of battery swapping-charging [EUR], MC_{infra_panto} – annual maintenance costs of pantograph charging [EUR], MC_{infra_stop} – annual maintenance costs of on bus-stop charging [EUR], MC_{infra_induct} – annual maintenance costs of in-motion inductive charging [EUR], MC_{other}- other annual costs (for example insurance, taxes) [EUR], i_{infra} – market interest rate of infrastructure [-], m_{infra_life} – time with maintenance costs of infrastructure, $Infra_{life}$ – infrastructure life [years].

6.11 External costs

TCO analysis includes external costs as well. The calculation of annual external costs is calculated using the following equation:

$$EC_{exter} = \; Bus_{oper\_ann}(Noise_{cost\_r} + EPoll_{cost\_r}) \; + \; Bus_{oper\_ann\_h} \; HPoll_{cost\_er} \qquad (35)$$

where: EC_{exter} – annual external costs of the bus fleet [EUR], Bus_{oper_ann} – annual transport work [vkm/year], $Bus_{oper\_ann\_h}$- annual transport work of bus fleet with oil heating [vkm/year], $Ener_{cons}$ – energy consumption [kWh/vkm], $EPoll_{cost\_er}$ – cost rate of air pollutant and GHG emission per vehicle-km [EUR/vkm], $Noise_{cost\_r}$ – cost rate of noise emission per 1 vehicle-km [EUR/vkm], $HPoll_{cost\_er}$ – cost rate of pollutant emission per 1 vkm (bus heating with oil) [EUR/vkm].

Cost rate for air pollutant emission includes marginal air pollution costs and margin-al well-to-tank costs.

In the next step annual external costs are discounted with a discount rate adequate for the year in which such cash flows take place and then Present Value is the sum of these discounted flows in time:

$$PVE_{exter} = \sum_{m_{EX}}^{Bus_{life}} (EC_{exter} \frac{1}{(1 + i_{bus})^{m_{ex}}}) \qquad (36)$$

where: PVE_{exter} – present value of the external costs [EUR], EC_{exter} – annual external costs of the bus fleet [EUR], i_{bus} – market interest rate for bus [-], Bus_{life} – bus lifetime [years], m_{EX} – time with external costs.

6.12 Liquidation value

Liquidation value of a bus is discounted with a discount rate adequate for the year in which such cash flows take place. The calculation of discounted proceeds of bus liquidation is carried out by the equation:

$$DPL_{liq\_bus_i} = \sum_{m_{liq}}^{Bus_{life}} AC_{bus\_nom_i} pl_r \frac{1}{(1 + i_{bus})^{m_{liq}}} \qquad (37)$$

where: Bus_{life} – bus life [years], m_{liq} – time with proceeds of bus liquidation, AC_{bus_nom} – nominal acquisition costs of bus [EUR], pl_r – residual value rate of the bus [-], i_{bus} – market interest rate [-].

The PV of proceeds of liquidation for e-bus fleet is calculated as sum of discounted cash flows in a given period for n buses as follows:

$$PVL_{liq\_bus} = \sum_{i=1}^{n} DPL_{liq\_bus_i} \quad (38)$$

where: DPL_{liq_bus} – discounted proceeds of i-th bus liquidation [EUR], i – number of consecutive bus (i-th bus number), n – number of purchased buses.

6.13 The application of TCO model – case study based on realistic operational scenario

The applicability of the developed socially-oriented dynamic model was verified using a case study based on realistic operational scenario. Conducted case study was based on assumptions that every 5 years starting from 2020 the fleet of new 10 electric buses is bought, and in total 40 buses. Each new batch of 10 buses had different funding structure:

- First batch: 15% self-financing and 85% subsidy,
- Second batch: 90% bank credit (decreasing installments) and 10% self-financing,
- Third batch: 30% self-financing and 70% in leasing,
- Fourth batch: 90% bank credit (annuities) and 10% subsidy.

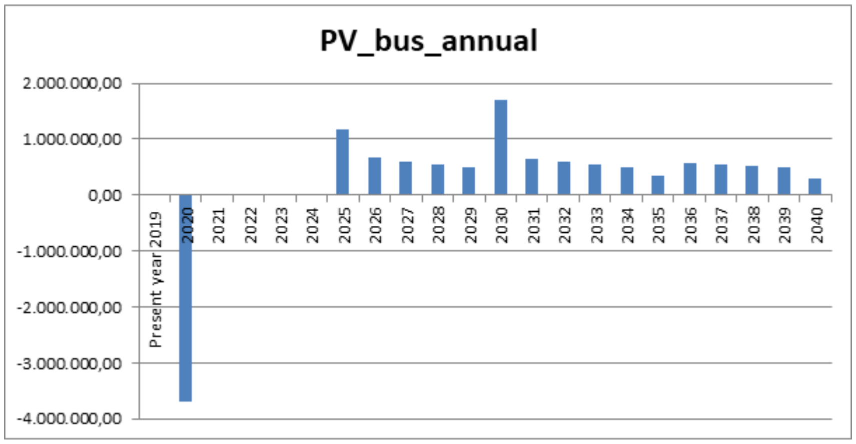

Figure 4: Present value of the acquisition costs of electric buses in given years of investment.

Table 11: Data used in case study of TCO calculation – bus fleet and infrastructure acquisition.

Item	Value	Unit
Nominal Acquisition Cost of Bus	554 000.00	EUR
Battery Capacity	240.00	kWh
Cost of Battery Unit Capacity	100.00	EUR/kWh
Bus cost (without battery)	500 000.00	EUR
Costs of double-layer capacitors	30 000.00	EUR
Acquisition costs of depot conductive plug-in charging (total 40 points)	1 000 000.00	EUR
Acquisition costs of on bus-stop charging (total 40 points)	1 200 000.00	EUR
Annual maintenance costs of depot conductive plug-in charging	400 000.00	EUR
Annual maintenance costs of on bus-stop charging	400 000.00	EUR
Other maintenance costs	40 000.00	EUR

The funding scenarios were diversified to examine the application of TCO model to different cases. Starting from 2025 also new spare batteries are bought, one spare battery for one bus. Assumed discount rate is 5% as recommended by the European Commission for the programming period 2014-2020 (European Commission, 2015).

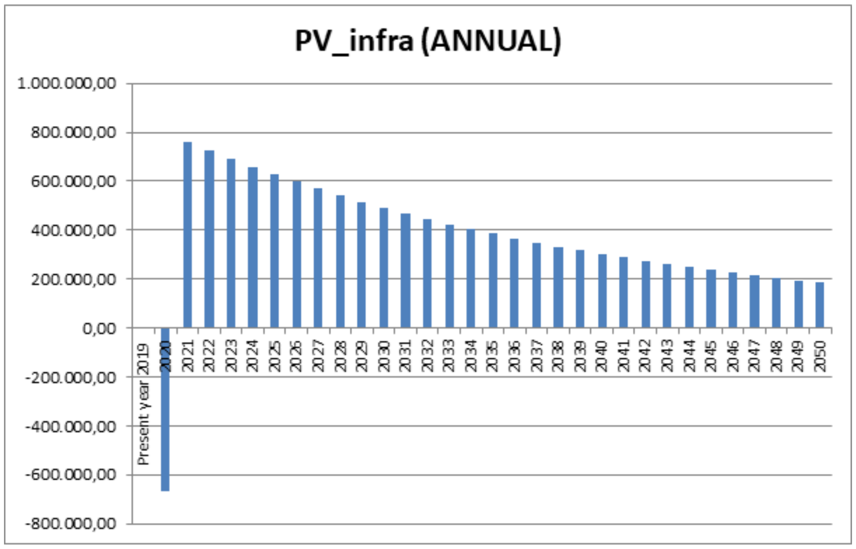

Figure 5: Present value of infrastructure in a given years of investment.

One depot conductive plug-in charger and on bus-stop charger per one bus is assumed. Infrastructure acquisition in 2020 and 2020 is also a first year of infrastructure maintenance and all costs related. The same infrastructure maintenance costs were assumed for 30 years (till 2050). The infrastructure acquisition is financed from own resources and subsidies (85% of the nominal cost of buses). It is assumed that all funds will be spent in one year. The entire subsidy is transferred in the same year as the buses were purchased. The main assumptions regarding bus and infrastructure acquisition made for the case study are presented in Table 11.

Results of conducted simulation regarding PV of acquisition costs of electric bus fleet and infrastructure are depicted in Fig. 4 and 5.

In the first year of investment (2020) the Present Value of Infrastructure (PV_{infra}) and acquisition costs of electric buses (PV_{bus}) is negative due to subsidy received which is higher than own expenditures. The main assumptions and data regarding operation and external costs are presented in Table 12.

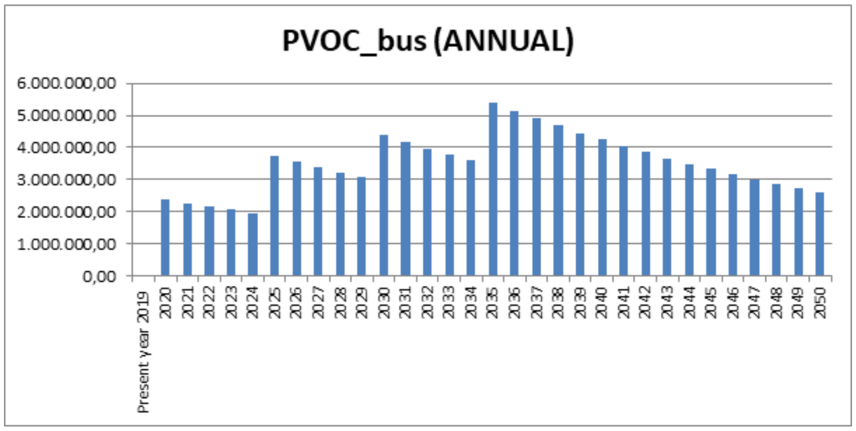

Figure 6: Present value of the buses operating costs in a given years of investment.

According to assumptions made one third of annual transport work is done with bus heating based on oil, which a source of external costs related to additional pollutant emission. This solution is used very often and results from business practice of public transport operators. Results of conducted simulation regarding PV of buses operating costs are depicted in Fig. 6. External costs rates that were assumed are presented in Table 13.

External costs rate are a certain average resulting from marginal costs rates (well-to-tank) for references cases in [12] European Union (European Commission, 2019). It was assumed that the liquidation value of all buses is in 2050 (assumed residual value rate of the bus 5%). The temporal course of annual TCO in the period from 2019 until 2050 is shown in Fig. 7. The final TCO value is over 234.6 million EUR, which means for this case it is 0.73 EUR per

Table 12: Operating and external costs.

Item	1st batch	2nd batch	3rd batch	4th batch
Years	2020-2024	2025-2029	2030-2034	2035-2050
Number of buses	10	20	30	40
Annual transport work	5 475 000	10 950 000	16 425 000	26 280 000
Staff service hours	21900	43800	65700	87600
Annual energy costs	1 182 600	2 365 200	3 547 800	5 676 480
Annual maintenance costs	109 500	219 000	328 500	438 000
Annual insurance costs	100 000	200 000	300 000	400 000
Annual costs of daily energy supply	1 095 000	2 190 000	3 285 000	5 256 000
Other costs	10 000	20 000	30 000	40 000
Annual Buses Operating Costs	2 497 100	4 994 200	7 491 300	11 810 480
Annual transport work of bus fleet with oil heating	1 825 000	3 650 000	5 475 000	8 760 000
Annual external costs	2 410 825	4 821 650	7 232 475	11 571 960

Table 13: External costs rates.

Item		
Cost rate of air pollutant and GHG emission per 1 vkm	0.32	EUR/vkm
Cost rate of noise emission per 1 vkm	0.067	EUR/vkm
Cost rate of pollutant emission per 1 vkm (bus heating with oil)	0.16	EUR/vkm

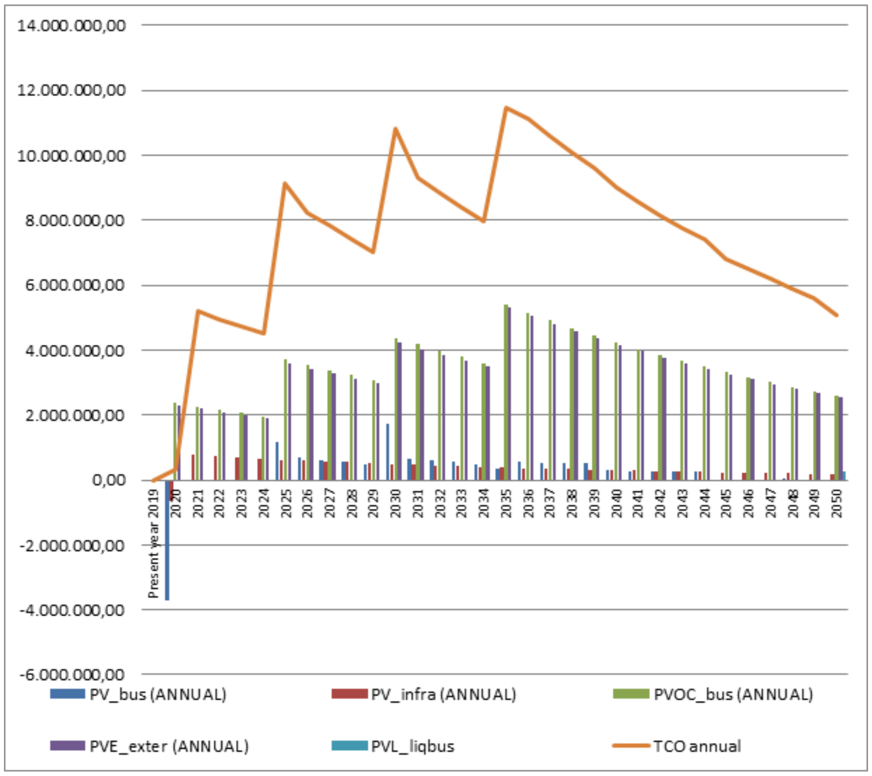

Figure 7: TCO annually in the period 2019-2050.

vehicle km based on the total mileage over the operational period (321.9 million of vehicle km in 2020-2050).

The simulation highlighted the importance of buses operating costs and external costs in the overall structure of TCO.

7 References

[1] Clerc, M. (2010) Particle Swarm Optimization.John Wiley & Sons.

[2] Garey, M.R., Johnson, D.S., 1979. Computers and intractability: A guide to the theory of NP-completeness. Freeman, San Francisco.

[3] Horn, W.A. (1974) Some simple scheduling algorithms. Naval Research Logistics Quarterly, 21, 177-185.

[4] Kennedy, J., Eberhart, R. (1995) Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942-1948.

[5] Pedersen, M.E.H., Chipperfield, A.J. (2010) Simplifying particle swarm optimization. Applied Soft Computing, 10, 618-628.

[6] Sengupta, S., Basak, S., Peters II, R.A. (2019) Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. arXiv:1804.05319.

[7] Steiner, G.,Yeomans, S. (1993) Level schedules for mixed-model, Just-In-Time assembly processes. Management Science, 39(6), 728-735.

[8] http://www.json.org/index.html.

[9] Bickert, S., Kuckshinrichs, W., 2011. Electromobility as a technical concept in an ecological mobility sector? An analysis of costs. 9th Int. Conf. Eur. Soc. Ecol. Econ. 2011 1–33.

[10] Ellram, L.M., 1999. Total Cost of Ownership, in: Handbuch Industrielles Beschaffungsmanagement. Gabler Verlag, pp. 595–607. https://doi.org/10.1007/978-3-322-99462-2_33.

[11] Ellram, L.M., 1995. Total cost of ownership; An analysis approach for purchasing. Int. J. Phys. Distrib. Logist. Manag. 25, 4–23. https://doi.org/10.1108/09600039510099928.

[12] European Commission, 2019. Handbook on the external costs of transport: Version 2019. Delft. https://doi.org/10.2832/27212.

[13] European Commission, 2015. Guide to Cost-Benefit Analysis of Investment Projects. https://doi.org/10.2776/97516.

[14] Hagman, J., Ritzén, S., Stier, J.J., Susilo, Y., 2016. Total cost of ownership and its potential implications for battery electric vehicle diffusion. Res. Transp. Bus. Manag. 18, 11–17. https://doi.org/10.1016/j.rtbm.2016.01.003.

[15] Lebeau, K., Lebeau, P., Macharis, C., Van Mierlo, J., 2014. How expensive are electric vehicles? A total cost of ownership analysis, in: 2013 World Electric Vehicle Symposium and Exhibition, EVS 2014. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/EVS.2013.6914972.

[16] Letmathe, P., Suares, M., 2017. A consumer-oriented total cost of ownership model for different vehicle types in Germany. Transp. Res. Part D Transp. Environ. 57, 314–335. https://doi.org/10.1016/j.trd.2017.09.007.

[17] Nurhadi, L., Borén, S., Ny, H., 2014. A sensitivity analysis of total cost of ownership for electric public bus transport systems in swedish medium sized cities, in: Transportation Research Procedia. Elsevier, pp. 818–827. https://doi.org/10.1016/j.trpro.2014.10.058.