Collaborative Interactive Wireless Charging in a Cyclic Mobispace

Rui Zhang[†], Sheng Zhang[†], Zhuzhong Qian[†], Mingjun Xiao[§], Jie Wu[‡], Jidong Ge[†], Sanglu Lu[†]
[†]State Key Lab. for Novel Software Technology, Nanjing University
[§]Suzhou Institute for Advanced Study, University of Science and Technology of China
[‡]Center for Networked Computing, Temple University
Corresponding author: sheng@nju.edu.cn

***Abstract*—Electric vehicle (EV) is a promising technological tool for diminishing environmental impact caused by gasoline-consumed transportation. Due to the limited battery capacity, EVs need to be charged frequently in a static charging station and thus waste large amounts of time being out of service. Research previously conducted in this topic have proposed solutions for deployment of charging lanes that can charge in-motion EVs. However, they cannot guarantee that every EV can be operational in their respective entire route. Meanwhile, we observe that EVs have repetitive motions and may cyclically encounter with each other which no prior research having been investigated. In addition, the development on the circuit design of energy transmit antennas can render EVs to be able to bi-directionally, highly efficiently transfer energy between themselves. These two observations enable us to distribute energy among EVs in a *collaborative* and *interactive* manner. We consider the cases of both loss-less and lossy energy transfer between EVs. In both cases, we formulate the problem of minimizing the time needed (or energy transferred) to reach a given energy distribution into a series of linear programming problems. When compared with a state-of-the-art algorithm, extensive simulation results show that the proposed algorithms can reduce the balancing time and energy loss by up to 70.60% and 36.59%, respectively.**

I. Introduction

With environmental concern increasing, novel energy technologies like electric battery have been developed along with the rapid evolution of electric vehicles (EVs) in the past few decades. The utilization of EVs has the potential to reduce greenhouse gases, which are largely caused by gasoline consumption of petrol-based vehicles [1][2]. However, due to the limitation of the battery capacity, EVs can only be operational for restricted distance until being charged by wired stationary chargers. In fact, by using traditional wired charging, EVs need to be frequently charged in static charging stations, thus waste large amounts of time being out of service.

To satisfy metropolitan transit demands, EVs must be continuously operable without recharging downtime [3]. With recent breakthroughs on wireless energy transfer [4] for in-motion EV charging, we can use wireless energy transfer to charge an EV when the EV is moving over a charging lane installed in the road [5]. In this way, EVs can be continuously operational. However, charging lanes are very expensive and thus cannot be widely deployed. Yan et al. proposed CatCharger [6] that solves the problem of deploying charging lanes with as less deploying cost as possible. However, in CatCharger, some EVs may visit charging lanes less frequently than others, which may cause them receive less energy that cannot enable them to work through the residual route. In this case, these EVs must change their routes to visit more charging lanes or stations, which is not desirable for metropolitan transit.

We are motivated by the following question: instead of relying on static charging stations or lanes, why not leverage the contacts between EVs to transfer energy between them so as to reach a desirable energy distribution? We have two key observations that inspire our work.

First, the development on the circuit design of energy transmit antennas can render an EV be able to *bi-directionally*, and highly efficiently, transfer energy [7][8]. Therefore, an EV becomes an energy transmitter as well as an energy receiver, which enables it to share energy with other EVs.

Second, Liu and Wu [9] suggested that most real objects have cyclic motion routines. For example, a factory produces a fixed amount of products in fixed time points and an EV is regulated to periodically transport these products to another fixed spot. Therefore, it is reasonable to assume that the movements of EVs have cyclic patterns. We use cyclic mobispace to denote such a scenario where EVs have this kind of predictable pattern. It is then feasible and reasonable to transfer energy from EVs that visit charging lanes frequently to those that visit charging lanes rarely.

These two observations enable us to distribute energy among EVs in a collaborative and interactive manner. We consider the cases of both loss-less and lossy energy transfer between EVs. In both cases, we formulate the problem of minimizing the time needed (or energy transferred) to reach a given energy distribution into a series of linear programming (LP) problems. We also show how to improve and generalize the proposed solutions. Our main contributions are summarized here:

- To the best of our knowledge, this is the first study of collaborative interactive wireless charging in a cyclic mobispace. We present a new problem statement regarding energy interaction.
- We consider the cases of both loss-less and lossy energy transfer between EVs and propose LP-based solutions to find the minimum time needed (or energy transferred) to reach a given energy distribution.

978–1–5386–2542–2/18/$31.00

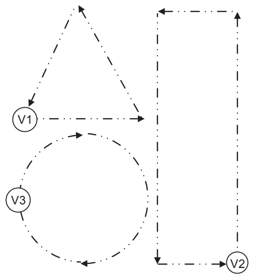


Fig. 1: An example of cyclic mobispace

TABLE I: Cyclic encounters

Time	EVs
5	v_1, v_3
30	v_1, v_3
34	v_1, v_2
39	v_2, v_3

- We evaluate our proposals using extensive simulations in several traces, and the results show the advantages of the proposed algorithms.

The rest of the paper is organized as follows: We first give brief background that motivates our problem and introduce the model in Section II. Then we present solutions for loss-less and lossy energy transfer in Sections III and IV, respectively. Extensions are discussed in Section V. Evaluation is given in Section VI. We survey related work in Section VII and conclude in Section VIII.

II. Model and Problem

A. Cyclic Mobispace

Human activities are usually cyclic [10][11], which makes vehicles inclined to have cyclic or repetitive mobility, since they are correlated with our activities and demands. We use *cyclic mobispace* to denote a scenario where vehicles have cyclic mobility and interaction patterns. More formally, a cyclic mobispace is a Euclidean space where EVs move in cyclic trajectories and thus interact with other EVs at fixed time slots in each cycle.

We denote by C the *common motion cycle* for all EVs. For example, the common motion cycle C for buses is a day. During each day, a bus is regulated to run in a fixed route at fixed time, and its contacts with other buses can be easily estimated. We divide time into small time slots of equal length. For a pair of vehicles, if their distance is no larger than the wireless charging distance at time slot t, they can interact with each other and exchange energy in this time slot.

Fig. 1 is a sample of a cyclic mobispace. Table I shows the encounters of the EVs in Fig 1 within one common motion cycle. In this figure, the three nodes[1] represent three EVs which do cyclic motion in their trajectories drawn with the dashed lines. As shown in this figure, vehicle v_1 and v_3 uniformly move in their triangular and circular trajectories respectively with a cycle time of 25 time slots each, while vehicle v_2 uniformly runs along its rectangle trajectory with a cycle time of 50 time slots. When the distance of the EVs is within their wireless charging range, the two EVs encounter each other and are capable of transmitting and receiving energy with each other. For example, in Fig. 1, at time slot 5, the distance between v_1 and v_3 is smaller than the charging distance, so they can share and interact energy at this time slot. Meanwhile, since the EVs do periodic motion along their trajectories, they repeat their movements and encounter patterns in the common motion cycle, which is the least common multiple of all cycles (e.g., $C = 50$ time slots in Fig. 1).

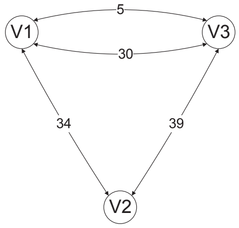


Fig. 2: The time-space graph of Fig. 1

We denote a cyclic mobispace by a time-space graph $G = (V, F, C)$, where V is the set of nodes representing EVs, F is the set of edges which represents the interaction time slots in a cycle C. Each edge $e \in F$ is a tuple (v_i, v_j, t_k) indicating that EVs v_i and v_j have a contact at time slot t_k.

We transform Fig. 1 into a time-space graph G shown in Fig. 2, where each edge represents an encounter between two corresponding vehicles. For example, v_1 and v_3 have two edges labelled 5 and 30, which indicates these two EVs meet at time slot 5 and 30 in one cycle.

B. Wireless Charging between EVs

Given a time-space graph G, each node represents an EV. Whenever two EVs meet, they can interact by exchanging energy between their respective battery cells bi-directionally.

We consider a set of n EVs $V = \{v_1, ..., v_n\}$. Obviously, due to the nature wireless energy technology, any transfer of energy ε induces *energy loss*. Similar to [7], we assume that the energy loss function $L(\cdot)$ satisfies a linear law:

$$L(\varepsilon) = \beta \cdot \varepsilon, \tag{1}$$

where $\beta \in (0, 1)$ is the wireless charging efficiency between two vehicles. As the EVs may remain moving when they encounter, the wireless charging efficiency keeps changing likewise. To simplify our problem, we assume β remains stable when EVs interact.

Therefore, if EVs v_i and v_j interact at the beginning of the t-th time slot (denoted by t^-) and v_i is planned to transfer energy ε to v_j, the new energy levels of v_i and v_j at the end of this time slot (denoted by t^+) become

$$\begin{aligned} E_{t^+}(i) &= E_{t^-}(i) - \varepsilon \\ E_{t^+}(j) &= E_{t^-}(j) + (1 - \beta) \cdot \varepsilon \end{aligned} \tag{2}$$

Whenever two vehicles interact, they can only transmit and receive limited energy so that their energy after interaction is neither too much to exceed the battery capacity nor too little to support their respective remaining movements. To ensure this, we introduce two energy thresholds: E_{max} and E_{min}; whenever there is an energy transfer, we make sure that the energy of either vehicle after interaction is between E_{min} and E_{max}. In general, we set E_{max} to be the capacity of a battery

[1] We will use node and EV interchangeably throughout the paper.

cell, and E_{min} to be the minimum energy that can support the remaining movement of a vehicle until the next cycle.

C. Problem

We assume that the initial energy level of vehicle v_i at the beginning is $E_0(v_i)$. In order to focus on the collaborative interactive charging between EVs, we assume the initial energy level of an EV represents the overall energy it can obtain from charging stations or lanes during the time period of interest.

Definition 1: Energy distribution — The energy distribution $\mathbf{D}_t$ of n EVs at time slot t is defined as the vector

$$[\frac{E_t(v_1)}{\sum_i E_t(v_i)}, \frac{E_t(v_2)}{\sum_i E_t(v_i)}, ..., \frac{E_t(v_n)}{\sum_i E_t(v_i)}]$$

.

Now we can present our problem:

Problem 1: Given a cycle mobispace denoted by $G = (V, F, C)$, and the initial energy level $E_0(v)$ of every EV v in V, how can we get an *interaction plan* X to reach a given energy distribution $\mathbf{D} = [d_1, d_2, ..., d_n]$, where $\sum d_i = 1$?

III. Loss-less Energy Transfer

In this section, we focus on minimizing the time needed to reach a given energy distribution in the case of loss-less energy transfer, i.e., $\beta = 0$ in Eq. (1).

A. Formulation

When two EVs meet to reach a given energy distribution, it is not enough to just take current contact into account. Future contacts and energy thresholds should also be taken into consideration. We use the example in Fig. 3 to illustrate this. There are 4 EVs with $E_{min} = 10$ and $E_{max} = 100$. The contacts in a common motion cycle $C = 50$ are shown in this figure. Without loss of generality, we present several examples throughout this paper to reach a uniform distribution of energy, i.e., $\mathbf{D} = [\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}]$.

Capacity thresholds should be taken into consideration. When v_1 and v_3 encounter with each other at time slot 9, although we know the final energy level of them is $\frac{90+18+90+90}{4} = 72$, we cannot directly let v_1 transfer 18 units of energy to v_3, because v_3 cannot store so much energy.

Future contacts should be considered. To reach a uniform energy distribution in Fig. 3, all of v_1, v_3, and v_4 should transfer (directly or indirectly) their respective energy to v_2, because only $E_0(v_2)$ is below the average energy level, which is 72. Since there is no direct contact between v_1 and v_2, v_1 has to rely on v_3 to indirectly transfer its energy to v_2. Hence, when v_1 and v_3 encounter with each other at time slot 9, a simple idea is to let v_1 transfer energy to v_3. However, in the optimal solution, which is shown in Table III, the energy transfer happens in the *opposite* direction.

These findings prohibit us from designing a combinatorial algorithm. Hence, we resort to linear programming.

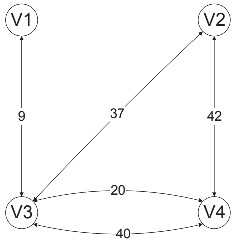


Fig. 3: A 4-node example used in the rest of the paper

TABLE II: Initial energy levels

EV	$E_0(v_i)$
v_1	90
v_2	18
v_3	90
v_4	90

We use $X_t(i, j)$ to denote the amount of energy transferred from vehicle v_i to vehicle v_j at the t-th time slot. All these $X_t(i, j)$s constitute an interaction plan X. Note that since wireless energy transfer only happens when two vehicles meet, we know

$$X_t(i, j) = 0, \quad \forall (v_i, v_j, t) \notin F. \tag{3}$$

The energy level of vehicle v_i at the end of time t is its initial energy plus the energy that is received from other EVs minus the energy that is transmitted to other EVs. $E_t(v_i)$ can be denoted as:

$$E_t(v_i) = E_0(v_i) + \sum_{\substack{t' \in [0,t] \\ (v_j, v_i, t') \in F}} X_{t'}(j, i) - \sum_{\substack{t' \in [0,t] \\ (v_i, v_j, t') \in F}} X_{t'}(i, j). \tag{4}$$

Our goal in this loss-less case is to minimize the number of time slots needed to reach the energy distribution $\mathbf{D} = [d_1, d_2, ..., d_n]$, where $\sum d_i = 1$. Denote by T the number of time slots needed. We can formulate this problem as follows:

[Loss-Less Problem, LLP]

$$\begin{aligned} \min \quad & T & \text{(5a)} \\ \text{s.t.} \quad & E_{min} \le E_t(v_i) \le E_{max}, \quad \forall t \in [0, T], \forall v_i \in V & \text{(5b)} \\ & E_T(v_i) = d_i \sum_{v_i \in V} E_0(v_i), \quad \forall v_i \in V & \text{(5c)} \\ & X_t(i, j) \ge 0, \quad \forall t \in [0, T], \forall (v_i, v_j, t) \in F & \text{(5d)} \end{aligned}$$

Eq. (5b) indicates the energy restriction for every vehicle at every time slot; Eq. (5c) ensures that the energy distribution at time slot T is the designed energy distribution $\mathbf{D}$. However, this optimization problem is not a standard linear programming (LP), which can be easily solved using well-developed methods.

In the following, we propose to transform LLP into a series of LP problems (the number of which is bounded) to find the minimum time T.

B. Solution

We introduce LAL (shown in Alg. 1) to solve this problem. The input variable B is a parameter we introduced to restrict the searching space in LAL, in order to avoid our algorithm from infinite searching loop. The BinaryFindT algorithm referred in LAL is shown as Alg. 2.

The key observation that enables us to develop LAL is that LLP with a fixed T is a standard LP problem. Based on this, we propose to employ a binary search to find the minimum T by checking whether Eq. (6) has a feasible solution. Given a fixed T, Eq. (6) can be solved efficiently using lpsolve [12].

Algorithm 1 LP-based Algorithm for LLP (LAL)

```
Input: time-space graph $G(V, F, C)$, initial energy levels $E_0(v_i)$, $E_{min}$, $E_{max}$, $B$
Output: $T$, $X$
1: $T \leftarrow C$, $cnt \leftarrow 0$
2: if LLP($T$, $E_{min}$) has a feasible solution then    ▷ lpsolve [12]
3:     $(T, X) \leftarrow$ BinaryFindT($T$, $E_{min}$, $cnt$)
4: else
5:     while $cnt < B$ do
6:         $T \leftarrow 2T$, $E_{min} \leftarrow 2E_{min}$, $cnt \leftarrow cnt + 1$
7:         if LLP($T$, $E_{min}$) has a feasible solution then
8:             $(T, X) \leftarrow$ BinaryFindT($T$, $E_{min}$, $cnt$)
9: if $cnt \geq B$ then
10:     return cannot reach the given distribution
11: return $T$, $X$
```

Algorithm 2 BinaryFindT

```
Input: $T$, $E_{min}$, $cnt$
Output: $T$, $X$
1: $t_h \leftarrow T$
2: if $cnt = 0$ then
3:     $t_l \leftarrow 0$
4: else
5:     $t_l \leftarrow 2^{cnt-1} \cdot C$
6: $t_m \leftarrow (t_h + t_l)/2$
7: while $(v_i, v_j, t) \in F$ and $t_l < t < t_h$ do
8:     if LLP($t_m$, $E_{min}$) has a feasible solution $(T', X')$ then
9:         $T \leftarrow T'$, $X \leftarrow X'$    ▷ Save results
10:        $t_h \leftarrow t_m$
11:    else
12:        $t_l \leftarrow t_m$
13:    $t_m \leftarrow (t_h + t_l)/2$
14: return $T$, $X$
```

LAL starts by checking whether Eq. (6) has a feasible interaction plan when T is equal to C, the common motion cycle (line 2). If yes, we use a binary search (Alg. 2) to find the minimum T between 0 and C that makes Eq. (6) have a feasible interaction plan. Otherwise, we check whether Eq. (6) with $T = 2C$ has a feasible interaction plan, and either use Alg. 2 to search between C and $2C$, or continue increase T. In this algorithm, we need to pay attention to three issues.

[LLP(T,E_{min})]

$$\min \quad 0 \tag{6a}$$

$$\text{s.t.} \quad E_{min} \leq E_t(v_i) \leq E_{max}, \quad \forall t \in [0, T], \forall v_i \in V \tag{6b}$$

$$E_T(v_i) = d_i \sum_{v_i \in V} E_0(v_i), \quad \forall v_i \in V \tag{6c}$$

$$X_t(i, j) \geq 0, \quad \forall t \in [0, T], \forall (v_i, v_j, t) \in F \tag{6d}$$

First, in LAL, if we cannot find a feasible interaction plan with current T, we will double it (line 6 in Alg. 1). Correspondingly, the minimum energy needed for an EV to support its movements is also increased, so we double E_{min} every time we double T (line 6 in Alg. 1).

Second, we constrain the searching space in LAL by parameter B. We currently set B to be 3, i.e., we search the optimal value of T within the range between 0 and $2^3C = 8C$. When the counter cnt increases to B, i.e., LLP cannot have a feasible interaction plan that needs time within 2^BC, we report that LLP cannot reach the given distribution (lines 9-10 in Alg. 1).

Third, in the binary search phase (lines 7-13 in Alg. 2), when will the algorithm stop searching? Our problem is different from a traditional binary search. We know two EVs can exchange energy only when there is a contact (i.e., an edge in the time-space graph denoting the cyclic mobisapce). Therefore, although we search T in the range of $[0, 2^BC]$, the final value of T must be in the form of

$$T = kC + t,$$

where

$$k \in \{0, 1, 2, ..., 2^B - 1\}, \text{ and } t \in \{t | \exists (v_i, v_j, t) \in F\}.$$

Hence, when there is no edge that occurs between the current search range $[t_l, t_h]$, we do not need to continue searching (line 7 in Alg. 2).

TABLE III: Interation plan for Fig. 3 using LAL

Time	Interaction	Transmitted energy	Energy levels (v_1, v_2, v_3, v_4)
0			(90, 18, 90, 90)
9	$v_3 \to v_1$	8	(98, 18, 82, 90)
20	$v_4 \to v_3$	18	(98, 18, 100, 72)
37	$v_3 \to v_2$	54	(98, 72, 46, 72)
59	$v_1 \to v_3$	26	(72, 72, 72, 72)

C. Example

For example, Table III is the energy interaction plan of Fig. 3 using LAL when $\mathbf{D} = [\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}]$. In general, each of v_1, v_3 and v_4 needs to transmit 18 units energy to v_2. However, v_1 can only interact with v_3 at time slot 9 when v_3 can only store 10 more units energy (since $E_{max} = 100$). Therefore, it is infeasible to distribute the energy evenly among nodes within only one cycle. When we expand the graph to two cycles, v_1 has one more opportunity to interact with v_3 at time slot 59. Hence, v_1 can transfer 18 units energy to v_3 within these two encounters[2].

D. Summary

For the running time of LAL, in the worst case, we need to solve the LLP(T, E_{min}) problem B times until we get the first feasible solution. Then we have to run BinaryFindT(T, E_{min}, $B - 1$) with $T \in [2^{B-1}C, 2^BC]$ to search for the minimum time needed. Therefore, we have to solve the linear programming problem $B + \log(2^BC - 2^{B-1}C) = B + \log(2^{B-1}C) = 2B - 1 + \log C$ times in the worst case.

[2] Note that, the optimal solution shown in Table III obtained by LAL makes v_3 transfer 8 units energy to v_1 at time slot 9 and v_1 transfer 26 units energy to v_3 at time slot 59.

As we mentioned before, we use the lpsolve tool to solve the linear programming problem. lpsolve is a free linear programming solver based on the revised simplex method [12]. Although linear programming may require an exponential number of steps in the worst case, in theory LP can be solved within a polynomial time [13]. Denote by $O(f(x))$ the time complexity of solving a LP problem with an input of size x in binary format, then the overall time complexity of LAL is $O((2B + \log C)f(x))$.

In this section, we formulate the collaborative interactive wireless charging problem in the loss-free case and design an LP-based solution LAL.

IV. Lossy Energy Transfer

In this section, we consider a more general case, where every transfer of energy ε induces non-zero energy loss $\beta\varepsilon$, $0 < \beta < 1$.

A. Formulation

Energy loss complicates the problem in three aspects.

First, when EVs interact with each other to collaboratively reach a given energy distribution, energy loss happens. Minimizing the time needed to reach an given energy distribution may increase energy loss. Hence, in contrast to minimizing the number of time slots, we seek to minimize the energy loss in this section. Note that energy loss is always proportional to energy transferred. Our goal is equivalent to minimizing the energy transferred to reach a given energy distribution.

Second, energy loss makes the overall energy of all EVs decrease. Unlike the loss-less case where we know the final energy level of each EV before we run our algorithm, in this lossy case, we cannot know the final energy level of each EV until we find an exact interaction plan.

Third, when two EVs have a contact, energy can be transferred in two directions in the loss-less case. However, in the lossy case, to minimize the energy loss, energy should be only transferred in one of the two directions.

In the light of these issues, we now present the problem formulation. When two EVs interact at time slot t and vehicle v_i transfer $X_t(i,j)$ energy to v_j, the energy level of vehicle v_i decreases by $X_t(i,j)$ while the energy level of vehicle v_j only increases by $(1-\beta)X_t(i,j)$. Therefore, the energy level of vehicle v_i at the end of time t is:

$$\begin{aligned} E_t(v_i) = {} & E_0(v_i) \\ & + (1-\beta) \sum_{\substack{t'\in[0,t] \\ (v_j,v_i,t')\in F}} X_{t'}(j,i) - \sum_{\substack{t'\in[0,t] \\ (v_i,v_j,t')\in F}} X_{t'}(i,j). \end{aligned} \tag{7}$$

Because of the energy loss, the available energy in the network decreases when there is an energy transfer between EVs. Hence, the total energy level of n EVs at time slot t is:

$$\sum_{v_i\in V} E_t(v_i) = \sum_{v_i\in V} E_0(v_i) - \beta \sum_{\substack{t'\in[0,t] \\ (v_i,v_j,t')\in F}} X_{t'}(i,j). \tag{8}$$

Our goal in this lossy case is to minimize energy loss to reach the given energy distribution $\mathbf{D} = [d_1, d_2, ..., d_n]$, where $\sum d_i = 1$. We assume that the time needed to reach the given energy distribution is bounded by T. We can formulate this problem as follows:

TABLE IV: Interaction plan for Fig. 3 by simply solving LYP(C,E_{min})

Time	Interaction	Transmitted energy	Energy level (v_1, v_2, v_3, v_4)
0			(90, 18, 90, 90)
9	$v_1 \rightarrow v_3$	46.15	(43.85, 18, **126.92**, 90)
9	$v_3 \rightarrow v_1$	26.92	(65.39, 18, 100, 90)
37	$v_3 \rightarrow v_2$	34.62	(65.39, 45.7, 65.38, 90)
42	$v_4 \rightarrow v_2$	24.62	(65.39, 65.39, 65.38, 65.38)

[LossY Problem, LYP(T,E_{min})]

$$\min \quad \beta \sum_{\substack{t\in[0,T] \\ (v_i,v_j,t)\in F}} X_t(i,j) \tag{9a}$$

$$\text{s.t.} \quad E_{min} \le E_t(v_i) \le E_{max}, \quad \forall t \in [0,T], \forall v_i \in V \tag{9b}$$

$$E_T(v_i) = d_i \sum_{v_i\in V} E_T(v_i), \quad \forall v_i \in V \tag{9c}$$

$$X_t(i,j) \ge 0, \quad \forall t \in [0,T], \forall (v_i, v_j, t) \in F \tag{9d}$$

B. Solution

Given a fixed T, LYP(T,E_{min}) is a standard LP problem and can be solved efficiently[3]. However, we find that if T is not set to a proper value, solving LYP(T,E_{min}) may return some unreasonable interaction plan. We use the following example to better explain this.

Table IV is the interaction plan of Fig. 3 by simply solving LYP(C,E_{min}) when $\mathbf{D} = [\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}]$. In order to transmit enough energy from v_1 to v_3, the energy is transmit *bi-directionally*. This is unreasonable for two reasons. First, extra energy transfer induces extra energy loss; second, the energy level of v_3 is temporarily beyond E_{max}. This situation occurs when the given time-space graph cannot distribute energy evenly within the given time (which is C in this Table).

The main reason for such unreasonable interaction plan is that the final energy level of each EV in the given distribution changes, when there is an energy transfer between two EVs; therefore, to reach the given energy distribution, two EVs in a contact can intentionally transfer energy to each other to quickly arrive at the given distribution. This motivates us to propose LAY as shown in Algorithm 3.

LAY starts by checking whether Eq. (9) has an *effective* interaction plan when T is equal to C, the common motion cycle (line 2 in Alg. 3). If yes, we return the interaction plan. Otherwise, we check whether Eq. (9) with $T = 2C$ has an effective interaction plan, and then use BinaryFindT2 to find the minimum T between C and $2C$ or continue increasing T. In this procedure, there are a few issues we would like to explain as follows.

First, what is an effective interaction plan? In an interaction plan for the lossy case, if there is an edge $(v_i, v_j, t) \in F$

[3] As we mentioned in Section III, E_{min} is proportional to T. For example, when T increases from C to $2C$, E_{min} doubles as well.

Algorithm 3 LP-based Algorithm for LYP (LAY)

```
Input: time-space graph $G(V, F, C)$, initial energy levels $E_0(v_i)$, $E_{min}$, $E_{max}$, $B$
Output: $T$, $X$
$T \leftarrow C$, $cnt \leftarrow 0$
if LYP($T$, $E_{min}$) has an effective[3] solution then
    return $T$, $X$
else
    while $cnt < B$ do
        $T \leftarrow 2T$, $E_{min} \leftarrow 2E_{min}$, $cnt \leftarrow cnt + 1$
        if LYP($T$, $E_{min}$) has an effective solution then
            $(T, X) \leftarrow$ BinaryFindT2($T$, $E_{min}$, $cnt$)
if $cnt \geq B$ then
    return cannot reach the given distribution
return $T$, $X$
```

such that $X_t(v_i, v_j) > 0$ and $X_t(v_j, v_i) > 0$, we say it is ineffective; otherwise, it is an effective interaction plan.

Second, what is the difference between BinaryFindT and BinaryFindT2? If we change line 8 of BinaryFindT (Alg. 2) to "LYP(t_m, E_{min}) has an effective solution (T', X')", then we have BinaryFindT2.

Third, if LYP(T, E_{min}) has an effective interaction plan with $T = C$, we just return the result. (Remember that in Algorithm 2, we continue to find the minimum T between 0 and C using BinaryFindT.) This is because, in the lossy case, our primary goal is to minimize the amount of energy loss; thus, if the time needed to reach the given energy distribution is not large (i.e., $T \leq C$), we would not try to optimize it. However, if the time is larger than C we still optimize it using BinaryFindT2.

C. Example and Summary

Table V shows the energy interaction plan returned by LAY on the example in Fig. 3 while $\mathbf{D} = [\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}]$. As shown in Table IV, we cannot have an effective solution within one cycle. Hence, we increase T and try to find an effective solution with $T = 2C$, and so forth. Once we find an effective interaction plan (bi-directional energy transfer does not exist), although minimizing energy cost is our primary concern, reducing time is another important objective. Therefore, we use BinaryFindT2(T, E_{min}, cnt) to minimize T. Table V gives the final interaction of our algorithm which presents a good tradeoff between energy loss and time.

As a brief summary of this section, we formulate the collaborative interactive wireless charging problem in the lossy case and design an LP-based solution LAY.

V. Discussion

In the current design of BinaryFindT, it searches a given range in a traditional binary search manner. We present here two alternatives that improve the original BinaryFindT.

TABLE V: Interaction plan for Fig. 3 using LAY

Time	Interaction	Transmitted energy	Energy level (v_1, v_2, v_3, v_4)
0			(90, 18, 90, 90)
9	$v_1 \rightarrow v_3$	12.5	(77.5, 18, 100, 90)
37	$v_3 \rightarrow v_2$	40	(77, 50, 60, 90)
42	$v_4 \rightarrow v_2$	22.22	(77, 67.78, 60, 67.78)
59	$v_1 \rightarrow v_3$	9.72	(67.78, 67.78, 67.78, 67.78)

First, in BinaryFindT as shown in Alg. 2, we actually search for a precise edge that provides the least time to reach the given energy distribution. In our present algorithm, we simply halve the search range. However, in some cases, the encounters of EVs may distribute unevenly in a cycle. When we halve the search range $[t_l, t_h]$, the subrange we are going to search may contain more than half of the edges in $[t_l, t_h]$. If this happens, we cannot arrive at the final result quickly, because the search range is not halved after each iteration.

We provide an alternative method for this issue: denote by $[t_l, t_h]$ the current range to be searched, then we

- sort the edges in $\{(v_i, v_j, t) | t_l \leq t \leq t_h\}$ in descending order of t;
- set t_m to t' such that
$$|\{(v_i, v_j, t) | t_l \leq t \leq t'\}| = |\{(v_i, v_j, t) | t' \leq t \leq t_h\}|,$$
that is, partition $\{(v_i, v_j, t) | t_l \leq t \leq t_h\}$ into two equal-size subsets by t'.

Second, we can take advantage of the interaction plan to further reduce the searching range. Denote the current range by $[t_l, t_h]$ to be searched. By inspecting the interaction plan, we find there is t' such that

$$X_t(v_i, v_j) = 0, \quad \forall (v_i, v_j, t) \in F, \forall t \in [t', t_h]$$

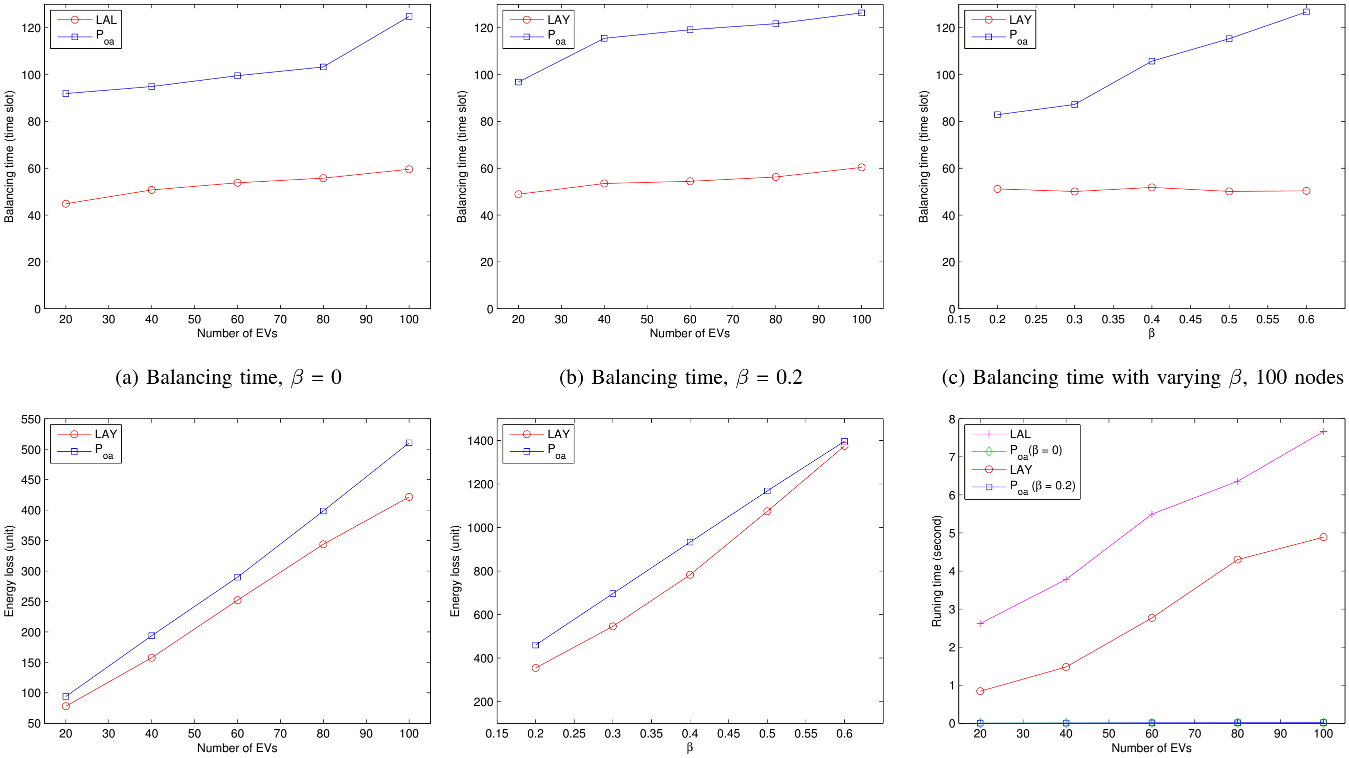
That is to say, no energy exchange happens between t' and t_h. Therefore, it is not necessary to search within the range between t' and t_h, and we successfully reduce the original range $[t_l, t_h]$ to $[t_l, t']$.

For example, as shown in Table IV, when LLP(T, E_{min}) cannot give a feasible interaction in one cycle, we set $T = 2C$. We find LLP($2T$, $2E_{min}$) has a feasible solution, then we use BinaryFindT to search for the minimum time needed. By the original design of BinaryFindT, we then have to solve LLP($1.5T$, $1.5E_{min}$), and so forth. However, after we carefully checking the returned interaction plan, we find that after $t' = 59$, there is no energy exchange. Thus, we quickly reduce the search range to $[50, 59]$. Since there is no edge with a timestamp between 50 and 59, we know the minimum time needed to reach a uniform distribution is exactly 59.

VI. Performance Evaluation

In this section, we evaluate our protocols in the context of another algorithm using two traces: random trace and synthetic bus trace under different settings and reveal insights of the performance.

[3] In an interaction plan for the lossy case, if there is an edge $(v_i, v_j, t) \in F$ such that $X_t(v_i, v_j) > 0$ and $X_t(v_j, v_i) > 0$, we say it is ineffective; otherwise, it is an effective interaction plan.



(a) Balancing time, β = 0

(b) Balancing time, β = 0.2

(c) Balancing time with varying β, 100 nodes

(d) Energy loss, β = 0.2

(e) Energy loss with varying β, 100 nodes

(f) Running time

Fig. 4: Random trace

A. *Simulation Setup*

We conduct several simulations using MATLAB R2014a. Since there currently are no existing works that focus on collaborative interactive charging in a cyclic mobispace, we use a heuristic algorithm P_{oa} [7] for comparison. P_{oa} is an interaction protocol that provides the direction and quantity of each energy transfer by every agent locally estimating the total energy of the whole agents without any previously provided information. Since the purpose of P_{oa} is to make energy balanced in the network of mobile agents, we set the final energy distribution $\mathbf{D}$ to be $[\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}]$ in our experiments. Obviously, P_{oa} cannot make use of the cyclic information of EVs, which makes it hard to reach an absolute balanced energy distribution as our algorithm does. For comparison purposes, we use the standard deviation σ as the parameter to monitor the distance between the energy level of EVs using P_{oa} and the absolute balanced energy distribution.

B. *Random Trace and Results*

We simulate random time-space graphs with different number of EVs with initial energy levels generated in a uniform distribution. The common motion cycle C is 50 time slots. We set the maximum energy level of EVs $E_{max} = 100$ units and the minimum energy level $E_{min} = 10\% \times E_{max}$. The bound coefficient B for iterative binary searching is set to be 4. For statistical smoothness, we repeat our experiments 100 times at each setting and depict the average result of each set of simulations. To compare with P_{oa}, when the standard deviation σ is less than 5, we consider the energy distribution obtained by P_{oa} almost reaches the balanced energy distribution.

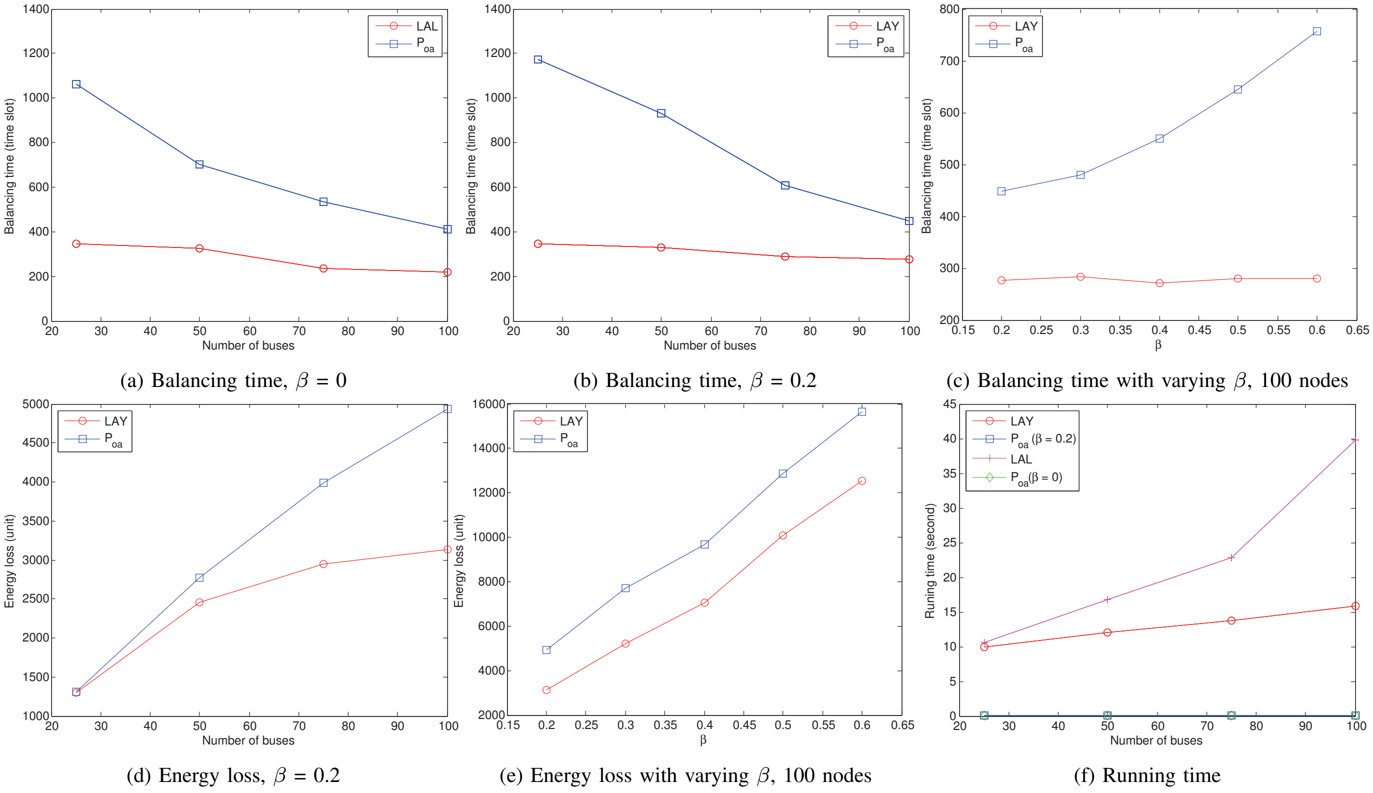
The results of the random trace are depicted as Fig. 4 and the details of analysis are shown as follows:

1) Balancing time: In this section, the time when the energy of EVs distributes evenly (almost evenly by P_{oa}) in the network is called balancing time.

Fig. 4(a), (b) show the balancing time in the random trace which respectively compared LAL and LAY with P_{oa} when $\beta = 0$ and $\beta = 0.2$. As shown in these figures, although we allow approximate energy balance for P_{oa}, our algorithms have overwhelming superiority in both lossless and lossy cases. Both LAL and LAY can return an optimal interaction plan in about one common motion cycle, while P_{oa} needs to use over two cycles to reach the approximate energy balance. When the number of EVs increases, the balancing time of all algorithms increases as well, which is reasonable because it is hard to arrive a uniform energy distribution among a larger number of nodes in random trace.

In addition, the balancing time of LAL is usually smaller than that of LAY. As we mentioned in Section IV, if we have an effective interaction plan when $T = C$, to minimize the energy loss, LAY will not keep searching for a smaller balancing time, while LAL does.

Fig. 4(c) shows the impact of the energy loss factor β on balancing time. When β increases, it has little influence on



(a) Balancing time, β = 0

(b) Balancing time, β = 0.2

(c) Balancing time with varying β, 100 nodes

(d) Energy loss, β = 0.2

(e) Energy loss with varying β, 100 nodes

(f) Running time

Fig. 5: Synthetic bus trace

the balancing time of LAY but greatly affects the balancing time of P_{oa}, which indicates that the proposed LAY has better tolerance for balancing time than P_{oa}.

2) Energy loss: In this section, the total energy loss due to energy transfer from the beginning to the balancing time is called energy loss.

Fig. 4(d), (e) show the energy loss with varying numbers of EVs and β in random trace, respectively. In Fig. 4(d), the increasing rate of energy loss in LAY is smaller than that of P_{oa}. In Fig. 4(e), although LAY costs less amount of energy loss compared with P_{oa}, the increasing rate of LAY is larger than that of P_{oa}. There are two reasons for this phenomenon: first, LAY not only minimizes energy loss, but also minimizes balancing time when the balancing time is larger than one cycle; second, the time-space graphs in random trace have more randomness which is not totally suitable in our problem (compared with synthetic bus trace shown in section VI-C which has more desirable results).

3) Running time: Fig. 4(f) shows the running time of LAL, LAY, and P_{oa} with varying number of EVs. Since LAL and LAY rely on solving a series of linear programming problems to reach the uniform energy distribution, their running time is much higher than that of P_{oa}. However, the problem we deal with is not an online problem, and the result we obtain can be used for multiple times or a very long period of time as long as the cyclic mobispace remains unchanged. Therefore, sacrificing running time for less energy loss or less balancing time is reasonable.

C. Synthetic Bus Trace and Results

We generate synthetic bus traces based on the realistic metropolitan bus routes in Nanjing, China. Each bus travels a route from the beginning station to the ending station. In order to imitate the periodical mobility of the cyclic mobispace, when one bus reaches the ending station, it immediately reverts to the beginning station along its route. Therefore, the motion cycle of a bus is the round trip time on its route. We assume the cost each bus takes to travel between two consecutive stations is 5 minutes and each bus stays at each station for 1 minute. When several buses stay in the same station at the same time slot, they can interact and exchange energy between them. We randomly select 25, 50, 70, 100 bus routes and assign motion cycles to these buses on their routes according to the number of stations on that route. The motion cycles for different routes are distributed from 240 minutes to 300 minutes. We set the common motion cycle C as 300 minutes. When a bus finishes a round trip on its route but it takes the time less than the common motion cycle, the bus needs to wait in its beginning station for the rest time of the common motion cycle.

We set $E_{max} = 1000$ units and $E_{min} = 10\% \times E_{max}$. The bound coefficient B is 4 and the common motion cycle C is 300 time slots. We generate the initial energy levels of the buses in a uniform distribution and repeat the experiments 100 times at each setting. The standard deviation threshold σ for P_{oa} is set as 50.

Intuitively speaking, the outcome of synthetic bus trace shown as Fig. 5 is better than that of random trace shown as Fig. 4. This is because synthetic bus trace is generated from realistic bus routes, while random trace is randomly generated with little realistic information. Obviously, the synthetic bus trace is more appropriate to simulate actual cyclic human activities and interactions in reality where our algorithms perform better in synthetic bus trace than in random trace.

The details of the results of synthetic bus trace depicted in Fig. 5 are analyzed as follows:

1) Balancing time: Fig. 5(a), (b) show the balancing time in synthetic bus trace which respectively compared LAL and LAY with P_{oa} when $\beta = 0$ and $\beta = 0.2$. With the increasing number of buses, the results of balancing time in our algorithms remain stable as about one common cycle, which represent the optimal results. Meanwhile, the balancing time of P_{oa} is over three times than the common cycle when there are 25 buses whenever $\beta = 0$ or $\beta = 0.2$ and gradually declines along with the increasing number of buses. The result of P_{oa} still remains a certain distance from our algorithms. In reality, the more buses we add, the more frequently these buses can encounter in the stations along their routes. In Fig. 5 (b), LAY reduces the balancing time by up to 70.60% compared with P_{oa}.

Fig. 5(c) shows the impact of β on LAY and P_{oa}. Apparently, the result of LAY remains stable, which indicates that LAY is rarely influenced by β while the result of P_{oa} increases drastically along with the increasing β. This result shows that LAY has a much batter tolerance on balancing time than P_{oa}.

2) Energy loss: Fig. 5(d) shows the energy loss of LAY and P_{oa} when $\beta = 0.2$. Although two algorithms have similar energy loss when there are 25 buses, as the number of buses increases, the energy loss using P_{oa} grows almost linearly while the energy loss of LAY gradually tends to be slow and flat. This indicates our algorithm has great superiority in reducing energy loss. When the number of buses is 100, LAY reduces the energy loss by up to 36.59% compared with P_{oa}.

Fig. 5(e) shows the impact of β on energy loss. Apparently, β which indicates the wireless charging efficiency significantly influences energy loss of both LAY and P_{oa}. With the increasing of β, the energy loss of LAY and P_{oa} both increases almost linearly. Interestingly, no matter how β changes, the energy loss of LAY always remains a stable number of energy less than that of P_{oa}.

3) Running time: As mentioned in section VI-B, although our algorithms need more time to complete, running time is not a vital consideration in our problem.

D. An Example of LAL and LAY on 10 EVs

Fig. 6 and Fig. 7 show the energy level of each EV over time using LAL and LAY, respectively. We make three interesting observations.

- First, obviously the overall energy trend of each EV is approaching the final average energy level. Over half of the lines in the figures are between their initial energy level and the final average energy level, while a small part of EVs have drastic energy variations between $[E_{min}, E_{max}]$. This phenomenon indicates that a small part of the EVs are energy carriers and transporters that redistribute the energy within the cyclic mobispace.

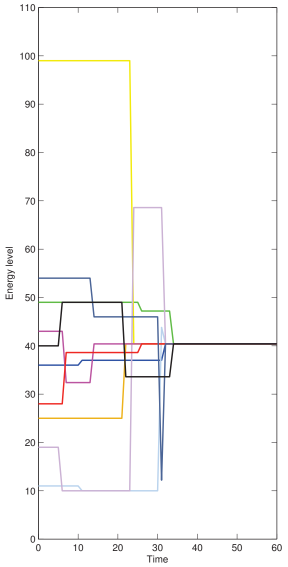


Fig. 6: Energy level of each EV over time using LAL (10 EVs are represented by 10 different colors)

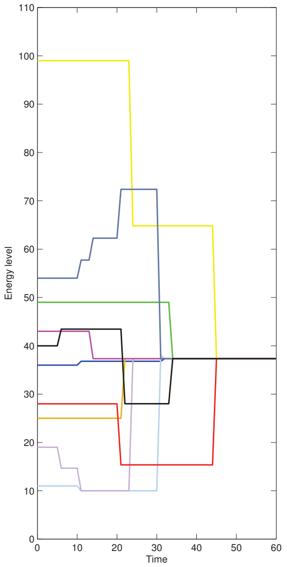


Fig. 7: Energy level of each EV over time using LAY (10 EVs are represented by 10 different colors)

- Second, some lines in Fig. 6 are more sharp than some lines in Fig. 7. Since LAL does not need to worry about energy loss, the energy in LAL can be transferred whenever necessary. But as LAY needs to minimize energy loss, the energy in LAY need to be used carefully, which results in smoother lines in LAY.
- Third, LAL reaches the uniform energy distribution more quickly than LAY. For example, the balancing time in Fig. 6 is 34, while it is 45 in Fig. 7. This indicates that LAL provides a lower bound of the balancing time.

E. Summary

In this section, we evaluate LAL and LAY with simulations compared with P_{oa}. While the balancing time using P_{oa} raises when the number of nodes increases or the wireless charging efficiency β increases, the balancing time of LAL or LAY changes little. Meanwhile, LAY costs less energy to reach the uniform energy distribution compared with P_{oa}. Although our algorithms runs more slowly than P_{oa}, we consider it worthwhile, because once we obtain an optimal interaction plan, the plan can be used as long as the network does not change.

VII. Related Work

Wireless energy transfer is a promising technology, which have been lately investigated both in static sensor networks and in mobile agent networks. Some of these works employ static or mobile chargers to periodically replenish rechargeable static sensors such as RFID tags with the purpose of optimizing charging quality [14], minimizing total charging delay [15],charging itinerary selection [16], etc. Although the proposed solutions are marvelous, they all assume that the nodes to be charged are fixed in location and ignore the cases of mobile agent network.

There are also a large amount of works focusing on wireless charging in networks where agents are mobile. Jang *el al.* formulated an optimization problem to deploy wireless charging lanes to support in-motion buses with the minimum cost [5]. Yan *el al.* investigated the problem of how to deploy charging lanes in a metropolitan road network to minimize the deployment cost [6]. Nikoletseas *el al.* proposed an interactive wireless charging for energy balance [7]. Chen *el al.* focused on charging path optimization and charger scheduling problems [17]. Different from these works, our work focus on how to distribute energy to reach a given distribution $\mathbf{D}$ using the techniques of bi-directional energy transfer while the movement of each EV is cyclic.

VIII. Conclusions and Future Work

With the development of wireless charging techniques, in-motion EVs can be charged along their moving trajectories and consequently remain operational. Previous works mainly focus on how to deploy wireless charging lanes, but they cannot guarantee that every EV can be operational in their respective entire route. In this paper, we allows EVs to receive energy from not only charging stations/lanes but also other EVs, which results in a collaborative and interactive energy distribution framework. To the best of our knowledge, we are the first to propose the notion of collaborative interactive wireless charging. We give a new problem formulation and consider it in cases of both loss-less and lossy for which we design two algorithms LAL and LAY. Extensive simulation results show that, compared with a state-of-the-art algorithm, the proposed algorithms can reduce the balancing time and energy loss by up to 70.60% and 36.59%, respectively.

There are a number of research directions we plan to investigate in our future work. First, we did not consider the deployment of charging lanes which is related to the uneven initial energy distribution. Therefore, the problem on how to deploy charging lanes in a cyclic mobispace to enable every EV keep operational with the minimum cost is an important work we intend to do. Second, in reality, the amount and efficiency of the energy interaction is not as idealistic as we assumed. On the one hand, we implicitly assumed that any amount of energy transfer can be finished when there is a contact between two EVs. But in reality, two EVs can only exchange a small amount of energy that depends on their meeting time. On the other hand, as the EVs keep moving when they encounter, the efficiency which is largely affected by the distance would be dynamic in the real world. Thus, we are also going to investigate the energy distribution problem with these constraints. Third, we assumed the motions of EVs are purely cyclic. But there are inevitable fluctuations in speed, congestion, etc in real world, which make the cyclic mobispace change. Consequently, how to adjust to the new mobispace is worth exploring.

IX. Acknowledge

This work was partially supported by National Key R & D Program of China (2017YFB1001801), NSFC (61502224, 61472181, 61472185), Ministry of Education & China Mobile Research Foundation (MCM20170307), Jiangsu NSF (BK20151392, BK20151390), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] A. Y. Lam, Y.-W. Leung, and X. Chu, “Electric vehicle charging station placement,” in *Proceedings of the 4th International Conference on Smart Grid Communications (SmartGridComm 2013)*, 2013, pp. 510–515.

[2] Y. Zheng, Z. Y. Dong, Y. Xu, K. Meng, J. H. Zhao, and J. Qiu, “Electric vehicle battery charging/swap stations in distribution systems: comparison study and optimal planning,” *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 221–229, 2014.

[3] K. C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury, L. Yu, C. Qiu, and V. Soundararaj, “A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC),” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 491–509, 2016.

[4] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljačić, “Wireless power transfer via strongly coupled magnetic resonances,” *Science*, vol. 317, no. 5834, pp. 83–86, 2007.

[5] Y. J. Jang, E. S. Suh, and J. W. Kim, “System architecture and mathematical models of electric transit bus system utilizing wireless power transfer technology,” *IEEE Systems Journal*, vol. 10, no. 2, pp. 495–506, 2016.

[6] L. Yan, H. Shen, J. Zhao, C. Xu, F. Luo, and C. Qiu, “Catcharger: Deploying wireless charging lanes in a metropolitan road network through categorization and clustering of vehicle traffic,” in *Proceedings of the 36th International Conference on Computer Communications (INFOCOM 2017)*, 2017.

[7] S. Nikoletseas, T. P. Raptis, and C. Raptopoulos, “Interactive wireless charging for energy balance,” in *Proceedings of the 36th International Conference on Distributed Computing Systems (ICDCS 2016)*. IEEE, 2016, pp. 262–270.

[8] M. del Prete, A. Costanzo, A. Georgiadis, A. Collado, D. Masotti, and Z. Popovic, “Energy-autonomous bi-directional wireless power transmission (WPT) and energy harvesting circuit,” in *Proceedings of 2015 International Microwave Symposium (IMS 2015)*, 2015, pp. 1–4.

[9] C. Liu and J. Wu, “Routing in a cyclic mobispace,” in *Proceedings of the 9th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2008)*. ACM, 2008, pp. 351–360.

[10] V. Srinivasan, M. Motani, and W. T. Ooi, “Analysis and implications of student contact patterns derived from campus schedules,” in *Proceedings of the 12th International Conference on Mobile Computing and Networking (MobiCom 2006)*. ACM, 2006, pp. 86–97.

[11] M. Motani, V. Srinivasan, and P. S. Nuggehalli, “Peoplenet: engineering a wireless virtual social network,” in *Proceedings of the 11th International Conference on Mobile Computing and Networking (MobiCom 2005)*. ACM, 2005, pp. 243–257.

[12] “lpsolve,” *http://lpsolve.sourceforge.net/5.5/*.

[13] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *Proceedings of the 16th ACM Symposium on Theory of Computing (STOC 1984)*. ACM, 1984, pp. 302–311.

[14] S. Zhang, Z. Qian, F. Kong, J. Wu, and S. Lu, “P^3: Joint optimization of charger placement and power allocation for wireless power transfer,” in *Proceedings of the 34th International Conference on Computer Communications (INFOCOM 2015)*. IEEE, 2015, pp. 2344–2352.

[15] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, “Minimizing charging delay in wireless rechargeable sensor networks,” in *Proceedings of the 32th International Conference on Computer Communications (INFOCOM 2013)*. IEEE, 2013, pp. 2922–2930.

[16] S. Zhang, Z. Qian, J. Wu, F. Kong, and S. Lu, “Optimizing itinerary selection and charging association for mobile chargers,” *IEEE Transactions on Mobile Computing*, pp. 2833–2846, 2016.

[17] L. Chen, S. Lin, and H. Huang, “Charge me if you can: charging path optimization and scheduling in mobile networks.” in *Proceedings of the 17th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2016)*, 2016, pp. 101–110.