Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

An Efficient Recharging Task Planning Method For Multi-Robot Autonomous Recharging Problem

Jingwen Xu, Jingchuan Wang, Weidong Chen

***Abstract*—As robots are increasingly employed in the warehouse operation, an important problem that needs to be solved is the Autonomous Recharging Problem (ARP); that is deciding when and where to recharge robots. Former research in ARP is disjoint that only focuses on one of above two questions, considering the high computation time. In this paper, we describe an efficient recharging task planning method to solve ARP and it includes two phases: In the first phase, recharging task planning based on task sequence segmentation is proposed for individual robot, in which both when to recharge and where to recharge are considered to maximize operational efficiency. And it also improves the computation efficiency by segmenting task sequence and planning each recharging task recursively. In the second phase, all robots task sequences with planned recharging tasks are centralized and selected recharging stations of some planned recharging tasks are reallocated, using maximum bipartite graph matching to reduce waiting time for other robots recharging and improve overall task execution efficiency. The proposed method is validated by means of simulation in a real warehouse environment.**

I. INTRODUCTION

In warehousing scenarios, multiple mobile robots are often used to perform tasks such as handling and transportation. Considering the limited battery capacity, robot needs to recharge during executing task sequence and Autonomous Recharging Problem influences task execution efficiency. For robots with given task allocation, solving ARP is equivalent to plan recharging tasks in the task sequence, where their selected recharging station should also be decided. And it depends on the task sequence itself for the individual robot and is also influenced by other robots, considering that multiple robots share recharging stations.

Paper [1] defined the autonomous recharging problem as two main problems: when to recharge and where to recharge. Current research in autonomous recharging is disjointed and focuses predominantly on addressing one of above two problems:

For solving when to recharge, a fixed threshold method is widely used in practical scenarios, such as [2], [3]. However, the method often has some deficiencies. The inaccuracy of threshold setting can easily lead to the failure of the robot. In order to overcome this problem, [4]-[6] developed the adaptive threshold method, [7] - [9] developed different power consumption models for different types of robots, avoiding recharging failure caused by inaccurate threshold. However,in the above research [2]-[9], recharging is considered only when the battery capability is not enough. A method based on mixed integer programming in [10] is proposed to solve when to recharge. In this method, the number of recharging tasks is given, and the order of recharging tasks in the task sequence is a decision variable aiming to minimize the task completion time. However, there are enough recharging stations with ideal one-to-one match for each robot, that is not correspondent with practice. Paper [1] proposes a method based on exhaustive search which is operating in the decentralized structure. In [1], each possible recharging task numbers and their orders in the task sequences are search out and optimal solution is the final result for individual robot. Recharging station selection of these recharging tasks in [1] always follows nearest neighbor principle. [1],[10] solved when to recharge optimally compared with [2]-[9]. However, there are also some deficiencies such as ignorance of where to recharge and influence of different robots. And computation complexity increases quickly when task numbers becomes large.

For solving where to recharge, [11] proposed a fuzzy inference based method where recharging station priority for each robot is decided by its residual power, distance and ideal recharging time. However, [11] concentrated on individual robot and ignore the influence of other robots, [12] proposes a market-based method. In this method, recharging tasks executed at the same time is regarded as a bidder, and bid for the shared recharging station in the single-round auction process. [13] proposed a centralized improved genetic algorithm to solve where to recharge. In this method, order of recharging task is determined by threshold method such as [2]-[3], the recharging station selection of each recharging task is regarded as an individual and forms population, method iterated repeatedly in the process of crossover and mutation. The is to minimize the overall task completion time. Although [12] - [13] coordinates the selection of recharging stations among multiple robots to improve the overall efficiency, ignoring when to recharge still limits the optimization of the method. However, the computation of this methods is complex and real-time performance is limited, considering that it scheduling recharging tasksselected recharging stations in the whole task sequence for all robots.

In this paper, when to recharge and where to recharge is simultaneously considered for solving ARP. And we described a efficient recharging task planning method. The method include two phases: In the first phase, for individual robot,recharging task planning based on the task sequence segmentation is proposed, it recursively segments the task sequence in the

This work is supported by the National Natural Science Foundation of China under grants No. 61773261.

Jingwen Xu, Jingchuan Wang and Weidong Chen are with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Institute of Medical Robotics, Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Corresponding author: Jingchuan Wang, email: jchwang@sjtu.edu.cn.

individual robot and plan single recharging task at each time; In the second phase, all robots task sequences are centralized and some recharging tasks are coordinated. A method based on maximum bipartite graph matching is proposed and reallocate these recharging tasks' recharging stations, aiming to improve overall recharging and task completion time.

The rest of this paper is organized as follows. Section II provides the main definitions of relevant variables in the method that will be used in this paper. In Section III, a detailed description of the method is given. Simulation validation results are reported in Section IV. Section V concludes our study.

II. PROBLEM STATEMENT

In this paper, the warehousing scenario involves N robots.Robots set is denoted as R= $[r_1, r_2...r_N]$. The allocated task sequence of robot r_i is given and denoted as T_i^c= $\left[task_i^1, task_i^2...task_i^{M_i}\right]$ where M_i represents the task number and r_i executes each task according to the order in T_i^c. Each task represents the depot that robot must arrive at in T_i^c. The original electricity is denoted as E_{0,r_i},original location is denoted as C_{0,r_i} and original time is denoted as t_{0,r_i} for r_i.Its velocity is denoted as v.The electricity consumption model is described as follows:

$$E_{r_i}(t) = \begin{cases} E_{0,r_i} - \beta vt & \text{if } v!{=}0 \\ E_{0,r_i} - \alpha t & \text{if } v{=}0 \end{cases} \tag{1}$$

In the (1), E_{0,r_i} represents r_i's residual electricity with t seconds. β and α represent the electricity consumption coefficients while robot is waiting or moving respectively.

In addition, H static recharging stations are also included in the warehousing scenario. Recharging station set is denoted as $S = [s_1, s_2...s_h]$. The unit time for one recharging station can only afford one robot recharging but allow multi robots waiting and its recharging rate per second is denoted as e .

In this research, recharging tasks are planned and coordinated in each T_i^c for solving ARP. The j th recharging task in T_i^c is denoted as $rt_{r_i}^j$. The global objective is to minimize the total task completion time such that

$$J = min\left(max[time(T_1^c), time(T_2^c)...time(T_N^c)]\right) \tag{2}$$

where T_i^c represent the time when r_i finish T_i^c.

III. AN EFFICIENT RECHARGING TASK PLANNING METHOD

Fig. 1 shows the architecture of our method including two phases: recharging task planning, multiple recharging tasks coordination. Firstly, single recharging task is recursively planned in T_i^c by task sequence segmentation for each r_i. Secondly, some of planned recharging tasks belonging to different robots are chose and coordinated considering the influence of multiple robots based on maximum bipartite graph matching.

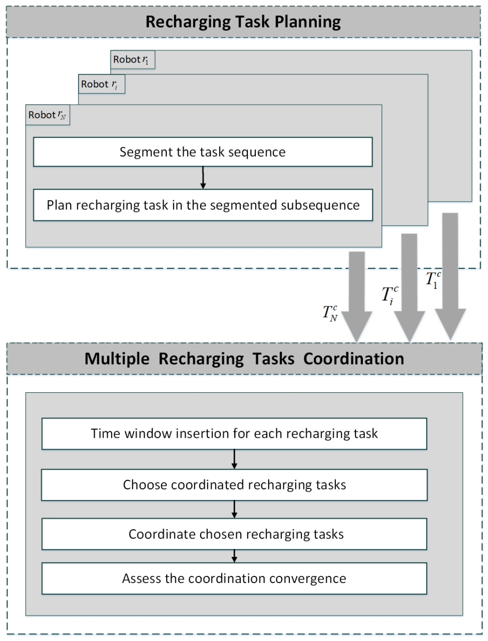
Recharging Task Planning
Robot r_1
Robot r_i
Robot r_N
Segment the task sequence
Plan recharging task in the segmented subsequence
T_N^c
T_i^c
T_1^c
Multiple Recharging Tasks Coordination
Time window insertion for each recharging task
Choose coordinated recharging tasks
Coordinate chosen recharging tasks
Assess the coordination convergence

Fig. 1. The method architecture

A. Recharging Task Planning

As showed in Fig. 1, $Phase\,A$ is decentralized and runs in each robot. In this phase, T_i^c is recursively segmented and planned single recharging task by configuring a integer programming. This method reduces computation complexity compared with planning all possible recharging tasks in the whole task sequence. It include two steps as follows:

1) Segment task sequence:

In this step, T_i^c is recursively segmented many times and task sequence that needs to be segmented is denoted as T_i^s which is equal to T_i^c for the first time. In the each time segmentation, T_i^s is divided into two subsequences, which represent respectively the sequence that need to plan recharging task and sequence that waits for next time segmentation and planning. A task sequence segmentation strategy is proposed as follows:

Given the T_i^s= $\left[task_{q_1}^i...task_q^i...task_{q_2}^i\right]$, find $task_q^i$ in T_i^s which satisfies Eq(3) which indicates that r_i can only execute tasks until $task_q^i$ without recharging. Then T_i^s is segmented into two parts with $task_q^i$ as boundary. For example, in the situation of Fig. 2, there are three times segmentation for r_0 in T_0^s and T_0^s is segmented into $T_0^{s1} = [task_1, task_2, task_3]$ and $T_0^{s2} = [task_4, task_5, task_6, task_7]$ in the first time segmentation and a recharging task rt_1 is planned in T_0^{s1}.

$$\begin{cases} E^i - \beta(\sum_{j=q_1}^{q-1} d_{task_j}^{task_{j+1}} + d_{task_q}^{C^i}) > 0 \\ E^i - \beta(\sum_{j=q_1}^{q} d_{task_j}^{task_{j+1}} + d_{task_q}^{C^i}) \leqslant 0 \end{cases} \tag{3}$$

In Eq(3), C^i and E^i represent the original location and electricity in the segmentation which is equal to C_{0,r_i}, E_{0,r_i} at the first time segmentation. And $d_{task_j}^{task_{j+1}}$ represents the shortest distance between $task^i_{j+1}$ and $task^i_j$.

2) Plan recharging task in the segmented subsequence:

In this step, single recharging task rt^i_p with its selected recharging station is planned and inserted in the T^{s1}_0 by building a integer formulation and solving it. The integer formulation includes constant, decision variable, constraint and objective function as follows:

- Constant:

C^i:robots original location before segmentation and planning

E^i:robots original electricity before segmentation and planning

T^{s1}_i:task sequence for recharging task planning and $T^{s1}_i = [task^i_{q_1}, task^i_{q_1+1}...task^i_q]$

N^{s1}:task numbers in T^{s1}_0

$d^{task_{i+1}}_{task_i}$:The shortest distance between $task_i$ and $task_i + 1$

$d^{task_i}_{s_h}$:The shortest distance from $task_i$ to s_h

- Decision Variable:

$X_{task_n - s_h}$:binary variable that tasks the value of 1 if robot decides to go to for recharging after finishing n th task in the T^{s1}_i and 0 otherwise.

- Constraints:

current planned recharging task number constraint:

It is needed to plan only one recharging task in the segmented task sequence just as Eq(4) shows.

$$\sum\sum X_{task_i - s_h} = 1, task_i \in T^{s1}_i, s_h \in S \tag{4}$$

residual power constraint:

Eq(5) indicates that planning solution should make sure that robot finishes the following tasks in the T^{s1}_0 after recharging.

$$100 - (\sum_{j=n+1}^{q-1} d^{task_{j+1}}_{task_j} + d^{task_{n+1}}_{s_h})\beta > 0, if X_{task_n - s_h} = 1 \tag{5}$$

total planned recharging task number constraint:

Compared with planning all possible recharging task in the overall task sequence,this method improve the computation complexity, but it can also influence the global optimization if the total planned recharging task number is far from the optimal solution. So the total recharging task number constraint is set to make sure that current planning solution would not lead global planned recharging task number to exceed estimated value. In the Eq(6),MAX represents the estimated total recharging task number in the T^c_i and can be computed in advance when $PhaseA$ starts.And p represents the current planned rt^i_p is the pth recharging task in the T^c_i.

$$\frac{|100 - (\sum_{j=n+1}^{M_i - 1} d^{task_{j+1}}_{task_j} + d^{task_n}_{s_h})\beta|}{100} < MAX - p \quad if X_{task_n - s_h} = 1 \tag{6}$$

$$MAX = (\frac{|E^i_0 - \beta(\sum_{j=0}^{M_i - 1|} d^{task_{j+1}}_{task_j})|}{100} * 1.5) + 1 \tag{7}$$

- Objective function:

The recharging task planning is to minimize the task completion time of T^{s1}_i ,such that

$$F = min \sum\sum X_{task_n - s_h} t^{task_n - s_h}_f \tag{8}$$

where $t^{task_n - s_h}_f$ represents task completion time if r_i decides to go to s_h for recharging after finishing $task_n$ in the T^{s1}_i such that:

$$\begin{cases} t^{task_n - s_h}_f &= t^i + t^{task_n - s_h}_{charging} + t^{task_n - s_h}_{run} \\ t^{task_n - s_h}_{charging} &= \frac{100 - (E_i - (\sum_{j=0}^{n-1} d^{task_{j+1}}_{task_j} + d^{s_h}_{task_n}))}{e} \\ t^{task_n - s_h}_{run} &= \frac{(\sum_{j=0}^{N} d^{task_{j+1}}_{task_j} + d^{s_h}_{task_n})}{v} \end{cases} \tag{9}$$

As Fig. 2 shows, $Phase$ *A-(1)* and $Phase$ *A-(2)* are recursively operating in the $PhaseA$ until to the end of T^c_i. Each time when current task subsequence finishes recharging task planning in $Phase$ *A-(2)*, C_i ,E_i are updated according to current planning result. At the same time, take the T^{s2}_i as T^s_i for next segmentation in $Phase$ *A-(1)*.

B. Multiple Recharging Tasks Coordination

Cause that $Phase\,A$ is decentralized and runs in each robot without considering influence of other robots, it may frequently happen that different robots wait for recharging at the same charging station in the same time, if task sequence are immediately executed after $Phase\,A$. So in the $Phase\,B$, all task sequences are centralized and some of these recharging tasks are coordinated based on bipartite graph matching to reduce robots unnecessary waiting time and improve efficiency. The process includes following four steps. Firstly,recharging station time window is defined and computed as follows for choosing the appropriate recharging tasks.

Definition Of Recharging Station Time Window: Similar to [11],recharging station time window is defined which aims to indicate occupied time interval of different recharging tasks including waiting time and recharging time. The set

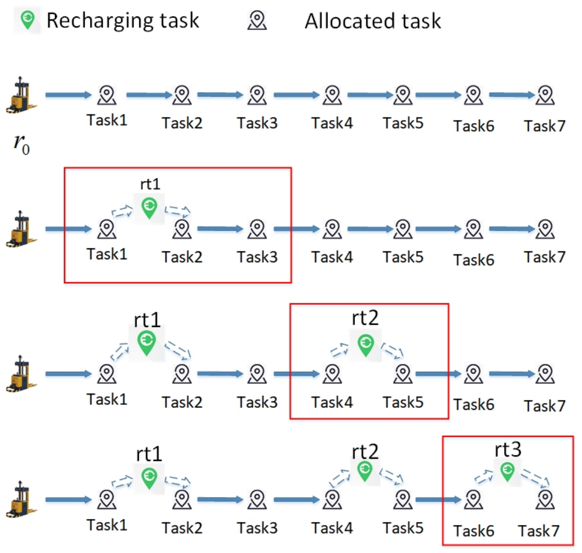

Fig. 2. The example of recharging task planning based on task sequence segmentation for robot r_0

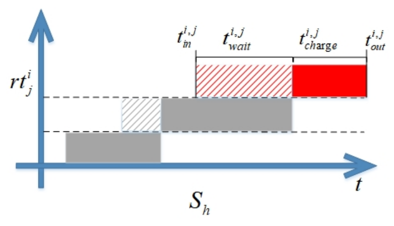

Fig. 3. Inserting time window in S_h for rt^i_j

of recharging station s_h time window is denote as TW_{s_h}. For recharging task rt^i_j which belongs to r_i, if its selected recharging station is s_h,the occupied time window is denoted as a vector $tw_{rt^i_j} = [t^{i,j}_{s_h,in}, t^{i,j}_{s_h,out}, t^{i,j}_{s_h,wait}, t^{i,j}_{s_h,charge}, s_h]$ contained in TW_{s_h}. The parameter $t^{i,j}_{s_h,in}$, $t^{i,j}_{s_h,out}$, $t^{i,j}_{s_h,wait}$ and $t^{i,j}_{s_h,charge}$ are defined as the arrival time, leaving time, waiting time interval and recharging time interval respectively in TW_{s_h}. Their relation is denoted as Eq(10).As Fig. 3 shows,there are four recharging tasks occupied time windows where texture rectangles represent their waiting time and solid rectangles represent recharging time.

$$t^{i,j}_{s_h,out} - t^{i,j}_{s_h,in} = t^{i,j}_{s_h,wait} + t^{i,j}_{s_h,charge} \tag{10}$$

Recharging tasks that need be coordinated are chosen by inserting time window. The selected recharging stations for these recharging tasks are reallocated based on maximum bipartite graph matching to improve overall recharging efficiency.The process includes following steps:

1) Insertion of time window for each recharging task: Time window insertion is to decide $tw_{rt^i_j}$ of each rt^i_j. And insertion order follows recharging tasks' order in the task sequence and of each robot r_i. As Fig. 3 shows, there is no overlap of $t^{i,j}_{charge}$ between different recharging tasks in S_h and that may cause some waiting time $t^{i,j}_{wait}$. The parameter in $tw_{rt^i_j}$ can be easily computed according to insertion order and electricity consumption model.The total waiting time of all recharging tasks is computed as Eq(11) and denoted as Sum_w when insertion is finished. This step is to make sure any recharging tasks occupied time interval and use this information to choose some recharging tasks that needed to coordinate.

$$Sum_w = \sum_{r_i \epsilon R} \sum_{rt^i_j \epsilon T^c_i} t^{i,j}_{wait} \tag{11}$$

2) Choose coordinated recharging tasks: Find the time interval with the most number of recharging tasks from TW_{s_h} in any s_h and set these recharging tasks as coordinated charging tasks.Delete the following recharging tasks time window whose order are behind the coordinated recharging tasks in the task sequences from TW_{s_h} .

3) Coordinate chosen recharging tasks: Multi robots waiting for recharging always comes from different recharging tasks selecting the same recharging stations.It is usually better to assign different recharging stations for these recharging tasks for solving this problem. Therefore, the issue in this step is to assign each coordinated recharging tasks to a different recharging station such that the accumulated time spent in recharging station is minimized. In graph theory, this problem is known as the matching problem.

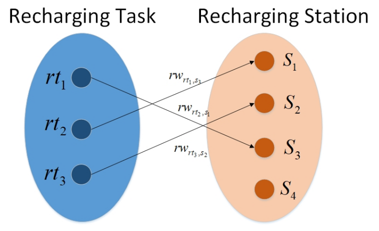

Fig. 4. The example of three charging tasks matching four charging stations

An example of a bipartite graph is shown in Fig. 4. In this figure there are 3 recharging tasks and 4 recharging stations which have to be assigned to the recharging tasks. An assignment between charging task and recharging station has a reward $rw^{rt^i_j}_{s_h}$.The objective is to assign all recharging tasks to recharging stations such that the accumulated rewards are maximized. It could be solved by the maximum bipartite matching algorithm.And $rw^{rt^i_j}_{s_h}$ can be computed in Eq(12) where parameter in are got from by the time window insertion similar to step1. In order to obtain a result that minimizes instead of maximizes the recharging time.

$$rw^{rt^i_j}_{s_h} = max_{\forall s_k \varepsilon S}(t^{i,j}_{s_k,out} - t^{i,j}_{s_k,in}) - (t^{i,j}_{s_h,out} - t^{i,j}_{s_h,in}) \tag{12}$$

The Eq(12) makes the maximum value of zero in $rw_{s_h}^{r_j^i}$ and the smaller values of positive value in $rw_{s_h}^{r_j^i}$. The positive values can now be seen as positive rewards.

It is possible that there are more recharging tasks than recharging stations. In that case, some of the recharging tasks are not assigned to a recharging stations when the maximum bipartite matching is performed. In order to guarantee that each recharging station will eventually be assigned to charging station, each time the maximum bipartite matching algorithm is executed, it is checked if there are unassigned recharging tasks. If this is the case, the previously assigned recharging tasks are removed from the algorithm, and the algorithm is executed again. This process is repeated until all recharging tasks are assigned to recharging stations.

When the process ends, the time window insertion is operated for the coordinated recharging tasks and the following recharging tasks whose order are behind the coordinated recharging tasks in each task sequence. And the total waiting time is recomputed and denoted as $Sum_w{}'$.

4) Assess the coordination convergence:

For coordinating more recharging tasks occupying the same time interval at the same recharging station and reducing more waiting time as much as possible, recharging tasks choosing and coordination in $Phase\, B$-*(2)* and $Phase\, B$-*(3)* is recursively operated with Sum_w updating until the convergence satisfies that the updated total waiting time $Sum_w{}'$ is not better than the former Sum_w.

IV. SIMULATION AND ANALYSIS

In order to evaluate the proposed method, simulations were implemented under the MATLAB environment with Intel 3.60GHz Core i7-4790U CPU and 8G RAM.The map of warehouse is depicted in Fig. 9 which includes the 20 recharging stations showed as red squares . The task execution efficiency, computing efficiency and recharging efficiency of proposed method are contrasted with that of the existing method in [1], [13] under a different number of tasks. The existing methods in [1],[13] only concentrate on solving when to recharge or where to recharge respectively.

For the warehouse environment in Fig. 5, the number of robots is 40. The total number of the tasks is set at 400, 500, 600, 700, 800 and the original task allocation is given by the method in [15]. The speed of each robot is 1m/s. For electricity consumption model in Eq(1), β is set at 0.5 and α is set at 0.1. The recharging rate for recharging station e is set to 1. For method in [13], the static threshold is set to the electricity consumption between in the simulated warehouse depicted in Fig. 5. The average of 20 repeated simulations were calculated.

A. Task execution efficiency

Fig. 6 shows the average task execution time of the three methods,and improvement percentage of proposed method compared with the existing methods in [1],[13]. As can be seen, the task execution time of the proposed method is less

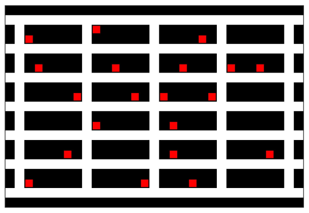

Fig. 5. Simulated warehouse environment.

than that of the existing method in [1], [13], And when task number becomes larger, the task execution efficiency improves a lot. And improvement percentage can be up to 20%.

The difference is mainly the result of improvement of proposed method in the recharging efficiency when task allocation is given. In the proposed method, when to recharge and where to recharge are considered simultaneously in the recharging task planning aiming to minimize the individual task execution time, whereas the existing methods in [1], [13] only focus on one of above two questions. In addition, seleted recharging stations of some planned recharging tasks are coordinated to reduce waiting time for recharging in the multiple recharging tasks phase, which also helps improve task efficiency for multi-robot, compared with the method in [1] ignoring the influence of multiple robots. And the improvement in task execution efficiency is more obvious when there are more tasks considering each robot needs more recharging tasks in that situation.

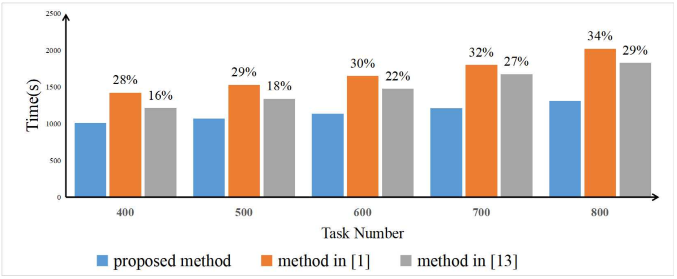

Fig. 6. Task execution time of three methods

B. Computation efficiency

Fig. 7 shows the average computing time of three methods, and improvement percentage of proposed method compared with the existing method [1], [13]. As can be seen, the computation time of the proposed method is less than that of the existing method in [1],[13]. And the proposed method is capable of computing the solutions in a short time, which satisfies the need of a multi-robot environment. When the task number becomes larger, the computing efficiency improves.

In the proposed method, unlike the existing methods in [1], [13] that search all possible recharging task order or schedule recharging tasks'selected recharging stations in the whole task sequence, the designed algorithm adopts the strategy of planning each single recharging task based on task sequence segmentation. Therefore, computation efficiency could be increased, compared to the existing methods in [1], [13].

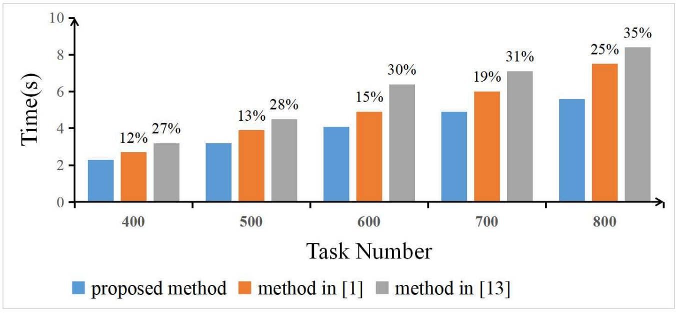

Fig. 7. Computation time of three methods

C. Recharging efficiency

In order to quantify the charging efficiency, three indexes are defined : average waiting time, average recharging time and average occupied time. Average waiting time and average recharging time represent the time spent on waiting and recharging in the recharging station for each robot respectively. Average occupied time represents time occupied by robots in recharging station. Fig. 8 shows the above three indices of three methods when task number is 800 and robot number is 40, and improvement percentage for proposed method compared with existing method in [1],[13]. As can be seen, proposed method obviously improves charging efficiency with less time spent on waiting and recharging in recharging station, compared with existing method in [8],[13].

The recharging performance of proposed method is guaranteed by planning order and selected recharging station for each recharging task simultaneously, and coordinating multiple recharging tasks from different robots. The former trys to solve when to recharge and where to recharge in the autonomous recharging problem at the same time whereas existing method in [1], [13] are confined to one of two above problems. The latter centralizes task sequences of all robots and reallocated multiple recharging tasks's selected recharging stations to shorten the overall waiting time for other robots recharging. This process considers recharging tasks planning results among different robots and would improve overall recharging efficiency.

V. CONCLUSION

This paper describes an efficient recharging task planning method, aiming to solve autonomous recharging problem of multiple robots in the warehouse environment. The method include two phases: In the first phase, recharging task is planning based on the task sequence segmentation for individual robot; in the second phase, all robots task sequences are centralized and some recharging tasks selected recharging station are coordinated based on maximum bipartite graph matching. Simulation results validate that our method can improve the overall task execution efficiency and charging efficiency with enough computational efficiency. Our future work will involve conducting experiments in a real warehouse to compare the performance of the proposed method with the existing method.

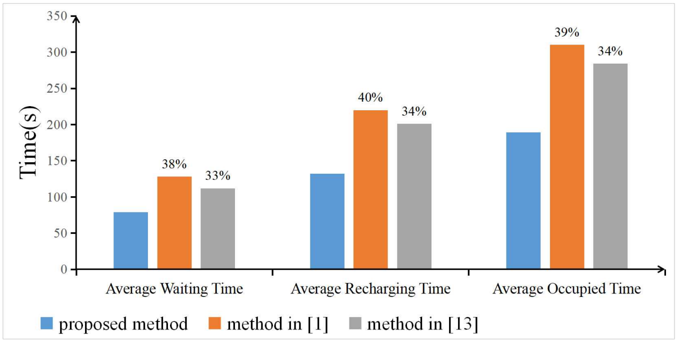

Fig. 8. Charging efficiency of three methods

REFERENCES

1 Kannan B , Marmol V , Bourne J , et al. The Autonomous Recharging Problem: Formulation and a market-based solution[C]// IEEE International Conference on Robotics Automation. IEEE, 2013.
2 Leonard J , Savvaris A , Tsourdos A . Energy Management in Swarm of Unmanned Aerial Vehicles[J]. Journal of Intelligent Robotic Systems, 2014, 74(1-2):233-250.
3 Silverman M C , Nies D , Jung B , et al. Staying Alive: A Docking Station for Autonomous Robot Recharging[C]// Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA. IEEE, 2002.
4 Rappaport M . Energy-Aware Mobile Robot Exploration with Adaptive Decision Thresholds[C]// Isr: International Symposium on Robotics. VDE, 2016.
5 Berenz V , Tanaka F , Suzuki K . Autonomous battery management for mobile robots based on risk and gain assessment[J]. Artificial Intelligence Review, 2012, 37(3):217-237.
6 Wawerla J , Vaughan R T . Near-Optimal Mobile Robot Recharging with the Rate-Maximizing Forager[J]. 2007.
7 F. Dressler, G. Fuchs, Energy-aware operation and task allocation of autonomous robots, Proc. IEEE Workshop Robot Motion Control (2005) 163168
8 B. Bethke, M. Valenti, J.P. How, An experimental demonstration with integrated health monitoring, IEEE Robot. Autom. Mag. (2008) 3944.
9 Kottas A , Drenner A , Papanikolopoulos N . Intelligent power management: Promoting power-consciousness in teams of mobile robots[C]// 2009 IEEE International Conference on Robotics and Automation. IEEE, 2009.
10 Wang I L , Wang Y , Lin P C . Optimal recharging strategies for electric vehicle fleets with duration constraints[J]. Transportation Research Part C: Emerging Technologies, 2016, 69:242-254
11 Siqueira, Fernando, Edson. (2016). A fuzzy approach to the autonomous recharging problem for mobile robots.1065-1070. 10.1109/FSKD.2016.7603326.
12 A Muoz-Melndez, F Semp, Drogoul A . Sharing a charging station without explicit communication in collective robotics[C]// International Conference on Simulation of Adaptive Behavior on from Animals to Animats. MIT Press, 2002.
13 Kim J , Song B D , Morrison J R . On the Scheduling of Systems of UAVs and Fuel Service Stations for Long-Term Mission Fulfillment[J]. Journal of Intelligent Robotic Systems, 2013, 70(1-4):347-359.
14 Ruochen T,Jingchuan W,Tian W,Weidong C.A Time-Efficient Approach to Solve Conflicts and Deadlocks for Scheduling AGVs in Warehousing Applications[C]// 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE, 2019.
15 Zhao W , Meng Q , Chung P W H . A Heuristic Distributed Task Allocation Method for Multivehicle Multitask Problems and Its Application to Search and Rescue Scenario[J]. IEEE Transactions on Cybernetics, 2015:1-1.