Mobile Nodes Scheduling in MANETs for Resource Assignment: from Hospital Assignment to Energy Charging

Peng Liu*, Biao Xu*, Zhen Jiang†‡, and Jie Wu‡
*Institute of Computer Application Technology, Hangzhou Dianzi University, China
†Dept. of Computer Science, West Chester University, USA
‡Dept. of Computer and Information Sciences, Temple University, USA

Abstract—**In the large scale MANETs, mobile nodes could act either as resource providers, relays or requestors in the network. Since the storage space, the channel capacity and the energy supply are often limited, it is always worth considering the assignment between requestors and resources. We model two scenarios in this chapter, namely hospital assignment and energy charging. In the first scenario, the mobile node carries the patient to a hospital. Patients with different condition may compete for the same hospital. In the battery powered MANETs, mobile nodes need constantly charging to remain active. The problem lies in that energy critical nodes may not find an available charger in short distance and a temporary stationary node being assigned to a mobile charger.**

In both case, our goal is to optimize QoS (in terms of latency) in assigning requestors to resources. We extend the scheme to consider resource reservation along the time scale by estimating from the past records in history. As a result, the occupancy is balanced in order to reduce the risk of critical nodes being delayed. Then we develop an local waiting queue to keep serious nodes waiting locally, when it costs more to reach another available resource. Simulation results show the substantial improvement of our approach in average delay and number of failure-of-assignment.

Index Terms—**Ambulance service, resource allocation, bipartite matching, wireless ad-hoc network.**

I. Introduction

Indicated by [1], pre-consultation is one of the most important facts of delay in emergency treatment in addition to transportation time of ambulances. After that, an emergent patient would be assigned a sickbed and a doctor for next step treatment. In China and India [2][3], when there are not enough sickbeds, some life-critical patients would die, which also leads to patient-doctor dispute [4]. Traditional local greedy patient-hospital assignment methods, which assign a patient to the available hospital with shortest delay in a first-come-first-serve (FCFS) manner. The problem is, a life-critical patient may encounter non-vacancy of sickbeds in nearby hospitals, which have been occupied by non-critical patients, forcing an ambulance choose a further hospital and causing deadly delay [5]. This is either because non-critical patients have smaller delay, or due to their earlier appearances before life-critical patients. In an outburst of emergent patients [6], the amount could be far more beyond the hospital accommodation. Therefore, we must balance the average delay of each patient and reserve some space for life-critical patients in advance. Also, the chargers in the Mobile Ad-hoc Networks have limited resources. Each one of them can only support a fixed number of mobile nodes. When there are multiple requests exist in the system, careless assignment such as greedy based method would bring extra delay and void assignment which is severe to energy critical nodes. To remain the network working in a green and lasting manner, the assignment between the nodes and chargers must be carefully considered.

In this paper, we first propose a hospital assignment for emergent patients in a big city, denoted by HAEP, to minimize the average delay for sending a patient to a hospital. Then apply the algorithm to the scenario of energy charing in MANETs. An ambulance can obtain a reserved sickbed in a certain hospital within limited distance without continuously waiting for pre-consultation. In this way, many life critical situation can be treated in the limited time. We take advantage of the recent technical advances in wireless networks and vehicular networks [7][8] to collect the real time information from hospitals and patients, and to solve the above optimization problem in a way that is derived from the Hungarian algorithm [9]. In our system, there are three kinds of patients, namely, life-critical, serious and cared patients. Two kinds of hospitals, i.e., premium and primitive hospitals are considered, where primitive hospitals can only treat non-critical patients. Furthermore, the occupied bed is not preemptable regardless life-critical or not, due to patient-doctor dispute. To avoid the situation of lack-of-bed, a number of preserved beds are set aside for life-critical patients, who are diagnosed by the ambulance. In our paper, the preserved beds are calculated based on history records. We propose our solution on the extension of Maximum Weight Bipartite Matching [10], a problem to assignment n resource to n resource requestors. We assign n patients to m hospitals with capacity C at each hospital, and provide the preserving sickbeds and waiting queue along the time scale. Our contribution is threefold:

1) We propose a new assignment to balance the bed requirement of life-critical and non-critical patients with

the purpose to reduce the delay of treatment for life-critical patient as well as the average delay for all patients.

2) We extend the solution from Hungarian algorithm, with the capacity and consideration of reservation along the time scale, then the application on cases of inadequate resources.
3) We simulate a real scenario of patient assignment in Shanghai [11]. The experimental results derived from real trace data show our substantial improvement in delay and failure-of-assignment in terms of their impact on efficient treatment on life-critical patients.
4) We further apply the scheme to an energy charging problem in MANETs where average cost for searching chargers are minimized.

II. Target problem and related works

In this section, we discuss the problem in existing system in the ambulance services. Greedy algorithms are often used in resource allocation problems. When the constraints determine a polymathic and the objective is linear, the greedy procedure results in an optimal solution [12]. However, most current research work ignores the fact that all the requests take place along the time scale which is not an one-time optimization problem or resource assignment. M. Xu studied another interesting phenomenon that service delay for non-emergent patients will be significantly affected due to arrival of emergent patients [13]. They conducted a retrospective study in real trace of a large hospital in Hong Kong and estimated waiting time and length of stay for individual non-emergent patients as a function of the presence of emergent patients and other related factors. The study convinces us that the competition of critical and non-critical patients must be carefully addressed.

As demonstrated in Fig. 1, there are two hospitals, in which a is premium hospital and b is primitive hospital. There are also two patients in which 1 is a non-critical patient and 2 is a life-critical patient. We use *number:time* in the patient ellipse to show patient number and his appearance time. The weight in the arrow denotes the delay. Assume available capacity in each hospital is 1, requests are submitted from patient 1 and 2 at time 0 as seen in Fig.1(a). They will compete for a, according to the FCFS greedy algorithm, 1 will be assigned to a since it has smallest delay, leaving 2 no place to go. As the resource is enough for both patients, the best solution is to assign 1 to b, and 2 to a. In Fig.1(b), when two patients raise requests one after another, FCFS results 1 occuping a although delay of 1 is greater than that of 2, and 2 has no hospital to go. The desired assignment is indicated in Fig.1(b).

We assume hospitals have fixed locations and patients will appear anywhere to call ambulance so that localizing resources [14][15] can not be used here to solve the problem. Through route planning or traffic control, we assume there's no extra delay in taking patient to hospitals [16][17][18], so that the delay can be calculated based on the distance and pre-consultation for any hospitals. Also, the resource allocation problem inside the hospitals are addressed in many literature, such as operation room planning [19], admission arrangement [20], improving use of Computed Tomography Facilities [21], and surgery scheduling [22], which enables us to study historic records and model capability of available beds. We also consider the ability of different hospitals to treat different kind of patients. Hospitals can be categorized into different classes according to their ability of medical care. Therefore, the problem becomes the one with multiple kinds of resources (each has different capacity) and multiple kinds of requesters, which includes the consideration of future reservation along the time scale.

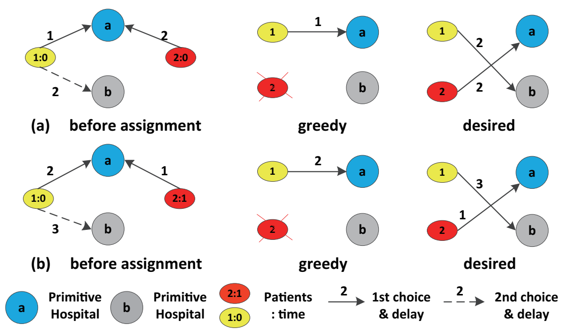


Fig. 1. A demonstration of patients and hospitals assignment

In this paper, we solve the optimized resource allocation problem using bipartite matching in [9], first to balance the requirements among critical and non-critical patients in terms of average delay, then to develop a preservation-based method on top of the bipartite matching along the time scale for life-critical patients, finally to introduce local waiting queue for serious patients. The number of available beds are estimated using possibility observation and history statistics. The waiting queue is build-up by leveraging local waiting time against re-assignment cost. The proposed method will greatly balance the need of both life-critical and non-critical patients along the time scale.

III. System Model

In our system, Patients, Hospitals and the Service Center are three main components. Ambulance can send the request to the Service Center via wireless Ad Hoc networks or cellular networks. Hospitals count vacancy and estimate the capacity growth with the advanced technology of [7][8]. All the ambulances are equipped with on-vehicle communication devices and can be guided by the Service Center to transport patients to the target hospital.

Tab.I summarizes all of the notations used in this paper, which will be explained in the following. There are three kinds of patients: life-critical, serious, and cared. Denote $x_i \in C_x \subseteq X$, as the ith critical patient, $x_j \in S_x \subseteq X$, as jth serious patient, and $x_k \in N_x \subseteq X$, as kth cared patients. There are also two kinds of hospitals, i.e., H_i^P means ith premium hospitals and H_j^p means jth primitive hospitals. As we consider the time scale, the in and out of patients from hospital y could be monitored, managed, and predicted, thus capacity of sickbeds at $t > 0$, denoted as C_y^t,

TABLE I
NOTATIONS

X	patient set $X = C_x \cup S_x \cup N_x = \{1, 2, 3, \cdots\}$
$\mid X \mid$	total number of patients ($\in X$) in schedule
Y	hospitals $Y = H^P \cup H^p = \{a, b, c, \cdots\}$
$\mid Y \mid$	number of in-patient beds
C_y^X	available beds (also called beds capacity) of $y \in Y$ where $X \in \{C_x, S_x, N_x\}$
$R(x, y)$	cost (≤ 0) for $x \in X$ to reach $y \in Y$ in terms of elapsed time where "$-$" indicates an initial/unreachable status
$m(x, y)$	bipartite matching between $x \in X$ and $y \in Y$ where 1 denotes a saturated assignment, 0 denotes a possible assignment, and "$-$" indicates not reachable currently
$L(v)$	labeling function of Hungarian algorithm [9], $v \in X \cup Y$
$L'(v)$	previous record of L for any given $v \in X \cup Y$
α	the difference between $L(v)$ and $L'(v)$ each time
S	patient set in the current consideration of allocation, $\subseteq X$
$N(S)$	hospitals ($\subseteq Y$) that are reachable by patients $\in S$, or arriving), i.e., $\{j \mid \exists m(i, j) = 0 \text{ or } 1\}$
ⓐ	common beds available at a hospital, that could be assigned to all types of patients, say $y \subseteq Y$, that have bed(s), denoted by ⓐy, i.e., $\{y \mid \textcircled{a}y = C_y - \sum_{x \in X} m(x, y)\}$
Ⓡ	Availability of persevered beds for critical patients at a hospital, e.g., Ⓡ$y = R_i$
ⓦ	Availability of waiting queue for serious patients at a hospital, e.g., ⓦ$y = w_k$
E_u^*	an alternating tree [9] derived from m, with the root u, simply called E-tree

is calculated based on the prediction of patient-leaving amount and allocation at round t. Especially, when $t = 0$, $C_y^0 = C_y$. In our method, we also predict the amount of critical patients in the future time slice. Since critical patients can appear for many reasons, e.g., a burst of a serious disease, a severe traffic accident, etc, although there are some mathematic tools [23][24] to estimate one kind of situation, it is hard to estimate superimposed situations. However, there are two key facts that we could use. First is that there will be regular days and peak days regarding the burst rate of critical patients due to epidemic seasons, holidays, or bad weather. Second, no matter the accident or disease, there is always a trend rather than sporadic ups and downs. Therefore, we can use the last three data to predict the future possibility. Our method can be described in two steps, i.e., cost matrix build-up, and hospital assignment.

The goal of the paper is to assign each proposed patient $\in X, X = C_x \cup S_x \cup N_x$ to a proper hospital $\in H, H = H^P \cup H^p$, so that the total delay (from the time a patient is picked up by an ambulance, waiting at hospitals, until he gets treatment in a hospital). By giving a critical patient more weight in delay cost, our method can balance the requirements between life-critical and non-critical patients. The impact to patient treatment is also measured by the failure-of-assignment of critical patients. Existing research utilizing bipartite matching often formalizes problems as maximum-weighted matching [10], while this problem is a minimum-weighted matching. By setting the delay cost table in [9] to its contrariety, such as $R(x, y) = -cost(x, y)$, our minimal delay problem could be implemented as max-weight matching. Inherited from the bipartite matching, we use matrix m to indicate whether there is an assignment from patient x to hospital y, e.g., $m(x, y) = 0, 1$. Then we formalize the problem as follows:

$$\begin{aligned} & \max \textstyle\sum_{x \in X} \sum_{y \in Y} \sum_{0 \leq t < \infty} R(x, y) m(x, y) & \\ s.t. \quad & \text{every } m(x, y) = 0,\ 1, \text{ or "}-\text{"} & i) \\ & \textstyle\sum_{y \in Y} m(x, y) = 1 \text{ for every } x \in X & ii) \\ & \textstyle\sum_{x \in X} m^t(x, y) \leq C_y^t \text{ for every } y \in Y \text{ and any } t & iii) \end{aligned}$$

"$-$" indicates an initial or unreachable status. Constraint i) ensures the sickbed assignment as a bipartite matching. Constraint ii) guarantees such an assignment without double-assignment. Constraint iii) asserts the use of resources under the capacity constraint. The problem cannot be solved by minimizing average delay in each patient level, but to minimize average delay for all patients, especially between non-critical and serious.

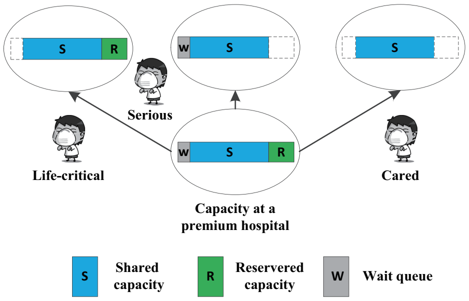


Fig. 2. Capacity illustration

When a hospital could have multiple capacities and more than one type of patients, the problem becomes complicated, since the current optimal does not mean global optimal. As shown in Fig. 2, different patients have different view of capacity of a premium hospital. In our method, we first consider the extension of capacity to bipartite matching, then the optimization across adjacent time periods. Based on the estimated rate of critical patients that will enter the hospital in the near future, we set up preservation in our algorithm HAEP (indicated as R in Fig. 2) to optimize allocation across time periods. Furthermore, for patients in serious condition, we arrange a special waiting room to have them stay rather than ask them to go (indicated as w in Fig. 2, capacity is based on the number of patients leaving the hospital and time cost to another nearest hospital).

IV. Method to solve multi-dimension hospital assignment

A. Cost matrix build-up

The basic requirement of bipartite matching is the build-up of cost matrix. In our scenario, the main cost is the response time from the ambulance submitting a request until the patient successfully checks in at a hospital. It is composed of two parts, one is the transportation time from where the patient is to the location of the destination hospital. The other is the waiting time at the destination hospital, which is necessary. To simplify

TABLE II
A SIMPLE EXAMPLE OF PATIENTS REQUESTS

patient	requested assignment time	situation	if no vacancy
U1	0	Cared	no wait
U2	0	Cared	no wait
U3	1	Serious	wait
U4	1	life-Critical	no wait

the model, we define Euclidean distance between them as the metric of transportation time. In practice, the transportation time can be controlled [16][17][18] so that this will not affect the proposed method.

We show an example in Shanghai as indicated in Fig. 3 and and table II. To quickly simplify the algorithm, we could use grid to achieve the same goal. We first divide the whole region into small grids. The size of the grids are closely related to the organization of the city, such as by living area, by postcode and so on. With grid, we can easily find out the hospital sequence for any grid according to distance.

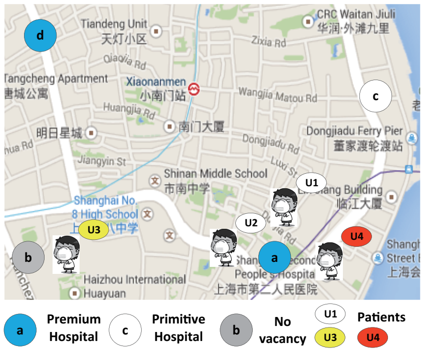


Fig. 3. A demonstration of patients and hospitals assignment

After grid participation from Fig.3, we get Fig.4. Then we can easily get the cost matrix as seen in the equation. As indicated here, for each time slot we can get one cost matrix where "-" indicates an impossible reach from patient to hospital. The initial capacity (C_a, C_b, C_c, C_d) at each hospital is $(1, 0, 2, 1)$.

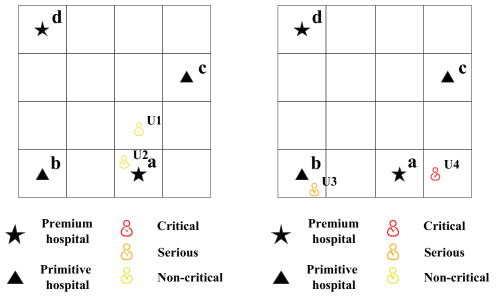


Fig. 4. Example illustration

$$CostMatrix(slot0) = \begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} U1 \\ U2 \end{array} & \begin{pmatrix} 1 & 3 & 2 & 4 \\ 0 & 2 & 3 & 5 \end{pmatrix} \end{array} \quad (1)$$

$$CostMatrix(slot1) = \begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} U3 \\ U4 \end{array} & \begin{pmatrix} 2 & 0 & 5 & 3 \\ 1 & - & - & 6 \end{pmatrix} \end{array} \quad (2)$$

For FCFS schedule, at slot 0, patient $U1$ will be allocated to hospital a, patient $U2$ will be allocated to hospital c. At slot 1, the capacity of each hospital (C_a, C_b, C_c, C_d) changes from $(1, 0, 2, 1)$ to $(0, 0, 1, 1)$. Therefore, conflict appears in which two emergent patients cannot be assigned to close hospitals. More terribly, patient $U4$ will have no hospital in which to stay, since the last eligible hospital d is assigned to patient $U3$ since he is earlier than patient $U4$.

B. Hospital assignment

For regular matching algorithm, at each time slot, there is an augment path found so that $U1$ and $U2$, $U3$ and $U4$ will switch assignments to get an optimization. Finally, after the switch, patient $U4$ could find a hospital with time cost 6. It is local optimal; however, in the view of slot 0 and slot 1 together, the assignment is not optimal.

In our algorithm, assume that there is a reservation of 1 bed for critical patients at hospital a at slot 0. Therefore, the capacity of (C_a, C_b, C_c, C_d) that the two patients see is $(0, 0, 2, 1)$. After applying a matching algorithm, the result of slot 0 is $U1$ to c and $U2$ to c.

At slot 1, consider the patient leaving rate at hospital b: it allows a waiting queue of 1 room, therefore, the capacity of (C_a, C_b, C_c, C_d) that the serious and critical patient see is $(0, 1, 0, 0)$ and $(1, -, -, 0)$, so the matching would be as shown in Fig.5. The total cost of our algorithm is $6 + waitingtime(lessthan1)$, while that of the regular matching is 13, and FCFS has no answer.

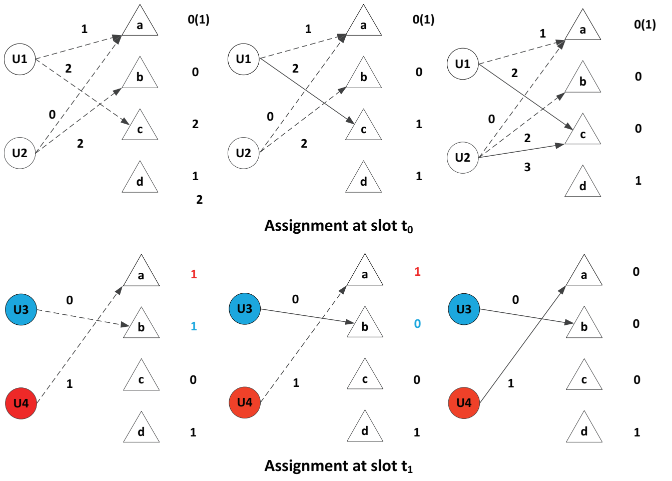


Fig. 5. Our matching at slot t_0 and t_1

Non-critical patients can switch with critical patients, but with limitations. In the experiment, we set the cost of life-critical patient three times as the same distance and non-critical patient. The value could be adjusted when applying the framework to practice. The assignment problem is actually a specific capacity to a different requester. As seen in Fig.2, each category of patients can use a different portion of the entire capacity of a hospital, which is designed to meet the requirement of each patient.

Reserved capacity is an evolutional parameter, which can only accommodate critical patients since there are less applicable hospitals, and critical patients obviously cannot wait. Over-prediction will cause waste, and less than enough will cause inefficiency. Wait-allowed capacity is a small list which is related to the hospital-leaving rate. For serious patients, the earlier the treatment, the better; therefore, it is good to have them have some basic treatment and to wait in a nearby hospital when it will cost additional time to reach a further hospital. The capacity of a queue in a waiting room is based on the number of patients leaving the hospital and the time cost to another nearest hospital. A non-critical patient is not allowed to wait, and will always be sent to the nearest hospital with capacity.

The schedule is expected to apply whenever there is a new request. However, to avoid local optimization, the method needs more requests together to perform the schedule. A time slice δ is adopted with trade-off of between waiting time and global cost. If the δ is too large, it will incur additional waiting time. If it is too small, the local optimization will incur global cost. We have the following definition:

Definition 1: Any $x \in X$ that has not seized the reservation is called unsaturated, and it has $m(x,y) \neq 1$ for every $y \in Y$. Any $y \in Y$ still available for allocation is called unsaturated and it has $\sum_{x \in X} m(x,y) < C_y$.

Definition 2: Any $y \in Y$ still available for allocation is called available and $\textcircled{a}y$ or $\textcircled{r}y$ or $\textcircled{w}y > 0$ according to category of $v \in S$ where $m(v,y) = 0$).

Our algorithm is shown as Alg.1. The first phase is to initialize cost matrix R, matching matrix m, and label L(v). The second phase is to check if the matching could stop and converge otherwise to build a new augmenting tree with root x. The remaining phases are similar to the Hungarian algorithm, except we use Def. 2 to implement multi-dimensional matching, and we alter the table construction phase to apply on multiple capacities.

For critical patients, first use reserved capacity. For any $C_y > 1$, multiple reservations are allowed on hospital y. Any existing reservation will be added into our records (E-tree and S), for later reservation shuffle with the augment path. This phase of hospital matching will continue until all patients have been checked under the capacity constraint. For serious and cared patients, they can only see the capacity at premium hospitals after removing critical preservation. For serious patients, they will be allowed to stay at a waiting list before getting a bed in a hospital which is denoted as $\textcircled{w}y$. All the capacity checks imply the above strategy. The entire process will stop at phase 2 when every patient finds its target hospital or all the beds are allocated (the method will stop when the sum of cost of allocated patients and un-allocated patients is minimal). Eq. 3 is used to calculate α for the greedy progress along the time scale of patient arrivals.

$$\alpha = \min\nolimits_{x \in S, y \in Y \setminus T} \{L(x) + L(y) - R(x,y)\} \tag{3}$$

$$L(v) = \begin{cases} L'(v) - \alpha, & \text{if } v \in S \\ L'(v) + \alpha, & \text{if } v \in Y \text{ has been considered} \\ & \text{before for } S\text{, i.e., } \{y \in Y \mid \\ & \exists m(x,y) = 1 \text{ where } x \in X\} \\ L'(v), & \text{otherwise} \end{cases} \tag{4}$$

$$m(x,y) = \begin{cases} 0 & L(x) + L(y) = R(x,y) \\ \text{“}-\text{”} & otherwise \end{cases} \tag{5}$$

Algorithm 1 Hospital assignment based on Hungarian algorithm with capacity and reservation

```
Require: $X$, $Y$, $R$, and $C_y$ for each $y \in Y$
Ensure: a bipartite matching in $Y$ for each $x \in X$
1: Initialization (i.e., R(i,j), m(i,j), and L(v) with Eq.(3)).
2: Completion check:
   For any unsaturated (Def. 1) patient $x \in X$, prepare $S$, $T$ and $E^*_x$ to start the matching process in the following; otherwise, successfully end the entire process.
3: Label update for new opportunities for $x$ to match with $y$ in $m$:
   IF $N(S) \neq T$ GOTO phase 4 ELSE
   Update reaching opportunities of patients by (first $\alpha$ with Eq.(3), and then $L$ with Eq.(4), in order to reset $m$ with Eq.(5).
4: Table construction ($E^*_x, m, T and\ S$):
   Find any $y \in N(S) \setminus T$; IF $y$ is available (Def. 2), an alternating path from $x$ (root of the tree $E^*_x$) to $y$ exists, apply augment matching along this path as the Hungarian algorithm, decrease capacity of $y$ by 1, GOTO phase 2.
   ELSE find all $z$ that are matched with $y$, update $E^*_x$ with $(z,y)$ respectively, $S = S \cup \{z\}$, $T = T \cup \{y\}$, GOTO phase 3.
```

When there are more patients than available resources, the original Hungarian algorithm cannot give a solution since the result is obviously not a perfect matching. When all the resources (say y) are allocated, there are still requesters that remain unsaturated (say x). Therefore, in our algorithm, we add a virtual resource hospital beyond the real ones. By setting the cost to be a very big number for each patient to reach along with infinite capacities, our algorithm will converge and assure an optimization for all allocated y. As indicated in equation (6), there are two hospitals and two patients; we add virtual hospital c and a cost of 99 time units for each patient. Assume the initial availability at each hospital a and b is (1, 0) without virtual hospital, the matching process would be as shown in the Fig.6. Without virtual hospital c, the final assignment from the extended Hungarian algorithm would be as left with a cost of 2 ($U1$ to a), while our algorithm will stop at right with a cost of 1 ($U2$ to a).

$$CostMatrix = \begin{array}{c} \\ U1 \\ U2 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 2 & 3 & 99 \\ 1 & 2 & 99 \end{pmatrix} \end{array} \tag{6}$$

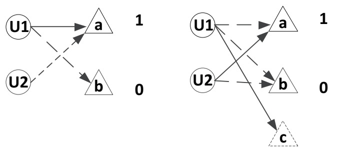


Fig. 6. illustration of virtual hospital

Theorem 1. *The bipartite matching achieved with Alg. 1 is optimal on total transportation time in R when each R_k is accurate.*

Proof: Alg.1 is derived from the Hungarian algorithm [10]; first, with multiple capacity, we can still guarantee optimality since in phase 4 every possible available capacity is considered and the multiple z assigned with y will be added to the augmenting tree properly. According to Def.2, if the estimation of R_k is accurate, the total cost will be minimized. As we add virtual node, the algorithm will finally stop at phase 2, no matter whether the resource is adequate or not. Therefore, we have this statement proven. ■

C. Parameter formulation

The waiting capacity of a hospital, i.e., w_k, is very important to serious patients, since they could make sure they would have beds after waiting, and get some basic treatment while waiting. It takes extra traffic time (may be greater than the delay cost of waiting) to get to the next hospital. The parameter w_k is related to the hospital-leaving rate and the time to the furthest hospital. We adopt poisson distribution [19], and find λ using the statistics. Since the average in-patient hours d_k is in a Poisson distribution [14] as $P(k, \lambda) = \frac{\lambda^k e^{-k}}{k!}$ and normally most hospitals are running in a high occupancy, we could get w_k as below:

$$\sum_{i=w}^{\infty} P(i, \frac{C_k}{d_k} \times Min\{T_k\}) > 0.8 \tag{7}$$

Where $Min\{T_k\}$ denotes the distance (represented by time) of nearest hospital to hospital y_k. C_k and d_k denote the capacity and average in-patient time of y_k respectively. Namely, we hope the total possibility of leaving w_k patients in $Min\{T_k\}$ time is larger than 80%. The larger the $Min\{T_k\}$ is, the larger the w_k. It is normal that patients would want to wait at the closer hospital, as opposed to going to the further one.

The reservation number of beds R_k at hospital y_k would greatly affect the performance of the framework. Cared patients can only use rest capacity excluding R, and critical patients will first use R then the rest of capacity. In our paper, we suggest a method that calculates R_k in a less complex way. Since the reservation depends on the distribution of in-patient possibility, a concrete method could be adopted according to a different realistic model. Here, we assume that the number of critical patients in a region is always smaller than the capacity of a premium hospital.

Most diseases have their unique distribution and disciplinarian. For any single illness, ARIMA [23] can be used to estimate the happen rate, or Markov process [24]. However, the in-patient requirements are also from incidents such as a car crash, fire, alcohol poisoning, etc. These incidents vary, but will remain stable in a period of time. Illness and incidents, altogether, will make an orderliness along the time scale, but will be smooth in a divided period of time. The goal is to try to match the reservation beds with the critical situations along a month scale.

The idea is simple, if we observe a waste of additional reservation beds, we cut the budget. If we continually encounter insufficient reservations, we increase the budget. The method is not as accurate as the ARIMA or Markov process, but it requires less training and history data, as well as a great reduction of computational complexity, while the accuracy is very acceptable. We record the last three data sets as history to estimate the future. Accuracy could be improved with a higher sample rate, and more history data. For example, let r_i denote the ith amount of in-patient critical patients, then the estimation of next time-slot's reservation could be calculated as

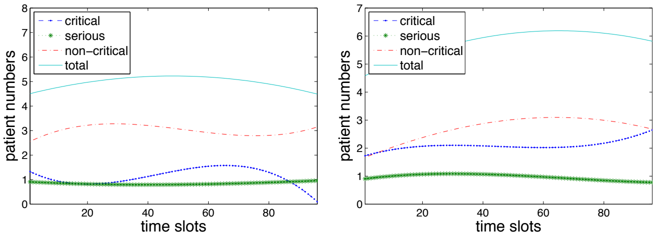
$$R_{i+1} = r_i + \frac{\alpha \times (r_i - r_{i-1}) + \beta \times (r_{i-1} - r_{i-2})}{2} + \gamma \tag{8}$$

Here, α, β, and γ are all a constant coefficient where $\alpha + \beta = 1$. α and β are the weight of the history data, where they are set to 0.6 and 0.4 in our experiment. γ is a compensatory factor, which is set to 0 in our experiment.

V. Experimental evaluation and Scenario overview

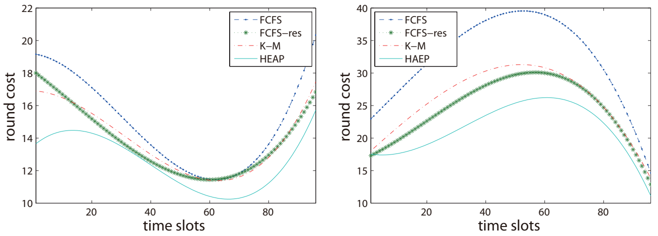
We evaluate our algorithm using simulations, but the data are derived from real statistics [20]. Here is the simulation parameter setup: a 4×4 grid (adjacent grid's distance is 5 minutes) with 4 hospitals (2 premium, 2 primitive), each having 120 beds. The average in-hospital time is 24 hours (shrink pro rata according to 10 days, on average, in practice [11]). We use 15 minutes as a schedule time slot. We use a Poisson distribution function to generate x patients including critical, serious, and non-critical every 15 minutes, and distribute them to 16 grids randomly. Generate data set, record the cost matrix. The patient number along the time scale is shown in Fig.7. Here, we use unit number rather than real number of patients. Therefore, one unit of patients could be hundreds of patients in a real scenario.

The first competitor is a local heuristic method denoted by $FCFS$ [12]. Whenever there is a new request, it will be assigned to the nearest appropriate hospital, i.e., a critical patient will be sent to a nearest premium hospital, while a non-critical patient will be sent to any nearest hospital with vacancy. The second competitor is a basic bipartite matching method [10] denoted by $K-M$ which divides time into slots, and uses bipartite matching to achieve global optimization. The third method is $FCFS$ with preservation, denoted as



(a) patient number of regular days (b) patient number of peak days

Fig. 7. patient number varies along time



(a) round cost of regular days (b) round cost of peak days

Fig. 8. cost for each round

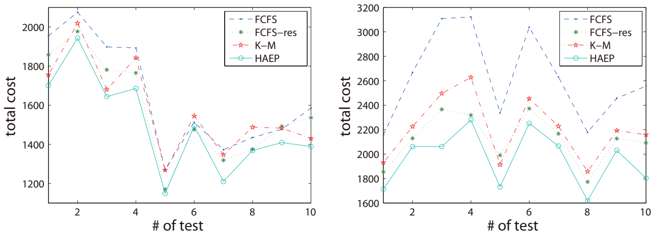
$FCFS-res$. The fourth method is our method which uses a critical safe reservation on top of $K-M$, denoted by $HAEP$. Apply $FCFS$, $K-M$ and our algorithm $HAEP$ onto the same data set. Run simulations for 24 hours, and repeat 10 times. If any of the algorithm could not allocate a critical patient, increase failure-of-assignment by 1 and add the maximum cost of the grid. Every 15 minutes, generate patients and try to assign them in 4 hospitals. To lever the different emergency situations, we set the coefficients as 1:1:3 and 2:1:3 for critical, serious, non-critical patients, respectively, as regular and peak days.

Fig.7 shows the dynamic of patient number in each round. The difference is that the number of critical patients rises one time in Fig.7(b) against Fig.7(a). All the dynamics abide by the λ of the Poisson distribution.

From Fig.8, we could compare the round waiting time for 4 methods. The costs vary since there are different patient requests in each round. $FCFS$ will cost more than $K-M$, $FCFS-res$ and $HAEP$. $K-M$ will be better than $FCFS$, but not as good as $FCFS-res$ and $HAEP$. They will give more benefit to serious and critical patients so that some non-critical patients will be affected. $HAEP$ is obviously the best.

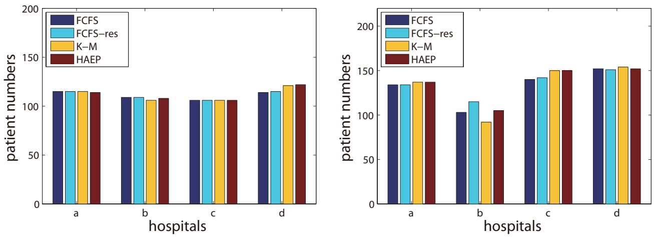
We repeat the test 10 times so that we get more visual facts. The total cost of four methods, can be seen in Fig.9. $HAEP$, denoted by a light blue line is about 25% less than $K-M$, which implies that we both have a good total cost, and better cares for critical patients. $FCFS-res$ also has preservation, so the total cost is a little bit lower than $K-M$. The more the critical patients, the better the performance of our method $HAEP$.

As a distribution of patients in each hospital, shown in Fig.10, $FCFS$ has nearly average distribution due to the fact that it is only based on distance. $FCFS-res$ and



(a) total cost of regular days (b) total cost of peak days

Fig. 9. total cost in different number of test



(a) distribution of regular days (b) distribution of peak days

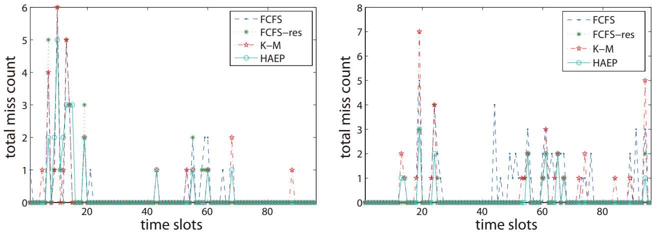
Fig. 10. distribution of patients to hospitals

$HAEP$ consider the requirements from critical patients so that premium hospital a and d gets more patients in. $HAEP$ uses the reservation so that some noncritical patients will go further, and primitive hospital will get more patients. However, the distribution also relies on the patients' distribution in grids.

When there are not enough resource (beds), there are always some patients that cannot be assigned to a hospital. We compare the total mismatch times of four algorithms, as shown in Fig.11. As for $HAEP$, critical patients are the first to consider; therefore, some non-critical patients maybe become mismatched. That is why the improvement is not very obvious. However, if we look into the details, $HAEP$ will have more critical patients scheduled than will the other two methods.

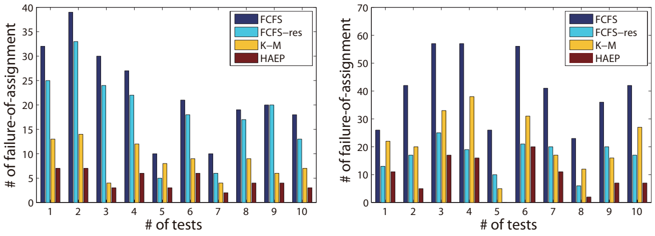
We also consider the failure-of-assignment of critical and serious patients. This is because the incoming of patients is based on the possibility that there could be times that the total resource cannot meet the requirement. Under this situation, we compare the total number of failure-of-assignment of four methods as shown in Fig.12. $HAEP$ definitely gets the minimum failure-of-assignment.

We also study the impact on our method using different



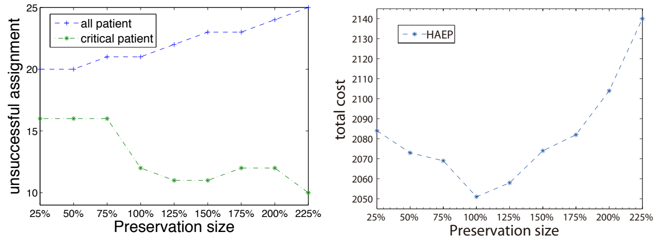
(a) distribution of regular days (b) distribution of peak days

Fig. 11. distribution of patients to hospitals



(a) distribution of regular days (b) distribution of peak days

Fig. 12. distribution of patients to hospitals



(a) distribution of regular days (b) distribution of peak days

Fig. 13. evaluation of different preservation size

reservation percentages. From Fig.13, we illustrate the unsuccessful assuagement and total cost change according to preservation adjustment. The experiment is based on one extremely completive test instance of 96 rounds. We name our choice as 100%, and decrease/increase accordingly. The unsuccessful assignments of critical patients decrease with the increasing of preservation beds, but the unsuccessful assignments of all patients are increasing in Fig.13(a). Meanwhile, the total cost increases very fast when the preservation is added beyond our value, as shown in Fig.13(b). It is obviously a tradeoff between total cost and benefit of critical patients to find best point of preservation size.

Our observations are summarized as follows: 1) From Fig.9, our results show that the HAEP always outperforms FCFS, FCFS-res and K-M, no matter in regular days or peak days. During regular days, the critical patients are not many, so that the competition is not serious. FCFS has the worst performance while FCFS-res and K-M are quite similar. During peak days, consider the ratio: HAEP is 30% better than FCFS and 16% better than K-M, and 10% better than FCFS-res. FCFS-res is rated as the second best which means critical patients have been well taken care of, and comprise a good portion of the total performance. 2) From Fig.12, regarding the deathrate, our algorithm HAEP has a very low failure-of-assignment over the other three algorithms. The FCFS is the worst algorithm, since it does not consider the future situation of critical patients at all. FCFS-res considers the requirement of critical patients, so it has a good performance as well. 3) Fig.10 shows the distribution of assigned patients. Since only successfully assigned patients will be calculated, the figure shows that HAEP has the best utilization of hospital beds, while FCFS has the worst. K-M comes as the second, since it has local optimal for each round.

VI. Charger assignment in MANETs

In Mobile Ad-hoc Networks, most mobile nodes such as s-mart phones, sensor nodes and UAVs are powered by batteries. To ensure their proper working conditions, constant charging is necessary. Wireless charing is a promising technology [25], where a power cord is not needed anymore between the charger and receiver. so that it is very convenient and easy to use for mobile nodes. The technology has many advantages such as no requirement of line-of sight, supporting multiple mobile devices and no additional alignment needed. The main challenge is how to satisfy all user's requirements while keep the cost low.

Due to fast energy transfer efficiency decreasing [26], a wireless charge can only cover a small range of area. Thus it is costly to deploy ample chargers in the whole area. Charging sequence arrangement [27] which schedules a mobile node to a serial of chargers at the specified time and location, is also not feasible, since it is impossible to gather all nodes' trajectory information not mention that many of which are stochastic. Wireless charging vehicle [28][29] won't work in our scenario because the trajectory planning need the location information for each node which is not trivial in MANETs. In most cases, a mobile node needs to move to the charger itself when it feels necessary. However, since each node has its own planned trajectory and energy consumption rate, it is not possible to schedule a node to a charger in advance. From the view point of the node itself, the desired situation is that a closest charger will be ready when it raises a request of energy replenishment. Therefore, additional movement could be minimized. Unlike above studies, in this paper the wireless chargers are already installed in the area with the respect of efficiency and cost. The main goal is to schedule a mobile node to a charger as close as possible when it posts an energy replenishment request so that the average delay(moving duration) and cost(moving distance) can be minimized.

A. Charger assignment problem in MANETs

Similar like the hospital assignment problem, there are resources and requestors, and we need to make the best match between them. Since the residual energy of each mobile node tends to be quite different when they post a request, we can also group them into three categories which are critical, serious and mild respectively. The energy depletes very quickly in some devices so that they need to find a charger as soon as possible. We call them critical nodes in that case. Some nodes would like to remain their residual energy above a threshold for the sake of safety. They will raise charging request more frequently. But, actually, they can bear some time before being charged. We call these nodes mild nodes. All other nodes between critical and mild could be regarded as serious. They can hold for some time but not too long. Otherwise they will also shut down due to flat energy. Regarding chargers, we consider they are identical in functioning so that the background is different from the one in the scenario of hospital assignment. Below we will surmise the feature of the scenario.

1) Capacity of chargers: Apparently, considering the energy storage and radiation capability, each charger will have a charging capacity, denoted by C_y. We assume that each energy receiver possesses same portion of resources so that C_y can be explained as the number of mobile nodes that the charger y can support at a time. For simplicity, C_y is a positive integer.

2) Effective charging distance: Theoretically, a charger can transfer energy to a receiver through wireless means within some range. However, the energy transfer efficiency drops quickly when the distance increases [25]. Therefore, to ensure the reception power at a receiver is larger than a threshold so that it is large enough to charge the node, mobile nodes requiring charging must come into the effective charging distance of a charger denoted by D. Because the size of the D will not affect the result of the assignment, we set the D as 0.

3) Mobility of chargers: In the first scenario, all hospitals are stationary while in this scenario the chargers can be small in size so that they are possible to be mounted on public transporters such as trains, buses, and ferries. Therefore, the mobility of chargers must be considered. However, the mobility is relative which dose not mean a charger will move towards a receiver in purpose. Let us consider a case that a user is waiting a subway at the platform and the train is coming. We should not assign the user to the charger in the platform, but to the one in the coach. We group chargers into two groups, one is moving with the mobile nodes (denoted by H^M, the other is not (denoted by H^S). Obviously that when a user is in the coach, it should not be assigned with the charger located in the platform. Then, we can model the scenario in the similar form with the previous one. Here we adopt a simple model where we do not consider the dynamic distance changing between charger and the node. Otherwise the speed and direction should be involved. The dynamic changing of distance will only affect the weight and it will be considered in the future work.

4) Charging duration: The algorithm will end in the first scenario when a patient is successfully assigned to a hospital. In this scenario, due to nodes mobility, such as a passenger riding a subway, a node may only stay with a charger for a short of time. During that short time duration, the job of charging may not be finished. Therefore, a node may be scheduled to multiple chargers until it is satisfied with the residual energy. We divide the time into T equal time slots indexed by $1, 2, ..., T$, and each time slot is denoted by U_t. Note that the battery capacity and energy consumption rates of different nodes may significantly vary, due to different hardware configuration and behaviors, we set a charger duration of U_x for each node x where U_x is an integer multiple of time slot U_t. Also, for some nodes, the charging duration could be shorter than U_t.

5) Appearance of charging request: A charging request could appear anywhere and anytime. It could be critical, serious and mild as well, just like the patient in a big city. The appearance pattern is similar so that we model the appearance of charging request with Poisson distribution.

6) Local waiting queue: When the delay to a further charger is greater than the expected delay of waiting locally, a node will be assigned to the waiting queue of the local charger.

$$\sum_{i=w}^{\infty} P(i, \frac{C_k}{d_k} \times Min\{T_k\}) > 0.9 \tag{9}$$

Where $Min\{T_k\}$ denotes the distance (represented by time) of the nearest charger to the charger y_k. C_k and d_k denote the capacity and average charging time of y_k respectively. Namely, we hope the total possibility of leaving w_k nodes in $Min\{T_k\}$ time is larger than 90%. Otherwise it is better for the node going to a further charger. We use parameter 90% which is more strict than that in hospital assignment. The larger the $Min\{T_k\}$ is, the larger the w_k. It is normal that nodes would want to wait at the closer charger, as opposed to going to the further one.

7) Reservation: The same contradiction exists in this scenario. For example, when a critical node posts a charging request, the nearest charger is all occupied. And the residual energy will run out before it reaches the further charger. Therefore, we need to hold some resources as the reservation for critical nodes. The calculation of the reservation amount can also use Eq.(8).

B. Bipartite matching based algorithm

From above discussions, we will formulate the charger assignment problem as follows:

$$\begin{array}{lll} & \max \sum_{x\in X}\sum_{y\in Y}\sum_{0\le t<\infty} R(x,y)m(x,y) & \\ s.t. & \text{every } m(x,y) = 0,\ 1, \text{ or ``}-\text{''} & i) \\ & \sum_{y\in Y} m(x,y) = 1 \text{ for every } x \in X & ii) \\ & \sum_{x\in X} m^t(x,y) \le C_y^t \text{ for every } y \in Y \text{ and any } t & iii) \end{array}$$

where x denotes mobile nodes, y denotes chargers, "−" indicates an initial or unreachable status. We use constraint i) to ensure the charger assignment as a bipartite matching. Constraint ii) does not allow a mobile node being charged by more than one charger at the same time. Constraint iii) ensure that the total number of charging mobile nodes does not exceed the limitation. It is not surprise that two different scenario could be modeled as a same problem. Therefore, we can also use the Alg. 1 derived from the first scenario to solve this problem. Here we will skip the steps of describing since it has been illustrated thoroughly in the first part of this paper.

C. Results analysis and discussion

The scenario and parameter are different in two system, but the key problem is the same, i.e., the optimization of the resources. When each requestor can only be assigned to one resource and has different weight for each resource, we could consider using the bipartite matching to solve it. In the hospital assignment, the patient is the requestor and the hospital is the resource while in charger assignment, the mobile node is the requestor and the charger is the resource. If the total number of requestor is not too big and central scheduling is available, the algorithm is compatible with respect to cost and efficiency. Resource assignment is a common problem in communication and network, in this paper, we propose solution extended from traditional bipartite matching which considers the mobility and time involving of the system. It is an online algorithm with less proactive knowledge requirement and the search space could be limited to sub area so that to bring down the time and space complexity.

VII. Conclusions

In this paper, we consider the assignment problem between requesters and resources in MANETs. The resource can either act as a data storage or a provider of energy. Two scenario are discussed, i.e., hospital assignment and energy charging. In both scenarios, there are multiple type of resources and requesters which introduce more strict assignment constraint. Also, time involving facts increase the difficulty of the problem. We propose a novel resource assignment to minimize the average delay of requesting nodes, as well as the amount of failure-of-assignment for critical nodes in the system, denoted by HAEP. The framework is composed of three components, i.e., Service Center, Mobile nodes, and Resource providers. To avoid the disadvantage of local competition, resource providers will submit their occupancy status while mobile nodes call the Service Center to declare their requirements so that the schedule could be done in the Service Center in a global view manner. The solution is built on the Hungarian algorithm, with the prediction and time scale, applied on a multi-dimension resource and requesters. The The simulation shows that our work is very efficient compared with the three other usual methods.

Acknowledgment

This work is supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry of China, and Chinese Scholarship Council (201208330096). This work is also supported in part by NSF grants CNS 149860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963.

References

[1] Y. Li, “The application of improved early pre-consultation assessment in emergency treatment,” *Guide of china Medicine*, vol. 9, no. 8, pp. 126–127, 2011.

[2] ChinaMobile, “You have to be lucky to find a hospital that has room,” ONLINE, http://labs.chinamobile.com/news/iot/86608, 2012.

[3] timesofindia, “Hospitals lack enoughbeds, admits government,” ONLINE, http://timesofindia.indiatimes.com/city/patna/Hospitals-lack-enoughbeds-admits-government/articleshow/37798085.cms, 2014.

[4] S. Yao, Q. zeng, M. Peng, S. Ren, G. Chen, and J. Wang, “Stop violence against medical workers in china,” *Journal of Thoracic Disease*, vol. 6, no. 6, pp. 141–145, 2014.

[5] Firstpost, “Footballer in india dies due to lack of hospital beds,” ONLINE, http://www.firstpost.com/sports/footballer-in-india-dies-due-to-lack-of-hospital-beds-372272.html, 2012.

[6] Hexun, “Nearly 1000 calls perday for ”120” emergency call,” ONLINE, http://news.hexun.com/2012-12-14/149027241.html, 2012.

[7] S. Krug, M. Siracusa, S. Schellenberg, P. Begerow, J. Seitz, T. Finke, and J. Schroeder, “Movement patterns for mobile networks in disaster scenarios,” in *WoWMoM*. IEEE, 2014, pp. 1–6.

[8] C.-Y. Chen, P.-Y. Chen, and W.-T. Chen, “A novel emergency vehicle dispatching system,” in *VTC Spring*. IEEE, 2013, pp. 1–5.

[9] J. Bondy and U. Murty, *Graph Theory with Applications*, 1st ed. Elsevier Science Publishing Co., Inc, 1976.

[10] xray, “Assignment problem and hungarian algorithm,” topcoder, https://www.topcoder.com/community/data-science/data-science-tutorials/assignment-problem-and-hungarian-algorithm/.

[11] W. Jin, “Analysis of in-patient difference between downtown and suburban in shanghai,” *China Medical Herald*, vol. 7, no. 6, pp. 150–151, 2010.

[12] A. Federgruen and H. Groenevelt, “The greedy procedure for resource allocation problems: necessary and sufficient conditions for optimality,” *Operations Research*, vol. 34, no. 6, pp. 909–918, 1986.

[13] M. Xu, T. Wong, S. Wong, K. Chin, K. Tsui, and R. Hsia, “Delays in service for non-emergent patients due to arrival of emergent patients in the emergency department: a case study in hong kong,” *The Journal of Emergence Medicine*, vol. 45, no. 2, pp. 271–280, 2013.

[14] P. Beraldi and M. E. Bruni, “A probabilistic model applied to emergency service vehicle location,” *European Journal of Operational Research*, vol. 196, no. 1, pp. 323–331, 2009.

[15] O. I. Alsalloum and G. K. Rand, “Extensions to emergency vehicle location models,” *Computers & OR*, vol. 33, pp. 2725–2743, 2006.

[16] J. Luo, J. Wang, and H. Yu, “A dynamic vehicle routing problem for medical supplies in large-scale emergencies,” in *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. IEEE, 2011, pp. 271–275.

[17] A. Senart, M. Bouroche, and V. Cahill, “Modelling an emergency vehicle early-warning system using real-time feedback,” *International Journal of Intelligent Information and Database Systems (IJIIDS), Special Issue on Information Processing in Intelligent Vehicles and Road Applications*, vol. 2, no. 2, pp. 222–239, 2008.

[18] H. Noori, “Modeling the impact of vanet-enabled traffic lights control on the response time of emergency vehicles in realistic large-scale urban area,” in *ICC*. IEEE Computer Society, 2013, pp. 526–531.

[19] B. Cardoen, E. Demeulemeester, and J. Beliën, “Operating room planning and scheduling: A literature review,” *European Journal of Operational Research*, vol. 201, no. 3, pp. 921–932, 2010.

[20] J. C. Lowery, “Design of hospital admissions scheduling system using simulation,” in *Winter Simulation Conference*. ACM, 1996, pp. 1199–1204.

[21] W. R. Reinus, A. Enyan, P. Flanagan, B. Pim, D. S. Sallee, and J. Segrist, “A proposed scheduling model to improve use of computed tomography facilities,” *Journal of Medical Systems*, vol. 24, no. 2, pp. 61–76, 2000.

[22] D. Min and Y. Yih, “An elective surgery scheduling problem considering patient priority,” *Computers & OR*, vol. 37, no. 6, pp. 1091–1099, 2010.

[23] S. Lu, K. Ju, and K. H. Chon, “A new algorithm for linear and nonlinear ARMA model parameter estimation using affine geometry [and application to blood flow/pressure data],” *IEEE Trans. Biomed. Engineering*, vol. 48, no. 10, pp. 1116–1124, 2001.

[24] Y. Huang, “Using markov model to predict the incidence of tuberculosis in taiwan for the next decade,” Master’s thesis, National Yang-Ming University, 2006.

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

[26] L. Xie, Y. Shi, Y. Hou, W. Lou, H. Sherali, and S. Midkiff, “Multi-node wireless energy charging in sensor networks,” *IEEE/ACM Transactions on Networks*, 2014.

[27] W. Xu, W. Liang, S. Hu, X. Lin, and J. Peng, “Charging your smartphones on public commuters via wireless energy transfer,” in *IPCCC*. IEEE Computer Society, 2015, pp. 1–8.

[28] W. Xu, W. Liang, X. Lin, G. Mao, and X. Ren, “Towards perpetual sensor networks via deploying multiple mobile wireless chargers,” in *ICPP*, 2014, pp. 80–89.

[29] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, “Maintaining sensor networks perpetually via wireless recharging mobile vehicles,” in *LCN*. IEEE, 2014, pp. 270–278.