Reserved optimisation: Handling Incident Priorities in Emergency Response Systems

Muralidhar Konda*
School of Information Systems
Singapore Management University
muralidhark@smu.edu.sg

Supriyo Ghosh*
School of Information Systems
Singapore Management University
supriyog.2013@phdis.smu.edu.sg

Pradeep Varakantham
School of Information Systems
Singapore Management University
pradeepv@smu.edu.sg

Abstract

Emergency (medical, fire or criminal) Management Systems (EMSs) are crucial for ensuring public safety and security. Typically in many cities, less than 20% of the cases received by EMSs belong to the extremely serious category and require immediate help. Rest of the incidents typically are less serious and thereby allow more flexibility in response time. Therefore, for efficient management of EMS requests, several EMSs now categorise an incoming emergency request into a priority level based on well studied "triaging" methods. Leading research on optimising emergency response has either focussed on data-driven models for settings with homogenous incidents or on generic heuristics (that are not data-driven) in multi-priority incident settings.

In this paper, we provide data-driven models that employ tiered optimisation of allocation and dispatch simultaneously to ensure high priority incidents are served effectively. To that end, we make the following contributions in this paper: (1) For a given dataset of historical incidents, we first provide an optimisation model that maximises the percentage of highest priority incidents served within a threshold response time, while ensuring threshold response times for other priority incidents. Apart from optimising the allocation, this optimisation model also provides a detailed dispatch strategy that fits the given set of historical incidents well; (2) To better handle high variance (spatial and temporal) in arrival of high priority incidents, we reserve a set of ERVs for high priority incidents. Our second contribution is in modifying our optimisation model to reserve a subset of ERVs for high priority incidents while considering a minor modification to nearest available ERV dispatch strategy; and (3) Finally, using a real-world EMS data set, we experimentally demonstrate that our solution with a detailed dispatch strategy outperforms the existing benchmark approach. Moreover, due to the presence of few high priority incidents and significant spatio-temporal uncertainty associated with them, we show that a simple dispatch strategy with reserved ERVs outperforms the detailed dispatch strategy.

1 Introduction

Emergency (e.g., medical, fire or criminal) Management Systems (EMSs) play an important role in maintaining public safety and health-care services. In an EMS, a set of base stations are strategically deployed throughout the city and a fleet of Emergency Response Vehicles, ERVs (e.g., ambulances, fire rescue bikes, police vehicles) are strategically positioned at those base stations. Once an emergency request arrives in the system, the operators typically dispatch the nearest available ERV to assist the victim. After providing an initial treatment, the ERV transfers the patient to a nearby hospital (if required) and returns back to the base from where it was dispatched. EMS is an extremely sensitive domain, as reducing the response times (i.e., the time taken by an ERV to reach the incident location after the request arrived in the system) for emergency needs by even a few seconds can save human lives.

Typically in many cities, less than 20% of the cases received by EMS operators belong to the extremely serious category and require immediate care. Rest of the cases belong to the serious category but can afford flexibility in terms of response time. In recent times, EMS operators have employed formal "triaging" methods[1] to assign priorities to incidents based on the information revealed over the phone with regards to the emergency. The next crucial step for effective utilisation of EMSs is to provide response proportional to the incident priorities.

Recent research (Yue, Marla, and Krishnan 2012; Saisubramanian, Varakantham, and Chuin 2015; Ghosh and Varakantham 2016) in improving effectiveness of EMS has utilised data-driven optimisation models for allocation of ERVs and placement of base stations. However, these data-driven optimisation models assume that all the emergency incidents are homogeneous in terms of criticality and therefore, just optimising allocation of ERVs with a typical dispatch strategy (i.e., dispatch the nearest available ERV) is sufficient to provide fast response. However, when there are multi-priority incidents present, ERV dispatch needs to be conditioned on the priority of the incident and hence it plays a significant role along with ERV allocation. Therefore, in this paper, we provide data-driven optimisation models that not only optimise allocation but also optimise dispatch conditioned on incident priorities.

There have also been other works (Kuisma et al. 2004; Sudtachat and Mayorga 2013; Gnanasekaran et al. 2013) that have considered multi-priority incidents in EMS. However, none of these works have considered optimisation of response time for high priority incidents while exploiting flexibility with lower priority incidents. In this paper, we address this crucial issue while considering incidents from past

*First two authors contributed equally to the paper. Most of the work was done while the second author was at Singapore Management University.

[1]Triaging questions are typically worked out in conjunction with emergency medical departments at hospitals.

data. Specifically, our goal is to maximise the percentage of requests served within a threshold time bound (i.e., bounded time response) for highest priority requests while meeting desired bounded risk response metric on other priority requests (e.g., assist 90% of lower priority requests within 13 minutes) through priority contingent dispatch and efficient allocation of ERVs. To that end, our key contributions are as follows:

1. We propose a mixed integer linear programming (MILP) model that maximises the bounded time response metric for high priority requests[2], while ensuring that the bounded risk response times for low priority requests are bounded by given threshold values. Furthermore, we also obtain a detailed ERV dispatch strategy from this optimisation model that fits the given set of historical incidents well.
2. Since high priority incidents are typically very few and there is a significant variance associated with them, reserving ERVs for high priority incidents can be useful. We provide a reserved optimisation model that reserves a subset of ERVs for high priority requests. Unlike the first contribution where we obtain an intricate ERV dispatch strategy, due to reservation of ERVs, we obtain a much simpler nearest available ERV based dispatch strategy[3] in this approach, where a high priority request is assisted by the nearest ERV (from the entire fleet of ERVs) and the low priority request is served by the nearest non-reserved ERV.
3. By employing an anonymous real-world EMS data set, we experimentally demonstrate that our solution with an intricate dispatch strategy outperforms the existing benchmark approach. However, we also illustrate that our second contribution with a simpler dispatch strategy outperforms the approach with a detailed dispatch strategy. This can be attributed to the presence of significantly lower ($\leq$ 20%) high priority incidents and significant spatio-temporal uncertainty associated with them.

2 Related Work

Due to the practical importance of EMS, a wide variety of research papers have studied ERV allocation and dispatch problems. We categorise them into three relevant threads of research. The first thread of research focuses on learning efficient ERV dispatch policies for effective emergency response. (Andersson and Värbrand 2007; Schmid 2012; Ibri, Nourelfath, and Drias 2012; Bjarnason et al. 2009) develop simulation approaches to discover an efficient strategy for dispatching ERVs. Furthermore, they provide techniques to frequently relocate ERVs to support the learnt dispatch strategy. However, frequent relocation of ERVs incurs substantial relocation cost and moreover, ERVs are not utilised during the relocation period. On the contrary, we learn a data-driven dispatch strategy that supports a fixed optimised allocation. (Carter, Chaiken, and Ignall 1972) show that typical nearest available ERV dispatch strategy is not ideal for EMS with multi-priority incidents. In this paper, we also provide evidence for this claim, while considering data-driven models and optimisation of allocation. (Culley et al. 1994) propose a heuristic based simulation model to learn a dispatch strategy for multi-priority incidents, by assuming that a request can be served from any base location. However, (Carter, Chaiken, and Ignall 1972) show that the effectiveness of EMS increased when a request is served from a base that lies within the geographical proximity of the demand location. Therefore, we assume that a request can only be served from a set of nearby feasible bases.

The second thread of research papers focus on strategic placement of base stations and allocation of ERVs. (Maxwell et al. 2010) provide a dynamic relocation model for single ERV. In contrast, our goal is to allocate an entire fleet of ERVs. (Brotcorne, Laporte, and Semet 2003; Gendreau, Laporte, and Semet 2006) propose mathematical optimisation or local search based heuristics to solve the allocation problem. However, these optimisation approaches do not capture the spatio-temporal dynamics of EMS. Recent research (Saisubramanian, Varakantham, and Chuin 2015; Yue, Marla, and Krishnan 2012; Restrepo, Henderson, and Topaloglu 2009; Ghosh and Varakantham 2016) has successfully employed data-driven optimisation models or event-driven simulators for ERV allocation. However, these approaches consider only homogeneous incidents and allocate ERVs without taking into account the ERV dispatch strategy. In contrast, we learn a dispatch strategy from the data-driven optimised solution.

The third thread of research focuses on effective handling of multi-priority incidents in EMS. (Kuisma et al. 2004) show that modifying the dispatch strategy in favour of high priority incidents do not affect the pre-hospital mortality of low priority requests. (Sudtachat and Mayorga 2013) propose a simulation model to find the allocation of ERVs in case of multi-priority incidents. However, their simulation approach is a trial and error approach and does not provide a mechanism to optimise key EMS metrics. They do indicate that dispatch policies are critical and just using nearest ERV dispatch is typically not sufficient. (Gnanasekaran et al. 2013) assume that only high priority requests need to be transferred to hospitals, which is not a realistic assumption. In addition, based on the model of (Andersson and Värbrand 2007), they propose to frequently relocate the ERVs. This paper builds on these works and is different in the following key ways: (i) we provide data-driven models that optimise key EMS metrics for high priority incidents while achieving a bounded performance for lower priority incidents; and (ii) more importantly provide a principled approach for obtaining dispatch policies to deal with multi-priority incidents.

3 Prioritized Emergency Response (PER)

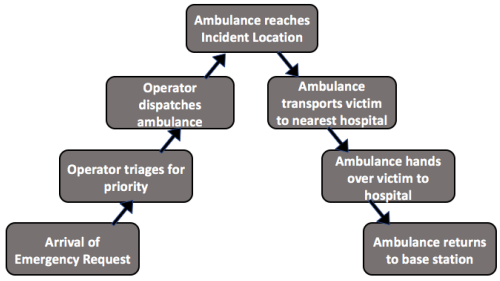
We now describe the overall process in the context of ambulance response. Other kinds of emergency response (fire, crime and traffic) have similar process flows. Figure 1(a) provides this process flow. Once the dispatch centre receives

[2]Bounded time response metric is the number of requests assisted within a threshold time.

[3]Note that, due to the presence of reserved ERVs, our priority contingent nearest available ERV dispatch strategy is slightly different than the existing typical nearest ERV dispatch strategy.

a call about an emergency request, the operator asks the caller a fixed set of questions to identify the priority of the incident. Depending on the priority of the incident and the dispatch strategy, operator dispatches an ambulance as early as possible to the incident location. Once the ambulance reaches the incident location, the paramedics provide an initial first aid and treatment before transporting the patient to the hospital. On reaching the hospital, the patient is handed over to the emergency department and the ambulance travels back to its home base station.

EMS operators store information about each of these phases including incident priority and our goal is to use past data of these incidents to compute new allocation and dispatch strategies. We employ a data-driven approach, where we compute best allocation and dispatch strategies for a set of training data samples and then use those allocation and dispatch strategies on a testing data set. Before providing the optimisation models used to compute allocation and dispatch strategies, we first provide a formal model for the problem on training data samples.



(a)

Frequency Matrix for Time Interval 12:00 - 3:00 PM on Monday

		Priority		
		P1	P2	P3
Base ID	B1	5	65	30
	B2	15	50	35
	...	...	...	...
	BN	...	...	...

(b)

Figure 1: (a) Process flow for ambulance response; and (b) An example frequency matrix obtained from dispatch variables, $\boldsymbol{x}$.

Formal Model

The formal model for Prioritized Emergency Response (PER) is defined using the following tuple:

$$\langle \mathcal{B}, \mathcal{A}, \mathcal{R}, \boldsymbol{T}, \boldsymbol{C}, \alpha, \boldsymbol{b} \rangle$$

$\mathcal{B}$ denotes the set of base stations and $\mathcal{A}$ is a fleet of ERVs. $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots\}$ denotes the set of emergency requests, with $\mathcal{R}_i$ representing requests with priority i. Each request $r \in \mathcal{R}_i$ contains the following information $\langle t, s, h, \mathcal{B}_r, \lambda_r, \mu_r \rangle$, where t is the arrival time, s is the origin location and h denotes the destination hospital and i denotes the priority of request r. $\mathcal{B}_r$ denotes a set of nearby feasible bases from which the request r can be assisted. μ_r represents the response times for each of the nearby bases in $\mathcal{B}_r$. Precisely, if $\mathcal{B}_r = \{l_1, l_2, ...\}$, then μ_r^i denotes the response time from base l_i. λ_r represents the round-about times (i.e., the difference between the time an ERV left the base for assisting an emergency request and the time when it returns back to the base) for the nearby bases. $\boldsymbol{T}$ provides the ERV travel time between any two locations. Precisely, $T_{l',l''}$ represents travel time between source location l' to destination l''. $\boldsymbol{C}$ represents the capacities of the base stations, where C_l denotes the capacity of base $l \in \mathcal{B}$. $\alpha = \{\alpha_1, \alpha_2, ...\}$ indicates the bounded risk percentile for different priorities. For example, if the constraint is to serve 90 percent of priority 2 requests in a threshold, then $\alpha_2 = 0.1$. Finally, b_i denotes the upper bound on the α-response time for priority i requests.

With the given input tuple, our goal is to find an effective dynamic[4] allocation of an entire fleet of ERVs that maximises the number of high priority requests served efficiently while ensuring that α-response time for priority i ($i > 1$) requests (say δ_i) is less than b_i.

All the aspects of the PER model are either directly or indirectly populated from historical data of incidents or are provided by EMS operators. In the next section, we provide an optimisation model that can be used to compute allocation and dispatch strategy for ERVs for a given set of training data samples.

4 Prioritised Time Bounded Optimisation

In this section, we describe an optimisation model that provides tiered response to incidents with multiple priorities. Specifically, we maximise the number of priority 1 requests served within a bounded time limit (for priority 1 requests) while ensuring that upper bounds on α-response times for requests with lower priority (i.e. priority > 1) are satisfied. We refer to this approach as Prioritised Time Bounded Optimisation (PTBO). As indicated earlier, PTBO is run on a given set of training data samples.

More concretely, PTBO is a Mixed Integer Linear Programming (MILP) model that not only computes an optimal allocation for a fleet of ERVs (to base stations) but also provides a dispatch strategy for the given set of training data samples. We first describe the objective and constraints in this MILP model and then explain how we derive dispatch strategy from the optimised solution. The key decision variables employed in the MILP are:

δ_i: is the α-response time for request priority i. δ_i^r denotes the response time for request r with priority i.

x_{rl}: is the binary decision variable that is set to 1 if request r is assigned to base station l. In other words, an ERV from

[4]For dynamic allocation, the allocation strategy changes on every weekday. For example, we consider the set of requests of past Mondays to find an allocation for a Monday.

base station l is dispatched to serve request r. Therefore, this set of decision variables are referred to as the dispatch variables and are used to compute a dispatch strategy.

a_l: is an integer decision variable which denotes the number of ERVs allocated to base l and can take any integer value between 0 to the capacity of base l, C_l.

Our objective is to maximise the number of priority 1 requests served efficiently and to facilitate this we define an utility function L which gives one unit of reward if a request is served within $\mathcal{T}$ minutes and is defined as follows:

$$L_{rl} = \begin{cases} 1 & \text{if } T_{l,r.s} \leq \mathcal{T} \text{ minutes} \\ 0 & \text{Otherwise} \end{cases}$$

Our goal is therefore:

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r \in \mathcal{R}_1} \sum_{l \in \mathcal{B}_r} x_{rl} L_{rl}$$

Intuitively, the constraints associated with the real dynamics and objective of EMS are as follows:

A request can be served from maximum one base station: These constraints enforce that a request can only be assisted from one of the neighbouring feasible base station. If all the feasible bases are empty, the request is assigned to a dummy base $\perp$ and we call it as a null assignment (i.e., the request cannot be served).

$$\sum_{l \in \{\mathcal{B}_r \cup \perp\}} x_{rl} = 1, \qquad \forall r \in \mathcal{R} \tag{1}$$

A request can be served from a base, only if it has at least one idle ERV: To define these constraints, we first introduce the notion of parent requests. Let, P_r^l be the set of parent requests of request r for base l. A request r' belongs to the set P_r^l only if it has arrived before request r and if an ERV is assigned for request r' from base l, then the ERV would be busy in serving request r' when request r has arrived. Specifically, if a request r arrives in the interval within the round-about time from the arrival time of r' and is served from base l then r' would belong to P_r^l. Therefore, these set of constraints enforce that we can assign a request r to base l only if there is at least one ERV which is not serving the parent requests (i.e., $\sum_{j \in P_r^l} x_{jl} < a_l$).

$$x_{rl} + \sum_{j \in P_r^l} x_{jl} \leq a_l, \qquad \forall r \in \mathcal{R}, l \in \mathcal{B}_r \tag{2}$$

Compute response times for emergency requests: These set of constraints are used to compute the response times for the requests according to the assignment decisions. If the request is assigned to the dummy base, then we impose a high penalty and set the response time to a large number $\hat{M}$.

$$\delta_i^r \geq \sum_{l \in \mathcal{B}_r} x_{rl} \cdot T_{l,r.s} + x_{r\perp} \cdot \hat{M}, \quad \forall i \in N, r \in \mathcal{R}_i \tag{3}$$

Enforce the conditions on α-response time: Let, z_i^r be a binary decision variable which is set to 1 if the response time for request r of priority i, δ_i^r exceeds the α-response time δ_i. Then, these set of constraints ensure that the proportion of priority i requests whose response time exceeds the α-response time δ_i, is bounded by α_i, where M represents a significantly large number.

$$\frac{\delta_i^r - \delta_i}{M} \leq z_i^r, \qquad \forall i \in N \setminus 1, r \in \mathcal{R}_i \tag{4}$$

$$\frac{\sum_{r \in \mathcal{R}_i} z_i^r}{|\mathcal{R}_i|} \leq \alpha_i, \qquad i \in N \setminus 1 \tag{5}$$

The entire fleet of ERVs is allocated: This constraint enforces that the total number of allocated ERVs is exactly equal to the ERV fleet size.

$$\sum_{l \in \mathcal{B}} a_l = |\mathcal{A}| \tag{6}$$

α-response times for requests with priorities > 1 follow the given bounds: Finally, these set of constraints ensure that the α-response times for low priority (priority > 1) requests satisfy the given upper bounds.

$$\delta_i < b_i, \quad \forall i \in N \setminus 1 \tag{7}$$

Regularization: To avoid the overfitting of the optimisation model to the given training request set $\mathcal{R}$ (and extend well to testing set of requests), we include a regularization term in our objective. We divide the request set $\mathcal{R}$ into two parts - a validation set $\mathcal{R}^1$ and a training set $\mathcal{R}^2$. First we run our optimisation model with validation request set $\mathcal{R}^1$ and generate an allocation strategy $\hat{\boldsymbol{a}}$. In the next phase, we use the training request set $\mathcal{R}^2$ to obtain the final allocation. Our modified objective function (8) includes a regularised term with a constant learning parameter γ to reduce the overfitting according to training incidents $\mathcal{R}^2$.

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r \in \mathcal{R}_1} \sum_{l \in \mathcal{B}_r} x_{rl} L_{rl} - \gamma \cdot \sum_{l \in \mathcal{B}} |a_l - \hat{a}_l| \tag{8}$$

Intuitively, this ensures that we find an allocation for training set that is also close to the optimal allocation for the validation set. This is in tune with the broad idea of regularization that penalizes overfitting to the training data alone.

The objective function (8) is linearised using the following set of expressions (9)-(11).

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r \in \mathcal{R}_1} \sum_{l \in \mathcal{B}_r} x_{rl} L_{rl} - \gamma \cdot \sum_{l \in \mathcal{B}} \beta_l \tag{9}$$

$$\textbf{s.t.} \ \beta_l \geq a_l - \hat{a}_l \qquad \forall l \in \mathcal{B} \tag{10}$$

$$\beta_l \geq \hat{a}_l - a_l \qquad \forall l \in \mathcal{B} \tag{11}$$

Deriving Dispatch Strategy from Dispatch Variables, x: As indicated earlier, we also derive a dispatch strategy from the MILP (in addition to the allocation strategy) using the dispatch variables $\boldsymbol{x}$. We employ PTBO to generate allocation strategies and dispatch variables, $\boldsymbol{x}$ for all the weekdays separately specialised to the training data samples. From the optimised assignment solution of weekday w, we first create a frequency matrix $\boldsymbol{d}_w$, where $d_{w,t}^l$ represents the count on the number of high priority requests that are served from

base l for weekday w and time-slot[5] t. Figure 1(b) provides an example of a frequency matrix for a time slot of 12:00 - 3:00 PM that is computed from an optimised solution.

Then we utilise these frequency matrixes for dispatching ERVs when dealing with new requests as follows:

- Priority 1 requests are always served from the nearest base which has at least one idle ERV.
- Other priority requests (i.e., with priority greater than 1) are served from one of the nearby (with response time less than the input bound for α-response time) bases based on current ambulances there and the numbers in the frequency matrix.

Specific details of this dispatch strategy are provided in the following simulator which is used for evaluating PTBO.

Simulator for Evaluating PTBO

We now describe an event-driven simulation model (inspired from Yue, Marla, and Krishnan 2012) that is used to evaluate the performance of our optimisation approach. The key difference from Yue *et al.*'s simulator is that we employ the dispatch strategy derived from the dispatch statistics, $\boldsymbol{d}_w$ obtained from PTBO.

Algorithm (1) demonstrates the key functions of our simulator. Let, ξ be an event list, where $e \in \xi$ represents an emergency request. ξ is sorted according to the arrival time of incidents. I denotes the set of available ERVs, which is initialised according to the given allocation strategy. Let, a_r be the assigned ERV id for request r, which is initialised as a null assignment. In each iteration, we pop the first element from ξ and if it is a new high priority request, then we dispatch the nearest available ERV to serve the request.

If the new request has a lower priority i, then we try to assign it from a base which does not impact the future higher priority incidents and also satisfy the given bounds on α-response time of priority i. Precisely, if the request arrives at weekday w and during the time-slot t, then we find the top high priority incident prone base l^* during that time period. If the base l^* has less than 2 ERVs available, then we do not serve the low priority request from l^*. Otherwise, we check the feasible bases of request r according to their response times and assign the nearest available ERV which satisfies the given threshold values for priority i. Once these conditions are checked for all the feasible bases, if the request is still unassigned, then we employ the typical nearest available ERV dispatch strategy. We then remove the assigned ERV, a_r from the available ERV set I and add a job-completion event in ξ at time $t_r(a_r)$, where $t_r(a_r)$ denotes the time when a_r will return back to the base. On the other hand, if the popped event is a job-completion event for request r, then we add the ERV, a_r to the available ERV set I. This iterative process continues until the event list becomes empty.

[5]The entire day is divided into few time-slots depending on the frequency of priority 1 requests.

Algorithm 1: PTBO-Simulator($\mathcal{R}, \mathcal{B}, \boldsymbol{A}$)

Initialize: $I \leftarrow \boldsymbol{A}$ // Initialise set of free ERV ;
$\xi \leftarrow \mathcal{R}$ sorted in arrival order ;
$\boldsymbol{a} = \{a_r | a_r \leftarrow \perp\}$ //Initialise as null assignment ;
repeat
  Pop next arriving event e from ξ;
  if *e=New Request r with weekday w, priority i* **then**
    if *i = 1* **then**
      $a_r \leftarrow Dispatch(r, I)$ // Dispatch nearest free ERV;
    else
      $t \leftarrow TimeSlot(\hat{t}_r)$ // compute time-slot from arrival time $\hat{t}_r$;
      $l^* \leftarrow \arg\max_{l \in \mathcal{B}} \boldsymbol{d}_{w,t}$ // find the top high priority incident prone base ;
      for $l \in \mathcal{B}_r$ *(according to response time)* **do**
        if $(l \neq l^* || I_{l^*} \geq 2) \& (T_{r.s,l} < b_i) \&$ $(I_l \geq 1)$ **then**
          $a_r \leftarrow Dispatch(r, I_l)$ // Dispatch ERV from base l ;
      if $a_r = \perp$ **then**
        $a_r \leftarrow Dispatch(r, I)$ // Dispatch nearest free ERV;
    $I \leftarrow I - \{a_r\}$ // Update free ERV;
    Push job completion event at time $t_r(a_r)$ into ξ;
  else if *e=job completion event for r* **then**
    $I \leftarrow I \cup \{a_r\}$ // Update free ERV;
until $(|\xi| > 0)$;
return $\{\boldsymbol{a_r}\}$

5 Reserved optimisation

PTBO exploits an intricate dispatch strategy that employs frequency values from optimisation over past incidents. Given the high variance (across space and time) in arrival of priority 1 requests, having ambulances statically set aside for priority 1 incidents is better than trying to dynamically allocate or dispatch ambulances. We refer to this approach as Prioritised Time Bounded Optimisation with Reserved ERVs (PTBO-RE). This approach can be generalised to the settings where EMS owns multiple classes of ERVs (i.e., high priority ERVs provide additional supports for life-threatening incidents).

We now describe a modified optimisation approach where we divide the ERVs into two sets - reserved ERVs, $\mathcal{A}^+$ that can only be used for high priority incidents, and non-reserved ERVs, $\mathcal{A}^-$ that can be employed for any incident. The MILP for the allocation problem with reserved ERVs is compactly shown in Table (1).

Let, a_l^+ be the number of ERVs allocated to base l from set $\mathcal{A}^+$ and a_l^- denote the number of allocated ERVs at base l from set $\mathcal{A}^-$. x_{rl}^+ denotes the assignment variable which is set to 1 if the reserved ERV is assigned for request r from

base l. Similarly, x^-_{rl} denotes the assignment variable which is set to 1 if a non-reserved ERV is assigned for request r from base l.

Our objective is to maximise the number of high priority requests that are assisted within $\mathcal{T}$ minutes (delineated in Expression 12), where we employ the same utility function L as mentioned previously. Constraints (23) enforce that the α-response times for low priority requests follow the given input bounds. Constraints (13) enforce that only one type (either reserved or non-reserved) of ERV from one of the feasible bases can be assigned to serve a request. Constraints (14) enforce that low priority requests cannot be assigned a reserved ERV. If a request r has high priority, then let, Q^l_r denotes the set of parent requests for high priority requests for a base l . Q^l_r helps to ensure the temporal dependencies between high priority requests in using the reserved ERVs are satisfied. A request r' belongs to Q^l_r if r' has high priority, it arrives in the system before request r and it is still active when request r has arrived and is served from base l. Therefore, constraints (15) ensure that a high priority request can be served with a reserved ERV from a base only if a free reserved ERV is present at that base. As non-reserved ERVs can be utilised for any type of request, the definition of parent set, P^l_r remains the same. Constraints (16) enforce that a request can be served with a non-reserved ERV from a base only if a free non-reserved ERV is present in that base. Constraints (17) compute the response times for the requests. Constraints (18)-(19) ensure the conditions for computing α-response times. Constraints (20) enforce that the total number of allocated ERV at base, l is bounded by its capacity, C_l. Finally, constraints (21)-(22) assure that the total number of allocated reserved and non-reserved ERVs is exactly equal to their fleet sizes, i.e., $|\mathcal{A}^+|$ and $|\mathcal{A}^-|$, respectively.

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r\in\mathcal{R}_1}\sum_{l\in\mathcal{B}_r}(x^+_{rl}+x^-_{rl})L_{rl} \quad (12)$$

$$\textbf{s.t.} \sum_{l\in\{\mathcal{B}_r\cup\perp\}}(x^+_{rl}+x^-_{rl})=1, \qquad \forall r\in\mathcal{R} \quad (13)$$

$$\sum_{i\in N\setminus 1}\sum_{r\in\mathcal{R}_i}\sum_{l\in\mathcal{B}_r}x^+_{rl}=0 \quad (14)$$

$$x^+_{rl}+\sum_{j\in Q^l_r}x^+_{jl}\le a^+_l, \qquad \forall r\in\mathcal{R}_1, l\in\mathcal{B}_r \quad (15)$$

$$x^-_{rl}+\sum_{j\in P^l_r}x^-_{jl}\le a^-_l, \qquad \forall r\in\mathcal{R}, l\in\mathcal{B}_r \quad (16)$$

$$\delta^r_i\ge\sum_{l\in\mathcal{B}_r}(x^+_{rl}+x^-_{rl})\cdot T_{l,r.s}+(x^+_{r\perp}+x^-_{r\perp})\cdot\hat{M}, \qquad \forall i\in N\setminus 1, r\in\mathcal{R}_i \quad (17)$$

$$\frac{\delta^r_i-\delta_i}{M}\le z^r_i, \qquad \forall i\in N\setminus 1, r\in\mathcal{R}_i \quad (18)$$

$$\frac{\sum_{r\in\mathcal{R}_i}z^r_i}{|\mathcal{R}_i|}\le\alpha_i, \qquad i\in N\setminus 1 \quad (19)$$

$$a^+_l+a^-_l\le C_l \qquad \forall l\in\mathcal{B} \quad (20)$$

$$\sum_{l\in\mathcal{B}}a^+_l=|\mathcal{A}^+| \quad (21)$$

$$\sum_{l\in\mathcal{B}}a^-_l=|\mathcal{A}^-| \quad (22)$$

$$\delta_i<b_i \qquad \forall i\in N\setminus 1 \quad (23)$$

$$a^+_l,a^-_l,\delta_i,\delta^r_i\ge 0, x^+_{rl},x^-_{rl}\in\{0,1\}, z^r_i\in\{0,1\} \quad (24)$$

Table 1: **TIMEALLOCATION**$(\mathcal{R},\mathcal{B},\mathcal{A},\alpha)$

Dispatch Strategy for PTBO-RE: Unlike the intricate dispatch strategy derived from dispatch variables for PTBO, we employ a dispatch strategy that is simple and contingent on the priority of the incident for PTBO-RE:

- If the incident is of priority 1, then we dispatch the nearest ERV from all available ERVs (reserved and non-reserved).
- If the incident is of priority > 1, then we dispatch the nearest ambulance among non-reserved ERVs.

Simulator for Evaluating PTBO-RE

We now present an event-driven simulation model that is used to evaluate the performance of PTBO-RE. Algorithm (2) demonstrates the key functionalities of the simulator. We use the simple priority contingent dispatch policy for PTBO-RE.

We employ two sets of available ERVs. I^+ denotes the available set of reserved ERVs and initialised according to allocation of $\mathcal{A}^+$. Similarly, I^- denotes the available set of non-reserved ERVs and initialised according to allocation of $\mathcal{A}^-$. The major difference between Algorithm (1) and Algorithm (2) is the dispatch strategy. Once we pop a new request r from the event list ξ, we dispatch the nearest available and permissible ERV for the request. In case of a high priority request, we dispatch the nearest available ERV (irrespective of the type of ERV) for the request. In case of a low priority request, we only dispatch the nearest available non-reserved ERV to serve the request. We remove the ERV from its corresponding available ERV set and add a job-completion event in the list ξ once the job is completed. On the other hand, if we encounter a job-completion event, then we add that ERV to available (according to the type of the ERV) ERV set. This iterative process continues until ξ becomes empty. Once the simulation is over, we have a valid solution for assignment of requests to bases, from which we can compute the percentage of requests served efficiently and α-response times for different priorities.

6 Experimental Results

We conduct our experiments using a simulated data set. The data set contains $|\mathcal{B}|$ base stations and $|\mathcal{A}|$ ERVs. We have 6 months of request logs in this data set with requests of three priorities. P1 requests are the most critical in nature and our goal is to maximise the number of these requests that are served within a threshold $\mathcal{T}$ while maintaining a bound on α-response times for P2 and P3 requests. We impose a tighter bound than the actual key performance indicator (KPI) for

Algorithm 2: PTBO-RE-Simulator($\mathcal{R}, \mathcal{B}, \boldsymbol{A}$)

Initialize: $I^+ \leftarrow \boldsymbol{A}^+$ //Initialise set of free reserved ERVs;
$I^- \leftarrow \boldsymbol{A}^-$ // Initialise set of free non-reserved ERVs;
$\xi \leftarrow \mathcal{R}$ sorted in arrival order ;
$\boldsymbol{a} = \{a_r | a_r \leftarrow \perp\}$ //Initialise as null assignment ;
repeat
  Pop next arriving event e from ξ;
  if *e=New Request r with priority i* **then**
    if *i = 1* **then**
      $a_r \leftarrow Dispatch(r, I^+, I^-)$ // Dispatch nearest free ERV from either reserved or non-reserved set of ERVs;
    else
      $a_r \leftarrow Dispatch(r, I^-)$ // Dispatch nearest free ERV from non-reserved set of ERVs only;
    if $a_r \in I^+$ **then**
      $I^+ \leftarrow I^+ - \{a_r\}$ // Update free ERV;
    else
      $I^- \leftarrow I^- - \{a_r\}$ // Update free ERV;
    Push job completion event at time $t_r(a_r)$ into ξ;
  else if *e=job completion event for request r* **then**
    if $a_r \in I^+$ **then**
      $I^+ \leftarrow I^+ \cup \{a_r\}$ // Update free ERV;
    else
      $I^- \leftarrow I^- \cup \{a_r\}$ // Update free ERV;
until $(|\xi| > 0)$;
return $\{\boldsymbol{a_r}\}$

P2 and P3 requests to ensure that the actual KPI bound [6] is satisfied on the testing set.

Over the 6 months of simulated requests, we have a small percentage of P1 requests and a significant number of P2 and P3 requests. It is usually the case in reality that the number of P1 incidents is few and the spatial, temporal uncertainty associated with P1 requests is quite high. Hence, the training and testing data may not have a consistent pattern of P1 requests. Therefore, our experimental results provide a worst-case analysis for our optimisation approaches. The performance of our approaches will only improve if the spatio-temporal distribution of P1 requests in training and testing data follows a consistent pattern.

In addition, we have the information about the location of base stations and hospitals. Each request log contains the following information (a) Incident location; (b) Priority of the request; (c) Arrival time; (d) A set of feasible nearby bases from where the request can be assisted; (e) Response time from each of the feasible base to scene location; and (f) Round-about time for each of the feasible base. While these specific details might not always be readily available, they can be estimated in a straight-forward approach (Ghosh and Varakantham 2016).

We compare our approaches against the data-driven optimisation model from Ghosh and Varakantham (2016)[7] that optimises a bounded time metric while discovering the allocation of ERVs to bases. As they considered homogeneous requests (i.e., all the requests have same priority), their objective is to maximise the number of requests that are served within a threshold time bound. Furthermore, they adopted a typical nearest available ERV dispatch strategy. Therefore, the performance of their ERV allocation policies can be evaluated by employing an event-driven simulator (adopted from Yue, Marla, and Krishnan 2012) that follows the nearest available ERV dispatch strategy. We refer to this approach as Time Bounded Optimisation (TBO). We evaluate the performance of the three approaches (PTBO, PTBO-RE and TBO) by employing their corresponding simulation model. However, it should be noted that all the simulators run through the requests in the same order (i.e., without having a knowledge of future requests).

We divide our 6 months of data set into two parts - first 3 months of data is used for training purpose to generate the allocation strategies and the performance of these allocations is tested on the last 3 months of data. We have generated separate allocations for each of the weekdays. For instance, the allocation for the Monday is generated using requests of all the Mondays on training data. For regularisation purpose (in case of PTBO), we further divide the training data into two parts - around 30% of the requests are used as the validation set and other 70% of requests are used to generate the allocations. We provide three threads of results on the testing data set: (a) Performance comparison of our approaches against the benchmark approach; (b) Effect of fleet size of ERVs on the performance of PTBO-RE; and (c) Runtime performance.

Performance Comparison: The first thread of results shows the performance comparison between our approaches (PTBO and PTBO-RE) while comparing to the benchmark approach (TBO). For these experiments, we reserve $|\mathcal{A}^+|$ ERVs for P1 requests for PTBO-RE. Figure (2) depicts the performance comparison for different priorities of requests. Figure 2(a) demonstrates the most important performance metric which is the extra percentage of P1 requests served within the given threshold time when compared to TBO. In the X-axis, we show different weekdays and Y-axis denotes the increased percentage of P1 requests served within the given threshold. As shown clearly, for all the weekdays, our approaches (PTBO and PTBO-RE) always outperform the existing TBO approach. On an average, PTBO serves 2% extra requests within the given threshold over the TBO approach. Most importantly, PTBO-RE provides 6% average gain over the benchmark approach, TBO in serving the P1 requests efficiently. While PTBO is proven to be effective than TBO, PTBO-RE always outperforms the PTBO approach. On an average, PTBO-RE is able to assist 4% extra P1 requests within the threshold time bound over PTBO.

[6] The bounds are decided through some preliminary experiments. A more thorough theoretical and empirical analysis for setting of bounds given KPI is left for future work.

[7] Refer to Table (1) of Ghosh and Varakantham (2016) for the details of the optimisation model, which is an extension of the approach proposed in Yue, Marla, and Krishnan (2012).

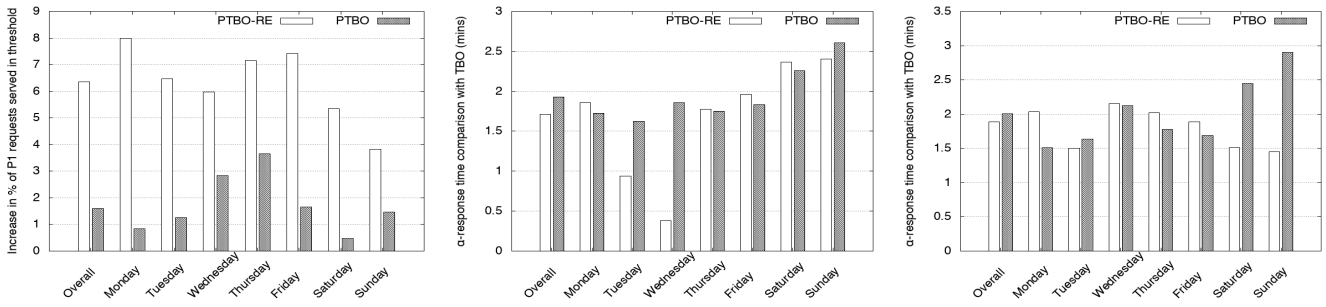


Figure 2: Performance comparison between TBO, PTBO, and PTBO-RE on: (a) Extra Percentage of Priority 1 requests served within threshold; (b) Difference in α-percentile response time for Priority 2 requests ; and (c) Difference in α-percentile response time for Priority 3 requests.

These results clearly indicate that dispatch strategy plays an important role when tackling multi-priority incidents and more importantly, due to uncertainties associated with lower number of P1 requests, a simple dispatch strategy with reserved ERVs provides better solutions than employing an intricate dispatch strategy.

Figure 2(b)-(c) delineate the performance comparison on P2 and P3 requests respectively. For these set of results, our goal is to bound the value of α-response times for P2 and P3 requests within our KPI. In the X-axis, we show different weekdays and Y-axis denotes the increase in α-response time when compared to TBO on the corresponding day of the week. The average α-response times for P2 and P3 requests for PTBO-RE were well within our KPI. As expected, TBO always provides better α-response times for P2 and P3 requests in comparison to our approaches. This is so because TBO assumes that all the requests are homogeneous and optimises the response times for P2 and P3 requests as well.

Effect of fleet size of ERVs: We evaluated the performance of PTBO-RE while varying the ERV fleet size. Specifically, we varied the number of reserved ERVs in intervals of 2 while we fix the total fleet size of ERVs. As the reserved ERVs are only used for serving P1 requests, and the number of non-reserved ERVs reduces when we increase the number of reserved ERVs, the performance fluctuates for P2 and P3 requests when the number of reserved ERVs is increased. However, as expected, the α-response time for P1 requests reduces monotonically. It should be noted that, while the α-response time for P1 requests reduces with increased number of reserved ERVs, it fails to satisfy our KPI for P2 and P3 requests (i.e., the α-response times for P2 and P3 requests are higher than the accepted threshold).

We also evaluate the performance of PTBO-RE when we fix the fleet size of reserved ERVs and vary the number of total ERVs from $|\mathcal{B}|$-10 to $|\mathcal{B}|$ in intervals of 2. We observe a consistent pattern in the performance for all the priorities of requests. As the non-reserved ERVs can be used to assist all priorities of requests, α-response times for all the priorities of requests reduce monotonically if we increase the number of non-reserved ERVs.

Timing results: In the last thread of results, we discuss the runtime performance of our approaches. It should be noted that the dynamic allocation of ERVs is an offline process for preparedness. In particular, we need to generate an allocation once in a day and therefore, the runtime is an important parameter for us. We observe that TBO is the fastest of all the three approaches as it does not consider the complex objectives for multi-priority incidents. However, even with additional complex constraints and objective, both our approaches (PTBO and PTBO-RE) scale gracefully with the increasing number of emergency requests. For our largest problem instance (i.e., with highest number of requests), PTBO and PTBO-RE are solved within 4 minutes and 6 minutes, respectively, and therefore, our approaches are suitable for solving the real-world large-scale problem instances.

7 Conclusion

In this paper, we provide two efficient data-driven optimisation approaches for effective emergency response with multi-priority incidents. For the first approach, we propose an optimisation approach to serve the high priority incidents efficiently while ensuring threshold response times for other priorities. We then learn an intricate dispatch strategy from the optimised solution and use that strategy on an event-driven simulator for evaluation. For the second approach, we propose an optimisation model by reserving a subset of ERVs for the high priority requests and employ a simple nearest available ERV dispatch strategy. The empirical results on a real-world data set demonstrate that both our optimisation approaches outperform the existing best-known approach from literature and are proven to be highly scalable. In future, this work can be extended with a robust optimisation technique which can take into account the high uncertainties associated with high priority requests.

8 Acknowledgements

This research was supported by the Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2 grant under research grant MOE2016-T2-1-174.

References

Andersson, T., and Värbrand, P. 2007. Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society* 58(2):195–201.

Bjarnason, R.; Tadepalli, P.; Fern, A.; and Niedner, C. 2009. Simulation-based optimization of resource placement and emergency response. In *IAAI*.

Brotcorne, L.; Laporte, G.; and Semet, F. 2003. Ambulance location and relocation models. *European journal of operational research* 147(3):451–463.

Carter, G. M.; Chaiken, J. M.; and Ignall, E. 1972. Response areas for two emergency units. *Operations Research* 20(3):571–594.

Culley, L. L.; Henwood, D. K.; Clark, J. J.; Eisenberg, M. S.; and Horton, C. 1994. Increasing the efficiency of emergency medical services by using criteria based dispatch. *Annals of Emergency Medicine* 24(5):867 – 872.

Gendreau, M.; Laporte, G.; and Semet, F. 2006. The maximal expected coverage relocation problem for emergency vehicles. *Journal of the Operational Research Society* 57(1):22–28.

Ghosh, S., and Varakantham, P. 2016. Strategic planning for setting up base stations in emergency medical systems. In *ICAPS*, 385–393.

Gnanasekaran, A. M.; Moshref-Javadi, M.; Zhong, H.; Moghaddam, M.; and Lee, S. 2013. Impact of patients priority and resource availability in ambulance dispatching. In *IIE Annual Conference. Proceedings*, 1727. Institute of Industrial and Systems Engineers (IISE).

Ibri, S.; Nourelfath, M.; and Drias, H. 2012. A multi-agent approach for integrated emergency vehicle dispatching and covering problem. *Engineering Applications of Artificial Intelligence* 25(3):554–565.

Kuisma, M.; Holmström, P.; Repo, J.; Määttä, T.; Nousila-Wiik, M.; and Boyd, J. 2004. Prehospital mortality in an ems system using medical priority dispatching: a community based cohort study. *Resuscitation* 61(3):297–302.

Maxwell, M. S.; Restrepo, M.; Henderson, S. G.; and Topaloglu, H. 2010. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing* 22(2):266–281.

Restrepo, M.; Henderson, S. G.; and Topaloglu, H. 2009. Erlang loss models for the static deployment of ambulances. *Health care management science* 12(1):67–79.

Saisubramanian, S.; Varakantham, P.; and Chuin, L. H. 2015. Risk based optimization for improving emergency medical systems. In *AAAI*.

Schmid, V. 2012. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research* 219(3):611–621.

Sudtachat, K., and Mayorga, M. E. 2013. A simulation model for dispatching emergency vehicles under multitiered response. In *IIE Annual Conference. Proceedings*, 3450. Institute of Industrial and Systems Engineers (IISE).

Yue, Y.; Marla, L.; and Krishnan, R. 2012. An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment. In *AAAI*.