Dispatch Guided Allocation Optimization for Effective Emergency Response

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

Effective emergency (medical, fire or criminal) response is crucial for improving safety and security in urban environments. Recent research in improving effectiveness of emergency management systems (EMSs) has utilized data-driven optimization models for efficient allocation of emergency response vehicles (ERVs) to base locations. However, these data-driven optimization models either ignore the dispatch strategy of ERVs (typically the nearest available ERV is dispatched to serve an incident) or employ myopic approaches (e.g., greedy approach based on marginal gain). This results in allocations that are not synchronised with the real evolution dynamics on the ground or can be improved significantly. To bridge this gap, we make the following contributions: (1) We first provide a novel exact optimization model for allocation of ERVs that incorporates the non-linear real-world dispatch strategy as linear constraints and ensures that optimization exactly imitates the real-world dynamics of EMS; (2) In order to improve scalability, we then provide two novel heuristic approaches to solve problems with large number of emergency incidents; and (3) Finally, using two real-world EMS data sets, we empirically demonstrate that our heuristic approaches provide significant improvement over the best known benchmark approach.

Introduction

Emergency Management Systems (EMSs) are an essential component of public safety and health-care services. The application domains for EMS range from emergency medical support to fire evacuation to crime management. In an EMS, a set of base stations are strategically placed throughout the city and a set of Emergency Response Vehicles, ERVs (e.g., ambulances, fire trucks, police vehicles) are stationed in each of those base stations. Upon arrival of an emergency request, the operators typically dispatch the nearest available ERV to assist the victim. The personnel on board the ERV provide initial treatment and move the victim to a neighbouring hospital. After transferring the patient over to hospital, ERV returns back to the base from where it was dispatched. EMS is an extremely sensitive and critical domain, as arriving at an incident location a few seconds early can save a human life. Therefore, EMS operators measure performance using metrics that are based on response time (i.e., the time taken by an ERV to reach the incident location after the request arrived into the system). One such widely adopted key performance metric is bounded time response (the percentage of requests served within a threshold response time), and will be the main focus in this paper.

Performance of an EMS can be improved either by optimizing allocations of ERVs to base stations or by improving strategies to dispatch ERVs to incidents. In this paper, we specifically focus on the optimization of ERV allocation. There are two types of data-driven approaches researchers have recently employed to improve ERV allocation: (a) Yue, Marla, and Krishnan (2012) proposed a greedy approach to incrementally allocate one ERV at a time using an event-driven simulator that follows the nearest available ERV dispatch strategy. They employ the bounded time objective where the goal is to maximize the number of incidents served within a fixed time. However, the solution of this greedy approach can be far away from the optimal due to its myopic nature (one request at a time) and as the bounded time response metric is not sub-modular; (b) On the other hand, Saisubramanian, Varakantham, and Chuin (2015) generated an allocation that minimizes response time with a bounded risk (i.e., percentage of incidents that can have response times higher than the objective) using a linear optimization model that follows an omniscient dispatch policy. They then evaluate the obtained solution using a simulator with the actual dispatch strategy to get the actual objective value. While the optimization approach with omniscient dispatch provides promising results, the use of omniscient dispatch is not realistic since the optimization occurs over a different evolution of emergency response dynamics than the one that happens on the ground.

Therefore, in order to reduce the response times for emergency requests, we need to consider both these operational (day-to-day) inefficiencies simultaneously: (a) allocation of all ERVs (and not one by one) to base stations; and (b) dispatch of the ”right” ERVs to the emergency requests. To tackle the above mentioned issues and to bridge the gap between the optimization model and the real-world scenarios, we provide a dispatch guided optimization approach for allocating all ERVs to base stations. We specifically consider the widely used bounded time objective employed by Yue *et al.* (Yue, Marla, and Krishnan 2012). To that end, our key

contributions are as follows:

- We provide a novel Integer Linear Programming (ILP) model for dynamic allocation of ERVs that incorporates the real-world ERV dispatch strategies as linear constraints. This allows for exactly imitating the real dynamics of EMS when optimizing the allocation.
- As the proposed ILP and its equivalent constraint programming (CP) models suffer with scalability issues when the number of emergency requests are increased, we provide two novel heuristic approaches to solve the problems with large number of incidents.
- By employing an event-driven simulation model based on two real-world EMS data sets, we empirically show that our proposed heuristic approaches can consistently and in some cases significantly improve the efficacy of EMS over the existing benchmark approach.

Related Work

Given the practical need for having efficient emergency response, a wide range of research papers have studied resource (e.g., base station) placement and ERV allocation problems in EMS. We focus on three relevant threads of research. The first thread of research focuses on resource placement for rare and large-scale disaster response (e.g., fire, natural disaster or catastrophic events), which enjoys a rich history. The conventional models for facility location problem in disaster response can be categorised into two groups: (a) location set covering problem [LSCP] (Toregas et al. 1971), where the goal is to cover all the demand points; and (b) maximal covering location problem [MCLP] (Church and Velle 1974), where the goal is to maximize the coverage of demand points within a given budget. In a survey, (Luis, Dolinskaya, and Smilowitz 2012) summarise the leading contributions in large-scale disaster response for strategic placement of ERV depots and allocation of ERVs to those depots. The recent research in designing decision support systems for disaster response (Jia, Ordóñez, and Dessouky 2007; Huang, Kim, and Menezes 2010) has employed mathematical optimization methods or dynamic programming. However, these solution approaches only consider the spatial distribution of demand locations and therefore, are not relevant for EMS. In EMS, the emergency incidents occur on a regular basis and spatio-temporal distribution of the demand points changes dynamically over time.

The second thread of papers focus on learning an efficient ERV dispatch strategy. (Andersson and Värbrand 2007; Schmid 2012; Ibri, Nourelfath, and Drias 2012; Bjarnason et al. 2009) develop techniques to learn a dispatch strategy while allocating ERVs. They also provide an ERV relocation model that suggests a destination location for ERVs in an online fashion. Although dispatch strategy plays an important role in improving EMS, the EMS operators typically follow the nearest available ERV dispatch strategy to avoid the inherent complexities such as uncertainties associated with future incident demand points, traffic congestion, and misinterpretation of incident criticality by the operators. Furthermore, continuous relocation of ERVs is inefficient both in terms of cost-effectiveness and utilisation of ERVs. Therefore, we employ a data-driven optimisation model to compute a fixed allocation strategy for the entire day and follow the typical nearest available ERV dispatch strategy.

The third thread of research focuses on allocation of ERVs. (Maxwell et al. 2010) propose an allocation and dynamic relocation model for single ERV. In contrast, we consider the allocation of an entire fleet of ERVs. (Brotcorne, Laporte, and Semet 2003; Gendreau, Laporte, and Semet 2006) propose mathematical optimization or local search based heuristics to solve the allocation problem, where the performance metrics are considered as the model parameter. However, these optimization models typically fail to capture the spatio-temporal dynamics of EMS (e.g., ERV travel times or response times for incidents change over time). To address these issues, recent research (Saisubramanian, Varakantham, and Chuin 2015; Yue, Marla, and Krishnan 2012; Restrepo, Henderson, and Topaloglu 2009; Ghosh and Varakantham 2016) has utilized data-driven optimization model or real-world event-driven simulator for ERV allocation and placement of base stations. However, these approaches either solve the problem greedily (add one ERV at a time) through simulation or ignore the ERV dispatch strategy in the optimization. In this paper, we propose a data-driven optimization approach by incorporating the real-world dispatch constraints for allocation of ERVs.

Motivation: Dynamic Allocation of ERVs

We now formally define the generic model for Dynamic ERV Allocation Problem (**DEAP**). We employ the following tuple to represent the *DEAP*:

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

$\mathcal{B}$ denotes the set of base stations and $\mathcal{A}$ represents a fleet of ERVs. $\mathcal{R}$ denotes a set of emergency requests for a particular weekday (e.g., incident logs of consecutive ten Mondays). Each request $r \in \mathcal{R}$ is a tuple $\langle t, s, h, \mathcal{B}_r, \mu_r, \lambda_r \rangle$. t, s, h denote the arrival time, source location and destination hospital for the particular request r. $\mathcal{B}_r$ represents a set of nearby base stations from which the request r can be served. μ_r provides the response times for each of the nearby bases in $\mathcal{B}_r$. Specifically, if $\mathcal{B}_r = \{l_1, l_2, ...\}$, then μ_r^i denotes the response time from base l_i. For the ease of representation, we assume that the nearby base set $\mathcal{B}_r$ is sorted according to the response times. That is to say, $\mu_r^i \leq \mu_r^{i+1}$. λ_r provides the round-about times (i.e., the total time required for an ERV to return back to the origin base after serving the request) for the nearby bases, where λ_r^i denotes the round-about time for base l_i. $\boldsymbol{T}$ provides the ERV travel time between any two locations. Specifically, T_{l_1,l_2} represents travel time for an ERV between source location l_1 to destination l_2. $\boldsymbol{C}$ denotes the capacities of the bases, where C_l represents the maximum number of ERVs the base $l \in \mathcal{B}$ can hold at a time. Finally, L represents the utility function, which is defined as follows:

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

Where, Δ denotes the threshold response time bound provided by the EMS operators. Intuitively, a reward of 1 unit is provided if a request is served within the threshold time. With the given *DEAP* input tuple, our objective is to find an efficient dynamic[1] allocation of an entire fleet of ERVs, $\mathcal{A}$ to a given set of base stations, $\mathcal{B}$ that maximizes the percentage of requests which can be served within the given threshold time bound, Δ. This is also referred to as the bounded time objective provided by (Yue, Marla, and Krishnan 2012).

Event-driven Simulation Model

We first present an event-driven simulation model (adopted from Yue, Marla, and Krishnan 2012), which is used to evaluate the performance of an allocation strategy. Algorithm (1) delineates the key functionalities of the event-driven simulator. Let us assume, ξ denotes a set of events where each event $e \in \xi$ represents an emergency request and the list is sorted according to the arrival time of incidents. Let we need to evaluate the performance of allocation strategy $\boldsymbol{A}$. Then, the set of available ERVs, I is initialized according to the allocation strategy $\boldsymbol{A}$. Let, a_r denotes the ERV which is assigned for request $r \in \mathcal{R}$, where each incident is initialized with a null assignment. In each iteration, the first element in the event list is popped. If the element is a new request r, then we assign the nearest available ERV, a_r for the request (a typical dispatch strategy followed by the real-world EMS operators) and that particular ERV is removed from the available ERV set I. In addition, we insert a job completion event in ξ at time $t_r(a_r)$, which denotes the time when the ERV, a_r will return back to the base after serving request r. On the contrary, if the popped event is a job completion event for request r, we add the ERV, a_r into the available ERV set I. This iterative process continues until the event list becomes empty. Once the simulation is over, we have a valid assignment for each request and therefore, we can compute the utility of the given allocation strategy, $\boldsymbol{A}$ in terms of the percentage of requests served with the given threshold response time bound, Δ.

Solution Approaches

We first propose an exact Integer Linear Programming (ILP) formulation for efficiently solving the *DEAP*. This exact formulation can also with minor modifications be converted to a Constraint Program (CP). However, as the two exact models do not scale to problems with large number of requests, we provide two novel heuristic approaches to improve scalability of our solutions[2].

[1]For dynamic allocation, the allocation strategy changes on every weekday. For instance, to generate the allocation strategy for a Monday, we consider $\mathcal{R}$ as the set of requests of past Mondays.

[2]Dynamic allocation of ERVs is a offline process for preparedness. As the allocation needs to be generated once in a day, our proposed approaches (which follow our imposed time-limit of two hours) are suitable for these scenarios. During the response phase, as the allocation and dispatch strategy are given as input, the operators can continuously dispatch the nearest available ERV for an incident by employing the simulator from Algorithm (1).

Algorithm 1: EDSimulator($\mathcal{R}, \mathcal{B}, \boldsymbol{A}$)

Initialize: $I \leftarrow \boldsymbol{A}$ // Initialize set of available ERV;
$\xi \leftarrow \mathcal{R}$ sorted in arrival order;
$\boldsymbol{a} = \{a_r | a_r \leftarrow \perp\}$ // Initialize as null assignment ;
repeat
  Pop next arriving event e from ξ;
  if e =*New Request* r **then**
    $a_r \leftarrow Dispatch(r, I)$ // Dispatch nearest free ERV;
    $I \leftarrow I - \{a_r\}$ // Update available 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 available ERV;
until $(|\xi| > 0)$;
return $\{\boldsymbol{a_r}\}$

Integer Linear Programming Formulation

We first provide a compact ILP formulation with novel dispatch related constraints to find an optimal allocation for a fleet of ERVs, $\mathcal{A}$ to the given set of bases $\mathcal{B}$. A request $r \in \mathcal{R}$ can be served from a feasible set of nearby bases, $\{\mathcal{B}_r \cup \perp\}$, where $\perp$ denotes a null assignment (i.e., the request cannot be served). Let, x_{rl_i} denotes a binary assignment variable, which is set to 1 if the request r is served from base $l_i \in \{\mathcal{B}_r \cup \perp\}$. Let, a_l is an integer variable which denotes the number of ERVs allocated to base $l \in \mathcal{B}$. a_l can be set to any value between 0 and the base capacity C_l. Our objective in the ILP is to find an efficient allocation that maximizes the utility function, L.

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r \in \mathcal{R}} \sum_{l_i \in \mathcal{B}_r} x_{rl_i} L_{rl_i}$$

To represent the evolution dynamics of EMS exactly, we now describe the constraints. Please note that the description of dispatch constraints is novel and the title of dispatch constraint description is highlighted in bold below:
A request can only be assigned to one base station: This set of constraints ensure that only one ERV from one of the feasible nearby bases is dispatched to assist an emergency incident. If all the nearby bases are empty when the request arrived into the system, the request is assigned to a dummy base $\perp$ and we label it as a null assignment.

$$\sum_{l_i \in \{\mathcal{B}_r \cup \perp\}} x_{rl_i} = 1, \qquad \forall r \in \mathcal{R}$$

A request can be served from a base if it has at least one ERV available: Let $P_r^{l_i}$ denotes the set of parent requests for r that are served from base l_i. More specifically, a requests r' belongs to the parent set $P_r^{l_i}$, if it has arrived in the system before request r and an ERV is still busy in serving r' when the request r has arrived if the ERV is assigned from base l_i for request r'. Therefore, these set of constraints enforce that if all the ERVs of a base l_i are busy in serving the parent

requests of r (i.e., $\sum_{j \in P_r^{l_i}} x_{jl_i} = a_{l_i}$), then the request r cannot be served from base l_i.

$$x_{rl_i} + \sum_{j \in P_r^{l_i}} x_{jl_i} \leq a_{l_i}, \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r$$

The entire fleet of ERVs has to be allocated: This constraint assures that each ERV is allocated to one of the base stations.

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

The nearest available ERV needs to be dispatched for assisting an emergency request: As mentioned previously, we assume that the set of nearby bases, $\mathcal{B}_r$ from which a request r can be served is sorted according their response times. So, the logical constraints (1) ensure that a request is always served from the nearest base with more than one idle ERV. Precisely, constraints (1) enforce that a request r must be assisted from a base $l_i \in \mathcal{B}_r$ where more than one ERV is present and all the other bases from which request r can be served faster are empty when the request has arrived.

$$\sum_{k \leq i} x_{rl_k} \geq 1 \qquad \textbf{if} \underbrace{\; a_{l_i} - \sum_{j \in P_r^{l_i}} x_{jl_i} \;}_{\text{\#ERV available at base } l_i} \geq 1 \qquad (1)$$

To linearise these constraints we introduce a binary variable b_{rl_i} which is set to 1 if more than one ERV is available in base $l_i \in \mathcal{B}_r$ when the request r has arrived.

$$b_{rl_i} = \begin{cases} 1 & \text{if } \sum_{j \in P_r^{l_i}} x_{jl_i} \leq a_{l_i} - 1 \\ 0 & \text{Otherwise} \end{cases}$$

The logical definition of these binary variables, $\boldsymbol{b}$ can easily be linearised using the following set of linear constraints, where C_{l_i} denotes the capacity of base l_i.

$$a_{l_i} - \sum_{j \in P_r^{l_i}} x_{jl_i} \leq C_{l_i} \cdot b_{rl_i} \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r$$

$$a_{l_i} - \sum_{j \in P_r^{l_i}} x_{jl_i} \geq C_{l_i} \cdot (b_{rl_i} - 1) + 1 \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r$$

We can now replace the logical and non-linear dispatch constraints (1) by using the following linear constraints.

$$\sum_{k \leq i} x_{rl_k} \geq b_{rl_i} \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r$$

An efficient alternative for the dispatch constraints: Let, $|\bar{\mathcal{B}}|$ denotes the average number of nearby bases for each of the incidents. To represent the dispatch policy according to the above mentioned approach, we need to introduce $|\mathcal{R}| \times |\bar{\mathcal{B}}|$ binary variables and $3 \times |\mathcal{R}| \times |\bar{\mathcal{B}}|$ linear constraints. Due to these large number of newly introduced binary variables and constraints, the prior approach for incorporating the dispatch constraints performs poorly. Therefore, in this section we provide a simplified and compact representation of the dispatch constraints (1). According to constraints (1), we just need to ensure that a request is served from a base with an idle ERV if other adjacent bases (from which the request can be served faster) are empty. As the assignment variables, $\boldsymbol{x}$ are binary, it would be adequate if we can ensure that the value of $\sum_{k \leq i} x_{rl_k}$ (i.e., sum of all the assignment variables for bases whose response times are less than or equals to the one for base l_i) is greater than zero and less than or equals to one. These conditions can be imposed using constraints (2), where C_{l_i} denotes the capacity of the base l_i (i.e., the maximum number of ERVs the base l_i can hold at a time). Specifically, we normalise the right-hand side value of constraints (2) to ensure that it is always bounded between 0 and 1. Note that, as C_{l_i} is a given input, the constraints (2) are linear in nature.

$$\sum_{k \leq i} x_{rl_k} \geq \frac{1}{C_{l_i}} \Big[\underbrace{\; a_{l_i} - \sum_{j \in P_r^{l_i}} x_{jl_i} \;}_{\text{\#ERV available at base } l_i} \Big] \; \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r \quad (2)$$

We show the entire ILP model for the ERV allocation problem compactly in Table (1). We refer to this approach as **ILP** in the later sections.

$$\max_{\boldsymbol{a}, \boldsymbol{x}} \sum_{r \in \mathcal{R}} \sum_{l_i \in \mathcal{B}_r} x_{rl_i} L_{rl_i} \qquad (3)$$

$$\textbf{s.t.} \sum_{l_i \in \{\mathcal{B}_r \cup \perp\}} x_{rl_i} = 1, \qquad \forall r \in \mathcal{R} \qquad (4)$$

$$x_{rl_i} + \sum_{j \in P_r^{l_i}} x_{jl_i} \leq a_{l_i}, \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r \quad (5)$$

$$\sum_{l \in \mathcal{B}} a_l = |\mathcal{A}| \qquad (6)$$

$$\sum_{k \leq i} x_{rl_k} \geq \frac{1}{C_{l_i}} \big[a_{l_i} - \sum_{j \in P_r^{l_i}} x_{jl_i} \big] \; \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r \quad (7)$$

$$a_l \in \{0, 1, ..., C_l\}, x_{rl_i} \in \{0, 1\} \qquad (8)$$

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

Constraint Programming

As the optimization model of Table (1) cannot be solved optimally with more than few hundred emergency requests using state-of-the-art black-box optimization solvers such as *CPLEX*, we now provide an alternative constraint programming (CP) model of Table (1). For every allocation variable, a_l we have created variable $\boldsymbol{allocation[l]}$ whose domain range is defined as $\{0, 1, ..., C_l\}$. Similarly, for the assignment variables, x_{rl} we created variable $\boldsymbol{assignment[r][l]}$ whose domain range is defined as $\{0, 1\}$. With these definition of variables, the equations (3)-(7) of Table (1) can be translated to CP. We refer to this approach as **CP**.

Continuous Assignment

Unfortunately, neither the ILP nor the CP model can be solved optimally within our threshold time-limit of 12 hours.

Therefore, we now provide an heuristic approach which can be solved within a minute with large number of emergency incidents. We essentially modify the ILP of Table (1) by relaxing the 0/1 dispatch variables to a probabilistic or continuous assignment. The revised optimization model is shown in Table (2), where we modified the assignments, $\boldsymbol{x}$ from discrete or binary to continuous variables. However, as the allocation variables, $\boldsymbol{a}$ remain integer, we are still allocating each ERV to exactly one base station. Therefore, the solution of the optimization problem will provide a valid ERV allocation, which can be executed on the event-driven simulator delineated in Algorithm (1) to obtain a valid and integral assignment for each request and to compute the actual utility of the allocation strategy. Although the objectives of the optimization problem and the simulation model might not be synchronised, we experimentally show that this approach provides better solution than the above mentioned exact approaches. This approach is referred to as **Relaxation**.

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r\in\mathcal{R}} \sum_{l_i\in\mathcal{B}_r} x_{rl_i} L_{rl_i}$$

s.t. Constraints (4), (5), (6) and (7) holds

$$a_l \in \{0, 1, ..., C_l\}, 0 \leq x_{rl_i} \leq 1 \tag{9}$$

Table 2: **FINDRELAXEDALLOCATION**$(\mathcal{R}, \mathcal{B}, \mathcal{A})$

Observation 1 *If all the base stations have single capacity (i.e., $a_l \in \{0, 1\}$), then the optimization model of Table* (2) *provides an optimal and integral solution.*

Proof: In case of single capacity base stations, the allocations $\boldsymbol{a}$ become binary variables. Therefore, when the first request r arrives in the system, constraints (7) enforce that the assignment variable x_{rl} is set to 1 if base l is the nearest base for request r and $a_l = 1$. If the nearest base is empty, then this logic is applicable for the second nearest base and so on. Henceforth, no request can be served from base l, until the ERV returns back to the base after serving the request. Due to this reasoning, the value of the right-hand side of constraints (7) can only be either 0 or 1. Hence, the assignment variables, $\boldsymbol{x}$ for all the requests can take either 0 or 1. Therefore, even with continuous assignment variables, $\boldsymbol{x}$ the optimization model of Table (2) provides an integral solution and is equivalent to our ILP model of Table (1). ■

Two-stage Optimization

In this section, we provide another heuristic approach to find an efficient ERV allocation. We propose a two stage hierarchical approach[3], where a preliminary allocation is generated for a subset of ERVs in the first stage and then we utilize that to guide the solution of the second stage for achieving allocation of the entire fleet of ERVs. In the first stage, we solve the ILP of Table (1) for the entire fleet of ERVs as a linear program (LP). That is to say, we relax both the allocation, $\boldsymbol{a}$ and assignment, $\boldsymbol{x}$ variables from integer to continuous one. The LP formulation for the first stage optimization problem is shown in Table (3).

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r\in\mathcal{R}} \sum_{l_i\in\mathcal{B}_r} x_{rl_i} L_{rl_i}$$

s.t. Constraints (4), (5), (6) and (7) holds

$$0 \leq a_l \leq C_l, 0 \leq x_{rl_i} \leq 1 \tag{10}$$

Table 3: **FINDLPALLOCATION**$(\mathcal{R}, \mathcal{B}, \mathcal{A})$

The LP solution provides a sense of best possible fractional allocation and therefore, we utilize this solution to compute the final integral and feasible solution in the second stage. Let $\hat{\boldsymbol{a}}$ denote the allocation obtained from the LP solution of Table (3). We use the following rounding approach to obtain an initial integral allocation, $\bar{\boldsymbol{a}}$ from $\hat{\boldsymbol{a}}$.

$$\bar{a_l} = \begin{cases} \lceil \hat{a_l} \rceil & \text{if } \hat{a_l} - \lfloor \hat{a_l} \rfloor \geq 0.95 \\ \lfloor \hat{a_l} \rfloor & \text{Otherwise} \end{cases}$$

$\bar{\boldsymbol{a}}$ provides a valid allocation for a subset of ERVs. As all the ERVs are homogeneous, it does not matter which specific subset of ERVs are allocated a priori. However, $\bar{\boldsymbol{a}}$ does not allocate the entire fleet of ERVs. We then utilize the values of $\bar{\boldsymbol{a}}$ to guide the original ILP of Table (1). In the second stage, we essentially solve the ILP of Table (1) with additional set of constraints (11), which enforce that at least $\bar{a_l}$ ERVs need to be allocated in base $l \in \mathcal{B}$. The second stage optimization model is shown compactly in Table (4). Note that, although the entire fleet of ERVs are employed in the optimization model of Table (4), this problem is less computationally challenging than the one of Table (1), because we manually fix the allocation for a subset of ERVs using constraints (11). Therefore, the optimizer needs to search for an allocation of only $|\mathcal{A}| - |\bar{\boldsymbol{a}}|$ ERVs. This approach is referred to as **TwoStage**.

$$\max_{\boldsymbol{a},\boldsymbol{x}} \sum_{r\in\mathcal{R}} \sum_{l_i\in\mathcal{B}_r} x_{rl_i} L_{rl_i}$$

s.t. Constraints (4), (5), (6), (7) and (8) holds

$$a_l \geq \bar{a_l} \qquad \forall l \in \mathcal{B} \tag{11}$$

Table 4: **FINDTWOSTAGEALLOCATION**$(\mathcal{R}, \mathcal{B}, \mathcal{A})$

Experimental Results

We conduct experiments[4] on two real-world data sets. We obtain the *dataset-1* from a real-world EMS in the form of anonymous and modified sample of request logs. The

[3]Note that, our two-stage optimisation is a single-shot (i.e., non-iterative) hierarchical approach.

[4]All the optimization models are solved using IBM ILOG Optimization Studio V12.5. incorporated within python code.

	Greedy	ILP	CP	Relaxation	TwoStage
Mon	58.97 %	57.96 %	57.56 %	60.10 %	60.46 %
Tue	59.15 %	44.95 %	56.12 %	60.76 %	60.16 %
Wed	59.27 %	47.51 %	57.43 %	61.76 %	62.52 %
Thu	60.27 %	59.32 %	57.96 %	62.86 %	62.42 %
Fri	59.87 %	59.81 %	52.83 %	61.68 %	61.92 %
Sat	63.65 %	63.16 %	63.47 %	66.69 %	66.80 %
Sun	65.76 %	67.44 %	67.05 %	70.06 %	69.46 %

Table 5: Performance (percentage of requests served within 8 minutes) comparison on testing data of *dataset-1*.

dataset-2 is adopted from Yue, Marla, and Krishnan (2012)[5]. Both the data sets provide details of emergency requests over a certain period. Each request log contains the following information (a) Incident location; (b) Arrival time; (c) A set of feasible nearby bases from where the request can be assisted; (d) Response time from each of the feasible base to incident location; and (e) Round-about time for each of the feasible base. While these specific details might not always be readily available for real deployment, as indicated in Ghosh and Varakantham (2016), we can estimate them using a straightforward method. As the geographical locations of the requests, hospitals and bases are available in the historical data sets, we can compute the set of feasible nearby bases and predict the response and round-about times for each of these bases.

Empirical results on dataset-1

The *dataset-1* contains a fleet of 35 ERVs and 35 base stations. We have an anonymous request sample over a period of six months. We divide our 6 months of data set into two parts - first 3 months is used for training purpose to generate the allocation strategies and the performance of these strategies are tested on other 3 months of data. We evaluate the performance of our approach by employing a real-life event-driven simulation model (refer to Algorithm 1 for the details of simulator) which follows the nearest available ERV dispatch rule. We compare our approach against the existing greedy approach which was introduced by Yue, Marla, and Krishnan (2012). The greedy approach incrementally adds one ERV in each iteration using the event-driven simulator until the entire fleet is allocated. We refer to this approach as **Greedy**. We do not provide comparison results against the approach proposed by Saisubramanian *et. al.* (Saisubramanian, Varakantham, and Chuin 2015), as their objective is to miminize response time for a fixed percentile of requests, which is different from the metric of interest in our paper (i.e., maximizing the number of requests served within a threshold time bound). Due to different objective functions, we experimentally observe that the approach from Saisubramanian *et. al.* (2015) produced worse quality solutions than the greedy approach proposed by Yue *et. al.* (2012).

Solution quality of the heuristic approaches: The ILP or CP cannot solve the large-scale problems optimally within our imposed time-limit of two hours. However, these exact approaches can be solved optimally for very small problems with only a few hundred requests. We experimentally observe that the our heuristic approaches provide good quality solutions in comparison to the optimal for these small instances. For instance, our two-stage optimization approach is only 1.5% away from the optimal. However, in our problem instances, we have a few thousands training incident requests. So, we can only get a sense of ILP optimum from the optimality gap provided by black-box solvers such as CPLEX. Unfortunately, these gaps are loose and are far away from the optimal solution (specifically for the ILP) and hence are unreliable.

Performance comparison: We now demonstrate the performance comparison between our approaches and the greedy approach on testing data of *dataset-1*. Table (5) shows the comparison results for all the weekdays. Our key performance metric is the percentage of requests that are served within 8 minutes. We observe that the solution quality of CP and ILP is worse than the existing greedy approach. This is so because we impose a time-limit of two hours for both the approaches and none of these approaches can be solved optimally within our time-limit. On an average, the optimality gap for ILP was more than 20%. However, both our heuristic approaches (i.e., two-stage optimization and relaxation approach) outperform the greedy approach. On an average, both these heuristics can serve around 63.4% requests within 8 minutes. Most importantly, for all the weekdays, our heuristic approaches serve around 2.4% additional requests within the threshold time bound (i.e., 8 minutes) over the existing greedy approach.

Effect of ERV fleet size: In this thread of results, we demonstrate the performance comparison between different approaches on *dataset-1* by varying the ERV fleet size. We use the same training and testing data set for these experiments. Figure 1(a)-(b) depict the performance comparison on training and testing data set for one of the weekdays. In the X-axis we vary the number of ERVs and Y-axis shows the percentage of requests served within 8 minutes. Due to the scalability issue, ILP and CP yield poor quality solution than our heuristic approaches for all the settings of ERV fleet size. Both the two-stage optimization and relaxation approach always outperforms the greedy approach. More interestingly, the gain over the greedy approach increases if we

[5]http://projects.yisongyue.com/ambulance_allocation/

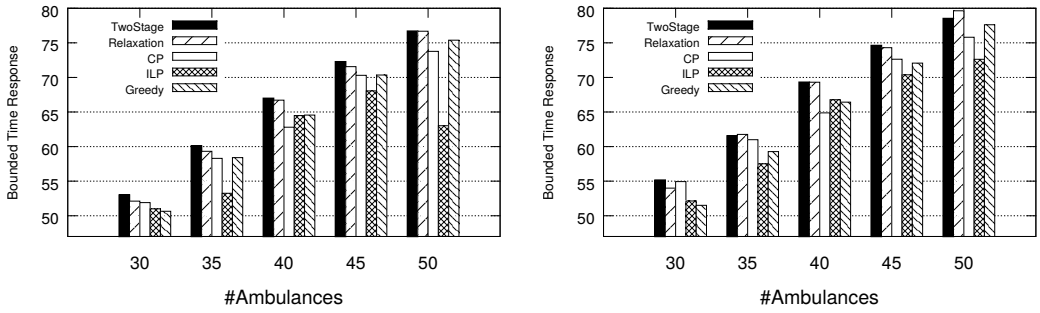

Figure 1: Performance comparison by varying ERV fleet size: (a) Training results on weekday; (b) Testing result on weekday.

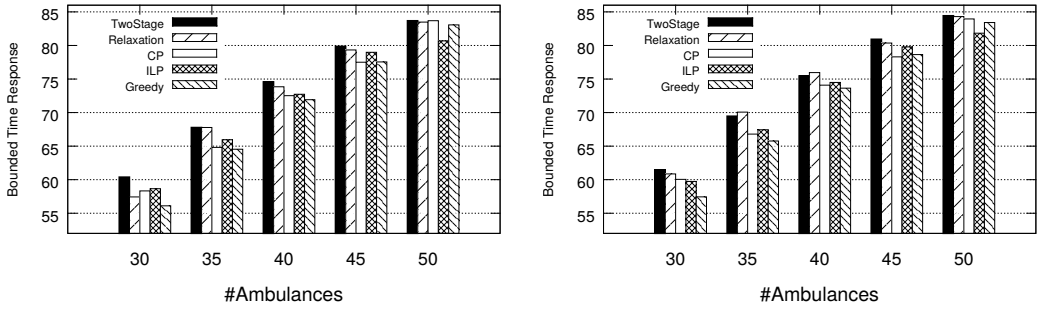

Figure 2: Performance comparison by varying ERV fleet size: (a) Training result on weekend; and (b) Testing result on weekend.

decrease the number of ERVs. This insight clearly indicates that the performance of the existing greedy approach degrades for EMS with limited resources and our approaches are suitable to tackle such scenarios.

Figure 2(a)-(b) demonstrate the performance comparison on training and testing data set for one of the weekends. We observe a similar pattern for these results. Our heuristic approaches always produce better solution than the greedy approach, specially when we have fewer ERVs. For instance, the performance gain of our two-stage optimization approach over the greedy approach on testing data set increases from 0.6% to 4.2% when the ERV fleet size is reduced from 50 to 30.

Empirical results on dataset-2

The *dataset-2* contains a fleet of 58 ERVs and 58 base stations. We have 1500 weeks of request logs which are generated using a memory-less stochastic process (i.e., poisson distribution). The parameters of this stochastic process are learnt from historical data[6]. We use Sample Average Approximation [SAA] (Verweij et al. 2003) for validation and performance estimation. We generate 10 policies for each of the weekdays, where each policy is generated using request logs of that particular weekday for 10 consecutive weeks (e.g., the second policy for Monday is generated using requests of all the Mondays from week 11 to week 20). Then we identify the policy with best validation performance for each of the weekdays separately over 500 weeks of request logs. Finally, we evaluate the performance of the validated policies on 3 testing data sets, each of which contains 300 weeks of request logs.

We now present the performance comparison results on the testing data of *dataset-2*. Figure 3 depicts the comparison on three testing data sets. The X-axis denotes the weekdays and the Y-axis represents the percentage of requests served within 15 minutes. As each of the testing data set involves 300 weeks of request logs, we report the average utility using SAA. Figure 3(a) plots the bounded time response value for the first testing data set. As shown clearly, our two-stage optimization almost always provides the best performance over other approaches, while our relaxation approach is proven to be highly competitive with two-stage optimization approach. Although, none of the CP and ILP can be solved optimally, CP provides a reasonably better quality solution than ILP within the time-limit of two hours. For all the weekdays, our two-stage optimization approach provides at least 1.5% gain in bounded time response value over the greedy approach.

Figure 3(b) and 3(c) depict the performance comparison results on second and third testing data sets. We observe a consistent pattern on the performance over all the three testing data sets. As shown in Figure 3(b), both the two-stage and relaxation heuristics are able to serve 1.6% extra requests within the threshold response time over the greedy approach for second testing data set. Figure 3(c) demonstrates that, for third testing data set, both the two-stage

[6]For dataset-2, refer to Section 3.1 and Section 7.1 of Yue *et. al.* (Yue, Marla, and Krishnan 2012) for the details of request sampling process and for the general settings of Sample Average Approximation (SAA), respectively.

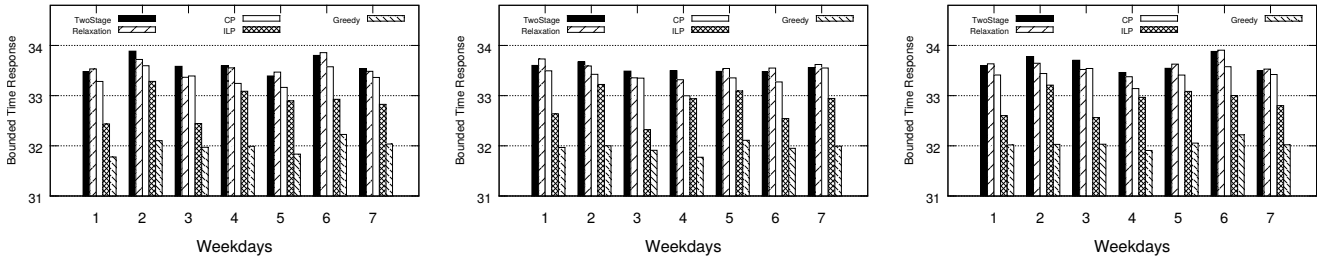

Figure 3: Performance (percentage of requests served within 15 minutes) comparison on *dataset-2*: (a) First testing set; (b) Second testing set; and (c) Third testing set.

and relaxation heuristics provide at least 1.45% performance gain on all the weekdays and improve the average bounded time response value by 1.6% over the greedy approach.

Discussion

We now discuss about directions that we explored in addition to the approaches described above. We believe that these approaches have potential and can be improved in the future.

Benders decomposition: By exploiting observation (1), we can ensure that continuous assignment guarantees to provide an optimal and integral solution for our original problem if all the bases have single capacity. A straightforward method to translate our problem into single capacity base station problem is to create C_l single capacity bases at the location of base l. The response and round-about times for a request r from all the C_l bases will be same as the response and round-about time for base l. The number of feasible nearby bases for a request r will now increase from $|\mathcal{B}_r|$ to $\sum_{l \in \mathcal{B}_r} C_l$. Once we have a continuous assignment problem with single capacity bases, it will be an ideal ground for applying Benders decomposition (Benders 1962), where master solves the allocation problem and slave takes the assignment decisions. Our initial experiments show that due to significant increases in the number of variables and constraints, this particular translation results in a large optimality gap even with Benders decomposition. However, these insights lead to a promising direction for improving our solutions in the future.

SAT representation: The reformulated single capacity base problem is a 0/1 integer program and can be translated to a satisfiability problem. As we have an optimization problem, we can translate it to a partial max-SAT (Argelich and Manya 2007; Koshimura et al. 2012) representation, where our objective function can be converted to following soft clauses (12):

$$\boldsymbol{x}_{rl}, \quad \forall r \in \mathcal{R}, \exists l, T_{r.s,l} \leq \Delta \tag{12}$$

Constraints (4) can be translated to a set of hard clauses (13). The clauses (14)-(15) are equivalent to constraints (5). The clauses (16) exactly represent the constraints (7).

$$\neg \boldsymbol{x}_{rl_i} \vee \neg \boldsymbol{x}_{rl_j} \qquad \forall r \in \mathcal{R}, l_i, l_j \neq l_i \in \mathcal{B}_r \tag{13}$$

$$\boldsymbol{a}_{l_i} \vee \neg \boldsymbol{x}_{rl_i} \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r \tag{14}$$

$$\neg \boldsymbol{x}_{rl_i} \vee \neg \boldsymbol{x}_{jl_i} \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r, j \in P_r^{l_i} \tag{15}$$

$$\neg \boldsymbol{a}_{l_i} \vee_{k \leq i} \boldsymbol{x}_{rl_k} \vee_{j \in P_r^{l_i}} \boldsymbol{x}_{jl_i} \qquad \forall r \in \mathcal{R}, l_i \in \mathcal{B}_r \tag{16}$$

However, to the best of our knowledge, there is no explicit way to represent the constraints (6) as SAT clauses. A brute-force approach would be employing the following set of hard clauses (17)-(19), where d_{cl} variable is set to 1 if the ERV c is allocated to base l. However, this brute-force approach increases the number of variables and clauses significantly and therefore, state-of-the-art partial max-SAT solvers fail to solve it efficiently. So, discovering an efficient and compact SAT representation would be a potential future direction.

$$\neg \boldsymbol{d}_{cl_i} \vee \neg \boldsymbol{d}_{cl_j} \qquad \forall c \in \mathcal{A}, l_i, l_j \neq l_i \in \mathcal{B} \tag{17}$$

$$\neg \boldsymbol{d}_{c_i l} \vee \neg \boldsymbol{d}_{c_j l} \qquad \forall c_i, c_j \neq c_i \in \mathcal{A}, l \in \mathcal{B} \tag{18}$$

$$\neg \boldsymbol{a}_l \vee_c \boldsymbol{d}_{cl} \qquad \forall l \in \mathcal{B} \tag{19}$$

Conclusion

In this paper, we provide dispatch guided optimization approaches for effective and dynamic allocation of ERVs to base locations. We propose a novel optimization model by incorporating the real-world ERV dispatch strategy and show that the optimization model follows the real evolution dynamics of EMS. As the proposed optimization model suffers scalability issues, we provide two novel heuristic approaches to increase scalability to large number of emergency incidents. The empirical results on two real-world EMS data sets demonstrate that our heuristic approaches always outperform the existing best known approach and therefore, improve the efficiency of EMS. In future, this work can be extended in the following two directions: (a) Improve the scalability of the solutions either by translation to an efficient partial Max-SAT or through well-known decomposition approaches; and (b) Extend our solutions for EMS that involves multi-tiered ERVs and multi-priority incidents, where dispatch strategy plays a crucial role.

Acknowledgments

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.

Argelich, J., and Manya, F. 2007. Partial max-sat solvers with clause learning. *SAT* 2007:28–40.

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik* 4(1):238–252.

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.

Church, R., and Velle, C. R. 1974. The maximal covering location problem. *Papers in regional science* 32(1):101–118.

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.

Huang, R.; Kim, S.; and Menezes, M. B. 2010. Facility location for large-scale emergencies. *Annals of Operations Research* 181(1):271–286.

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.

Jia, H.; Ordóñez, F.; and Dessouky, M. 2007. A modeling framework for facility location of medical services for large-scale emergencies. *IIE transactions* 39(1):41–55.

Koshimura, M.; Zhang, T.; Fujita, H.; and Hasegawa, R. 2012. Qmaxsat: A partial max-sat solver. *Journal on Satisfiability, Boolean Modeling and Computation* 8:95–100.

Luis, E.; Dolinskaya, I. S.; and Smilowitz, K. R. 2012. Disaster relief routing: Integrating research and practice. *Socio-economic planning sciences* 46(1):88–97.

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.

Toregas, C.; Swain, R.; ReVelle, C.; and Bergman, L. 1971. The location of emergency service facilities. *Operations Research* 19(6):1363–1373.

Verweij, B.; Ahmed, S.; Kleywegt, A. J.; Nemhauser, G.; and Shapiro, A. 2003. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications* 24(2-3):289–333.

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