Online Supplement for "Emergency Medical Service Resource Allocation in a Mass Casualty Incident by Integrating Patient Prioritization and Hospital Selection Problems"

by Kyohong Shin and Taesik Lee

A Simulation

We construct a discrete event simulation of the operation of ambulances and EDs in an MCI. This simulation is used to obtain the state trajectories of the system required in the ADP algorithm and to evaluate the performance of the policy obtained from the ADP and other heuristic policies tested in this work.

The simulation starts with N_A ambulances arriving at the accident site. An ambulance transports a patient to a destination hospital as instructed by a chosen decision logic (ADP policy or heuristic policy). After bringing a patient to a hospital, the ambulance returns to the accident site for its next transport task. If the hospital is currently busy, the patient joins its queue and waits until treatment becomes available. Once his/her treatment starts, the survival probability of the patient is recorded. These processes are repeated until all patients are transported and begin treatment at the hospitals.

A discrete event simulation proceeds by sequentially executing events in an event list. Events registered in the event list are sorted in order of event start time, and the simulation executes the first event in the list. The simulation clock is advanced to the start time of the current event, and the system state is updated according to the results of the event execution. Execution of an event may generate another event, which will be inserted into the event list at an appropriate slot. After executing an event, we repeat the process and execute the next event in the event list. The simulation runs continuously until the event list is empty.

Figure A.1 shows a flowchart of the simulation model. First, the simulation clock and system states are initialized. Then, we construct an initial event list by inserting N_A *Ambulance arrival (at the accident site) event*s with an event start time of 0, where N_A is the number of ambulances. The initialization phase creates an environment in which ambulances are ready to transport patients.

There are three types of events: *Ambulance arrival at the site*, *Ambulance arrival at hospital j*, and *Treatment completion*. When executed, each event changes the relevant system state and possibly generates another event to update the event list. This is shown in the square boxes in the middle of Figure A.1.

For the *Ambulance arrival at the site* event, the simulation first identifies the number of patients at the site from the system state information. If there are patients waiting at the site, a decision is made in regard to which class of patients will be transported and to which hospital. This decision is made using the policy solution applied in the simulation. The system state is updated accordingly; for example, the number of patients in the selected class decreases by one, the ambulance status changes to 'transporting,' and so on. Finally, a new event, *Ambulance arrival at hospital j*, is generated. The start time of this event is determined by the probability distribution of its travel time, and the event is added to the event list in the proper order. If there are no more patients at the site, nothing happens.

The *Ambulance arrival at hospital j* event updates the system state as follows. The number of patients (in the class of the transported patient) in hospital j increases by one, the ambulance status changes to 'returning,' and other relevant state information is updated. Then, a new event, *Ambulance arrival at the site*, is added to the event list, with its start time set according to the travel time distribution. If there are no patients in hospital j (i.e., hospital j is empty), another new event, *Treatment completion*, is created. The start time of this event is determined by the probability distribution of the service time of hospital j.

In the case of the *Treatment completion* event, the number of patients in the hospital is reduced by one. When there is a patient waiting in the queue, the *Treatment completion* event is generated and inserted into the event list and its start time is given by the service time distribution.

B ADP Algorithm Development

To solve the problem considered in this paper, Equation (2) is computed for every possible $s \in S$ until the value of each state converges to its optimal value. For our problem, the state space is too large, rendering the iterative process computationally prohibitive. ADP is a powerful tool for managing the curse of dimensionality. Unlike DP, which computes values for all possible states, ADP randomly generates sample paths and only computes values for the states on the generated paths by stepping forward from the initial state. This is done by solving the following

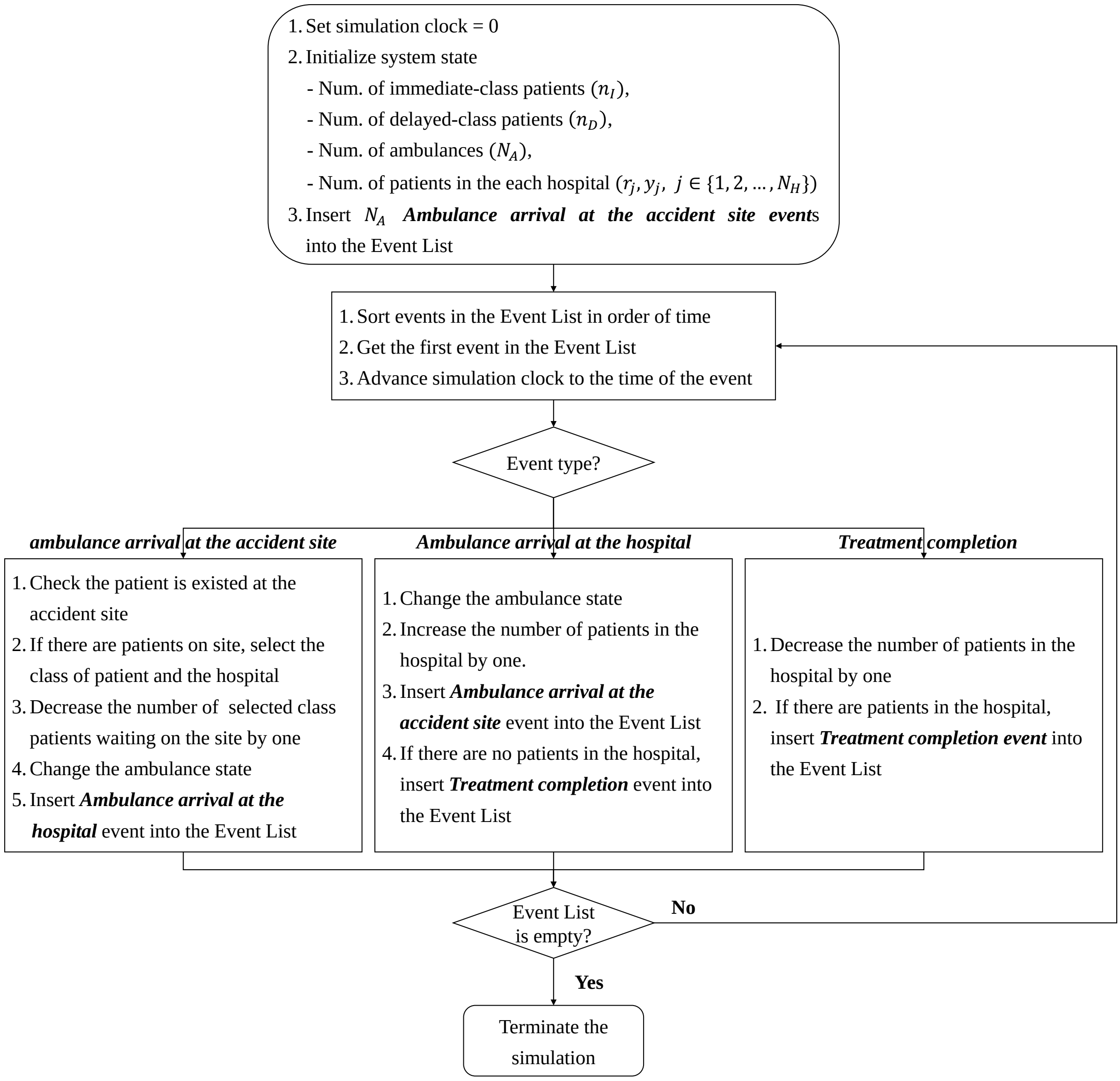

Figure A.1: Simulation flow chart

equation at each iteration:

$$\hat{v}^n = \max_{x \in X}[R(s^n, x^n) + E\{\bar{V}^{n-1}(s'^n)|s^n, x^n\}], \tag{B.1}$$

where n denotes the current iteration number, $\hat{v}^n$ is the sample estimate of the value of the current state s^n, and $\bar{V}^{n-1}(s'^n)$ is the approximate value of state s'^n updated at $(n-1)$. Equation (B.1) is the same as Equation (2), except that an approximate value $\bar{V}^{n-1}(s'^n)$ is used for the next state s'. The value of the current state s^n is then approximated by the weighted sum of the sample estimate obtained from this iteration ($\hat{v}^n$) and its approximate value on $(n-1)^{th}$ iteration, $\bar{V}^{n-1}(s'^n)$, as follows:

$$\bar{V}^n(s) = (1 - \alpha_n)\bar{V}^{n-1}(s) + \alpha_n \hat{v}^n. \tag{B.2}$$

In Equation (B.2), the stepsize α_n determines the weight of the sample estimate obtained in this iteration when calculating the approximate value of the current state.

To solve equations (B.1) and (B.2), we construct a simulation-based ADP algorithm. The simulation model (described in Section A) performs two functions in the algorithm. First, it generates a random sample path in the state space. The algorithm computes estimated values while stepping through the states chosen in the sample path. Second, the simulation model estimates the expectation $E\{\bar{V}^{n-1}(s'^n)|s^n, x^n\}$ required to solve Equation (B.1). The overall process is described in **Algorithm 1**.

In the first iteration cycle ($n = 1$), the algorithm computes the estimate $\hat{v}^n$ and decides action x^n of the initial state (lines 12 and 15). In lines 12, 13, and 16, $E\{\bar{V}^{n-1}(s'^n)|s_0^n, x^n\}$ must be computed. As we do not have explicit knowledge of the transition probabilities $p(s, x, s')$, the simulation model is used to estimate the expectation. Specifically, we evaluate the expectation by running the simulation model N_{sim} times for each possible action $x \in X(s_0^n)$ until the current state transitions to the next state (line 10). The value of the initial state is then updated to compute $\bar{V}^n(s_0^n)$ (line 18). The simulation model is now executed to progress to the next decision epoch, and returns the next state whose value is going to be updated (line 19). This process is repeated until the first sample path is completed (i.e., all patients are being treated in hospitals), and the values for the states included in the sample path are updated. The next iteration then proceeds with a new sample path. The overall process is repeated for N iterations.

Algorithm 1 Simulation-based ADP

1: $\bar{V}^0(s) \leftarrow 0 \quad \forall s$

2: Set the maximum iteration number N

3: *Choose an Initial state* s_0^1

4: **for** $n = 1$ to N **do**

5: Initialize the discrete event simulation configuration

6: $TPH \leftarrow$ *total patients in hospitals*

7: $PA \leftarrow$ *patients in ambulances*

8: **while** $n_I > 0$ or $n_D > 0$ or $TPH > 0$ or $PA > 0$ **do**

9: $ran \leftarrow rand()$

10: Compute $E\{\bar{V}^{n-1}(s'^n)|s^n, x\}$ by running the simulation model N_{sim} times for each possible action $x \in X(s^n)$ through the next decision epoch

11: **if** $ran \geq e^{-\varepsilon \times n}$ **then**

12: $x^n \leftarrow \arg\max_{x \in X(s)}[R(s^n, x^n) + E\{\bar{V}^{n-1}(s'^n)|s^n, x^n\}] \quad (exploitation)$

13: $\hat{v}^n \leftarrow \max_{x \in X(s)}[R(s^n, x^n) + E\{\bar{V}^{n-1}(s'^n)|s^n, x^n\}]$

14: **else**

15: $x^n \leftarrow$ randomly selected $x \in X(s) \quad (exploration)$

16: $\hat{v}^n \leftarrow R(s^n, x^n) + E\{\bar{V}^{n-1}(s'^n)|s^n, x^n)\}$

17: **end if**

18: $\bar{V}^n(s^n) \leftarrow (1 - \alpha_n(s^n))\bar{V}^{n-1}(s^n) + \alpha_n(s^n)\hat{v}^n$

19: $s^n \leftarrow S^M(s^n, x^n, w(s^n, x^n))$

20: $TPH \leftarrow$ total patients in hospitals

21: $PA \leftarrow$ patients in ambulances

22: **end while**

23: **end for**

B.1 Exploitation vs Exploration

At the start of **Algorithm 1**, the values of all states are initialized to zero. The values of the states that happen to be included in the sample path are updated to nonzero values, while the values of the other states remain at zero. This causes a bias in the choice of actions (line 13 of **Algorithm 1**), because actions that drive the next transition toward nonzero-valued states are preferred. When this happens early in the iterative process, the algorithm only updates a small portion of the state space, leaving most of the states with zero values. Clearly, this will lead to a poor solution. To prevent this problem, we need to implement a mechanism that intentionally chooses actions that ensure as many states as possible are visited and their values updated. This is particularly important during the early stages of the iteration. As these actions may not necessarily maximize the sample estimate $\hat{v}^n$, we need to use an *exploration* mechanism that allows an action to be selected even if it does not maximize $\hat{v}^n$. This is implemented in line 15 of **Algorithm 1**, where we randomly choose an action x^n.

If this biased selection of actions occurs during the later stages of the iteration, it works to the advantage of the algorithm. Once a reasonably large number of states have been updated, we want the algorithm to focus on the portion of the state space whose values are relatively high. This *exploitation* increases the accuracy of the values of those states, improving the quality of the approximation.

To induce early exploration and later exploitation in the iterative process, we use the Epsilon-Greedy Exploration technique (Schmid, 2012), setting an exploration rate $e^{-\varepsilon \times n}$ as a threshold to choose between exploration and exploitation. The initial exploration rate is high, but decreases as the number of iterations increases. The parameter ε in the exploration rate is known to affect the overall performance of the ADP algorithm.

B.2 Stepsize

In line 18 of **Algorithm 1**, the parameter α_n is used to control the weight of the sample estimate obtained from this iteration, $\hat{v}^n$, in the approximation of the state value. A common rule is that, if there is little knowledge of state s, i.e., s has been visited few times, then a high weight is assigned to the newly obtained sample estimate $\hat{v}^n$, and vice versa. This practice is often implemented through the harmonic stepsize rule (Powell, 2007; Fang et al., 2013):

$$\alpha_n = \frac{c}{c + n' - 1}, \tag{B.3}$$

where c is a constant and n' is the number of visits to state s. The larger the constant c in Equation (B.3), the slower the rate at which α_n falls to zero.

B.3 Aggregation

Although an exploration strategy has been adopted in the algorithm, the large state space in our problem makes it possible that many states will never be visited during the iterative process. Thus, we need to develop a more aggressive approach to approximate the values of these states. We choose the aggregation method, which is a value function approximation technique. The aggregation method groups states that are considered to be similar–if the value of one state in the group is updated by the algorithm, the states of other members of the group are also updated to have the same value (Powell, 2007). We use τ_j, the start time of the treatment for the current patient, as the criterion for grouping the states. Specifically, given an aggregation level L, we aggregate the care start time τ_j for the current patient in the following way: $[0, L), [L, 2L), ...$ For example, when $L = 5$, we aggregate the states which have $\tau_j \in [0, 5)$ and have the same values for the rest of the state elements.

Based on preliminary experiments, we set these parameters to $\varepsilon = 0.001, c = 5, L = 5$.

C Sensitivity Analysis for Delta

A comment on δ_j is in order. In our problem setting, 'tier-1' denotes hospitals with a lower care capability, where limited facilities or care provider skill levels lead to a less than the desired level of care for class-I patients. To differentiate the care capabilities between tier-1 and tier-2 hospitals, we use the adjustment factor δ_j ($\delta_j < 1.0$) when hospital j is a tier-1 hospital. Certainly, δ_j influences the reward function, thereby affecting the derived policy solution. One may wonder whether a high value of δ_j would change this observation; for example, $\delta_j = 1.0$ would mean that there is no difference in survival probability outcomes between tier-1 and tier-2 hospitals. However, it should be emphasized that the reward function value is not only determined by δ_j, but also by the waiting time at the hospital. These two factors collectively determine the reward function value based on which the optimal policy is computed. Our experiments, in which δ_j changed from 0.2 to 1.0, show that even for larger values of δ_j – tier-1 and tier-2 hospitals have similar care capabilities – the hospital selection decision plays an important role in the overall success of EMS operations (see Table C.1) because the survival probability depends on the waiting time at the destination hospital; choosing a destination hospital that ensures timely care provision is still important.

Table C.1: Average number of survivors of three decision rules for each value of δ_j

δ_j	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
① (optimal, optimal)	6.430	6.401	6.462	6.461	6.471	6.48	6.506	0.493	6.509
② (simple, optimal)	6.077	6.059	6.077	6.043	6.066	6.091	6.091	6.057	6.101
③ (optimal, simple)	5.947	5.987	5.95	5.95	5.965	5.968	5.968	5.973	5.987
① - ③ Effect of optimal hospital selection	0.483	0.414	0.512	0.511	0.506	0.512	0.528	0.52	0.522
① - ② Effect of optimal transport priority	0.353	0.342	0.385	0.418	0.405	0.389	0.415	0.436	0.408

D D. Sensitivity Analysis for the Service Time Ratio Between Tier-1 and Tier-2

As mentioned in Section 3.3.1, the mean service time of a hospital is inversely proportional to the total number of medical staff in the hospital. The personnel requirement set by the Ministry of Health and Welfare of South Korea states that the number of medical staff in tier-2 hospitals is 1.5 times higher than the number of medical staff in tier-1 hospitals. Thus, the service time of the tier-1 hospital is set to be 1.5 times longer than that of the tier-2 hospital.

To examine how this factor (service time ratios between tier-1 and tier-2, ρ) may change the experimental outcomes, we conduct sensitivity analysis with respect to the service time difference of two hospitals. Using the values from the experiments described above as a reference ($\rho = 1.5$), we vary ρ from 1, 1.25, 1.75, to 2. Table D.1 show that (optimal, optimal) is the best policy and also that the effect of optimizing each decision increases as ρ increases. This means that as the service time ratio between tier-1 and tier-2, ρ, becomes larger, the effect of sending to hospitals with potentially fewer patients waiting becomes more important. In addition, as ρ increases, the service time of the tier-1 hospital becomes longer and completing the treatment is delayed. Similar to the sensitivity analysis for service time, it may be better to transport class-D patients than class-I patients with a significant decrease in survival over time. Therefore, as ρ increases, the effect of prioritizing patient transport also becomes more important.

Table D.1: Average numbers of survivors of three decision rules for service time ratios between tier-1 and tier-2

ρ	1	1.25	1.5	1.75	2
① (optimal, optimal)	6.495	6.45	6.430	6.412	6.395
② (simple, optimal)	6.284	6.141	6.077	6.052	5.962
③ (optimal, simple)	6.299	6.141	5.947	5.830	5.810
① - ③ Effect of optimal hospital selection	0.196	0.310	0.483	0.582	0.585
① - ② Effect of optimal transport priority	0.211	0.309	0.353	0.360	0.433

E Transition Probabilities for the Reduced Problem

The reduced problem considers the accident environment in which patients are transported to two hospitals by one ambulance.

Problem description. There are two classes of patients, class-I and class-D, denoted by n_I and n_D, respectively. These patients have a limited survival time within which they must receive care at a hospital. The survival time for the patients in each class follows an exponential distribution with mean survival times of $\frac{1}{r_I}$ and $\frac{1}{r_D}$, respectively, where r_i denotes the abandonment rate. We assume that $\frac{1}{r_I} < \frac{1}{r_D}$.

Two hospitals, tier-1 and tier-2, are accessible from the accident site. The tier-j hospital is located at an average distance of $\frac{1}{\mu_j}$ minutes by ambulance. The tier-2 hospital has a higher capability in that it can treat patients faster and with a better survival outcome. This is represented by a higher service rate ($w_2 > w_1$) and an adjustment factor δ to give a higher probability of survival in the immediate reward.

System state. The system state is defined as $S = (n_I, n_D, h_1, h_2, a)$, where n_I (n_D) denotes the number of class-I (class-D) patients remaining on the accident site, $h_j (j = 1, 2)$ is the number of patients waiting in the tier-j hospital, and a represents the ambulance states and is defined as in the original problem; however, because there is only one ambulance, it can be explicitly stated as $a \in \{a_1^0, a_1^1, a_2^0, a_2^1\}$, where a_i^0 indicates the ambulance is traveling to the tier-i hospital and a_i^1 denotes that it is on the return trip.

State transition probability. The model has three events – patient death, ambulance arrival, and patient discharge. A patient dies if he or she is not picked up by an ambulance before the survival time expires. The ambulance arrival event refers to the ambulance's arrival at a destination hospital or its return to the accident site. Finally, the patient discharge event occurs when the patient's treatment at a hospital is complete.

The system state transitions to other states upon the occurrence of these three events. *Patient death*, *discharge*, and *ambulance arrival at a hospital* events are natural processes that do not involve decision making; only the event of *an ambulance arriving at the accident site* requires a decision to be made.

As described above, this model assumes that the inter-arrival times for all three events follow an exponential distribution. Under this assumption, the event generation process of this model follows the Poisson process (Ross et al., 1996). Therefore, the system state transition probability is given by the arrival rate of each event divided by the sum of all rates, which is:

$$\gamma = (\mu_j) + (r_I n_I + r_D n_D) + (1_{h_1>0} w_1 + 1_{h_2>0} w_2). \tag{E.1}$$

where the first term is the rate at which ambulances arrive at the scene or at a hospital, the second term is the rate at which the patients die at the accident site and the final term denotes the rate at which patients are discharged from the hospital. The indicator function $1_{h>0}$ takes a value of 1 when there is at least one patient in the hospital and a value of 0 when there are no patients. Therefore, when an ambulance returns to the accident site, the transition probabilities for the three events are as follows:

$$\text{Transition probability by patient death event} = \frac{r_i n_i}{\gamma}, i \in \{I, D\}, \tag{E.2}$$

$$\text{Transition probability by patient discharge event} = \frac{1_{h_j>0} w_j}{\gamma}, j \in \{1, 2\}, \tag{E.3}$$

$$\text{Transition probability by ambulance arrival event} = \frac{\mu_j}{\gamma}, j \in \{1, 2\}. \tag{E.4}$$

Finally, the resulting state transition diagram is shown in Figure E.1.

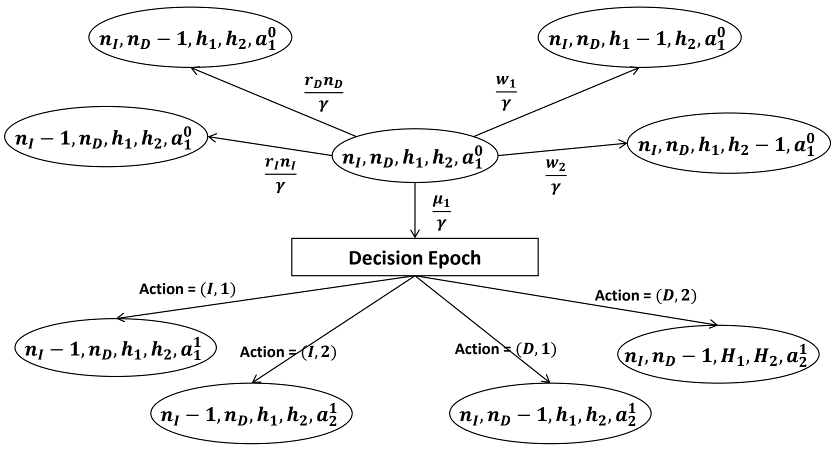

Figure E.1: A State transition when the ambulance is returning to the scene. The value of γ is the sum of the rates of all state transition events, $\gamma = (\mu_j) + (r_I n_I + r_D n_D) + (1_{h_1>0} w_1 + 1_{h_2>0} w_2)$.

F Proof of Proposition 1

Proposition 1. Suppose that $\mu^k > 2w^{k'}$ for $k, k' \in \{A, B\}$, and let us define $r^* = \frac{\mu^B w^B (h^A \mu^A + w^A) - \mu^A w^A (h^B \mu^B + w^B)}{\mu^A w^A h^B - \mu^B w^B h^A}$. r^* is defined only when $\mu^A w^A h^B \neq \mu^B w^B h^A$. When there is only one patient at the site whose abandonment rate is r, the optimal hospital to transport the patient to is determined as follows:

(i) If $\mu^A w^A h^B = \mu^B w^B h^A$, transport the patient to hospital k that has the smaller $(\frac{1}{\mu^k} + \frac{h^k}{w^k})$.

(ii) If $r^* = 0$, transport the patient to hospital k that has the smaller $(\frac{1}{\mu^k} \frac{h^k}{w^k})$.

(iii) If $r^* < 0$, transport the patient to hospital k that has the smaller $(\frac{1}{\mu^k} + \frac{h^k}{w^k})$.

(iv) If $0 < r \leq r^*$, transport the patient to hospital k that has the smaller $(\frac{1}{\mu^k} + \frac{h^k}{w^k})$.

(v) If $0 < r^* < r$, transport the patient to hospital k that has the larger $(\frac{1}{\mu^k} + \frac{h^k}{w^k})$.

Proof. Proposition 1 applies to the following subsets of the state space: $(0, 1, h^A, h^B, a)$ and $(1, 0, h^A, h^B, a)$. We first look at state $(0, 1, h^A, h^B, a)$. As there is only one patient at the scene, when transporting the patient to either of the two hospitals, the optimal decision is to choose an action that yields the largest immediate reward.

Under the assumption of $\mu^k > 2w^{k'}$ for $k, k' \in \{A, B\}$, the reward for transporting a class-D patient to hospital A or B is given by Equation (6) in Section 4.2 as:

$$R(s, A) = \frac{\mu^A w^A}{(h^A r_D + w^A)(r_D + \mu^A)}, \tag{F.1}$$

$$R(s, B) = \frac{\mu^B w^B}{(h^B r_D + w^B)(r_D + \mu^B)}. \tag{F.2}$$

Comparing $R(s, A)$ and $R(s, B)$ can be written as determining inequality of the following:

$$\frac{\mu^A w^A}{(h^A r_D + w^A)(r_D + \mu^A)} \gtreqless \frac{\mu^B w^B}{(h^B r_D + w^B)(r_D + \mu^B)} \tag{F.3}$$

Rearranging Equation (F.3) and by letting m_k and c_k denote $\frac{h^k}{\mu^k w^k}$ and $\frac{h^k}{w^k} + \frac{1}{\mu^k}$, we have

$$m_B r_D + c_B \gtreqless m_A r_D + c_A \tag{F.4}$$

Action A is optimal if RHS of Equation (F.4) is smaller, and action B is optimal if LHS is smaller. When equality holds, either of the actions is optimal. Note that both the left- and right-hand side of Equation (F.4) can be seen as a linear function of r_D ($r_D > 0$), where m_k is the slope and c_k is its intercept. r^*, declared in the Proposition, can be written as $\frac{c_A - c_B}{m_B - m_A}$, and it makes equality hold in Equation (F.4). Graphically, it is where where the two lines represented by the LHS and RHS intersect.

<u>(i) If $\mu^A w^A h^B = \mu^B w^B h^A$</u>, $m_B = m_A$ and we have

$$\frac{h^B}{w^B} + \frac{1}{\mu^B} \gtreqless \frac{h^A}{w^A} + \frac{1}{\mu^A}. \tag{F.5}$$

Thus, an optimal action can be determined by comparing LHS and RHS of Equation (F.5) and choosing the one with a smaller value. Graphical interpretation is that since the two lines have the same slope, we only need to compare their intercepts and choose the one with a smaller intercept.

<u>(ii) If $r^* = 0$</u>, $\mu^B w^B (h^A \mu^A + w^A) = \mu^A w^A (h^B \mu^B + w^B)$, which gives $c_B = c_A$. Then Equation (F.5) reduces to

$$\frac{1}{\mu^B} \frac{h^B}{w^B} r_D \gtreqless \frac{1}{\mu^A} \frac{h^A}{w^A} r_D. \tag{F.6}$$

Thus, an optimal action can be determined by comparing LHS and RHS of Equation (F.6) and choosing the one with a smaller value. Graphically, we have two lines that have the same intercepts, and thus we only need to compare their slopes to choose the one with a smaller slope.

(iii) If $r^* = \frac{c_A - c_B}{m_B - m_A} < 0$, it requires either of the followings to be true: ($m_B > m_A$ and $c_B > c_A$) or ($m_B < m_A$ and $c_B < c_A$).

For ($m_B > m_A$ and $c_B > c_A$), since $r_D > 0$, Equation (F.4) holds true with the following inequality:

$$m_B r_D + c_B > m_A r_D + c_A. \tag{F.7}$$

Therefore, action A is optimal under these conditions. Graphically, both the slope and intercept of the RHS line are smaller than the LHS line, and therefore the RHS line lies below the LHS line for all $r_D > 0$.

Similarly, for ($m_B < m_A$ and $c_B < c_A$),

$$m_B r_D + c_B < m_A r_D + c_A, \tag{F.8}$$

making action B optimal.

From the conditions, ($m_B > m_A$ and $c_B > c_A$) and ($m_B < m_A$ and $c_B < c_A$), we see that optimal action is determined by only comparing c_A and c_B, which is Equation (F.5).

(iv) If $0 < r_D \leq r^* = \frac{c_A - c_B}{m_B - m_A}$, it requires either of the following to be true: ($m_B > m_A$ and $c_B < c_A$) or ($m_B < m_A$ and $c_B > c_A$).

For ($m_B > m_A$ and $c_B < c_A$), we have

$$m_B r_D + c_B \leq m_A r_D + c_A, \tag{F.9}$$

and action B is optimal. Similarly, it easy to see action A is optimal when ($m_B < m_A$ and $c_B > c_A$). Thus, as in (iii), an optimal action can be determined by only comparing c_A and c_B, which is Equation (F.5). Graphical explanation is that smaller/larger relationship at the intercept between the two lines remains the same for interval from 0 to the intersect r^*.

(v) If $r_D > r^* = \frac{c_A - c_B}{m_B - m_A} > 0$, it requires either ($m_B > m_A$ and $c_B < c_A$) or ($m_B < m_1$ and $c_B > c_A$).

If ($m_B > m_A$ and $c_B < c_A$),

$$m_B r_D + c_B > m_A r_D + c_A, \tag{F.10}$$

and if ($m_B < m_A$ and $c_B > c_A$),

$$m_B r_D + c_B < m_A r_D + c_A. \tag{F.11}$$

Thus, in this case, an optimal action is the one with larger intercept, which is the opposite to the result in (iv). Graphically, when r_D is beyond the intercept of the two lines, the smaller/larger relationship is the inverse of that at the intercept.

The optimal action for state $(1, 0, h^A, h^B, a)$ can be derived in exactly the same way. This completes the proof of Proposition 1. □

G Derivation of Equation (7)

When policy π_{I2D1} is executed in state $s = (1, 1, h_1, h_2, a_j^1)$, the value obtained by this policy is as follows:

$$\begin{aligned} V_{\pi_{I2D1}}(s) = &\left(\frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)}\right) + \left(\frac{\mu_2^2}{(\mu_2 + w_1 + r_D)^2} \times \frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D + \mu_1)}\right) \\ &+ \left(\frac{2\mu_2^2 w_1}{(\mu_2 + w_1 + r_D)^3} \times \frac{\mu_1 w_1}{((h_1 - 1) r_D + w_1)(r_D + \mu_1)}\right) + \cdots . \end{aligned} \tag{G.1}$$

where the first term in parentheses is the reward for transporting a class-I patient to a tier-2 hospital;terms in the second, third and thereafter parentheses represent the future value of $(D, 1)$ portion of the policy. That is, we compute the reward for sending the class-D patient to a tier-1 hospital under each possible state of the tier-1 hospital. In each set of parentheses, the second term is the immediate reward for sending the class-D patient to a tier-1 hospital in a certain state, whereas the first term is the probability that the tier-1 hospital is in that particular state. For example, the first term in the second set of parentheses is the probability that the class-D patient lives and there is no patient discharged from the tier-1 hospital during the round trip of the ambulance's earlier transportation task (i.e., taking a class-I patient to the tier-2 hospital and returning to the site to pick up the class-D patient). Then, the immediate reward of taking the class-D patient to the tier-1 hospital is expressed by the second term in the parentheses.

We can further simplify Equation (G.1). First, the fourth and later terms can be ignored because the probabilities are very small under the assumption of $\mu > 2w$. Second, as r_D is very small, the reward terms in the second and third sets of parentheses are approximately the same:

$\frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D+\mu_1)} \approx \frac{\mu_1 w_1}{((h_1-1) r_D + w_1)(r_D+\mu_1)}$. Thus, we can rewrite Equation (G.1) as:

$$\begin{aligned} V_{\pi_{I2D1}}(s) \approx & \left(\frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)} \right) \\ & + \left(\frac{\mu_2^2}{(\mu_2 + w_1 + r_D)^2} + \frac{2\mu_2^2 w_1}{(\mu_2 + w_1 + r_D)^3} \right) \times \frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D + \mu_1)}. \end{aligned} \tag{G.2}$$

The summation of the probabilities shown in parentheses on the second line is approximately the same as the probability that the class-D patient lives. Therefore, the value when policy π_{I2D1} is executed in the current state $s = (1, 1, \cdot, \cdot, a_j^1)$ can be approximated as follows:

$$V_{\pi_{I2D1}}(s) = \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + r_D)^2} \times \frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D + \mu_1)}. \tag{G.3}$$

The opposite case of $V_{\pi_{D1I2}}$ is derived similarly as:

$$V_{\pi_{D1I2}}(s) = \frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D + \mu_1)} + \frac{\mu_1^2}{(\mu_1 + r_I)^2} \times \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)}, \tag{G.4}$$

where π_{D1I2} denotes the policy in which the class-D patient is first transported to the tier-1 hospital, and then the class-I patient is transported to the tier-2 hospital.

We now compare $V_{\pi_{I2D1}}$ and $V_{\pi_{D1I2}}$ to derive a condition for which π_{I2D1} is superior to π_{D1I2}:

$$\begin{aligned} & \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + r_D)^2} \times \frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D + \mu_1)} \\ & \qquad > \frac{\mu_1 w_1}{(h_1 r_D + w_1)(r_D + \mu_1)} + \frac{\mu_1^2}{(\mu_1 + r_I)^2} \times \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)}. \end{aligned} \tag{G.5}$$

After rearranging Equation (G.5) and applying the assumption that r_i for $i \in \{I, D\}$ is very small, we obtain the following approximate condition:

$$\frac{\mu_2 w_2}{\mu_1 w_1} h_1 + \frac{w_2}{r_D} \left(\frac{\mu_2}{\mu_1} - \frac{r_D}{r_I} \right) > h_2. \tag{G.6}$$

Finally, as r_i is very small for $i \in \{I, D\}$, $\mu_1 \times r_D \approx \mu_2 \times r_I$, and we finally state the condition for which π_{I2D1} is superior to π_{D1I2} as:

$$\frac{\mu_2 w_2}{\mu_1 w_1} h_1 > h_2. \tag{G.7}$$

H Derivation of Equation (8)

We follow the same procedure as in Section G to derive Equation (8). The value function when policy π_{I2D2} is executed in the current state $s = (1, 1, h_1, h_2, a_j^1)$ is:

$$V_{\pi_{I2D2}}(s) = \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + w_2 + r_D)^2} \times \frac{\mu_2 w_2}{((h2+1)r_D + w_2)(r_D + \mu_2)} + \frac{2\mu_2^2 w_2}{(\mu_2 + w_2 + r_D)^3} \times \frac{\mu_2 w_2}{(h_2 r_D + w_2)(r_D + \mu_2)} + \cdots . \tag{H.1}$$

Using the same argument as in Section G, Equation (H.1) can be approximated as follows:

$$V_{\pi_{I2D2}}(s) = \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + r_D)^2} \times \frac{\mu_2 w_2}{(h_2 r_D + w_2)(r_D + \mu_2)}. \tag{H.2}$$

That is, if the class-D patient at the accident site remains alive while the ambulance transports the patient and returns, the reward for the next decision can be approximated as $\frac{\mu_2 w_2}{(h_2 r_D + w_2)(r_D + \mu_2)}$ and the probability of this condition is $\frac{\mu_2^2}{(\mu_2 + r_D)^2}$. Likewise, $V_{\pi_{D2I2}}$ is given by:

$$V_{\pi_{D2I2}}(s) = \frac{\mu_2 w_2}{(h_2 r_D + w_2)(r_D + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + r_I)^2} \times \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)}. \tag{H.3}$$

Thus, the condition under which π_{I2D2} is superior to π_{D2I2} is:

$$\frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + r_D)^2} \times \frac{\mu_2 w_2}{(h_2 r_D + w_2)(r_D + \mu_2)} > \frac{\mu_2 w_2}{(h_2 r_D + w_2)(r_D + \mu_2)} + \frac{\mu_2^2}{(\mu_2 + r_I)^2} \times \frac{\mu_2 w_2}{(h_2 r_I + w_2)(r_I + \mu_2)}. \tag{H.4}$$

Finally, rearranging Equation (H.4) and applying the assumption that r_i for $i \in \{I, D\}$ is very small, Equation (H.4) can be simplified as follows:

$$\frac{w_2}{r_I r_D} > h_2. \tag{H.5}$$

References

Fang, J., Zhao, L., Fransoo, J.C., and Van Woensel, T., 2013. Sourcing strategies in supply risk management: An approximate dynamic programming approach. *Computers & Operations Research*, 40(5):1371–1382.

Powell, W.B., 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality.*

John Wiley & Sons.

Ross, S.M., Kelly, J.J., Sullivan, R.J., Perry, W.J., Mercer, D., Davis, R.M., Washburn, T.D., Sager, E.V., Boyce, J.B., and Bristow, V.L., 1996. *Stochastic processes*, volume 2. Wiley New York.

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