J SYST SCI SYST ENG
Vol. 34, No. 1, February 2025, pp. 55–77
DOI: https://doi.org/10.1007/s11518-024-5625-9
ISSN: 1004-3756 (paper), 1861-9576 (online)
CN 11-2983/N

Simulation Optimization for Inpatient Bed Allocation with Sharing

Jie Li,[a] Sichen Li,[b] Jun Luo,[b, c] Haihui Shen[a, c]

[a]Sino-US Global Logistics Institute, Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200240, China
lj9418@126.com

[b]Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200240, China
l-schen@sjtu.edu.cn

[c]Data-Driven Management Decision Making Lab, Shanghai Jiao Tong University, Shanghai 200240, China
jluo_ms@sjtu.edu.cn (✉), shenhaihui@sjtu.edu.cn

Abstract. The inpatient bed allocation that allows beds shared among different departments is an important and challenging problem for a healthcare system. When the objective function(s) and (some) constraints need to be estimated via expensive and noisy stochastic simulation, a simulation optimization algorithm is required to solve this problem. In literature, there is a heuristic algorithm highly customized for one specific inpatient bed allocation problem, and it performs quite well on that problem. However, its lack of theoretical convergence and high specialization may not give practitioners enough confidence to apply it on real inpatient bed allocation problems. To mitigate such issues, this paper proposes to use the empirical stochastic branch-and-bound (ESB&B) algorithm, which is theoretically convergent and suitable for relatively general problems. A modest improvement for the original ESB&B algorithm is made and how to adapt the ESB&B algorithm to inpatient bed allocation problems is presented. Numerical experiments reveal the generality and fairly satisfying performance of the ESB&B algorithm, and the superiority of the improved ESB&B algorithm over the original one.

Keywords: Healthcare management, resource sharing, bed allocation, simulation optimization, empirical stochastic branch-and-bound (ESB&B)

1. Introduction

The number of inpatient beds is the most fundamental measure of capacity for a healthcare system (Green 2004), since it can not be arbitrarily increased without matching the resources of staff, facilities, space, etc. The insufficiency of inpatient beds will often cause congestion in upstream departments (or wards) and lead to long waiting time of patients. So, how to optimally allocate the limited inpatient beds among different departments is an important issue. However, it is also a difficult issue due to the complexity of the healthcare system and the multiple objectives and constraints that need to be considered. A healthcare system, which can be viewed as an integrated and adaptive set of people, processes and products, is a typical complex service system (Tien and Goldschmidt-Clermont 2009). Patients arrive at the healthcare system, flow among different departments, and depart after service completion or waiting too long, forming a complicated queueing network (Bhattacharjee and Ray 2014). Only to effectively evaluate the queueing-related performance (such as utiliza-

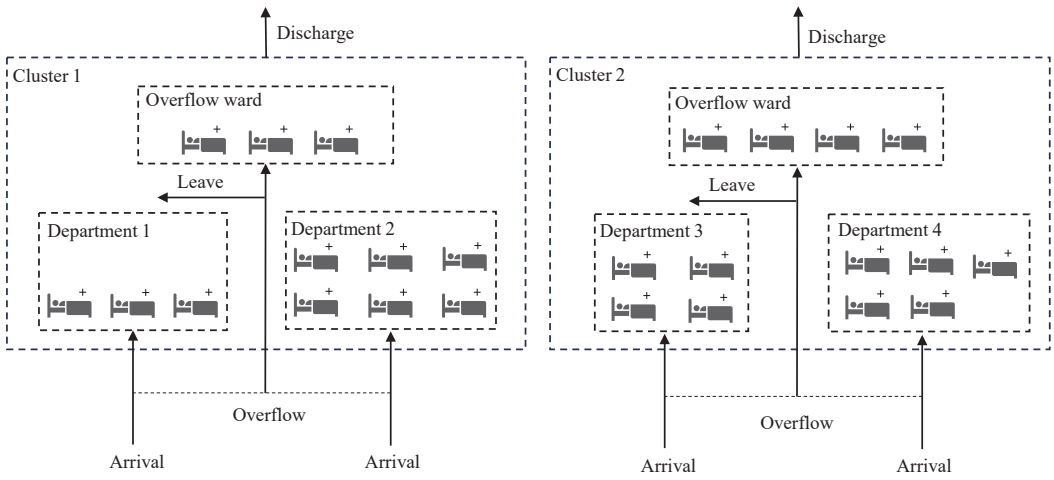

Figure 1 An Example of CO Configuration of 25 Inpatient Beds among 4 Departments

tion of inpatient beds and waiting time of patients) under a specific inpatient bed allocation scheme (without actually trying it in the real system) is already a challenging problem. One may consider simplifying the queueing network by imposing some assumptions so that closed-form expressions for interested performance exist, suffering from the gap between the model and the reality. Alternatively, one can resort to simulation that ensures high fidelity but is time-consuming. Oftentimes, the decision makers also have to consider many other objectives or constraints in addition to the basic queueing-related performance when optimizing the inpatient bed allocation, e.g., profitability, quality, patient satisfaction, and social equity (Porter 2010, Zhou et al. 2018).

Recently, the emerging attention to the shared (or pooled) inpatient beds among departments further increases the difficulty of the inpatient bed allocation (Wang et al. 2024). In the past, each department was allocated a certain number of dedicated inpatient beds, which are occupied and used exclusively by itself. In such a situation, the decision variables of the inpatient bed allocation optimization problem are merely the numbers of inpatient beds for all departments. Since there are well-known benefits of resource pooling in many systems, it is natural to anticipate that sharing (or pooling) some inpatient beds among departments may reduce congestion and patients' waiting time. However, the nursing cost is usually higher when shared inpatient beds exist, because the shared inpatient beds must be equipped with nurses who are able to nurse different types of patients. Therefore, the optimal allocation of inpatient beds may be neither the fully dedicated configuration nor the fully shared configuration, but an intermediate one.

In general, one can consider the following flexible inpatient bed configuration. All the departments are divided into several (at least one) clusters, and in every cluster each department has some dedicated inpatient beds and there are some other inpatient beds (that form a so-called overflow ward) to be shared with all departments in this cluster. When a patient arrives, s/he will be assigned a dedicated inpatient bed in his destination department if

there are available dedicated beds in that department; otherwise, s/he will be assigned a shared inpatient bed in the cluster to which his destination department belongs given that there are available shared beds in that cluster. If all dedicated and shared inpatient beds are occupied, s/he will wait until her/his waiting time threshold is reached, in which case s/he will leave without service (for simplicity, say this patient is rejected). Such flexible configuration is referred to clustered overflow (CO) configuration in Izady and Mohamed (2021) and (Gong et al. 2022). Figure 1 shows an example of CO configuration with 25 inpatient beds among 4 departments, wherein two clusters are formed and there are 3 and 4 shared beds in the two clusters respectively. It is easy to see that the aforementioned fully dedicated configuration and fully shared configuration are just two specific instances of the CO configuration. So, if one optimizes the inpatient bed allocation under the CO configuration, the result will be better than (or at least the same as) that in the fully dedicated configuration or fully shared configuration. However, such benefits come at a cost. The optimization under the CO configuration is much more challenging due to the larger space of feasible solutions.

Undoubtedly, it is quite attractive to both scholars and practitioners to design an efficient algorithm that is capable of solving the inpatient bed allocation problem under the CO configuration. There are indeed some attempts in the literature, but drawbacks or limitations of the existing approaches make the problem still unsolved (or at least partially unsolved); see detailed literature review in Section 2. This paper aims to introduce a general simulation optimization algorithm (named empirical stochastic branch-and-bound or ESB&B for short) for the inpatient bed allocation problem under the CO configuration, which has several advantages compared to the existing ones.

The rest of the paper is organized as follows. Detailed literature review on the existing approaches for inpatient bed allocation with sharing and the simulation optimization technique adopted in the paper is given in Section 2. Section 3 illustrates the general bed allocation problem, and introducing the problem from Gong et al. (2022) as an example. In Section 4, the ESB&B algorithm is introduced with a modest improvement, then how to adapt the ESB&B algorithm to the inpatient bed allocation problems under CO configuration is discussed. In Section 5, numerical experiments are conducted to show the generality and fairly satisfying performance of ESB&B algorithm in contrast to other algorithms, and the superiority of the improved ESB&B algorithm over the original one. Finally, Section 6 concludes the paper.

2. Literature Review

Some researchers have considered configurations that allow shared beds but are less flexible than the CO configuration. Best et al. (2015) consider a configuration called wing formation (WF), where departments with similar functions are brought together to form a wing, sharing inpatient beds. In other words, WF is a special case of CO in which each department has no dedicated inpatient beds. The optimization objective is defined as the total expected utility gained from patients and the main constraint is the total number of inpa-

tient beds. To evaluate the expected utility for each cluster, they simplify each cluster to an $M/M/n + M$ (Erlang-A) queueing model, i.e., the classical $M/M/n$ queueing model with exponentially distributed waiting time threshold, for which closed-form expression exists. Even so, they find that the feasible region is still too large to afford an exact algorithm to find the optimal solution (although WF is already less flexible than CO). They reduce the feasible region by imposing the restriction that clusters are formed by making cuts in a fixed sequence of the departments, and then a dynamic programming (DP) approach is used to find the exact optimal solution. Bekker et al. (2017) consider a configuration called earmarking (EAR), where each department has dedicated beds and all departments share some beds. In other words, EAR is a special case of CO that there is only one cluster. However, they do not consider the issue of optimizing the inpatient bed allocation under EAR configuration.

Izady and Mohamed (2021) first consider the general CO configuration. They propose two formulations for the inpatient bed allocation under CO configuration. The first one is the total cost minimization (TCM) formulation, which seeks to minimize the total expected daily costs of rejected patients and nursing teams subject to the total number of inpatient beds. The second one is the constrained blocking minimization (CBM) formulation, which aims to minimize the total expected number of rejected patients subject to the total number of beds and the expected nursing cost falling below a given threshold. To evaluate the objective functions, they simplify the problem by imposing assumptions that (i) patients arrive following the Poisson process; (ii) the length of stay once admitted is exponentially distributed; and (iii) the waiting time threshold is zero, then the approximation method is adopted. They also restrict the clusters to cuts of a fixed sequence of the departments as in Best et al. (2015), in order to reduce the feasible region. The TCM formulation is solved using DP approach and conjugate direct orthogonal shift (CDOS) heuristic, while the CBM formulation is solved using an integer linear programming approach adapted with CDOS heuristic and enumeration.

Gong et al. (2022) revisit the general CO configuration and try to solve it in a more realistic setting. In particular, they do not impose assumptions on the distributions of patients' interarrival times and length of stay. The optimization objective is defined as the weighted total cost of rejecting patients, holding patients waiting, and nursing cost (for dedicated beds and shared beds). They also consider an equity-of-access constraint that is important to public hospitals in China. Both the above objective and constraint can not be evaluated analytically or even with some fair approximation. So, an inpatient simulation model is developed and the optimization is based on the inputs and outputs of the stochastic simulation, which is known as a simulation optimization problem (Fu 2015). Besides, they do not restrict the feasible region as in Best et al. (2015) and Izady and Mohamed (2021). These factors together make their problem quite challenging to solve. A simulation-based metaheuristic approach (SMA) is proposed to search for the optimal solution in the entire feasible re-

gion. In particular, a niching genetic algorithm (GA) framework is proposed to optimize the cluster partition, and each partition is evaluated by optimizing the bed allocation through an adaptive hyperbox algorithm-based local search. The practical performance of SMA is quite good as demonstrated in their simulation experiments.

There are many other studies on the healthcare system that utilize simulation techniques to deal with the complexity, see Shirazi et al. (2021) and Ghasemi et al. (2023) for instance. However, in these studies, simulation models are not directly used for objective function evaluation. Instead, simulation models are run to produce some input parameters or constraints for the mathematical model. In the end, the problems are transformed into classical deterministic optimization problems, and then exact or heuristic methods are designed to solve the problems. Such simulation techniques are also used in addressing the complexity of some disaster occurrences (Khalili-Damghani et al. 2022, Ahmadi Choukolaei et al. 2024). But for the above mentioned inpatient bed allocation problem, the simulation is run to evaluate the objective function and verify the constraint for a given solution. It is not possible to transform the formulation into deterministic optimization, and the optimization has to be carried out based on the inputs and noisy outputs of the stochastic simulation.

This paper focuses on general inpatient bed allocation problems with (a) flexible sharing, which refers to the CO configuration; (b) realistic queueing network setting, which means the above objective and some constraints have to be evaluated via time-consuming simulation. Among all the literature including those mentioned above, to the best of our knowledge, the SMA of Gong et al. (2022) is the only algorithm capable for such problems. However, there are still two drawbacks or limitations in the SMA. First, the SMA is a heuristic simulation optimization algorithm, which means there is no guarantee that the solved solution will converge to the optimal solution when the solving time (essentially the simulation time) goes to infinity. Second, the SMA is highly customized for the specific problem setting considered in Gong et al. (2022). In other words, when the setting is changed (no matter the objective, constraints, or the operation rules of the healthcare system), SMA may no longer perform well or may need adjustment. These two issues will make a decision maker not confident enough when applying SMA to real problems that are not exactly the same as in Gong et al. (2022).

This paper aims to introduce a simulation optimization algorithm for inpatient bed allocation problems under CO configuration (i.e., the most flexible configuration allowing bed sharing), which mitigates the two issues of SMA mentioned above. In particular, such an algorithm needs to be convergent theoretically, in addition to good performance in finite time. Besides, it should be suitable for relatively general inpatient bed allocation problems under CO configuration where the objective and (some) constraints need to be evaluated via stochastic simulation, including but not limited to the problem considered in Gong et al. (2022). In other words, such an algorithm should perform fairly well for a variety of problems without specific adjustments or parameter tuning. These two properties may

reassure the decision makers when applying the algorithm in real inpatient bed allocation problems.

Simulation optimization has been a challenging but active research topic in recent decades. Depending on the decision variables, simulation optimization can be mainly classified as continuous simulation optimization and discrete simulation optimization, and for each type various algorithms have been proposed in the literature; see Amaran et al. (2016) for a review. Apparently, the general inpatient bed allocation problems under CO configuration (like the one considered in Gong et al. (2022)) belong to the discrete simulation optimization category. However, many existing algorithms for discrete simulation optimization problems that are proved to be (locally or globally) convergent and have satisfying finite sample performance, e.g., the COMPASS algorithm of Hong and Nelson (2006) and the GPS algorithm of Sun et al. (2014), are not suitable for inpatient bed allocation problems. The main reason is that these discrete simulation optimization algorithms assume that the feasible region is simply the intersection of the integer lattice $\mathbb{Z}^d$ with a closed set in $\mathbb{R}^d$, where the dimensionality d is fixed. Moreover, there is a well-defined neighborhood structure in the feasible region, i.e., for two solutions whose (Euclidean) distance is short their function values should be close (which is analogous to the smoothness assumption for a continuous surface). However, for the inpatient bed allocation problems under CO configuration, the feasible region is much more complicated. First, a feasible solution consists of how the clusters are formed, the number of shared beds in each cluster, and the number of dedicated beds in each department. Clearly, the dimensionality of the feasible solution varies when the number of clusters varies. Second, the distance (no matter Euclidean distance, Manhattan distance, or others) between two solutions (even given that they have the same dimensionality) does not give any information about the difference of their function values. For example, by merely swapping two departments in two clusters, the resulting two solutions tend to have a small distance, but their function values may be dramatically different. So, it remains to be investigated what simulation optimization algorithm is capable of solving the inpatient bed allocation problems under CO configuration, which is theoretically convergent and has satisfying finite sample performance.

Essentially, the inpatient bed allocation problems under CO configuration are combinatorial optimization problems where the objective function and (some) constraints need to be estimated via expensive and noisy stochastic simulation. Recall that for deterministic combinatorial optimization problems (i.e., the objective function and constraints can be analytically evaluated), the branch-and-bound (B&B) algorithmic framework is a widely-used method for producing exact solution (Lawler and Wood 1966). Norkin et al. (1998a) and Norkin et al. (1998b) adapt the B&B idea to stochastic optimization problems and propose the stochastic B&B (SB&B) method. By assuming that the lower and upper bounds of the subregions can be estimated more and more precisely with increasing simulation effort, the SB&B method is proved to be globally convergent for problems with finite feasible solutions.

However, for general simulation optimization problems, there may not exist such bound estimators. To make SB&B more practical, Xu (2009) and Xu and Nelson (2013) propose the empirical SB&B (ESB&B) algorithm for discrete simulation optimization, which estimates the bounds based on the performance of sampled solutions and is still proved to be globally convergent. Although in Xu (2009) and Xu and Nelson (2013) they focus on discrete simulation optimization problems where the feasible region is the intersection of the integer lattice $\mathbb{Z}^d$ with a closed set in $\mathbb{R}^d$, the theoretical results hold for the arbitrary feasible region as long as the number of feasible solutions is finite. Moreover, due to the inherited spirit from the classical B&B algorithmic framework, we believe that the ESB&B algorithm can be extended to the inpatient bed allocation problems under CO configuration and its finite sample performance should be fairly satisfying.

The main contributions of this paper are as follows. First, for general inpatient bed allocation problems under CO configuration where the objective and (some) constraints need to be evaluated via stochastic simulation, we propose to use ESB&B algorithm, which mitigates the two drawbacks of the existing SMA algorithm. Second, a modest improvement for the original ESB&B algorithm is made and how to adapt the ESB&B algorithm to inpatient bed allocation problems is presented. The convergence of ESB&B is guaranteed by its theoretical property and is also reflected in simulation experiments. The generality of ESB&B over a variety of problems is also demonstrated in simulation experiments, which outperforms the existing SMA. Moreover, the superiority of the improved ESB&B algorithm over the original ESB&B is also observed.

3. Problem Description

Suppose a healthcare system consists of $I \in \mathbb{Z}^+$ departments and $B \in \mathbb{Z}^+$ inpatient beds. The decision maker needs to divide the departments into $J \in \mathbb{Z}^+$ clusters, where $J \le I$ and one department can only belong to one cluster. Let $\mathbb{I} = \{1, \cdots, I\}$ be the set of departments, where the corresponding index is i (or h sometimes when necessary). Let $\mathbb{J} = \{1, \cdots, J\}$ be the set of clusters, where the corresponding index is j. The decision maker also needs to decide the set of departments that belong to cluster $j \in \mathbb{J}$, which is denoted as Γ_j. In addition to the cluster partition, the decision maker also needs to determine the number of dedicated beds for department $i \in \mathbb{I}$, which is denoted as $n_i \in \mathbb{N}$, and the number of shared beds for cluster $j \in \mathbb{J}$, which is denoted as $o_j \in \mathbb{N}$. This paper considers the following general inpatient bed allocation optimization problem under CO configuration:

$$\min_{\substack{J,\{\Gamma_1,\cdots,\Gamma_J\},\\ \{n_1,\cdots,n_I\},\\ \{o_1,\cdots,o_J\}}} f(J, \Gamma_1, \cdots, \Gamma_J, n_1, \cdots, n_I, o_1, \cdots, o_J)$$

s.t.

$$\sum_{i\in\mathbb{I}} n_i + \sum_{j\in\mathbb{J}} o_j = B$$

$$g_k(J, \Gamma_1, \cdots, \Gamma_J, n_1, \cdots, n_I, o_1, \cdots, o_J) \le c_k, \quad \forall k \in \{1, \cdots, K\}$$

$$n_i, o_j \in \mathbb{N}, \quad \forall i \in \mathbb{I}, j \in \mathbb{J}$$

$$J \in \mathbb{I}$$

where f represents a general objective function, whose value given all the decision vari-

ables needs to be evaluated via stochastic simulation; g_k, $k \in \{1, \cdots, K\}$ represents K other performances which need to be controlled under upper bounds c_k (note that a lower bound can be transformed to an upper bound by adding a negative sign to the performance). It is possible that some g_k also needs to be evaluated via stochastic simulation.

To show how complicated the problem can be, we use the same inpatient bed allocation problem as in Gong et al. (2022) as an illustrative example, while emphasizing that our simulation optimization algorithm is proposed for more general inpatient bed allocation problems under CO configuration where the objective and (some) constraints need to be evaluated via stochastic simulation. The key details of this illustrative problem are summarized as follows. When a patient arrives at the healthcare system, s/he will have a destination department. If that department has available dedicated beds, s/he will be admitted immediately and assigned a dedicated bed in that department. If dedicated beds are unavailable but there are available shared beds in the cluster to which that department belongs, s/he will also be admitted immediately and assigned a shared bed in that cluster. Otherwise, the patient needs to wait in line of that department. When later a dedicated bed (in that department) becomes available, it will be assigned to the waiting patients (in that department) according to the first-come-first-served (FCFS) rule. When later a shared bed (in that cluster) becomes available, it will be assigned to the waiting patients in that cluster according to a specific priority rule (specified later). When a patient's waiting time reaches his waiting threshold (i.e., the maximum time s/he is willing to wait), s/he will leave the system immediately without service (i.e., rejected). There are some additional assumptions to simplify the problem:

(a) For department $i \in \mathbb{I}$, there is only one type of patients whose destination department is i (call them type i patients for short).

(b) For type $i \in \mathbb{I}$ patients: 1) They arrive randomly and their interarrival times are independent and identically distributed (i.i.d.) random variables with mean $1/\lambda_i$ and probability density (pdf) f_i; 2) They have the same waiting time threshold D_i; 3) Their lengths of stay (i.e., the time lengths they occupy the inpatient beds) once admitted are i.i.d. random variables with mean $1/\mu_i$ and pdf g_i, which is irrelevant to the inpatient bed type (dedicated or shared).

(c) The ratio of nurses to beds is a constant φ, which is the same for any department and overflow ward. The average nursing cost per time unit of dedicated beds in department $i \in \mathbb{I}$ is ξ_i, and that of shared bed in cluster $j \in \mathbb{J}$ is ξ_{Γ_j}.

(d) Different types of patients arrive independently of each other, and a patient who finishes service (in either a department or an overflow ward) will leave the system immediately without entering another department or overflow ward.

(e) The inpatient beds are fully flexible and their allocation is not limited by anything else like bed class (e.g., single, double, etc.) or special requirement (e.g. same sex or isolation requirement, etc.).

Let $W_i(t)$ denote the total waiting time of type $i \in \mathbb{I}$ patients by time t, and $R_i(t)$ the total number of type i patients rejected. For a type $i \in \mathbb{I}$ patient, let C_i^W denote the cost of waiting per time unit, and C_i^R the cost of being rejected. For type $i \in \mathbb{I}$ patients, let $P_i(t)$ denote the admission rate by time t, which is the ratio of the number of admitted type i patients to the number of arrived type i patients by time t. To ensure the equity of access to different departments, there is an upper bound $\delta > 0$ for $\lim_{t\to\infty}(P_i(t) - P_h(t))$ (i.e., the long-run admission rate difference) for any $i, h \in \mathbb{I}$. It is a hard constraint in the sense that if an inpatient bed allocation scheme leads to the violation of $\lim_{t\to\infty}(P_i(t) - P_h(t)) \le \delta, \forall i, h \in \mathbb{I}$, then it is an infeasible scheme. The priority rule of assigning a newly available shared bed to the waiting patients within the same cluster $j \in \mathbb{J}$ also accommodates such consideration of equity of access. In particular, for cluster $j \in \mathbb{J}$, if the equity-of-access constraint is satisfied for all departments in this cluster (i.e., Γ_j), then higher priority is given to patient type $i \in \Gamma_j$ with larger cost saving rate $C_i^W \mu_i$; otherwise, higher priority is given to patient type $i \in \Gamma_j$ with the smallest admission rate $P_i(t)$. Once a patient type is chosen, inside the queue of that patient type the FCFS rule is still adopted. It is worth mentioning that the equity-of-access constraint is useful to ensure fairness when optimizing the inpatient bed allocation. Without such a hard constraint, it is possible that the optimal bed allocation will allocate more beds to departments whose patients' waiting cost and rejection cost are higher and the nursing cost is lower. An extreme case is that some departments will not be allocated beds at all. With the equity-of-access constraint, such a solution will be infeasible, and the admission rate among all departments will not be far too different. See more details of the impact of δ in Gong et al. (2022). Finally, the optimization of inpatient bed allocation can be formulated as

$$\min_{\substack{J,\{\Gamma_1,\cdots,\Gamma_J\},\\ \{n_1,\cdots,n_I\},\\ \{o_1,\cdots,o_J\}}} \left\{ \lim_{t\to\infty} \frac{1}{t} \sum_{i\in\mathbb{I}} \left(C_i^R R_i(t) + C_i^W W_i(t)\right) + \sum_{i\in\mathbb{I}} \xi_i \lceil \varphi \cdot n_i \rceil + \sum_{j\in\mathbb{J}} \xi_{\Gamma_j} \lceil \varphi \cdot o_j \rceil \right\} \tag{1}$$

s.t.

$$\sum_{i\in\mathbb{I}} n_i + \sum_{j\in\mathbb{J}} o_j = B \tag{2}$$

$$\lim_{t\to\infty}(P_i(t) - P_h(t)) \le \delta, \quad \forall i, h \in \mathbb{I} \tag{3}$$

$$n_i, o_j \in \mathbb{N}, \quad \forall i \in \mathbb{I}, j \in \mathbb{J} \tag{4}$$

$$J \in \mathbb{I} \tag{5}$$

Note that $\lim_{t\to\infty} \frac{1}{t} \sum_{i\in\mathbb{I}} \left(C_i^R R_i(t) + C_i^W W_i(t)\right)$ is equivalent to the expectation of sum of patients' waiting costs and rejecting costs per time unit, and $\lim_{t\to\infty}(P_i(t) - P_h(t))$ is equivalent to the expectation of admission rate difference, in the steady state of the queueing network. They both need to be estimated by running a simulation model in finite time for certain replications and calculating the average over replications. To summarize, the objective (1) is to minimize the total cost per time unit, which consists of the expectation of the sum of patients' waiting costs and rejecting costs per time unit, the nursing cost of all dedicated beds per time unit, and the nursing cost of all shared beds per time unit. The constraint (2) states the total number

of inpatient beds to be allocated, constraint (3) means the expectation of admission rate difference among all departments should not exceed a certain level, constraint (4) means the number of dedicated beds for each department and the number of shared beds for each cluster are natural numbers (including 0), and constraint (5) means the number of clusters is an integer between 1 and the total number of departments.

As emphasized before, the above specific problem and model are directly from Gong et al. (2022) and are used to illustrate what kind of inpatient bed allocation problems we are focusing on. The proposed algorithm is not especially customized for this specific problem. And as demonstrated later, it performs fairly well for a variety of problems including this one without particular parameter tuning. So it is convincing that the proposed algorithm is capable for more general inpatient bed allocation problems under CO configuration where the objective and (some) constraints need to be evaluated via stochastic simulation. For example, one may consider different structures and admission/operation rules of healthcare systems, different patient related issues (e.g., deterioration during waiting), and even different objective(s) or constraint(s). The performance of our proposed algorithm should still be satisfying.

4. ESB&B and Adaption to Inpatient Bed Allocation

We first summarize the original ESB&B algorithm proposed in Xu (2009) and Xu and Nelson (2013), and then a modest improvement for the ESB&B algorithm is proposed. After that, we introduce how to adapt the ESB&B algorithm to the inpatient bed allocation problems under CO configuration.

4.1 Original ESB&B Algorithm

The original ESB&B algorithm proposed in Xu (2009) and Xu and Nelson (2013) aims to solve the discrete simulation optimization problems $\max_{x\in\mathcal{X}} \mathbb{E}[Y(\boldsymbol{x})]$, where $\mathcal{X}$ is the intersection of the integer lattice $\mathbb{Z}^d$ with a closed set in $\mathbb{R}^d$, and $|\mathcal{X}| < \infty$, where $|\cdot|$ denotes the cardinality of a set. Note that $\mathbb{E}[Y(\boldsymbol{x})]$ cannot be analytically calculated, and it can only be estimated via i.i.d. observations of random variable $Y(\boldsymbol{x})$ via expensive stochastic simulation, which are denoted as $Y_1(\boldsymbol{x}), Y_2(\boldsymbol{x}), \cdots$. The detailed steps of the original ESB&B algorithm are presented in Algorithm 1.

Partition means diving a region into a group of smaller non-empty subregions that are mutually exclusive and collectively exhaustive. In the original ESB&B algorithm, the dimension (or more precisely, coordinate axis) with the maximal span is selected, and the region is divided into ω (a predetermined value) approximately even parts by cuts perpendicular to that dimension, which is subject to the rounding issue (see more details in online Appendix A of Xu and Nelson (2013)). We refer to such a partitioning strategy as the maximal edge partitioning strategy. In iteration $k \geq 1$, the number of solutions to be sampled from each subregion that does not belong to the record set, i.e., $\theta(X^P)$ for each $X^P \in \mathcal{P}_k \backslash \{R^k\}$ is randomly determined based on the previous information. In particular, $\{\theta(X^P) : X^P \in \mathcal{P}_k \backslash \{R^k\}\}$ is a random sample from a multinomial distribution with ϑ_O

Algorithm 1 Original ESB&B Algorithm

Step 0 *Initialization*.

Set iteration counter $k = 0$, initial partition $\mathcal{P}_0 = \{\mathcal{X}\}$, and record set $R^0 = \mathcal{X}$.

Step 1 *Partitioning*:

If the record set R^k is not a singleton, construct a partition of the record set, $\mathcal{P}''_k(R^k)$, and define the new full partition by $\mathcal{P}'_k = \left(\mathcal{P}_k \backslash \{R^k\}\right) \bigcup \mathcal{P}''_k(R^k)$; otherwise, set $\mathcal{P}''_k(R^k) = \{R^k\}$ and $\mathcal{P}'_k = \mathcal{P}_k$.

Step 2 *Bounding*.

2.1. Solution sampling: For each subregion $X^P \in \mathcal{P}''_k(R^k)$, randomly and uniformly sample ϑ_R solutions. If $k > 0$, for subregion $X^P \in \mathcal{P}_k \backslash \{R^k\}$ randomly and uniformly sample $\theta(X^P)$ solutions, where $\theta(X^P)$ depends on the observations at solutions in $\mathfrak{S}^{k-1}$. Aggregate all of the sampled solutions into a set, S^k. If $k > 0$, let $\mathfrak{S}^k = \mathfrak{S}^{k-1} \cup S^k$; else let $\mathfrak{S}^k = S^k$.

2.2. Bound estimation: Simulate Δn_{F} observations from each solution in S^k that has not been encountered before and simulate Δn_{A} additional observations from each solution that has been encountered before. For subregion $X^P \in \mathcal{P}'_k$, set $\eta^{k+1}(X^P) = \max_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^k} \bar{Y}(\boldsymbol{x})$, where $\bar{Y}(\boldsymbol{x})$ is the cumulative sample mean of all observations at solution $\boldsymbol{x}$.

Step 3 *Updating partition and record set*.

Update the record set $R^{k+1} = \operatorname{argmax}_{X^P \in \mathcal{P}'_k} \eta^{k+1}(X^P)$ and partition $\mathcal{P}_{k+1} = \mathcal{P}'_k$. Set $k = k + 1$ and go to Step 1.

trials and success probabilities $\boldsymbol{\rho} = \{\rho(X^P) : X^P \in \mathcal{P}_k \backslash \{R^k\}\}$, where $\vartheta_O \in \mathbb{Z}^+$ is predetermined and $\boldsymbol{\rho}$ is calculated based on the observations at solutions in $\mathfrak{S}^{k-1}$. The $\mathfrak{S}^k$ is a set of all solutions that have been sampled through iteration k. The intuition behind the calculation of $\boldsymbol{\rho}$ is to sample more solutions in a subregion with better potential, which is important to the finite sample performance of the algorithm. Also note that the choice of $\boldsymbol{\rho}$ will not affect the convergence of the algorithm as long as all elements of $\boldsymbol{\rho}$ are bounded away from zero. Xu and Nelson (2013) consider several strategies and find from numerical experiments that the probability-based allocation using normal distribution performs better. Here we summarize this strategy without derivation details. For each $X^P \in \mathcal{P}_k \backslash \{R^k\}$, let $\eta^* = \max_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^{k-1}} \bar{Y}(\boldsymbol{x})$ be the optimal value through iteration k, $m = |X^P \cap \mathfrak{S}^{k-1}|$ be the total number of solutions in the subregion that have been sampled and simulated through iteration k. The sample mean and sample variance of all solutions in the subregion are denoted as $\bar{Y} = m^{-1} \sum_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^{k-1}} \bar{Y}(\boldsymbol{x})$ and $S_Y^2 = (m-1)^{-1} \sum_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^{k-1}} (\bar{Y}(\boldsymbol{x}) - \bar{Y})^2$, respectively. Let $S_p^2 = v^{-1} \sum_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^{k-1}} \sum_{s=1}^{n(\boldsymbol{x})} \left(Y_s(\boldsymbol{x}) - \bar{Y}(\boldsymbol{x})\right)^2$ the pooled sample variance within the subregion, where $v = \sum_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^{k-1}} (n(\boldsymbol{x}) - 1)$ and $n(\boldsymbol{x})$ denotes the total number of observations obtained at $\boldsymbol{x}$. Also, define the effctive degrees of freedom n^* such that $1/n^* = m^{-2} \sum_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^{k-1}} 1/n(\boldsymbol{x})$. If $m < |X^P|$, calculate quantiles α_1 and α_2 that solve

$$t_{1-\alpha_1, m-1} = t_{1-\alpha_2, v} = \frac{\eta^* - \bar{Y}}{S_Y \sqrt{1 + (1/m)} + \left(S_p / \sqrt{n^*}\right)}$$

and let $p(X^P) = \alpha_1 + \alpha_2$; otherwise calculate $\alpha(\boldsymbol{x})$ that solves

$$t_{1-\alpha(\boldsymbol{x}), n(\boldsymbol{x})-1} = \frac{\eta^* - \bar{Y}(\boldsymbol{x})}{S_P / \sqrt{n^*}}$$

and let $p(X^P) = \max_{\mathbf{x} \in X^P \cap \mathfrak{S}^{k-1}} \alpha(\mathbf{x})$. Let $T = \sum_{X^P \in \mathcal{P}_k \backslash \{R^k\}} \max\left\{\epsilon, p(X^P)\right\}$, where ϵ is a predetermined constant that $0 < \epsilon \ll 1$.

For each $X^P \in \mathcal{P}_k \backslash \{R^k\}$, calculate $\rho(X^P) = \max\left\{\epsilon, p(X^P)\right\}/T$.

The ESB&B algorithm summarized above is globally convergent under mild conditions. Specifically, assume $|\mathbb{E}[Y(\boldsymbol{x})]| < \infty$ and $\mathrm{Var}(Y(\boldsymbol{x})) < \infty$ for all $\boldsymbol{x} \in \mathcal{X}$. Denote $\mathcal{X}^* = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[Y(\boldsymbol{x})]$. Then with probability one, there exists an iteration number k_0 such that for all $k \le k_0$, the record sets R^k produced by the ESB&B algorithm summarized above are singletons and $R^k \subseteq \mathcal{X}^*$. It is worth mentioning that the ESB&B algorithm does not actually prune any subregion, since there are no exact upper bound and lower bound for each subregion but the estimated ones. If one prunes some subregions based on the estimated upper and lower bounds, s/he will have some probability that mistakenly prunes the subregions containing $\mathcal{X}^*$, in which case the asymptotic convergence will no longer hold. Instead, the ESB&B algorithm inherits the spirit of B&B methods by dividing the feasible region into smaller and smaller subregions and allocating the sampling budget according to the potential (i.e., estimated bounds) of the subregions.

For subregions whose estimated bounds show that they are inferior, although they are not pruned directly, their probability of being sampled in the future will be smaller than other subregions. Note that the convergence analysis holds for a variety of partitioning strategies and solution sampling strategies. However, it can be anticipated that the choice of partitioning strategy and solution sampling strategy will affect the finite sample performance of the ESB&B algorithm. Xu and Nelson (2013) consider the maximal edge partitioning strategy for simplicity. However, such a strategy is static and its performance may not be the best. Intuitively, a partitioning strategy that incorporates the previous sampling information should be better than such a static strategy, since it can locate the most promising subregions more efficiently. Besides, Xu and Nelson (2013) consider sampling ϑ_R solutions for each subregion of the partition of the record set and sampling ϑ_O solutions among all subregions that do not belong to the record set according to the multinomial distribution. For the subregions of the partition of the record set, the sampling budget is simply fixed, and it may be improved by also considering the potential of each subregion.

4.2 A Modest Improvement for ESB&B

As mentioned above, we consider improving the finite sample performance of the original ESB&B algorithm by adopting an adaptive partitioning strategy and a global solution sampling strategy. We call the ESB&B algorithm with such Adaptive partitioning strategy and Global solution sampling strategy ESB&B-AG algorithm. Since the new partitioning strategy and solution sampling strategy are still in the analysis framework of the asymptotic convergence, the established global convergence of the original ESB&B algorithm still holds for the ESB&B-AG algorithm.

4.2.1 Adaptive partitioning strategy

In the original ESB&B algorithm, the maximal edge partitioning strategy is used to cut the record set into ω parts, which only considers the geometric feature of the record set and does not utilize the previous sampling information. Inspired by the work of Lu et al.

(2021), we adopt the following adaptive partitioning strategy. Consider a general record set R that is represented as follows:

$$\begin{aligned} h_j(\boldsymbol{x}) &\le b_j, \quad j = 1, \cdots, q \\ l_i \le x_i &\le u_i, \quad i = 1, \cdots, d \\ x_i, l_i, u_i &\in \mathbb{Z}, \quad i = 1, \cdots, d \end{aligned}$$

We consider all the possible partitions that divide R into $\omega = 2$ parts using cuts perpendicular to one coordinate axis). Specifically, define set $\{R_{wk}, R'_{wk}\}$ for $k = 1, \cdots, u_w - l_w$, $w = 1, \cdots, d$, such that R_{wk} is represented as

$$\begin{aligned} h_j(\boldsymbol{x}) &\le b_j, \quad j = 1, \cdots, q \\ l_i \le x_i &\le u_i, \quad i = 1, \cdots, d \text{ and } i \ne w \\ l_i \le x_i &\le l_i + k - 1, \quad i = w \\ x_i, l_i, u_i &\in \mathbb{Z}, \quad i = 1, \cdots, d \end{aligned}$$

and R'_{wk} is represented as

$$\begin{aligned} h_j(\boldsymbol{x}) &\le b_j, \quad j = 1, \cdots, q \\ l_i \le x_i &\le u_i, \quad i = 1, \cdots, d \text{ and } i \ne w \\ l_i + k \le x_i &\le u_i, \quad i = w \\ x_i, l_i, u_i &\in \mathbb{Z}, \quad i = 1, \cdots, d \end{aligned}$$

Define a set $\mathfrak{P}$ which is a collection of all the sets $\{R_{wk}, R'_{wk}\}$ such that $R_{wk} \cap \mathfrak{S} \ne \emptyset$ and $R'_{wk} \cap \mathfrak{S} \ne \emptyset$, where $\mathfrak{S}$ denotes the set of all sampled solutions right before the partition. Note that $\mathfrak{P}$ will be nonempty as long as $|R \cap \mathfrak{S}| \ge 2$. Then, the adaptive partitioning strategy chooses the partition in $\mathfrak{P}$ that the most similar solutions are likely to be grouped together. Mathematically, the final chosen partition is

$$\underset{\mathcal{P}''(R) \in \mathfrak{P}}{\operatorname{argmin}} \sum_{X^P \in \mathcal{P}''(R)} \sum_{\boldsymbol{x} \in X^P \cap \mathfrak{S}} (\bar{Y}(\boldsymbol{x}) - \bar{Y})^2$$

which can be simply solved by enumeration, since $|\mathfrak{P}|$ is linear to the dimensionality d. Compared with the static maximal edge partitioning strategy, such adaptive partitioning strategy tends to construct subregions wherein inferior solutions and superior solutions cluster respectively, so that the subregion with a cluster of inferior solutions can be less likely sampled while the subregion with a cluster of superior solutions can be more likely sampled in the future.

4.2.2 Global solution sampling strategy

In the original ESB&B algorithm, ϑ_R solutions are sampled in each subregion $X^P \in \mathcal{P}''_k(R^k)$, and ϑ_O solutions are sampled among all subregions that do not belong to the record set according to the multinomial distribution, where the probability in each subregion is determined by its potential estimated via observations in it. Intuitively speaking, for subregion $X^P \in \mathcal{P}''_k(R^k)$ it should be beneficial if the sampling budget is also allocated based on its potential instead of simply fixed. Recall the strategy of constructing the multinomial sampling distribution for subregions that do not belong to the record set described in Section 4.1. It is clear that same method can be applied to $X^P \in \mathcal{P}''_k(R^k)$ to calculate $p(X^P)$. So, it is totally workable if one considers allocating the $\omega\vartheta_R + \vartheta_O$ solution sampling budget globally to the entire subregions in $\mathcal{P}'_k$. In particular, after calculating $p(X^P)$ for each $X^P \in \mathcal{P}'_k$, one now needs to define $T = \sum_{X^P \in \mathcal{P}'_k} \max\{\epsilon, p(X^P)\}$ and calculate $\rho(X^P) = \max\{\epsilon, p(X^P)\}/T$ for each $X^P \in \mathcal{P}'_k$. Then, let $\{\theta(X^P) : X^P \in \mathcal{P}'_k\}$ be a random sample from a multinomial distribution with $\omega\vartheta_R + \vartheta_O$ trials and success probabilities $\boldsymbol{\rho} = \{\rho(X^P) : X^P \in \mathcal{P}'_k\}$, and randomly and uniformly sample $\theta(X^P)$ solutions for subregion $X^P \in \mathcal{P}'_k$. Note that in iteration 0, since

no solution has been sampled, one can simply let $\rho(X^P) = 1/|\mathcal{P}'_k|$. Using such a global solution sampling strategy may help the ESB&B algorithm to concentrate more on subregions that have higher potential estimated from previous observations, which may increase the efficiency when searching for a better solution. The detailed steps of the improved ESB&B algorithm, i.e., ESB&B-AG algorithm, is presented in Algorithm 2.

4.3 Adaption to Inpatient Bed Allocation

As discussed in the introduction, the inpatient bed allocation problems under CO configuration are combinatorial optimization problems where the objective function and (some) constraints need to be estimated via expensive and noisy stochastic simulation. The dimensionality of the feasible solution varies when the number of clusters varies. For the ESB&B algorithm, although the theoretical analysis does not restrict the form of the feasible region, the original ESB&B in Xu (2009) and Xu and Nelson (2013) is designed for the feasible region that is the intersection of the integer lattice $\mathbb{Z}^d$ with a closed set in $\mathbb{R}^d$. Therefore, some adaption is required in order to make the original ESB&B algorithm or the improved ESB&B-AG algorithm applicable to the inpatient bed allocation problems under CO configuration. Moreover, how to efficiently sample solutions uniformly in the feasible region is also an issue to be addressed, since now the region is no longer simply a hypercube with linear constraints.

4.3.1 Two-layer branches

Note that the dimensionality of the feasible solution will be fixed once the number of clusters is fixed. So, a natural idea is to consider the

Algorithm 2 ESB&B-AG Algorithm

Step 0 *Initialization*.

Set iteration counter $k = 0$, initial partition $\mathcal{P}_0 = \{\mathcal{X}\}$, and record set $R^0 = \mathcal{X}$. Let $\mathfrak{S}^0 = \emptyset$.

Step 1 *Partitioning*:

If the record set R^k is not a singleton:

(a) If $|R^k \cap \mathfrak{S}^k| \geq 2$, apply the adaptive partitioning strategy to construct a partition of the record set, $\mathcal{P}''_k(R^k)$;

(b) otherwise, apply the maximal edge partitioning strategy with $\omega = 2$ to construct $\mathcal{P}''_k(R^k)$.

And define the new full partition by $\mathcal{P}'_k = \left(\mathcal{P}_k \backslash \{R^k\}\right) \bigcup \mathcal{P}''_k(R^k)$. Otherwise, set $\mathcal{P}''_k(R^k) = \{R^k\}$ and $\mathcal{P}'_k = \mathcal{P}_k$.

Step 2 *Bounding*.

2.1. Solution sampling: For each subregion $X^P \in \mathcal{P}'_k$, randomly and uniformly sample $\theta(X^P)$ solutions, where $\theta(X^P)$ is determined as described in Section 4.2.2. Aggregate all of the sampled solutions into a set, S^k. If $k > 0$, let $\mathfrak{S}^k = \mathfrak{S}^{k-1} \cup S^k$; else let $\mathfrak{S}^k = S^k$.

2.2. Bound estimation: Simulate Δn_{F} observations from each solution in S^k that has not been encountered before and simulate Δn_{A} additional observations from each solution that has been encountered before. For subregion $X^P \in \mathcal{P}'_k$, set $\eta^{k+1}(X^P) = \max_{\boldsymbol{x} \in X^P \cap \mathfrak{S}^k} \bar{Y}(\boldsymbol{x})$, where $\bar{Y}(\boldsymbol{x})$ is the cumulative sample mean of all observations at solution $\boldsymbol{x}$.

Step 3 *Updating partition and record set*.

Update the record set $R^{k+1} = \operatorname{argmax}_{X^P \in \mathcal{P}'_k} \eta^{k+1}(X^P)$ and partition $\mathcal{P}_{k+1} = \mathcal{P}'_k$. Set $k = k + 1$ and go to Step 1.

two-layer branches as follows. For the inpatient bed allocation problems under CO configuration, denote its feasible region as $\mathcal{X}$. Recall that the total number of departments is I. Initially, the feasible region $\mathcal{X}$ is divided into $I-1$ parts, wherein the number of clusters J is $1, \cdots, I-1$, respectively. Note that there is no need to explicitly consider the subregion wherein the number of clusters is 0 or I, because such a case means the fully dedicated configuration, which is contained in the subregion wherein the number of clusters is 1 and the number of shared beds for cluster 1 is 0. All the future partitions are within these $I-1$ subregions, which can ensure that the dimensionality in each subregion is consistent. It is like that we predetermine the first-layer branches (which are the branches directly from the root node $\mathcal{X}$, and only let the algorithm adaptively determine the branches under this layer. In practice, one only needs to let the partition of the record set $R^0 = \mathcal{X}$, i.e., $\mathcal{P}_0''(R^0)$, be exactly the collection of the aforesaid $I-1$ parts in iteration 0, and in the later iterations the partition is normally conducted as described before.

4.3.2 Structure of solutions

When the number of clusters is fixed, the solution can be explicitly written. For example, suppose $I = 3$ and given that the number of clusters is 2, then a feasible (only in terms of the maximal number of inpatient beds) solution can be written as $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ such that $\sum_{i=1}^{5} x_i = B$, where x_1, x_2, x_3 denote the number of dedicated inpatient beds for departments 1, 2 and 3 respectively, x_4, x_5 denote the number of shared inpatient beds in clusters 1 and 2 respectively, and x_6, x_7, x_8 denote how the 3 departments are partitioned into 2 clusters. For (x_6, x_7, x_8) there are a variety of expression forms. One simple way is to assign integers to (x_6, x_7, x_8) to indicate to which cluster a department belongs. For example, $(x_6, x_7, x_8) = (1, 2, 1)$ means departments 1 and 3 form one cluster, and department 2 forms one cluster. Note that when a cluster only contains one department, the shared beds are actually also dedicated to that department. Also note that the above expression will produce duplicated solutions. For example, $(x_6, x_7, x_8) = (2, 1, 2)$ also means the same partition of departments. To reduce the feasible solution, such duplications should be eliminated using some rule. A simple rule is to assign the cluster numbers according to the ascending order of the minimal department index in each cluster. Under such rule, $(x_6, x_7, x_8) = (2, 1, 2)$ is not allowed and $(x_6, x_7, x_8) = (1, 2, 1)$ is the unique expression when departments 1 and 3 form one cluster and department 2 forms one cluster. Besides, to make the expression of the feasible region easier, other constraints in addition to the number of inpatient beds are moved to the objective function. For example, for problem (1), the hard constraint (3) can be moved to the objective function with an indicator function (which returns zero when the constraint is satisfied and infinity otherwise).

4.3.3 Sampling of solutions

Based on the solution structure described in Section 4.3.2, to randomly and uniformly sample a solution in a subregion that is contained in one of the $I-1$ parts defined in Section 4.3.1 involves two steps. The first step is to sample the bed allocation to each department and each cluster, subject to the constraints on each

number and the total number. The acceptance-rejection technique is the most straightforward choice for such a purpose. However, in this case, the acceptance rate may be unacceptably low. We adopt the Dirichlet-rescale (DRS) algorithm proposed in Griffin et al. (2020), which can randomly and uniformly generate a d-dimensional vector $\boldsymbol{u} = (u_1, \cdots, u_d)$ that $\sum_{i=1}^{d} u_i = U$ and $l_i \le u_i \le u_i, i = 1, \cdots, d$, for given l_i, u_i and U with high efficiency. Rounding is required since the number of inpatient beds is an integer. After rounding, the summation of the integers may be not equal to B. We simply do the following. If the summation exceeds B, then randomly pick a positive element and decrease it by one; repeat until the summation equals B. Otherwise, randomly pick an element that could be positive or zero and increase it by one; repeat until the summation equals B. The second step is to sample the partition of departments under a given number of clusters, say $I' \in \{1, \cdots, I-1\}$. To ensure efficiency, a random permutation of the department indices $1, \cdots, I$ is generated and then $I'-1$ cuts are randomly added to divide the index sequence into I' non-empty parts. After that, the rule of assigning the cluster numbers as described in Section 4.3.2 is applied to determine the values of the corresponding elements in the solution.

5. Numerical Experiments

The purpose of this section is twofold. The first purpose is to investigate the range of applications of the ESB&B and ESB&B-AG algorithms compared with the SMA of Gong et al. (2022). Recall that the SMA is highly customized for the specific inpatient bed allocation problem (1), so it will be anticipated that SMA is not suitable for other discrete simulation optimization problems. This is to be verified via numerical experiments. On the other hand, since the ESB&B and ESB&B-AG algorithms are designed for general discrete simulation optimization problems, whether they indeed perform fairly well for a variety of problems including inpatient bed allocation problems is also to be verified. The second purpose is to compare the performance of the ESB&B-AG algorithm with the original ESB&B algorithm to show the practical value of the improvement proposed in this paper. Throughout the numerical experiments, both ESB&B and ESB&B-AG algorithms are implemented in Python with $\omega = 2$, $\Delta n_{\mathrm{F}} = 10$, $\Delta n_{\mathrm{A}} = 2$, $\vartheta_R = 10$, $\vartheta_O = 20$, $\epsilon = 0.1$; the exactly same implementation of SMA with the same parameters as in Gong et al. (2022) is used. Since the SMA adopts a GA framework in its outer layer, we also try the basic GA, where binary encoding is employed with a population size of 50, the crossover probability is 0.9, and the mutation probability is 0.005 as suggested in Schaffer et al. (1989), and for each solution in the population 5 observations are simulated in each iteration. Note that the ESB&B is originally defined for a maximization problem while the SMA is for a minimization problem. Also note that a maximization problem can be easily transformed to a minimization problem by adding a negative sign, and vice versa. So in the following, we will implicitly do such transformation without specific mention.

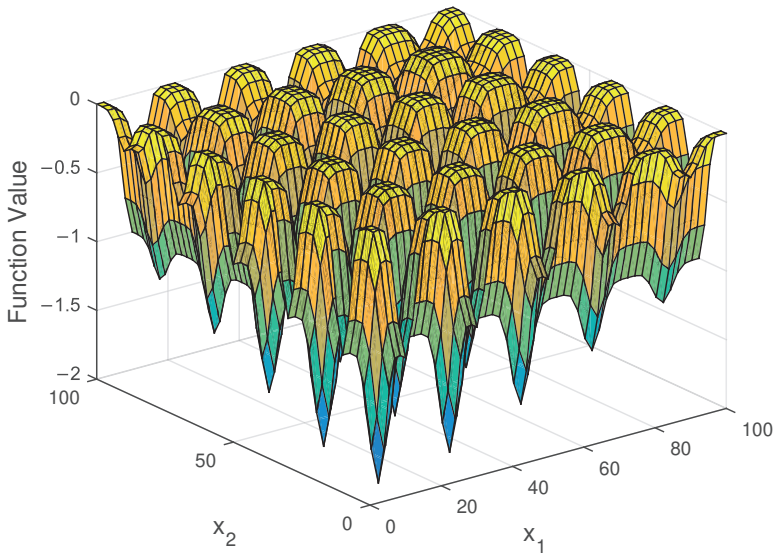

Figure 2 The Miller and Shaw Function in Two Dimensions

5.1 Results for Synthetic Problems

The first test problem is exactly from Xu (2009) and Xu and Nelson (2013) that originally propose the ESB&B algorithm, which is a modification of the multimodal function F_2 used in Miller and Shaw (1996). Specifically, consider

$$\min_{(x_1,x_2)\in\mathcal{X}} -\frac{\sin^6(0.05\pi x_1)}{2^{2((x_1-10)/80)^2}} - \frac{\sin^6(0.05\pi x_2)}{2^{2((x_2-10)/80)^2}} \quad (6)$$

where $\mathcal{X} = \mathbb{Z}^2 \cap [0, 100]^2$, and the objective function is observed with normally distributed noises with mean of 0 and a standard deviation of 0.3. This problem has a global optima $(10, 10)$ with the objective value of -2. The response surface is bumpy with 25 local optimal, which is shown in Figure 2. It is a challenging test problem for simulation optimization algorithms.

The second test problem is to minimize the Griewank function, which is adapted from Salemi et al. (2019). Specifically, consider

$$\min_{(x_1,\ldots,x_d)\in\mathcal{X}} \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (7)$$

where $\mathcal{X} = \mathbb{Z}^d \cap [-50, 50]^d$, and the objective function is observed with normally distributed noises with a mean of 0 and a standard deviation of 0.1. This problem has a global optima $(0, \cdots, 0)$ with the objective value of 0. It also has multiple local optima and it is a commonly used test problem for optimization algorithms.

Figures 3 and 4 show the performance of GA, SMA, original ESB&B, and ESB&B-AG for the Miller and Shaw problem and Griewank problem with $d = 3$, respectively. Each curve is the average of 50 repetitions of each algorithm, and the horizontal axis is the sample size, i.e., the number of noisy observations of the objective function. Several conclusions can be drawn from the results. First, both ESB&B and ESB&B-AG algorithms perform fairly well in two problems, while the performance of SMA is less satisfying. For the Miller and Shaw problem, SMA quickly finds good solutions in the early stage but does not improve too much afterward. For the Griewank problem, the performance of SMA is worse and it behaves just like the basic GA. It seems like some features in SMA that are highly customized for problem

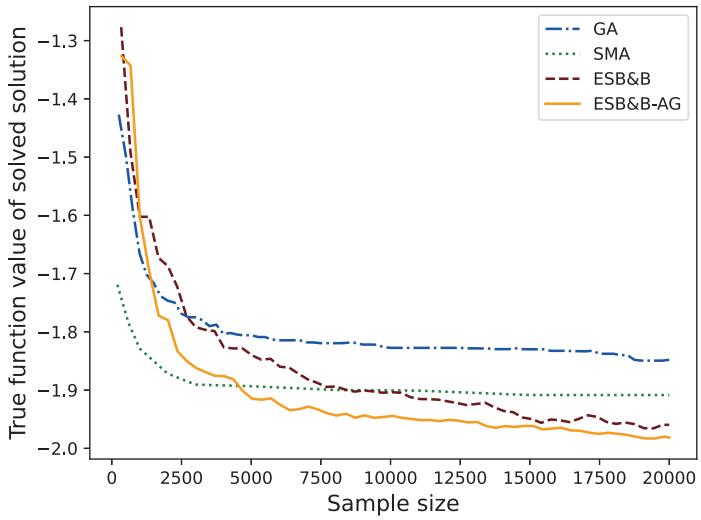

Figure 3 Performance of GA, SMA, ESB&B, ESB&B-AG for the Miller and Shaw Problem

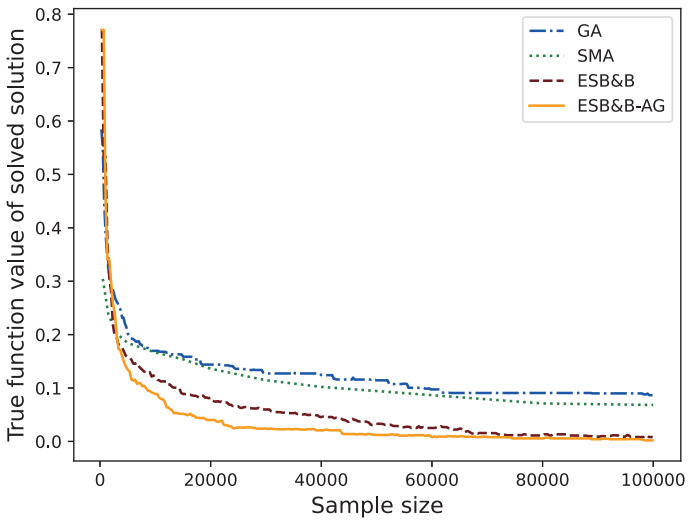

Figure 4 Performance of GA, SMA, ESB&B, ESB&B-AG for the Griewank Problem

(1) do not work for these general simulation optimization problems. Second, the ESB&B-AG evidently outperforms the original ESB&B on the two problems.

It is desirable to establish the complexity of the algorithm. For deterministic optimization algorithms, it is sometimes possible to derive the computation complexity. But for simulation optimization algorithms, deriving the computation complexity is both impossible (since the computation in the algorithm is too complicated) and unnecessary (since the most time-consuming part is running the simulation model). So, a more reasonable and practical metric is the sample complexity, where one sample means running the simulation model once to obtain one noisy observation. In particular, one can consider the sam-

Table 1 Average Sample Size of Different Algorithms for the Griewank Problem

	$d = 1, N_1 = 20000$		$d = 2, N_2 = 40000$		$d = 3, N_3 = 100000$	
	exact opt.	near opt.	exact opt.	near opt.	exact opt.	near opt.
GA	673 (4 failed)	442	17500 (27 failed)	6211	12000 (29 failed)	23145 (5 failed)
SMA	571	60	6388 (9 failed)	1170	– (30 failed)	44460
ESB&B	1666	933	23166	4749	53245	24799
ESB&B-AG	1358	798	15526	3931	29345	11801

ple size required by a simulation algorithm to reach a certain level of solution accuracy; see for example Zhang et al. (2023ab). We conduct some numerical experiments to compare the sample size of different algorithms for the Griewank problem with varying dimensions. More specifically, for $d = 1, 2, 3$, each algorithm is run until reaching the exact optimum (which means the found solution is the true global optima) or near optimum (which means the true function value of a found solution is smaller than 0.1 while the true global optimum is 0). The algorithm will also stop if the sample size reaches the limit N_d, which is denoted as a failure.

At each solution accuracy level, each algorithm is repeatedly run 30 times, and the results are reported in Table 1. The average sample size is computed among the successful replications. If the number of failures among 30 replications is larger than 0, it is also reported (shown in parentheses). When the number of failures is 30, the average sample size is not defined (denoted with –). It can be observed that when $d = 1$, the sample size of SMA is the lowest at each solution accuracy level, which means its sample efficiency is the highest without being trapped in the local optimum. GA has the second highest sample efficiency, however, it fails sometimes to find the global optimum. However, as d increases (which means the problem becomes more difficult), the performance of SMA deteriorates largely. For $d = 2$, it has a large probability of being trapped in the local optimum; while for $d = 3$, it can not find the global optimum among 30 replications, and its sample efficiency becomes the lowest for finding the near optimum. GA also deteriorates similarly. In contrast, both ESB&B and ESB&B-AG perform robustly in all cases. They can always find the global optimum, and the sample size becomes more competitive at each solution accuracy level when the problem becomes more difficult. Moreover, ESB&B-AG consistently outperforms ESB&B in terms of sample size.

All the above numerical results show that SMA is not suitable for general simulation optimization problems, since it is highly customized for a specific inpatient bed allocation problem. It may find some good solution quickly at early stage, but may get trapped in some local optimum, since there is no theoretical guarantee of global convergence for it. The plain GA performs worse than SMA without particular parameter tuning. Both ESB&B and ESB&B-AG perform fairly well without par-

Table 2 Department-Related Parameters

Department	i	λ_i	μ_i	C_i^R	C_i^W	ξ_i	D_i
Cardiac surgery	1	3.28	21.79	4350.58	579.80	196.5	10
Minimally invasive surgery	2	3.53	2.54	503.45	289.90	192.4	6
Gastrointestinal surgery	3	7.85	12.67	2557.33	434.85	200.1	10
Ophthalmology	4	6.98	4.36	869.47	289.90	167.6	6
Burns surgery	5	2.07	15.56	3095.00	579.80	193.3	4

ticular parameter tuning, since they are not highly customized algorithms and have theoretical convergence property.

5.2 Results for Inpatient Bed Allocation Problem

We now investigate the practical performance of the ESB&B-AG algorithm on inpatient bed allocation problems under CO configuration. The problem (1) of Gong et al. (2022) is used as an illustrative example. Specifically, the problem instance is exactly from Gong et al. (2022), which is based on real data collected from a public hospital in Shanghai, China. In this problem instance, the time unit is day, there are $I = 5$ departments with $B = 240$ beds to allocate, and the related parameters are presented in Table 2. Each ξ_i is the sum of the average daily wage of nurses in Shanghai ξ^W (146.2 RMB) and the training cost amortized into one day, which is denoted as ξ_i^T. For the 5 departments, ξ_i^T equals 50.3, 46.2, 53.9, 21.4, and 47.1 RMB, respectively. Then the ξ_{Γ_j} is set as $\xi_{\Gamma_j} = (1 + 0.1 \times (|\Gamma_j| - 1))\xi^W + \sum_{i\in\Gamma_j} \xi_i^T$. For f_i and g_i, $i = 1, \cdots, 5$, empirical distributions that are constructed from real data are used. Moreover, $\varphi = 0.25$, and $\delta = 0.1$. The exact same implementation of the simulation model for this problem instance as in Gong et al. (2022) is used. When estimating the objective function in (1) and the constraint (3), we call the output after running the simulation model for 10^4 days (which balances the simulation cost and the simulation fidelity) as one noisy observation (i.e., one sample). And we will replicate the simulation run independently to obtain independent noisy observations and then the sample average will be calculated for estimation.

Since we have shown that the ESB&B-AG algorithm outperforms the original ESB&B algorithm, here we only investigate the performance of the ESB&B-AG algorithm for the inpatient bed allocation problem and compare it with the SMA that is highly customized for this problem. Since the basic GA is also a general optimization algorithm, we also try it in this case. Note that one important feature of SMA is that it can find good initial solutions by utilizing detailed information of the specific inpatient bed allocation problem. However, the ESB&B-AG algorithm simply starts from randomly generated solutions. As a remedy, we let the ESB&B-AG algorithm first solve the problem within a smaller feasible region (under WF configuration instead of CO configuration). Then after 4×10^4 samples are used, switch to the original problem (i.e., under CO configuration) with the so far found solution as an initial solution. For SMA, SA, and ESB&B-AG, we run the algorithm until 10^5 samples are exhausted and output the final found solu-

Table 3 Achieved Total Costs when 10^5 Samples Are Exhausted Using SMA, GA and ESB&B-AG

Algorithm	Mean	Standard deviation	Gap of mean with SMA
SMA	2.95×10^4	1.7×10^2	–
GA	3.28×10^4	3.0×10^3	11.36%
ESB&B-AG	2.99×10^4	1.4×10^2	1.54%

tion. To evaluate the solution quality, run the high-fidelity simulation model with 10^6 days (which is time-consuming and not affordable during the optimization) at this solution and record the simulated objective function value (i.e., the achieved total cost). Each algorithm is repeated 30 times and the mean and standard deviation of the achieved total costs are reported in Table 3. It can be seen that the performance of ESB&B-AG is close to SMA. Given that ESB&B-AG is a general optimization algorithm while SMA is highly customized, the performance of ESB&B-AG is fairly good. Note that also as a general optimization algorithm, GA has a much larger gap with SMA compared to ESB&B-AG. Recall that ESB&B-AG also has stable and satisfying performance on the synthetic problems, there is reason to believe that ESB&B-AG will also perform well on other inpatient bed allocation problems (e.g., when the objective, constraints, and/or operation rules of the healthcare system are different to those in problem (1)).

6. Conclusions

In this paper, we propose to use the improved ESB&B algorithm (i.e., ESB&B-AG algorithm) to solve the inpatient bed allocation problems under CO configuration where the objective function and (some) constraints need to be estimated via expensive and noisy stochastic simulation. Compared to the SMA which is a highly customized heuristic algorithm for one specific inpatient bed allocation problem, the original ESB&B algorithm and the ESB&B-AG algorithm are theoretically convergent and suitable for relatively general problems, and the performance of the ESB&B-AG algorithm on the inpatient bed allocation problem for which SMA is customized is close to SMA. It provides practitioners with a good choice when solving a real inpatient bed allocation problem that is not identical to the one for which SMA is customized. A possible direction to extend this paper is to improve or modify the original ESB&B algorithm more substantially for better solving combinatorial simulation optimization problems. For example, pruning the inferior branches is a core idea for classical B&B framework, but how to inherit it in ESB&B while still guarantee the global convergence is still an open question.

Acknowledgments

We would like to thank the Executive Editor, Yongbo Xiao, and the two anonymous reviewers for their insightful and detailed comments that have significantly improved this paper. We are particularly grateful to Gong et al. (2022) for providing us with the source C++ codes of the simulation model for the specific problem instance. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 72031006,

72031007, and 72394375.

Data Availability

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Conflict of Interest

We would like to disclose that Jun Luo, one of the co-authors of this study, serves as an editor for the Journal of Systems Science and Systems Engineering.

References

Ahmadi Choukolaei H, Jahangoshai Rezaee M, Ghasemi P (2024). Data-driven modeling using system dynamics simulation to provide relief in earthquake based on different scenarios. *Environmental Science and Pollution Research* 31(24): 35266-35282.

Amaran S, Sahinidis N V, Sharda B, Bury S J (2016). Simulation optimization: A review of algorithms and applications. *Annals of Operations Research* 240(1): 351-380.

Bekker R, Koole G, Roubos D (2017). Flexible bed allocations for hospital wards. *Health Care Management Science* 20: 453-466.

Best T J, Sandıkçı B, Eisenstein D D, Meltzer D O (2015). Managing hospital inpatient bed capacity through partitioning care into focused wings. *Manufacturing & Service Operations Management* 17(2): 157-176.

Bhattacharjee P, Ray P K (2014). Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: A review and reflections. *Computers & Industrial Engineering* 78: 299-312.

Fu M C, (ed)(2015). *Handbook of Simulation Optimization*, volume 216. Springer.

Ghasemi P, Goodarzian F, Simic V, Tirkolaee E B (2023). A DEA-based simulation-optimisation approach to design a resilience plasma supply chain network: A case study of the COVID-19 outbreak. *International Journal of Systems Science: Operations & Logistics* 10(1): 2224105.

Gong X, Wang X, Zhou L, Geng N (2022). Managing hospital inpatient beds under clustered overflow configuration. *Computers & Operations Research* 148: 106021.

Green L V (2004). Capacity planning and management in hospitals. In Brandeau M L, Sainfort F, Pierskalla W P (eds). *Operations Research and Health Care. A Handbook of Methods and Applications*: 15-42. Kluwer Academic Publishers, Norwell, MA.

Griffin D, Bate I, Davis R I (2020). Generating utilization vectors for the systematic evaluation of schedulability tests. In *2020 IEEE Real-Time Systems Symposium (RTSS)*: 76-88. IEEE.

Hong L J, Nelson B L (2006). Discrete optimization via simulation using COMPASS. *Operations Research* 54(1): 115-129.

Izady N, Mohamed I (2021). A clustered overflow configuration of inpatient beds in hospitals. *Manufacturing & Service Operations Management* 23(1): 139-154.

Khalili-Damghani K, Tavana M, Ghasemi P (2022). A stochastic bi-objective simulation-optimization model for cascade disaster location-allocation-distribution problems. *Annals of Operations Research* 309(1): 103-141.

Lawler E L, Wood D E (1966). Branch-and-bound methods: A survey. *Operations Research* 14(4): 699-719.

Lu J, Zhou T, Osorio C (2021). Adaptive partitioning strategy for high-dimensional discrete simulation-based optimization problems. *arXiv Preprint arXiv: 2104.14119*.

Miller B L, Shaw M J (1996). Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *Proceedings of IEEE International Conference on Evolutionary Computation*: 786-791.

Norkin V I, Ermoliev Y M, Ruszczyński A (1998a). On optimal allocation of indivisibles under uncertainty. *Operations Research* 46(3): 381-395.

Norkin V I, Pflug G C, Ruszczyński A (1998b). A branch and bound method for stochastic global optimization. *Mathematical Programming* 83(3): 425-450.

Porter M E (2010). What is value in health care. *The New England Journal of Medicine* 363(26): 2477-2481.

Salemi P, Song E, Nelson B L, Staum J (2019). Gaussian Markov random fields for discrete optimization via simulation: Framework and algorithms. *Operations Research* 67(1): 250-266.

Schaffer J D, Caruana R, Eshelman L J, Das R (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms*: 51-60.

Shirazi H, Kia R, Ghasemi P (2021). A stochastic bi-objective simulation-optimization model for plasma supply chain in case of COVID-19 outbreak. *Applied Soft Computing* 112: 107725.

Sun L, Hong L J, Hu Z (2014). Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Operations Research* 62(6): 1416-1438.

Tien J M, Goldschmidt-Clermont P J (2009). Healthcare: A complex service system. *Journal of Systems Science and Systems Engineering* 18: 257-282.

Wang Z, Chen H, Luo J, Wang C, Xu X, Zhou Y (2024). Sharing service in healthcare systems: A recent survey. *Omega* 129: 103158.

Xu W L (2009). Flexibility, lifecycle planning and simulation-based optimization in integrated supply chains. PhD thesis, Northwestern University.

Xu W L, Nelson B L (2013). Empirical stochastic branch-and-bound for optimization via simulation. *IIE Transactions* 45(7): 685-698.

Zhang H, Zheng Z, Lavaei J (2023a). Gradient-based algorithms for convex discrete optimization via simulation. *Operations Research* 71(5): 1815-1834.

Zhang H, Zheng Z, Lavaei J (2023b). Stochastic localization methods for convex discrete optimization via simulation. *Operations Research*, online.

Zhou L, Geng N, Jiang Z, Wang X (2018). Multi-objective capacity allocation of hospital wards combining revenue and equity. *Omega* 81: 220-233.

Jie Li is a graduate of Sino-US Logistics Research Institute, Shanghai Jiao Tong University. Her research interests are simulation modeling, optimization and its applications

Sichen Li is studying in Antai College of Economics and Management at Shanghai Jiao Tong University for his master degree of Management Science. His research interests include simulation optimization and data-driven optimization.

Jun Luo received his PhD degree from the Hong Kong University of Science and Technology and bachelors degree from Nanjing University. He currently is a professor in Antai College of Economics and Management at Shanghai Jiao Tong University. His research interests include stochastic modeling, simulation optimization, and statistical learning, with applications in service operations management, supply chain management and financial risk management.

Haihui Shen is an associate professor in Sino-US Global Logistics Institute, Antai College of Economics and Management at Shanghai Jiao Tong University. He received his Ph.D. from the Department of Management Sciences, College of Business at City University of Hong Kong in 2018. His research focuses on simulation modeling, analysis and optimization, together with their application in areas including production, management and healthcare.