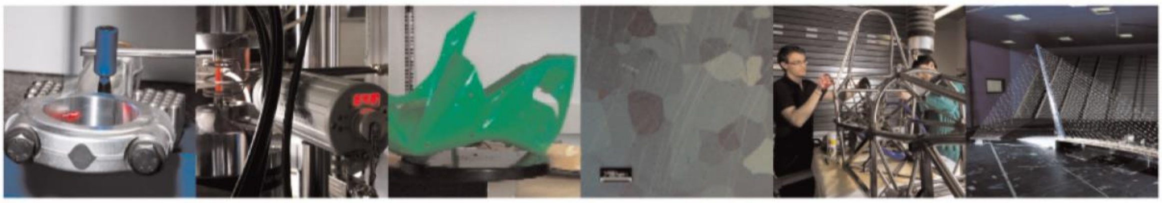


# Optimal budget allocation policy for tabu search in stochastic simulation optimization

Yu, Cl; Lahrichi, N; Matta, A

This is a post-peer-review, pre-copyedit version of an article published in COMPUTERS & OPERATIONS RESEARCH. The final authenticated version is available online at: http://dx.doi.org/10.1016/j.cor.2022.106046

# A Budget Allocation Policy for Tabu Search in Stochastic Simulation Optimization

Chunlong Yu[a,*], Nadia Lahrichi[b], Andrea Matta[c]

[a]*School of Mechanical Engineering, Tongji University, Shanghai 201804, China*
[b]*Département de Mathématiques et Génie Industriel, École Polytechnique de Montréal, Montréal H3C 3A7, Canada*
[c]*Department of Mechanical Engineering, Politecnico di Milano, Milan 20156, Italy*

---

**Abstract**

Tabu search (TS) is a powerful method for solving combinatorial optimization problems. However, when TS is adopted for stochastic simulation optimization, the simulation noises may mislead the search direction and prevent TS from converging to high-quality solutions. This issue can be mitigated by increasing the number of simulation samples used for solution evaluation, however, real-world applications are generally constrained by a finite computing budget. Therefore, it is critical to reducing the noise effect by an efficient simulation budget allocation, i.e., splitting the total number of samples on solutions. Studies on the budget allocation problem of TS are sparse. Most of the works employ equal allocation or simple policies which are not optimal. Based on the large deviations framework, we propose a new budget allocation policy to enhance the performance of TS under stochastic settings. The proposed allocation policy, provided in closed-form formulas, is asymptotically optimal for maximizing the probability that TS performs the correct move in a single iteration. The efficiency of the proposed method is validated by solving different problems from manufacturing and healthcare scenarios, including an inventory control problem, a throughput maximization problem for the production line, and a physician scheduling problem for the radiotherapy center. The numerical results show that, with the proposed method, TS can obtain better results using the same amount of computational efforts.

*Keywords:* metaheuristics; simulation optimization; tabu search; optimal computing budget allocation; ranking and selection.

---

*Corresponding author
*Email address:* chunlong_yu@tongji.edu.cn (Chunlong Yu)

## 1. Introduction

Originated from Glover (1986), Tabu search (TS) is a deterministic local search procedure for global optimization. Distinguishing from a pure local search, firstly, TS can escape from a local optimum by allowing moves to a neighborhood solution with increased cost; secondly, TS adopts the *tabus* as adaptive memory to avoid cycling when moving away from local optima. The basic framework of TS is rather simple. The search starts from an initial solution. At each iteration, it moves to the best solution of the current neighborhood. The reversing move is recorded in a tabu list, and is prevented for some future iterations unless the aspiration criteria is satisfied. This procedure iterates until the termination condition is met. The efficiency of TS can be further improved by coupling to the basic framework some additional search elements, such as intensification and diversification, in a proper way. See Gendreau (2002) for some advances of the method.

Despite its simplicity, TS is among the most effective methods to tackle difficult combinatorial optimization problems, and succeeds to provide quasi-optimal solutions in problems emerging from a variety of domains such as manufacturing (Costa et al., 2015), transportation (Qiu et al., 2018), healthcare (Ferland et al., 2001), among others. In most of the studies, TS is applied to tackle deterministic optimization problems. However, in many real-world cases, the existence of randomness drives the use of stochastic models for a better representation of the real system (Juan et al., 2015). There is a trend of solving stochastic optimization problems with TS. Gendreau et al. (1996) considered a vehicle routing problem with stochastic customer locations and demands, and developed an efficient TS which produces the optimal solution in most of the testing cases. Tsai and Gemmill (1998) proposed a TS to provide good solutions to a resource-constrained project scheduling problem with stochastic activity durations. Lutz et al. (1998) used TS to solve the buffer allocation problem in production lines with stochastic processing times. TS has also been adopted to find the optimal number of kanbans in a Just-In-Time system (Dengiz and Alabas, 2000), to tackle stochastic knapsack problem (Konak and Kulturel-Konak, 2005) and physician scheduling problem (Niroumandrad and Lahrichi, 2018).

When solving stochastic optimization problems with TS, the quality of solutions is usually evaluated by stochastic simulation models. Simulation is a common tool for modeling real-world systems because they are complex and rarely satisfy the assumptions of analytical models. Optimization via simulation model is known as simulation optimization (SO) (Xu et al., 2015, 2016). In this case, the objective value of a solution is no longer deterministic, but a random variable subjected to simulation noises. This introduces the risk that TS fails to pick the correct neighborhood solution and performs wrong solution updates. As a consequence, TS encounters two difficulties impeding its efficiency in finding the optimum: (1) The optimal solution is not visited due to the misled search direction; (2) The optimal solution is visited, but without being recognized as the best.

In this paper we define the budget as the number of simulation samples available for solution evaluation. Generally, simulation noises can be mitigated by increasing the budget allocated on solutions. However, due to the cost or time of a simulation, the total budget of an optimization procedure is limited. As a result, when coupled with a stochastic model, TS faces the following dilemma: allocating budget for precise evaluations to recognize the best visited solutions (estimation), or exploring more non-visited solutions to find the global optimum (exploration). This dilemma remains a challenge for the SO literature. In the case of TS, the budget allocation problem can be decomposed into two levels. The top level is the budget allocation along the search path, i.e., to decide the budget assigned to each search iteration. The second level is to split the budget to the neighborhood solutions. In this research, we focus on the latter problem. More specifically, with a total budget for one TS iteration, we focus on developing a budget allocation policy to minimize the probability of performing false moves due to simulation noises, and finally improve the TS efficiency.

In summary, this article aims at enhancing the search performance of TS in SO by optimizing the simulation budget allocation during the solution evaluation. The research contributions of this article are twofold:

1. We propose an asymptotically optimal budget allocation policy for TS. In the literature, the budget allocation of TS is generally performed by equal allocation or simple allocation rules which are not optimal. To the best of our knowledge, this is the first research that seamlessly integrates the concept of ranking and selection (R&S) into TS, and provides theoretical results on the asymptotic optimality of the budget allocation.

2. Closed-form formulas are developed to allocate budget to solutions, aiming at maximizing the probability that TS performs the correct move in an iteration. Thus, the developed algorithm is easy to implement in practice. We also propose a sequential allocation procedure for better using the information collected during the budget allocation procedure.

Although the policy is proposed for allocating budget in a single TS iteration, we validate its efficiency in improving the final solution quality of TS with applications from the literature and the real-world scenario. The proposed method is tested on three different problems. We use equal allocation and the optimal computing budget allocation approach (Chen et al., 2000) as two benchmarks. The comparisons are made in different cases in terms of simulation budget and simulation noise level. Results show that the proposed budget allocation procedure can further enhance the search performance of TS using the same amount of simulation budget.

This article is organized as follows. Section 2 reviews the literature. Section 3 provides the preliminaries of TS and formulates the budget allocation problem. Section 4 derives the budget allocation policy. Section 5 gives the numerical results. Section 6 concludes the paper.

## 2. Literature review

When TS is coupled with simulation, the search is guided by a sequence of decisions drawn from noised observations. The problem of improving decision accuracy by sampling allocation is closely related to the R&S problem. In this section, we first review the approaches proposed for the R&S problem, then investigate the studies using R&S approaches to enhance the search efficiency of metaheuristics, and finally, summarize the works on the budget allocation problem of TS.

R&S aims at identifying the best from a finite set of alternatives whose performance is estimated by simulations or experiments. A well-researched paradigm to tackle this problem is the indifference zone (IZ) formulation. It guarantees the probability of correct selection (PCS) whenever the best alternative is better than the other alternatives by at least a user-defined indifference-zone. This research stream started from the problem with basic features like normally and independently distributed sampling outputs, known and equal variance (Bechhofer, 1954), to more general settings. The followings are some typical methods. Rinott's two-stage selection procedure was proposed for alternative sets with unknown variances (Rinott, 1978). The fully sequential procedure developed in Kim and Nelson (2001), known as KN, allows unknown variances and the use of common random numbers that help to reduce variances in stochastic simulations. The KN++ procedure (Goldsman et al., 2002), as an extension of KN, can be applied when observations are non-normal and dependent within each alternative. Tackling a large alternative set is also an active research direction for R&S. Nelson et al. (2001) used a first-stage subset selection to screen out alternatives that are not competitive, whilst Luo et al. (2015) and Ni et al. (2017) take advantage of the parallel computing to tackle large scale R&S problems. The research stream of IZ formulation was summarized in

Bechhofer (1995), then Kim and Nelson (2006) provides recent reviews. In the IZ formulation, the primary task is to guarantee the PCS, whilst the efficiency of selecting the best is secondary. For this reason, the IZ approaches tend to allocate more replications than necessary to guarantee a PCS in the worst-case configuration. See Branke et al. (2007) for a comprehensive comparison between the IZ approaches and other methods.

As a consequence, many recent works have focused on the simulation budget allocation procedures with a primary goal to enhance the efficiency of selecting the best alternative. These studies can be categorized into two frameworks, the frequentist and the Bayesian. In the frequentist framework, the budget allocation is formulated as a static optimization problem to maximize the PCS with the given simulation budget, assuming the solution information such as true means and variances of alternatives are known in prior. By solving this problem, Chen et al. (2000) proposed the optimal computing budget allocation (OCBA) approach for maximizing the PCS. It provides a closed-form approximation of the sampling ratios which are asymptotically optimal when the budget tends to infinite. Glynn and Juneja (2004) solved this problem with the large deviations (LD) formulation, and obtained the same asymptotically optimal sampling ratios as OCBA under the normality assumption. Further, variants of the OCBA approach have been developed for different goals such as subset selection (Chen et al., 2008), multi-objective problem (Lee et al., 2010), among others. See Chen and Lee (2011) for more details of OCBA. In the frequentist framework, the sampling ratios are dependent on the unknown solution information. For this reason, in practical implementations, the allocation is usually sequential and the sampling ratios are estimated iteratively using plug-in estimators for the unknown parameters. Surprisingly, the sequential OCBA implementation using plug-in estimated parameters generally outperforms the static OCBA approach using perfect information, especially when the total budget is small or medium. These results can be found in Chen et al. (2006), as well as in a recent review of Peng et al. (2018a). This encourages the research stream using Bayesian models to capture the value of posterior information for better low-budget performance.

In the Bayesian framework, the budget allocation problem is formulated by incorporating the solution information accumulated from the sequential sampling procedure. More specifically, the problem is to optimize a conditional expectation of a certain performance indicator, e.g., a loss function, subjected to the posterior or predictive distributions of simulation outputs. To this end, several allocation procedures have been proposed, including the classic expected improvement(EI)

(Jones et al., 1998), some variants of EI such as the expected value of information (EVI) (Chick and Inoue, 2001; Chick et al., 2010), the knowledge gradient (KG)(Frazier et al., 2008), and the Bayesian OCBA (Chen, 1996; Chen et al., 2009). These procedures try to maximize the posterior information gains by the end of the next sampling stage and therefore, are considered as myopic heuristics. Although with these procedures the PCS can converge to one as the budget goes to infinity, the budget allocation may not necessarily converge to the asymptotically optimal sampling ratios when the variances are unknown (Ryzhov, 2016). This implies a suboptimal convergence rate and may limit their large-budget performance. To solve this, Chen and Ryzhov (2019b) modified a recently proposed variant of EI, known as 'complete EI', to allow the convergence to the asymptotically optimal sampling ratios. Chen and Ryzhov (2019a) proposed a dynamic allocation that learns the asymptotically optimal sampling ratios from the LD optimality conditions (Glynn and Juneja, 2004). Recently, Peng et al. (2018b) formulated the R&S procedure as a Markov decision process and proposed an allocation policy. The proposed method achieves the asymptotically optimal sampling ratios meanwhile it is shown more efficient than some EI methods.

As an enumeration approach, R&S is not efficient for optimization problems with large solution space, which is usually encountered in many real-world applications. As a result, there emerges a trend to integrate R&S with metaheuristics to find the optimal or high-quality solutions efficiently. R&S is incorporated into metaheuristics such as genetic algorithm (GA) (Schmidt et al., 2006; Lee et al., 2008), nested partition (Chew et al., 2009), cross entropy (He et al., 2010) and particle swarm optimization (Pan et al., 2006; Zhang et al., 2016). In some cases, such integration is straightforward. This happens when the search mechanism, or a part of it, aligns to the purpose of R&S. For example, in multi-objective evolutionary algorithms, it is required to decide at each iteration the non-dominated set, from which the parents are selected for reproduction. To this end, Lee et al. (2008) incorporated the multi-objective OCBA (Lee et al., 2010) to correctly recognize the non-dominated set. For the same purpose, multi-objective OCBA was coupled directly to the nested partition algorithm in Chew et al. (2009). However, when such mechanism alignment does not exist, modifications on the budget allocation are required. For example, He et al. (2010) derived an OCBA approach for the cross entropy algorithm to select the elite set. Although the form is quite similar to the mOCBA given in Chen et al. (2008), the proposed rule aims at a different objective to minimize the mean-squared error of the cross entropy weight function. Schmidt et al. (2006) found that simulation noises affect the correctness of the selection and replacement procedure in GA. The authors proposed a sampling allocation framework that focuses only on the pairwise comparisons that are adopted by the GA mechanism, and adapted the Bayesian OCBA approach (Chen, 1996) to improve the budget efficiency. Recently, Zhang et al. (2016) formulated the budget allocation problem of particle swarm optimization, and proposed an asymptotically optimal budget allocation under the LD framework. Particle swarm optimization with the proposed budget allocation is shown more efficient than that integrated with the standard OCBA.

Studies on the budget allocation problem of TS for stochastic optimization problems are quite sparse. Most of the applications utilize the equal allocation method, which allocates an equal number of replications (or equal simulation length) to the solutions. These are, e.g., Lutz et al. (1998), Tsai and Gemmill (1998), Yang et al. (2004), Dengiz and Alabas (2000) and Niroumandrad and Lahrichi (2018). In Konak and Kulturel-Konak (2005), TS was applied to solve the stochastic knapsack problem. The authors showed by empirical results that "the performance of a simulation optimization using TS may highly depend on how the simulation is conducted during the search and how its output is used to guide the search". The authors proposed several budget allocation policies which obtained some improvements from the equal allocation. However, these are simple heuristics and there is no evidence of optimality.

To the best of our knowledge, the attempt of combining R&S and TS has not been made. Due to the mechanism discrepancy, the R&S techniques cannot be directly applied with TS. In TS, the problem is not only about selecting the best solution from the neighborhood, one also needs to consider the tabu mechanism and the update of the best-known solution. More specifically, the tabu mechanism is twofold. On the one hand, it prevents a move to a tabu solution; on the other hand, it allows a move to a tabu solution if the aspiration rule is triggered. Intuitively, the budget allocated on tabu solutions should be less than that on non-tabu solutions, but should be enough to assure a correct triggering of the aspiration rule. What is the appropriate budget for tabu solutions? After moving to a neighborhood solution, the best-known solution should be updated when necessary. To assure a correct update, how much additional budget should be allocated to the best-known solution? The optimal budget allocation becomes not so straightforward when considering all these factors.

## 3. Problem formulation

The simulation optimization problem can be expressed as

$$\min_{x \in \Theta} y(x), \tag{1}$$

where $\Theta$ is the search space, $y(x) = E[Y(x)]$ is the expectation of the simulation output $Y(x)$, which is a random variable. Generally, $y(x)$ can only be estimated by the sample mean $\bar{Y}(x;n) = \frac{1}{n}\sum_{j=1}^{n} Y_j(x)$, where $n$ is the number of simulation observations. The estimation accuracy increases with $n$, and $\lim_{n\to\infty} \bar{Y}(x;n) \to y(x)$. The following assumptions are made.

**Assumption 1.** *Assume that* $Var[Y(x)] < \infty$ *for all* $x \in \Theta$*, and we can collect independent and identically distributed (i.i.d.) observations,* $Y_1(x), Y_2(x), \ldots$ *at any* $x$*.*

**Assumption 2.** *The simulation observations* $Y_j(x)$*'s are normally distributed, or namely,* $Y_j(x) \sim \mathcal{N}(y(x), \sigma^2(x))$*,* $\forall j$*.*

Both assumptions are widely adopted in the SO literature. More specifically, Assumption 1 holds when the observations are collected from different independent simulation replications. Whilst when the observations are drawn from different segments of the same simulation replication, the correlation may exist among samples from consecutive segments, but it can be reduced by increasing the segment length (Law and Kelton, 2000). Assumption 2 holds in many applications because the performance metrics are generally taken over some averages of a sample path or a batch of samples (e.g., averaged system time of customers). Even when the normality is not significant, one can improve the normality through the batch means technique (Law and Kelton, 2000) .

### *3.1. Tabu search*

TS functions similarly as the ordinary local search, proceeding iteratively from one point (solution) to another until a chosen termination criterion is satisfied. Each $x \in \Theta$ has an associated neighborhood $\mathbb{N}(x)$, and each solution $x' \in \mathbb{N}(x)$ can be reached from $x$ by an operation called *move*. Local search applies the descent move, i.e., it moves to a neighborhood solution $x'$ such that $y(x') < y(x)$. The evident shortcoming of local search is that it stops at a local optimum. To overcome this, TS introduces the following adjustments. First, it moves to the best solution in the neighborhood, no matter the move is descent or ascent. Second, a subset of neighbors in $\mathbb{N}(x)$ are tabu according to a tabu list $T$, which prevents cycles and guides the search to escape from local optima. The tabu list is considered as a short-term memory, which is updated based on the recently visited solutions. The elements in $T$ could have various formats: solutions, solution attributes, operators for generating solutions, etc. Typically, moving to a tabu solution is forbidden, but if this move satisfies certain aspiration criteria, it will still be accepted. We consider the simplest and most common aspiration criteria: the move to a tabu solution is acceptable if this solution outperforms the best solution visited in the search path, i.e., the best-known solution.

We use the following notations to better describe the move in a single TS iteration:

$x$: Current solution before performing the move;

$x_0$: Best-known solution, i.e., the best solution among all visited solutions in the previous iterations;

$x_i$: The $i$-th neighborhood solution of $x$;

$N$: Index set of neighborhood solutions, $N = 1, 2, \ldots, k$;

$J$: Index set of tabu neighborhood solutions, i.e., $J = \{j \in N | x_j \text{ is tabu according to } T\}$;

$I$: Index set of non-tabu neighborhood solutions, i.e., $I = N \setminus J$;

$b$: Index of the best non-tabu neighborhood solution, i.e., $b = \arg\min_{i \in I} y(x_i)$;

$g$: Index of the best neighborhood solution, i.e., $g = \arg\min_{i \in N} y(x_i)$.

The admissible neighborhood solution set, i.e., a set consisting of all the neighborhood solutions to which the move is allowed, is given by $A = I \cup J^*$, where $J^* = \{j \in J | y(x_j) < y(x_0)\}$ is the set of tabu neighborhood solutions satisfying the aspiration criteria. If $\min_{i \in N} y(x_i) \geq y(x_0)$, we have $J^* = \emptyset$, then the admissible neighborhood solution set $A$ reduces to the non-tabu neighborhood set $I$. In this case, TS will move to the best solution in $I$, i.e., $x_b$. Otherwise, the move to the best solution in $A$ will be adopted. Since $\arg\min_{i \in I \cup J^*} y_i = \arg\min_{i \in I \cup J} y_i = \arg\min_{i \in N} y_i = g$, TS performs the move to $x_g$.

The complete TS procedure is given in Algorithm 1. The search starts at an initial solution $x^0$. At each iteration, the move is performed, and the best-known solution $x_0$ will be updated if necessary. Then, $T$ is updated by adding the new tabu and deleting the oldest tabu if a certain removal condition, e.g., the length of $T$ exceeds a limit, is met. This procedure repeats until the termination condition is met. Note that Algorithm 1 coincides with the standard TS framework given in Gendreau et al. (2010), except that we rephrase it by distinguishing the moves in two scenarios to facilitate the derivation of budget allocation policy.

**Algorithm 1:** Tabu search

**Require:** initial solution $x^0$; maximum iteration $t_{max}$

**Ensure:** the best-known solution $x_0$

1 **Initialize**: iteration $t \leftarrow 1$, current solution $x \leftarrow x^0$, best-known solution $x_0 \leftarrow x$, tabu list $T \leftarrow \emptyset$;<br>
2 **while** $t < t_{max}$ **do**<br>
3     Evaluate the neighborhood solutions of $x$, obtaining $y(x_i)$ for $i \in N$;<br>
4     **if** $\min_{i \in N} y(x_i) \geq y(x_0)$ **then**<br>
5         Move to the best non-tabu neighborhood solution, i.e., $x \leftarrow x_b$, where $b = \arg\min_{i \in I} y(x_i)$;<br>
6     **else**<br>
7         Move to the best neighborhood solution, i.e., $x \leftarrow x_g$, where $g = \arg\min_{i \in N} y(x_i)$;<br>
8     **end**<br>
9     **if** $y(x) < y(x_0)$ **then**<br>
10         Update the best-known solution $x_0 \leftarrow x$;<br>
11     **end**<br>
12     Record tabu for the current move, and add the tabu into $T$;<br>
13     **if** *Tabu removal condition is met* **then**<br>
14         Remove the oldest tabu from $T$;<br>
15     **end**<br>
16     $t \leftarrow t + 1$;<br>
17 **end**

*3.2. The budget allocation problem*

When TS is applied for SO, during each iteration, a finite simulation budget is allocated for estimating $y(x_0), y(x_1), \dots, y(x_k)$. Due to simulation noises, estimation errors are inevitable and would affect the search decisions. According to Algorithm 1, the search decisions based on sample means are twofold: choosing the neighborhood solution (Line 5 and 7) to move, and updating the best-known solution (Line 10). Both are essential for the TS performance, the former decides the search direction, whilst the latter determines the final output. To model these decisions jointly, a *TS move* is defined as below.

**Definition 1.** *Let $s^t = (x, x_0)$ be the state of TS at the $t$-th iteration, where $x$ is the current solution at the $t$-th iteration, and $x_0$ is the best-known solution at the $t$-th iteration. A TS move,* *denoted as $m$, is defined as a transition of $s^t \to s^{t+1}$.*

Let $m(s^t)$ represent the correct TS move in state $s^t$ based on true objective values $y(x_0), \dots, y(x_k)$, and $\hat{m}(s^t, n_0, \dots, n_k, \xi)$ be the actual move based on $\bar{Y}(x_0; n_0), \dots, \bar{Y}(x_k; n_k)$, where $\bar{Y}(x_i; n_i)$ is the sample mean calculated from $n_i$ simulation samples, and $\xi$ is a general term representing the simulation randomness. The probability of correct move (PCM) is given by

$$\text{PCM} = P(\hat{m}(s^t, n_0, \dots, n_k, \xi) = m(s^t)).$$

The sampling allocation has a direct impact on the estimation accuracy and, therefore, the move. Then, with a fixed budget, optimizing the allocation policy is an essential way to improve PCM. The problem considered in this research is to optimize the simulation budget allocation at a single iteration of TS. Given a simulation budget $n$ for one iteration, say $t$, the problem is

$$\begin{aligned} \max_{\alpha_0, \dots, \alpha_k} \quad & \text{PCM}, \\ \text{s.t.} \quad & \alpha_0 + \alpha_1 + \ldots + \alpha_k = 1, \\ & \alpha_i > 0, i = 0, 1, \dots, k. \end{aligned} \tag{2}$$

Here, $\alpha_i, i = 0, 1, \dots, k$ is the sampling ratio, $n_i = \alpha_i n, i = 1, \dots, k$ is the budget allocated to the $i$-th neighborhood solution, $n_0 = \alpha_0 n$ is budget of the best-known solution (we ignore that $\alpha_i n$ is not an integer).

Solving the budget allocation problem (2) is not identical to solving the simulation optimization problem (1) optimally. In stochastic settings, the optimality gap of TS could be the result of two factors: the misleading effect of simulation noises and the metaheuristic nature of TS. This article focuses on mitigating the first factor through solving (2), so as to enhance the TS performance. Whilst tackling the second factor by improving the search mechanism of TS via, e.g., hybridization, advanced intensification and diversification, is beyond the scope of this work.

## 4. Simulation budget allocation

At an iteration, TS selects the search mode depending on the neighborhood features. When $y(x_g) \geq y(x_0)$, $g = \arg\min_{i \in N} y(x_i)$, the search is guided by the tabu list, which facilitates the hill-climbing and exploration of new regions (Algorithm 1 Line 5). This scenario is named *Best-Holding*. Otherwise, TS acts as a simple local search algorithm and moves to the best neighborhood solution (Algorithm 1 Line 7). This scenario is called *Best-Improving*.

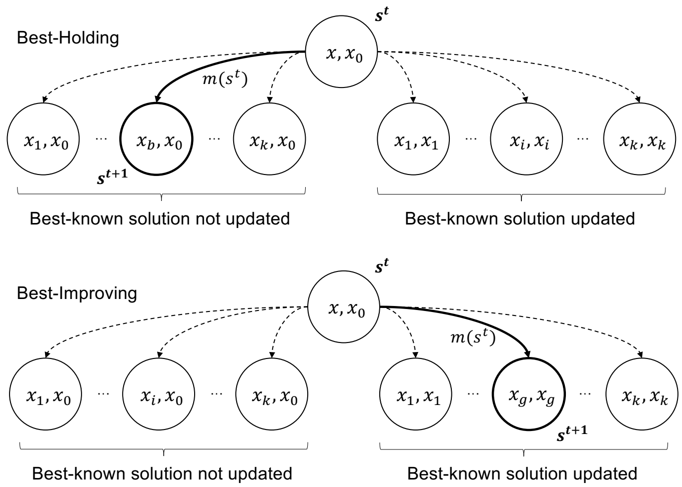



Figure 1: Correct moves of tabu search in different scenarios

The correct TS moves in two scenarios are given in Figure 1. For Best-Holding, TS replaces the current solution $x$ with the best non-tabu neighborhood solution $x_b$ and maintains the current best-known solution $x_0$, the correct move is thus $m(s^t) : (x, x_0) \to (x_b, x_0)$. For Best-Improving, TS replaces the current solution $x$ with the best neighborhood solution $x_g$, and set $x_g$ as the best-known solution. Hence, the correct move is $m(s^t) : (x, x_0) \to (x_g, x_g)$. In the following subsections, we derive the budget allocation policy for each scenario.

### *4.1. Budget Allocation of Scenario Best-Holding*

For Best-Holding, the correct move $m(s^t) : (x, x_0) \to (x_b, x_0)$ can be misled by the following *wrong move events*:

(a) *Wrong pick.* This happens when a non-tabu neighborhood solution $x_i$ is esimated to be better than $x_b$, or formally, $\bar{Y}(x_i; n_i) < \bar{Y}(x_b; n_b), i \in I \setminus b$. In this case, TS fails to pick $x_b$ thus the correct move cannot happen.

(b) *Wrong aspiration.* This occurs when any tabu neighborhood solution is estimated to be better than the best-known solution, or formally, $\bar{Y}(x_j; n_j) < \tilde{y}(x_0; n_0), j \in J$. In this case, the aspiration rule would be wrongly triggered. TS moves to $x_j$ instead of $x_b$ and updates the best-known solution.

(c) *Wrong update.* The best non-tabu neighborhood solution $x_b$ is estimated to be better than $x_0$, or formally, $\bar{Y}(x_b; n_b) < \tilde{y}(x_0; n_0)$, the best-known solution would be wrongly updated.

Thus, the probability of false move (PFM) is given by:

$$\text{PFM} = P\Big(\bigcup_{i \in I \setminus b} (\bar{Y}(x_i; n_i) \leq \bar{Y}(x_b; n_b)) \cup \bigcup_{j \in J} (\bar{Y}(x_j; n_j) \leq \tilde{y}(x_0)) \cup (\bar{Y}(x_b; n_b) \leq \tilde{y}(x_0))\Big),$$

where $\tilde{y}(\cdot)$ is the realized value of the sample mean from the last iteration. Since an accurate estimation of the best-known solution is important for a correct move, we consider the possibility to allocate budget to improve the estimation accuracy of $x_0$ at the current iteration, and thus $\tilde{y}(x_0)$ is replaced with $\bar{Y}(x_0; n_0)$. Then we have

$$\text{PFM} = P\Big(\bigcup_{i \in I \setminus b} (\bar{Y}(x_i; n_i) \leq \bar{Y}(x_b; n_b)) \cup \bigcup_{j \in J} (\bar{Y}(x_j; n_j) \leq \bar{Y}(x_0; n_0)) \cup (\bar{Y}(x_b; n_b) \leq \bar{Y}(x_0; n_0))\Big), \tag{3}$$

and $\text{PCM} = 1 - \text{PFM}$.

The major difficulty for solving Model (2) is that no closed-form expression exists for PFM. The common idea is to approximate PFM with its bounds. For R&S problem, Chen et al. (2000) approximated PCS with a lower bound whose closed form is available. By optimizing the lower bound, they obtain analytical expressions for the optimal sampling ratios when $n \to \infty$. Despite the allocation is asymptotically optimal, it brings significant improvement to the simulation efficiency with finite budget. However, this approach requires the normality assumption of simulation outputs. A more general approach allowing various distribution types is based on the LD framework (Dembo and Zeitouni, 2010). Glynn and Juneja (2004) showed that the problem of maximizing PCS can be converted to determining the allocation $\alpha^* = (\alpha_0, \ldots, \alpha_k)$ that maximizes the LD rate function associated with the probability of false selection (PFS), which is defined by $\lim_{n\to\infty} -\frac{1}{n} \log \text{PFS}$. This paradigm is then used for solving various simulation budget allocation problems arising in, e.g., particle swarm optimizaiton (Zhang et al., 2016), identifying the Pareto set (Li et al., 2017).

Similarly, to solve Model (2), we have to decide the allocation that maximizes the rate function associated with PFM, namely,

$$\begin{aligned} \max_{\alpha_0, \ldots, \alpha_k} \lim_{n\to\infty} & -\frac{1}{n} \log \text{PFM}, \\ \text{s.t.} \quad & \alpha_0 + \alpha_1 + \ldots + \alpha_k = 1, \\ & \alpha_i > 0, i = 0, 1, \ldots, k. \end{aligned} \tag{4}$$

Given that PFM is intractable, we approximate it with the bounds provided in Lemma 1.

**Lemma 1.** *The PFM in Eq.(3) is bounded by* $P^* \leq PFM \leq kP^*$, *where*

$$P^* = \max\{\max_{i\in I\setminus b} P(\bar{Y}(x_i;n_i) \leq \bar{Y}(x_b;n_b)), \max_{j\in J} P(\bar{Y}(x_j;n_j) \leq \bar{Y}(x_0;n_0)), P(\bar{Y}(x_b;n_b) \leq \bar{Y}(x_0;n_0))\}. \tag{5}$$

*Proof.* See Appendix A. □

With any positive allocation $\alpha = (\alpha_0, \ldots, \alpha_k)$ where $\alpha_i > 0, \forall i = 0, \ldots, k$, when $n \to \infty$, the solutions tend to receive infinite number of samples and thus, $\bar{Y}(x_i;n_i) \to y(x_i)$ for all $i = 0, \ldots, k$. As a result, the probabilities of all wrong move events decay to zero and thus, $P^* \to 0$. Since $P^* \leq \text{PFM} \leq kP^*$ , PFM $\to 0$ at the same rate of $P^*$. The goal of maximizing $\lim_{n\to\infty} -\frac{1}{n}\log \text{PFM}$ is equivalent to maximizing $\lim_{n\to\infty} -\frac{1}{n}\log P^*$.

To derive the rate function of $P^*$, we introduce the rate functions of the wrong move events as follows. For any wrong pick event, $P(\bar{Y}(x_i;n_i) \leq \bar{Y}(x_b;n_b)), i \in I \setminus b$ decreases to 0 exponentially at a rate of

$$G_{ib}(\alpha_i, \alpha_b) = \lim_{n\to\infty} -\frac{1}{n}\log P(\bar{Y}(x_i;n_i) \leq \bar{Y}(x_b;n_b)) = \inf_{\tau}(\alpha_i \mathscr{I}_i(\tau) + \alpha_b \mathscr{I}_b(\tau)), \tag{6}$$

for any wrong aspiration event,

$$G_{j0}(\alpha_j, \alpha_0) = \lim_{n\to\infty} -\frac{1}{n}\log P(\bar{Y}(x_j;n_j) \leq \bar{Y}(x_0;n_0)) = \inf_{\tau}(\alpha_j \mathscr{I}_j(\tau) + \alpha_0 \mathscr{I}_0(\tau)), \tag{7}$$

and for the wrong update,

$$G_{b0}(\alpha_b, \alpha_0) = \lim_{n\to\infty} -\frac{1}{n}\log P(\bar{Y}(x_b;n_b) \leq \bar{Y}(x_0;n_0)) = \inf_{\tau}(\alpha_b \mathscr{I}_b(\tau) + \alpha_0 \mathscr{I}_0(\tau)), \tag{8}$$

where $\mathscr{I}_i(\tau)$ is defined in LD theory as the rate function of $P(\bar{Y}(x_i;n_i) < \tau)$ for $y(x_i) > \tau$ or $P(\bar{Y}(x_i;n_i) > \tau)$ for $y(x_i) < \tau$. Eq.(6) - (8) hold for light-tailed underlying distributions such as the Normal, Bernoulli, Poisson and Gamma family. See Section 2.2 in Glynn and Juneja (2004) for the detailed derivation of the rate functions.

Then, we obtain the rate function of $P^*$ as

$$\lim_{n\to\infty} -\frac{1}{n}\log P^* = \min\{\min_{i\in I\setminus b}\{G_{ib}(\alpha_i, \alpha_b)\}, \min_{j\in J}\{G_{j0}(\alpha_j, \alpha_0)\}, G_{b0}(\alpha_b, \alpha_0)\},$$

Therefore, the budget allocation optimization problem (4) becomes

$$
\begin{aligned}
&\max_{\alpha_0,\ldots,\alpha_k} \min_{i\in I\setminus b, j\in J} \{G_{ib}(\alpha_i,\alpha_b), G_{j0}(\alpha_j,\alpha_0), G_{b0}(\alpha_b,\alpha_0)\} \\
\text{s.t.} \quad & \sum_{i\in I\setminus b} \alpha_i + \sum_{j\in J} \alpha_j + \alpha_0 = 1, \\
& \alpha_0, \alpha_b, \alpha_i, \alpha_j > 0, \forall i \in I \setminus b, j \in J.
\end{aligned} \tag{9}
$$

The rate function $G_{ij}(\alpha_i, \alpha_j), i \neq j$ is a continuously differentiable concave function and is monotonically increasing with $(\alpha_i, \alpha_j)$ (Glynn and Juneja, 2004). The objective function of Model (9) is a concave function w.r.t $\alpha_0, \ldots, \alpha_k$, because the minimum of a series of concave functions is also concave. Thus, Model (9) is a concave optimization model, which can be re-written into

$$
\begin{aligned}
&\max z \\
\text{s.t.} \quad & G_{ib}(\alpha_i,\alpha_b) \geq z, \quad \forall i \in I \setminus b, \\
& G_{j0}(\alpha_j,\alpha_0) \geq z, \quad \forall j \in J, \\
& G_{b0}(\alpha_b,\alpha_0) \geq z, \\
& \sum_{i\in I\setminus b} \alpha_i + \sum_{j\in J} \alpha_j + \alpha_b + \alpha_0 = 1, \\
& \alpha_0, \alpha_b, \alpha_i, \alpha_j > 0, \forall i \in I \setminus b, j \in J.
\end{aligned} \tag{10}
$$

Model (10) is a constrained convex optimization model satisfying the constraint qualification (Lemma 2) and hence, the Karush-Kuhn-Tucker (KKT) conditions are sufficient and necessary for the optimality. By applying the KKT conditions, we obtain the optimality conditions of the sampling ratios $\alpha_0, \alpha_1, \ldots, \alpha_k$ in Theorem 1.

**Lemma 2.** *The mathematical model (10) satisfies the constraint qualification.*

*Proof.* See Appendix B. □

**Theorem 1.** *The budget allocation $\alpha_0, \alpha_1, \ldots, \alpha_k$ is optimal for Model (10) if the following condi-*

*tions are satisfied:*

$$\sum_{j\in J}\frac{\partial G_{j0}(\alpha_j,\alpha_0)/\partial\alpha_0}{\partial G_{j0}(\alpha_j,\alpha_0)/\partial\alpha_j}+(1-\sum_{i\in I\setminus b}\frac{\partial G_{ib}(\alpha_i,\alpha_b)/\partial\alpha_b}{\partial G_{ib}(\alpha_i,\alpha_b)/\partial\alpha_i})\frac{\partial G_{b0}(\alpha_b,\alpha_0)/\partial\alpha_0}{\partial G_{b0}(\alpha_b,\alpha_0)/\partial\alpha_b}=1, \tag{11a}$$

$$G_{ib}(\alpha_i,\alpha_b)=G_{j0}(\alpha_j,\alpha_0)=G_{b0}(\alpha_b,\alpha_0),\forall i\in I\setminus b, j\in J, \tag{11b}$$

$$1-\sum_{i\in I\setminus b}\frac{\partial G_{ib}(\alpha_i,\alpha_b)/\partial\alpha_b}{\partial G_{ib}(\alpha_i,\alpha_b)/\partial\alpha_i}\geq 0, \tag{11c}$$

$$\sum_{i\in I\setminus b}\alpha_i+\sum_{j\in J}\alpha_j+\alpha_b+\alpha_0=1, \tag{11d}$$

$$\alpha_i,\alpha_j,\alpha_b,\alpha_0>0,\forall i\in I\setminus b, j\in J. \tag{11e}$$

*where* $b=\arg\min_{i\in I} y(x_i)$.

*Proof.* See Appendix C. □

**Remark 1.** *Conditions (11b) indicate that the asymptotically optimal budget allocation is a way to balance the rate functions of all wrong move events. Since any of these events may lead to a wrong move, the best way is to balance their occurrences and prevent the existence of a "bottleneck". This is found as a common insight for the decision-making based on sampling allocation. Similar results can be found in the R&S problem (Glynn and Juneja, 2004), and the OCBA problem of particle swarm optimization (Zhang et al., 2016).*

To derive the optimal values of $\alpha_i$'s, the closed-form expressions of $G_{ij}(\alpha_i,\alpha_j)$'s are provided in Lemma 3.

**Lemma 3.** *Suppose* $Y(x_i)\sim\mathcal{N}(y(x_i),\sigma_i^2)$ *for* $i=0,\dots,k$ *, then*

$$G_{ij}(\alpha_i,\alpha_j)=\frac{(y(x_i)-y(x_j))^2}{\sigma_i^2/\alpha_i+\sigma_j^2/\alpha_j},\quad \forall i\neq j. \tag{12}$$

*Proof.* See Appendix D □

The optimal allocation $\alpha_0,\dots,\alpha_k$ can be obtained by solving the system of equations in Theorem 1. However, due to the high nonlinearity of the rate functions, solving such a system requires a numerical approach based on certain iterative search algorithm, which introduces extra computational burden. For a fast and easy implementation, based on some mild assumptions, we provide the closed-form formulas of the asymptotically optimal budget allocation in Proposition 1.

**Proposition 1.** *Assume the simulation output of each solution follows a normal distribution, $\alpha_b \gg \alpha_i, \forall i \in I \setminus b, \alpha_0 \gg \alpha_j, \forall j \in J$, and $\alpha_b \gg \alpha_0$, the asymptotically optimal budget allocation $\alpha_0, \alpha_1, \ldots, \alpha_k$ for tabu search to maximize the probability of correct move in a single iteration of Scenario Best-Holding is given by:*

$$\frac{\alpha_i}{\alpha_{i'}} = \frac{\sigma_i^2/(y(x_b) - y(x_i))^2}{\sigma_{i'}^2/(y(x_b) - y(x_{i'}))^2}, \forall i, i' \in I \setminus b, i \neq i', \tag{13a}$$

$$\frac{\alpha_j}{\alpha_{j'}} = \frac{\sigma_j^2/(y(x_0) - y(x_j))^2}{\sigma_{j'}^2/(y(x_0) - y(x_{j'}))^2}, \forall j, j' \in J, j \neq j', \tag{13b}$$

$$\frac{\alpha_i}{\alpha_j} = \frac{\sigma_i^2/(y(x_b) - y(x_i))^2}{\sigma_j^2/(y(x_0) - y(x_j))^2}, \forall i \in I \setminus b, j \in J, \tag{13c}$$

$$\alpha_0 = \max\{\sigma_0 \sqrt{\sum_{i \in J} \alpha_j^2/\sigma_j^2}, \alpha_i \frac{\sigma_0^2/(y(x_b) - y(x_0))^2}{\sigma_i^2/(y(x_i) - y(x_b))^2}\}, \forall i \in I \setminus b, \tag{13d}$$

$$\alpha_b = \sigma_b \sqrt{\alpha_0^2/\sigma_0^2 - \sum_{j \in J} \alpha_j^2/\sigma_j^2 + \sum_{i \in I \setminus b} \alpha_i^2/\sigma_i^2}, \tag{13e}$$

$$\alpha_0 + \alpha_b + \sum_{i \in I \setminus b} \alpha_i + \sum_{j \in J} \alpha_j = 1. \tag{13f}$$

*where $b = \arg\min_{i \in I} y(x_i)$.*

*Proof.* See Appendix E. □

According to Proposition 1, the budgets allocated to the solutions depend both on their true mean and variances. Generally, solutions with a higher variability receive more budget. This coincides with the aim of reducing the noise by increasing sample size. Besides, several insights are summarized in the following remarks.

**Remark 2.** *Eq.(13a) show that the budget allocated to a non-tabu neighborhood solution, say $x_i$, is inversely proportional to the gap between its true mean and that of the best non-tabu neighborhood solution. Indeed, as such gap gets smaller, it becomes harder to recognize the best non-tabu neighborhood solution. For this reason, $x_i$ receives more budget to reduce its output variability, so as to prevent the wrong pick.*

**Remark 3.** *Eq.(13b) indicate that the budget allocated to a tabu neighborhood solution is inversely proportional to the gap between its true mean and that of the best-known solution. Similar to Remark 2, this can be interpreted as a measure to prevent the wrong aspiration.*

**Remark 4.** *Eq.(13c) reveal the competition of budget between the non-tabu and tabu neighborhood solutions. Since $y(x_0) \leq y(x_b)$ in Best-Holding, tabu neighborhood solutions are less competitive than non-tabu ones.*

**Remark 5.** *Eq.(13d) show that the best-known solution receives more budget than any tabu neighborhood solution. This is because $x_0$ has to be compared to each tabu neighborhood solution for checking the aspiration criteria, the accuracy of $x_0$ is more critical and should be guaranteed. Also, as indicated by the second term of Eq.(13d), the budget allocated to $x_0$ increases as the gap between $y(x_0)$ and $y(x_b)$ decreases. This acts as a measure to prevent the wrong update.*

**Remark 6.** *By plugging Eq.(13d) into Eq.(13e), one can see that the best non-tabu neighborhood solution $x_b$ receives more budget than any other non-tabu neighborhood solution, and the budget allocated on $x_b$ is inversely proportional to the gap between $y(x_0)$ and $y(x_b)$. Similar to Remark 5, these can be understood as measures to avoid the wrong pick and wrong update, respectively.*

In summary, the proposed allocation policy clearly reveals the critical solutions for budget allocation, and provides insights on how to guarantee a correct move. A numerical approach to calculate the sampling ratios is given in Appendix F.

*4.2. Budget Allocation of Scenario Best-Improving*

For Best-Improving, the events misleading the correct move $m(s^t) : (x, x_0) \to (x_g, x_g)$ are as follows:

(a) *Wrong pick.* The best neighborhood solution $x_g$ is estimated to be worse than another neighborhood solution, or formally, $\bar{Y}(x_i; n_i) < \bar{Y}(x_g; n_g), i \in N \setminus g$. In this case, a wrong neighbor would be picked.

(b) *Wrong update.* The best-known solution $x_0$ is estimated to be better than the best neighborhood solution $x_g$, or formally, $\bar{Y}(x_0; n_0) < \bar{Y}(x_g; n_g)$. In this case, the best-known solution would not be updated correctly.

As a result, the PCM can be formulated as:

$$\begin{aligned}\text{PCM} = 1 - \text{PFM} &= 1 - P\Big(\bigcup_{i \in N \setminus g} (\bar{Y}(x_i; n_i) < \bar{Y}(x_g; n_g)) \cup (\bar{Y}(x_0; n_0) < \bar{Y}(x_g; n_g))\Big) \\ &= 1 - P\Big(\bigcup_{i \in N \cup \{0\} \setminus g} (\bar{Y}(x_i; n_i) < \bar{Y}(x_g; n_g))\Big). \end{aligned} \tag{14}$$

As shown, a false move is made when the best solution $g$ is not recognized in the set $N \cup \{0\}$. The problem of maximizing the PCM in Eq.(14) coincides with the problem of maximizing the probability of selecting the best from a finite set, i.e., the R&S problem. The budget allocation for R&S is tackled by the well-known OCBA approach proposed in Chen et al. (2000). For this reason, the following proposition is provided.

**Proposition 2.** *Assume the simulation output of each solution follows a normal distribution,* $\alpha_g \gg \alpha_i, \forall i \in N \cup \{0\} \setminus g$, *the asymptotically optimal budget allocation* $\alpha_0, \alpha_1, \ldots, \alpha_k$ *for the tabu search to maximize the probability of correct move in a single iteration of Scenario Best-Improving is given by:*

$$\frac{\alpha_i}{\alpha_j} = \frac{\sigma_i^2/(y(x_i) - y(x_g))^2}{\sigma_j^2/(y(x_j) - y(x_g))^2}, \quad \forall i, j \in N \cup \{0\} \setminus g, i \neq j \tag{15a}$$

$$\alpha_g = \sigma_g \sqrt{\sum_{i \in N \cup \{0\} \setminus g} \alpha_i^2/\sigma_i^2}, \tag{15b}$$

$$\alpha_g + \sum_{i \in N \cup \{0\} \setminus g} \alpha_i = 1. \tag{15c}$$

*where* $g = \arg\min_{i \in N} y(x_i)$.

*Proof.* See Chen et al. (2000). □

*4.3. Sequential budget allocation procedure*

The allocation policy given in Proposition 1 and 2 is named optimal computing budget allocation for tabu search (TSOCBA). The sampling ratios $\alpha_i$'s in TSOCBA are calculated via unknown parameters $y(x_i)$'s and $\sigma_i$'s. Moreover, the selection of allocation rule depends on the scenario $\mathscr{S} \in \{\text{Best-Holding}, \text{Best-Improving}\}$, which, however, is unknown before the sampling procedure. To facilitate the implementation, Algorithm 2 provides a sequential allocation procedure that uses information accumulated during the budget allocation for estimating the unknown scenario and parameters. It starts with an initial allocation phase that collects $n^0$ samples for each of the neighborhood solution to get a first estimation of the parameters $y(x_i)$'s, $\sigma_i$'s and $\mathscr{S}$. Then, based on the estimated scenario and values, the allocation rule is selected, and the sampling ratio of each solution is calculated. After, a small increment of simulation budget, $\Delta$, is allocated to the solutions based on the calculated sampling ratios. Then, the estimated scenario and parameters are updated.

This procedure is continued until the total budget $n$ is exhausted. Ideally, as the estimation gets better, each allocation step should bring us closer to the correct budget allocation.

One possible concern about the sequential allocation procedure is that the poor estimation of $y(x_i)$'s, $\sigma_i$'s at the beginning phase may mislead the budget allocation. The situation is even worse when $\mathscr{S}$ is wronly estimated thus a wrong allocation rule will apply. However, it has been shown in the literature that the sequential implementation using the posterior estimations of $y(x_i)$'s and $\sigma_i$'s would not necessarily worsen the performance, and sometimes, it obtains better performance than allocation with perfect information. The reason, according to (Peng et al., 2018a), is that the posterior mean-variance trade-off happens to lead to a desirable allocation-and-sampling policy in certain circumstances. The same phenomenon happens for the scenario $\mathscr{S}$, which we will show in our numerical experiments in Section 5.1.2 and 5.1.3.

TSOCBA can be integrated into TS conveniently. In Algorithm 1, during the evaluation of the neighborhood solutions (Line 3), the sequential TSOCBA (Algorithm 2) is employed to decide the number of samples for the solutions. Then, the sample means are calculated and used for the estimation of $y(x_0), \ldots, y(x_k)$. All other steps remain the same.

In Algorithm 1, we need to decide the initial number of samples $n^0$, and the budget increment $\Delta$. The selection of these two parameters are well discussed in Chen et al. (2000). The value $n^0$ should not be too small to avoid poor estimations of the mean and the variance, which misleads the budget allocation. Also, a large $n^0$ may reduce the number of samples that can be allocated according to the policy. Both may lower the PCM. A suitable choice for $n^0$ is between 5 and 20. On the other hand, the compromise exists for $\Delta$: a large $\Delta$ can result in the waste of budgets to obtain an unnecessarily high accuracy of some solutions; whilst if $\Delta$ is small, one may need to update the sampling ratios for many times, increasing the overall computational time. A suggested choice for $\Delta$ is a value bigger than 5 but smaller than 10% of the neighborhood size.

## 5. Numerical results

In this section, we first show numerically the efficiency of TSOCBA on improving the PCM in a single iteration budget allocation problem. Different neighborhood landscapes are considered. Then, to validate the usefulness of TSOCBA on improving the performance of TS, we choose the following problems: (1) An (s,S) inventory problem (Koenig and Law, 1985). This problem is a well-known testing bed in the SO literature. With its 2-dimensional integer lattice solution space,

**Algorithm 2:** Sequential budget allocation with TSOCBA

**Require:** initial budget $n^0$; total budget $n$; budget increment $\Delta$;

**Ensure:** number of samples $n_0, n_1, \ldots, n_k$;

**Initial allocation**: Collect $n^0$ simulation samples $Y_1(x_i), \ldots, Y_{n^0}(x_i)$ for each neighborhood solution; set $n_i \leftarrow n^0, \forall i = 1, \ldots, k$; set the total budget counter $n^* \leftarrow kn^0$;

**while** $n^* < n$ **do**

  **Updating**: Compute the sample mean $\bar{Y}(x_i; n_i)$ and the sample variance $S_i^2 = \sum_{j=1}^{n_i} (Y_j(x_i) - \bar{Y}(x_i; n_i))^2/(n_i - 1)$, for $i = 0, 1, \ldots, k$;

  **Allocation**:

  Set $\Delta^* \leftarrow \min\{\Delta, n - n^*\}$, update $n^* \leftarrow n^* + \Delta^*$;

  **if** $\min_{i=1,\ldots,k} \bar{Y}(x_i; n_i) \geq \bar{Y}(x_0; n_0)$ **then**

    Compute the sampling ratios $\alpha_0, \alpha_1, \ldots, \alpha_k$ using Eq.(13) in Proposition 1;

  **else**

    Compute the sampling ratios $\alpha_0, \alpha_1, \ldots, \alpha_k$ using Eq.(15) in Proposition 2;

  **end**

  Calculate the number of samples $n_i' = \alpha_i n^*, \forall i = 0, 1, \ldots, k$;

  **Sampling**:

  **for** $i = 1$ **to** $\Delta^*$ **do**

    Select the most starving solution $i^*$ in terms of the need of additional budget, where $i^* = \arg\max_{i \in 0,1,\ldots,k} n_i' - n_i$;

    Collect one additional simulation sample for solution $i^*$, and update $n_{i^*} \leftarrow n_{i^*} + 1$;

  **end**

**end**

we can visualize and compare the search trajectory of TS under different budget allocation policies. (2) A production line throughput maximization problem (Pasupathy and Henderson, 2011). This problem is quite important in manufacturing, and is also a common benchmark problem for SO approaches. Due to its large solution space, solving it with traditional R&S approaches requires a great simulation effort. Hence, it is worthwhile to investigate the potential of TSOCBA on solving this problem. (3) A physician scheduling problem (Niroumandrad and Lahrichi, 2018). This real-world problem is taken from our previous work which is critical for the patient wait time and processing time. This problem is of high complexity due to different decision types and a huge solution space. This serves for testing the performance of TSOCBA in scheduling problems with complex solution structures. All experiments are coded in C++ and run in a computing cluster with Intel Gold Skylake @2.4 GHz processors.

The following two budget allocation methods are adopted as benchmarks:

**EA**: Equal Allocation. At each iteration, the budget is allocated equally to the neighborhood solutions, i.e., $\alpha_i = 1/k, \forall i = 1, \ldots, k$.. EA is the most frequently used policy when TS is adopted for SO.

**OCBA**: Optimal Computing Budget Allocation. At each iteration, the budget is allocated to the neighborhood solutions using the OCBA formulas proposed by Chen et al. (2000):

$$\frac{\alpha_i}{\alpha_j} = \frac{\sigma_i^2/(y_i - y_g)^2}{\sigma_j^2/(y_j - y_g)^2}, \quad \forall i \neq j \neq g, \tag{16a}$$

$$\alpha_g = \sigma_g \sqrt{\sum_{i \neq g} \alpha_i^2/\sigma_i^2}, \tag{16b}$$

$$\alpha_g + \sum_{i \neq g} \alpha_i = 1. \tag{16c}$$

where $g = \arg\min_{i \in [1,\ldots,k]} y(x_i)$. There are works that apply the original OCBA to metaheuristics for handling the budget allocation. For example, Horng et al. (2012) incorporated directly OCBA into particle swarm optimization. Although OCBA is proposed for the R&S problem rather than the specific search algorithm, according to the results of Zhang et al. (2016), some improvements can still be obtained. Therefore, we use the direct incorporation of OCBA into TS as a benchmark. Note that OCBA is quite similar, but not identical, to Proposition 2. The OCBA approach aims only at selecting the best neighborhood solution, thus no additional budget is spent on the best-known solution. We implement OCBA as a sequential allocation procedure as Algorithm 2, except that Eq.(16) is used for the Allocation step.

### *5.1. A single iteration budget allocation problem*

We consider the budget allocation problem in a single TS iteration. To simulate different neighborhood landscapes, four instances are introduced in Table 1. Instance A1 represents a case of Scenario Best-Holding, i.e., the neighborhood solutions are not better than the best-known solution. Here, $x$ and $x_0$ are the current and the best-known solution respectively, and $x_1, \ldots, x_9$ are the nine neighborhood solutions. The simulation output $Y(x_i) \sim \mathcal{N}(i, \sigma^2)$ for $i = 0, \ldots, 9$, where $\sigma = 6$. The tabu neighborhood solution set is $J = \{1, 2\}$. The correct TS move is therefore

Table 1: Test instances of the single iteration allocation problem

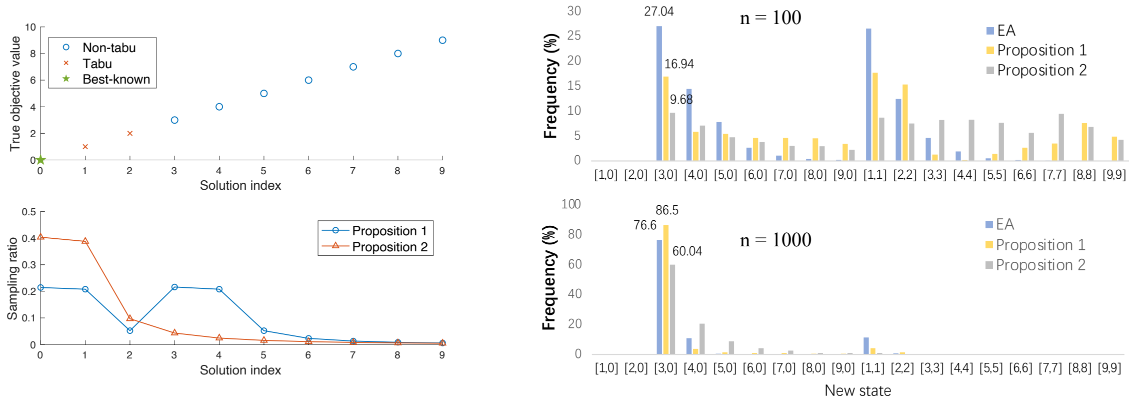
| Instance ID | Scenario | $y(x_0)$ | $y(x_1),\ldots,y(x_9)$ | $\sigma$ | Tabu neighbor set $J$ | Correct TS Move |
|---|---|---|---|---|---|---|
| A1 | Best-Holding | 0 | 1,2,3,4,5,6,7,8,9 | 6 | $1,2$ | $(x,x_0)\to(x_3,x_0)$ |
| A2 | Best-Holding | 0 | 1,2,3,4,5,6,7,8,9 | 6 | $3,4$ | $(x,x_0)\to(x_1,x_0)$ |
| B1 | Best-Improving | 1 | 0,2,3,4,5,6,7,8,9 | 6 | $3,4$ | $(x,x_0)\to(x_1,x_1)$ |
| B2 | Best-Improving | 1 | 0,2,3,4,5,6,7,8,9 | 6 | $1,2$ | $(x,x_0)\to(x_1,x_1)$ |

$m(s^t) : (x,x_0) \to (x_3,x_0)$. Instance A2 is also of Best-Holding, but compared to A1, the gap between $y(x_0)$ and $y(x_b)$ is much smaller. This case is often faced during the escaping from the local optimum. Instance B1 and B2 are of Best-Improving. B1 is the case when the best neighborhood solution $x_g$ is not tabu. On the contrary, B2 is a special case when $x_g$ is prevented by the tabu list, and thus, the aspiration criteria is triggered. These instances almost cover all types of neighhorhood landscapes during a TS procedure.

### *5.1.1. Static allocation implememtation*

In this subsection, we study the performance of the proposed polices in static allocation implememtation. By "static allocation" , we refer to the ideal case where the perfect information of $y(x_i)$'s and $\sigma_i$'s are known in advanced, and thus, the correct sampling ratios can be calculated. A finite budget $n$ is allocated by a single step according to the correct sampling ratios. We compare the performance of Proposition 1 and 2, as well as EA in such ideal case, with two different levels of $n = 100$ and $n = 1000$. Note that this ideal experiment aims only at showing the difference between the allocations and their impacts on the TS move.

We run the experiment only on Instance A1. Figure 2(a) illustrates the budget allocation of Proposition 1 and 2. Figure 2(b) depicts the probability of moving to each new state when different allocations are adopted. This probability is evaluated by Monte Carlo simulations with 5000 replications. For simplicity, the figure uses $[i,j]$ to represent the state $(x_i,x_j)$. We see that it is difficult to recognize the correct state $(x_3,x_0)$ when $n$ is small. As $n$ increases to 1000, the asymptotically optimal policy, namely, Proposition 1, obtains the best performance. However, we also observe that the PCM of Proposition 1 is worse than EA when $n = 100$. This implies that the asymptotically optimal sampling ratio per se does not guarantee a good performance when the simulation budget is not large enough, which coincides with the finding in Peng et al. (2018a).




(a) Sampling ratios of propositions

(b) Probability of TS moves

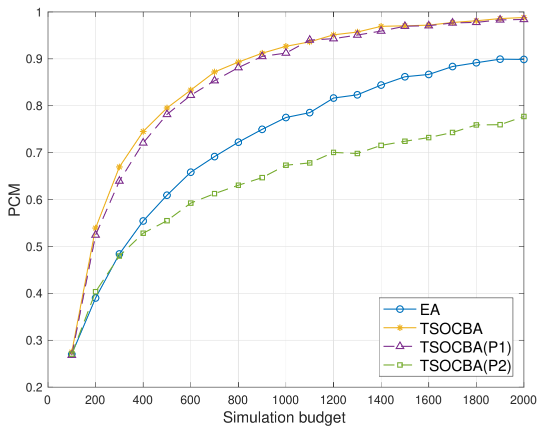
Figure 2: Comparison of difference policies with static allocation implememtation on Instance A1

*5.1.2. Sequential allocation implememtation*

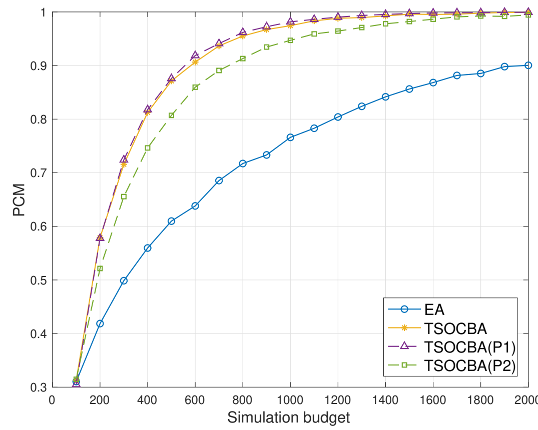
In actual cases, the $y(x_i)$'s, $\sigma_i$'s and the scenario $\mathscr{S}$ are unknown. A static allocation with perfect information is not allowed, and hence, the sequential procedure utilizing plug-in estimated parameters is adopted. In the following experiment, we study the performance of three sequential allocation procedures. TSOCBA is implemented as Algorithm 2. TSOCBA(P1) is implemented as TSOCBA except that the scenario $\mathscr{S}$ is fixed as Best-Holding and Proposition 1 is adopted for budget allocation. TSOCBA(P2) is implemented in a similar logic. We set $n^0 = 10$ and $\Delta = 1$, $n$ increases from 100 to 2000. The PCM is evaluated empirically by Monte Carlo simulations with 10000 replications.

The result of Instance A1 is given in Figure 3 (a). As expected, TSOCBA(P1) outperforms EA and TSOCBA(P2). TSOCBA has almost identical performance as TSOCBA(P1). TSOCBA(P2) converges slower than EA. We notice that when $n = 1000$, the PCM of the sequential approach TSOCBA(P1) is 0.912, which is actually higher than the PCM obtained by the static allocation approach with the same policy (0.865 as shown in Figure 2(b)). This validates the finding that the sequential procedure using posterior estimations of $y(x)_i$'s and $\sigma_i$'s can outperform the optimal static allocation using perfect information (Chen et al., 2006).

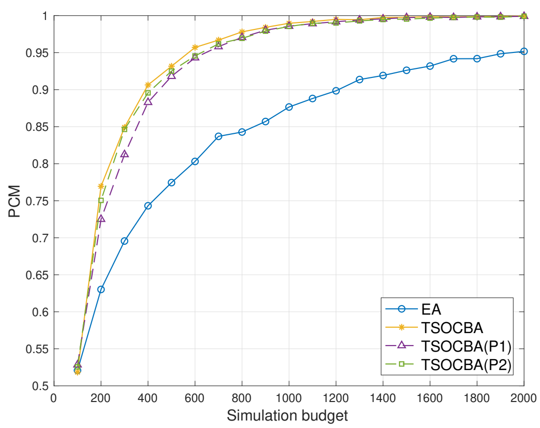
The results of A2, B1 and B2 are given in Figure 3. Basically, TSOCBA(P1) achieves the best performance for Best-Holding, whilst TSOCBA(P2) is the best for Best-Improving. However,



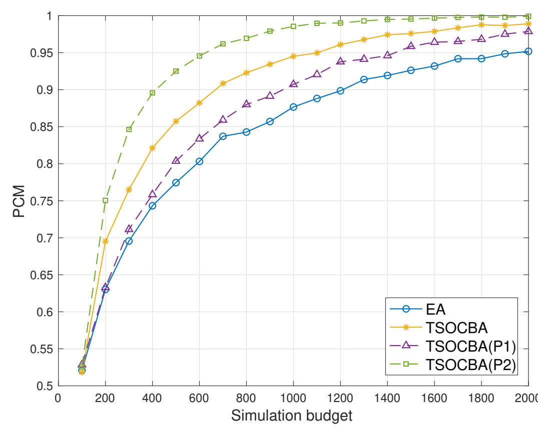

(a) Instance A1




(b) Instance A2




(c) Instance B1



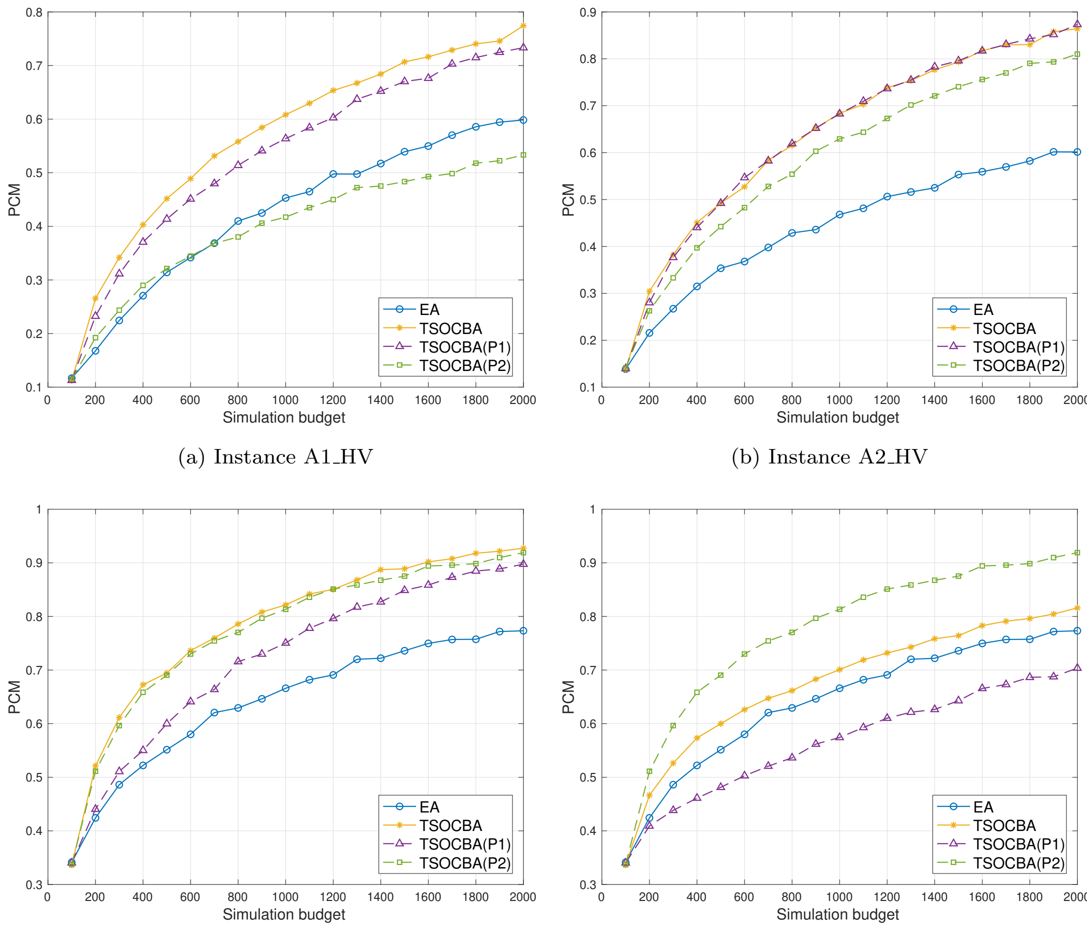

(d) Instance B2

Figure 3: PCM as a function of the simulation budget on the instances with $\sigma = 6$

these single-policy procedures cannot perform well when the scenario is not consistent. TSOCBA is robust and almost the best, except in instance B2 where it is outperformed by TSOCBA(P2).

*5.1.3. Sequential allocation implememtation - high simulation noises*

To test the performance in the case of high simulation noises, we generate the variants of the testing instances. Instances A1_HV, A2_HV, B1_HV and B2_HV have the same parameters as A1, A2, B1 and B2 respectively, except that $\sigma = 12$. The results are reported in Figure 4. Generally, the performance of all procedures becomes worse than those in the low-variance cases. This can be well




(a) Instance A1_HV

(b) Instance A2_HV

(c) Instance B1_HV

(d) Instance B2_HV

Figure 4: PCM as a function of the simulation budget on the instances with $\sigma = 12$

understood because the high noise level increases the difficulty in selecting the correct move. The rankings of different procedures remain almost the same in each instance. It is interesting to observe that in A1_HV, TSOCBA has an advantage over TSOCBA(P1) that uses perfect information on $\mathscr{S}$. This implies that the posterior estimation of $\mathscr{S}$ leads to a desirable allocation policy in certain circumstances. However, this is not always the case. As shown in B2_HV, TSOCBA is outperformed by TSOCBA(P2) by a large extent, which shows the penalty of not knowing the true $\mathscr{S}$. While in A2_HV and B1_HV, there is no significant difference between the performance of TSOCBA and the procedure using perfect information on $\mathscr{S}$. Indeed, in many classic R&S problems, the sequential implementation using posterior estimations of parameters could lead to a superior performance. However, it could also lead to misleading results in the low-confidence scenario characterizing by: the differences between the means of solutions are small, the variances are large, and the simulation budget is small (Peng et al., 2018a). As observed in our case, the sequential implementation based on estimated $\mathscr{S}$ is not worse than that using true $\mathscr{S}$ in most of the neighborhood configurations, even with large simulation noises.

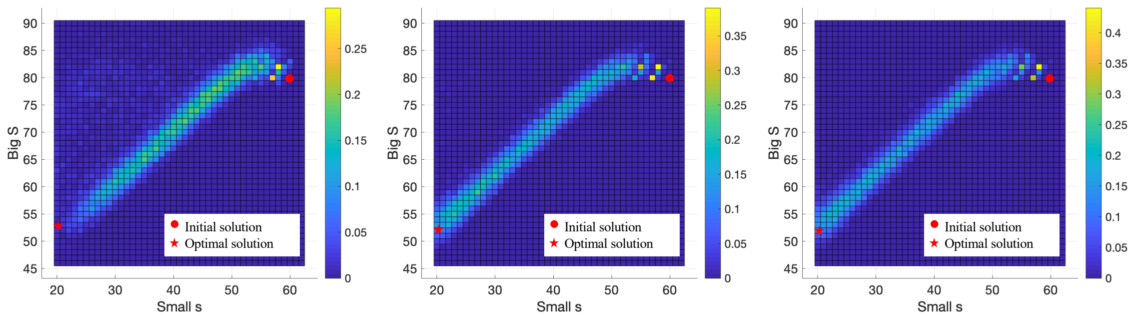
In summary, the proposed TSOCBA outperforms EA and the two single-policy procedures in almost all the test instances of the single iteration budget allocation problem.

### *5.2. The (s,S) Inventory problem*

We consider an inventory system with zero delivery lag and backlogging, which is introduced in Koenig and Law (1985). At the beginning of each period, if the inventory level is below $S_{small}$, an order is placed and the inventory level is replenished to $S_{big}$. Placing an order incurs a constant ordering cost of 32, and a variable ordering cost of 3 per unit. The demand in each period is i.i.d. and follows the Poisson distribution with mean 25. After subtracting the demand, for each unit remaining in the inventory, a holding cost of 1 per unit is incurred; unsatisfied demand incurs a penalty of 3 per unit, and becomes the backlog for the next period. The goal is to find a policy $x = (S_{small}, S_{big}) \in \Theta$, where $\Theta = \{(S_{small}, S_{big}) \in \mathbb{Z}^2 : 20 \leq S_{small} \leq 80, 40 \leq S_{big} \leq 100, S_{small} < S_{big}\}$, that minimizes the long-term expected inventory cost per period. The optimal solution is $x^* = (20, 53)$ with $y(x^*) = 111.1265$, which is found by a complete enumeration and using the Markov chain method presented in Zheng and Federgruen (1991) to evaluate exactly the solutions.

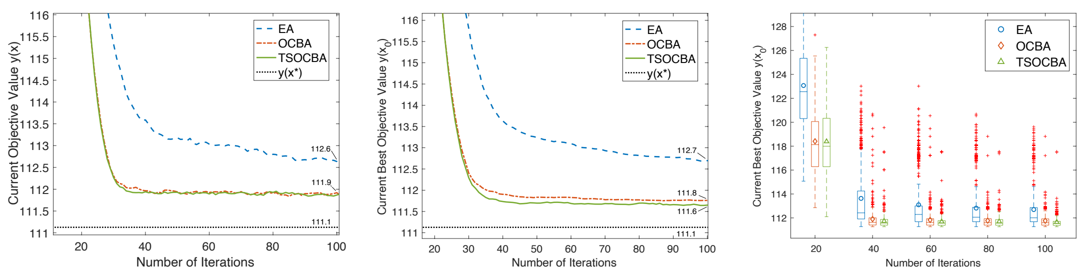
To apply TS, the neighborhood is defined as $\mathbb{N}(x, rad) = \{x' \in \Theta : ||x' - x|| \leq rad, x' \neq x\}$, where $rad$ is the radius of the neighborhood, and $||x||$ is the norm of vector $x$. After the move $x \to x'$, the reverse direction vector $d_r = -(x' - x)$ is added into the tabu list $T$. Let $x$ be the current solution, given a vector $d_r$, any neighborhood solution is tabu if located in the cone $R(x, d_r, \theta) = \{x' \in \mathbb{N}(x, rad) : \arccos \frac{(x'-x)\cdot d_r}{||x'-x|| ||d_r||} \leq \theta\}$, where $\theta$ is an user-defined tabu angle. In this way, the tabu neighborhood subset is $\bigcup_{d_r \in T} R(x, d_r, \theta)$.

We set $rad = 3$, $\theta = 15^{\circ}$, $t_{max} = 100$, $x^0 = (60, 80)$, and the maximum length of $T$ as 4. To avoid solution re-evaluating, the solutions visited in the last three iterations are removed from the current neighborhood. $y(x)$ is evaluated by simulation with a total simulation length of $l$




(a) EA (b) OCBA (c) TSOCBA

Figure 5: A visualization of the TS trajectory in the solution space of the (s,S) inventory problem ($n = 300$ and $l = 10$). The brightness of a pixel represents the $\rho_{30}(x)$ value estimated from 500 replications.




(a) Objective value of current solution (b) Objective value of the best-known solution (c) Boxplot of the best-known solution value

Figure 6: Tabu search performance on the (s,S) inventory problem, $n = 300$, $l = 10$ (Data from 500 replications)

periods. A warm-up of 50 periods is used to guarantee the steady-state performance. To study the TS performance with different budget allocation procedures, levels of simulation noise and total budget, a full-factorial design of experiments (DOE) is performed with the following factors and levels: budget allocation procedure ($Alloc$) - EA, OCBA, TSOCBA; budget per iteration ($n$) - 200, 250, 300; simulation length ($l$) - 10, 30, 50. Each setting runs for 500 replications. For OCBA and TSOCBA, $n^0 = 5$ and $\Delta = 1$. The simulation noise level corresponding to each $l$ level is evaluated by the Coefficient of Variance (CV). For $l = 10$, 30 and 50, the CV of $Y(x^*)$ is 0.0695, 0.0372 and 0.0275, respectively.

There are 3*3=9 different cases in terms of $n$ and $l$. Take the case of ($n = 300$, $l = 10$) as an example. Figure 5 illustrates the search trajectory of TS. It depicts the probability that TS visits a specific solution along its search path $[x^0, x^1, \ldots, x^t]$. The probability of visiting a solution by

Table 2: Probability of visiting, recognizing, and maintaining the optimal solution of the (s,S) Inventory problem (Estimated from 500 replications)

| Methods | $\rho_{30}(x^*)$ | $\rho^*_{30}(x^*)$ | $\rho^0_{30}(x^*)$ | $\rho_{100}(x^*)$ | $\rho^*_{100}(x^*)$ | $\rho^0_{100}(x^*)$ |
|---|---|---|---|---|---|---|
| EA | 0.008 | 0.004 | 0.002 | 0.774 | 0.05 | 0.008 |
| OCBA | 0.13 | 0.04 | 0.032 | 0.982 | 0.2 | 0.056 |
| TSOCBA | 0.136 | 0.068 | 0.044 | 0.994 | 0.572 | 0.074 |

iteration $t$, given by $\rho_t(x) = P(x \in [x^0, x^1, \ldots, x^t])$, is estimated empirically from 500 replications. A brighter spot indicates a higher frequency of visiting. As shown, when EA is adopted, the search can hardly reach the optimum. Also, we observe some bright spots scattered in the left-up corner of the solution space. Both indicate the misleading effect of the simulation noises on the search path. Whilst when OCBA or TSOCBA is used, a clear path links the initial solution to the optimal solution region. The estimated $\rho_{30}(x^*)$ values are reported in Table 2, TSOCBA leads to the highest chance of visiting the optimum.

Figure 6(a) depicts the averaged quality of the search path. As shown, OCBA and TSOCBA converge faster than EA. Since $x_0$ is the final output of TS, we report the averaged $y(x_0)$ value in Figure 6(b). TSOCBA has a small advantage over OCBA. This gap is due to a more accurate update of $x_0$, which can be shown in Table 2. Here, $\rho^*_t(x^*) = P(x^* \in [x^0_0, x^1_0, \ldots, x^t_0])$ is defined as the probability that TS has stored the optimal solution $x^*$ as the best-known solution during the search, $\rho^0_t(x^*) = P(x^t_0 = x^*)$ is the probability that TS has maintained $x^*$ as the best-known solution at iteration $t$. Figure 6(c) provides the boxplot of $y(x_0)$. Here, the averages of the data are labeled by markers. As shown, TSOCBA has the smallest gap between the first and third quartile (except for the 20th iteration), showing a stable performance. Further, it can be seen from the outliers that TSOCBA achieves the best worst-case performance.

Define *Optimality Gap at Iteration* as $OptGap_t = (y(x^t_0) - y(x^*))/y(x^*) * 100$. We use $OptGap_{30}$ and $OptGap_{100}$ for measuring the convergence speed and quality, respectively. Table 3 reports the comparisons results in all cases. The mean and standard deviation of the data are reported. Besides, EA and OCBA are compared to TSOCBA via the Mann-Whitney U test with a significance level of 5%. The symbols "+", "=" and "-" indicate that the method is statistically worse, equal and better than TSOCBA, respectively. As shown, TSOCBA is the best among the competitors in terms of both indicators, regardless of the total budget number and simulation noise level. The

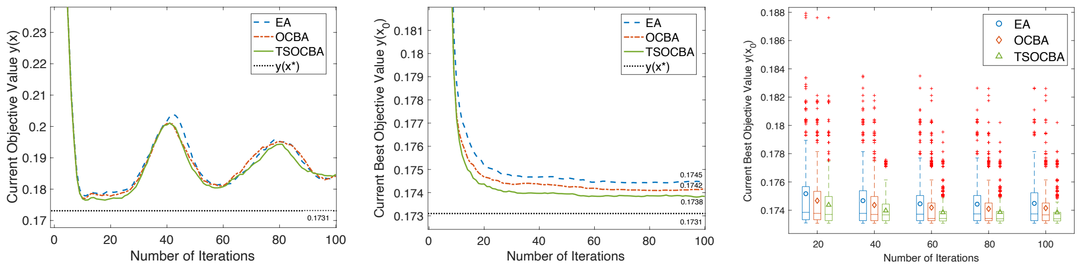
Table 3: Comparison results of the (s,S) inventory problem

| #Budget | Sim.length | $OptGap_{30}$ Mean (Std) | | | $OptGap_{100}$ Mean (Std) | | |
|---|---|---|---|---|---|---|---|
| | | EA | OCBA | TSOCBA | EA | OCBA | TSOCBA |
| 200 | 10 | 6.54 (3.9) + | 3.71 (2.8) = | **3.51** (2.5) | 1.86 (2) + | 0.832 (1.1) = | **0.789** (0.91) |
| | 30 | 0.791 (1) + | 0.456 (0.59) = | **0.406** (0.45) | 0.459 (0.38) + | 0.357 (0.3) + | **0.285** (0.25) |
| | 50 | 0.421 (0.47) = | 0.26 (0.23) = | **0.259** (0.22) | 0.341 (0.27) + | 0.25 (0.21) + | **0.215** (0.18) |
| 250 | 10 | 5.17 (3.3) + | 2.13 (2) = | **1.99** (1.8) | 1.6 (1.8) + | 0.69 (0.84) + | **0.593** (0.68) |
| | 30 | 0.599 (0.75) + | 0.319 (0.34) + | **0.265** (0.19) | 0.4 (0.33) + | 0.324 (0.28) + | **0.231** (0.19) |
| | 50 | 0.352 (0.41) + | 0.221 (0.19) = | **0.2** (0.14) | 0.309 (0.24) + | 0.228 (0.21) + | **0.164** (0.12) |
| 300 | 10 | 4.22 (2.9) + | 1.15 (1.2) + | **1.05** (1.1) | 1.41 (1.6) + | 0.57 (0.55) + | **0.471** (0.47) |
| | 30 | 0.484 (0.72) + | 0.275 (0.21) + | **0.248** (0.17) | 0.355 (0.27) + | 0.27 (0.23) + | **0.215** (0.15) |
| | 50 | 0.322 (0.39) + | 0.21 (0.17) + | **0.181** (0.13) | 0.274 (0.22) + | 0.215 (0.18) + | **0.152** (0.1) |
| Average | | 2.1 (3.1) + | 0.969 (1.7) + | **0.901** (1.6) | 0.779 (1.2) + | 0.415 (0.57) + | **0.346** (0.48) |

overall performance of TSOCBA is statistically better than EA and OCBA. Moreover, in all cases, TSOCBA is more stable than EA and OCBA due to a smaller standard deviation. A detailed ANOVA can be found in Appendix H.

### *5.3. The throughput maximization problem*

We consider the throughput maximization problem taken from SimOpt.org (Pasupathy and Henderson, 2011). There is a three-machine flow line with finite capacity buffer in front of Machine 2 and 3. The number of buffer storage (including the one in service at the machine) are denoted as $\zeta_4$ and $\zeta_5$. The service time of machine $i$ is exponentially distributed with service rate $\zeta_i, i = 1, 2, 3$. A full buffer causes the blockage of the upstream machine, and an empty buffer causes the starvation of the downstream machine. The *block-after-service* behavior is considered, i.e., a finished job cannot be released from a machine if the downstream buffer is full. Machines are considered to be reliable and no failure can happen. The first machine is never starved, and the last machine is never blocked. The total buffer capacity and the total service rates are limited. The goal is to find an allocation of buffer capacities and service rates $x = \{\zeta_1, \ldots, \zeta_5\}$ such that the steady-state throughput (TH) of the flow line is maximized, which can be formulated as a minimization problem: $\min_x 1/TH(x)$, subjected to $\zeta_1 + \zeta_2 + \zeta_3 = 20$; $\zeta_4 + \zeta_5 = 20$; $1 \leq \zeta_i \leq 20$ and $\zeta_i \in \mathbb{Z}^+$, for $i = 1, \ldots, 5$. The throughput is evaluated by simulation with a warm-up of 2000 jobs followed by a period of $l$ jobs. This problem has totally 21660 feasible solutions. The optimal solutions are known as $(6, 7, 7, 12, 8)$ and $(7, 7, 6, 8, 12)$. The optimal throughput is 5.776 and hence, $y(x^*) = 1/5.776 = 0.1731$.




(a) Objective value of current solution (b) Objective value of the best-known solution (c) Boxplot of the best-known solution value

Figure 7: Tabu search performance on the throughput maximization problem, $n = 100$, $l = 50$. (Data from 500 replications)

To apply TS, define the neighborhood by $\mathbb{N}(x) = \{x(i,j) \in \Theta, \forall i,j \in 1,...,5, i \neq j\}$, where $x = \{\zeta_1, \ldots, \zeta_5\}$ is the current solution, $x(i,j)$ is the solution obtained by moving one unit from $\zeta_i$ to $\zeta_j$, i.e., $\zeta_i \leftarrow \zeta_i - 1$ and $\zeta_j \leftarrow \zeta_j + 1$. $|\mathbb{N}(x)| = 8$ if $x$ is not on $\Theta$'s boundary. After moving one unit from $\zeta_i$ to $\zeta_j$, we tabu the reversing move from $\zeta_j$ to $\zeta_i$ in the next 2*#stations = 6 iterations. We remove the last three visited solutions from the current neighborhood, if applied. Set $x^0 = (14, 3, 3, 17, 3)$, $t_{max} = 100$. We perform a full-factorial DOE with the following factors and levels: budget allocation procedure ($Alloc$) - EA, OCBA, TSOCBA; budget per iteration ($n$) - 50, 100, 150; simulation length ($l$) - 30, 50, 70. Each experiment setting runs for 500 replications. For OCBA and TSOCBA, $n^0 = 5$ and $\Delta = 1$. The CV of $Y(x^*)$ is 0.1668, 0.1244 and 0.1022 for $l = 30$, 50 and 70, respectively.

The results of the case ($n = 100$, $l = 50$) is given in Figure 7, where the true value $y(x^t)$ is estimated by the sample mean from 500 replications of long-term simulation of 20000 jobs. Figure 7(a) shows that the search path quality is similar for all methods, which may due to a relatively small neighborhood size. However, TSOCBA can output a better solution (as depicetd in 7(b)), which is the result of a more accurate $x_0$ updating. Futher, Figure 7(c) shows that TSOCBA has a lower variability and a better worst-case performance than EA and OCBA.

The case-by-case comparison results are reported in Table 4. TSOCBA obtains the best averaged performance for both indicators in all cases, and in most of them the advantages of TSOCBA are statistically significant. The overall performance of TSOCBA statistically outperforms the two benchmarks in terms of the optimality gap in both early and late search phase. Furthermore, the standard deviation of TSOCBA is the smallest in all cases, which indicates a better performance

Table 4: Comparison results of the throughput maximization problem

| #Budget | Sim.Length | $OptGap_{20}$ Mean (Std) | | | $OptGap_{100}$ Mean (Std) | | |
|---|---|---|---|---|---|---|---|
| | | EA | OCBA | TSOCBA | EA | OCBA | TSOCBA |
| 50 | 30 | 2.91 (3.3) + | 2.71 (3) + | **2.36** (2.8) | 1.48 (1.7) = | 1.77 (2.4) + | **1.45** (1.9) |
| | 50 | 1.99 (2.4) + | 1.81 (2) + | **1.56** (1.9) | 1.17 (1.2) + | 1.24 (1.5) + | **0.966** (1.1) |
| | 70 | 1.6 (1.8) + | 1.33 (1.6) + | **1.12** (1.5) | 1.03 (1.2) + | 1.13 (1.3) + | **0.751** (0.99) |
| 100 | 30 | 1.73 (2.1) + | 1.37 (1.7) + | **1.04** (1.3) | 1.07 (1.2) + | 0.896 (1.1) + | **0.615** (0.73) |
| | 50 | 1.21 (1.5) + | 0.921 (1.2) + | **0.748** (1) | 0.81 (0.98) + | 0.619 (0.84) + | **0.438** (0.58) |
| | 70 | 0.896 (1.1) + | 0.689 (0.92) + | **0.481** (0.62) | 0.653 (0.8) + | 0.538 (0.77) + | **0.356** (0.43) |
| 150 | 30 | 1.38 (1.7) + | 0.961 (1.3) = | **0.823** (1) | 0.855 (1) + | 0.657 (1.1) + | **0.484** (0.61) |
| | 50 | 1.06 (1.2) + | 0.647 (0.84) + | **0.48** (0.65) | 0.605 (0.76) + | 0.483 (0.61) + | **0.363** (0.48) |
| | 70 | 0.752 (1) + | 0.489 (0.67) = | **0.414** (0.53) | 0.549 (0.68) + | 0.431 (0.58) + | **0.267** (0.32) |
| Average | | 1.5 (2) + | 1.21 (1.7) + | **1** (1.5) | 0.914 (1.1) + | 0.862 (1.3) + | **0.632** (1) |

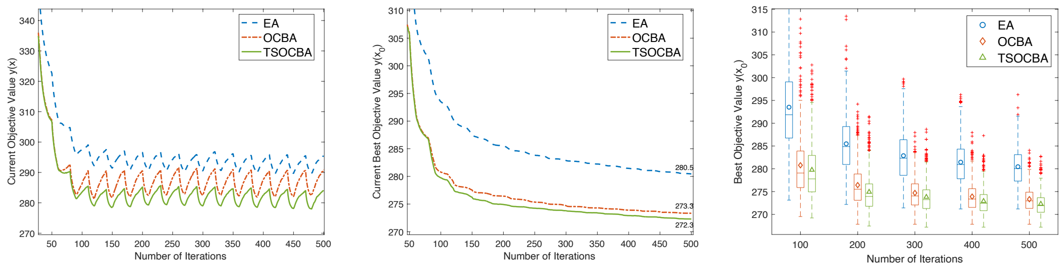
robustness. See Appendix H for the ANOVA details.

### *5.4. The physician scheduling problem*

In this section, we consider a real-world physician scheduling problem in the pretreatment phase for cancer patients (Niroumandrad and Lahrichi, 2018). See Appendix G for the details of the problem and adaption of the TS algorithm.

A full-factorial DOE is performed with the following factors and levels: budget allocation procedure (*Alloc*) - EA, OCBA, TSOCBA; budget per iteration ($n$) - 2000, 3000, 4000; objective weight ($w$) - 0.2, 0.5, 0.8 ( the simulation noise level increases with $w$). The CV of a randomly generated initial solution is 0.0101, 0.0203 and 0.0399 for $w = 0.2$, 0.5 and 0.8, respectively. Each experiment setting runs for 500 replications. For OCBA and TSOCBA, $n^0 = 5$ and $\Delta = 1$.

The results of the case ($n = 3000$, $w = 0.5$) is depicted in Figure 8. Here, the value of $y(x^t)$ is estimated by simulation with 10000 replications. As shown in Figure 8(a), OCBA and TSOCBA converge faster than EA due to a better descent direction. OCBA and TSOCBA have nearly the same performance at the early search phase, while after the 60th iteration, TSOCBA can reach a better solution space. Figure 8(b) shows that TSOCBA outperforms OCBA in terms of $y(x_0)$ after the 80th iteration. Indeed, at the early search phase, it is easier to gain improvements of the objective value and TS performs like a greedy algorithm picking the best neighborhood solution, which aligns to the goal of OCBA. As the procedure continues and the landscape becomes more complex, TS keeps entering and escaping from local optima under the tabu mechanism. Without




(a) Objective value of current solution (b) Objective value of the best-known solution (c) Boxplot of the best-known solution value

Figure 8: Tabu search performance on the physician scheduling problem, $n = 3000$, $w = 0.5$. (Data from 500 replications)

considering such ad-hoc mechanism in the budget allocation, OCBA is not able to perform as well as TSOCBA. Figure 8(c) shows that TSOCBA is more reliable and achieves a better worst-case performance than EA and OCBA.

The case-by-case comparison results are reported in Table 5. TSOCBA obtains the best averaged $OptGap_{100}$ and $OptGap_{500}$ in almost all cases. Exceptions are observed in ($w = 0.2$, $n = 3000$) and ($w = 0.2$, $n = 4000$), which are considered as the "easy" cases because of the low noise level and large amount of budget. The overall performance of TSOCBA statistically outperforms the two benchmarks in terms of optimality gap both in early and late search phase. Moreover, in almost all cases, TSOCBA obtains a smaller standard deviation than EA and OCBA, which implies a more stable performance. The ANOVA details are provided in Appendix H.

### *5.5. A Comparison to OptQuest*

To further validate the performance of TS with the proposed budget allocation policy (short as TSOCBA) on solving SO problems, besides TS with equal allocation (short as EA), we use the OptQuest® optimization engine as a benchmark, whose performance is well-recognized by simulation optimization practitioners. The main optimization procedure of OptQuest is based on scatter search, with various techniques included to complement the default search mechanisms, such as genetic algorithm, particle swarm optimization, cross entropy and simultaneous perturbation stochastic approximation.

The performance comparison is made on the throughput maximization problem with $n = 100$, $l = 50$, and the physician scheduling problem with $n = 3000, w = 0.5$. For a fair comparison,

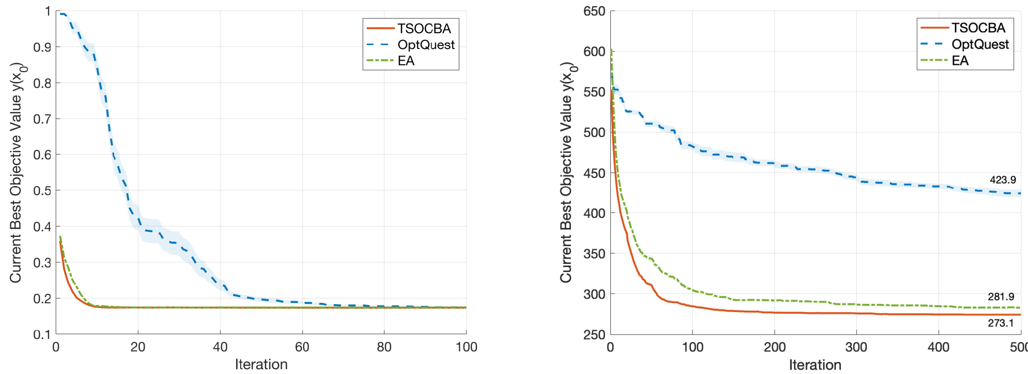
Table 5: Comparison results of the physician scheduling problem

| Weight | #Budget | *OptGap*$_{100}$ Mean (Std) | | | *OptGap*$_{500}$ Mean (Std) | | |
|---|---|---|---|---|---|---|---|
| | | EA | OCBA | TSOCBA | EA | OCBA | TSOCBA |
| 0.2 | 2000 | 2.15 (1) + | 1.32 (0.83) = | **1.25** (0.73) | 1.4 (0.63) + | 0.748 (0.53) = | **0.72** (0.49) |
| | 3000 | 1.84 (0.9) + | **1.11** (0.7) = | 1.14 (0.65) | 1.21 (0.58) + | **0.545** (0.44) - | 0.685 (0.44) |
| | 4000 | 1.66 (0.82) + | **1.1** (0.75) - | 1.16 (0.64) | 1.02 (0.53) + | **0.531** (0.47) - | 0.693 (0.46) |
| 0.5 | 2000 | 8.41 (2.5) + | 5.05 (2.1) + | **4.25** (2) | 6.71 (2) + | 3.97 (1.7) + | **3.13** (1.5) |
| | 3000 | 6.84 (2.4) + | 3.45 (1.7) + | **2.89** (1.6) | 5.33 (1.8) + | 2.51 (1.2) + | **2.14** (1.1) |
| | 4000 | 5.94 (2.1) + | 2.93 (1.6) + | **2.6** (1.5) | 4.63 (1.6) + | 2.21 (1.2) + | **1.83** (1) |
| 0.8 | 2000 | 14.4 (3.1) + | 11.5 (2.9) + | **10.9** (2.7) | 11.6 (2.4) + | 9.05 (2.4) + | **8.57** (2.5) |
| | 3000 | 12.5 (2.9) + | 8.53 (2.4) + | **7.5** (2.4) | 10.1 (2.2) + | 6.43 (1.9) + | **5.39** (1.8) |
| | 4000 | 11.8 (2.6) + | 7.75 (2.2) + | **7.02** (2.1) | 9.07 (2.1) + | 5.81 (1.9) + | **4.91** (1.7) |
| Average | | 7.27 (5.1) + | 4.75 (4) + | **4.3** (3.7) | 5.67 (4.1) + | 3.53 (3.2) + | **3.12** (2.9) |

all algorithms consume the same simulation budget $n$ at each search iteration, and use the same termination condition. We run the algorithms for 50 replications on the throughput maximization problem, and 10 replications for the physician scheduling problem. Note that the initial solution of TSOCBA is generated randomly, while OptQuest uses its default initialization mechanism.

The results are shown in Figure 9. Both the mean and 95% confidence intervals of the best-known objective value are illustrated. For the throughput maximization problem, all algorithms converge to the optimal objective value $y(x^*)$, but the speed of TSOCBA is much faster than OptQuest. The difference between TSOCBA and EA is relatively small. The confidence interval of TSOCBA is much narrower than OptQuest, showing a stable performance. For the complex physician scheduling problem, TSOCBA outperforms OptQuest to a large extent in terms of convergence speed and final solution quality, and there is a considerable performance gap between TSOCBA and EA.

Notably, the performance gap between EA and OptQuest is due to the different search mechanisms, while that between EA and TSOCBA is caused by the different simulation budget allocation policies. On the one hand, comparing to the general purpose optimization engine, we greatly improve the performance by designing ad-hoc TS algorithms for the problems; on the other hand, the performance can be further improved by optimizing the simulation budget allocation, and such benefit tends to increase with the problem complexity. Indeed, when tackling complex SO problems with large solution space, mitigating the misleading effect of simulation noises becomes more critical for an efficient search.




(a) Throughput maximization problem

(b) Physician scheduling problem

Figure 9: Comparison of tabu search coupled with TSOCBA, EA, and OptQuest Solver on the throughput maximization problem ($n = 100$, $l = 50$) and physician scheduling problem ($n = 3000, w = 0.5$). Results are averaged from multiple replications of experiments. 95% confidence intervals are illustrated as shaded bands.

### *5.6. Discussion*

The benefits of optimizing the budget allocation for TS come from two aspects: (a) A better search direction; (b) A higher accuracy on updating the best-known solution. These tackle, to some extent, the first and second difficulties in SO, respectively. More specifically, in the (s,S) inventory problem, TSOCBA outperforms EA because of both (a) and (b), and its advantage over OCBA is due to (b). In the throughput maximization problem, due to the small neighborhood size, all methods have similar search path qualities, and hence, (b) is the main factor distinguishing the methods. While in the real-world physician scheduling problem, due to (a), TSOCBA can better guide the search in the complex solution landscape, and helps TS reaching a more promising region than EA and OCBA. From a larger perspective, these benefits from optimizing the budget allocation by considering the specific search mechanisms should hold not only for TS, but also for other metaheuristics.

TS shows the potential of solving large-scale SO problems. Coupled with TSOCBA, TS can obtain quasi-optimal solutions for the (s,S) inventory problem and the throughput maximization problem. More specifically, using 100(iterations)$*$300(budget/iteration) $= 3 \times 10^4$ simulation samples (Sim.length $= 50$ periods), the proposed method results in an averaged optimality gap of 0.152 (%) for the (s,S) inventory problem. For the throughput maximization problem, the averaged optimality gap is 0.363 (%), which corresponds to an averaged TH of 5.756, using in total $1.5 \times 10^4$ simulation samples (Sim.length = 50 jobs). The consumed number of samples is significantly less than that by the R&S approaches. For example, Ni et al. (2017) tackled the same throughput maximization problem with R&S. Even for finding the best in a small subset of the solution space (3249 out of 21660 solutions), the proposed selection procedure spends $5.5 \times 10^5$ simulation samples (Sim.length = 50 jobs) to guarantee an indifference zone $\delta = 0.1$ (i.e., TH $\geq 5.766$) with a PCS of 0.95. Indeed, leveraging the topology of the solution space instead of a complete enumeration, TS is more efficient. However, unlike R&S, there is no evidence that TS would converge to the optimal solution as the budget goes to infinity, which is a common shortage as many metaheuristics. Still, the powerfulness and usefulness of TS have been shown in various difficult combinatorial optimization problems. Therefore, TS is quite promising for solving SO problems if the misleading effects of simulation noises can be properly mitigated.

TSOCBA reveals the optimal budget allocation on the tabu, non-tabu and best-known solutions. This policy remains valid no matter how the neighborhood solutions and tabus are generated. This implies that various techniques can be incorporated into the basic TS + TSOCBA framework to enhance its performance in SO. For example, metamodels can be adopted to recommend promising neighbors, avoiding an evaluation of the entire neighborhood. Other example is shown in Shylo and Shams (2018), where a machine learning model is trained for predicting high-quality solution attributes. These predictions are then used for controlling the update of tabus, guiding the search to promising regions.

## 6. Conclusions

In this research, we tackle the simulation budget allocation problem when tabu search (TS) is applied for stochastic simulation optimization. With the presence of simulation noises, the search direction of TS can be misled and thus, converging to high-quality solutions becomes difficult. The goal of this article is to improve the search efficiency by a proper allocation of the finite simulation budget. We formulate the problem as a maximization problem of the probability of correct move (PCM) in a single search iteration. A correct move refers to the move indicated by perfect information on the solution quality. Under the large deviations framework, we develop a surrogate model to maximize the exponential convergence rate of the PCM as the budget increases. By solving this surrogate model, we propose a new budget allocation policy for TS, and provide theoretical results on its asymptotic optimality. Given as closed-form formulas, the proposed policy, named TSOCBA, can be easily implemented in practice.

The efficiency of TSOCBA is first validated in a single iteration budget allocation problem with different neighborhood landscapes. Then, TSOCBA is coupled with TS to solve stochastic simulation optimization problems arising from different application scenarios. Design of experiments (DOE) is carried out to study the impacts of budget allocation policy, simulation noise level and the number of available budget. Through extensive numerical experiments, we show that TSOCBA can enhance the TS performance both in terms of convergence speed and final solution quality.

TSOCBA is compared to the conventional OCBA proposed in Chen et al. (2000). In the (s,S) inventory problem, TS guided by TSOCBA has a better chance to recognize the optimal solution than TS with OCBA. In the complex physician scheduling problem, TS coupled with TSOCBA can reach a more promising region in the search space and obtains a smaller optimality gap than OCBA. Also, TSOCBA is more reliable with lower performance variability. Moreover, as found in many cases, the advantage of TSOCBA over OCBA increases with the level of simulation noise, which implies a greater benefit of using TSOCBA in high-stochasticity settings. Finally, we show that the performance of TS coupled with TSOCBA succeeds the OptQuest®, which is a well-known solver for simulation optimization problems, in the throughput maximization problem and physician scheduling problem.

The allocation policy derived in this research is asymtoptically optimal and hence, its low-budget performance may not be guaranteed. Also, the budget allocated to each iteration is simply set as constant, which may not be the best philosophy. Therefore, it remains a future work to develop a dynamic sampling allocation policy for TS. Recently, machine learning models are used to guide the simulation sampling and successfully improve the probability of correct selection in ranking & selection problem (Goodwin et al., 2022). We will study the potential of using machine learning predictions to enhance the tabu search performance in simulation optimization.

**Acknowledgements**

We thank Chun Cheng, Jiaqi Liang, and Xiangyi Zhang for helpful discussions. We thank the referees for comments that improve the manuscript. This work is supported by IVADO and the Canada First Research Excellence Fund, and partially by Shanghai Pujiang Program (21PJ1413300).

**References**


Avriel, M., 2003. Nonlinear programming: analysis and methods. Courier Corporation.

Bechhofer, R.E., 1954. A single-sample multiple decision procedure for ranking means of normal populations with known variances. The Annals of Mathematical Statistics , 16–39.

Bechhofer, R.G., 1995. Design and analysis of experiment for statistical selection, screening, and multiple comparisons. 04; QA279, B4.

Branke, J., Chick, S.E., Schmidt, C., 2007. Selecting a selection procedure. Management Science 53, 1916–1932.

Chen, C.H., 1996. A lower bound for the correct subset-selection probability and its application to discrete-event system simulations. IEEE Transactions on Automatic Control 41, 1227–1231.

Chen, C.H., He, D., Fu, M., 2006. Efficient dynamic simulation allocation in ordinal optimization. IEEE Transactions on Automatic Control 51, 2005–2009.

Chen, C.H., He, D., Fu, M., Lee, L.H., 2008. Efficient simulation budget allocation for selecting an optimal subset. INFORMS Journal on Computing 20, 579–595.

Chen, C.h., Lee, L.H., 2011. Stochastic simulation optimization: an optimal computing budget allocation. volume 1. World Scientific.

Chen, C.H., Lin, J., Yücesan, E., Chick, S.E., 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. Discrete Event Dynamic Systems 10, 251–270.

Chen, C.H., Yücesan, E., Dai, L., Chen, H.C., 2009. Optimal budget allocation for discrete-event simulation experiments. IIE Transactions 42, 60–70.

Chen, Y., Ryzhov, I.O., 2019a. Balancing optimal large deviations in ranking and selection, in: 2019 Winter Simulation Conference (WSC), IEEE. pp. 3368–3379.

Chen, Y., Ryzhov, I.O., 2019b. Complete expected improvement converges to an optimal budget allocation. Advances in Applied Probability 51, 209–235.

Chew, E.P., Lee, L.H., Teng, S., Koh, C.H., 2009. Differentiated service inventory optimization using nested partitions and mocba. Computers & Operations Research 36, 1703–1710.

Chick, S.E., Branke, J., Schmidt, C., 2010. Sequential sampling to myopically maximize the expected value of information. INFORMS Journal on Computing 22, 71–80.

Chick, S.E., Inoue, K., 2001. New two-stage and sequential procedures for selecting the best simulated system. Operations Research 49, 732–743.

Costa, A., Alfieri, A., Matta, A., Fichera, S., 2015. A parallel tabu search for solving the primal buffer allocation problem in serial production systems. Computers & Operations Research 64, 97–112.

Dembo, A., Zeitouni, O., 2010. Large deviation techniques and Applications. volume 38. Springer-Verlag Berlin Heidelberg.

Dengiz, B., Alabas, C., 2000. Simulation optimization using tabu search, in: 2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165), IEEE. pp. 805–810.

Ferland, J.A., Ichoua, S., Lavoie, A., Gagné, E., 2001. Scheduling using tabu search methods with intensification and diversification. Computers & Operations Research 28, 1075–1092.

Frazier, P.I., Powell, W.B., Dayanik, S., 2008. A knowledge-gradient policy for sequential information collection. SIAM Journal on Control and Optimization 47, 2410–2439.

Gendreau, M., 2002. Recent advances in tabu search, in: Essays and surveys in metaheuristics. Springer, pp. 369–377.

Gendreau, M., Laporte, G., Séguin, R., 1996. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. Operations Research 44, 469–477.

Gendreau, M., Potvin, J.Y., et al., 2010. Handbook of metaheuristics. volume 2. Springer.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13, 533–549.

Glynn, P., Juneja, S., 2004. A large deviations perspective on ordinal optimization, in: Proceedings of the 36th conference on Winter simulation, Winter Simulation Conference. pp. 577–585.

Goldsman, D., Kim, S.H., Marshall, W.S., Nelson, B.L., 2002. Ranking and selection for steady-state simulation: Procedures and perspectives. INFORMS Journal on Computing 14, 2–19.

Goodwin, T., Xu, J., Celik, N., Chen, C.H., 2022. Real-time digital twin-based optimization with predictive simulation learning. Journal of Simulation , 1–18.

He, D., Lee, L.H., Chen, C.H., Fu, M.C., Wasserkrug, S., 2010. Simulation optimization using the cross-entropy method with optimal computing budget allocation. ACM Transactions on Modeling and Computer Simulation (TOMACS) 20, 1–22.

Horng, S.C., Yang, F.Y., Lin, S.S., 2012. Applying pso and ocba to minimize the overkills and re-probes in wafer probe testing. IEEE Transactions on Semiconductor Manufacturing 25, 531–540.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. Journal of Global Optimization 13, 455–492.

Juan, A.A., Faulin, J., Grasman, S.E., Rabe, M., Figueira, G., 2015. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Operations Research Perspectives 2, 62–72.

Kim, S.H., Nelson, B.L., 2001. A fully sequential procedure for indifference-zone selection in simulation. ACM Transactions on Modeling and Computer Simulation (TOMACS) 11, 251–273.

Kim, S.H., Nelson, B.L., 2006. Selecting the best system. Handbooks in Operations Research and Management Science 13, 501–534.

Koenig, L.W., Law, A.M., 1985. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. Communications in Statistics-Simulation and Computation 14, 719–734.

Konak, A., Kulturel-Konak, S., 2005. Simulation optimization using tabu search: an empirical study, in: Proceedings of the Winter Simulation Conference, 2005., IEEE. pp. 7–pp.

Law, A.M., Kelton, W.D., 2000. Simulation modeling and analysis. volume 3. McGraw-Hill New York.

Lee, L.H., Chew, E.P., Teng, S., Chen, Y., 2008. Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. European Journal of Operational Research 189, 476–491.

Lee, L.H., Chew, E.P., Teng, S., Goldsman, D., 2010. Finding the non-dominated pareto set for multi-objective simulation models. IIE Transactions 42, 656–674.

Li, J., Liu, W., Pedrielli, G., Lee, L.H., Chew, E.P., 2017. Optimal computing budget allocation to select the nondominated systems—a large deviations perspective. IEEE Transactions on Automatic Control 63, 2913–2927.

Luo, J., Hong, L.J., Nelson, B.L., Wu, Y., 2015. Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. Operations Research 63, 1177–1194.

Lutz, C.M., Davis, K.R., Sun, M., 1998. Determining buffer location and size in production lines using tabu search. European Journal of Operational Research 106, 301–316.

Nelson, B.L., Swann, J., Goldsman, D., Song, W., 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. Operations Research 49, 950–963.

Ni, E.C., Ciocan, D.F., Henderson, S.G., Hunter, S.R., 2017. Efficient ranking and selection in parallel computing environments. Operations Research 65, 821–836.

Niroumandrad, N., Lahrichi, N., 2018. A stochastic tabu search algorithm to align physician schedule with patient flow. Health Care Management Science 21, 244–258.

Pan, H., Wang, L., Liu, B., 2006. Particle swarm optimization for function optimization in noisy environment. Applied Mathematics and Computation 181, 908–919.

Pasupathy, R., Henderson, S.G., 2011. Simopt: A library of simulation optimization problems, in: Proceedings of the 2011 Winter Simulation Conference (WSC), IEEE. pp. 4075–4085.

Peng, Y., Chen, C.H., Chong, E.K., Fu, M.C., 2018a. A review of static and dynamic optimization for ranking and selection, in: 2018 Winter Simulation Conference (WSC), IEEE. pp. 1909–1920.

Peng, Y., Chong, E.K., Chen, C.H., Fu, M.C., 2018b. Ranking and selection as stochastic control. IEEE Transactions on Automatic Control 63, 2359–2373.

Qiu, M., Fu, Z., Eglese, R., Tang, Q., 2018. A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups. Computers & Operations Research 100, 102–116.

Rinott, Y., 1978. On two-stage selection procedures and related probability-inequalities. Communications in Statistics-Theory and methods 7, 799–811.

Ryzhov, I.O., 2016. On the convergence rates of expected improvement methods. Operations Research 64, 1515–1528.

Schmidt, C., Branke, J., Chick, S.E., 2006. Integrating techniques from statistical ranking into evolutionary algorithms, in: Workshops on Applications of Evolutionary Computation, Springer. pp. 752–763.

Shylo, O.V., Shams, H., 2018. Boosting binary optimization via binary classification: A case study of job shop scheduling. arXiv preprint arXiv:1808.10813 .

Tsai, Y.W., Gemmill, D.D., 1998. Using tabu search to schedule activities of stochastic resource-constrained projects. European Journal of Operational Research 111, 129–141.

Xu, J., Huang, E., Chen, C.H., Lee, L.H., 2015. Simulation optimization: A review and exploration in the new era of cloud computing and big data. Asia-Pacific Journal of Operational Research 32, 1550019.

Xu, J., Huang, E., Hsieh, L., Lee, L.H., Jia, Q.S., Chen, C.H., 2016. Simulation optimization in the era of industrial 4.0 and the industrial internet. Journal of Simulation 10, 310–320.

Yang, T., Kuo, Y., Chang, I., 2004. Tabu-search simulation optimization approach for flow-shop scheduling with multiple processors: a case study. International Journal of Production Research 42, 4015–4030.

Zhang, S., Xu, J., Lee, L.H., Chew, E.P., Wong, W.P., Chen, C.H., 2016. Optimal computing budget allocation for particle swarm optimization in stochastic optimization. IEEE Transactions on Evolutionary Computation 21, 206–219.

Zheng, Y.S., Federgruen, A., 1991. Finding optimal (s, s) policies is about as simple as evaluating a single policy. Operations Research 39, 654–665.

## Appendix A. Proof of Lemma 1

Let $P^*$ be the maximum probability of the wrong move events, namely,

$$P^* = \max\{\max_{i\in I\setminus b} P(\bar{Y}(x_i;n_i) \le \bar{Y}(x_b;n_b)), \max_{j\in J} P(\bar{Y}(x_j;n_j) \le \bar{Y}(x_0;n_0)), P(\bar{Y}(x_b;n_b) \le \bar{Y}(x_0;n_0))\},$$

it can be seen that PFM $\ge P^*$. The upper bound of PFM is obtained by the Bonferroni inequality:

$$\begin{aligned}
\text{PFM} &\le \sum_{i\in I\setminus b} P(\bar{Y}(x_i;n_i) \le \bar{Y}(x_b;n_b)) + \sum_{j\in J} P(\bar{Y}(x_j;n_j) \le \bar{Y}(x_0;n_0)) + P(\bar{Y}(x_b;n_b) \le \bar{Y}(x_0;n_0)) \\
&\le (|I|-1)P^* + |J|P^* + P^* \\
&= kP^*.
\end{aligned}$$

Lemma 1 is proved.

## Appendix B. Proof of Lemma 2

The proof is based on Slater's condition (Avriel, 2003). It indicates that for a convex problem

$$\begin{aligned}
&\max_x y(x), \\
\text{s.t.}\quad &\boldsymbol{\omega}(x) \le \mathbf{0}, \\
&\boldsymbol{h}(x) = \mathbf{0},
\end{aligned}$$

where $\boldsymbol{\omega}(x) = (\omega_1(x), \omega_2(x), \ldots)^T$, $\boldsymbol{h}(x) = (h_1(x), h_2(x), \ldots)^T$ and $\mathbf{0} = (0, \ldots)^T$, the CQ is valid if there exists a point $x^\dagger$ satisfying $\boldsymbol{\omega}(x^\dagger) < \mathbf{0}$ and $\boldsymbol{h}(x^\dagger) = \mathbf{0}$. Firstly, we show that Model (10) is convex. This can be easily seen because: (1) the inequality constraint functions $\omega_{ib}(z, \alpha_0, \ldots, \alpha_k) = z - G_{ib}, \forall i \in I \setminus b$, $\omega_{j0}(z, \alpha_0, \ldots, \alpha_k) = z - G_{j0}, \forall j \in J$, and $\omega_{b0}(z, \alpha_0, \ldots, \alpha_k) = z - G_{b0}$ are convex due to the rate functions $G$'s are concave; (2) the equality constraint $h(z, \alpha_0, \ldots, \alpha_k) = 1 - \sum_{i\in I\setminus b} \alpha_i - \sum_{j\in J} \alpha_j - \alpha_b - \alpha_0$ is affine; (3) the objective function $z$ is linear. Secondly, we show that there exists a feasible point satisfying Slater's condition. Given a set of positive values $\{\alpha_0, \ldots, \alpha_k : \alpha_0 + \ldots + \alpha_k = 1\}$, the condition $h(z, \alpha_0, \ldots, \alpha_k) = 0$ holds regardless the value of $z$. Due to the concavity and positivity of $G$'s, one can always find $G^* = \min_{i\in I\setminus b, j\in J}\{G_{ib}, G_{j0}, G_{b0}\} > 0$. Then, for a point $x^\dagger = (z^*, \alpha_0, \ldots, \alpha_k)$ such that $0 < z^* < G^*$, we have $\omega_{ib}(x^\dagger) = z^* - G_{ib} \le z^* - G^* < 0, \forall i \in I \setminus b$, $\omega_{j0}(x^\dagger) = z^* - G_{j0} < 0, \forall j \in J$ and $\omega_{b0}(x^\dagger) = z^* - G_{b0} < 0$. In summary, because $\boldsymbol{\omega}(x^\dagger) < 0$ and $\boldsymbol{h}(x^\dagger) = 0$, the point $x^\dagger$ satisfies the Slater's condition. Lemma 2 is proved.

## Appendix C. Proof of Theorem 1

Theorem 1 can be obtained by solving Model (10) via the KKT conditions. For simplicity, we use $G_{ij}$ to denote the rate function $G_{ij}(\alpha_i, \alpha_j)$ in the following context. Let $L$ be the Lagrangian function of Model (10), we have

$$\begin{aligned} L =& z - \sum_{i \in I \setminus b} \lambda_i (z - G_{ib}) - \sum_{j \in J} \lambda_j (z - G_{j0}) - \lambda_b (z - G_{b0}) - \\ & v(\alpha_0 + \alpha_b + \sum_{i \in I \setminus b} \alpha_i + \sum_{j \in J} \alpha_j - 1) - \sum_{i \in I \setminus b} r_i(-\alpha_i) - \sum_{j \in J} r_j(-\alpha_j) - r_b(-\alpha_b) - r_0(-\alpha_0), \end{aligned}$$

where $\lambda$, $v$ and $r$ are Lagrangian dual variables. The KKT conditions are as follows:

(1) The primal feasibility:

$$z - G_{ib} \leq 0, \forall i \in I \setminus b, \tag{C.1a}$$

$$z - G_{j0} \leq 0, \forall j \in J, \tag{C.1b}$$

$$z - G_{b0} \leq 0, \tag{C.1c}$$

$$\sum_{i \in I \setminus b} \alpha_i + \sum_{j \in J} \alpha_j + \alpha_b + \alpha_0 = 1, \tag{C.1d}$$

$$\alpha_i, \alpha_j, \alpha_b, \alpha_0, z \geq 0, \forall i \in I \setminus b, j \in J. \tag{C.1e}$$

(2) The dual feasibility:

$$\lambda_i, \lambda_j, \lambda_b, r_i, r_j, r_b, r_0, v \geq 0, \forall i \in I \setminus b, j \in J. \tag{C.2}$$

(3) Complementary slackness:

$$\lambda_i (z - G_{ib}) = 0, \forall i \in I \setminus b, \tag{C.3a}$$

$$\lambda_j (z - G_{j0}) = 0, \forall j \in J, \tag{C.3b}$$

$$\lambda_b (z - G_{b0}) = 0, \tag{C.3c}$$

$$r_i \alpha_i = 0, \forall i \in I \setminus b, \tag{C.3d}$$

$$r_j \alpha_j = 0, \forall j \in J, \tag{C.3e}$$

$$r_b \alpha_b = 0, \tag{C.3f}$$

$$r_0 \alpha_0 = 0. \tag{C.3g}$$

(4) Lagrangian stationary:

$$\frac{\partial L}{\partial z} = 1 - \sum_{i \in I \setminus b} \lambda_i - \sum_{j \in J} \lambda_j - \lambda_b = 0,, \tag{C.4a}$$

$$\frac{\partial L}{\partial \alpha_i} = \lambda_i \frac{\partial G_{ib}}{\partial \alpha_i} - v + r_i = 0, \quad \forall i \in I \setminus b, \tag{C.4b}$$

$$\frac{\partial L}{\partial \alpha_j} = \lambda_j \frac{\partial G_{j0}}{\partial \alpha_j} - v + r_j = 0, \quad \forall j \in J, \tag{C.4c}$$

$$\frac{\partial L}{\partial \alpha_b} = \sum_{i \in I \setminus b} \lambda_i \frac{\partial G_{ib}}{\partial \alpha_b} + \lambda_b \frac{\partial G_{b0}}{\partial \alpha_b} - v + r_b = 0, \tag{C.4d}$$

$$\frac{\partial L}{\partial \alpha_0} = \sum_{j \in J} \lambda_j \frac{\partial G_{j0}}{\partial \alpha_0} + \lambda_b \frac{\partial G_{b0}}{\partial \alpha_0} - v + r_0 = 0. \tag{C.4e}$$

The KKT system contains $k+2$ decision variables $z, \alpha_0, \alpha_1, \ldots, \alpha_k$ and $2k+2$ dual variables $\lambda$'s, $r$'s and $v$, as well as $3k+4$ equations. In the following, we try to obtain the relationship between the sampling ratios $\alpha_0, \ldots, \alpha_k$ by eliminating the dual variables through substitutions.

Firstly, the expressions of dual variables are obtained. Given that the sampling ratios are positive, i.e., $\alpha_i, \alpha_j, \alpha_b, \alpha_0 > 0$, from the complementary slackness (Eq.(C.3d)-(C.3g)) we have that $r_i, r_j, r_b, r_0 = 0, \forall i \in I, j \in J$. Then, the expressions of $\lambda$'s are derived as follows.

From Eq.(C.4b), we have:

$$\lambda_i = \frac{v}{\frac{\partial G_{ib}}{\partial \alpha_i}}, i \in I \setminus b.$$

And from Eq.(C.4c):

$$\lambda_j = \frac{v}{\frac{\partial G_{j0}}{\partial \alpha_j}}, j \in J.$$

Substitute $\lambda_i$ in Eq.(C.4d) we get:

$$\begin{aligned} \sum_{i \in I \setminus b} \frac{v}{\frac{\partial G_{ib}}{\partial \alpha_i}} \frac{\partial G_{ib}}{\partial \alpha_b} + \lambda_b \frac{\partial G_{b0}}{\partial \alpha_b} - v = 0, \\ \lambda_b = v[1 - \sum_{i \in I \setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i}] \frac{1}{\partial G_{b0}/\partial \alpha_b}. \end{aligned} \tag{C.5}$$

The expressions of $\lambda$'s are related with the dual variable $v$. If $v = 0$, then all $\lambda$'s would be 0. However, this is not possible, because from Eq.(C.4a) it is obvious that there must be some $\lambda' s > 0$. So we have $v > 0$. Note that $\partial G_{ib}/\partial \alpha_i > 0$ and $\partial G_{j0}/\partial \alpha_j > 0$ (Glynn and Juneja, 2004), it can be sure that $\lambda_i, \lambda_j > 0, \forall i \in I \setminus b, j \in J$. Whilst for $\lambda_b$, its value depends on $[1 - \sum_{i \in I \setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i}]$ , which can be equal or greater than 0. We discuss both cases as below.

**Case** $\lambda_b > 0$: Substitute $\lambda_b$ and $\lambda_j$ into Eq.(C.4e), we have

$$\sum_{i \in J} \frac{v}{\frac{\partial G_{j0}}{\partial \alpha_j}} \frac{\partial G_{j0}}{\partial \alpha_0} + v[1 - \sum_{i \in I \setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i}] \frac{\partial G_{b0}/\partial \alpha_0}{\partial G_{b0}/\partial \alpha_b} - v = 0.$$

After some algebra we get the following condition, denoted by $C_1$, as

$$\sum_{j \in J} \frac{\partial G_{j0}/\partial \alpha_0}{\partial G_{j0}/\partial \alpha_j} + (1 - \sum_{i \in I \setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i}) \frac{\partial G_{b0}/\partial \alpha_0}{\partial G_{b0}/\partial \alpha_b} = 1. \tag{C.6}$$

Since all $\lambda' s > 0$, from the complementary slackness (Eq.(C.3)) we get the condition $C_2$:

$$\begin{aligned} z - G_{ib} = z - G_{j0} = z - G_{b0} = 0, \quad \forall i \in I \setminus b, j \in J, \\ G_{ib} = G_{j0} = G_{b0}, \quad \forall i \in I \setminus b, j \in J. \end{aligned} \tag{C.7}$$

Given that $\lambda_b > 0$, $v > 0$ and $\partial G_{b0}/\partial \alpha_b > 0$, from Eq.(C.5) the condition $C_3$ is obtained:

$$1 - \sum_{i \in I \setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i} > 0. \tag{C.8}$$

**Case** $\lambda_b = 0$: Similarly, by substituting $\lambda_b$ and $\lambda_j$'s into Eq.(C.4e), we have

$$\sum_{i \in J} \frac{v}{\frac{\partial G_{j0}}{\partial \alpha_j}} \frac{\partial G_{j0}}{\partial \alpha_0} - v = 0.$$

After some algebra we obtain the condition $C_4$:

$$1 - \sum_{j \in J} \frac{\partial G_{j0}/\partial \alpha_0}{\partial G_{j0}/\partial \alpha_j} = 0. \tag{C.9}$$

Since $\lambda_i > 0, \forall i \in I \setminus b$ and $\lambda_j > 0, \forall j \in J$, from Eq.(C.3) we have

$$\begin{aligned} z - G_{ib} = z - G_{j0}, \quad \forall i \in I \setminus b, j \in J, \\ G_{ib} = G_{j0}, \quad \forall i \in I \setminus b, j \in J. \end{aligned} \tag{C.10}$$

As $\lambda_b = 0$, from Eq.(C.3) we know that the constraint $z - G_{b0} \leq 0$ is not forced. Assume this constraint is inactive, i.e., $z - G_{b0} < 0$. Since $z - G_{ib} = 0$ and $z - G_{j0} = 0$, we get that

$$G_{b0} > G_{ib} = G_{j0} = z. \tag{C.11}$$

Because the rate functions $G_{ib}$, $G_{j0}$ and $G_{b0}$ are concave and strictly increasing with $\alpha$'s, for any budget allocation $\boldsymbol{\alpha} = (\alpha_0, \dots, \alpha_k)$ satisfying Eq.(C.11), there must exist an $\boldsymbol{\alpha}' = (\alpha'_0, \dots, \alpha'_k)$ rendering that $G_{b0}(\alpha'_b, \alpha'_0) = G_{ib}(\alpha'_i, \alpha'_b) = G_{j0}(\alpha'_j, \alpha'_0) = z'$, and $z' > z$. More specifically, this can be done by decreasing $\alpha_0$ or $\alpha_b$ meanwhile improving some $\alpha_i$'s or $\alpha_j$'s. This implies that Eq.(C.11) is not an optimal condition. Therefore, when $\lambda_b = 0$, if an allocation $\boldsymbol{\alpha}$ is optimal, the constraint $z - G_{b0} \le 0$ has to be active, i.e., $z - G_{b0} = 0$. Thus we get the condition $C_5$:

$$G_{b0} = G_{ib} = G_{j0}, \quad \forall i \in I \setminus b, j \in J. \tag{C.12}$$

Since $v > 0$, $\partial G_{b0}/\partial \alpha_b > 0$ and $\lambda_b = 0$, from Eq.(C.5) we obtain the condition $C_6$:

$$1 - \sum_{i \in I \setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i} = 0 \tag{C.13}$$

Now we summarize the optimality conditions from both the cases of $\lambda_b > 0$ and $\lambda_b = 0$, and obtain a general condition set $\mathbb{C}$. It can be easy to see that $C_2 = C_5$, $C_6 = C_1 \cap C_4$, and $C_4 = C_1 \cap C_6$. Then,

$$\begin{aligned}
\mathbb{C} &= \{C_1 \cap C_2 \cap C_3\} \cup \{C_4 \cap C_5 \cap C_6\} \\
&= \{C_1 \cup (C_4 \cap C_5 \cap C_6)\} \cup \{(C_2 \cap C_3) \cup (C_4 \cap C_5 \cap C_6)\} \\
&= \{(C_1 \cup (C_4 \cap C_6)) \cap (C_1 \cup C_5)\} \cap \{(C_2 \cup (C_4 \cap C_5 \cap C_6)) \cap (C_3 \cup (C_4 \cap C_5 \cap C_6))\}.
\end{aligned}$$

Given that $C_6 = C_1 \cap C_4$, we have $C_4 \cap C_6 = C_4 \cap C_1 \cap C_4 = C_1 \cap C_4$; also due to $C_2 = C_5$, we have $C_2 \cup (C_4 \cap C_5 \cap C_6) = C_2$, then

$$\begin{aligned}
\mathbb{C} &= \{C_1 \cap (C_1 \cup C_5)\} \cap \{C_2 \cap ((C_3 \cup C_6) \cap (C_3 \cup (C_4 \cap C_5)))\} \\
&= C_1 \cap C_2 \cap (C_3 \cup C_6) \cap (C_3 \cup (C_4 \cap C_5)).
\end{aligned}$$

Since

$$\begin{aligned}
C_1 \cap (C_3 \cup (C_4 \cap C_5)) &= (C_1 \cap C_3) \cup (C_1 \cap C_4 \cap C_5) \\
&= (C_1 \cap C_3) \cup (C_1 \cap C_6 \cap C_2) \\
&= C_1 \cap (C_3 \cup (C_6 \cap C_2)) \\
&= C_1 \cap (C_3 \cup C_6) \cap (C_2 \cup C_3),
\end{aligned}$$

then

$$\begin{aligned}\mathbb{C} &= C_1 \cap (C_3 \cup C_6) \cap (C_2 \cup C_3) \cap C_2 \cap (C_3 \cup C_6)\\ &= C_1 \cap C_2 \cap (C_3 \cup C_6).\end{aligned}$$

As a result, the optimality conditions indicated by $\mathbb{C}$ are:

$$\begin{aligned}\sum_{j\in J} \frac{\partial G_{j0}/\partial \alpha_0}{\partial G_{j0}/\partial \alpha_j} + (1 - \sum_{i\in I\setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i}) \frac{\partial G_{b0}/\partial \alpha_0}{\partial G_{b0}/\partial \alpha_b} &= 1,\\ G_{ib} = G_{j0} = G_{b0}, \quad \forall i \in I \setminus b, j \in J,&\\ 1 - \sum_{i\in I\setminus b} \frac{\partial G_{ib}/\partial \alpha_b}{\partial G_{ib}/\partial \alpha_i} &\geq 0.\end{aligned} \tag{C.14}$$

After adding the constraint of total budget fraction, Theorem 1 is obtained.

## Appendix D. Proof of Lemma 3

Under the normality assumption, for each solution $x_i$ we have

$$\mathscr{I}_i(\tau) = \frac{1}{2}(\frac{y(x_i) - \tau}{\sigma_i})^2$$

(see Chapter 2 of Dembo and Zeitouni (2010)). Define $\Phi(\tau) = \alpha_i \mathscr{I}_i(\tau) + \alpha_j \mathscr{I}_j(\tau)$. Since $\mathscr{I}_i(\tau)$ is strictly convex on $\tau$, $\Phi(\tau)$ is also convex. Hence, the stationary point $\tau^*$ is the global minimum. Through $\Phi'(\tau) = 0$ we obtain

$$\tau^* = \frac{\alpha_i/\sigma_i^2}{\alpha_i/\sigma_i^2 + \alpha_j/\sigma_j^2} y(x_i) + \frac{\alpha_j/\sigma_j^2}{\alpha_i/\sigma_i^2 + \alpha_j/\sigma_j^2} y(x_j).$$

Then, having $G_{ij}(\alpha_i, \alpha_j) = \inf_{\tau}(\Phi(\tau)) = \Phi(\tau^*)$, we can obtain Lemma 3.

## Appendix E. Proof of Proposition1

Proposition1 is obtained by solving the equation system in Theorem 1. Under the normality assumption, the following closed-form formulas can be obtained according to Lemma 3:

$$G_{ib} = \frac{(y(x_i) - y(x_b))^2}{\sigma_i^2/\alpha_i + \sigma_b^2/\alpha_b}, \quad i \in I \setminus b, \tag{E.1}$$

$$G_{j0} = \frac{(y(x_j) - y(x_0))^2}{\sigma_j^2/\alpha_j + \sigma_0^2/\alpha_0}, \quad j \in J, \tag{E.2}$$

$$G_{b0} = \frac{(y(x_b) - y(x_0))^2}{\sigma_b^2/\alpha_b + \sigma_0^2/\alpha_0}. \tag{E.3}$$

We first derive the expression of $\alpha_b$. Plug the closed-form expressions of rate functions into Eq.(11a), we get

$$\sum_{j\in J}\frac{\sigma_0^2(y(x_0)-y(x_j))^2}{\alpha_0^2(\sigma_0^2/\alpha_0+\sigma_j^2/\alpha_j)^2}\frac{\alpha_j^2(\sigma_0^2/\alpha_0+\sigma_j^2/\alpha_j)^2}{\sigma_j^2(y(x_0)-y(x_j))^2}+[1-\sum_{i\in I\setminus b}\frac{\sigma_b^2(y(x_i)-y(x_b))^2}{\alpha_b^2(\sigma_b^2/\alpha_b+\sigma_i^2/\alpha_i)^2}\frac{\alpha_i^2(\sigma_b^2/\alpha_b+\sigma_i^2/\alpha_i)^2}{\sigma_i^2(y(x_i)-y(x_b))^2}]$$
$$\cdot\frac{\sigma_0^2(y(x_b)-y(x_0))^2}{\alpha_0^2(\sigma_0^2/\alpha_0+\sigma_b^2/\alpha_b)^2}\frac{\alpha_b^2(\sigma_0^2/\alpha_0+\sigma_b^2/\alpha_b)^2}{\sigma_b^2(y(x_b)-y(x_0))^2}=1.$$

After some algebra we have

$$\alpha_b^2/\sigma_b^2+\sum_{j\in J}\alpha_j^2/\sigma_j^2=\alpha_0^2/\sigma_0^2+\sum_{i\in I\setminus b}\alpha_i^2/\sigma_i^2, \tag{E.4}$$

then the expression of $\alpha_b$ is obtained as

$$\alpha_b=\sigma_b\sqrt{\alpha_0^2/\sigma_0^2-\sum_{j\in J}\alpha_j^2/\sigma_j^2+\sum_{i\in I\setminus b}\alpha_i^2/\sigma_i^2}. \tag{E.5}$$

Eq.(E.5) requires that $\alpha_0^2/\sigma_0^2-\sum\limits_{j\in J}\alpha_j^2/\sigma_j^2+\sum\limits_{i\in I\setminus b}\alpha_i^2/\sigma_i^2\geq 0$. This is actually guaranteed by the condition of Eq.(11c) as illustrated below. By substituting the rate function $G_{ib}$ in Eq.(11c) with Eq.(E.1) we have

$$\alpha_b^2/\sigma_b^2\geq\sum_{i\in I\setminus b}\alpha_i^2/\sigma_i^2. \tag{E.6}$$

Then, plug the inequality (E.6) into (E.4) we get

$$\alpha_0^2/\sigma_0^2\geq\sum_{j\in J}\alpha_j^2/\sigma_j^2, \tag{E.7}$$

which means $\alpha_0^2/\sigma_0^2-\sum\limits_{j\in J}\alpha_j^2/\sigma_j^2+\sum\limits_{i\in I\setminus b}\alpha_i^2/\sigma_i^2\geq 0$. Inequalities (E.6) and (E.7) indicate the lower bounds for $\alpha_b$ and $\alpha_0$.

Now, we derive the expression of $\alpha_i$'s. From Eq.(11b) we know that $G_{ib}=G_{i'b},\forall i,i'\in I,i\neq i'$. By substituting the rate functions $G_{ib}$ and $G_{i'b}$ we have:

$$\frac{(y(x_b)-y(x_i))^2}{\sigma_b^2/\alpha_b+\sigma_i^2/\alpha_i}=\frac{(y(x_b)-y(x_{i'}))^2}{\sigma_b^2/\alpha_b+\sigma_{i'}^2/\alpha_{i'}},\forall i,i'\in I\setminus b. \tag{E.8}$$

Under the assumption $\alpha_b>>\alpha_i,\forall i\in I\setminus b$ (see Proposition1), the above formula can be simplified to

$$\frac{(y(x_b)-y(x_i))^2}{\sigma_i^2/\alpha_i}=\frac{(y(x_b)-y(x_{i'}))^2}{\sigma_{i'}^2/\alpha_{i'}}. \tag{E.9}$$

Then, the ratio between $\alpha_i$ and $\alpha_{i'}$ is obtained

$$\frac{\alpha_i}{\alpha_{i'}} = \frac{\sigma_i^2/(y(x_b) - y(x_i))^2}{\sigma_{i'}^2/(y(x_0) - y(x_{i'}))^2}, \forall i, i' \in I \setminus b, i \neq i'. \tag{E.10}$$

Similarly, the expression of $\alpha_j$'s is derived as below. From $G_{j0} = G_{j'0}$ and $\alpha_0 >> \alpha_j, j \in J$ we get

$$\frac{\alpha_j}{\alpha_{j'}} = \frac{\sigma_j^2/(y(x_0) - y(x_j))^2}{\sigma_{j'}^2/(y(x_0) - y(x_{j'}))^2}, \forall j, j' \in I, j \neq j'. \tag{E.11}$$

Additionally, from $G_{ib} = G_{j0}$, $\alpha_b >> \alpha_i$ and $\alpha_0 >> \alpha_j$ we obtain the ratio between $\alpha_i$ and $\alpha_j$ as

$$\frac{\alpha_i}{\alpha_j} = \frac{\sigma_i^2/(y(x_b) - y(x_i))^2}{\sigma_j^2/(y(x_0) - y(x_j))^2}, \forall i \in I \setminus b, j \in J. \tag{E.12}$$

Finally, the expression of $\alpha_0$ is obtained as follows. From $G_{ib} = G_{b0}$ we derive:

$$\frac{(y(x_b) - y(x_i))^2}{\sigma_b^2/\alpha_b + \sigma_i^2/\alpha_i} = \frac{(y(x_b) - y(x_0))^2}{\sigma_b^2/\alpha_b + \sigma_0^2/\alpha_0}. \tag{E.13}$$

According to the assumption $\alpha_b >> \alpha_0$ and $\alpha_b >> \alpha_i$, we have

$$\frac{\alpha_0}{\alpha_i} = \frac{\sigma_0^2/(y(x_b) - y(x_0))^2}{\sigma_i^2/(y(x_i) - y(x_b))^2}, \forall i \in I \setminus b. \tag{E.14}$$

Combining Eq.(E.7) and (E.14) we obtain the expression of $\alpha_0$ as

$$\alpha_0 = \max\{\sigma_0 \sqrt{\sum_{i \in J} \alpha_j^2/\sigma_j^2}, \alpha_i \frac{\sigma_0^2/(y(x_b) - y(x_0))^2}{\sigma_i^2/(y(x_i) - y(x_b))^2}\}, \forall i \in I \setminus b. \tag{E.15}$$

Proposition1 is obtained.

## Appendix F. Calculation of the sampling ratios

The sampling ratios $\alpha_i, i = 0, 1, \ldots, k$ can be obtained by solving a set of linear equations (13a) - (13f). In our implementation, we first set a seed sample ratio as 1, then calculate other sampling ratios by their relationships. Finally, we normalize the sampling ratios to ensure that the sum equals 1. The detailed steps are:

Step1 : Set the seed $\alpha_p = 1$, where $p$ is the first element in the set $I \setminus b$;

Step2 : Calculate $\alpha_i, \forall i \in I, i \neq p, i \neq b$ by Eq.(13a);

Step3 : Calculate $\alpha_j, \forall j \in J$ by Eq.(13c);

Step4 : Calculate $\alpha_0$ and $\alpha_b$ by Eq.(13d) and Eq.(13e), respectively.

Step5 : Normalize the values: $\alpha_i \leftarrow \frac{\alpha_i}{\sum_{i=0}^{k} \alpha_i}, i = 0, 1, \ldots, k$;

## Appendix G. Description of the physician scheduling problem

Physicians are critical resources for the radiotherapy center. The scheduling of physicians has a great impact on the efficiency of the center and the treatment duration of patients. After arriving at the radiotherapy center, a patient will be assigned to a physician based on the cancer specialization and the physician's quota for new patients. Before the patient starts the treatment plan in the linear accelerator, the patient would undergo a pretreatment phase. As shown in Figure G.10, this phase consists of a series of tasks. Four main tasks, i.e., consultation, scan contouring, dosimetry and preparation of a treatment plan, are provided by the assigned physician. To fulfill the needs of patients, the physician resources are organized by a task-based schedule. An example of physician schedule is given in Figure 12 of Niroumandrad and Lahrichi (2018). In the schedule, each day of the physician is divided into two time slots. Each slot is dedicated to a single task. The number of patients that a physician can meet in a time slot is limited. For each physician, each task type is scheduled for at least once in a week. The goal is to generate a physician schedule which repeats weekly for a long-term period. The objective is to minimize the total pretreatment duration and maximize the preference of physicians, with the following objective function:

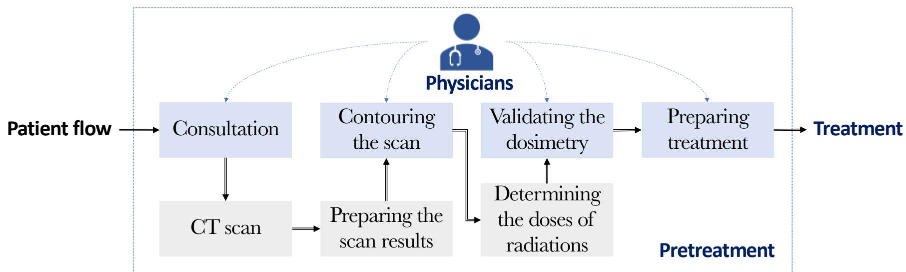



Figure G.10: Patient flow for the pretreatment phase in the radiotherapy center

$$y(x) = w \sum_{j \in \mathbb{J}} \phi_j(x) - (1 - w) \sum_{i \in \mathbb{I}} p_i(x). \tag{G.1}$$

where $w$ is the weight of the pretreatment duration objective, $\phi_j$ is the pretreatment duration of patient $j$, and $p_i$ is the preference score of physician $i$ on the schedule $x$. Given that uncertainties exist in the patient arrival time and type of cancer, a schedule is subjected to randomness in the objective value. This characterizes the problem as a stochastic SO problem. Remind that the objective function (G.1) is composed by a stochastic term, i.e., the pretreatment duration, and a deterministic term of preference index. A greater $w$ means a higher simulation noise level.

In our experiment, we use the real case of Center Intégré de Cancérologie de Laval (CICL). Located in the Montréal region, this cancer treatment center operates four linear accelerators and treats more than 1700 patients each year. There are nine physicians, and each has a quota of nine new patients per week. In practice, besides the four main tasks described above, the physicians will be designated to other tasks such as research, teaching and administration duties. For simplicity, we consider the schedule composing by only the four main tasks. The number of arriving patients is set as 60/week. We consider only curative patients. The patient arrival times are sampled from the Poisson distribution with mean 4, and the patient cancer type is generated based on historical data. The number of patients a physician can treat in a time slot is set as three.

The adopted TS is based on the one proposed in Niroumandrad and Lahrichi (2018) for the stochastic case. Two neighborhood structures, $\mathbb{N}_1(x)$ and $\mathbb{N}_2(x)$, are applied alternatively. $\mathbb{N}_1(x)$ contains all solutions obtained by swapping two tasks of a physician, while $\mathbb{N}_2(x)$ includes the solutions obtained by altering one repeated task of a physician. TS changes from $\mathbb{N}_1(x)$ to $\mathbb{N}_2(x)$ after every 20 consecutive iterations, then switches back to $\mathbb{N}_2(x)$ after 10 iterations. The tabu list contains a set of triplets. For $\mathbb{N}_1(x)$, $(i, d_1, d_2)$ prevents the swapping of the tasks in time slot $d_1$ and $d_2$ of physician $i$; for $\mathbb{N}_2(x)$, $(i, d, t)$ forbids changing physician $i$'s task on time $d$ to task $t$. Each triplet stays in the tabu list for $|\mathbb{I}|+ 2|\mathbb{J}| + |\mathbb{D}| = 99$ iterations, where $\mathbb{I}$, $\mathbb{J}$ and $\mathbb{D}$ are the sets of physicians, patients and time slots in a week, respectively. The solutions visited in the recent fifty iterations are removed from the current neighborhood. The initial solution is randomly generated, and the $t_{max}$ is set as 500.

## Appendix H. ANOVA analysis

This section provides the ANOVA details for the DOEs. The ANOVA tables for the DOEs are given in Figure H.11. The main effects plots are provided in Figure H.12, and the interaction plots are given in Figure H.13. All ANOVA models adopt the main factors and all their interactions as explanatory variables. The box-cox transformation is adopted to eliminate the trend in residuals. For each ANOVA, we test the normality and equal-variance assumptions. However, they do not hold in all the six ANOVA tests. This is mainly due to the range of the response $OptGap_t$ is limited in $[0, +\infty]$. Indeed, with such a truncation in the left tail, it is more difficult to have normally distributed residuals when the fitted level is closed to 0. Also, the fitted levels distant from 0 would have higher variances than those closed to 0. These lead to the poor fitness (indicated by the R-sq(adj) value) in some ANOVA tables, and prevent the further use of conventional post-ANOVA methods, like Tukey test, to compare the levels of factors. Even though, the F-test, i.e, to identify whether the effect of a factor is significant, is still considered to be robust.

**(s,S) Inventory problem**: Table H.11(a) and (b) indicate that for $OptGap_{30}$ and $OptGap_{100}$, the effects of all main factors are significant as the p-values are closed to 0; whilst the interactions are less influential because of smaller F-values. As shown by the main effects plots, increasing the total budget $n$ or the simulation length $l$ improves the performance. The influence of allocation method *Alloc* is also obvious. TSOCBA achieves the best averaged performance for both indicators. The $n - l$ interaction plot shows that, the influence of the simulation noise is more significant when the amount of budget is small. From the $l - Alloc$ interaction plot we observe that, the simulation noise has a greater impact on EA than on OCBA and TSOCBA, which reveals a better robustness of OCBA and TSOCBA to the degree of stochasticity.

**Throughput maximization problem**: According to Table H.11(c) and (d), all main factors are significant for both $OptGap_{20}$ and $OptGap_{100}$. The factor $n$ has the largest influence, followed by $l$ and *Alloc*. For *Alloc*, TSOCBA obtains the best mean for $OptGap_{20}$ and $OptGap_{100}$. The only obvious interaction effect is observed in the $n - Alloc$ plot. Figure H.13 (d) indicates that when the simulation budget is low ($n = 50$), the performance of OCBA is even worse than EA, whilst TSOCBA is still reliable in a low-budget situation.

**Physician scheduling problem**: As shown in Table H.11(e) and (f), for both $OptGap_{100}$ and $OptGap_{500}$, the main factors are significant. The interactions are also significant but less influential. The weight $w$ is the most influential factor, followed by *Alloc* and $n$. TSOCBA achieves the best mean performance for both indicators. From the $w - Alloc$ interaction plots (Figure H.13 (e) and (f)) we see that the advantage of TSOCBA over OCBA increases with the simulation noise level for both $OptGap_{100}$ and $OptGap_{500}$. More specifically, for $OptGap_{100}$, the averaged advantage of TSOCBA over OCBA is -0.0065, 0.5645 and 0.8089 when $w = 0.2$, 0.5 and 0.8, respectively. For $OptGap_{500}$, these values are -0.0915, 0.5324 and 0.8094, respectively. This reveals the benefit of using TSOCBA when the stochasticity is large.

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| n | 2 | 1.1276 | 0.5638 | 500.48 | 0 |
| l | 2 | 23.5408 | 11.7704 | 10448.59 | 0 |
| Alloc | 2 | 2.1064 | 1.0532 | 934.92 | 0 |
| n*l | 4 | 0.4122 | 0.1031 | 91.48 | 0 |
| n*Alloc | 4 | 0.0439 | 0.011 | 9.75 | 0 |
| l*Alloc | 4 | 0.36 | 0.09 | 79.9 | 0 |
| n*l*Alloc | 8 | 0.1299 | 0.0162 | 14.41 | 0 |
| Error | 13473 | 15.1774 | 0.0011 | | |
| Total | 13499 | 42.8982 | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.033563 | 64.62% | 64.55% | 64.48% | | |

(a) (s,S) inventory problem (OptGap_30)

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| n | 2 | 3.179 | 1.5897 | 113.56 | 0 |
| l | 2 | 88.184 | 44.0922 | 3149.65 | 0 |
| Alloc | 2 | 18.345 | 9.1726 | 655.23 | 0 |
| n*l | 4 | 0.084 | 0.0211 | 1.5 | 0.198 |
| n*Alloc | 4 | 0.091 | 0.0227 | 1.62 | 0.166 |
| l*Alloc | 4 | 1.588 | 0.3969 | 28.35 | 0 |
| n*l*Alloc | 8 | 0.147 | 0.0184 | 1.31 | 0.231 |
| Error | 13473 | 188.61 | 0.014 | | |
| Total | 13499 | 300.229 | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.118318 | 37.18% | 37.06% | 36.93% | | |

(b) (s,S) inventory problem (OptGap_100)

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| n | 2 | 42.821 | 21.4105 | 519.17 | 0 |
| l | 2 | 18.191 | 9.0954 | 220.55 | 0 |
| Alloc | 2 | 8.267 | 4.1336 | 100.23 | 0 |
| n*l | 4 | 0.527 | 0.1316 | 3.19 | 0.012 |
| n*Alloc | 4 | 0.869 | 0.2173 | 5.27 | 0 |
| l*Alloc | 4 | 0.022 | 0.0055 | 0.13 | 0.97 |
| n*l*Alloc | 8 | 0.369 | 0.0461 | 1.12 | 0.347 |
| Error | 13473 | 555.626 | 0.0412 | | |
| Total | 13499 | 626.691 | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.203076 | 11.34% | 11.17% | 10.98% | | |

(c) Throughput maximization problem (OptGap_20)

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| n | 2 | 64.62 | 32.3077 | 446.23 | 0 |
| l | 2 | 19.28 | 9.6392 | 133.14 | 0 |
| Alloc | 2 | 14.3 | 7.1506 | 98.76 | 0 |
| n*l | 4 | 0.3 | 0.0738 | 1.02 | 0.395 |
| n*Alloc | 4 | 1.7 | 0.425 | 5.87 | 0 |
| l*Alloc | 4 | 0.12 | 0.0297 | 0.41 | 0.801 |
| n*l*Alloc | 8 | 0.4 | 0.0496 | 0.69 | 0.705 |
| Error | 13473 | 975.46 | 0.0724 | | |
| Total | 13499 | 1076.17 | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.269075 | 9.36% | 9.18% | 8.99% | | |

(d) Throughput maximization problem (OptGap_100)

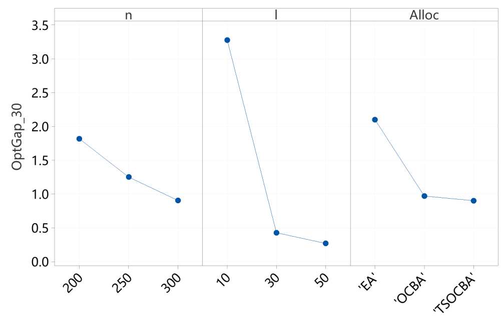
| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| n | 2 | 227.39 | 113.7 | 1020.34 | 0 |
| w | 2 | 6077.06 | 3038.53 | 27268.43 | 0 |
| Alloc | 2 | 668.35 | 334.17 | 2998.94 | 0 |
| n*w | 4 | 54.18 | 13.54 | 121.55 | 0 |
| n*Alloc | 4 | 4.63 | 1.16 | 10.39 | 0 |
| w*Alloc | 4 | 83.92 | 20.98 | 188.28 | 0 |
| n*w*Alloc | 8 | 9.63 | 1.2 | 10.8 | 0 |
| Error | 13473 | 1501.3 | 0.11 | | |
| Total | 13499 | 8626.45 | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.333812 | 82.60% | 82.56% | 82.53% | | |

(e) Physician scheduling problem (OptGap_100)

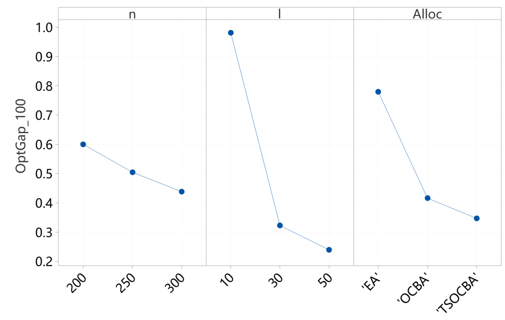
| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| n | 2 | 201.45 | 100.72 | 1268.44 | 0 |
| w | 2 | 4936.11 | 2468.06 | 31081.06 | 0 |
| Alloc | 2 | 591.28 | 295.64 | 3723.09 | 0 |
| n*w | 4 | 44.08 | 11.02 | 138.78 | 0 |
| n*Alloc | 4 | 5.59 | 1.4 | 17.61 | 0 |
| w*Alloc | 4 | 63.39 | 15.85 | 199.59 | 0 |
| n*w*Alloc | 8 | 11.87 | 1.48 | 18.68 | 0 |
| Error | 13473 | 1069.85 | 0.08 | | |
| Total | 13499 | 6923.63 | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.281793 | 84.55% | 84.52% | 84.49% | | |

(f) Physician scheduling problem (OptGap_500)

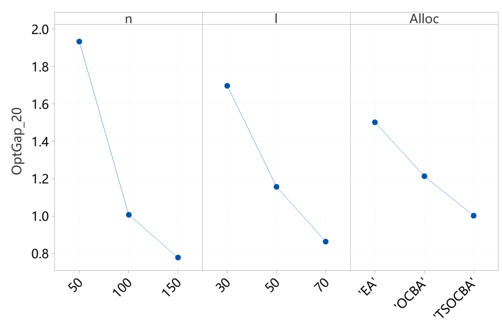
Figure H.11: ANOVA tables



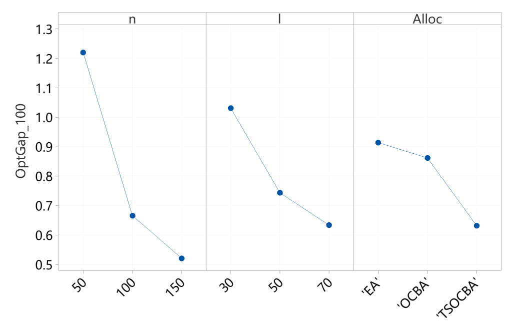

(a) (s,S) inventory problem (OptGap_30)



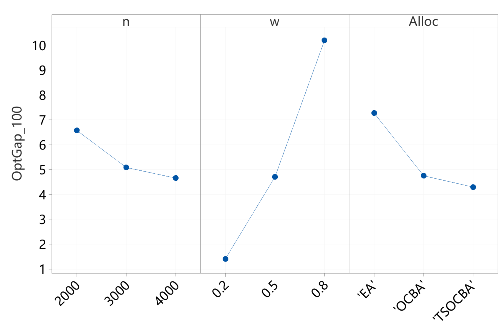

(b) (s,S) inventory problem (OptGap_100)



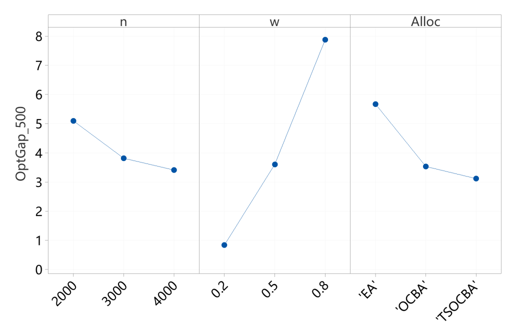

(c) Throughput maximization problem (OptGap_20)




(d) Throughput maximization problem (OptGap_100)

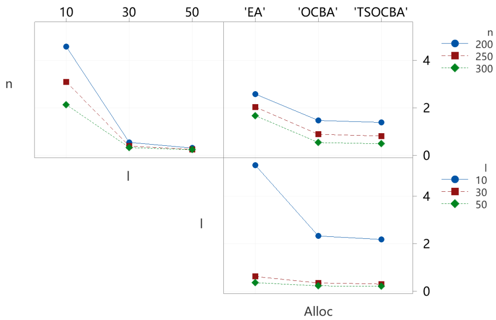



(e) Physician scheduling problem (OptGap_100)

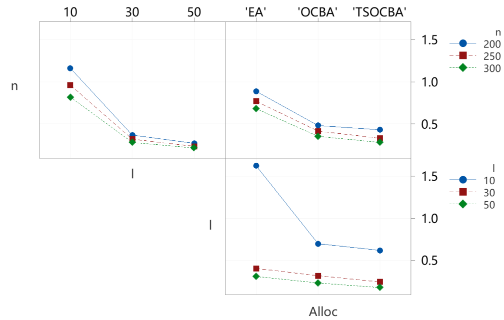



(f) Physician scheduling problem (OptGap_500)

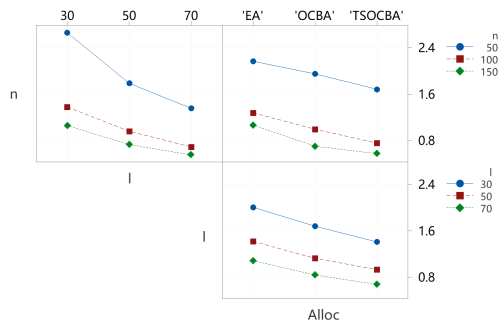
Figure H.12: Main effects plots



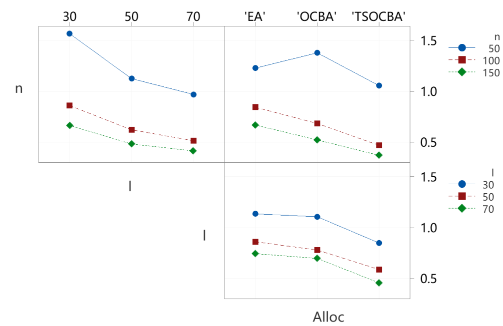

(a) (s,S) inventory problem (OptGap_30)



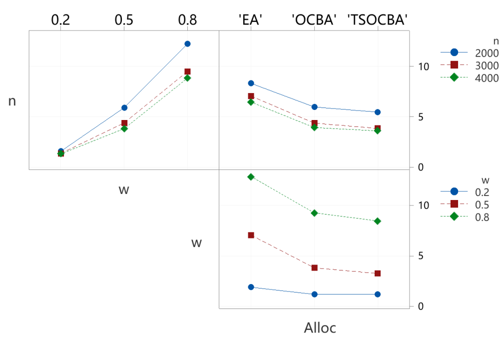

(b) (s,S) inventory problem (OptGap_100)



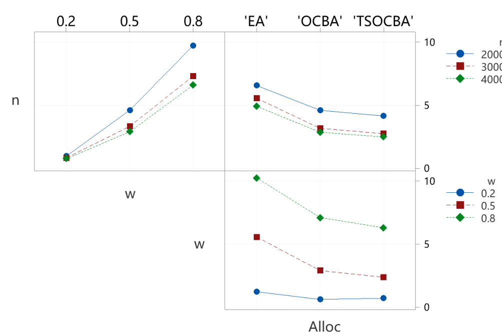

(c) Throughput maximization problem (OptGap_20)




(d) Throughput maximization problem (OptGap_100)




(e) Physician scheduling problem (OptGap_100)




(f) Physician scheduling problem (OptGap_500)

Figure H.13: Interaction plots