Taylor & Francis
Taylor & Francis Group

IISE Transactions

ISSN: 2472-5854 (Print) 2472-5862 (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

A budget allocation strategy minimizing the sample set quantile for initial experimental design

Ziwei Lin, Andrea Matta & Shichang Du

To cite this article: Ziwei Lin, Andrea Matta & Shichang Du (2021) A budget allocation strategy minimizing the sample set quantile for initial experimental design, IISE Transactions, 53:1, 39-57, DOI: 10.1080/24725854.2020.1748771

To link to this article: https://doi.org/10.1080/24725854.2020.1748771

Published online: 18 May 2020.

IISE TRANSACTIONS
2021, VOL. 53, NO. 1, 39–57
https://doi.org/10.1080/24725854.2020.1748771

A budget allocation strategy minimizing the sample set quantile for initial experimental design

Ziwei Lin[a,b], Andrea Matta[b], and Shichang Du[a]

[a]Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai, P.R. China; [b]Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy

ABSTRACT

The increased complexity of manufacturing systems makes the acquisition of the system performance estimate a black-box procedure (e.g., simulation tools). The efficiency of most black-box optimization algorithms is affected significantly by initial designs (populations). In most population initializers, points are spread out to explore the entire domain, e.g., space-filling designs. Some population initializers also consider exploitation procedures to speed up the optimization process. However, they are either application-dependent or require an additional budget. This article proposes a generic method to generate, without an additional budget, several good solutions in the initial design. The aim of the method is to optimize the quantile of the objective function values in the generated sample set. The proposed method is based on a clustering of the solution space; feasible solutions are clustered into groups and the budget is allocated to each group dynamically based on the observed information. The asymptotic performance of the proposed method is analyzed theoretically. The numerical results show that, if proper clustering rules are applied, an unbalanced design is generated in which promising solutions have higher sampling probabilities than non-promising solutions. The numerical results also show that the method is robust to wrong clustering rules.

ARTICLE HISTORY
Received 24 June 2019
Accepted 20 March 2020

KEYWORDS
Budget allocation; quantile minimization; initial designs; optimization

1. Introduction

As the complexity of manufacturing systems increases, the estimation of a system's performance (e.g., throughput and lead time) using a closed-form analytical formula becomes extremely difficult, even impossible. Some tools for complex system performance evaluation have been proposed and have demonstrated their power, such as aggregation methods (Li and Meerkov, 2009), decomposition methods (Gershwin, 1987; Dallery *et al.*, 1988, 1989) and simulation models. In these cases, the acquisition of the system performance estimates is a black-box process.

To find the optimal or near-optimal system configuration, black-box optimization methods allocate the budget (one budget allocation means one acquisition of the objective function value) to numerous, even infinite, feasible solutions. Most black-box optimization algorithms for engineering problems require warm start solutions that, if properly selected, may help to reduce the computational effort and improve the quality of the solution. Having a good initial population can increase the convergence speed of population-based searching algorithms, such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) and Genetic Algorithm (GA) (Srinivas and Patnaik, 1994). Model-based methods, e.g., model reference adaptive search (Hu *et al.*, 2007) and cross-entropy method (Rubinstein, 1999), can have a more accurate initial sampling probability distribution, if the initial design points are properly allocated. In surrogate-based methods, e.g., efficient global optimization (Jones *et al.*, 1998), the quality of the initial surrogate model also depends on the initial design.

Currently, uniform sampling and space-filling designs (e.g., Latin hypercube sampling (McKay *et al.*, 1979), the minimax and maximin distance design (Johnson *et al.*, 1990)) are the most utilized initial designs. In addition, in the field of evolutionary algorithms, several population initialization techniques are used to increase the diversity, or uniformity, of the generated design, such as chaotic maps (dos Santos Coelho and Mariani, 2008; Alatas, 2010), quasi-random sequences (Maaranen *et al.*, 2004), centroidal voronoi tessellation (Richards and Ventura, 2004) and simple sequential inhibition (Maaranen *et al.*, 2007). A review of population initializers can be found in Kazimipour *et al.* (2014). In the field of surrogate-based optimization algorithms, some criterion-based designs are proposed for specific estimators, e.g., the D-optimal design (Smith, 1918) minimizes the confidence ellipse volume of the regression parameters of least-square estimators, whereas the maximum entropy design (Santner *et al.*, 2003) maximizes the entropy difference of Gaussian-process estimators' parameters before and after experiments. All these designs attempt to decentralize initial design points to explore the space, i.e., only exploration is considered.

Indeed, exploring the entire space helps the algorithm to maintain a global perspective. However, the efficiency of the

CONTACT Andrea Matta andrea.matta@polimi.it

optimization algorithms is critical in reality, such as in real-time optimization problems. A good solution, which may not be the optimum, is required in a short time. In this case, initial designs taking into account the exploitation, i.e., having good solutions in the initial population, can help to speed up the optimization process. Some application-specific methods have been proposed to create good solutions in the initial population to improve the efficiency of evolutionary algorithms, these include Pezzella *et al.* (2008) and Zhang *et al.* (2011) for flexible job-shop scheduling problems, and Deng *et al.* (2015) for symmetric traveling salesmen problems. These methods are developed based on the features of the studied problems, which may not be applicable in other problems. Some methods generate an original population randomly and update the population using the objective function as a guideline before it is used as the initial population in evolutionary algorithms. For example, Rahnamayan *et al.* (2007a) apply opposition-based learning (Tizhoosh, 2005) to replace the original points by their opposite points that have better fitness values; instead of opposite points, quasi-opposite points are used in Rahnamayan *et al.* (2007b); de Melo and Delbem (2012) re-sample the initial population, applying a machine-learning technique. In these methods, a pre-optimization process is performed to replace old points with new sampled points with better fitness values. This means non-negligible additional computational effort (i.e., an additional budget in addition to the initial population size) is required, if the acquisition of the objective function is expensive.

The problem investigated in this article is to develop a generic population initializer for applications in deterministic optimization problems, in which the objective function is a black box. A small computational budget is allocated to a broad feasible domain, with the goal of identifying promising alternatives from which the search for the optimum will start. We address the situation in which the acquisition of the objective function is time-consuming. For example, a long simulation length is required to estimate the quantile of the order tardiness, due to the high variability of the system. No extra budget, except for the initial design size, is available. A user-defined number of good solutions are expected to be contained in the generated design. More specifically, we try to optimize the corresponding quantile of the objective function values in the created design.

This article proposes a budget allocation method to create designs based on an existing clustering on the solution space. The feasible solutions are clustered into groups based on specific clustering rules. The budget is allocated to each group dynamically based on the observed group information, i.e., this is a sequential design problem. Different clustering rules can be applied, such as partitioning the feasible domain into several regions evenly or clustering feasible solutions using techniques such as the k-means method (MacQueen, 1967). The clustering can also be executed in transformed spaces. For example, in the problem where the acquisition of the objective function is time-consuming (high-fidelity models) but the objective function can also be roughly and quickly estimated by analytical methods (low-fidelity models), the alternative solutions can be clustered based on their low-fidelity outputs. The knowledge embedded in low-fidelity methods may help to separate promising solutions from bad solutions. A similar idea appears in MO2TOS (Xu *et al.*, 2016), in which the original solution space is transformed into a one-dimensional ordinal space based on a queuing model.

There are two other classic types of budget allocation problems. One is the multi-armed bandit problem, in which the budget needs to be allocated to competing choices. Each selection provides a random reward from an unknown distribution specific to the selected choice. The goal of the budget allocation policy is to maximize the cumulative reward. The policy will face a dilemma: allocating more budget to the current best choice, i.e., the choice having the current highest average payoff, to gain more (exploitation); or allocating more budget to search for the real best choice (exploration). The ε-greedy method (Sutton *et al.*, 1998) selects the current best choice with probability $1-\varepsilon$, while with probability ε, other choices are randomly selected. Wiering (1999) combines the ε-greedy method with SoftMax function assigning weights to choices that are not the current best one. The UCB family (e.g., UCB1, UCB-Tuned (Auer *et al.*, 2002) and KL-UCB (Maillard *et al.*, 2011) and Thompson sampling (Thompson, 1933; Russo *et al.*, 2018) are also frequently used to solve this problem type. A survey on the multi-armed bandit approach can be found in Burtini *et al.* (2015).

Ranking and Selection (R&S) procedures are applied to deal with the other type of budget allocation problems. Budget is allocated to a set of alternative solutions, which have a stochastic performance, to select the best one. The Probability of Correct Selection is commonly constrained. The goal of the budget allocation policy is to separate the real best solution from the others. In classic R&S, the best solution refers to the solution with the lowest (or highest) performance expectation. Optimal Computing Budget Allocation (OCBA), proposed by Chen *et al.* (2000), allocates a simulation budget of a certain size to maximize the approximate probability of correctly selecting the best solution. Some other algorithms to handle this problem type are expected value of information (Chick and Inoue, 2001), knowledge gradient procedure (Frazier *et al.*, 2009) and indifference-zone procedure (Hong and Nelson, 2005). Recently, several approaches have been proposed to deal with this type of problems with different definitions of the best solution. For example, in Linz *et al.* (2016) and Peng *et al.* (2019), the best solution is defined as the one with the optimal quantile.

The goal of the budget allocation method proposed in this article is to minimize the quantile of the sampled values (i.e., the objective function values of the sampled solutions). This is different from the goals of the multi-armed bandit problem and the R&S, which maximize the cumulative sampled values and separate the best group (considering one group is one solution with stochastic performance) from other groups, respectively. The research contributions of this article are summarized below:

1. A generic population initializer is proposed to generate an unbalanced design, in which more budget is

allocated to promising regions. Currently, most initial designs consider only exploration by decentralizing design points. Some existing methods are able to exploit the promising regions, but they are either application-specific or require an additional budget in addition to the initial design size.

2. Closed-form formulas are developed to allocate budget to existing groups of different solutions, in order to minimize the quantile of the sampled solutions' objective function values. Thus, the developed algorithm is easy to implement in practice.
3. The asymptotic performance of the proposed method and its robustness to wrong clustering rules are analyzed theoretically.

The proposed method is tested in designed cases and applied to a transfer line buffer allocation problem and a multi-stage manufacturing system server allocation problem. Numerical results show that, in the studied cases, the proposed method behaves as expected and improves the performance of the applied search algorithms.

The proposed budget allocation method can be used as a population initializer for optimization problems in several fields of engineering. It can also be used as a strategy to allocate budget to competitive choices in other problems aiming at minimizing a certain quantile, or the minimum, of all the obtained values. For instance, budgets with different sizes can be allocated to search different neighborhoods based on the quality of the regions in multi-start algorithms; different numbers of solutions can be sampled from different sub-regions in partition-based search algorithms, e.g., nested partition (Shi and Ólafsson, 2000, 2009).

This article is organized as follows. Section 2 describes the problem in a mathematical way. Section 3 describes the proposed method and analyzes its asymptotic performance. Section 4 applies the proposed method to cases designed for testing purposes. Section 5 presents applications of the proposed method to a transfer line buffer allocation problem and a multi-stage manufacturing system server allocation problem. Finally, conclusion and guidelines for future developments are drawn in Section 6.

2. Notation and problem description

A minimization problem, in which the objective function $y(\cdot)$ is a black box, is considered in this article. The objective function is deterministic, i.e., no noise is involved in the acquisition of the objective function values. The decision variable vector $\boldsymbol{x} = [x_1, \ldots, x_d]^T$ is a d-dimensional vector and $\mathcal{D}$ is the feasible domain. An optimization algorithm (e.g., GA, PSO), in which an initial design is created at the first step, is applied to find the (near) optimal solution.

Assume that all the feasible solutions have been clustered into K groups based on specific clustering rules. A budget of size N is allocated to these K groups applying a budget allocation policy $\mathscr{S} = [n_1, \ldots, n_K]$, where n_k is the total budget size allocated to group k. A corresponding number of solutions are sampled from different groups to create the initial design $\mathcal{S}(n_1, \ldots, n_K, \xi)$, where ξ presents the random noise caused by the sampling. The policy used inside one group to sample new solutions is user-defined. Denote the ith solution sampled from group k as $\boldsymbol{x}_{k,i}$. We introduce the following assumption:

Assumption 1. *The objective function values of solutions sampled from group k, i.e., $y(\boldsymbol{x}_{k,i}), \forall i$, are absolutely continuous random variables, which are independently and identically distributed with probability density function (pdf) $f_k(\cdot)$ and cumulative distribution function (cdf) $F_k(\cdot)$, where $f_k(\cdot)$ is positive and differentiable at any point.*

A user-defined number, denoted as r ($r < N/2$), of good solutions are expected to be contained in the generated sample set $\mathcal{S}(n_1, \ldots, n_K, \xi)$. This means that the objective function value of the rth best solution in the sample set, which is the r/N-quantile of the objective function values in the created sample set, is expected to be minimized. Let $\alpha = r/N$, where $\alpha < 0.5$, and $q_\alpha(\mathcal{S}(n_1, \ldots, n_K, \xi))$ be the α-quantile of the objective function values in the created initial design, the problem investigated in this article can be formulated as follows:

$$\begin{aligned} \min_{n_1,\ldots,n_K} \quad & E(q_\alpha(\mathcal{S}(n_1, \ldots, n_K, \xi))), \\ s.t. & \sum_{k=1}^{K} n_k = N, \\ & n_k \in \mathbb{N}, \forall k, \end{aligned} \tag{1}$$

where

$$q_\alpha(\mathcal{S}(n_1, \ldots, n_K, \xi)) = \min_\tau \left\{ \tau \,\middle|\, \sum_{k=1}^{K} \sum_{i=1}^{n_k} I_{\{\boldsymbol{x}|y(\boldsymbol{x})\le\tau\}}(\boldsymbol{x}_{k,i}) \ge \alpha N; \tau \in \mathbb{R} \right\},$$

and $I_{\{\boldsymbol{x}|y(\boldsymbol{x})\le\tau\}}(\cdot)$ is the indicator function. The objective function of the problem in expression (1) is the expectation of the α-quantile. The first constraint shows the budget constraint. The second constraint indicates that $n_k, \forall k$ are non-negative integers.

For the sake of simplicity, the notations used in the proposed method are listed below.

N	: the total budget size
K	: number of groups clustered
n_k	: the total budget size allocated to group k, $\sum_{k=1}^{K} n_k = N$
$\mathcal{S}(n_1, \ldots, n_K, \xi)$	: the final sample set
α	: the fraction of good solutions in $\mathcal{S}(n_1, \ldots, n_K, \xi)$, $\alpha = r/N$
N_s	: the total allocated budget size after stage s
$n_{s,k}$	: the total allocated budget size in group k after stage s
$\hat{\mu}_k, \hat{\sigma}_k^2$	: the sample mean and sample variance of $y(\cdot)$ in group k
$\hat{b}$	: the current best group
$\hat{\tau}$	: the estimated threshold

3. Budget allocation for quantile minimization

The problem in expression (1) is difficult to solve, due to the calculation of the order statistic expectation of samples from different distributions. Therefore, we approximate the problem to a simpler one. According to the definition of the $q_\alpha(\mathcal{S}(n_1, ..., n_K, \xi))$ in Section 2, $r = \alpha N$ solutions in the final sample set have objective function values smaller than or equal to $q_\alpha(\mathcal{S}(n_1, ..., n_K, \xi))$. Instead of minimizing the expectation of $q_\alpha(\mathcal{S}(n_1, ..., n_K, \xi))$ as in expression (1), we propose an effective approximate formulation, which is minimizing a threshold τ so that expected αN solutions in the final sample set have objective function values less than or equal to τ:

$$\begin{aligned} \min_{\tau, n_1, ..., n_K} \quad & \tau, \\ s.t. \quad & E\left(\sum_{k=1}^{K}\sum_{i=1}^{n_k} I_{\{\boldsymbol{x}|y(\boldsymbol{x})\le\tau\}}(\boldsymbol{x}_{k,i})\right) = \alpha N, \\ & \sum_{k=1}^{K} n_k = N, \\ & \tau \in \mathbb{R}, n_k \in \mathbb{N}, \forall k, \end{aligned} \tag{2}$$

where $I_{\{\boldsymbol{x}|y(\boldsymbol{x})\le\tau\}}(\cdot)$ is the indicator function, $\boldsymbol{x}_{k,i}$ is the ith solution sampled from group k and $y(\boldsymbol{x}_{k,i})$ is the objective function value at $\boldsymbol{x}_{k,i}$. The first constraint limits the expected number of solutions in the final sample set whose objective function values are under the threshold τ. The second constraint shows the budget constraint. The third constraint indicates that τ is real and $n_k, \forall k$ are non-negative integers. The above problem approximation moves the expectation from the objective function to the constraint. In this way, the calculation of the order statistic expectation is avoided, which makes this problem easier to be solved. Under Assumption 1, the approximated problem can be further simplified as shown in Proposition 1 and the proof of Proposition 1 can be found in Appendix A. Numerical results show that the problem in expression (3) is a reasonable approximation of the problem in expression (1) (more details can be found in Appendix B).

Proposition 1. *Under Assumption 1, the problem in expression (2) can be simplified as*

$$\begin{aligned} \min_{\tau, n_1, ..., n_K} \quad & \tau, \\ s.t. \quad & \sum_{k=1}^{K} n_k F_k(\tau) = \alpha N, \\ & \sum_{k=1}^{K} n_k = N, \\ & \tau \in \mathbb{R}, n_k \in \mathbb{N}, \forall k. \end{aligned} \tag{3}$$

In the following, Section 3.1 presents the optimal budget allocation solution of the approximated problem in expression (3) when the group information, i.e., $F_k(\cdot), \forall k$, are known in advance. Section 3.2 proposes a heuristic algorithm to allocate budget to each group dynamically when the group information are unknown, and analyzes the asymptotic performance of the proposed heuristic algorithm as the budget size increases.

3.1. Optimal policy when group information are known

After the clustering, if Assumption 1 holds and we know all the group information in advance, the optimal solution of the approximated problem in expression (3) has a closed form as shown in Theorem 1.

Theorem 1. *Assumption 1 holds, all the group α-quantiles, i.e., $F_k^{-1}(\alpha), \forall k$, are known and $F_i^{-1}(\alpha) \ne F_j^{-1}(\alpha), \forall i \ne j$, where $F_k^{-1}(\cdot)$ is the inverse function of $F_k(\cdot)$, the optimal solution of the approximated problem in expression (3) is*

$$\begin{cases} b = \arg\min_k F_k^{-1}(\alpha) \\ \tau^* = F_b^{-1}(\alpha) \\ n_b^* = N \\ n_k^* = 0, \forall k \ne b \end{cases},$$

i.e., allocating all the budget to the best group b, which has the smallest group α-quantile.

The problem in expression (3) is a mixed-integer problem. To prove Theorem 1, we first relax the integer constraints on n_k. In the relaxed problem, only one solution satisfies the Fritz John conditions (Bazaraa *et al.*, 2013), which is the one provided by Theorem 1. Therefore, if this relaxed problem has a local optimum, it must be this solution. Then, we use the KKT second-order sufficient conditions (Bazaraa *et al.*, 2013) to prove that this solution is a strict local optimum of the relaxed problem, which is also the global optimum since it is the only local optimum. This solution also satisfies the relaxed integer constraints. Thus, it is the global optimal solution of the problem in expression (3).

Proof of Theorem 1. Relax the integer constraints in expression (3) from $n_k \in \mathbb{N}, \forall k$ to $n_k \ge 0, n_k \in \mathbb{R}, \forall k$. The decision variables of the relaxed problem are defined in the real number set. Under Assumption 1, the objective function and the constraints of the relaxed problem are all continuously differentiable at all feasible solutions. Denote the equality constraints as $h_i(\cdot) = 0, i = 1, 2$ and the inequality constraints as $g_k(\cdot) \ge 0, k = 1, ..., K$:

$$\begin{aligned} h_1(\tau, n_1, ..., n_K) &= \sum_{k=1}^{K} n_k F_k(\tau) - \alpha N, \\ h_2(\tau, n_1, ..., n_K) &= \sum_{k=1}^{K} n_k - N, \\ g_k(\tau, n_1, ..., n_K) &= n_k, \forall k. \end{aligned}$$

The gradients of the above equations are

$$\begin{aligned} \nabla h_1(\tau, n_1, ..., n_K) &= \left[\sum_{k=1}^{K} n_k f_k(\tau), F_1(\tau), ..., F_K(\tau)\right]^T, \\ \nabla h_2(\tau, n_1, ..., n_K) &= [0, 1, ..., 1]^T, \\ \nabla g_k(\tau, n_1, ..., n_K) &= \boldsymbol{e}_{k+1}, \forall k, \end{aligned}$$

where $\boldsymbol{e}_{k+1}$ is a $(K+1)$-dimensional vector whose $(k+1)$th element is one and the rest elements are zero.

If $[\tau^*, n_1^*, ..., n_K^*]^T$ is a local optimum of the relaxed problem, it must satisfy the Fritz John conditions (Bazaraa *et al.*, 2013), i.e., there exists a non-zero vector $\boldsymbol{\lambda} = [\lambda_0, \lambda_k, \forall k \in \mathcal{I}_a, \lambda_{K+1}, \lambda_{K+2}]^T$, in which $\mathcal{I}_a = \{k : n_k^* = 0\}$ and $\lambda_k \geq 0, \forall k \in \mathcal{I}_a \cup \{0\}$, such that:

$$\lambda_0 \nabla f(\tau^*, n_1^*, \ldots, n_K^*) - \sum_{k \in \mathcal{I}_a} \lambda_k \nabla g_k(\tau^*, n_1^*, \ldots, n_K^*) - \sum_{i=1,2} \lambda_{K+i} \nabla h_i(\tau^*, n_1^*, \ldots, n_K^*) = \mathbf{0},$$

where $\nabla f(\tau^*, n_1^*, \ldots, n_K^*) = \boldsymbol{e}_1$ is the gradient of the objective function of the relaxed problem.

From the Fritz John conditions and the constraints of the relaxed problem, we have the following equations:

$$\begin{cases} \lambda_0 - \lambda_{K+1}\left(\sum_{k=1}^{K} n_k^* f_k(\tau^*)\right) = 0 & \text{(4a)} \\ \lambda_k + \lambda_{K+1} F_k(\tau^*) + \lambda_{K+2} = 0, \forall k \in \mathcal{I}_a & \text{(4b)} \\ \lambda_{K+1} F_k(\tau^*) + \lambda_{K+2} = 0, \forall k \notin \mathcal{I}_a & \text{(4c)} \\ \sum_{k \notin \mathcal{I}_a} n_k^* F_k(\tau^*) - \alpha N = 0 & \text{(4d)} \\ \sum_{k \notin \mathcal{I}_a} n_k^* - N = 0. & \text{(4e)} \\ n_k^* > 0, \forall k \notin \mathcal{I}_a & \text{(4f)} \\ n_k^* = 0, \forall k \in \mathcal{I}_a & \text{(4g)} \\ \lambda_k \geq 0, \forall k \in \mathcal{I}_a \cup \{0\} & \text{(4h)} \\ \exists k \in \mathcal{I}_a \cup \{0, K+1, K+2\}, s.t. \lambda_k \neq 0 & \text{(4i)} \end{cases}$$

$f_k(\cdot)$ is positive for all feasible solutions in the relaxed problem under Assumption 1. From equations (4a) (4b) (4c) (4h) (4i), we have $\lambda_0 > 0$ (i.e., a regularity condition holds for all feasible solutions, the Fritz John conditions are equivalent to the KKT conditions), $\lambda_{K+1} > 0$ and $\lambda_{K+2} < 0$. From equations (4c) (4d) (4e), we have $F_k(\tau^*) = \alpha, \forall k \notin \mathcal{I}_a$. Due to the uniqueness assumption on the group quantiles, only one element, denoted as b, is not in the set $\mathcal{I}_a$, i.e., $\mathcal{I}_a = \{k | k \neq b\}, n_b^* = N, n_k^* = 0, \forall k \neq b,$ and $\tau^* = F_b^{-1}(\alpha)$. From equations (4b) (4c)(4h) and the uniqueness assumption of group quantiles, we have $\lambda_k > 0, F_k^{-1}(\alpha) > \tau^*, \forall k \in \mathcal{I}_a$. Therefore, we have only one solution that satisfies the Fritz John conditions:

$$\begin{cases} b = \arg\min_k F_k^{-1}(\alpha) \\ \tau^* = F_b^{-1}(\alpha) \\ n_b^* = N \\ n_k^* = 0, \forall k \neq b \end{cases}.$$

This solution is a KKT point and $\lambda_k > 0, \forall k \in \mathcal{I}_a$. The objective function and the constraints of the relaxed problem are all twice differentiable under Assumption 1. To prove this solution satisfies the KKT second-order sufficient conditions (Bazaraa *et al.*, 2013), we define the cone:

$$\mathcal{G} = \left\{ \boldsymbol{d} \left| \begin{array}{l} \boldsymbol{d} \neq \mathbf{0} \\ \nabla g_k(\tau^*, n_1^*, \ldots, n_K^*)^T \boldsymbol{d} = 0, \forall k \neq b \\ \nabla h_i(\tau^*, n_1^*, \ldots, n_K^*)^T \boldsymbol{d} = 0, i = 1, 2 \end{array} \right. \right\}.$$

$\mathcal{G}$ is an empty set, which means this solution satisfies the KKT second-order sufficient conditions and it is the strict and the only local optimum, i.e., the global optimum, of the relaxed problem. In addition, this solution satisfies the relaxed integer constraints, so it is also the global optimal solution of the problem in equation (3). Theorem 1 is proved. □

3.2. Allocation strategy when group information are unknown

In practice, it is unlikely to know in advance the group information before a large number of observations are available. In this situation, the most intuitive approach is to estimate the group quantiles by allocating a trial budget to each group at the first stage. Then, all the remaining budget is allocated to the current best group according to the observed information. Nevertheless, this approach has two drawbacks: (i) numerous samples are required to estimate a quantile, whereas only a small budget is available for the initial design; and (ii) the estimated group information could be biased due to the sampling noise, which means, under a certain probability, this approach will allocate all the remaining budget to a wrong group.

To simplify the estimation of the group quantiles and reduce the sample size required, we further restrict Assumption 1 as follows:

Assumption 2. *The objective function values of solutions sampled from group k, i.e., $y(\boldsymbol{x}_{k,i})$, are independently and identically distributed as normal distribution with mean μ_k and positive variance σ_k^2, in which μ_k and σ_k are unknown.*

Under Assumption 2, the best group in Theorem 1 has a closed-form expression: $b = \arg\min_k\{\mu_k + z_\alpha \sigma_k\}$, where z_α is the α-quantile of the standard normal distribution. In this way, non-parametric quantile estimation can be avoided and the best group can be easily estimated, with few samples, using the group sample means $\hat{\mu}_k$ and group sample variances $\hat{\sigma}_k$:

$$\hat{b} = \arg\min_k\{\hat{\mu}_k + z_\alpha \hat{\sigma}_k\}. \quad (5)$$

Despite Assumption 2 being quite strong, the numerical results show that the proposed budget allocation method works well even if Assumption 2 is not satisfied. From equation (5), we can find that the selection of the α value reflects the preference among the group mean and the group variance. If a low α value is used (i.e., only a few good solutions are required), groups with a high variance are regarded as good groups, i.e., the proposed method will take the risk of searching in the group with not only a high mean but also a high variance, in order to have the chance of obtaining a very good solution. If a high α value is used (i.e., nearly half of the initial solutions are of interest), groups with a low mean are regarded as good groups, i.e., the proposed method will behave conservatively to ensure that most of the sampled solutions are acceptable.

The current best group, estimated from the first stage sampling, may be wrong, due to the small sample size. Therefore, we are facing a dilemma similar to the multi-armed bandit problem: to allocate more budget to the current best group, or to allocate more budget to search for the real best group. The difference between our problem and the multi-armed problem is the definition of the best group. We are interested in the group with the smallest quantile, whereas the multi-armed problem is looking for the group with the highest mean. To cope with this dilemma, a simple idea is proposed: let the total budget size allocated to group k be proportional to the posterior probability that group k is the best group b, given the observed information.

A heuristic algorithm is developed to allocate the remaining budget dynamically taking into account the noise introduced by previous samplings. At a new stage (denoted as s), a fixed size (denoted as Δ) of budget is added into the total budget size:

$$N_s = \min\{N_{s-1} + \Delta, N\},$$

and the estimated group means $\hat{\mu}_k$, the group variances $\hat{\sigma}_k^2$ and the current best group $\hat{b}$ are updated. Under Assumption 2, from Theorem 1 we have $\sigma_k > 0, \forall k$ and $\tau^* = \mu_b + z_\alpha \sigma_b \le \mu_k + z_\alpha \sigma_k, \forall k$, i.e., $\mu_b - \tau^* = -z_\alpha \sigma_b$ and $\mu_k - \tau^* \ge -z_\alpha \sigma_k, \forall k$. Since $\alpha < 0.5$ (i.e., $z_\alpha < 0$), we have:

$$0 < \frac{\sigma_b}{\mu_b - \tau^*} = -\frac{1}{z_\alpha} \quad \text{and} \quad 0 < \frac{\sigma_k}{\mu_k - \tau^*} \le -\frac{1}{z_\alpha}, \forall k,$$

that is, the real best group b has the largest $\frac{\sigma_k}{\mu_k - \tau^*}$. Therefore, we first estimate the value of the optimal threshold τ^*, based on the information estimated from previous sampling, as:

$$\hat{\tau} = \min_k\{\hat{\mu}_k + z_\alpha \hat{\sigma}_k\} = \hat{\mu}_{\hat{b}} + z_\alpha \hat{\sigma}_{\hat{b}}. \tag{6}$$

Then, let $n_{s,k}$ be proportional to the posterior probability that group k has the largest $\frac{\sigma_k}{\mu_k - \hat{\tau}}$:

$$n_{s,k} = N_s \cdot P\left(\frac{\sigma_k}{\mu_k - \hat{\tau}} \ge \frac{\sigma_i}{\mu_i - \hat{\tau}}, \forall i \ne k | \hat{\mu}_j, \hat{\sigma}_j, n_{s-1,j}, \forall j\right), \tag{7}$$

where $n_{s,k}$ is the total budget size allocated to group k after stage s and $\hat{\mu}_j, \hat{\sigma}_j^2$ are the sample mean and sample variance of group j based on $n_{s-1,j}$ previous observations. Proposition 2 provides an approximate way to estimate expression (7) when there are only two groups.

Proposition 2. *If only two groups are available, i.e.,* $K = 2$, *Assumption 2 holds and* $0 < \frac{\sigma_k}{\mu_k - \hat{\tau}} < 0.33, k = 1, 2,$ *the ratio of the total budget sizes allocated to these two groups according to expression (7) can be approximated as*

$$\frac{n_{s,1}}{n_{s,2}} \approx \frac{F(C_{1,2}; n_{s-1,1} - 1, n_{s-1,2} - 1)}{F(C_{2,1}; n_{s-1,2} - 1, n_{s-1,1} - 1)},$$

where

$$C_{i,j} = \frac{1 + \frac{1}{\hat{c}_j^2} - \frac{1}{n_{s-1,j}}}{1 + \frac{1}{\hat{c}_i^2} - \frac{1}{n_{s-1,i}}}, \quad i, j = 1, 2,$$

$$\hat{c}_k = \frac{\hat{\sigma}_k}{\hat{\mu}_k - \hat{\tau}}, \quad k = 1, 2,$$

$F(\cdot; v_1, v_2)$ is the cdf of the F-distribution with degrees of freedom v_1, v_2 and $\hat{\mu}_k, \hat{\sigma}_k^2, k = 1, 2$ are the group sample means and group sample variances based on $n_{s-1,k}$ observations.

Proposition 2 is proved based on the McKay's chi-square approximation for the coefficient of variation (McKay, 1932).

Proof of Proposition 2. Let random variables:

$$X_k = y(\boldsymbol{x}_{k,i}) - \hat{\tau}, k = 1, 2.$$

Under Assumption 2:

$$X_k \sim N(\mu_k - \hat{\tau}, \sigma_k^2) \quad \text{and} \quad c_k = \frac{\sigma_k}{\mu_k - \hat{\tau}}$$

is the coefficient of variation of the random variable X_k. McKay's approximation (McKay, 1932) shows that if $0 < c_k < 0.33$, the statistic:

$$W_k = \left(1 + \frac{1}{c_k^2}\right) \frac{(n_{s-1,k} - 1)\hat{c}_k^2}{1 + (n_{s-1,k} - 1)\hat{c}_k^2 / n_{s-1,k}}$$

is approximately distributed as the χ^2 distribution with degree of freedom $(n_{s-1,k} - 1)$, where $\hat{c}_k = \hat{\sigma}_k / (\hat{\mu}_k - \hat{\tau})$ is the estimate of the coefficient of variation based on $n_{s-1,k}$ observations. Under the assumption of independence, we can say that when $0 < c_i, c_j < 0.33$:

$$W_{i,j} = \frac{W_i/(n_{s-1,i} - 1)}{W_j/(n_{s-1,j} - 1)} = C_{i,j} \cdot \frac{1 + \frac{1}{c_i^2}}{1 + \frac{1}{c_j^2}},$$

is approximately distributed as the F-distribution with degrees of freedom $(n_{s-1,i} - 1)$ and $(n_{s-1,j} - 1)$, where:

$$C_{i,j} = \frac{1 + \frac{1}{\hat{c}_j^2} - \frac{1}{n_{s-1,j}}}{1 + \frac{1}{\hat{c}_i^2} - \frac{1}{n_{s-1,i}}}.$$

Therefore,

$$\begin{aligned} &P\left(\frac{\sigma_i}{\mu_i - \hat{\tau}} \ge \frac{\sigma_j}{\mu_j - \hat{\tau}} | \hat{\mu}_i, \hat{\sigma}_i, n_{s-1,i}, \hat{\mu}_j, \hat{\sigma}_j, n_{s-1,j}\right) \\ &= P(c_i \ge c_j | \hat{c}_i, n_{s-1,i}, \hat{c}_j, n_{s-1,j}) \\ &= P(W_{i,j} \le C_{i,j} | \hat{c}_i, n_{s-1,i}, \hat{c}_j, n_{s-1,j}) \\ &\approx F(C_{i,j}; n_{s-1,i} - 1, n_{s-1,j} - 1), \end{aligned}$$

where $F(\cdot; v_1, v_2)$ is the cdf of the F-distribution with degrees of freedom v_1 and v_2. When only two groups are available, we have:

$$\begin{aligned} \frac{n_{s,1}}{n_{s,2}} &= \frac{N_s \cdot P\left(\frac{\sigma_1}{\mu_1 - \hat{\tau}} \ge \frac{\sigma_2}{\mu_2 - \hat{\tau}} | \hat{\mu}_i, \hat{\sigma}_i, n_{s-1,i}, i = 1, 2\right)}{N_s \cdot P\left(\frac{\sigma_2}{\mu_2 - \hat{\tau}} \ge \frac{\sigma_1}{\mu_1 - \hat{\tau}} | \hat{\mu}_i, \hat{\sigma}_i, n_{s-1,i}, i = 1, 2\right)} \\ &\approx \frac{F(C_{1,2}; n_{s-1,1} - 1, n_{s-1,2} - 1)}{F(C_{2,1}; n_{s-1,2} - 1, n_{s-1,1} - 1)}, \end{aligned}$$

i.e., Proposition 2 is proved. □

When $K = 2$, Figure 1 shows the percentage of the total budget that should be allocated to Group 1 after stage s, according to Proposition 2. As shown in the left figure, when both groups have the same budget size after the

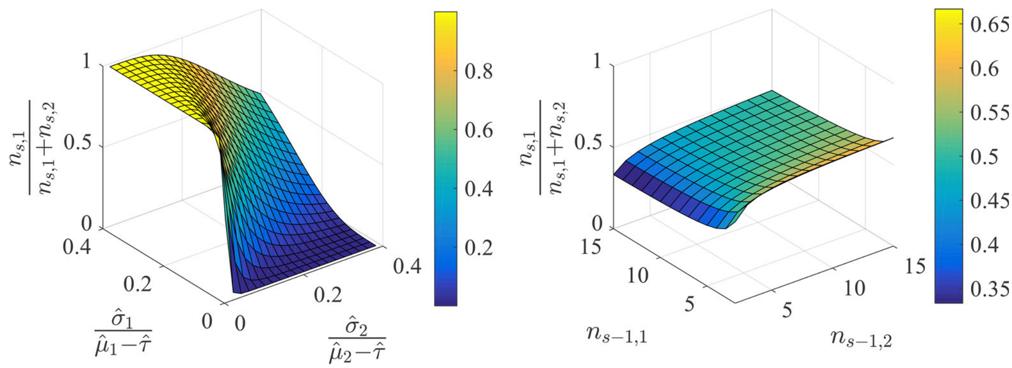

Figure 1. The percentage of the total budget allocated to Group 1. $K = 2$. In the left figure, $n_{s-1,k} = 5, k = 1, 2$ and in the right figure, $\frac{\hat{\sigma}_k}{\hat{\mu}_k - \hat{\tau}} = 0.1, k = 1, 2$.

previous stage ($n_{s-1,k} = 5, \forall k$), more budget should be allocated to the group with a larger $\frac{\hat{\sigma}_k}{\hat{\mu}_k - \hat{\tau}}$ (i.e., higher $\hat{\sigma}_k$ and lower $\hat{\mu}_k$). In the right figure, when the $\frac{\hat{\sigma}_k}{\hat{\mu}_k - \hat{\tau}}$ of both groups are the same, more budget should be allocated to the group with a smaller $n_{s-1,k}$, since less data are observed. Nevertheless, the gap becomes insignificant as both budget sizes increase.

When more than two groups are available, i.e., $K > 2$, we extend Proposition 2 by using the budget size allocated to the current best group $\hat{b}$ as the reference:

$$\frac{n_{s,k}}{n_{s,\hat{b}}} = \frac{F(C_{k,\hat{b}}; n_{s-1,k} - 1, n_{s-1,\hat{b}} - 1)}{F(C_{\hat{b},k}; n_{s-1,\hat{b}} - 1, n_{s-1,k} - 1)}, \quad \forall k \neq \hat{b}, \tag{8}$$

$$n_{s,\hat{b}} = N_s / \left(\sum_{k=1}^{K} \frac{F(C_{k,\hat{b}}; n_{s-1,k} - 1, n_{s-1,\hat{b}} - 1)}{F(C_{\hat{b},k}; n_{s-1,\hat{b}} - 1, n_{s-1,k} - 1)} \right), \tag{9}$$

where

$$C_{i,j} = \frac{1 + \frac{1}{\hat{c}_j^2} - \frac{1}{n_{s-1,j}}}{1 + \frac{1}{\hat{c}_i^2} - \frac{1}{n_{s-1,i}}}, \forall i, j, \qquad \hat{c}_k = \frac{\hat{\sigma}_k}{\hat{\mu}_k - \hat{\tau}}, \forall k, \tag{10}$$

and $F(\cdot; \nu_1, \nu_2)$ is the cdf of the F-distribution with degrees of freedom ν_1 and ν_2. $n_{s,k}, \forall k$ are rounded to integers. Then, additional $\max\{0, n_{s,k} - n_{s-1,k}\}$ new solutions are sampled from group k in the order of the additional sample size descending.

Algorithm 1 describes in detail how to implement the proposed method. A small trial budget is allocated to each group to estimate the group performance at the first-stage sampling. At each new sampling stage, the total budget size increases by a fixed size. The estimated group means, group variances, the current best group and the estimated threshold are updated. The new budget is allocated according to equations (8) (9) (10). Given the group budget sizes, uniform sampling or other sampling methods considering exploration can be applied to sample new solutions from each group. This process is repeated until the budget is exhausted.

Algorithm 1

$N, K, \alpha, n_{1,k}, \forall k, \Delta \quad \leftarrow \quad$ User-defined parameters.
Cluster the feasible solutions into K groups.
$s = 1$.
for $k = 1$ to K **do**
  Sample $n_{s,k}$ solutions from group k.
  Calculate group sample mean $\hat{\mu}_k$ and group sample variance $\hat{\sigma}_k^2$.
end for

$$N_s = \sum_k n_{s,k}.$$

Calculate $\hat{b}$ and $\hat{\tau}$ using equations (5) and (6).
while $N - N_s > 0$ **do**
  $s = s + 1$.
  $N_s = \min\{N_{s-1} + \Delta, N\}$.
  Calculate $n_{s,k}, \forall k$ using equations (8) (9) and (10).
  Sample $\max\{0, n_{s,k} - n_{s-1,k}\}$ solutions from group $k, \forall k$.
  Update group sample mean $\hat{\mu}_k$ and group sample variance $\hat{\sigma}_k^2$.
  Update $\hat{b}$ and $\hat{\tau}$ using equations (5) and (6).
end while

3.2.1. Asymptotic performance

The sampling noise is considered in Algorithm 1 for the situation where group information is unknown. Theorem 2 shows that, under Assumption 2, Algorithm 1 converges to the optimal budget allocation policy in Theorem 1, in which the exact groups information are known in advance, when the number of sampling stages approaches to infinity.

Theorem 2. *Under Assumption 2, if $\alpha < 0.5, \mu_i + z_\alpha \sigma_i \neq \mu_j + z_\alpha \sigma_j, \forall i \neq j$ and the number of sampling stages approaches to infinity, Algorithm 1 converges to the optimal allocation policy in Theorem 1, which is allocating all the budget to the group having the smallest α-quantile:*

$$\lim_{s \to \infty} n_{s,b} / N_s = 1,$$

where $b = \arg\min_k \{\mu_k + z_\alpha \sigma_k\}$.

We first prove that, as the number of sampling stages approaches to infinity, the budget sizes allocated to all groups $n_k, \forall k$ approach to infinity. This involves that the current best group $\hat{b}$ approaches to the real best group b. Then, we prove that as the number of sampling stages approaches to infinity, the ratio of the budget size allocated to the current best group $\hat{b}$ to the total budget size N_s approaches to unity. The detailed proof can be found in Appendix A.

3.2.2. Robustness to wrong clustering

When an inappropriate clustering rule is applied, it is possible that all the groups have similar group performances, i.e., the clustering rule cannot separate the promising solutions from bad ones. Theorem 3 shows that when all groups have the same means and the same variances, Algorithm 1 tends to allocate equal budget sizes to all groups as the number of sampling stages approaches to infinity. This means that if all groups have the same performance, instead of exploiting one of the groups, Algorithm 1 focuses more on the exploration searching all the groups. The detailed proof can be found in Appendix A.

Theorem 3. *If $\alpha < 0.5, \mu_k = \mu_b, \sigma_k = \sigma_b, \forall k$ and the number of sampling stages approaches to infinity, Algorithm 1 converges to allocating equal budget sizes to all the groups, i.e.,*

$$\lim_{s\to\infty} n_{s,k}/n_{s,\hat{b}} = 1, \forall k \neq \hat{b}.$$

4. Numerical results

The proposed method is tested in designed cases to analyze its performance under different circumstances and the influence of the user-defined parameters. Then, it is applied to a Griewank function to show the effect of the applied clustering rules.

4.1. Comparison of budget allocation strategies among groups

In this section, we assume that all the feasible solutions are already clustered. We would like to investigate the benefit of using Algorithm 1 to allocate the budget among groups. The following five budget allocation strategies are applied:

BAQM: Budget Allocation for Quantile Minimization. The budget is allocated using Algorithm 1, i.e., the proposed method.

AATB: Allocate all the added budget to the current best group $\hat{b}$ at each stage s:

$$n_{s,\hat{b}} - n_{s-1,\hat{b}} = N_s - N_{s-1} \quad \text{and} \quad n_{s,k} - n_{s-1,k} = 0, \forall k \neq \hat{b}.$$

Modified ε-greedy: The new budget is allocated to the current best group $\hat{b}$ with probability $1-\varepsilon$ and allocated to other groups with probability ε. The ε-greedy method is frequently used for classic multi-armed bandit problems. It is applied here with the definition of the best group modified using the quantile instead of the mean as the criterion. The aforementioned AATB method can be regarded as an extreme ε-greedy method with $\varepsilon = 0$. In the following experiments, the ε value is selected as 0.1 according to preliminary analysis.

Table 1. The distribution parameters of each group.

System ID	μ	σ
1	[10, 15, 20, 25, 30]	[4, 4, 4, 4, 4]
2	[10, 15, 20, 25, 30]	[6, 6, 6, 6, 6]
3	[10, 15, 20, 25, 30]	[3, 4, 5, 6, 7]
4	[10, 15, 20, 25, 30]	[7, 6, 5, 4, 3]
5	[10, 10, 10, 10, 10]	[7, 6, 5, 4, 3]
6	[10, 15, 20, 25, 30]	[1, 5, 5, 5, 5]

OCBA: Optimal Computing Budget Allocation. The budget is allocated, at each stage s, using the OCBA formulas proposed by Chen *et al.* (2000):

$$\frac{n_{s,i}}{n_{s,j}} = \left(\frac{\hat{\sigma}_i/(\hat{\mu}_{b'} - \hat{\mu}_i)}{\hat{\sigma}_j/(\hat{\mu}_{b'} - \hat{\mu}_j)}\right)^2, \forall i \neq j \neq b',$$

and

$$n_{s,b'} = \hat{\sigma}_{b'} \sqrt{\sum_{i \neq b'} \frac{n_{s,i}^2}{\hat{\sigma}_i^2}},$$

where $b' = \arg\min\{\hat{\mu}_k, \forall k\}$. The OCBA method is frequently applied to stochastic problems to maximize the approximate probability that the selected design b' is the best design (the design having the lowest mean). From another perspective, the OCBA method attempts to separate the mean of the current best group from the other groups, which is different from the goal of this article. However, the OCBA method also assigns more budget to the group with a higher variance and a lower mean, which is similar to our algorithm. Thus, it is also applied in the experiments as a budget allocation pattern reference.

EBA: Equal Budget Allocation. The budget is allocated equally to all groups:

$$n_k = N/K, \forall k = 1, \ldots, K,$$

where the indivisible budget is allocated arbitrarily.

The first four strategies are applied dynamically with a fixed budget size added to the total budget size at each sampling stage. The EBA is applied in one stage. The EBA is used as a benchmark in the following experiments.

In the following experiments, 10 000 replications are executed. For the sake of simplicity, the first-stage sampling sizes are assumed to be the same among all groups, i.e., $n_{1,i} = n_{1,j}, \forall i, j$.

4.1.1. Effect of group parameters

Six cases are considered in this section. In each case, five groups are clustered. Assumption 2 holds, i.e., the objective function values in each group are independently and normally distributed. Group means and group standard deviations are presented in Table 1. These parameters are designed to test the behavior of different strategies in

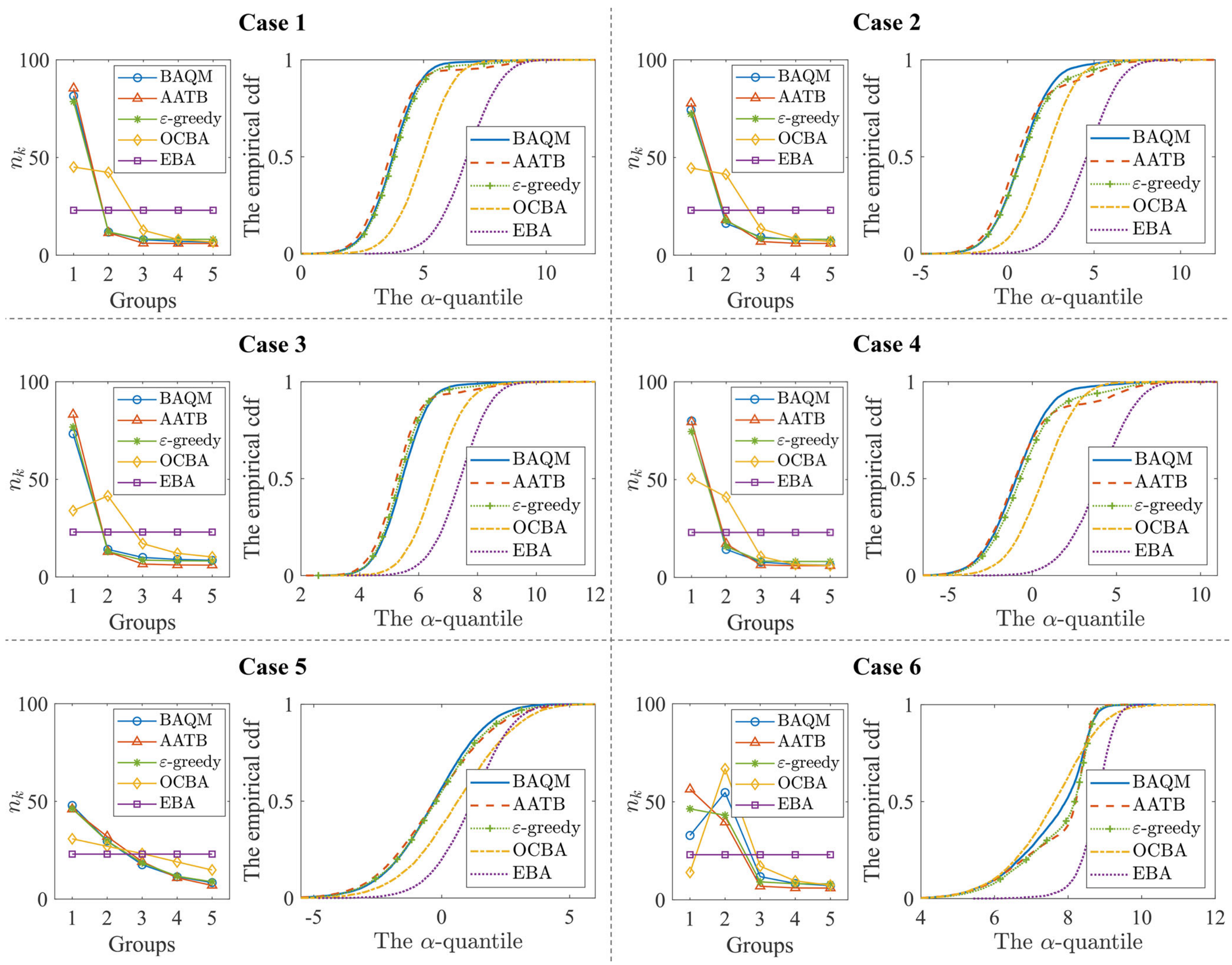

Figure 2. The average group budget size (left) and the empirical cdf of the α-quantile (right). $N = 100, \alpha = 0.05, n_{1,k} = 3, \forall k, \Delta = 1$. A total of 10 000 replications are executed.

different conditions. A budget of $N = 100$ is available and the first stage sampling sizes are $n_{1,k} = 3, \forall k$. We are interested on the α-quantile ($\alpha = 0.05$) of the objective function values in the final sample set, i.e., the fifth smallest observation, and only one budget size is added at each stage (i.e., $\Delta = 1$). Under the given α value, the first group is the best group in all the cases except the last one in which the second group is the best group.

Figure 2 shows the average total budget sizes allocated to different groups, i.e., n_k, using different budget allocation strategies (the left figures) and the empirical cdf of the α-quantile of the objective function values in the final sample set among 10 000 replications (the right figures). The first conclusion is that the first four strategies all allocate, on average, most of the budget to the best group, except the OCBA in Case 3, the AATB and the modified ε-greedy in Case 6.

In most cases, the AATB behaves slightly better than the BAQM under a certain probability (the first part of the AATB's empirical cdf is on the left-hand side of that of the BAQM). This is because the AATB will not waste budget exploring other groups once it identifies the best group. However, under a certain probability, the AATB does not identify the real best group and allocates most of the budget to a wrong group. This is the reason why the tail of the AATB's empirical cdf collapses. As shown in Case 2 and Case 4, the AATB has more difficulty in identifying the best group when the performances of good groups have large variability. Compared with the AATB, the BAQM wastes a small budget in exploration, but significantly reduces the probability that poor designs are generated, especially when the group performances are highly varied.

It can be found that compared with the AATB, the modified ε-greedy also tries to improve the identification of the best group by using some budget for exploration. The performance of the modified ε-greedy is highly sensitive to the selection of the ε value. A large ε value can improve the situation that the tail of the empirical cdf collapses, but it will waste too much budget once the real best group is identified. The modified ε-greedy can be improved by reducing the ε value as the sampling stage increases. Nevertheless, additional effort are required to determine the adaptive function and to tune the adaptive parameter for the ε. The BAQM has a better performance than the modified ε-greedy when good groups have large variability, as in Case 2, Case 4, and Case 6. When the variability of good groups is small,

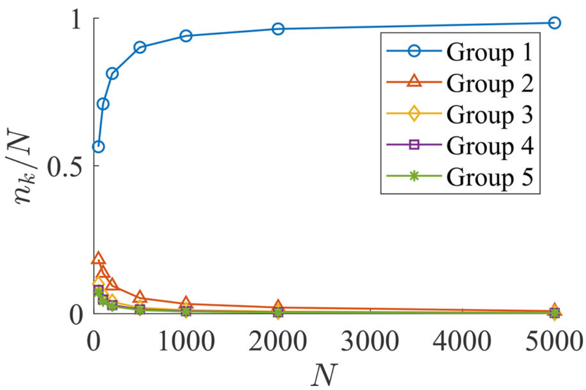

Figure 3. The proportions of the total budget allocated to different groups as N increases. $\alpha = 0.05, n_{1,k} = 3, \forall k$ and $\Delta = 1$. A total of 1000 replications are executed.

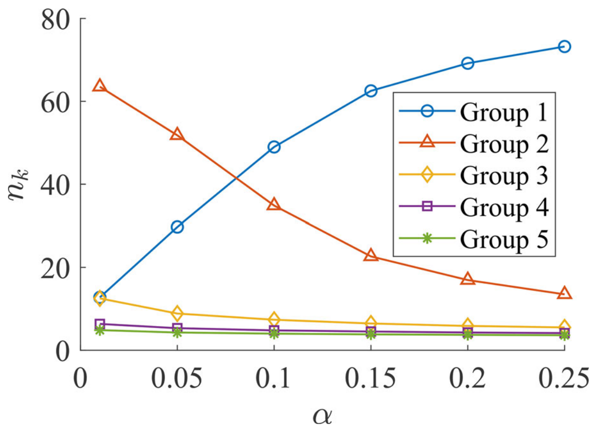

Figure 4. The average group budget sizes with varying α values. $N = 100, n_{1,k} = 3, \forall k$ and $\Delta = 1$. A total of 10 000 replications are executed.

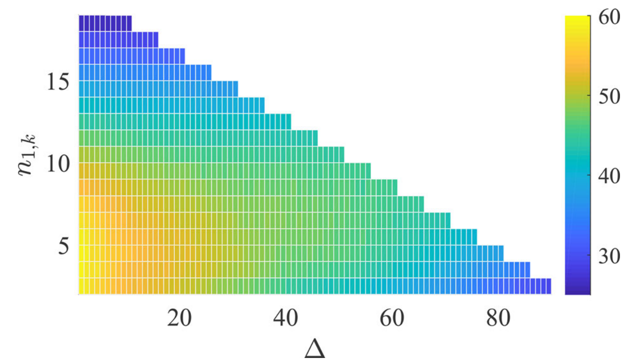

Figure 5. The average budget size allocated to the best group with varying $n_{1,k}$ and Δ values. $N = 100$ and $\alpha = 0.05$. A total of 10 000 replications are executed.

the BAQM can achieve a similar performance to that of the modified ε-greedy without the tuning of the parameters.

The OCBA does not perform well when the best group has a small variability. In Case 3, the OCBA allocates more budget to the second group, however, the first group is the best group. Indeed, the goal of the OCBA is to separate the mean of the best group from the means of the other groups. In this case, the variance of the first group is small. Allocating more budget to the second group enables this goal to be achieved more quickly. This is also the reason why the OCBA has a good performance in Case 6, in which the best group is the group that has the largest variance and a small mean.

4.1.2. Effect of algorithm parameters

In this section, we investigate the effect of the user-defined parameters $(N, \alpha, n_{1,k}, \forall k, \Delta)$ in the proposed budget allocation method.

In Figure 3, the proposed method is applied to Case 2 with parameters $\alpha = 0.05, n_{1,k} = 3, \forall k, \Delta = 1$ and different total budget sizes N. The y-axis shows the proportions of the total budget being allocated to different groups, i.e., n_k/N. As the total budget size increases, the proportion of the total budget allocated to group 1 (the best group) climbs quickly at the first iterations, then slowly tends to one. This is consistent to Theorem 2 that, as the number of sampling stages increases, the proposed method tends to allocate more budget to the real best group and converges to the optimal budget allocation in Theorem 1.

In Figure 4, Case 6 is considered, in which group 1 has the lowest mean and a much smaller variance compared with the other groups. The proposed method is applied with parameters $N = 100, n_{1,k} = 3, \forall k, \Delta = 1$. The average budget size allocated to each group by the proposed method with different α values is presented in the figure. A higher budget size is assigned to group 1 (the group with the lowest mean) when the α value is large. When the α value is small, the method focuses more on group 2 (the group with the second lowest mean but a larger variance). This is consistent with the discussion about the selection of the α value in Section 3.2.

In Figure 5, the proposed method is applied to Case 2 with parameters $N = 100, \alpha = 0.05$, $n_{1,k}$ values varying between 2 and 18, and Δ value varying between 1 and 90. The color presents the average budget size allocated to the best group among 10 000 replications. The lighter the color, the higher the budget size assigned correctly.

A small Δ value helps to allocate budget correctly independent of the first stage sampling sizes $n_{1,k}$. Nevertheless, a small Δ value, i.e., many sampling stages, may be a drawback in some cases, such as a fixed setup time is required to start a simulation model or when extensive parallel computing power is available. Given the Δ value, the selection of the first stage sizes $n_{1,k}$ also affects the budget allocation results. Large $n_{1,k}$ values waste budget in the first stage sampling whereas small $n_{1,k}$ values may result in wrong allocation in the following stages, due to of the biased estimated group information. As the selected Δ value increases, the corresponding optimal $n_{1,k}$ values increase as well.

4.1.3. Effect of group distribution types

In this section, we investigate the effect of the group distribution types in the proposed budget allocation method. The goal is to test the impact of Assumption 2.

We keep the same group means and group variances as in Case 2 and change the distribution type of the values in each group. Six new cases are considered. Table 2 shows the distribution types of different cases and Figure 6 shows the pdf shapes of the values in group 1 in different cases. In Case 9 and Case 10, the beta distributions are scaled to meet the same group means and group variances as in Case 2. Multi-modal distributions are considered in Case 11 whose pdf presents the shape of two triangles connected together as shown in Figure 6. In Case 12, the five groups follow different distribution types, which are the distribution types from Case 7 to Case 11, respectively. As in the previous analysis, all the strategies are executed dynamically with $N = 100, \alpha = 0.05, n_{1,k} = 3, \forall k, \Delta = 1$.

Figure 7 shows the empirical cdf of the α-quantile of the objective function values in the final sample set in different cases. Compared with the AATB, the BAQM significantly reduces the probability that poor designs are generated by wasting only a limited amount of budget on exploration. In most cases, the BAQM performs better than the modified ε-greedy. It also has better behavior than the OCBA if the group distribution types are unimodal. If the group distribution type has a shape that is particularly different from the normal distribution, the BAQM identifies the wrong best group under a certain probability, but it is still more robust than the AATB. Overall, even if Assumption 2 does not hold, the proposed budget allocation method works well.

Table 2. The distribution types in different cases.

Case ID	*Distribution type*
7	Symmetric triangular distribution
8	Uniform distribution
9	Scaled beta(1,2)
10	Scaled beta(2,1)
11	Multi-modal distribution
12	Mixed distribution types among groups

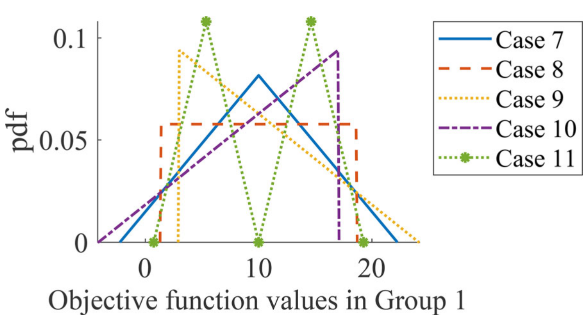

Figure 6. The pdf of the objective function values in Group 1 in different cases.

4.2. Effect of clustering rules

In this section, the proposed budget allocation method is applied to the Griewank function:

$$y = 1 + \frac{1}{4000}x_1^2 + \frac{1}{4000}x_2^2 - \cos(x_1)\cos\left(\frac{x_2}{\sqrt{2}}\right),$$

and the goal is to minimize the Griewank function in the feasible domain $[x_1, x_2]^T \in [-4, 4]^2$. In this case, two clustering rules are applied by partitioning the original solution space into different regions as shown in Figure 8. In the left figures, the objective function values at different locations are presented by different colors (blue indicates promising regions) and the numbers are the groups' labels. The right figures show the box-plot of the total budget size allocated to each group among 10 000 replication, using the proposed method with parameters $N = 100, \alpha = 0.05, n_{1,k} = 3, \forall k, \Delta = 1$.

The first clustering rule partitions the feasible domain into four regions that have similar objective function performances. This partitioning cannot separate good solutions from bad

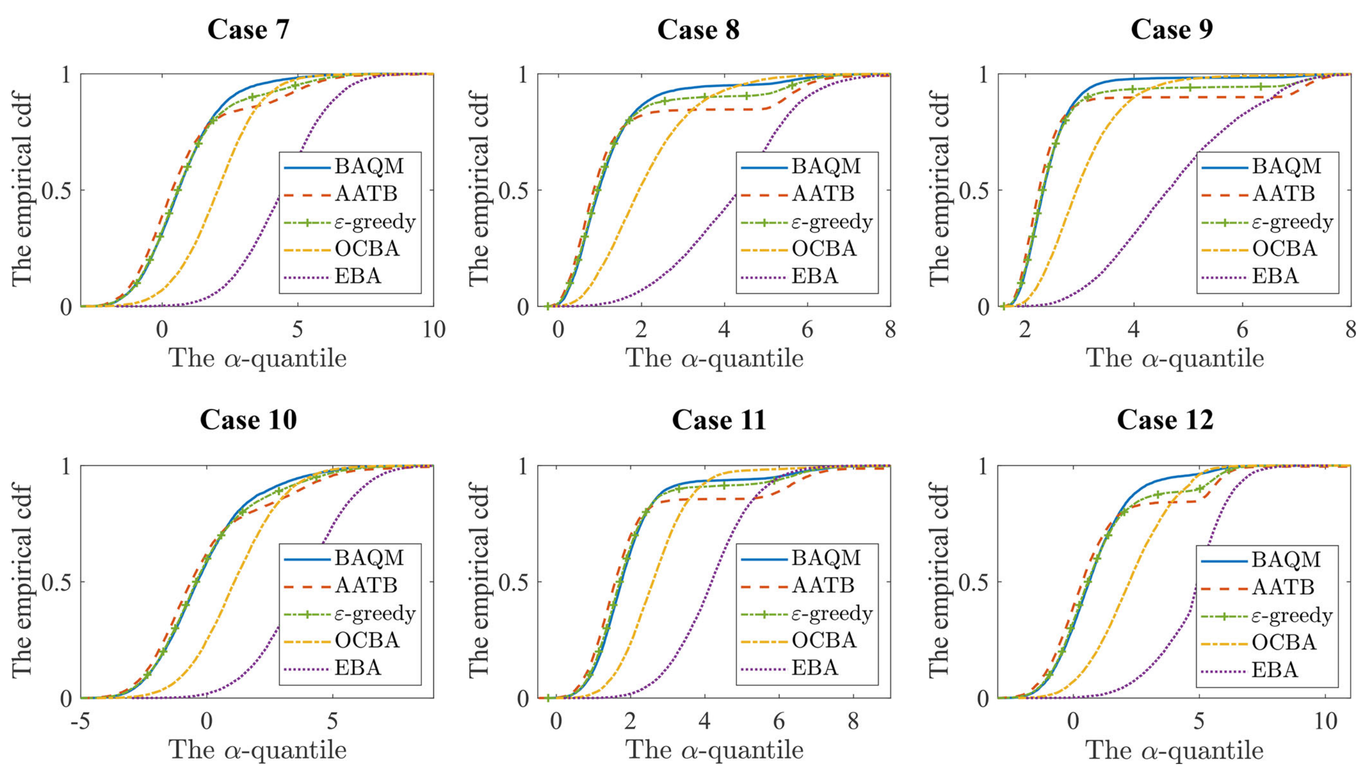

Figure 7. The empirical cdf of the α-quantile. $N = 100, \alpha = 0.05, n_{1,k} = 3, \forall k, \Delta = 1$. A total of 10 000 replications are executed.

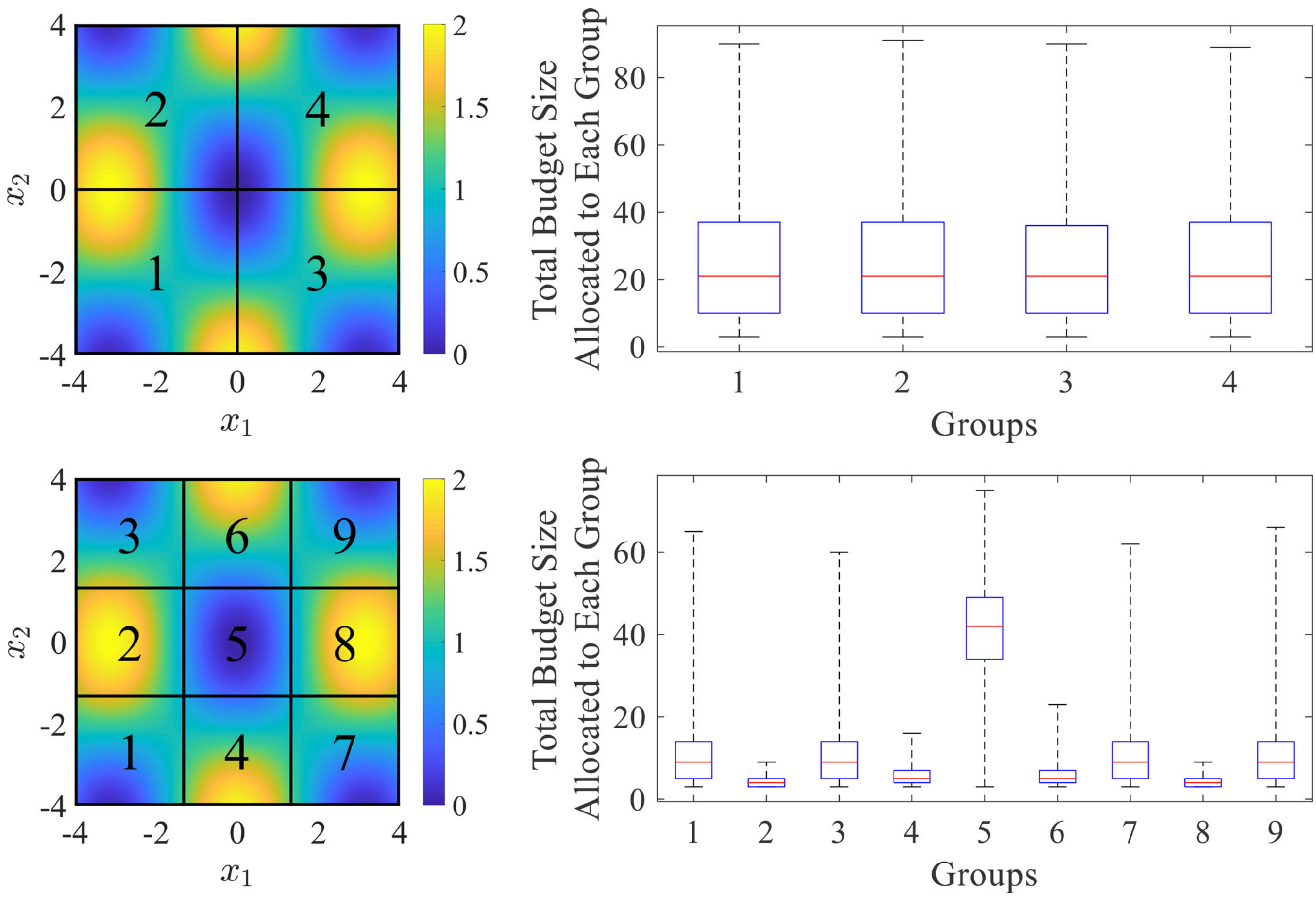

Figure 8. The clustering rules on the contour of the Griewank function (left) and their corresponding budget allocation information (right). In the left figures, the objective function values at different locations are presented by different colors. A total of 10 000 replications are executed.

ones and no region (group) dominates other regions. The proposed method allocates, on average, equal budget sizes to each group. This is consistent with Theorem 3. The proposed method focuses more on the exploration in this situation.

The second clustering rule partitions the feasible solution space into nine regions that almost separate promising areas from non-promising ones. The proposed method allocates, on average, most of the budget to group 5, which is the best group. Groups 1, 3, 7, and 9 also have higher budget sizes than the other four groups since better performances are identified.

5. Applications to manufacturing systems

In this section, the proposed method is tested on a buffer allocation problem of a transfer line and a server allocation problem of a multi-stage manufacturing system with re-entries to generate initial designs for selected black-box optimization algorithms. In addition, an example of using multi-fidelity information to cluster feasible solutions is presented in Section 5.2.

5.1. A transfer line buffer allocation problem

This section is to test the impact of using the proposed method as a population initializer in optimization problems. In the experiments, a transfer line is considered with data collected from a vehicle injector assembly line. It is composed of 13 stations that are connected by buffers with capacities denoted as $\boldsymbol{x} = [x_1, \ldots, x_{12}]^T$. The block after service rule is applied. It is assumed that the first station is never starved and the last station is never blocked. The processing times are deterministic and the line is well balanced in terms of the processing time, i.e., the processing times on all stations are close. For the sake of simplicity, we assume that the processing times, are all equal to the average processing time and this time is taken as the time unit. Each station has a probability p_i that it will stop in one time unit, and the repair probability in one time unit is denoted as r_i. Table 3 presents the estimated p_i, r_i values and the availability of each station $e_i, \forall i$. The mean throughput of the studied line is evaluated by the DDX algorithm (Dallery *et al.*, 1988), denoted as $th_{\text{DDX}}(\boldsymbol{x})$.

The problem investigated in this section is to minimize the total buffer capacity with a throughput satisfaction: $th_{\text{target}} = 0.83$. We formulate the problem using a penalty function:

$$\min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{12} x_i + h(th_{\text{target}} - th_{\text{DDX}}(\boldsymbol{x}))^+ \middle| \sum_{i=1}^{12} x_i \leq 400, x_i \geq 3, x_i \in \mathbb{N}, \forall i \right\},$$

where $h(th_{\text{target}} - th_{\text{DDX}}(\boldsymbol{x}))^+$ indicates that the value of the penalty function is equal to zero if the throughput target is

Table 3. The stoppage information of each station.

Station ID	S5	S7	S8	S1-S4, S6, S9-S13
p_i	0.0038	0.0044	0.0022	0.0009
r_i	0.0209	0.0542	0.0444	0.0976
$e_i = r_i/(r_i + p_i)$	0.846	0.925	0.953	0.991

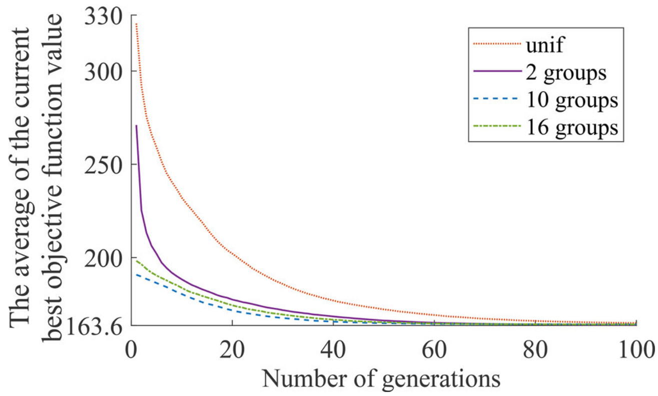

Figure 9. The average of the current best objective function value as the number of generations increases. A total of 200 replications are executed.

satisfied and equal to the gap multiplied by a large value h, if the throughput target is not satisfied. The value of h is selected as 5000 in the experiments. Similar results can be found with the h value varying from 4500 to 10 000. The upper bound of the total buffer capacity is set to narrow down the search space. The lower bound of each buffer is set to avoid errors in the execution of the DDX method. A GA is applied to search for the optimal buffer allocation with population size $N=50$ and other settings being the default values in Matlab.

The initial population is generated using the proposed method with $\alpha = 0.2, n_{1,k} = 3, \forall k, \Delta = 1$. Feasible solutions are clustered into two groups, 10 groups and 16 groups in different experiments based on two criteria: (i) whether the capacity of each buffer in the solution is correlated to the mean availability of the adjacent machines; and (ii) the range to which total buffer capacity of the solution belongs. More details can be found in Appendix C. Uniform sampling is applied within a group.

The best solution found during the experiments is $\boldsymbol{x}^* = [3, 3, 8, 17, 46, 44, 26, 4, 3, 3, 3, 3]$ with objective function value 163.6 and throughput 0.8299. Figure 9 shows the average of the current best objective function values at different generations in the GA algorithm. A total of 200 replications are executed and the initial population is re-sampled in each replication. The label "2 groups" indicates that in the proposed method, the feasible solutions are clustered using only the first criterion. Compared with uniformly sampling the initial population, using the initial population generated by the proposed method improves the efficiency of the GA algorithm significantly in the studied case, even though only the first criterion is used for clustering. Further separating the solutions based on the total buffer capacity (e.g., label "10 groups") can make the characteristics of good solutions more specific, which also helps the optimization. However, if there are too many groups (e.g., "16 groups" in Figure 9), this advantage is counterbalanced by the loss of the ability for exploitation in the proposed method, because most of budget is used for exploration in the first stage sampling.

Table 4. The production sequence, the mean and the standard deviation of the processing times.

Part type 1			Part type 2		
Workstation	Mean (h)	Std (h)	Workstation	Mean (h)	Std (h)
1	2	0.25	2	5	0.8
3	1.6	0.2	4	4.5	0.19
4	3.5	0.15	3	2.5	0.28
5	4	0.25	1	2.2	0.25
6	3.5	0.4	6	3	0.45
3	2.5	0.28	–	–	–

5.2. A multi-stage manufacturing system server allocation problem

The proposed budget allocation method is applied to a server allocation problem of a multi-stage manufacturing system with re-entries to generate the initial design. A fixed number of servers are assigned to six workstations to minimize the 80% quantile of the order lead times. The order lead time quantile is estimated by a highly-detailed simulation model (high-fidelity model) and its execution is time-consuming. Two analytical methods (low-fidelity models), which can provide fast but biased estimates on the order lead time, are used to cluster the feasible solutions.

In real-world systems, it is common that the acquisition of the high-fidelity estimation of the system performance (highly-detailed simulation response or data from the field) is expensive (or time-consuming). Low-fidelity estimates (from analytical methods or low-detailed simulation models) are biased, but easy and fast to obtain. In this section, we present a way to cluster feasible solutions in this kind of application and show the benefits of using the proposed method as a population initializer in search algorithms.

5.2.1. System description

A multi-stage manufacturing system composed of six workstations is producing two types of parts. Each workstation has several identical servers (machines or operators) and each part type passes through different workstations following a specific sequence. Table 4 presents the production sequence of each part type, the mean and the standard deviation of their processing time at each workstation. The processing times are independently and lognormally distributed. The transportation times between workstations are assumed to be negligible and the buffer capacities in between are assumed to be infinite. The first-come first-served rule is applied to each workstation.

Orders arrive in batches. A single order contains only one part type. Once an order arrives, the system starts to produce this part type and when all the parts in this order are produced, they are delivered at the same time. The arrival rates of different part types are 1/130 and 1/70 with batch sizes 150 and 100, respectively. The inter-arrival times of the orders are independently and exponentially distributed.

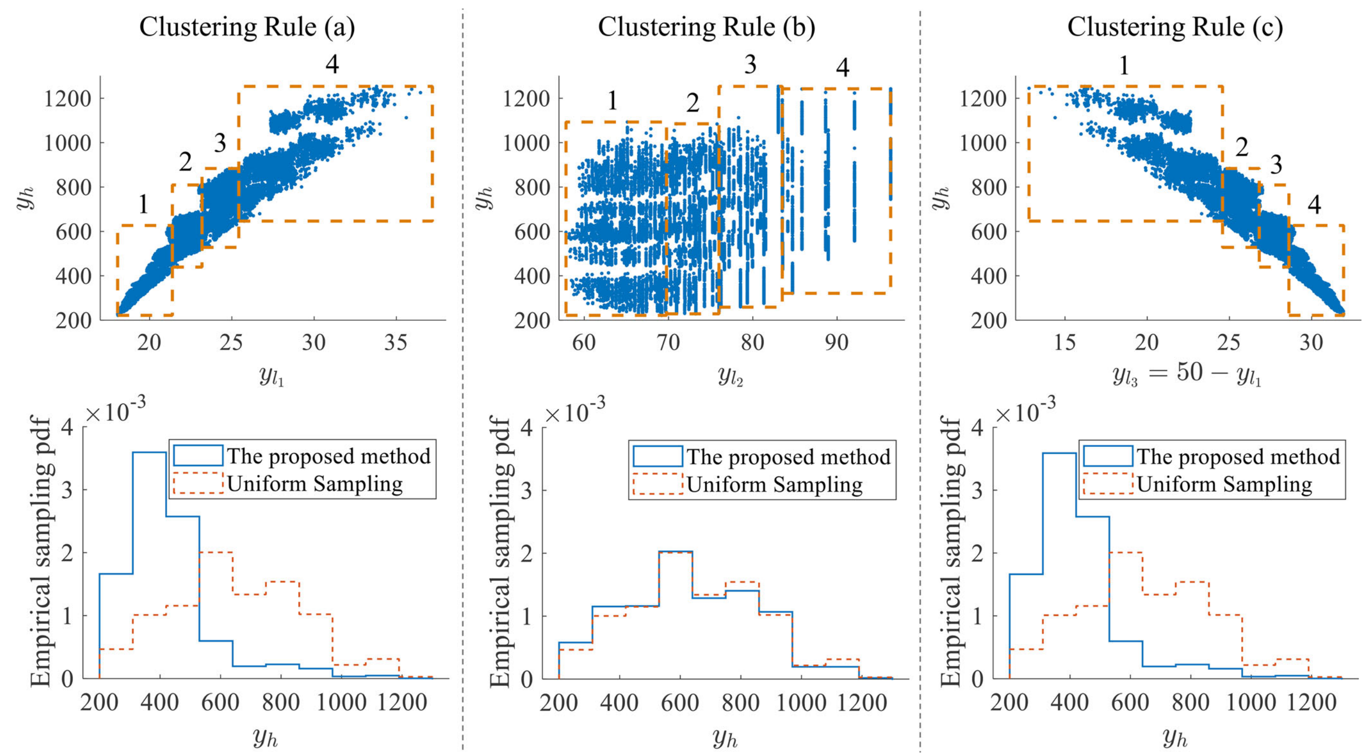

Figure 10. The clustering rules (top figures) and their empirical sampling pdf on the simulation responses (bottom figures). Clustering rules (a) (b) and (c) are based on the low-fidelity models y_{l_1}, y_{l_2} and y_{l_3}, respectively. The numbers from "1" to "4" indicate the group IDs. $N = 100, \alpha = 0.05, n_{1,k} = 2, \forall k, \Delta = 1$. A total of 10 000 replications are executed.

The key performance indicator is the order lead time, i.e., the duration between the order arrival time and the order delivery time. The goal of this problem is to assign a total of 65 servers to these six workstations to minimize the 80% quantile of the order lead times. Denote the decision variables, i.e., the server numbers at different workstations, as $\boldsymbol{x} = [x_1, ..., x_6]^T$. In order to satisfy the demands, server numbers have lower bounds, which are $[6, 8, 9, 11, 5, 9]$ for the six workstations, respectively. Therefore, a total of 26 334 assignment solutions are feasible.

The objective function (i.e., the 80% quantile of the order lead times) is calculated using a highly-detailed simulation model, denoted as $y_h(\cdot)$, with simulation length 1×10^7 hour, warm-up length 1.5×10^5 hour and 10 independent replications. The average length of the half confidence interval of the mean lead time is about 2% of its estimate. All the feasible solutions are evaluated and their simulation responses are regarded as the exact objective function values in the following experiments, i.e., we assume the objective function is deterministic in execution. On the average, it takes about 9.5 minutes to obtain a simulation output using a laptop (Intel(R) Core(TM) i7-6600U CPU @ 2.6GHz 2.81 GHz, RAM 16GB), which means about 174 days to evaluate all the feasible solutions.

In addition to the simulation model, two kinds of analytical method are also applied. One is an Open Jackson Network, denoted as $y_{l_1}(\cdot)$, which assumes all inter-arrival times and processing times are independently and exponentially distributed. Also, the parts in the same order are assumed to arrive one by one rather than in batches. The $M/M/c$ formulas are used in the Open Jackson Network and the average system time is used as the estimation of the objective function. The other analytical method, denoted as $y_{l_2}(\cdot)$, is developed assuming there is no interaction between different orders and that the processing times are deterministic. Compared with the simulation model, these two analytical methods are fast in execution. They require about a total of 5 seconds and 0.5 seconds to run all the feasible solutions in the same laptop, respectively.

In the analyzed case, the optimal solution is $\boldsymbol{x}^* = [8, 11, 12, 14, 8, 12]^T$ with high-fidelity estimate of the 80% quantile of the order lead times $y_h(\boldsymbol{x}^*) = 222$ hours. The low-fidelity estimates of the optimal solution are $y_{l_1}(\boldsymbol{x}^*) = 18$ hours and $y_{l_2}(\boldsymbol{x}^*) = 69$ hours.

5.2.2. Effect of low-fidelity information

$K = 4$ groups are clustered based on the two analytical methods described above, as well as a third low-fidelity model which is defined as

$$y_{l_3}(\boldsymbol{x}) = 50 - y_{l_1}(\boldsymbol{x}).$$

The scatter plots of the analytical method outputs (the x-axis) versus the simulation responses y_h (the y-axis) and the clustering information are as shown in Figure 10 (top figures), in which the numbers from "1" to "4" indicate the group IDs. Notice that Assumption 2 does not hold in the analyzed case.

As shown in Figure 10, y_{l_1} shows a significant positive correlation with y_h and provides a good reference for the selection of promising solutions. However, this knowledge cannot be known in advance and it could vary case by case.

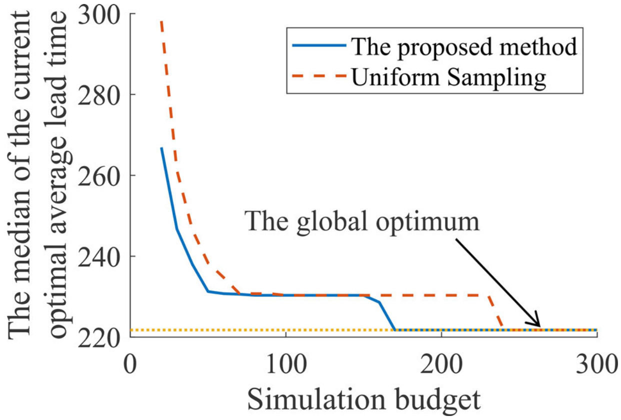

Figure 11. The the median of the current optimal objective function value. A total of 5000 replications are executed.

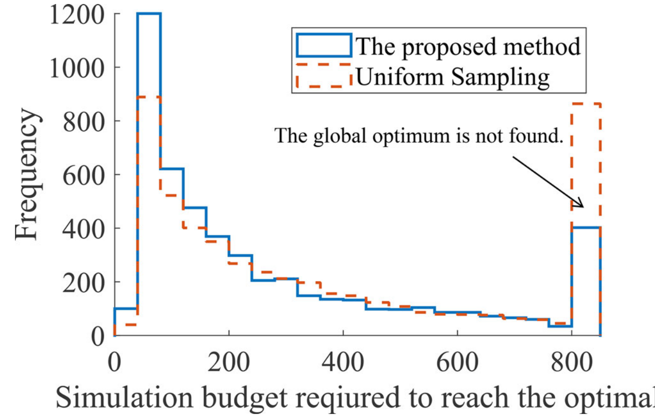

Figure 12. The histogram of the total simulation budget required to reach the global optimum. 5000 replications are executed.

The selected low-fidelity models may give useless information, like y_{l_2}, or worse, erroneous information is provided, such as y_{l_3}. In this situation, poor initial designs will be generated if we pick up solutions with good low-fidelity estimates.

The bottom figures in Figure 10 show the empirical sampling pdf on the simulation responses when different budget allocation methods are applied. The proposed method is applied with parameters $N = 100, \alpha = 0.05, n_{1,k} = 2, \forall k, \Delta = 1$ and uniform sampling is applied within a group. The label "Uniform Sampling" means solutions are uniformly sampled from the whole feasible domain.

Using the proposed budget allocation method enables the solutions with a lower objective function value to have higher sampling probabilities, whereas less promising solutions are less likely to be sampled, if proper clustering rules are applied, such as clustering rule (a) and (c). The average budget sizes allocated to different groups in clustering rule (a) and (c) are around $[88.8, 3.7, 3.6, 3.9]$ and $[3.9, 3.6, 3.7, 88.8]$, respectively. The case of clustering policy (c) shows that the proposed method can also have a good performance even if the selected low-fidelity provides a wrong ranking on the feasible solutions.

If the selected clustering rule does not provide much help, the proposed method tends to allocate similar budget sizes to all groups and behaves similarly to uniform sampling, such as clustering rule (b). The average budget sizes allocated to different groups in clustering rule (b) are around [31,35,23,11].

5.2.3. Benefit of good initial designs

A local search algorithm is applied to this problem to find the optimal server allocation with a total simulation budget size of 800 for the optimization. At the first step, $N = 20$ initial points are sampled according to a specific sampling policy. The sampled points are selected as starting points in ascending order of their high-fidelity performance. When a starting point is selected, the algorithm searches its neighborhood and moves to the best of its neighbor points in each iteration until it reaches a local optimum. Then, the algorithm moves to the next starting point and repeats the local search procedure until the simulation budget is exhausted. The neighborhood of a point $\boldsymbol{x}^0$ is defined as:

$$N(\boldsymbol{x}^0) = \{\boldsymbol{x} | \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{e}_i - \boldsymbol{e}_j, \forall i, j, s.t. ||i - j|| = 1\},$$

where $\boldsymbol{e}_i$ is a six-dimensional vector whose ith element is one and the remaining elements are all zero.

The performance of the applied local search algorithm with starting points sampled by the proposed budget allocation method is compared to that with starting points uniformly sampled from the feasible domain. The proposed sampling method is applied with parameters $N = 20, \alpha = 0.05, n_{1,k} = 2, \forall k, \Delta = 1$ and the feasible solutions are clustered using clustering rule (a) as shown in Figure 10. The experiment is repeated 5000 times and the starting points are re-sampled in each replication.

Figure 11 shows the median of the current optimal objective function value among the 5000 replications, as the simulation budget increases. Compared with uniformly sampling the starting points, using the proposed budget allocation method in the applied local search algorithm can save about 29% of the simulation budget, according to the median value among the 5000 replications, if proper clustering rules are applied.

Figure 12 shows the histogram of the total simulation budget required to reach the global optimum among the 5000 replications. The last column indicates the number of replications in which the global optimum is not found within the given simulation budget. Using the proposed budget allocation method in this case can save the budget required to reach the global optimum. It also reduces the frequency that the global optimum is not found within the given budget.

5.3. A note on application

The proposed method is useful when a clustering strategy, which is developed based on some knowledge on the studied problem, is applicable but the goodness of the groups cannot be guaranteed. The proposed method can identify which groups are good in terms of the quantile of the objective function values and allocate larger evaluation budgets, which are expensive, to better groups.

Using low-fidelity models, i.e., coarse but fast estimators, is an effective way to cluster the feasible solutions when the high-fidelity budget is expensive, e.g., Xu *et al.* (2016). In practice, it may happen that multiple low-fidelity models are available for the studied problem and suggest different clustering strategies (as in Section 5.2). In this case, we can use all the low-fidelity models for the clustering, i.e., cluster the solutions in a transformed multidimensional space in which each dimension is the output of one low-fidelity model. The useless low-fidelity models will be finally ignored by the proposed method as the number of observations increases.

In practice, too many groups may result in wasting budget in the first-stage exploration whereas not enough groups may be not able to separate the good solutions from bad ones. Therefore, when the simulation budget size is low, a more effective way is to use few groups at the beginning. Then, good groups, i.e., groups have higher budget sizes, can be further partitioned during the sampling process.

6. Conclusion

This article proposes a method to allocate budget to existing clusters of the solution space. The goal of the budget allocation is to minimize the quantile of the objective function values among all the sampled solutions. The proposed method can be easily applied, since closed-form formulas are provided. It can be used to generate good solutions in the initial design for optimization problems, in which the efficiency is critical, i.e., a good solution but not the optimum is needed in a short time. It can also be applied to other problems in which budget is allocated to competitive choices in order to minimize the quantile of all sampled values.

The proposed method is tested in designed cases and applications. The numerical results show that the proposed method works well even if the given assumptions are not satisfied. It also shows that using the method as the population initializer can improve the performance of the applied search algorithms in the studied cases, when proper clustering rules are used. In addition, an example of how to cluster solutions using low-fidelity information (e.g., outputs from analytical methods) is provided.

Future works have several directions. One is extending the proposed method to stochastic problems. Currently, it is applied considering the objective function is deterministic, i.e., no randomness is involved in the estimation of the objective function. The second direction is to develop a dynamic clustering strategy while sampling with the proposed method so that the promising groups can be further exploited.

Notes on contributors

Ziwei Lin is a Ph.D. candidate in the Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University, China, and the Department of Mechanical Engineering, Politecnico di Milano, Italy. She holds a bachelor's degree in industrial engineering (2015) from Shanghai Jiao Tong University. Her thesis topic is system performance evaluation and optimization via simulation.

Andrea Matta is a professor of manufacturing at the Department of Mechanical Engineering at Politecnico di Milano, Italy. He currently teaches integrated manufacturing systems and manufacturing. His research area includes analysis and design of manufacturing and health care systems. He is editor-in-chief of *Flexible Services and Manufacturing Journal*.

Shichang Du is a professor of Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University, China. His research interest is about product quality modeling and control in complex manufacturing processes.

ORCID

Ziwei Lin http://orcid.org/0000-0002-9732-6415
Andrea Matta http://orcid.org/0000-0003-3902-2007
Shichang Du http://orcid.org/0000-0003-2408-722X

References

Alatas, B. (2010) Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, **37**(8), 5682–5687.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, **47**(2-3), 235–256.

Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (2013) *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Hoboken, New Jersey.

Burtini, G., Loeppky, J. and Lawrence, R. (2015) A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757*.

Chen, C.-H., Lin, J., Yücesan, E. and Chick, S.E. (2000) Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, **10**(3), 251–270.

Chick, S.E. and Inoue, K. (2001) New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, **49**(5), 732–743.

Dallery, Y., David, R. and Xie, X.-L. (1988) An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE Transactions*, **20**(3), 280–283.

Dallery, Y., David, R. and Xie, X.-L. (1989) Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on Automatic Control*, **34**(9), 943–953.

de Melo, V.V. and Delbem, A.C.B. (2012) Investigating smart sampling as a population initialization method for differential evolution in continuous problems. *Information Sciences*, **193**, 36–53.

Deng, Y., Liu, Y. and Zhou, D. (2015) An improved genetic algorithm with initial population strategy for symmetric TSP. *Mathematical Problems in Engineering*, **2015**, 1–6.

dos Santos Coelho, L. and Mariani, V.C. (2008) Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Systems with Applications*, **34**(3), 1905–1913.

Frazier, P., Powell, W. and Dayanik, S. (2009) The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, **21**(4), 599–613.

Gershwin, S.B. (1987) An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research*, **35**(2), 291–305.

Hong, L.J. and Nelson, B.L. (2005) The tradeoff between sampling and switching: New sequential procedures for indifference-zone selection. *IIE Transactions*, **37**(7), 623–634.

Hu, J., Fu, M.C. and Marcus, S.I. (2007) A model reference adaptive search method for global optimization. *Operations Research*, **55**(3), 549–568.

Johnson, M.E., Moore, L.M. and Ylvisaker, D. (1990) Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, **26**(2), 131–148.

Jones, D.R., Schonlau, M. and Welch, W.J. (1998) Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, **13**(4), 455–492.

Kazimipour, B., Li, X. and Qin, A.K. (2014) A review of population initialization techniques for evolutionary algorithms, in *Proceedings of*

the 2014 *IEEE Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 2585–2592.
Kennedy, J. and Eberhart, R. (1995) Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, IEEE Press, Piscataway, NJ, pp. 1942–1948.
Kerman, J. (2011) A closed-form approximation for the median of the beta distribution. *arXiv preprint arXiv:1111.0433*.
Li, J. and Meerkov, S.M. (2009) *Production Systems Engineering*. Springer, New York.
Linz, D.D., Huang, H. and Zabinsky, Z.B. (2016) A quantile-based nested partition algorithm for black-box functions on a continuous domain, in *Proceedings of the 2016 Winter Simulation Conference*, IEEE Press, Piscataway, NJ, pp. 638–648.
Maaranen, H., Miettinen, K. and Mäkelä, M.M. (2004) Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications*, **47**(12), 1885–1895.
Maaranen, H., Miettinen, K. and Penttinen, A. (2007) On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, **37**(3), 405–436.
MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, University of California Press, Berkeley, Calif, pp. 281–297.
Maillard, O.-A., Munos, R. and Stoltz, G. (2011) A finite-time analysis of multi-armed bandits problems with Kullback-Leibler divergences, in *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19, JMLR Workshop and Conference, Budapest, pp. 497–514.
McKay, A. (1932) Distribution of the coefficient of variation and the extended "t" distribution. *Journal of the Royal Statistical Society*, **95**(4), 695–698.
McKay, M.D., Beckman, R.J. and Conover, W.J. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**(2), 239–245.
Peng, Y., Chen, C.-H., Fu, M.C., Hu, J. and Ryzhov, I.O. (2019) Efficient sampling allocation procedures for optimal quantile selection. Submitted to *Journal on Computing*.
Pezzella, F., Morganti, G. and Ciaschetti, G. (2008) A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, **35**(10), 3202–3212.
Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M. (2007a) A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, **53**(10), 1605–1614.
Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M. (2007b) Quasi-oppositional differential evolution, in Proceedings of the *2007 IEEE Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 2229–2236.
Richards, M. and Ventura, D. (2004) Choosing a starting configuration for particle swarm optimization, in *Proceedings of the IEEE International Joint Conference on Neural Networks,* volume 3, IEEE Press, Piscataway, NJ, pp. 2309–2312.
Rubinstein, R. (1999) The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, **1**(2), 127–190.
Russo, D.J., van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2018) A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning*, **11**(1), 1–96.
Santner, T.J., Williams, B.J. and Notz, W.I. (2003) *The Design and Analysis of Computer Experiments*, Springer Science & Business Media, New York.
Shi, L. and Ólafsson, S. (2000) Nested partitions method for global optimization. *Operations Research*, **48**(3), 390–407.
Shi, L. and Olafsson, S. (2009) *Nested Partitions Method, Theory and Applications*. Springer, New York.
Smith, K. (1918) On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, **12**(1-2), 1–85.
Srinivas, M. and Patnaik, L.M. (1994) Genetic algorithms: A survey. *Computer*, **27**(6), 17–26.
Sutton, R.S., and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
Thompson, W.R. (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, **25**(3–4), 285–294.
Tizhoosh, H.R. (2005) Opposition-based learning: A new scheme for machine intelligence, in *Proceeding of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, volume 1, IEEE Press, Piscataway, NJ, pp. 695–701.
Wiering, M.A. (1999) Explorations in efficient reinforcement learning. PhD Thesis of University of Amsterdam.
Xu, J., Zhang, S., Huang, E., Chen, C.H., Lee, L.H. and Celik, N. (2016) MO2TOS: Multi-fidelity optimization with ordinal transformation and optimal sampling. *Asia-Pacific Journal of Operational Research*, **33**(3), 1–26.
Zhang, G., Gao, L. and Shi, Y. (2011) An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications*, **38**(4), 3563–3573.

Appendices

Appendix A: Proofs

Proof of Proposition 1. Given the group budget size n_k and a threshold τ, for any solution $\boldsymbol{x}_{k,i}$ sampled from group k, the probability that its objective function value is smaller than or equal to τ is $F_k(\tau)$ under Assumption 1. Therefore,

$$E(I_{\{\boldsymbol{x}|y(\boldsymbol{x})\le\tau\}}(\boldsymbol{x}_{k,i})) = F_k(\tau),$$

and the expected number of the solutions in the final sample set $\mathcal{S}(n_1, \dots, n_K, \xi)$ whose objective function values are smaller than or equal to τ is

$$E\left(\sum_{k=1}^{K}\sum_{i=1}^{n_k} I_{\{\boldsymbol{x}|y(\boldsymbol{x})\le\tau\}}(\boldsymbol{x}_{k,i})\right) = \sum_{k=1}^{K}\sum_{i=1}^{n_k} F_k(\tau) = \sum_{k=1}^{K} n_k F_k(\tau).$$

Thus, the problem in expression (2) becomes the problem in expression (3). Proposition 1 is proved. □

Lemma 1. *If a random variable X follows the F-distribution $F(\nu_1, \nu_2)$, the random variable $\nu_1 X$ follows the Chi-square distribution $\chi^2(\nu_1)$ as ν_2 approaches to infinity and*

$$\lim_{\nu_2\to\infty} F(x; \nu_1, \nu_2) = F_{\chi^2}(\nu_1 x; \nu_1),$$

where $F(\cdot; \nu_1, \nu_2)$ is the cdf of the F-distribution with degrees of freedom ν_1, ν_2 and $F_{\chi^2}(\cdot; \nu_1)$ is the cdf of the Chi-square distribution with degree of freedom ν_1.

Lemma 2. *If a random variable X follows the F-distribution $F(\nu_1, \nu_2)$,*

$$\lim_{\nu_1,\nu_2\to\infty} \mathrm{Var}(X) = \lim_{\nu_1,\nu_2\to\infty} \frac{2\nu_2^2(\nu_1+\nu_2-2)}{\nu_1(\nu_2-2)^2(\nu_2-4)} = 0,$$
$$\lim_{\nu_1,\nu_2\to\infty} E(X) = \lim_{\nu_1,\nu_2\to\infty} \frac{\nu_2}{\nu_2-2} = 1,$$

which means that $\lim_{\nu_1,\nu_2\to\infty} F(x; \nu_1, \nu_2) = 0, \forall x < 1$.

Proof of Theorem 2. Since $\hat{\tau} = \hat{\mu}_{\hat{b}} + z_\alpha\hat{\sigma}_{\hat{b}} \le \hat{\mu}_k + z_\alpha\hat{\sigma}_k, \forall k, \alpha < 0.5$ (i.e., $z_\alpha < 0$) and $\hat{\sigma}_k > 0, \forall k$, then,

$$\frac{\hat{\sigma}_{\hat{b}}}{\hat{\mu}_{\hat{b}} - \hat{\tau}} = -\frac{1}{z_\alpha} \text{ and } \frac{\hat{\sigma}_k}{\hat{\mu}_k - \hat{\tau}} \le -\frac{1}{z_\alpha}, \forall k,$$

i.e., the inequality $0 < \hat{c}_k \le \hat{c}_{\hat{b}} = -\frac{1}{z_\alpha}$ always holds as the number of sampling stages s approaches to infinity.

At least one group, denoted as γ, has infinite budget size when the number of sampling stages approaches to infinity, i.e., $\lim_{s\to\infty} n_{s-1,\gamma} \to \infty$. If the series $\{n_{s,\hat{b}}\}$ is bounded, then $\exists n_c | 2 \le n_c < \infty$, s.t. $\lim_{s\to\infty} n_{s-1,\hat{b}} \le n_c$ and using equation (10) we have:

$$1<\frac{1+1/\hat{c}_\gamma^2}{1+1/\hat{c}_{\hat{b}}^2-1/n_c}\le\lim_{s\to\infty}C_{\hat{b},\gamma}=\lim_{s\to\infty}\frac{1+1/\hat{c}_\gamma^2-1/n_{s-1,\gamma}}{1+1/\hat{c}_{\hat{b}}^2-1/n_{s-1,\hat{b}}}\le\frac{1+1/\hat{c}_\gamma^2}{1/2+1/\hat{c}_{\hat{b}}^2}<\infty.$$

Denote $\lim_{s\to\infty}C_{\hat{b},\gamma}$ as $C^\infty_{\hat{b},\gamma}$ and $\lim_{s\to\infty}n_{s-1,\hat{b}}$ as $n^\infty_{\hat{b}}$. According to Lemma 1, we have:

$$0<\lim_{s\to\infty}F(C_{\hat{b},\gamma};n_{s-1,\hat{b}}-1,n_{s-1,\gamma}-1)=F_{\chi^2}\left((n^\infty_{\hat{b}}-1)C^\infty_{\hat{b},\gamma};n^\infty_{\hat{b}}-1\right)<1.$$

Then, using equation (8) we have:

$$\begin{aligned}\lim_{s\to\infty}\frac{1/n_{s,\hat{b}}}{1/n_{s,\gamma}}&=\lim_{s\to\infty}\frac{F(C_{\gamma,\hat{b}};n_{s-1,\gamma}-1,n_{s-1,\hat{b}}-1)}{F(C_{\hat{b},\gamma};n_{s-1,\hat{b}}-1,n_{s-1,\gamma}-1)}\\&=\lim_{s\to\infty}\frac{1-F(C_{\hat{b},\gamma};n_{s-1,\hat{b}}-1,n_{s-1,\gamma}-1)}{F(C_{\hat{b},\gamma};n_{s-1,\hat{b}}-1,n_{s-1,\gamma}-1)}\\&=\frac{1}{F_{\chi^2}\left((n^\infty_{\hat{b}}-1)C^\infty_{\hat{b},\gamma};n^\infty_{\hat{b}}-1\right)}-1>0.\end{aligned}$$

The two series $\{1/n_{s,\hat{b}}\}$ and $\{1/n_{s,\gamma}\}$ are infinitesimals of the same order, which contradicts the assumption that $\{n_{s,\hat{b}}\}$ is bounded. Therefore, $\lim_{s\to\infty}n_{s,\hat{b}}\to\infty$. Similarly, we can prove that $\lim_{s\to\infty}n_{s,k}\to\infty,\forall k\neq\hat{b}$, since $\lim_{s\to\infty}n_{s,\hat{b}}\to\infty$.

The group sample means and group sample variances approach to the real group means and real group variances as the total budget sizes allocated to all the groups approach to infinity. Thus, $\hat{c}_k$ converges to $c_k,\forall k$ and the current best group $\hat{b}$ converges to the real best group b as the number of sampling stages approaches to infinity.

Since $\lim_{s\to\infty}n_{s-1,k}\to\infty,\forall k$ and $\mu_i+z_\alpha\sigma_i\neq\mu_j+z_\alpha\sigma_j,\forall i\neq j$:

$$\lim_{s\to\infty}C_{k,\hat{b}}=\lim_{s\to\infty}\frac{1+1/\hat{c}_{\hat{b}}^2-1/n_{s-1,\hat{b}}}{1+1/\hat{c}_k^2-1/n_{s-1,k}}=\frac{1+1/c_b^2}{1+1/c_k^2}<1,\forall k\neq\hat{b}.$$

According to Lemma 2 we have:

$$\lim_{s\to\infty}F(C_{k,\hat{b}};n_{s-1,k}-1,n_{s-1,\hat{b}}-1)=0,\forall k\neq\hat{b}.$$

Therefore, according to equation (9):

$$\begin{aligned}\lim_{s\to\infty}n_{s,\hat{b}}/N_s&=\lim_{s\to\infty}1/\left(\sum_k\frac{F(C_{k,\hat{b}};n_{s-1,k}-1,n_{s-1,\hat{b}}-1)}{F(C_{\hat{b},k};n_{s-1,\hat{b}}-1,n_{s-1,k}-1)}\right)\\&=\lim_{s\to\infty}1/\left(1+\sum_{k\neq\hat{b}}\frac{F(C_{k,\hat{b}};n_{s-1,k}-1,n_{s-1,\hat{b}}-1)}{1-F(C_{k,\hat{b}};n_{s-1,k}-1,n_{s-1,\hat{b}}-1)}\right)\\&=1.\end{aligned}$$

In summary, when the number of sampling stages approaches to infinity, Algorithm 1 converges to allocating all the budget to the current best $\hat{b}$ while the current best group converges to the real best group b, i.e., $\lim_{s\to\infty}n_{s,b}/N_s=1$. Theorem 2 is proved. □

Lemma 3. *As both the degrees of freedom ν_1, ν_2 approach to infinity, the cdf of the F-distribution has the following properties:* $\lim_{\nu_1,\nu_2\to\infty|\nu_1<\nu_2}F(1,\nu_1,\nu_2)>0.5$; $\lim_{\nu_1,\nu_2\to\infty|\nu_1=\nu_2}F(1,\nu_1,\nu_2)=0.5$; $\lim_{\nu_1,\nu_2\to\infty|\nu_1>\nu_2}F(1,\nu_1,\nu_2)<0.5$.

Proof of Lemma 3. When the argument is equal to one, the cdf of the F-distribution with degrees of freedom ν_1, ν_2 can be transformed to the cdf of the beta distribution as follows:

$$F(1,\nu_1,\nu_2)=F_B\left(\frac{\nu_1}{\nu_1+\nu_2},\frac{\nu_1}{2},\frac{\nu_2}{2}\right),$$

where $F_B(\cdot,\frac{\nu_1}{2},\frac{\nu_2}{2})$ is the cdf of the beta distribution with shape parameters $\frac{\nu_1}{2},\frac{\nu_2}{2}$. An approximate closed-form of the median of the beta distribution with both shape parameters larger than one is provided by Kerman (2011):

$$F_B\left(\frac{\frac{\nu_1}{2}-\frac{1}{3}}{\frac{\nu_1}{2}+\frac{\nu_2}{2}-\frac{2}{3}},\frac{\nu_1}{2},\frac{\nu_2}{2}\right)\approx0.5,$$

and the error rapidly decreases to zero as the shape parameters increase. If $1<\nu_1<\nu_2$:

$$\frac{\nu_1}{\nu_1+\nu_2}-\frac{\frac{\nu_1}{2}-\frac{1}{3}}{\frac{\nu_1}{2}+\frac{\nu_2}{2}-\frac{2}{3}}=\frac{2(\nu_2-\nu_1)}{(\nu_1+\nu_2)(3\nu_1+3\nu_2-4)}>0.$$

Then,

$$\lim_{\nu_1,\nu_2\to\infty|\nu_1<\nu_2}F(1,\nu_1,\nu_2)>\lim_{\nu_1,\nu_2\to\infty}F_B\left(\frac{\frac{\nu_1}{2}-\frac{1}{3}}{\frac{\nu_1}{2}+\frac{\nu_2}{2}-\frac{2}{3}},\frac{\nu_1}{2},\frac{\nu_2}{2}\right)=0.5.$$

Similarly,

$$\lim_{\nu_1,\nu_2\to\infty|\nu_1=\nu_2}F(1,\nu_1,\nu_2)=0.5\text{ and }\lim_{\nu_1,\nu_2\to\infty|\nu_1>\nu_2}F(1,\nu_1,\nu_2)<0.5.$$

Lemma 3 is proved. □

Proof of Theorem 3. It is proved in the proof of Theorem 2 that, $\lim_{s\to\infty}n_{s-1,k}\to\infty,\forall k$, which means $\hat{c}_k/\hat{c}_{\hat{b}},\forall k\neq\hat{b}$ approach to one since $\mu_k=\mu_b,\sigma_k=\sigma_b,\forall k$. Therefore, using equation (10) we have:

$$\lim_{s\to\infty}C_{\hat{b},k}=\lim_{s\to\infty}\frac{1+1/\hat{c}_k^2-1/n_{s-1,k}}{1+1/\hat{c}_{\hat{b}}^2-1/n_{s-1,\hat{b}}}=1,$$

and according to equation (8):

$$\begin{aligned}\lim_{s\to\infty}\frac{n_{s,k}}{n_{s,\hat{b}}}&=\lim_{s\to\infty}\frac{F(C_{k,\hat{b}};n_{s-1,k}-1,n_{s-1,\hat{b}}-1)}{F(C_{\hat{b},k};n_{s-1,\hat{b}}-1,n_{s-1,k}-1)}\\&=\lim_{s\to\infty}\frac{1}{F(C_{\hat{b},k};n_{s-1,\hat{b}}-1,n_{s-1,k}-1)}-1\\&=\lim_{n_{s-1,k},n_{s-1,\hat{b}}\to\infty}\frac{1}{F(1;n_{s-1,\hat{b}}-1,n_{s-1,k}-1)}-1,\forall k\neq\hat{b}.\end{aligned}$$

Combined with Lemma 3, if

$$\lim_{s\to\infty}\frac{n_{s-1,k}}{n_{s-1,\hat{b}}}<1,\ \lim_{s\to\infty}\frac{n_{s,k}}{n_{s,\hat{b}}}>1;$$

if

$$\lim_{s\to\infty}\frac{n_{s-1,k}}{n_{s-1,\hat{b}}}>1,\ \lim_{s\to\infty}\frac{n_{s,k}}{n_{s,\hat{b}}}<1;$$

if

$$\lim_{s\to\infty}\frac{n_{s-1,k}}{n_{s-1,\hat{b}}}=1,\ \lim_{s\to\infty}\frac{n_{s,k}}{n_{s,\hat{b}}}=1.$$

In addition,

$$\lim_{s\to\infty}\frac{n_{s-1,k}}{n_{s-1,\hat{b}}}=\lim_{s\to\infty}\frac{n_{s-1,k}}{n_{s-1,\hat{b}}+\Delta}\le\lim_{s\to\infty}\frac{n_{s,k}}{n_{s,\hat{b}}}\le\lim_{s\to\infty}\frac{n_{s-1,k}+\Delta}{n_{s-1,\hat{b}}}=\lim_{s\to\infty}\frac{n_{s-1,k}}{n_{s-1,\hat{b}}}.$$

Therefore,

$$\lim_{s\to\infty}\frac{n_{s,k}}{n_{s,\hat{b}}}=1,\forall k\neq\hat{b}.$$

Theorem 3 is proved. □

Appendix B: Numerical analysis on approximation (3)

Numerical experiments are executed to support the goodness of the problem approximation in expression (3). Given a problem of expression (1) and its approximation in expression (3) (i.e., fix N,α,K, values and $F_k,\forall k$ distributions), for any solution $\boldsymbol{n}_i$ (i.e., an $n_k,\forall k$ combination), its ranking according to the objective function value in expression (1) is denoted as $r_i^{(1)}$ and its ranking according to the objective function value in expression (3) is denoted as $r_i^{(3)}$. The main idea is to show that the sequences $\{r_i^{(1)}\}$ and $\{r_i^{(3)}\}$ are highly correlated. If the correlation coefficient is equal to one, the solution rankings in expression (1) and in expression (3) are the same, which includes that the optimal solution in expression (3) is the optimal solution in expression (1). If the correlation coefficient is close to one, it means that the optimal solution in expression (3) may not be the optimal solution in expression (1), but it still has a good performance in expression (1), i.e., it is a near-optimal solution.

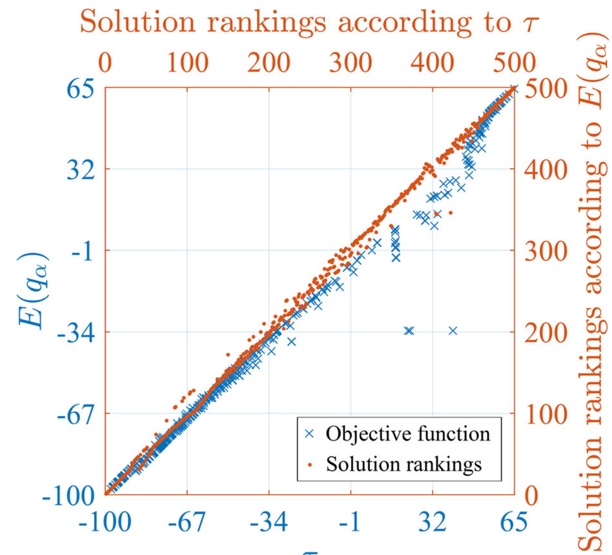

Figure 13. The scatter plots of the objective function values and the solution rankings in a problem. $N = 50, K = 5, \alpha = 0.1427$.

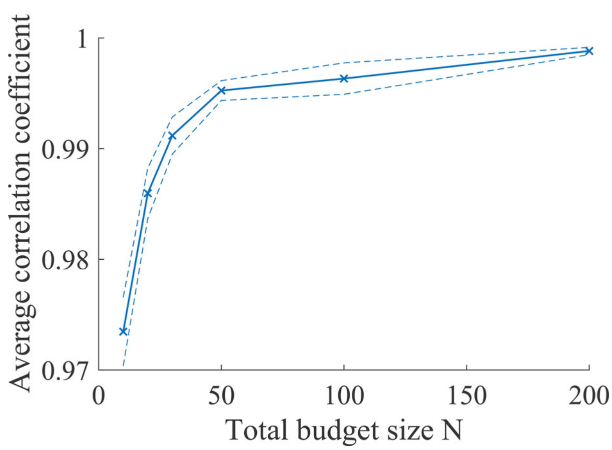

Figure 14. The average correlation coefficient of $\{r_i^{(1)}\}$ and $\{r_i^{(3)}\}$ and its 95% confidence intervals.

An example, in which 500 solutions are randomly sampled, is presented in Figure 13 with the objective function values and the solution rankings of the sampled solutions. In this problem, $N = 50, K = 5, \alpha = 0.1427$ and the group distributions are U(40.59,218.21), N(-73.71,796.77), Tria(19.93,136.46,155.35) and two piecewise linear distributions with two pieces, respectively. The expectation of the q_α, i.e., the objective function value of the original problem, is estimated using the Monte Carlo method with 50 000 replications. It is possible to see, in Figure 13, that the objective function values of the original problem (i.e., $E(q_\alpha(\mathcal{S}(n_1, ..., n_K, \xi)))$ and the approximated problem (i.e., $\tau : \sum_{k=1}^{K} n_k F_k(\tau) = \alpha N$) are highly correlated, especially for good solutions, i.e., the points with smaller $E(q_\alpha)$ values.

Figure 14 presents the average of the correlation coefficient between $r_i^{(1)}$ and $r_i^{(3)}$. Every point is obtained with 500 problems. In each problem, $\alpha, K, F_k, \forall k$ are randomly selected and 500 different solutions are sampled. The K value is chosen from 2 to 10 and the α value varies from 0.0001 to 0.45. The F_k is randomly selected from Normal distribution $N(\mu \in [-100, 100], \sigma^2 \in [0, 2500])$, Uniform distribution $U(a, b; -250 \le a < b \le 250)$, Triangular distribution $Tria(a, b, c; -250 \le a < b < c \le 250)$, and piecewise linear distribution composed by two to five pieces within [-250, 250].

As shown in Figure 14, a significant correlation is observed even though the N value is small (the average correlation coefficient is larger than 0.97 even if $N = 10$). Furthermore, the average correlation coefficient converges to one rapidly as the total budget size N increases, which means that the solution rankings in the original problem and in the approximated problem tend to be the same as the total budget size increases.

In addition, 400 randomly selected problems are tested with N varying from 10 to 30, K varying from three to five, the other settings the same as above and the optimal solution identified through enumeration. In 92.75% of the tested problems, the optimal solution is the solution in Theorem 1. In the rest of the 29 problems, the minimal group quantile and the second minimal group quantile are quite close and the optimal solution is that some budget are allocated to the group with the second minimal quantile. In these 29 problems, the mean absolute relative difference between the objective function value of the optimal solution and that of the solution in Theorem 1 is only 2.9% (including the sampling noise). Therefore, expression (3) is a reasonable approximation of expression (1).

Table 5. The categories of total buffer capacity ranges.

Number of groups	*Categories*
2	[36,400]
10	[36,108], [109,181], [182,254], [255,327], [328,400]
16	[36,81], [82,126], [127,172], [173,218], [219,263], [264,309], [310,354], [355,400]

Appendix C: Clustering rule for BAP

This Appendix describes in detail how the feasible solutions are clustered in Section 5.1 and the algorithm used to sample solutions.

First, two clusters are generated. The buffer capacity of the solutions in the first cluster is correlated to the average availability of the corresponding adjacent machines, whereas the solutions in the second cluster do not have any specific pattern. Then, each cluster is further divided into more groups based on the range to which the total buffer capacity belongs. Table 5 shows the categories of total buffer capacity ranges when different numbers of groups are required.

During the sampling phase, the value of the total buffer capacity is firstly sampled, then the combination of each buffer slot is randomly generated. The sampling probability of the total buffer capacity is proportional to the corresponding total number of combinations of the buffer slots, i.e., high values have high probabilities to be sampled.

Algorithm 2 describes how the solutions, in which the buffer capacity is correlated to the machine availability, are generated in Section 5.1. For each new solution, the m value, which controls the ratio of the highest buffer capacity to the lowest buffer capacity, is randomly selected from [1,16]. The total buffer capacity is sampled considering the number of combinations of buffer slots. The capacity of each buffer is determined in proportion to the weight calculated based on the machine availability. Finally, noise is added to increase the randomness of the sampling.

Algorithm 2 Buffer capacity sampling algorithm

n ← Number of solutions to be sampled
d ← Number of buffers in each solutions
for $i = 1 \cdots n$ **do**
  Sample the total buffer capacity B_i^{tot}.
  Randomly select $m_i \in [1, 16]$.
  for $j = 1 \cdots d$ **do**
    Calculate the reciprocal of the mean availability of the adjacent machines: $c_{i,j} = \frac{2}{e_j + e_{j+1}}$.
    Calculate the initial weight of each buffer: $w_{i,j}^0 = c_{i,j} - \frac{m_i \min_j\{c_{i,j}\} - \max_j\{c_{i,j}\}}{m_i - 1}$.
    Add noise to the weight: $w_{i,j} = (1 + u_{i,j}) w_{i,j}^0$ where $u_{i,j}$ is randomly selected from [-0.1,0.1].
    Determine the capacity of buffer j: $x_{i,j} = \lfloor B_i^{tot} * w_{i,j} / \sum_j w_{i,j} \rfloor$ and the rest buffer capacity is allocated arbitrarily.
  end for
end for