European Journal of Operational Research 320 (2025) 628–641

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Stochastics and Statistics

Reference alternatives based knockout-tournament procedure for ranking and selection

Ying Zhong [a], Jianzhong Du [b], Deng-Feng Li [a], Zhaolin Hu [c,*]

[a] School of Management and Economics, University of Electronic Science and Technology of China, Chengdu, 611731, China
[b] International Institute of Finance, School of Management, University of Science and Technology of China, Hefei, 230026, China
[c] School of Economics and Management, Tongji University, Shanghai, 200092, China

ARTICLE INFO

Keywords:
Simulation
Ranking and selection
Improved knockout-tournament
Parallel computing
Resource allocation

ABSTRACT

The knockout-tournament ($\mathcal{KT}$) procedure is an efficient parallel procedure recently developed to solve large-scale ranking and selection (R&S) problems. The procedure adopts a selection structure which is commonly used in many sports tournaments, and eliminates alternatives by conducting "matches" between paired alternatives round-by-round. In this paper, to further improve the procedure's performance in solving large-scale problems, we propose a major modification of the procedure. Specifically, in each round of the selection, before pairing the surviving alternatives and conducting the matches, we first choose an alternative as the reference alternative and then add the reference alternative to each match. We call the new procedure Procedure i-$\mathcal{KT}$, where i-$\mathcal{KT}$ stands for "improved knockout-tournament". We show that by carefully choosing the reference alternative and designing the pairing scheme for the remaining surviving alternatives in each round of the selection, Procedure i-$\mathcal{KT}$ can achieve significant improvements on both the average sample size required in each match and the total number of matches required during the entire selection process. In the meantime, we demonstrate that after the modifications, Procedure i-$\mathcal{KT}$ still fits parallel computing environments well. We compare Procedure i-$\mathcal{KT}$ with various procedures on different test examples and numerically justify our theoretical analysis.

1. Introduction

In a decision making process, one often faces the problem of comparing a finite set of k alternatives based on certain performance measures. In this paper, we consider a comparison problem with the goal of selecting the best alternative, where the best is defined to have the largest (or smallest) mean performance. In particular, the mean performance of each alternative is unknown, and can only be estimated by running simulation experiments (taking samples) and observing the random outputs. Such a selection of the best problem is also called the ranking-and-selection (R&S) problem in the literature. The problem was first proposed by Bechhofer (1954) and has since then been studied extensively in the areas of statistics and operations research, see Kim and Nelson (2006a), Chen et al. (2015), and Hong et al. (2021) for comprehensive reviews.

Because the outputs from each alternative are random, to obtain a good estimate of the best alternative, one needs to sample every alternative many times. A key issue in solving the R&S problem is to design statistical procedures that determine the number of observations each alternative should take so that the true best alternative can be selected with high probability. Typically, it is impossible for a procedure to achieve a *probability of correct selection* (PCS) of one for any finite sampling budget. To avoid such difficulty, two types of formulations have been proposed (Hunter & Nelson, 2017). For one type of formulation, procedures are designed to deliver a PCS higher than a pre-determined level upon stopping. These procedures are called the fixed-precision procedures (*e.g.*, Andradóttir and Lee 2021, Bechhofer 1954, Chen et al. 2023, Frazier 2014, Hong 2006, Kim and Nelson 2001, Lee et al. 2018, Paulson 1964, Rinott 1978, Yoon and Bekker 2019). For the other type of formulation, procedures are designed to deliver a PCS as high as possible given that the total sampling budget is fixed. These procedures are called the fixed-budget procedures (*e.g.*, Chen et al. 2000, Chick and Inoue 2001, Frazier et al. 2008, Glynn and Juneja 2004, Groves and Branke 2019). Essentially, fixed-precision procedures are designed to conduct *reliable* selections, and fixed-budget procedures are designed

[*] Corresponding author.
E-mail addresses: yzhong4@uestc.edu.cn (Y. Zhong), jianzhodu2-c@my.cityu.edu.hk (J. Du), lidengfeng@uestc.edu.cn (D.-F. Li), russell@tongji.edu.cn (Z. Hu).

https://doi.org/10.1016/j.ejor.2024.08.031
Received 7 February 2024; Accepted 30 August 2024
Available online 7 September 2024

to conduct *best-effort* selections. In this paper, we focus on designing fixed-precision procedures.

Thus far, various fixed-precision procedures have been developed. Traditionally, fixed-precision procedures are often designed and implemented in single-processor computing environments. Due to the limited computing power of a single processor, the procedures may only be used to solve small-scale problems (*e.g.*, the number of alternatives $k \leq 1,000$). In recent years, with the growing needs in practice, developing procedures to solve large-scale problems in parallel computing environments has become an emerging research topic (Hunter & Nelson, 2017). However, developing parallel procedures is non-trivial. In practice, the performances of different procedures are often examined and compared by measuring the wall-clock times[1] that the procedures take to solve different problems. Unlike single-processor computing environments, in parallel computing environments, besides the computational efforts spent on generating observations, communications and coordinations among the processors, unbalanced workloads across different processors, and non-parallelizable calculations can greatly affect the wall-clock time of a procedure. As pointed out by Luo et al. (2015), directly implementing some traditional procedures in parallel computing environments may encounter problems from both practical and theoretical aspects. In view of this, several prominent parallel procedures have been proposed (*e.g.*, Luo et al. 2015, Ni et al. 2017, Pei et al. 2022, Zhong and Hong 2022, Zhong et al. 2022). Among these procedures, the knockout-tournament ($\mathcal{KT}$) procedure developed by Zhong and Hong (2022) is a very competitive one.

Procedure $\mathcal{KT}$ adopts a selection structure which is commonly used in many sports tournaments and sequentially eliminates inferior alternatives in a round-wise manner. In each round of the selection, the procedure first pairs the alternatives that are still in contention. For each pair of alternatives, a "match" will be conducted by using an existing fixed-precision procedure, namely Procedure $\mathcal{KN}$ proposed by Kim and Nelson (2001). The winner advances to the next round of the selection, and the other one is eliminated. The procedure continues until there is only one alternative left and selects that alternative as the best. As demonstrated in Zhong and Hong (2022), the selection structure of Procedure $\mathcal{KT}$ is well-suited for parallel computing environments. Specifically, while running in parallel computing environments, the procedure requires almost no communications and coordinations among the processors, the workloads of different processors can be easily balanced, and the proportion of non-parallelizable calculations in the procedure is minimized. More importantly, the authors prove that, as the number of alternatives k increases, the expected total sample size of Procedure $\mathcal{KT}$ can grow linearly in k which is in a lower order than those of many existing procedures and reaches the optimal order. It establishes a theoretical foundation for the procedure to solve large-scale problems and suggests that the procedure tends to use fewer observations to identify the best alternative than many existing procedures do when the problem size, *i.e.*, the number of alternatives, is sufficiently large. Due to these advantages, Procedure $\mathcal{KT}$ successfully demonstrates its ability to solve large-scale problems in parallel computing environments and is capable of solving problems with more than one million alternatives on commercial clouds.

However, there are some potential issues for Procedure $\mathcal{KT}$. Since the selection structure of Procedure $\mathcal{KT}$ fits parallel computing environments well, the performance of Procedure $\mathcal{KT}$, *i.e.*, the wall-clock time, highly depends on the computational time spent on generating observations. Even though Procedure $\mathcal{KT}$ shows its superiority in solving large-scale problems from the aspect of the growth rate of the expected total sample size with respect to k, the finite-time performance of Procedure $\mathcal{KT}$ may not be satisfactory in terms of the total sample size until the number of alternatives is more than 10^5 or even 10^6. The main reason is that, for Procedure $\mathcal{KT}$, the average sample size of a match and the total number of matches are two major factors affecting the total sample size of the procedure, yet the procedure has some potential drawbacks in these two aspects. First, in Procedure $\mathcal{KT}$, only two alternatives participate in a match. As the selection proceeds, in many matches, the two alternatives may be clearly inferior alternatives but have similar or even equal means. It poses great challenges to the procedure trying to distinguish them. In this situation, the procedure needs to take a very long time and a large number of observations to conduct the matches. It significantly increases the average sample size of a match. Second, for Procedure $\mathcal{KT}$, to correctly select the best alternative, we only need to ensure that the best alternative wins the matches it participates in. During the selection process, most matches are unimportant matches, *i.e.*, the ones that do not contain the best alternative. In fact, in these matches, whichever alternatives win has little impact on the PCS of the procedure. However, the procedure still needs to spend a large amount of time on finding the winners of these matches, and wastes a lot of sampling efforts. It further dampens the practical performance of the procedure. These issues greatly limit the use of Procedure $\mathcal{KT}$ in practice. Therefore, it is of both theoretical and practical interests to ask whether we can make modifications of Procedure $\mathcal{KT}$ to improve its performance from these two aspects, but keep its nature as a parallel-friendly procedure and theoretical superiority, *i.e.*, linear growth rate on the expected total sample size as k increases, intact.

In this paper, to address the aforementioned issues, we propose a new fixed-precision parallel procedure. The new procedure adopts the same selection structure as that of Procedure $\mathcal{KT}$. Meanwhile, compared with Procedure $\mathcal{KT}$, one distinct feature of our procedure is that in each round of the selection, before pairing the alternatives that are still in contention, based on the sampling information collected in previous rounds of selections, the procedure would first choose an alternative as the reference alternative. Then, the procedure pairs the remaining alternatives and adds the reference alternative to each match. In each round, the reference alternative can only be eliminated in one designated match. For any other match, if the reference alternative wins the match, no alternative can advance to the next round of the selection. We call the new procedure Procedure i-$\mathcal{KT}$, where i-$\mathcal{KT}$ stands for "improved knockout-tournament". We prove that similar to Procedure $\mathcal{KT}$, Procedure i-$\mathcal{KT}$ still satisfies the PCS guarantee, *i.e.*, delivers a PCS higher than a pre-determined level upon stopping. Moreover, we show that due to the introduction of the reference alternative, the performance of Procedure i-$\mathcal{KT}$ can be significantly improved. To understand why, we would like notice that, for Procedure i-$\mathcal{KT}$, the presence of the reference alternative in every match is mainly for the purpose of speeding up the comparisons and eliminating additional alternatives. We prove that, in each round of the selection, choosing a reference alternative with a large mean enables us to quickly finish a variety of matches with a few observations. As a result, the average sample size of a match in Procedure i-$\mathcal{KT}$ tends to be smaller than that of Procedure $\mathcal{KT}$. Furthermore, from the aspect of reducing the total number of matches, we prove that, if the reference alternative has a relatively large mean, then by carefully designing a pairing scheme for the surviving alternatives, we may achieve a considerably high chance of eliminating the other two alternatives all at once if they are both inferior in a match. As a result, compared with Procedure $\mathcal{KT}$, in Procedure i-$\mathcal{KT}$, many fewer alternatives can survive in each round of the selection. It suggests that the procedure no longer needs to conduct many unnecessary matches in the following rounds of selections, and the total number of matches of the procedure is much smaller than that of Procedure $\mathcal{KT}$.

We are aware that to implement Procedure i-$\mathcal{KT}$ in parallel computing environments, choosing the reference alternative and adding it to every match are non-parallelizable calculations and require additional communications and coordinations among the processors. Furthermore, because a random number of alternatives survive in each round of

[1] Wall-clock time measures the real time from the start to the end of a procedure.

the selection, re-balancing the workloads of different processors in a round-wise manner is also necessary. These operations may affect the utilizations of processors and diminish the benefit of parallel computing. However, such impact is limited. We prove that for Procedure *i*-$\mathcal{KT}$, if the reference alternative is properly chosen and the tolerable probability of incorrect selection is set to be relatively small (which is often the case), the procedure tends to identify the best alternative in two rounds of selections. It suggests that the frequency of conducting these operations remains low and thus the impact is tolerable.

Lastly, it is worth noting that, to implement Procedure $\mathcal{KT}$ in parallel computing environments and fully utilize the processors, when the number of surviving alternatives falls below the number of processors in the parallel computing environments, instead of continuing to conduct matches among the alternatives, one needs to adopt another procedure, namely the Rinott's procedure proposed by Rinott (1978), to compare the surviving alternatives all at once. As a result, even though the expected total sample size of Procedure $\mathcal{KT}$ grows linearly in k, while implementing the procedure in parallel computing environments, the constant on the growth rate additionally depends on the number of processors in parallel computing environments. The more processors, the more observations the procedure needs to identify the best alternative. In this situation, the marginal effect of adding more processors on reducing the wall-clock time decreases. In this paper, we prove that, while implementing Procedure *i*-$\mathcal{KT}$ in parallel computing environments, in each round of the selection, there is always a chance that the reference alternative eliminates all the other surviving alternatives and terminates the selection. In this situation, not only the expected total sample size of the procedure can grow linearly in k, but also the constant on the growth rate is immune to the change of the number of processors in parallel computing environments and thus much smaller than that of Procedure $\mathcal{KT}$. With comprehensive numerical studies, we show that our procedure indeed performs much better than Procedure $\mathcal{KT}$ does across different types of problems.

The rest of the paper is organized as follows. In Section 2, we provide a mathematical formulation for the problem we aim to solve. In Section 3, we propose Procedure *i*-$\mathcal{KT}$ and verify its statistical validity. In Section 4, we analyze the average sample size of a match and total number of matches for Procedure *i*-$\mathcal{KT}$, followed by discussions on the parallel implementation of the procedure in Section 5. Numerical results and concluding remarks are presented in Sections 6 and 7 respectively.

2. Problem formulation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the underlying probability space. We denote by $\mathcal{K} = \{1, 2, \ldots, k\}$ the set of alternatives that are in contention at the very beginning of the selection process. Let $X_{i,\ell} = \mu_i + \varepsilon_{i,\ell}$ be the ℓ-th observation from alternative $i \in \mathcal{K}$, where μ_i is the mean of alternative i and $\varepsilon_{i,\ell}$ is the random error that follows a normal distribution with mean 0 and variance σ_i^2. Define σ_{ij}^2 as the variance of the difference between any two alternatives i and j, *i.e.*, $\sigma_{ij}^2 = \mathrm{Var}(\varepsilon_{i,\ell} - \varepsilon_{j,\ell})$. In this paper, we assume that variances σ_i^2 for $i = 1, 2, \ldots, k$ and σ_{ij}^2 for $i, j \in \mathcal{K}$ with $i \neq j$ are unknown. Further assume that the observations generated from the same alternative are independent with each other, i.e, for each $i \in \mathcal{K}$, $\varepsilon_{i,1}, \varepsilon_{i,2}, \ldots$, are independent. Because common random numbers (CRNs) are allowed in our procedure, we do not require the observations generated from different alternatives to be independent. Specifically, fixing ℓ, random errors $\varepsilon_{i,\ell}$ and $\varepsilon_{j,\ell}$ can be correlated for all $i, j \in \mathcal{K}$ with $i \neq j$. Without loss of generality, throughout this paper, we assume that the means of the alternatives are in an ascending order, *i.e.*, $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$, and alternative k is the true best alternative. We need to identify which alternative is alternative k.

Without any other assumptions on the configuration of the means, the goal of selecting alternative k can be stringent, because in practice the mean difference between the best and the second-best alternatives can be arbitrarily small. In this situation, the best alternative may not be identifiable in a finite number of observations. To avoid such difficulty, some compromises have to be made. Typically, there are two ways to address this issue. One way is to assume that there exists a minimal gap $\delta > 0$ between the means of the best and the second-best alternatives, where δ is the user-specified indifference-zone (IZ) parameter and the smallest difference the user feels worth detecting. Once the IZ assumption is made, it implies that $\mu_k > \mu_{k-1}$ and alternative k is the unique best alternative. Under the IZ assumption, procedures are designed to ensure that when the selection stops, the procedures can select alternative k with probability at least $1 - \alpha$, i.e, if $\mu_k - \mu_{k-1} \geq \delta$,

$$\mathbb{P}(\text{select alternative } k) \geq 1 - \alpha,$$

where α $(0 < \alpha < 1 - 1/k)$ is the tolerable probability of incorrect selection (PICS) and set by the user. This type of statistical guarantee is called the PCS guarantee. Instead of making assumptions on the configuration of the means, the other way is to soften the original goal of selecting the best alternative. For the procedures developed in this way, they are required to select an alternative within δ to the best with probability at least $1-\alpha$ upon stopping. This type of statistical guarantee is called the probability of good selection (PGS) guarantee. The PGS guarantee is always stronger than the PCS guarantee. As pointed out by Hunter and Nelson (2017), the procedure with the PGS guarantee always satisfies the PCS guarantee, but not vice versa. In this paper, our objective is to devise a procedure that only satisfies the PCS guarantee. However, with similar techniques discussed in Zhong and Hong (2022), one can modify the procedure to satisfy the PGS guarantee.

3. The procedure

In this section, we first briefly review Procedure $\mathcal{KT}$. Then, we present our procedure and verify its statistical validity.

3.1. Procedure $\mathcal{KT}$

Procedure $\mathcal{KT}$ is a parallel procedure recently developed by Zhong and Hong (2022) and implemented in a round-wise manner. In each round of the selection, the procedure first pairs the alternatives that are still in contention. Then for each pair of alternatives, an existing procedure with the PCS guarantee is employed to conduct a match between the two alternatives. The winner proceeds to the next round of the selection, and the other one is eliminated. Procedure $\mathcal{KT}$ continues until there is only one alternative left and selects that alternative as the best. The procedure used to conduct the matches should ensure that in the rth round of the selection, falsely eliminating alternative k happens with probability less than $\alpha_r = \alpha/2^r$. By doing so, the probability that alternative k is falsely eliminated during the entire selection process can be upper bounded by α, and thus Procedure $\mathcal{KT}$ can satisfy the PCS guarantee. In Fig. 1, we give an illustration of the selection of eight alternatives by Procedure $\mathcal{KT}$. As demonstrated in Zhong and Hong (2022), if the procedure used to conduct the matches is properly chosen, the growth rate of the expected total sample size of Procedure $\mathcal{KT}$ can be upper bounded by the order of k which is proven to be the optimal order. In this situation, Procedure $\mathcal{KT}$ tends to outperform many existing procedures in terms of the total sample size as long as k is large enough.

Procedure $\mathcal{KT}$ has a relatively simple selection structure. One advantage of such a selection structure is that the procedure can be easily implemented in parallel computing environments and works well. For example, to implement the procedure in a parallel computing environment with m processors, one only needs to divide the alternatives into m subsets at the very beginning of the selection process, and assign each processor with one subset of alternatives. Then, following the selection structure of Procedure $\mathcal{KT}$, every processor can independently conduct the selection until it finds the local best alternative within the

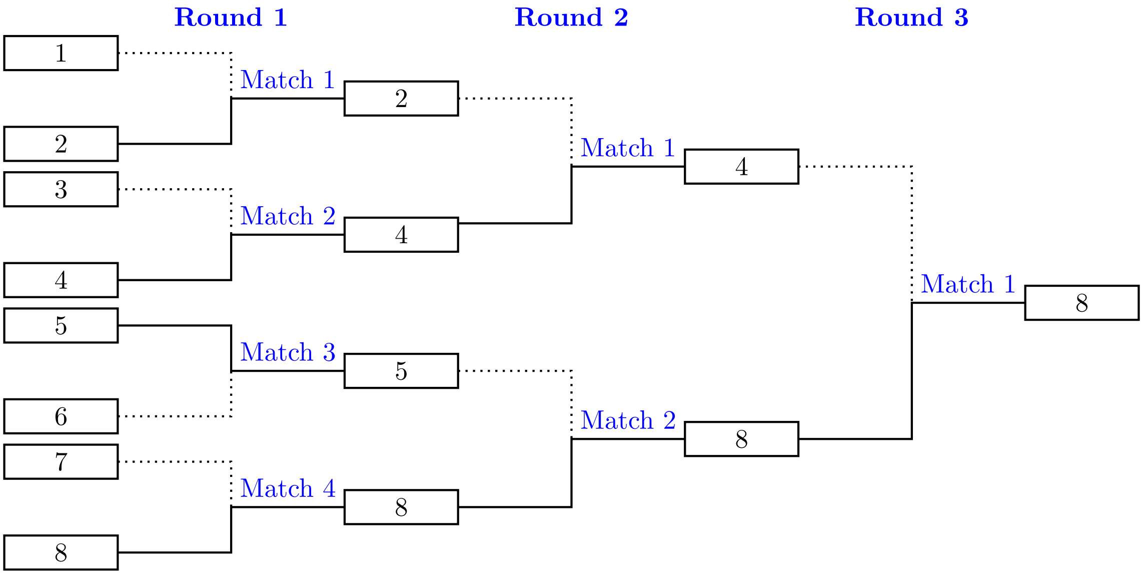


Fig. 1. An illustration of the selection of eight alternatives by Procedure $\mathcal{KT}$.

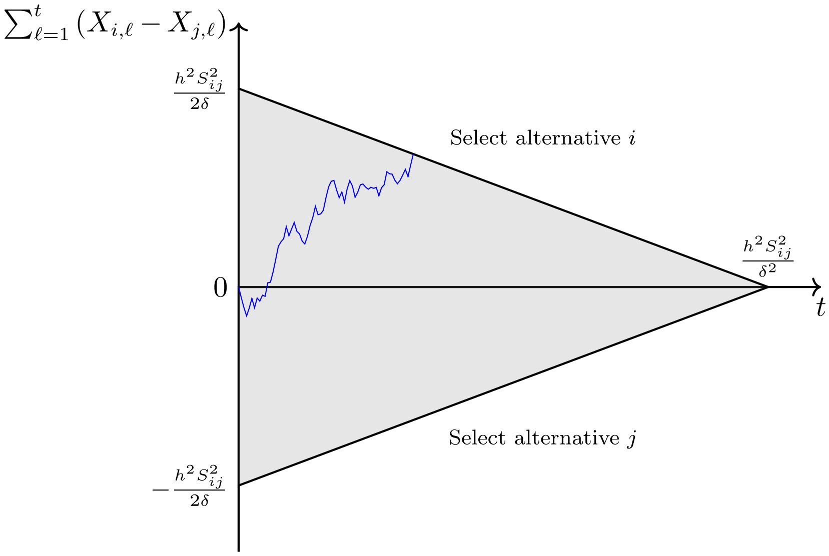
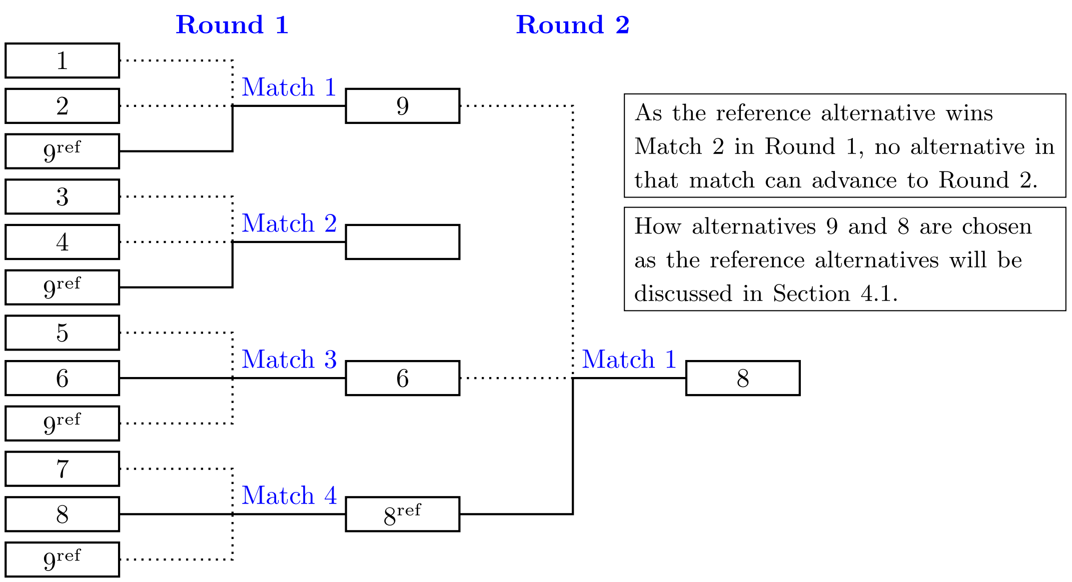

$\sum_{\ell=1}^{t}(X_{i,\ell}-X_{j,\ell})$
$\frac{h^2S_{ij}^2}{2\delta}$
Select alternative i
$\frac{h^2S_{ij}^2}{\delta^2}$
0
t
Select alternative j
$-\frac{h^2S_{ij}^2}{2\delta}$

Fig. 2. An illustration of the comparisons between two alternatives i and j in Procedure $\mathcal{KN}$.

subset as assigned. After every processor finds its local best alternative, if one keeps using the round-wise selection structure to find the final best alternative, in the following rounds of selections, there are more processors in the parallel computing environments than the surviving alternatives. As a result, some processors may stay idle for a relatively long time which may affect the utilizations of the processors. To address this issue, (Zhong & Hong, 2022) suggest using another procedure with the PCS guarantee, namely the Rinott's procedure (Rinott, 1978), to compare these local best alternatives all at once. It requires every processor to generate some additional observations from each local best alternative and calculate the sample mean of each local best alternative based on these additional observations. Then, we can select the one with the largest sample mean as the final best alternative. With such an implementation scheme, the workloads of different processors are roughly the same. Furthermore, only two rounds of communications and coordinations among the processors are needed, and there are almost no non-parallelizable calculations in the procedure.

In Procedure $\mathcal{KT}$, Procedure $\mathcal{KN}$ (Kim & Nelson, 2001) is employed to conduct the matches. Procedure $\mathcal{KN}$ is a well-known and efficient procedure with the PCS guarantee. Notice that we can directly use Procedure $\mathcal{KN}$ to select the best alternative from set $\mathcal{K}$ with probability at least $1-\alpha$. If that is the case, Procedure $\mathcal{KN}$ works as follows. Before the selection starts, Procedure $\mathcal{KN}$ would collect some initial observations from each alternative $i\in\mathcal{K}$ to estimate the variance of the difference between every pair of alternatives, i.e, first-stage sampling. Based on the estimated sample variance, Procedure $\mathcal{KN}$ constructs a continuation region for the partial sum of difference process between each pair of alternatives. The partial sum of difference process between any two alternatives i and j is defined as $\{Z_{ij}(t)=\sum_{\ell=1}^{t}(X_{i,\ell}-X_{j,\ell}): t=1,2,\dots,\}$. After the first-stage sampling, Procedure $\mathcal{KN}$ proceeds in iterations. In each iteration, the procedure first collects an additional observation from each surviving alternative. Then, the procedure updates the partial sum of difference between every pair of alternatives. Once the partial sum of difference exits the continuation region, one alternative can be eliminated accordingly. The procedure stops when only one alternative is left and chooses that alternative as the best. Typically, the continuation regions are constructed to ensure that for the comparisons between the best alternative and any other alternative in set $\mathcal{K}$, falsely eliminating the best alternative happens with probability less than $\alpha/(|\mathcal{K}|-1)$, where $|\mathcal{K}|$ is the cardinality of set $\mathcal{K}$.



As the reference alternative wins Match 2 in Round 1, no alternative in that match can advance to Round 2.

How alternatives 9 and 8 are chosen as the reference alternatives will be discussed in Section 4.1.

Fig. 3. An illustration of the selection of nine alternatives by Procedure *i*-$\mathcal{KT}$, where alternatives 9 and 8 are the reference alternatives in the 1st and 2nd rounds of selections respectively.

Then, the overall probability of falsely eliminating the best alternative can be upper bounded by α. For example, let n_0 denote the first-stage sample size, S_{ij}^2 be the estimated sample variance of the difference between alternatives i and j based on the first-stage sampling, *i.e.*, $S_{ij}^2 = \sum_{\ell=1}^{n_0}\left[X_{i,\ell} - X_{j,\ell} - (\sum_{\ell=1}^{n_0} X_{i,\ell}/n_0 - \sum_{\ell=1}^{n_0} X_{j,\ell}/n_0)\right]/(n_0-1)$, and $\Lambda(\gamma)$, where $\gamma > 0$, be the region formed by lines $\gamma - \delta t/2$ and $-\gamma + \delta t/2$ for $t \geq 0$. Then, $\Lambda(h^2 S_{ij}^2/(2\delta))$ is the continuation region for the comparisons between alternatives i and j where $h^2 = (n_0-1)[(2\alpha/(|\mathcal{K}|-1))^{-2/(n_0-1)} - 1]$. In Fig. 2, we give an illustration of the comparisons between two alternatives i and j in Procedure $\mathcal{KN}$, where the gray area is the continuation region $\Lambda\left(h^2 S_{ij}^2/(2\delta)\right)$.

3.2. Procedure i-$\mathcal{KT}$

In this subsection, we present Procedure *i*-$\mathcal{KT}$. In particular, we consider the case where the procedure is implemented in single-processor computing environments and extend the idea to the case of parallel computing environments in Section 5.

To facilitate the presentation, we first introduce some notations. As Procedure *i*-$\mathcal{KT}$ adopts the same selection structure as that of Procedure $\mathcal{KT}$ and proceeds round-by-round, here and throughout, we let $\{X_{i,\ell}^r = \mu_i + \varepsilon_{i,\ell}^r : \ell = 1, 2, \ldots\}$ denote the random outputs from alternative $i \in \mathcal{K}$ in the rth round of the selection, where $\varepsilon_{i,\ell}^r$ follows a normal distribution with mean 0 and variance σ_i^2. Further define $\bar{X}_i^r(t) = \sum_{\ell=1}^t X_{i,\ell}^r/t$ as the sample average of alternative i calculated by using the first t observations in that round. While using Procedure $\mathcal{KN}$ to conduct the matches in the rth round of the selection, we let $\mathcal{KN}\left(C, \alpha_r, \delta, n_0\right)$ denote the output of Procedure $\mathcal{KN}$ which can correctly select the best alternative from a set of alternatives C with probability at least $1 - \alpha_r$ if the first-stage sample size is n_0 and the IZ assumption holds for the alternatives in C with parameter δ. Because in Procedure *i*-$\mathcal{KT}$, at most three alternatives participate in a match, while constructing the continuation regions, we always assume that there are three alternatives in set C, *i.e.*, $|C| = 3$. The detailed descriptions of $\mathcal{KN}\left(C, \alpha_r, \delta, n_0\right)$ are listed as follows.

Procedure $\mathcal{KN}\left(C, \alpha_r, \delta, n_0\right)$

Step 1. (Initialization): *Let* $h_r^2 = \left(n_0 - 1\right)[\alpha_r^{-2/(n_0-1)} - 1]$. *For each alternative* $i \in C$, *simulate* n_0 *observations* $X_{i,1}^r, X_{i,2}^r, \ldots, X_{i,n_0}^r$ *and compute* $\bar{X}_i^r\left(n_0\right) = \sum_{\ell=1}^{n_0} X_{i,\ell}^r/n_0$. *For all* $i, j \in C$ *with* $i \neq j$, *compute*

$$S_{ij}^2(r) = \frac{1}{n_0-1}\sum_{\ell=1}^{n_0}\left[X_{i,\ell}^r - X_{j,\ell}^r - \left(\bar{X}_i^r\left(n_0\right) - \bar{X}_j^r\left(n_0\right)\right)\right]^2 \text{ and } N_{r,i,j} = \frac{h_r^2 S_{ij}^2(r)}{\delta^2}.$$

Let $N_{r,\max} = \max_{i,j\in C, i\neq j} N_{r,i,j}$. *If* $n_0 > \lfloor N_{r,\max} \rfloor$, *then stop and select the alternative with the largest sample mean as the best. Otherwise, set* $t = n_0$ *and go to* ***Step 2.***

Step 2. (Screening): *Set* $C^{old} = C$. *Compute*

$$C = \left\{ i : i \in C^{old} \text{ and } \sum_{\ell=1}^{t}\left(X_{i,\ell}^r - X_{j,\ell}^r\right) \geq -\max\left\{\frac{h_r^2 S_{ij}^2(r)}{2\delta} - \frac{\delta t}{2}, 0\right\}, \forall j \in C^{old}, j \neq i \right\}.$$

Step 3. (Stopping Rules):

- *If* $|C| = 1$, *stop and select the alternative whose index is in set* C *as the best.*
- *Else if* $t > \lfloor N_{r,\max} \rfloor$, *stop and select the alternative with the largest sample mean in set* C *as the best.*
- *Otherwise, take an additional observation* $X_{i,t+1}^r$ *from each alternative* $i \in C$, *set* $t = t + 1$ *and go to* ***Step 2.***

Procedure *i*-$\mathcal{KT}$ differs from Procedure $\mathcal{KT}$ in the following way. In the rth round of the selection, before pairing the surviving alternatives, based on their sampling information collected in previous rounds of selections, we first choose an alternative as the reference alternative, denoted by k_r. Then, we pair the remaining alternatives, and add alternative k_r to each match. Therefore, after the modification, there are three alternatives competing with each other in every match, and all the matches share a common alternative, *i.e.*, alternative k_r. Even though alternative k_r participates in every match, we only allow it to advance to the next round if it wins a single, designated match. For any other match, if alternative k_r wins the match, no alternative can advance. This restriction avoids the situation where an inferior alternative is chosen as the reference alternative and it can survive to the next round of the selection with relatively high probability. In this paper, we let the first match that the reference alternative involves in each round of the selection to be the designated match. In Fig. 3, we give an illustration of the selection of nine alternatives by Procedure *i*-$\mathcal{KT}$, and list the detailed descriptions of Procedure *i*-$\mathcal{KT}$ as follows.

Procedure *i*-$\mathcal{KT}$

Step 1. (Initialization): *Set PICS* α *and IZ parameter* δ. *Let* $\mathcal{I}_r$ *be the set of surviving alternatives at the beginning of the rth round of the selection. Set* $r = 1$ *and* $\mathcal{I}_r = \{1, 2, \ldots, k\}$.

Step 2. (Choosing The Reference Alternative): *Choose the reference alternative, i.e., alternative k_r, from set $\mathcal{I}_r$. Set $\mathcal{I}_r = \mathcal{I}_r \backslash \{k_r\}$ and $\mathcal{I}_{r+1} = \emptyset$.*

Step 3. (Screening): *Let $\mathcal{I}_r^q$ be the set of alternatives that compete in the qth match. Set $\alpha_r = \alpha/2^r$ and $q = 1$.*

- **a.** *If $|\mathcal{I}_r| \geq 2$, pick two alternatives i and j from set $\mathcal{I}_r$, let $\mathcal{I}_r^q = \{i, j, k_r\}$, and update $\mathcal{I}_r = \mathcal{I}_r \backslash \{i, j\}$. If $|\mathcal{I}_r| = 1$, let i be the alternative in $\mathcal{I}_r$, set $\mathcal{I}_r^q = \{i, k_r\}$, and update $\mathcal{I}_r = \emptyset$.*[2]
- **b.** *Let $\hat{i}^*_{r,q}$ be the index of the alternative that wins the qth match in the rth round of the selection and compute*

  $$\hat{i}^*_{r,q} = \mathcal{KN}\left\{\mathcal{I}_r^q, \alpha_r, \delta, n_0\right\}. \tag{1}$$

  If $q = 1$, update $\mathcal{I}_{r+1} = \mathcal{I}_{r+1} \cup \{\hat{i}^*_{r,q}\}$. If $q > 1$ and $\hat{i}^*_{r,q} \neq k_r$, update $\mathcal{I}_{r+1} = \mathcal{I}_{r+1} \cup \{\hat{i}^*_{r,q}\}$.
- **c.** *If $\mathcal{I}_r = \emptyset$, go to **Step 4**. If $\mathcal{I}_r \neq \emptyset$, set $q = q + 1$ and go to **Step 3.a**.*

Step 4. (Stopping Rule): *Set $r = r + 1$. If $|\mathcal{I}_r| = 1$, stop and select the alternative whose index is in $\mathcal{I}_r$ as the best. If $|\mathcal{I}_r| > 1$, go to **Step 2**.*

Notice that it is not always necessary to use Procedure $\mathcal{KN}$ to conduct matches in Procedure *i*-$\mathcal{KT}$. For example, if one only requires the procedure to satisfy the asymptotic statistical validity as $\delta \to 0$, we may employ a variant of Procedure $\mathcal{KN}$, *i.e.*, Procedure $\mathcal{KN}$++ proposed in Kim and Nelson (2006b), to conduct the matches. In this situation, the variant may perform better than the original one. For Procedure *i*-$\mathcal{KT}$, to ensure that the probability of falsely eliminating alternative k can be controlled in each round of the selection, the sampling information obtained prior to the rth round of the selection can only be used to choose the reference alternative. While conducting matches in that round, no previous observations can be used. We choose the reference alternative in **Step 2** and pair the alternatives in **Step 3.a**. The exact choosing and pairing methods are not listed in the procedure. We will discuss them in detail in Section 4. Furthermore, in each round of the selection, it is possible that there exists a match that only contains two alternatives (including the reference alternative). Because in Procedure $\mathcal{KN}$ the continuation regions are constructed by assuming that there are three alternatives in every match, in this situation, the true best alternative has a higher chance to win if it participates in a match with fewer than three alternatives. In the rest of this paper, while carrying out our analysis on Procedure *i*-$\mathcal{KT}$, we ignore such a case and assume that there are three alternatives in every match. It does not affect our theoretical results.

Similar to Procedure $\mathcal{KT}$, CRNs can be applied in Procedure *i*-$\mathcal{KT}$ without much difficulty. In the R&S literature, CRNs are often used to speed up the selection process by introducing positive correlations among the observations generated from different alternatives. To implement CRNs in Procedure *i*-$\mathcal{KT}$, in the rth round of the selection, one needs to predetermine the random number stream used to generate every individual observation and keep the random number stream the same while generating the ℓ-th observation from every alternative. By doing so, the observations generated from different alternatives can be positively correlated.

In each round of the selection, while using Procedure $\mathcal{KN}$ to compare the reference alternative, *i.e.*, alternative k_r, with the other two alternatives in different matches, we can reuse the observations for alternative k_r. Specifically, let set $\mathcal{X}_r$ record the sequence of observations generated from alternative k_r in the rth round of the selection, and $\mathcal{X}_r$ is set to be empty initially. While conducting the qth match in that round, to generate the ℓ-th observation from alternative k_r, we first check whether $\mathcal{X}_r$ has ℓ observations, i.e, the procedure has already generated ℓ observations from alternative k_r in previous matches in that round. If it is true, we can simply pick the ℓ-th observation in $\mathcal{X}_r$ to use. If there are not enough observations in $\mathcal{X}_r$, we generate new observations from alternative k_r and append them to $\mathcal{X}_r$ in order. By doing so, we avoid repeatedly generating new observations from alternative k_r in every match. It does not affect the statistical validity of the procedure.

3.3. Statistical validity

In this subsection, we show that Procedure *i*-$\mathcal{KT}$ is statistically valid. We define two partial sum processes $\{Z_{ij}^r(t) = \sum_{\ell=1}^t \left(X_{i,\ell}^r - X_{j,\ell}^r\right) : t = 1, 2, \ldots,\}$ and $\{\hat{Z}_{ij}^r(t) = \sum_{\ell=1}^t \left(\varepsilon_{i,\ell}^r - \varepsilon_{j,\ell}^r + \delta\right) : t = 1, 2, \ldots,\}$ between any two alternatives i and j in a match in the rth round of the selection. Let $\mathbf{T}_{ij}^r = \inf\{t \geq n_0 : Z_{ij}^r(t) \notin \Lambda(h_r^2 S_{ij}^2(r)/(2\delta))\}$ and $\hat{\mathbf{T}}_{ij}^r = \inf\{t \geq n_0 : \hat{Z}_{ij}^r(t) \notin \Lambda(h_r^2 S_{ij}^2(r)/(2\delta))\}$. To show that Procedure *i*-$\mathcal{KT}$ is statistically valid, we need the following lemma which is the theoretical foundation for Procedure $\mathcal{KN}$ satisfying the PCS guarantee, and provides a tight upper bound on the first-time hitting probability of $\{\hat{Z}_{ij}^r(t) : t = 1, 2, \ldots,\}$ exiting $\Lambda(h_r^2 S_{ij}^2(r)/(2\delta))$ for $t \geq n_0$.

Lemma 1. *(Kim & Nelson, 2001) For any two alternatives i and j, we have*

$$\mathbb{P}\left(\hat{Z}_{ij}^r\left(\hat{\mathbf{T}}_{ij}^r\right) \leq 0\right) \leq \frac{\alpha_r}{2}.$$

Further let $i_r^* = \operatorname{argmax}_{i \in \mathcal{I}_r} \mu_i$. Define Q_r as the event that alternative i_r^* survives in the rth round of the selection, *i.e.*, $\mathsf{Q}_r = \{i_r^* \in \mathcal{I}_{r+1}\}$, and $\bar{\mathsf{Q}}_r$ as the complement of Q_r. Then, with the results stated in Lemma 1, we prove the following theorem.

Theorem 1. *If $\mu_k - \mu_{k-1} \geq \delta$, then for Procedure i-$\mathcal{KT}$,*

$\mathbb{P}$(select alternative k) $\geq 1 - \alpha$.

Proof. Proof We consider the probability that alternative k is falsely eliminated and have

$$\begin{aligned}\mathbb{P}(\text{alternative } k \text{ is eliminated}) &= \sum_{r=1}^{\infty} \mathbb{P}\left(\mathsf{Q}_1 \cap \mathsf{Q}_2 \cap \cdots \cap \mathsf{Q}_{r-1} \cap \bar{\mathsf{Q}}_r\right) \\ &= \sum_{r=1}^{\infty} \mathbb{P}\left(\bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) \mathbb{P}\left(\bar{\mathsf{Q}}_r \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right).\end{aligned} \tag{2}$$

We focus on analyzing the second term in Eq. (2). Let alternatives i' and j' be alternatives that compete with alternative i_r^* in the rth round of the selection. If alternative i_r^* is the reference alternative, let alternatives i' and j' be the alternatives that compete with the reference alternative in the designated match of round r. For any $i \in \{i', j'\}$, $Z_{i_r^* i}^r(t) - \hat{Z}_{i_r^* i}^r(t) = \left(\mu_{i_r^*} - \mu_i - \delta\right) t$. Because event $\cap_{r'=1}^{r-1} \mathsf{Q}_{r'}$ implies that alternative k is in set $\mathcal{I}_r$, *i.e.*, $i_r^* = k$, in this case, $\mu_{i_r^*} - \mu_i - \delta \geq 0$ and $Z_{i_r^* i}^r(t) \geq \hat{Z}_{i_r^* i}^r(t)$ for all $t \geq 1$. Therefore, if event $\{Z_{i_r^* i}^r(\mathbf{T}_{i_r^* i}^r) \leq 0\}$ happens, event $\{\hat{Z}_{i_r^* i}^r(\hat{\mathbf{T}}_{i_r^* i}^r) \leq 0\}$ must happen. Then we have,

$$\begin{aligned}\mathbb{P}\left(\bar{\mathsf{Q}}_r \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) &= \mathbb{P}\left(\bigcup_{i \in \{i', j'\}} \left\{Z_{i_r^* i}^r\left(\mathbf{T}_{i_r^* i}^r\right) \leq 0\right\} \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) \\ &\leq \mathbb{P}\left(\bigcup_{i \in \{i', j'\}} \left\{\hat{Z}_{i_r^* i}^r\left(\hat{\mathbf{T}}_{i_r^* i}^r\right) \leq 0\right\} \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) \\ &\leq \mathbb{P}\left(\hat{Z}_{i_r^* i'}^r\left(\hat{\mathbf{T}}_{i_r^* i'}^r\right) \leq 0 \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) \\ &\quad + \mathbb{P}\left(\hat{Z}_{i_r^* j'}^r\left(\hat{\mathbf{T}}_{i_r^* j'}^r\right) \leq 0 \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right).\end{aligned} \tag{3}$$

Because no observations generated in previous rounds of selections are used to conduct the matches in the rth round of the selection, for $i \in$

[2] In the rth round of the selection, we pick alternatives from set $\mathcal{I}_r$ to conduct matches until there are no alternatives left in set $\mathcal{I}_r$.

$\{i', j'\}$, $\{\hat{Z}^r_{i^*_r i}(t) : t = 1, 2, \dots, \}$ exiting $\Lambda(h_r^2 S^2_{i^*_r i}(r)/(2\delta))$ is independent of $\bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}$. Then, we can rewrite Eq. (3) as follows,

$$\begin{aligned}(3) &= \mathbb{P}\left(\hat{Z}^r_{i^*_r i'}\left(\hat{\mathbf{T}}^r_{i^*_r i'}\right) \leq 0\right) + \mathbb{P}\left(\hat{Z}^r_{i^*_r j'}\left(\hat{\mathbf{T}}^r_{i^*_r j'}\right) \leq 0\right) \\ &\leq \frac{\alpha_r}{2} + \frac{\alpha_r}{2} \\ &= \alpha_r, \end{aligned} \tag{4}$$

where the first inequality holds due to Lemma 1. Combining the results listed in Eqs. (3) and (4), we have

$$\mathbb{P}\left(\bar{\mathsf{Q}}_r \middle| \bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) \leq \alpha_r. \tag{5}$$

Plug the results in Eq. (5) into Eq. (2) yielding

$$\mathbb{P}\,(\text{alternative } k \text{ is eliminated}) \leq \sum_{r=1}^{\infty} \alpha_r \mathbb{P}\left(\bigcap_{r'=1}^{r-1} \mathsf{Q}_{r'}\right) \leq \sum_{r=1}^{\infty} \alpha_r \leq \alpha.$$

Therefore, $\mathbb{P}\,(\text{select alternative } k) \geq 1 - \alpha$. It concludes the proof. □

We close this section with a remark that with similar techniques discussed in Zhong and Hong (2022), we can modify Procedure i-$\mathcal{KT}$ to satisfy the PGS guarantee. Specifically, besides the PICS α, one should allocate the tolerable error δ to each round of the selection and employ a procedure with the PGS guarantee to conduct the matches. By doing so, the procedure may satisfy the stronger statistical guarantee. However, these modifications can cause efficiency loss on the total sample size.

4. Analyzing the total sample size

In this section, we aim to analyze the total sample size of Procedure i-$\mathcal{KT}$, and compare it with that of Procedure $\mathcal{KT}$. For both procedures, the average sample size of a match and the total number of matches are two major factors affecting the total sample size. We may compare the performances of the procedures from these two aspects.

4.1. The average sample size of a match

In this subsection, we show that for Procedure i-$\mathcal{KT}$, by choosing a reference alternative with a relatively large mean, the average sample size of a match in the procedure tends to be smaller than that of Procedure $\mathcal{KT}$.

To compare the performances of the two procedures from the aspect of the average sample size of a match, we can fix two alternatives, let them participate in a match in the rth round of the selection in each procedure, and compare the sample sizes of the two matches. Specifically, let ξ and ζ be the indices of two alternatives. Without loss of generality, we assume that $\mu_\xi \geq \mu_\zeta$. Define $\mathbf{N}^*_\xi$ and $\mathbf{N}^*_\zeta$ as the sample sizes of alternatives ξ and ζ respectively if they participate in a match in the rth round of the selection of Procedure $\mathcal{KT}$. We also define $\mathbf{N}_\xi$ and $\mathbf{N}_\zeta$ as the sample sizes of alternatives ξ and ζ respectively if they participate in a match in the rth round of the selection of Procedure i-$\mathcal{KT}$ and alternative k_r is the reference alternative in the match. For Procedure i-$\mathcal{KT}$, the observations generated from alternative k_r are shared by all the matches in the rth round of the selection. It suggests that even though there are three alternatives competing with each other in the match of Procedure i-$\mathcal{KT}$, as we record the sample size of the match, we may ignore the observations generated from alternative k_r and only count the observations generated from alternatives ξ and ζ. Then, to compare the average sample sizes of a match in Procedure $\mathcal{KT}$ and Procedure i-$\mathcal{KT}$, we may compare $\mathbb{E}[\mathbf{N}^*_\xi + \mathbf{N}^*_\zeta]$ and $\mathbb{E}[\mathbf{N}_\xi + \mathbf{N}_\zeta]$.

One may argue that for Procedure i-$\mathcal{KT}$, as the selection proceeds, the observations generated from alternative k_r are shared by fewer and fewer matches. Furthermore, as we will show later, while implementing Procedure i-$\mathcal{KT}$ in parallel computing environments and conducting the matches in different processors, we may need to generate different sequences of observations from alternative k_r. In this situation, the number of observations generated from alternative k_r may have significant impact on the sample size of a match. However, we would like to notice that only a small portion of matches are affected because in Procedure i-$\mathcal{KT}$ most matches are conducted in the early rounds of the selection process and the observations generated from alternative k_r are shared by a relatively large number of matches in these rounds of selections.

To analyze $\mathbb{E}\left[\mathbf{N}^*_\xi + \mathbf{N}^*_\zeta\right]$ and $\mathbb{E}[\mathbf{N}_\xi + \mathbf{N}_\zeta]$, we consider the situation where $\alpha_r \to 0$. Firstly, let $G_1, G_2, \dots,$ be a sequence of independently and identically distributed (i.i.d.) normal random variables with mean Δ and variance σ_G^2, and $\{C(t) = \sum_{\ell=1}^{t} G_\ell : t = 1, 2, \dots, \}$ be a partial sum process. Further let $\mathbf{T}_G = \inf\{t \geq n_0 : C(t) \notin \Lambda\left(\beta S_G^2\right)\}$, where $\beta > 0$, $S_G^2 = \sum_{\ell=1}^{n_0}\left(G_\ell - \bar{G}\left(n_0\right)\right)^2 / \left(n_0 - 1\right)$, and $\bar{G}\left(n_0\right) = \sum_{\ell=1}^{n_0} G_\ell / n_0$. To characterize the relationships among β, $\mathbf{T}_G$ and $C\left(\mathbf{T}_G\right)$, we prove the following lemma. Its proof is included in the appendix.

Lemma 2. *As $\beta \to \infty$, $\mathbf{T}_G \to \infty$ with probability one (w.p.1). Furthermore, if $\Delta > 0$, as $\beta \to \infty$, $C\left(\mathbf{T}_G\right) > 0$ w.p.1.*

Then, with the results listed in Lemma 2, we show that if $\Delta \geq 0$ and β is sufficiently large, $\mathbb{E}\left[\mathbf{T}_G\right]$ can be approximated by the intersection point between lines $l(t) = \Delta t$ and $u(t) = \beta\sigma_G^2 - \delta t/2$. We summarize the results in the following lemma. Its proof is included in the appendix.

Lemma 3. *Suppose that $\Delta \geq 0$. Then, as $\beta \to \infty$, $\mathbf{T}_G/\beta \to 2S_G^2/(2\Delta + \delta)$ w.p.1. Furthermore, $\lim_{\beta\to\infty} \mathbb{E}\left[\mathbf{T}_G\right]/\beta = 2\sigma_G^2/(2\Delta + \delta)$.*

Let $\Delta_1 = \mu_{k_r} - \mu_\xi$, $\Delta_2 = \mu_{k_r} - \mu_\zeta$, and $\Delta' = \mu_\xi - \mu_\zeta$. Further define $\sigma^2_{\max} = \max\{\sigma^2_{\xi\zeta}, \sigma^2_{\xi k_r}, \sigma^2_{\zeta k_r}\}$. Based on Lemmas 2 and 3, we prove the following proposition. Its proof is included in the appendix.

Proposition 1. *If $\Delta_1 > 0$ and $\Delta' \geq 0$, we have*

$$\liminf_{\alpha_r \to 0} \frac{\mathbb{E}\left[\mathbf{N}^*_\xi + \mathbf{N}^*_\zeta\right]}{\mathbb{E}\left[\mathbf{N}_\xi + \mathbf{N}_\zeta\right]} \geq \frac{\delta + 2\Delta_1}{\delta + 2\Delta'} \cdot \frac{\sigma^2_{\xi\zeta}}{\sigma^2_{\max}} \cdot 2^{-\frac{2}{n_0 - 1}}.$$

Since there are only three alternatives competing in a match in Procedure i-$\mathcal{KT}$, we expect that the difference between $\sigma^2_{\xi\zeta}$ and $\sigma^2_{\max}$ may not be large. In Procedure i-$\mathcal{KT}$, while using Procedure $\mathcal{KN}$ to conduct the matches, the size of the continuation region used to compare any two alternatives is larger than that of Procedure $\mathcal{KT}$, and there is some efficiency loss on the sample size. Term $2^{-\frac{2}{n_0-1}}$ quantifies such efficiency loss. As one may observe, if $n_0 \geq 20$, then $2^{-2/(n_0-1)} > 0.9$, suggesting that such efficiency loss is small. Therefore, we may conclude that, when α_r is small, i.e, α is set to be small or r is relatively large, as long as $\Delta_1 > \Delta'$, the sample size of a match in Procedure i-$\mathcal{KT}$ tends to be smaller than that of Procedure $\mathcal{KT}$. The rationale behind the results is that if there are only alternatives ξ and ζ participating in a match and their means are close, while using Procedure $\mathcal{KN}$ to conduct the match, the comparisons between the two alternatives may not stop until the partial sum of difference process reaches the ending point of the continuation region. By adding a reference alternative with a large mean to the match, the reference alternative may quickly eliminate both alternatives ξ and ζ with a few observations and there is no need to wait for the completion of the comparisons between alternatives ξ and ζ. It is arguable that, in practice, while solving large-scale problems, there are a large number of clearly inferior alternatives. To use Procedure $\mathcal{KT}$ to solve these problems, during the selection process, the procedure is very likely to pair two alternatives whose means are small but close. Conducting such a match requires a lot of sampling efforts. However, by adding a reference alternative with a large mean to the match, we can significantly reduce the sample size of the match.

In this paper, we propose the following way to choose the reference alternative. Let $\bar{\mathfrak{X}}_{r,i}$ denote the sample average of alternative i calculated based on all its observations obtained prior to the rth round of the selection. Then, in the rth round of the selection, we may choose the alternative with the largest $\bar{\mathfrak{X}}_{r,i}$ as the reference alternative, *i.e.*, $k_r = \text{argmax}_{i\in\mathcal{I}_r} \bar{\mathfrak{X}}_{r,i}$, *i.e.*, we may replace **Step 2** in Procedure i-$\mathcal{KT}$ by the following one,

Step 2'. (Choosing The Reference Alternative): *Choose alternative* $k_r = \text{argmax}_{i\in\mathcal{I}_r} \bar{\mathfrak{X}}_{r,i}$ *as the reference alternative. Set* $\mathcal{I}_r = \mathcal{I}_r \backslash \{k_r\}$ *and* $\mathcal{I}_{r+1} = \emptyset$.

In order to apply such a choosing method in the first round of the selection, before the selection starts, we can add an initial stage to Procedure i-$\mathcal{KT}$ and generate $\mathfrak{n}_0$ observations from each alternative to estimate its sample mean. Notice that, in practice, without any foreknowledge on the configuration of the means, determining the IZ parameter is often very challenging. With this initial-stage sampling, we may have a better understanding on the means of the alternatives and thus can set a proper value for δ. The statistical validity of the procedure remains intact as these observations are not employed in the comparison and elimination of alternatives.

In what follows, we show that by using the method listed above to choose the reference alternatives, as long as α is set to be small, starting from the second round of the selection, Procedure i-$\mathcal{KT}$ tends to choose alternative k as the reference alternative and maximizes the value for Δ_1. We summarize the results in the following proposition and include its proof in the appendix.

Proposition 2. *If the method specified in **Step 2'** is used to choose the reference alternative in each round of the selection and* $\mu_k - \mu_{k-1} > 0$, *as* $\alpha \to 0$, *then* $k_r = k$ *for* $r \geq 2$ *w.p.1.*

We note that Proposition 2 does not require the IZ assumption. The proposition holds as long as alternative k is the unique best alternative, *i.e.*, $\mu_k - \mu_{k-1} > 0$.

4.2. The total number of matches

In this subsection, we compare the performances of the two procedures from the aspect of the total number of matches. For both procedures, the more alternatives the procedures can eliminate in each round of the selection, the fewer matches the procedures have to conduct. Therefore, to compare the performances of the two procedures from the aspect of the total number of matches, we may compare the number of alternatives the procedures can eliminate in each round of the selection. For Procedure $\mathcal{KT}$, about half of the surviving alternatives are eliminated in each round of the selection. In what follows, we show that for Procedure i-$\mathcal{KT}$, many more alternatives can be eliminated.

To proceed the analysis, we first define two types of matches, namely the Type A match and the Type B match.

Definition 4.1. Let $\{\eta, \rho\}$ be the set of alternatives that compete with the reference alternative k_r in a match. Then, we call the match a Type A match if

$$\mu_{k_r} \geq \mu_\eta + \delta \text{ and } \mu_{k_r} \geq \mu_\rho + \delta,$$

and call the match a Type B match if

$$\mu_{k_r} \geq \mu_\eta \text{ and } \mu_{k_r} \geq \mu_\rho.$$

According to the definition, a Type A match is the match where the mean of the reference alternative is of at least δ larger than those of the other two alternatives in the match, and a Type B match is the match where the mean of the reference alternative is larger than those of the other two alternatives in the match. Type A matches are a subset of Type B matches. We focus on analyzing the probabilities that the reference alternative can win in these two types of matches. To conduct the analysis, we need the following lemma.

Lemma 4 (Tamhane 1977). *Let* $V_1, V_2, \dots, V_k$ *be independent random variables, and let* $g_j(v_1, v_2, \dots, v_k)$, *for* $j = 1, 2, \dots, p$, *be nonnegative, real-valued functions, each one nondecreasing in each of its arguments. Then*

$$\mathbb{E}\left[\prod_{j=1}^{p} g_j\left(V_1, V_2, \dots, V_k\right)\right] \geq \prod_{j=1}^{p} \mathbb{E}\left[g_j\left(V_1, V_2, \dots, V_k\right)\right].$$

Then, we prove the following two lemmas. They basically state that in the rth round of the selection, for a Type A match, with probability no less than $1 - \alpha_r$, the reference alternative can win the match, and for a Type B match, if no CRNs are used, with probability no less than $1/4$, the reference alternative can win the match. Their proofs are included in the appendix.

Lemma 5. *For Procedure* i-$\mathcal{KT}$, *in the* r*th round of the selection, if the* q*th match is a Type* A *match, then*

$$\mathbb{P}\left(\hat{i}^*_{r,q} = k_r\right) \geq 1 - \alpha_r.$$

Lemma 6. *Suppose that no CRNs are used. For Procedure* i-$\mathcal{KT}$, *in the* r*th round of the selection, if the* q*th match is a Type* B *match, then*

$$\mathbb{P}\left(\hat{i}^*_{r,q} = k_r\right) \geq \frac{1}{4}.$$

The lower bound listed in Lemma 6 is tight. Suppose that in a match $\mu_\eta = \mu_\rho = \mu_{k_r}$, $\sigma_\eta \gg 0$, and $\sigma_\rho = \sigma_{k_r} \approx 0$. In such a case, the comparisons between alternatives ρ and k_r would finish with a few observations. Then the winner of alternatives ρ and k_r would compete with alternative η. For alternatives ρ and k_r, the probability of winning is approximately $1/4$. For alternative η, the probability of winning is approximately $1/2$.

Define $\mathsf{M}^A_{r,q}$ and $\mathsf{M}^B_{r,q}$ as the events that, in the rth round of the selection, the qth match is a Type A match and a Type B match respectively. Further let $\mathbf{p}^r_A$ and $\mathbf{p}^r_B$ be the proportions of Type A and Type B matches in the rth round of selection respectively, *i.e.*, $\mathbb{E}[\mathbb{1}\{\mathsf{M}^A_{r,q}\}|\mathcal{I}_r] = \mathbf{p}^r_A$ and $\mathbb{E}[\mathbb{1}\{\mathsf{M}^B_{r,q}\}|\mathcal{I}_r] = \mathbf{p}^r_B$, where $\mathbb{1}\{\cdot\}$ is the indicator function. With Lemmas 5 and 6, we prove the following proposition.

Proposition 3. *For Procedure* i-$\mathcal{KT}$, *if no CRNs are used, we have*

$$\mathbb{E}\left[|\mathcal{I}_{r+1}|\,\big|\,|\mathcal{I}_r|\right] \leq 1 + \left(\left\lceil \frac{|\mathcal{I}_r| - 1}{2} \right\rceil - 1\right)\left[1 - \frac{1}{4}\mathbf{p}^r_B - \left(\frac{3}{4} - \alpha_r\right)\mathbf{p}^r_A\right],$$

and if CRNs are used, we have

$$\mathbb{E}\left[|\mathcal{I}_{r+1}|\,\big|\,|\mathcal{I}_r|\right] \leq 1 + \left(\left\lceil \frac{|\mathcal{I}_r| - 1}{2} \right\rceil - 1\right)\left[1 - \left(1 - \alpha_r\right)\mathbf{p}^r_A\right].$$

Proposition 3 suggests that in the worst case scenario, *i.e.*, the alternative with the smallest mean is chosen as the reference alternative and $\mathbf{p}^r_A = \mathbf{p}^r_B = 0$ for all $r \geq 1$, Procedure i-$\mathcal{KT}$ eliminates about half of the surviving alternatives in each round of the selection. As long as $\mathbf{p}^r_A > 0$ and $\mathbf{p}^r_B > 0$ for some $r \geq 1$, the procedure eliminates more alternatives than Procedure $\mathcal{KT}$ does. In this situation, fewer matches are needed to identify the best alternative. As stated in Proposition 2, if α is set to be small, starting from the second round of the selection, Procedure i-$\mathcal{KT}$ tends to choose alternative k as the reference alternative. In such a case, $\mathbf{p}^r_A = \mathbf{p}^r_B = 1$, and no matter whether CRNs are used or not, fewer than $\alpha_r|\mathcal{I}_r|/2$ alternatives can survive, on average. Also notice that, if no CRNs are used, to efficiently eliminate alternatives, it is always beneficial to add the reference alternative to each match even if the mean of the reference alternative is not significantly larger than those of others. For example, consider a problem where the means of the alternatives follow the slippage configuration, *i.e.*, $\mu_1 = \mu_2 = \cdots = \mu_k - \delta = 0$, and in each round of the selection, the procedure fails to choose alternative k as the reference alternative. In this situation, $\mathbf{p}^r_A = 0$ and $\mathbf{p}^r_B \approx 1$ for all $r \geq 1$. Based on Proposition 3, the procedure still eliminates more than half of the surviving alternatives in each round of the selection and performs better than Procedure $\mathcal{KT}$.

According to Proposition 3, we may conclude that the larger values for $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$, the more alternatives the procedure can eliminate in each round of the selection and fewer matches the procedure needs to conduct. Therefore, to improve the performance of Procedure i-$\mathcal{KT}$ from the aspect of the total number of matches, increasing the values for $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$ is critical. To make it happen, choosing a reference alternative with a large mean is necessary. However, how the surviving alternatives are paired may also affect the values for $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$. For example, in the rth round of the selection, suppose that the alternatives in set $\mathcal{I}_r$ are sorted in a descending order based on their true means, and the reference alternative, i.e, alternative k_r, is the median in set $\mathcal{I}_r$. Consider the following two ways to pair the alternatives in set $\mathcal{I}_r$:

1. We let $\mathcal{I}_r[1]$ and $\mathcal{I}_r[2]$ form a match, $\mathcal{I}_r[3]$ and $\mathcal{I}_r[4]$ form a match, and so on, where $\mathcal{A}[\ell]$ is the ℓ-th element in set $\mathcal{A}$;
2. We let $\mathcal{I}_r[1]$ and $\mathcal{I}_r[|\mathcal{I}_r|]$ form a match, $\mathcal{I}_r[2]$ and $\mathcal{I}_r[|\mathcal{I}_r|-1]$ form a match, and so on.

It can be checked that in the former case, $\mathbf{p}_A^r \geq 0$ and $\mathbf{p}_B^r = 1/2$, and in the latter case, $\mathbf{p}_A^r = \mathbf{p}_B^r = 0$. In fact, in each round of the selection if the reference alternative is chosen and fixed, using the first way to pair the alternatives maximizes the values for $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$. Therefore, in this paper, to ensure that the values for $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$ can be as large as possible, after choosing the reference alternative in each round of the selection, we sort the alternatives in set $\mathcal{I}_r$ based on their values for $\bar{\mathfrak{X}}_{r,i}$ and adopt the first way to pair the alternatives, *i.e.*, we may replace **Step 3.a** in Procedure i-$\mathcal{KT}$ by the following one,

Step 3'. (Screening): *Let $\mathcal{I}_r^q$ be the set of alternatives that compete in the qth match. Set $\alpha_r = \alpha/2^r$ and $q = 1$. Sort the alternatives in set $\mathcal{I}_r$ in a descending order based on their values for $\bar{\mathfrak{X}}_{r,i}$.*

a. *If $|\mathcal{I}_r| \geq 2$, let i and j be the first two elements in set $\mathcal{I}_r$ and $\mathcal{I}_r^q = \{i, j, k_r\}$. Update $\mathcal{I}_r = \mathcal{I}_r \backslash \{i, j\}$. If $|\mathcal{I}_r| = 1$, let i be the alternative in $\mathcal{I}_r$, set $\mathcal{I}_r^q = \{i, k_r\}$, and update $\mathcal{I}_r = \emptyset$.*

We are aware that using different pairing methods can in turn affect the average sample size of a match. Then, it may become unclear whether the average sample size of a match in Procedure i-$\mathcal{KT}$ is still smaller than that of Procedure $\mathcal{KT}$. Precisely estimating the impact of such changes on the average sample size of a match may be difficult. However, we conjecture that the impact is limited. The main reason is that in each round of the selection, if the mean of the reference alternative is large, most alternatives are eliminated by the reference alternative. In other words, whichever matches these alternatives participate in, the number of observations generated from these alternatives do not change too much.

5. Implementing procedure i-$\mathcal{KT}$ in parallel computing environments

In this section, we discuss on how to implement Procedure i-$\mathcal{KT}$ in parallel computing environments. Because there are some differences between Procedure $\mathcal{KT}$ and Procedure i-$\mathcal{KT}$, to run Procedure i-$\mathcal{KT}$ in parallel computing environments, we need a slightly different implementation scheme.

Suppose that we want to implement Procedure i-$\mathcal{KT}$ in a parallel computing environment with m processors. At the beginning of each round of the selection, if there are more than $2m$ alternatives in contention, we first specify a processor to do the tasks of choosing the reference alternative, pairing the remaining alternatives, and scheduling the matches. Then, the specified processor equally divides the matches into m subsets and assigns every processor (including the specified processor itself) one subset of matches. By doing so, every processor is responsible for conducting one subset of matches. Within a processor, the matches are conducted sequentially. After finishing the matches as assigned, every processor submits the selection results back to the specified processor, and the procedure proceeds to the next round of the selection.

As the selection proceeds, if there are fewer than $2m$ alternatives left, it is not possible to assign every processor a match in the rest rounds of selections and some processors may stay idle for a relatively long time. To address this issue, we adopt the same techniques as illustrated in Procedure $\mathcal{KT}$. In particular, we suggest not using CRNs and employing the Rinott's procedure to compare these surviving alternatives all at once. In such a case, the procedure equally allocates surviving alternatives to all the processors.[3] After the allocation, every processor generates some additional observations from the alternatives as assigned and submits the sampling information back to the specified processor. The specified processor calculates the sample mean of each alternative based on these additional observations and selects the alternative with the largest sample mean as the final best alternative. In terms of total sample size, the Rinott's procedure is not a very efficient procedure to compare alternatives. In this paper, we restrict from invoking the Rinott's procedure in the second round of the selection. As stated in Proposition 2, starting from the second round of the selection, Procedure i-$\mathcal{KT}$ tends to select alternative k as the reference alternative. Then, it is worth trying to test whether the reference alternative can eliminate all the other alternatives in that round and terminate the selection without employing the Rinott's procedure.

To use the Rinott's procedure, we need to calculate the Rinott's constant with which we can determine the number of observations generated from each surviving alternative. Suppose that there are m' alternatives left while employing the Rinott's procedure. Define $h(\alpha_r, m', n_0)$ as the Rinott's constant calculated by letting PICS, the number of alternatives, and the first-stage sample size be α_r, m', and n_0 respectively. In what follows, we provide detailed descriptions on Procedure i-$\mathcal{KT}$ if it is implemented in a parallel computing environment with m processors and the methods proposed in Sections 4.1 and 4.2 are used to choose the reference alternative and pair the alternatives in every processor respectively. We call the parallel version of the procedure Procedure i-$\mathcal{KT}^+$.

Procedure i-$\mathcal{KT}^+$

Step 1. (Initialization): *Set PICS α, IZ parameter $\delta > 0$, initial-stage sample size $\mathfrak{n}_o$, first-stage sample size in a match n_0, and the number of processors m. Let $\mathcal{I}_r$ be the set of surviving alternatives at the beginning of the rth round of the selection and $\mathcal{I}_{r,s}$ be the set of alternatives assigned to processor s for $s = 1, 2, \ldots, m$ in that round. Set $r = 1$ and $\mathcal{I}_r = \{1, 2, \ldots, k\}$.*

Step 2. (Initial-Stage Sampling): *Equally allocate k alternatives to m processors. In processor $s = 1, 2, \ldots, m$: generate $\mathfrak{n}_o$ observations from each alternative allocated to the processor.*

Step 3. (The rth Round of Selection):

a. *Update $\bar{\mathfrak{X}}_{r,i}$ for all $i \in \mathcal{I}_r$. Sort the alternatives in set $\mathcal{I}_r$ in a descending order based on their values for $\bar{\mathfrak{X}}_{r,i}$. Let $k_r = \mathcal{I}_r[1]$ and $\mathcal{I}_r = \mathcal{I}_r \backslash \{k_r\}$. For $s = 1, 2, \ldots, m$, let $\mathcal{I}_{r,s} = \{\mathcal{I}_r[\ell] : \ell \leq |\mathcal{I}_r|$ and $\ell = s, s+1, 2m+s, 2m+s+1, 3m+s, 3m+s+1, \ldots, \}$*[4]*. Pass the values of k_r and $\mathcal{I}_{r,s}$ to processor s. Set $\mathcal{I}_{r+1} = \emptyset$.*

b. *In processor $s = 1, 2, \ldots, m$: let $\mathcal{I}_{r,s}^q$ be the set of alternatives that compete in the qth match in processor s in the rth round of the selection. Set $\alpha_r = \alpha/2^r$ and $q = 1$.*

i. *If $|\mathcal{I}_{r,s}| \geq 2$, let $i = \mathcal{I}_{r,s}[1]$, $j = \mathcal{I}_{r,s}[2]$, $\mathcal{I}_{r,s}^q = \{i, j, k_r\}$, and $\mathcal{I}_{r,s} = \mathcal{I}_{r,s} \backslash \{i, j\}$. If $|\mathcal{I}_{r,s}| = 1$, let $i = \mathcal{I}_{r,s}[1]$, $\mathcal{I}_{r,s}^q = \{i, k_r\}$, and $\mathcal{I}_{r,s} = \emptyset$.*

[3] If there are fewer than $2m$ surviving alternatives, it may not be possible to allocate every processor the same number of surviving alternatives. The numbers of surviving alternatives allocated to any two processors may differ by 1.

[4] We let every two consecutive alternatives in set $\mathcal{I}_r$ form a match and sequentially allocate the matches to the processors until there is no match left. By doing so, the workloads of different processors are roughly the same.

- ***ii.*** *Let* $\hat{i}^*_{r,s,q} = \mathcal{KN}\left\{\mathcal{I}^q_{r,s}, \alpha_r, \delta, n_0\right\}$*. If* $s = 1$ *and* $q = 1$*, update* $\mathcal{I}_{r+1} = \mathcal{I}_{r+1} \cup \{\hat{i}^*_{r,s,q}\}$*. For* $s \neq 1$ *or* $q \neq 1$*, if* $\hat{i}^*_{r,s,q} \neq k_r$*, update* $\mathcal{I}_{r+1} = \mathcal{I}_{r+1} \cup \{\hat{i}^*_{r,s,q}\}$*.*
- ***iii.*** *If* $\mathcal{I}_{r,s} = \emptyset$*, selection completes in processor* s*. If* $\mathcal{I}_{r,s} \neq \emptyset$*, set* $q = q + 1$ *and go to* ***Step3.b.i****.*

Step 4. (Stopping Rule):

- ***a.*** *If* $|\mathcal{I}_{r+1}| = 1$*, select the alternative whose index is in* $\mathcal{I}_{r+1}$ *as the best.*
- ***b.*** *If* $|\mathcal{I}_{r+1}| \geq 2m$ *or* $r < 2$*, set* $r = r + 1$ *and go to* ***Step 3****.*
- ***c.*** *If* $1 < |\mathcal{I}_{r+1}| < 2m$ *and* $r \geq 2$*, let* $m' = |\mathcal{I}_{r+1}|$ *and equally allocate alternatives in set* $\mathcal{I}_{r+1}$ *to* m *processors. Set* $r = r + 1$ *and* $\alpha_r = \alpha/2^r$*. In processor* $s = 1, 2, \ldots, m$*: let* $\mathcal{W}_s$ *be the set of alternatives allocated to processor* s*. For each* $i \in \mathcal{W}_s$*,*
  - ***i.*** *Generate* n_0 *observations from alternative* i *and calculate its sample variance* $S_i^2(r)$ *based on these* n_0 *observations.*
  - ***ii.*** *Generate additional*

$$\max\left\{0, \left\lceil \left(\frac{h(\alpha_r, m', n_0)}{\delta}\right)^2 S_i^2(r) \right\rceil - n_0 \right\},$$

observations from alternative i *and calculate the sample mean of alternative* i *based on the observations generated in* ***Step 4.c.i*** *and* ***Step 4.c.ii****.*

Select the alternative with the largest sample mean in set $\cup_{s=1,2,\ldots,m}\mathcal{W}_s$ *as the best.*

For Procedure *i*-$\mathcal{KT}^+$, the Rinott's Procedure is employed in **Step 4.c**. We do not balance the workloads of the processors in this step because we find that Procedure *i*-$\mathcal{KT}$ often terminates without invoking the Rinott's procedure in practice. Furthermore, in each round of the selection, the designated match is set to be the first match that the reference involves in processor 1. Only when the reference alternative wins this match, it can advance to the next round of the selection. In Procedure *i*-$\mathcal{KT}^+$, the observations generated from the reference alternative can still be reused in different matches. However, to avoid unnecessary communications and coordinations among the processors, we only reuse the observations generated from the reference alternative within each individual processor. While conducting matches in different processors, we generate different sequences of observations from the reference alternative. Next, we show that Procedure *i*-$\mathcal{KT}^+$ is statistically valid. We first summarize the property of the Rinott's procedure as follows.

Lemma 7. *(Rinott, 1978) Suppose that for Procedure i-*$\mathcal{KT}^+$*, in* ***Step 4.c****,* $k \in \cup_{s=1,2,\ldots,m}\mathcal{W}_s$ *and* $\mu_k - \mu_j \geq \delta$ *for all* $j \in \cup_{s=1,2,\ldots,m}\mathcal{W}_s$ *and* $j \neq k$*. If each alternative* $i \in \cup_{s=1,2,\ldots,m}\mathcal{W}_s$ *has more than*

$$\max\left\{n_0, \left\lceil \left(\frac{h(\alpha_r, m', n_0)}{\delta}\right)^2 S_i^2(r) \right\rceil \right\},$$

observations, then with probability at least $1-\alpha_r$*, alternative* k *has the largest sample mean.*

Then, with Lemma 7, we prove the following theorem. Its proof is included in the appendix.

Theorem 2. *If* $\mu_k - \mu_{k-1} \geq \delta$*, then for Procedure i-*$\mathcal{KT}^+$*,*

$\mathbb{P}$*(select alternative* k*)* $\geq 1 - \alpha$*.*

As one may observe, in Procedure *i*-$\mathcal{KT}^+$, the workloads of different processors are re-balanced in each round of the selection. Furthermore, choosing the reference alternative, pairing the remaining surviving alternatives, and scheduling the matches are non-parallelizable calculations and require additional communications and coordinations among the processors. These operations may affect the utilizations of the processors and diminish the benefit of parallel computing. In what follows, we show that as α approaches to zero, Procedure *i*-$\mathcal{KT}^+$ tends to stop in two rounds of selections. It suggests that the frequency of conducting these operations is low. We also prove that in this situation, not only the expected sample size of Procedure *i*-$\mathcal{KT}^+$ can grow linearly in k but also the constant on the growth rate is immune to the change of the number of processors. To conduct the analysis, we need following lemma which provides an upper bound for the Rinott's constant.

Lemma 8. *(Zhong & Hong, 2022) Let* $h(\alpha_r, m', n_0)$ *be the Rinott's constant determined by* α_r*,* m'*, and* n_0*. Then,*

$$h(\alpha_r, m', n_0) \leq \left\{2(n_0 - 1)\left[\left(\frac{2(m'-1)}{\alpha_r}\right)^{\frac{2}{n_0-1}} - 1\right]\right\}^{1/2}.$$

Let $\mathbf{N}^+$ denote the total sample size of Procedure *i*-$\mathcal{KT}^+$ less the initial stage sample size, *i.e.*, $n_0 k$. We summarize the analysis results in the following theorem. Its proof is included in the appendix.

Theorem 3. *If* $\alpha \to 0$ *and* $\mu_k - \mu_{k-1} > 0$*, Procedure i-*$\mathcal{KT}^+$ *stops in two rounds of selections w.p.1. Furthermore, if* σ_{ij}^2 *is upper bounded by a constant* $\sigma^2_{upper} > 0$ *for all* $i \neq j \in \mathcal{K}$ *and* $n_0 \geq 3$*,*

$$\limsup_{\alpha \to 0} \mathbb{E}\left[\mathbf{N}^+\right] \alpha^{\frac{2}{n_0-1}} \leq \frac{36\sigma^2_{upper} n_0}{\delta^2} k.$$

Notice that to implement Procedure $\mathcal{KT}$ in parallel computing environments, the Rinott's procedure is always invoked to compare m surviving alternatives in the last round of the selection. As a result, for Procedure $\mathcal{KT}$, even though the expected total sample size grows linearly in k, the constant on the growth rate additionally depends on the number of processors. However, for Procedure *i*-$\mathcal{KT}^+$, it may sometimes complete the selection without invoking the Rinott's procedure. For example, consider a situation where there are more than $2m$ alternatives in contention at the beginning of the rth round of the selection, and after conducting the matches in different processors, the reference alternative eliminates all the other alternatives. In this situation, only the reference alternative survives and Procedure *i*-$\mathcal{KT}^+$ can directly select the reference alternative as the final best alternative. When α is set to be small, the procedure is very likely to select the true best alternative as the reference alternative in the second round of the selection and use the reference alternative to eliminate all the other alternatives. Therefore, compared to Procedure $\mathcal{KT}$, adding more processors in parallel computing environments has less impact on the total sample size of Procedure *i*-$\mathcal{KT}^+$.

6. Numerical experiments

In this section, we aim to compare the performances of our procedure with some existing procedures under different settings. We conduct experiments on a personal computer (PC) with Intel(R) Xeon(R) Gold 6226R central processing unit (CPU) which contains 32 processors, 64 Gigabytes (GB) memory, and Ubuntu 20.04.5 LTS Linux operating system. The programming language is Java. All the code and results can be retrieved from https://github.com/biazhong/iKT. In the code, the pseudo random number seeds are specified so that all the results can be replicated. In this section, the desired PICS α is set to be 5%. Unless otherwise noted, while implementing Procedure *i*-$\mathcal{KT}$ and Procedure *i*-$\mathcal{KT}^+$, the initial-stage sample size, *i.e.*, n_0, is set to be 20. For the experiments conducted in the first subsection, we consider a simple setting where the observations are generated from normal distributions, and for the experiments conducted in the second subsection, the observations are obtained by conducting real simulation tasks.

Table 1
Comparisons of Procedure i-$\mathcal{KT}$ and Procedure $\mathcal{KT}$: normal observations.

			Problem 1		
Procedure	PCS	Total Number of Matches	Average Sample of a Match	Maximum Rounds of Selections	Total Sample Size ($\times$ 10^6)
$\mathcal{KT}$	0.954	1999 ± 0	2056 ± 2	11 ± 0	4.11 ± 0.01
i-$\mathcal{KT}$	0.955	1465 ± 12	2064 ± 15	3.55 ± 0.06	3.06 ± 0.04
			Problem 2		
Procedure	PCS	Total Number of Matches	Average Sample of a Match	Maximum Rounds of Selections	Total Sample Size ($\times$ 10^6)
$\mathcal{KT}$	0.999	1999 ± 0	840 ± 1	11 ± 0	1.68 ± 0.01
i-$\mathcal{KT}$	0.990	1051 ± 2	554 ± 8	2.68 ± 0.04	0.59 ± 0.01

6.1. Problems with normal observations

In this subsection, we consider a simple setting where the observations are drawn from normal distributions. We test the performances of Procedure i-$\mathcal{KT}$ and Procedure $\mathcal{KT}$ with two problems. For Problem 1, the means of the alternatives follow the slippage configuration, where $\mu_1 = \mu_2 = \cdots = \mu_k - \delta = 0$, and all the alternatives share a common but unknown variance 1, *i.e.*, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2 = 1$. Let $\text{Unif}\,(l, u)$ be the uniform distribution with lower bound l and upper bound u. For Problem 2, the means of the alternatives spread out and follow a configuration where $\mu_i = \text{Unif}\,(0, 16)\,\delta$, for $i = 1, 2, \ldots, k-1$, and $\mu_k = 17\delta$. The variances of the alternatives follow a configuration where $\sigma_i^2 = \text{Unif}\,(0.5, 1.5)$ for $i = 1, 2, \ldots, k$. We compare the performances of the two procedures in solving each problem by setting the number of alternatives k to be 2,000. While implementing these procedures, IZ parameter δ is set to be 0.1. No CRNs are used in this experiment. After using each procedure to solve the problems, we report the estimated PCS, the total number of matches, the average sample size of a match, the maximum rounds of selections, the total sample size, and 95% confidence intervals for the latter four estimates based on 1,000 independent macro replications in Table 1. For Procedure i-$\mathcal{KT}$, while counting the number of observations generated from alternative k_r in the rth round of the selection, we evaluate the length of set $\mathcal{X}_r$ at the end of that round.

From Table 1, we have several findings. First, while solving different problems, the two procedures can always deliver a PCS higher than the predetermined one, i.e, 0.95. Second, in terms of the total sample size, Procedure i-$\mathcal{KT}$ performs uniformly better than Procedure $\mathcal{KT}$ does in these two problems. Particularly, in Problem 1, because the means of the alternatives follow the slippage configuration, it is very difficult for Procedure i-$\mathcal{KT}$ to choose the true best alternative as the reference alternative in each round of the selection. As a result, the average sample size of a match in Procedure i-$\mathcal{KT}$ is slightly larger than that of Procedure $\mathcal{KT}$. However, Procedure i-$\mathcal{KT}$ still performs better than Procedure $\mathcal{KT}$ does in this situation because Procedure i-$\mathcal{KT}$ eliminates more alternatives than Procedure $\mathcal{KT}$ does in each round of the selection and the total number of matches of Procedure i-$\mathcal{KT}$ is much smaller than that of Procedure $\mathcal{KT}$. Third, in Problem 2, the means of the alternatives spread out. In this situation, Procedure i-$\mathcal{KT}$ can easily choose the true best alternative as the reference alternative, and on average the procedure can stop in less than three rounds of selections. In fact, while conducting the experiment, we find that in most cases, once the true best alternative is chosen as the reference alternative at the beginning of a round of selection, the procedure can finish the selection in that round.

To further investigate on how Type A matches and Type B matches affect the selection of Procedure i-$\mathcal{KT}$, while using Procedure i-$\mathcal{KT}$ to solve the two problems, we also record the average proportions of Type A matches and Type B matches in the first round of the selection, *i.e.*, $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$ for $r = 1$, the number of surviving alternatives after the first round of the selection, *i.e.*, $|\mathcal{I}_{r+1}|$ for $r = 1$, and their 95% confidence intervals. We summarize the results in Table 2.

Notice that there are 2,000 alternatives in contention at the beginning of the first round of the selection. As suggested by Proposition 3, if $\mathbf{p}_A^r = 0.001$ and $\mathbf{p}_B^r = 0.999$ for $r = 1$, on average, fewer than 750 alternatives can survive after the first round of the selection. If $\mathbf{p}_A^r = 0.829$ and $\mathbf{p}_B^r = 0.915$ for $r = 1$, on average, fewer than 171 alternatives can survive after the first round of the selection. Therefore, the results listed in Table 2 are consistent with our analysis in Proposition 3.

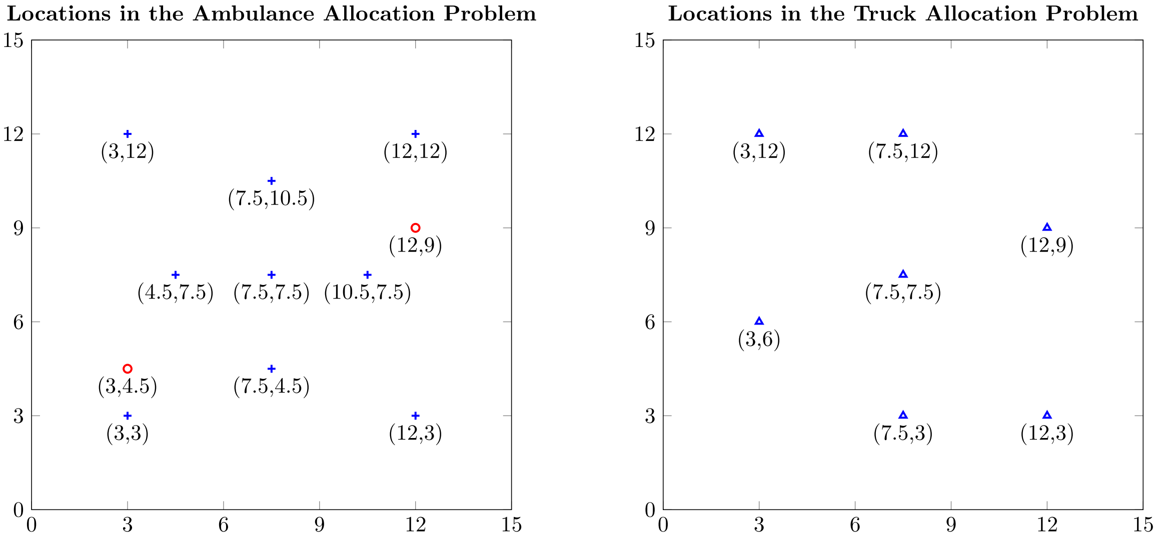
6.2. Resource allocation optimization

In practice, many resource allocation problems can be formulated as R&S problems. In these problems, one needs to decide the distribution of finite resources to a set of locations in a way that a certain performance measure of a system can be optimized. To formulate these problems as R&S problems, we can treat every possible way of allocating resources as an alternative. In this subsection, we implement Procedure i-$\mathcal{KT}$, Procedure $\mathcal{KT}$, Procedure $\mathcal{KN}$, and a well-known parallel procedure, the good selection procedure (GSP) proposed by Ni et al. (2017), to solve two resource allocation problems and compare their performances. In this subsection, the observations are generated by running real simulation tasks.

6.2.1. The ambulance allocation problem

The first problem is to allocate ambulances in the emergency medical service system. This problem is also considered in Fan et al. (2016) and Ni et al. (2012). We adopt the setting in Fan et al. (2016) with some modifications. Consider a square city area with size 15 miles by 15 miles. In the city, the distance from one point to another is measured by the Manhattan metric. There are 9 candidate ambulance bases and 2 hospitals. We try to allocate 4 ambulances to the bases. Each base can have more than one ambulance. Suppose that the inter-arrival time between two consecutive emergency calls follows exponential distribution with mean 55 min, and every emergency call can occur in any place of the square with a uniform distribution. Once a call arrives, the nearest base with at least one available ambulance responds to the call, and sends an ambulance to the call site.[5] The ambulance travels at a constant speed 24 miles per hour. Upon arriving, the ambulance crews take a random amount of time, which is exponentially distributed with mean 10 minutes, to serve the patient. After the crews serve the patient, the ambulance takes the patient to the nearest hospital and then travels back to the original base. A patient is well served if an ambulance can arrive within 20 min after s/he makes the emergency

[5] If all ambulances are dispatched, the call will not be responded immediately until there is one ambulance traveling back to its base.


Locations in the Ambulance Allocation Problem

Locations in the Truck Allocation Problem

Fig. 4. Location information in the resource allocation optimization problems.
Left: + stands for a possible location of ambulances and ∘ stands for a hospital. Right: △ stands for a possible location of trucks.

Table 2
Estimated $\mathbf{p}_A^r$, $\mathbf{p}_B^r$, and $|\mathcal{I}_{r+1}|$ in Procedure i-$\mathcal{KT}$ for $r = 1$: normal observations.

	Proportion of type A matches: $\mathbf{p}_A^r$	Proportion of type B matches: $\mathbf{p}_B^r$	Number of surviving alternatives after the first round: $|\mathcal{I}_{r+1}|$
Problem 1	0.001 ±0.002	0.999 ±0	694 ±17
Problem 2	0.829 ±0.004	0.915 ±0.004	89 ±4

call. For this problem, there are 495 allocation policies (alternatives), and we aim to select the policy which maximizes the expected fraction of well served calls. The locations of hospitals and candidate ambulance bases are shown in the left panel of Fig. 4. While simulating an observation, we warm up the system with 250 calls. After the first 250 calls are responded, we observe the fraction of well served calls for the subsequent 250 calls.

Since the size of the problem is relatively small, we use Procedure i-$\mathcal{KT}$, Procedure $\mathcal{KT}$, and Procedure $\mathcal{KN}$ to solve the problem in a single-processor computing environment. As discussed in Zhong and Hong (2022), properly pairing the surviving alternatives and allowing slightly more alternatives to participate in a match can improve the performance of Procedure $\mathcal{KT}$. In this experiment, besides the original $\mathcal{KT}$ procedure, we also consider two variants of Procedure $\mathcal{KT}$. We call the variants Procedure $\mathcal{KT}^s$ and Procedure $\mathcal{KT}^*$ respectively. The only difference between Procedure $\mathcal{KT}$ and Procedure $\mathcal{KT}^s$ is that, for Procedure $\mathcal{KT}^s$, while scheduling the matches in the rth round of the selection, we first sort the surviving alternatives in a descending order based on their sample means calculated by using the observations obtained prior to that round. Then, we pair alternatives $\mathcal{I}_r[1]$ and $\mathcal{I}_r[|\mathcal{I}_r|/2+1]$, alternatives $\mathcal{I}_r[2]$ and $\mathcal{I}_r[|\mathcal{I}_r|/2+2]$, and so on.[6] By doing so, we lower the chance that two equally "bad" alternatives are allocated to the same match. Similar to Procedure i-$\mathcal{KT}$, to estimate the sample means of different alternatives in the first round of the selection, we generate 20 observations from each alternative before the selection starts. The only difference between Procedure $\mathcal{KT}$ and Procedure $\mathcal{KT}^*$ is that we allow three alternatives to participate in a match in Procedure $\mathcal{KT}^*$.

Table 3
Comparisons of Procedures i-$\mathcal{KT}$, $\mathcal{KT}$, $\mathcal{KT}^s$, $\mathcal{KT}^*$, and $\mathcal{KN}$: ambulance allocation problem.

Procedure	PCS	Total sample size (× 10^5)	Wall-clock time (s)
i-$\mathcal{KT}$	0.97	1.29 ±0.07	12.18 ±0.67
$\mathcal{KT}$	0.98	7.00 ±0.15	64.62 ±1.42
$\mathcal{KT}^s$	0.95	3.83 ±0.06	34.42 ±0.51
$\mathcal{KT}^*$	0.99	5.30 ±0.12	48.09 ±1.05
$\mathcal{KN}$	0.99	1.96 ±0.06	18.03 ±0.52

In this experiment, no CRNs are used, IZ parameter δ is set to be 0.001, and the first-stage sample size n_0 is set to be 20. Via Monte Carlo simulation, we find that the optimal policy is to allocate one ambulance to each of the bases $(3.0, 3.0)$, $(7.5, 10.5)$, $(12.0, 3.0)$, and $(12.0, 12.0)$, and its mean performance is 0.8165. There is no other policy whose mean performance is within 0.001 to that of the best policy. After using every procedure to solve the problem, we report the estimated PCS, the wall-clock time with 95% confidence interval, and the total sample size with 95% confidence interval based on 100 independent macro replications in Table 3.

From Table 3, we can find that all five procedures can deliver a PCS higher than the predetermined one. In terms of the total sample size and wall-clock time, Procedure i-$\mathcal{KT}$ performs better than the other four procedures do. Comparing Procedure $\mathcal{KT}$, Procedure $\mathcal{KT}^s$, and Procedure $\mathcal{KT}^*$, we can conclude that, adopting a proper pairing method for the surviving alternatives in each round of the selection and allowing slightly more alternatives to participate in a match indeed improve the performance of Procedure $\mathcal{KT}$. However, Procedure $\mathcal{KT}^s$

[6] Such a pairing method may not work well in Procedure i-$\mathcal{KT}$ because the pairing method is not designed for the purpose of maximizing the values for $\mathbf{p}_A^r$ and $\mathbf{p}_B^r$.

and Procedure $\mathcal{KT}^*$ still perform worse than Procedure i-$\mathcal{KT}$ does. Lastly, Procedure $\mathcal{KN}$ is often considered to be an efficient procedure to solve small-scale problems in single-processor computing environments. Procedure i-$\mathcal{KT}$ outperforms Procedure $\mathcal{KN}$ in this experiment, demonstrating its competitiveness even in solving small-scale problems. The good performance of Procedure i-$\mathcal{KT}$ in terms of the total sample size can be explained as follows. In Procedure i-$\mathcal{KT}$, because each match only involves three alternatives, compared to Procedure $\mathcal{KN}$, the sizes of the continuation regions used to compare alternatives are relatively small. Therefore, if the reference alternative has a relatively large mean, Procedure i-$\mathcal{KT}$ tends to remove inferior alternatives with a few observations. Furthermore, with the pairing method discussed in Section 4.2, Procedure i-$\mathcal{KT}$ can eliminate as many alternatives as possible in each round of the selection and terminates in a few rounds. In such a case, the procedure greatly mitigates the negative impact of throwing away of observation data after each round of selection on the total sample size.

6.2.2. The truck allocation problem

Suppose that a company selling groceries has 15 trucks and wants to allocate the trucks to its 7 warehouses in the city area which is the same as that in the ambulance allocation problem. The locations of the warehouses are shown in the right panel of Fig. 4, and each warehouse can have more than one truck. The demand call occurs uniformly in the city area from 8 AM to 8 PM. The inter-arrival time between two consecutive calls follows exponential distribution with mean 4 minutes. Once a call arrives, the nearest warehouse with at least one available truck responds to the call and prepares the package that needs to be delivered. The preparation time follows exponential distribution with mean 2 min. After the package is prepared, a truck is sent to the call site. The truck travels at the speed 20 miles per hour. Upon arriving at the call site, it takes the truck driver some additional time to unload the package. The unloading time follows exponential distribution with mean 2 min. After the driver successfully unloads the package, the truck returns to the warehouse where it is originally allocated. In this problem, there are 54,264 ways to allocate the trucks, *i.e.*, 54,264 alternatives, and we aim to select the alternative that minimizes the expected customer waiting time, *i.e.*, the time between a demand call occurring and the truck driver successfully unloading the package. Monte Carlo simulation is adopted to evaluate the mean performance of each alternative. While generating an observation for an alternative, we simulate the calls in a day and record the average customer waiting times of the calls. Because for this problem the number of alternatives is relatively large, parallel computing is adopted to solve the problem. All the 32 processors in the PC are used to solve this problem. The underlying parallel computing platform is Apache Spark.

We compare Procedure i-$\mathcal{KT}^+$ with Procedure $\mathcal{KT}$ and GSP in this experiment. IZ parameter δ is set to be 1 minute. To investigate how CRNs can improve the performance of Procedure i-$\mathcal{KT}^+$, we consider two versions of Procedure i-$\mathcal{KT}^+$, *i.e.*, Procedure i-$\mathcal{KT}^+$ with CRNs and Procedure i-$\mathcal{KT}^+$ without CRNs. For GSP, the batch-size parameter is set to be 200 as recommended by the authors. The IZ assumption is violated in this problem. There are multiple alternatives whose mean performances are within 1 min to that of the best alternative. As long as a procedure can select an alternative whose mean performance is within 1 min to that of the best, we count it as a correct selection. After using the procedures to solve the problem, we report the estimated PGS, the total sample size, the utilization, the wall-clock time, and 95% confidence intervals for the latter three estimates based on 100 independent macro replications in Table 4. Notice that the utilization measures how efficient a procedure uses processors to simulate observations, and is calculated as follows,

$$\text{Utilization} = \frac{\text{total simulation time across all processors}}{\text{number of processors} \times \text{wall-clock time}}.$$

The larger value for the utilization, the less time the processors tend to stay idle.

From Table 4, we can observe that all four procedures deliver a PGS higher than the predetermined one in this problem. From the aspects of the total sample size and the wall-clock time, Procedure i-$\mathcal{KT}^+$ (without CRNs) performs much better than Procedure $\mathcal{KT}$ and GSP do. In particular, the total sample size and the wall-clock time of Procedure i-$\mathcal{KT}^+$ are only about one sixth of that of Procedure $\mathcal{KT}$ and one fourth of that of GSP. The utilization of Procedure i-$\mathcal{KT}^+$ is slightly smaller than those of Procedure $\mathcal{KT}$ and GSP. However, the utilization of Procedure i-$\mathcal{KT}^+$ is still above 85%. We may conclude that Procedure i-$\mathcal{KT}^+$ is indeed very suitable for solving large-scale problems in parallel computing environments. Comparing Procedure i-$\mathcal{KT}^+$ (with CRNs) and Procedure i-$\mathcal{KT}^+$ (without CRNs), we may conclude that in this experiment, once CRNs are used, the total sample size and the wall-clock time of Procedure i-$\mathcal{KT}^+$ are reduced a lot.

In Table 4, we do not include the results of Procedure $\mathcal{KN}$ because it does not work well on Apache Spark. However, as a reference, we implement the procedure in a single-processor computing environment and use it to solve the problem. Based on 100 independent macro replications, we find that the average sample sizes of Procedure $\mathcal{KN}$ with CRNs and without CRNs are 2.36×10^6 and 7.86×10^6 respectively. Similar to the results listed in Table 3, the total sample size of Procedure $\mathcal{KN}$ is still larger than that of Procedure i-$\mathcal{KT}$ if no CRNs are used. However, the total sample size of Procedure i-$\mathcal{KT}$ is slightly larger than that of Procedure $\mathcal{KN}$ if CRNs are used. It is because CRNs make the problem much easier to solve and Procedure i-$\mathcal{KT}$ requires more than 10^6 observations to conduct initial stage sampling. Moreover, there is also some efficiency loss on the total sample size of Procedure i-$\mathcal{KT}$ while implementing it in parallel computing environments.

7. Concluding remarks

In this paper, we have developed Procedure i-$\mathcal{KT}$ that improves upon Procedure $\mathcal{KT}$ proposed by Zhong and Hong (2022). The idea is to add a well chosen reference alternative to every match and to carefully pair the surviving alternatives in each round of the selection. By doing so, Procedure i-$\mathcal{KT}$ tends to outperform Procedure $\mathcal{KT}$ from the aspects of the average sample size of a match and the total number of matches, and thus achieves overall sample size and wall-clock time reductions. The numerical results verify our theoretical analysis and show the competitiveness of our Procedure i-$\mathcal{KT}$ on solving various problems.

CRediT authorship contribution statement

Ying Zhong: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Jianzhong Du:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Deng-Feng Li:** Writing – review & editing, Writing – original draft, Validation, Resources, Project administration, Investigation, Formal analysis, Conceptualization. **Zhaolin Hu:** Writing – review & editing, Writing – original draft, Validation, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Acknowledgments

We appreciate the editor Emanuele Borgonovo and the anonymous referees for the valuable comments and suggestions that helped significantly improve the paper. This research was partially supported by the National Natural Science Foundation of China [Grants 72101047, 72071148, 72021002] and the Natural Science Foundation of Sichuan (Grants 2023NSFSC0516).

Table 4
Comparisons of Procedures i-$\mathcal{KT}^+$, $\mathcal{KT}$, and GSP in a parallel computing environment with 32 processors: truck allocation problem.

Procedure	PGS	Total sample size ($\times 10^6$)	Wall-clock time (s)	Total simulation time ($\times 10^2$ s)	Utilization
i-$\mathcal{KT}^+$ (with CRNs)	1.00	3.28 ±0.14	30.44 ±1.37	8.70 ±0.39	89.40% ±0.35%
i-$\mathcal{KT}^+$ (without CRNs)	0.99	5.52 ±0.24	51.26 ±2.39	14.18 ±0.68	86.42% ±0.75%
$\mathcal{KT}$	1.00	32.86 ±0.03	288.24 ±5.77	83.52 ±1.29	90.80% ±0.71%
GSP	1.00	21.34 ±0.19	194.57 ±2.99	54.49 ±0.73	87.62% ±0.50%

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2024.08.031, which includes the proofs of the lemmas, propositions and theorems in the article.

References

Andradóttir, S., & Lee, J. S. (2021). Pareto set estimation with guaranteed probability of correct selection. *European Journal of Operational Research*, *292*(1), 286–298.

Bechhofer, R. E. (1954). A single-sample multiple decision procedure for ranking means of normal populations with known variances. *The Annals of Mathematical Statistics*, *25*(1), 16–39.

Chen, C. H., Chick, S. E., Lee, L. H., & Pujowidianto, N. A. (2015). Ranking and selection: efficent simulation budget allocation. In M. Fu (Ed.), *Handbook of simulation optimization* (pp. 45–80). Springer.

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

Chen, Z., Luo, J., & Wu, R. (2023). On the finite-sample statistical validity of adaptive fully sequential procedures. *European Journal of Operational Research*, *307*(1), 266–278.

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

Fan, W., Hong, L. J., & Nelson, B. L. (2016). Indifference-zone-free selection of the best. *Operations Research*, *64*(6), 1499–1514.

Frazier, P. I. (2014). A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operations Research*, *62*(4), 926–942.

Frazier, P. I., Powell, W., & Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, *47*(5), 2410–2439.

Glynn, P., & Juneja, S. (2004). A large deviations perspective on ordinal optimization. In R. G. Ingalls, M. D. Rossetti, J. S. Smith, & B. A. Peters (Eds.), *Proceedings of the 2004 winter simulation conference* (pp. 577–585). Piscataway, New Jersey: IEEE.

Groves, M., & Branke, J. (2019). Top-κ selection with pairwise comparisons. *European Journal of Operational Research*, *274*(2), 615–626.

Hong, L. J. (2006). Fully sequential indifference-zone selection procedures with variance-dependent sampling. *Naval Research Logistics*, *53*(5), 464–476.

Hong, L. J., Fan, W., & Luo, J. (2021). Review on ranking and selection: A new perspective. *Frontiers of Engineering Management*, *8*(3), 321–343.

Hunter, S. R., & Nelson, B. L. (2017). Parallel ranking and selection. In *Advances in Modeling and Simulation* (pp. 249–275). Springer.

Kim, S. H., & Nelson, B. L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, *11*(3), 251–273.

Kim, S. H., & Nelson, B. L. (2006a). Selecting the best system. In S. G. Henderson, & B. L. Nelson (Eds.), *Handbooks in operations research and management science: Simulation* (pp. 501–534). Elsevier.

Kim, S. H., & Nelson, B. L. (2006b). On the asymptotic validity of fully sequential selection procedures for steady-state simulation. *Operations Research*, *54*(3), 475–488.

Lee, M. L., Park, C., & Park, D. U. (2018). Self-adjusting the tolerance level in a fully sequential feasibility check procedure. *European Journal of Operational Research*, *271*(2), 733–745.

Luo, J., Hong, L. J., Nelson, B. L., & Wu, Y. (2015). Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. *Operations Research*, *63*(5), 1177–1194.

Ni, E. C., Ciocan, D. F., Henderson, S. G., & Hunter, S. R. (2017). Efficient ranking and selection in parallel computing environments. *Operations Research*, *65*(3), 821–836.

Ni, E. C., Hunter, S. R., Henderson, S. G., & Topaloglu, H. (2012). Exploring bounds on ambulance deployment policy performance. In *Proceedings of the 2012 winter simulation conference* (pp. 1–12). IEEE.

Paulson, E. (1964). A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, *35*(1), 174–180.

Pei, L., Nelson, B. L., & Hunter, S. R. (2022). Parallel adaptive survivor selection. *Operations Research*, *72*(1), 336–354.

Rinott, Y. (1978). On two-stage selection procedures and related probability-inequalities. *Communications in Statistics. Theory and Methods*, *A7*, 799–811.

Tamhane, A. C. (1977). Multiple comparisons in model i: One-way anova with unequal variances. *Communications in Statistics. Theory and Methods*, *6*, 15–32.

Yoon, M., & Bekker, J. (2019). Considering sample means in Rinott's procedure with a Bayesian approach. *European Journal of Operational Research*, *273*(1), 249–258.

Zhong, Y., & Hong, L. J. (2022). Knockout-tournament procedures for large-scale ranking and selection in parallel computing environments. *Operations Research*, *70*(1), 432–453.

Zhong, Y., Liu, S., Luo, J., & Hong, L. J. (2022). Speeding up Paulson's procedure for large-scale problems using parallel computing. *INFORMS Journal on Computing*, *34*(1), 586–606.