# Adaptive simulation budget allocation in Simulation assisted Differential Evolution algorithm

**Author:**
Akl, AM; Sarker, RA; Essam, DL

**Publication details:**
Applied Soft Computing Journal
v. 83
1568-4946 (ISSN); 1872-9681 (ISSN)

**Publication Date:**
2019-10-01

**Publisher DOI:**
https://doi.org/10.1016/j.asoc.2019.105678

**License:**
https://creativecommons.org/licenses/by-nc-nd/4.0/
Link to license to see what you are allowed to do with this resource.

Downloaded from http://hdl.handle.net/1959.4/unsworks_66193 in https://unsworks.unsw.edu.au on 2024-05-01

# Adaptive Simulation Budget Allocation in Simulation Assisted Differential Evolution Algorithm

Amany M. Akl*, Ruhul A. Sarker and Daryl L. Essam
School of Engineering and Information Technology
University of New South Wales at Canberra
Canberra 2600, Australia
Email: {r.sarker, d.essam} @adfa.edu.au
*Corresponding author email: amany.mamdouh@student.adfa.edu.au
*Corresponding author phone: +61415680784

## Abstract

Recently, the literature on Simulation assisted Optimization for solving stochastic optimization problems has been considerably growing. In the optimization context, the population based meta-heuristics algorithms, such as, Differential Evolutionary (DE), has shown tremendous success in solving continuous optimization problems. While in the simulation context, Monte-Carlo Simulation for Sample Average Approximation is one of the successful approaches in handling the stochastic parameters of such problems. However, the intertwined computational burden, when combining these two approaches is amplified, and that encourages new research in this topic. In this problem, the challenge is to maintain high quality stochastic solutions by minimizing the computational cost to a reasonable level. To deal with this challenge, we propose a novel Adaptive Segment Based Scheme (ASBS) algorithm, for allocating the MCS budget in a Simulation assisted Differential Evolution (Sim-DE) Algorithm. This allows the algorithm to adaptively control the simulation budget based on a performance measure**.** The performance of the proposed ASBS algorithm is compared with other simulation budget allocation techniques while using the same base algorithm. The experimental study has been conducted by solving a modified set of IEEE-CEC'2006 test problems and a wind-thermal power systems application. The experimental results reveal that the ASBS algorithm is able to substantially reduce the simulation budget, with an insignificant effect in solution quality.

Keywords: Simulation Optimization; Evolutionary Computations; Adaptive simulation budget; Stochastic Programming

## 1. Introduction

Many real-world decisions are made through optimization problem solving. These problems usually contain functional constraints and some of the parameters in those problems may be stochastic in nature. These problems are recognized as Stochastic Constrained Optimization Problems (SCOPs). The stochastic parameters may occur, either in the objective function, or in constraints functions or in both. In this research, we focus on studying such problems with the involvement of stochastic parameters in the objective function, as shown in the following equations (1)

$$\underset{x \in X}{\text{Min}} \; f = E(f(\vec{x}, \vec{\xi}_N^L))$$

$$subject\ to:$$

$$q_k(\vec{x}) \leq 0, \qquad k = 1,2, \dots, K$$

$$h_e(\vec{x}) = 0, \qquad e = 1,2, \dots, E$$

$$\underline{x}_j \leq x_j \leq \overline{x}_j, \quad j = 1,2, \dots, D \tag{1}$$

Where $\vec{x} = x_1, x_2, \dots, x_D$ is a vector of *D* decision variables defined on a continuous domain of real values. $E(f(\vec{x}, \vec{\xi}_N^L))$ is the expected value of the stochastic objective function with *L* stochastic parameters, each of size *N* independent identically distributed (*iid*) samples/scenarios of the random vector ξ, with given probability distribution *P*, $q_k(\vec{x})$ is the $k^{th}$ inequality constraint, $h_e(\vec{x})$ is the $e^{th}$ equality constraint and each $x_j$ has a lower limit ($\underline{x}_j$) and an upper limit ($\overline{x}_j$).

Large-scale stochastic optimization problem have inherent analytical complexities and high computational requirements. Simulation Optimization (Sim-Opt) is a prominent paradigm in solving such problems through combining two inveterate approaches, a detailed Sim-Opt review can be found in [1-3]. It has been adopted in many real and substantial applications such as, for production sectors and water distribution systems as proposed in [4, 5].

Various Optimization and Simulation techniques have been intertwined together to solve stochastic optimization problems. For instance, Meta-heuristics such as, population based Evolutionary Algorithms (EAs), are frequently employed to solve stochastic problems, which is recognized as Sim-heuristics in [6, 7]. Among the EAs, Differential Evolution (DE) [8] has become very popular, due to its excellent performance in solving complex optimization problems [9]. DE is a population-based stochastic algorithm, its algorithmic steps start with Initialization along with Mutation, Crossover and Selection. These steps are managed through multiple DE control parameters and operators. Due to the importance of improving the performance of DE, many studies have proposed different ways of self-adaptively managing these control parameters and/or a mix of operators of DE, such as [9-11]. Recently Elsayed et al. [12] proposed an enhanced version of the Self Adaptive Multi-Operator DE (SAMO-DE) algorithm that dynamically emphasizes on the best-performing DE variants, based on the quality of the fitness values and/or constraint violations. This approach was found to outperform the state-of-the-art algorithms by introducing new improvement measures and putting more emphasis on the best performing DE operators [12].

DE has been extensively studied for solving constrained problems in the deterministic context, but it requires further study to be adopted in the stochastic context. To deal with the stochastic parameters (which can be uncertainty with stochastic behaviour) within such population based algorithms, it is well-accepted to integrate a simulation based approach. Among the simulation approaches, Monte-Carlo Simulation (MCS) is popular and has been successfully employed in different applications for stochastic optimization [13-15]. Note that MCS is frequently used with Sample Average Approximation (SAA) [13, 16] to evaluate an approximated expected value for the stochastic objective function, as in (1). The evaluated expected values are used as feedback to EA's evolution process, for effectively solving this type of problem through a hybrid Sim-Opt framework, as studied in [17].

It is worthwhile to mention here that the simulation and optimization approaches are applied independently with very little interaction between them [18], as simulation focuses on the evaluation of the stochastic function with the values of the decision variables given from the optimization process, while using specific sample size or number of simulation replications. While the optimization process focuses on searching for better values of the decision variables, while considering the simulation evaluation process as a black box. In this context, the Optimal Computing Budget Allocation (OCBA) techniques [19] and their enhanced versions, such as OCBAm+ [20], have studied the allocation of the simulation budget to different alternatives/solutions as an independent study. However, they suggested that the budget allocation techniques can be combined with population based optimization algorithms, such as DE [20], where the solutions at each specific generation can be considered as a set of solutions for the OCBA algorithm. This approach has been studied in [17, 21].

OCBAm+ operates as an optimization technique to maximize the probability of the solution's Correct Selection P(CS), which is asymptotically estimated, based on multiple assumptions [19]. A sequential heuristic procedure is used to apply the allocation rule given by OCBAm+, its details are in [20]. Therefore, it is observed that using OCBAm+ at each generation of the EA leads to another optimization issue. This nested optimization problem adds further computational complexity to the overall Sim-DE framework. Furthermore, recent studies have provided a remarkable hypothesis of non-monotonicity of P(CS), which means that an increase in sample size will not necessarily lead to higher P(CS) [22].

There are a few papers that have studied different simulation budgets and their adoption in SAA from the simulation point of view [23]. The Sim-Opt joint paradigm is growing and requires further study, especially on the mutual benefits from the joint exploitation of the two approaches. Obviously, there are indications that the complexity of the Monte-Carlo Simulation grows faster with any increase of the number of generations or the population size of the EAs, which amplifies the overall computational budget enormously [6, 24]. Based on our preliminary investigation on the number of segments, that are representative enough for producing a good quality solution, it was found that five was the best [25]. The results are supported by the five-number summary statistical concept proposed in [26-28]. The five-number summary is a set of descriptive statistics that provide information about a dataset. It consists of the five most important sample percentiles: the sample minimum (smallest observation), the lower quartile or first quartile, the median (the middle value), the upper quartile or third quartile, and the sample maximum (largest observation) [28]. However, although this method showed efficient performance, the segments' sizes remained fixed. Consequently, the challenge of choosing appropriate sample sizes adaptively, is an important component in our algorithm framework. Inappropriate sample size (or simulation budget) can lead to low quality solutions, as much time can be spent in simulation, rather than optimization [23]. Thus, we particularly focus our research on this issue, as we aim to analyse the evolutionary process, to efficiently and adaptively, determine the simulation budget.

To address this challenge, this paper proposes a new approach to enhance the performance of the basic MCS for SAA, by incorporating an Adaptive Segment Based Scheme (ASBS) strategy within a proposed Sim-Opt algorithmic framework. In basic SAA, a constant "conservative" large number of scenarios is generated in each iteration and the objective function is evaluated iteratively until the Sim-DE termination condition is reached. In our approach, the large number of scenarios is considered as an initial sample size and ASBS is employed to reduce this large sample size into a small number of segments. We assume that most representative segments can be extracted from this initial large sample after ordinal transformation by our novel Segments Extraction Technique (SET). The size of each segment, called Window Size (WS), is adaptively determined, based on its performance within the evolutionary algorithm. We then represent those segments as scenarios and use them to solve stochastic problems by our proposed Simulation assisted Differential Evolutionary framework (Sim-DE).

ASBS is a novel idea, that can be considered as an inspiration from the importance sampling concept for MCS [29], which transforms the probability distribution to a biased distribution that emphasizes important values. However, unlike importance sampling, ASBS keeps the original distribution and emphasizes on important and representative values, from an initial large sample of scenarios. It consequently forms a smaller and representative sample of scenarios.

It is different from prior studies where the sample size is kept either fixed or increased monotonically [17, 23]. Our approach provides the opportunity to adaptively control the size of the scenarios, based on the computational performance obtained through Sim-DE iterations.

This approach is experimentally validated in the context of the constrained benchmark test problems of IEEE-CEC'2006 [30], with modified stochastic objective functions, as shown in Appendix A. A comparison is performed between the proposed ASBS, basic SAA and OCBAm+ for simulation budget allocation, where the results are compared based on the Optimality Gap (OG), simulation budget, computational time and significance test analysis. A well-known real-world problem such as the Dynamic Economic Dispatch (DED) wind-thermal problem [23, 31], is solved to demonstrate the usefulness of the proposed algorithm. From the overall benchmark results, the proposed approach showed its superiority in terms of reducing the simulation budget by 90% and the computational time by 42%, while keeping OG to a minimum value compared to others, which reveals the highest solutions quality.

## 2. Proposed methodology

In this paper, a new Simulation assisted Differential Evolution (Sim-DE) algorithm framework is proposed. It combines a DE algorithm, known as SAMO-DE [12], and Monte-Carlo Simulation (MCS) [15] for SAA, with our proposed Adaptive Segment Based Scheme (ASBS) and Segments Extraction technique (SET). For convenience of explaining the algorithm, we define a number of notations, as shown in Table 1.

Table 1. Notations

| Notation | Description |
|---|---|
| $R$ | Total number of independent runs of Sim-DE, where *r* =1, .., *R*. This generates *R* best solutions (/near optimal solutions), one from each run |
| $G$ | Maximum number of Sim-DE generations in each run *r* |
| $PS$ | Population size, the number of individuals |
| $N_{max}$ | The maximum number for the scenarios (generated based on uncertain parameters) that can be used to evaluate the individuals in each generation |
| $N_{indv}$ | A subset of scenarios used to evaluate each individual, where the upper bound is $N_{max}$ |
| $T$ | Total simulation budget for all individuals in a population per generation,<br>Where $T = PS * N_{indv}$ (2) |
| TCB | Total Computational Budget for function evaluations of Sim-DE,<br>where $TCB = G * T$ (3) |
| $(\mu_{ir})$ | $E\big(f(\overline{X_{ir}})\big) = \sum_{j=1}^{N_{indv}} \left(\frac{f(X_{ijr})}{N_{indv}}\right)$, $Expected\ fitness\ of\ the\ i^{th}\ solution\ of\ the\ r^{th}\ run,$<br>$where\ i = 1,..,PS, r = 1,\dots,R$ |

## 2.1. Adaptive Segment Based Scheme (ASBS) and Segments Extraction Technique (SET)

ASBS is designed to adaptively choose a subset of scenarios, to reduce the simulation budget, based on the algorithm's performance. It aims to encapsulate important inferences about the expected fitness values, to efficiently guide the DE algorithm in the search space. ASBS allows Sim-DE to adaptively determine the scenario size $N_{indv}$ and the chosen scenarios, as shown in Algorithm1. Specifically, we use an ordinal transformation and then segment a large sample size to extract few smaller segments of the scenarios, by using the proposed SET in Algorithm 2. The selected segments are then combined together to form a useful subset of scenarios. We then use these scenarios to obtain an upper bound for our SAA problem, which is also an upper bound for the underlying problem.

To illustrate the fundamental idea behind ASBS, we have considered an example stochastic problem with four decision variables ($x_1$, $x_2$, $x_3$, and $x_4$). The fitness component for each variable (i.e. $f(x_1)$, $f(x_2)$, $f(x_3)$, and $f(x_4)$) is independent and the overall fitness value is the simple sum of the four components. The problem has been solved and its fitness has been evaluated for ($N_{max} = 1000$) independently and identically distributed (*iid*) scenarios following a Normal distribution. In Figure 1, we plotted the obtained $f(x_1)$ against the scenario numbers (in the order of scenario generation), which does not show any specific pattern of behaviour. In Figure 2, all four fitness components were plotted after an ordinal transformation, which shows interesting patterns of behaviour for each variable against the scenarios. Ordinal transformation is concerned with the ordering of the scenarios, generated based on the stochastic parameters, from lowest to highest fitness, as shown in eq. (4), where a list of $l^{th}$ scenarios is scaled in increasing order.

$$\uparrow \xi_N^l = \left(\xi_i^l, such\ that\ \xi_i^l \leq \xi_{i+1}^l, \forall\ i = 1,..,N\right)\ where\ l = 1,..,L, \tag{4}$$

The level of contribution is different for different variables and some scenarios are more sensitive than others. Based on the insights uncovered in Figure 2, we propose the idea of using certain segments of scenarios (as a sub-set) instead of all 1000 scenarios for simulation replications. As an example, we have shown five segments in Figure 2, that are assumed to be representative scenarios for our purpose (say, two extreme segments because of their high sensitivity and three in between). While the algorithm is running, the segment size (also called here as window size, WS) can be varied adaptively, based on the performance of the segments.

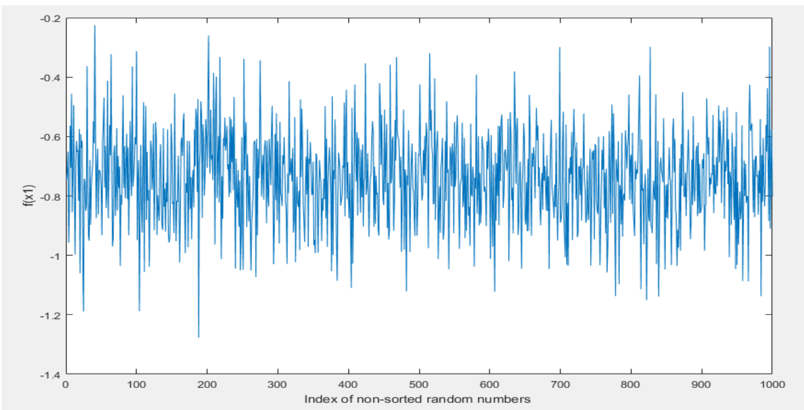


Figure 1. Fitness values of $f(x_1)$ by $N_{max}$ non-ordinal scenarios

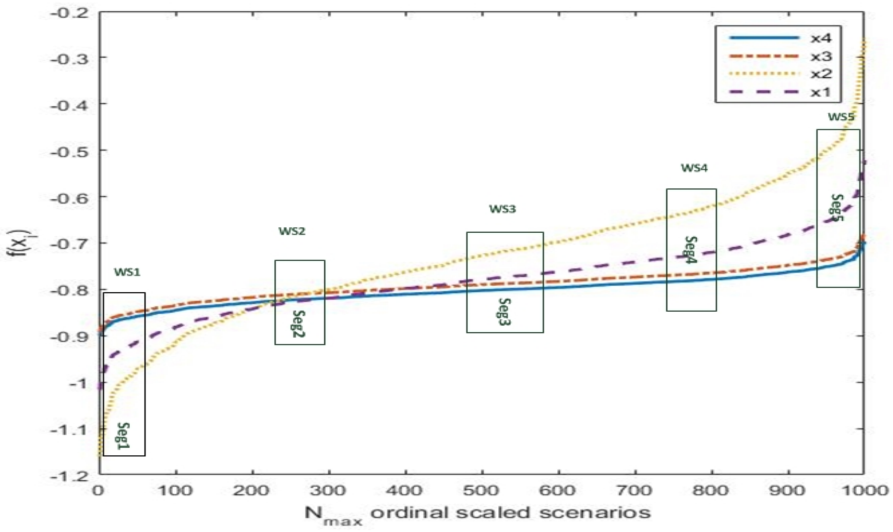


Figure 2. ASBS over fitness values of $f(x_i)$ by $N_{max}$ ordinal scaled scenarios

Algorithm1 illustrates the ASBS process. Based on the solutions obtained after every 100 generations, a Fitness Test (FT) is performed for each segment of NS segments. Depending on the relative performances of the segments, the number of individuals in the segments can be changed adaptively (increased or decreased) with a predetermined step size. In the FT, the expected fitness for the PS individuals is re-evaluated using $N_{max}$ scenarios and the best individual is selected, along with its index, as the best index. This best index is then compared with the best individual's index, that is selected after decreasing each segment's WS independently by a predetermined step. If the best indices' selection is matched, then this WS decrement is kept, otherwise the WS is penalized with an increased step. Then the rectified WS for all segments is determined and assigned to SET Algorithm 2, to extract the new representative segments of scenarios.

Algorithm 2 illustrates the steps of the representative Segments Extraction Technique (SET), where $N_{max}$ initial (*i.i.d.*) scenarios are generated by following a given probability distribution. These scenarios are ordinal scaled, as shown in Fig.2. Five NS representative and sparse segments, each with predetermined WS by using Algorithm 1, are extracted from the ordinal transformed initial large sample. The start points of these segments are located around the first quartile, median and third quartiles of the initial data sample, as well as at the beginning and end of it to include extreme scenarios. It is assumed that by using ordinal transformation and quartiles' perspective, with the five-number summary concept discussed earlier, the potential of targeting the most representative and sparse scenarios is heightened for these ranges, as shown in Fig.2. Consequently five segments (NS=5) are targeted for extraction of sample scenarios, as that showed superior performance over others, in our preliminary investigation [25]. Afterwards these segments are combined into a subset of scenarios (C) with size $N_{indv}$ , $C_{Nindv} \in \uparrow \xi^{l}_{Nmax}$, $here( N_{indv} \leq N_{max})$

The adaptive segments' WS attempts to capture enough fitness values' variations from important segments, for directing the DE algorithm with an efficient simulation budget. ASBS and SET allow the DE to determine the sizes of the segments adaptively, based on their performance, and to emphasize and extract important segments that require high WS, and efficiently save the simulation budget through the segments that can be represented by a low WS.

Algorithm 1: steps of applying proposed ASBS

Input: WS, $N_{max}$
Every 100 generations: perform Fitness Test (FT)
  Re-evaluate PS individuals with $N_{max}$ scenarios, and record best individual index (BI)
  Step ←Determine decrement or increment step for each WS
  **for** i =1 to NS segments ←Test reduction in each WS respectively
    Re-evaluate population of individuals with reduced $\mathrm{WS(i)} = \mathrm{WS(i)} - \mathrm{Step}$
    Record each segment's new best individual index SBI(i)
    Compare best individual indices respectively
    **if** $\mathrm{BI} == \mathrm{SBI(i)}$
      Update $\mathrm{WS(i)}$ with the reduction
    **else**
      Penalize $\mathrm{WS(i)} = \mathrm{WS(i)} + \mathrm{Step}$
    **end** if
  **end** for
**Return** updated $(WS)$

Algorithm 2: steps of representative SET

Input: WS, Probability Distribution (PD), $N_{max}$
RS ← generate an initial i.i.d. sample of $N_{max}$ scenarios given a PD.
SRand ← ordinal scaled/sorted list of the initial (RS).
$S_n$ ← [1,.., NS] segment's number n=1,…NS.
Subset C ← segments selection and subset formation

$$Subset\ C = \left[SRand\left(1{:}\,W(S_1)\right), SRand\left(i{:}\,i + WS(S_2)\right), SRand\left(j{:}\,j + WS(S_3)\right), SRand\left(k{:}\,k + WS(S_4)\right), SRand(N_{max} - WS(S_5){:}\,N_{max})\right], where\ \ i = q1 * N_{max}, j = q2 * N_{max}, k = q3 * N_{max}$$

$q1, q2\ and\ q3\ are\ percentiles\ within\ first, median\ and\ third\ quartile\ ranges\ respectively$

$N_{indv}$← Calculate the total number of scenarios, $N_{indv} = \sum_{i=1}^{5} WS(i)\ , where\ \ N_{indv} < N_{max}$

return $(N_{indv}\ , Subset\ C)$

## 2.2. The Proposed Algorithm

The flowchart of the proposed simulation optimization (Sim-DE) algorithm is presented in a flowchart, as shown in Figure3. It is mainly divided into three components, the evolutionary component, the simulation component of MCS along with ASBS and SET (Algorithms 1 and 2), and the performance measure component.

Sim-DE starts with an initial random population of size PS, where the individuals are generated within the lower and upper bounds of *X*. As discussed earlier, the process starts with five equal segments. The expected fitness value is evaluated for each individual by using the simulation component. The simulation component includes ASBS Algorithm 1, SET Algorithm 2 and MCS for SAA. The algorithm then continues with the DE component which applies self-adaptive crossover and mutation to generate the new population, based on the improvement measure, as suggested in SAMO-DE [12]. Improvement and diversity measurements for the self-adaptive mutation and crossover rates for the evolutionary process is one of the key aspects. The parameters F and CR depend on a historical memory of the successful values of the operators, while the improvement and diversity measurements assess the improvement in each operator over generations and adopts the successful values. The evolutionary selection process between any offspring and its parent follows one of three scenarios: (1) for two infeasible solutions, the one with a smaller sum of constraint violations is selected; (2) a feasible point is always better than an infeasible one; (3) for two feasible solutions, the fittest one (according to the expected fitness value) is selected. Consequently, SAMO-DE algorithm has the feature of implicitly minimizing the violation function through the selection process and hence targeting the minimum violation. Where the sum of constraints violation function is calculated as follows [12].

$$\psi_z = \sum_{k=1}^{K} \max(0, g_k(\overrightarrow{x_z})) + \sum_{e=1}^{E} \max(0, |h_e(\overrightarrow{x_z})| - \epsilon_e \quad (5)$$

where every equality constraint ($h_e$), $\epsilon_e$ is initialized with a large value and is then reduced to 0.0001. Setting the initial value of $\epsilon_e$ is problem dependent, as indicated in [32].

Then the simulation component is re-called to evaluate the expected fitness for the new PS individuals. The stochastic nature of the results in the SCOP problem requires consideration inside the design of the DE algorithm. In traditional DE for deterministic problems, the fitness value is sufficient as the solutions' quality measure, but in the stochastic context, the solution's expected fitness value is used along with its standard deviation ($\sigma$) to express quality. Therefore, we penalized the stochastic fitness function with the upper bound of its $\beta$ Confidence Interval (CI) [13] in the following form in eq. (6).

$$\underset{x \in X}{\text{Min}} \left( E(f(\vec{x}, \vec{\xi}_n^L)) + z_\beta \left(\frac{\sigma_{N_{indv}}}{\sqrt{N_{indv}}}\right)\right) \quad (6)$$

where $E(f(\vec{x}, \vec{\xi}_n^L) = \sum_{j=1}^{N_{indv}} \left(\frac{f(\vec{x}, \xi_j^L)}{N_{indv}}\right), \quad l = 1, .., L$

The evolutionary process continues until the DE stopping convergence criterion holds, that is after a period where it can no longer improve its fitness. This is repeated for R runs. Sim-DE is assumed to be converged, at the end of each run r, to its best solution ($x_{br}$), with its associated expected fitness value, denoted as ($\mu_{br}$), where $(\mu_{br}) < (\mu_{jr})$, $j = 1, .., PS$, $j \neq b$. The final Average Outcome/Expected fitness of Sim-DE over all runs is $\mu = \sum_{r=1}^{R}(\mu_{br})$.

It is assumed that assigning intensive simulation budget $N_{max}$ scenarios, results in more accurate approximation for the true function value, as of the Law of Large Numbers (LLN) [33]. As LLN indicates that higher numbers of scenarios will result in more accurate estimation for the stochastic function, with probability tending to one. Thus, we considered $N_{max}$ scenarios as the Basic Strategy (BS), with a fixed intensive simulation budget for Sim-DE, as a reference strategy. Consequently, for comparison reasons, after the proposed ASBS Sim-DE terminates, the fitness values of the ($x_{br}$) best solutions, are re-evaluated using the $N_{max}$ scenarios, as shown in eq. (7).

$$(\mu')_{ASBS}: \sum_r \sum_{j=1}^{Nmax} \left(\frac{f(X_{bjr}, \ \xi_j^l)_{ASBS}}{Nmax}\right), where\ r = 1, \ldots, R, l = 1, .., L \quad (7)$$

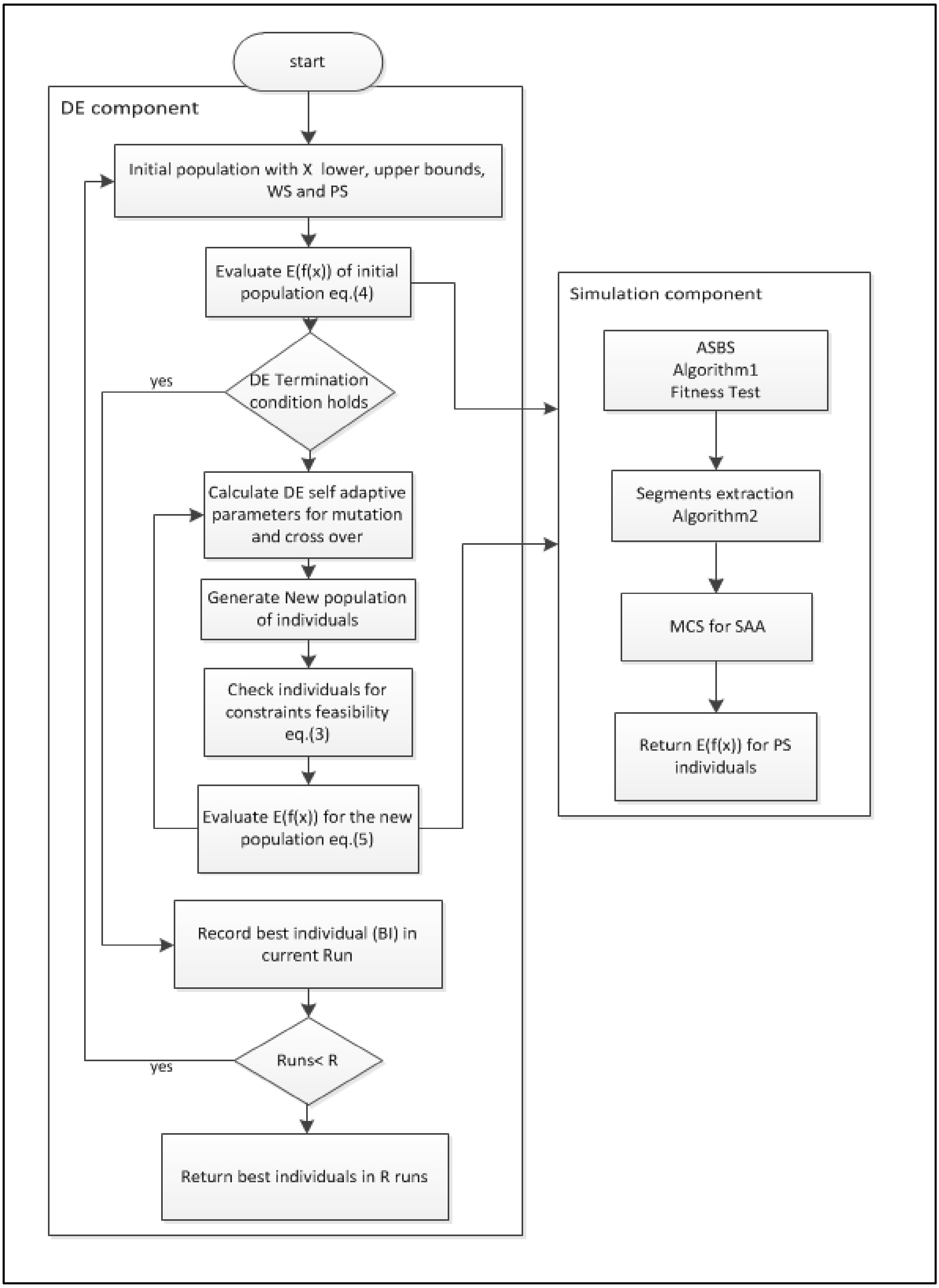


Figure 3. Flowchart of Sim-DE main algorithm

## 3. Experimental study

The experimental study with the proposed methodology is discussed in the following three subsections. The first one is experimental design, then the experimental results of the test problems follows and finally the experimental results of the practical problem known as the Dynamic Economic Dispatch Wind-Thermal problem is given.

### 3.1. Experimental design

The problem setting is defined as choosing the minimum mean among population size (PS) individuals at each Sim-DE generation. The Sim-DE algorithm settings are defined in Table 2, as recommended in [12, 14].

Table 2. Sim-DE settings

| Parameter | Value |
|---|---|
| **R** | 30 |
| **G** | 2,000 |
| **PS** | 100 |
| **WS** | [20, 20, 20, 20, 20] |
| **$N_{max}$** | 1000 |
| **Step** | 2 |

The performance of the proposed ASBS approach is compared with three distinct simulation budget strategies within the Sim-DE main framework. The first with the reference **BS**, has an intensive simulation budget with T = 100K. From eq. (3) the maximum TCB for BS is 200 million function evaluations in a single run, which is a formidable budget that requires efficient reduction while preserving the quality of the solutions. The second strategy follows the **OCBAm+** technique [20] to determine each $N_{indv}$ with a maximum T = 100k. Finally, the **Fixed WS** strategy uses SET Algorithm 2 separately from the ASBS Algorithm 1, with initial constant WS, as shown in Table 1, throughout all Sim-DE iterations, thus T = 10k. Algorithm 2 parameters, q1, q2 and q3, are set to 20%, 50% and 80% respectively, as initial start points, for the adaptive segments, in the first, median and third quartiles' ranges. The Sim-DE algorithm is terminated for a single run when at least one of the following conditions is met, a) stability state of convergence, b) the maximum number of generations G is reached.

The algorithm has been coded using MATLAB (R2015b) and was implemented on a PC with a 3.60-GHz Core™ i7 processor, with 16.0-GB RAM and Windows 7. For each strategy and algorithm, we conducted 30 independent runs. As per the minimization characteristics of the problem and the stochastic nature of the results, the mean values are coupled with their associated standard deviations for meaningful analysis of the output. Consequently, we are concerned with the $\mu'$ Upper Limit (UL) of $\beta$ confidence interval, as shown in eq. (8).

$$\mu'_{UL} = \mu' + \ z_\beta(\frac{\sigma_{N_{max}}}{\sqrt{N_{max}}})),\ where\ \ \beta = .95 \tag{8}$$

The Optimality Gap (OG) is used as a performance measure to compare different studied simulation budget strategies within the proposed Sim-DE framework. As the optimal for the true stochastic problem can't be calculated exactly [13], therefore the approximated estimates are calculated in eq. 5 by MCS. In addition, meta-heuristics (i.e. DE algorithm) do not guarantee optimality and its solution is possibly only a near optimal solution. Hence, the Optimality/near optimality Gap assumption is posited to calculate the difference between the best reached near optimal solution average fitness $(\mu'_{UL})_{Best}$ by one of the studied strategies, and near optimal solutions' average finesses by the rest of the studied strategies, at each test problem, as shown in eq. (9).

$$OG = (\mu'_{UL})_{St} - (\mu'_{UL})_{Best}, \ \ where\ St = [\ BS, Fixed\ WS,\ \ ASBS,\ \ OCBAm+] \tag{9}$$

### 3.2. Solving the Test Problems

Experiments were performed on a modified version of the IEEE-CEC'2006 single objective constrained optimization test problems [30], as no suitable constrained stochastic optimization set of test problems, with such characteristics, were found in the literature. Therefore, we modified these test problems by adding stochastic parameters to the objective functions, based on a Gaussian Normal distribution with specific mean and standard deviations N(Mean, Standard deviation). Where the CI level is set to be $\beta$=0.95. The first ten minimization problems, from G01 to G10, were used to test the proposed approach, with modified objective functions and deterministic constraints, as shown in Appendix A. The number of constraints of each problem, number of stochastic parameters (L) and dimension (D) are shown in Appendix A. The rest of the details about the constraints' functions and decision variables' $(x_i)$ limitations, can be found in the IEEE-CEC'2006 technical report [30].

For each test problem, detailed results are recorded in Table 3, which are respectively: the computational time given by CPU seconds per run, the average expected fitness value with its standard deviation, the expected fitness upper limit $(\mu'_{UL})$, OG and T. Note that the solutions' feasibility ratios (FS) equal 1 for all performed experiments.

Figure (4) depicts the adaptive behaviour of the segments' window sizes, e.g. for G05 run 30, where the WS for each segment adaptively changed through the DE generations. It shows that the proposed ASBS algorithm is capable of managing the simulation budget according to the DE needs through the Fitness Test. It does this by emphasizing important segments that require higher budget at different DE stages while saving simulations in other segments. We averaged the adaptive segments, WSs, over all runs and test problems, and found that it ranges between $9 \leq WS(i) \leq 34\ , for\ i = 1, .., 5$ segments. Consequently, $N_{indv}$ ranges between $45 \leq N_{indv} \leq 170 \leq N_{max}$.

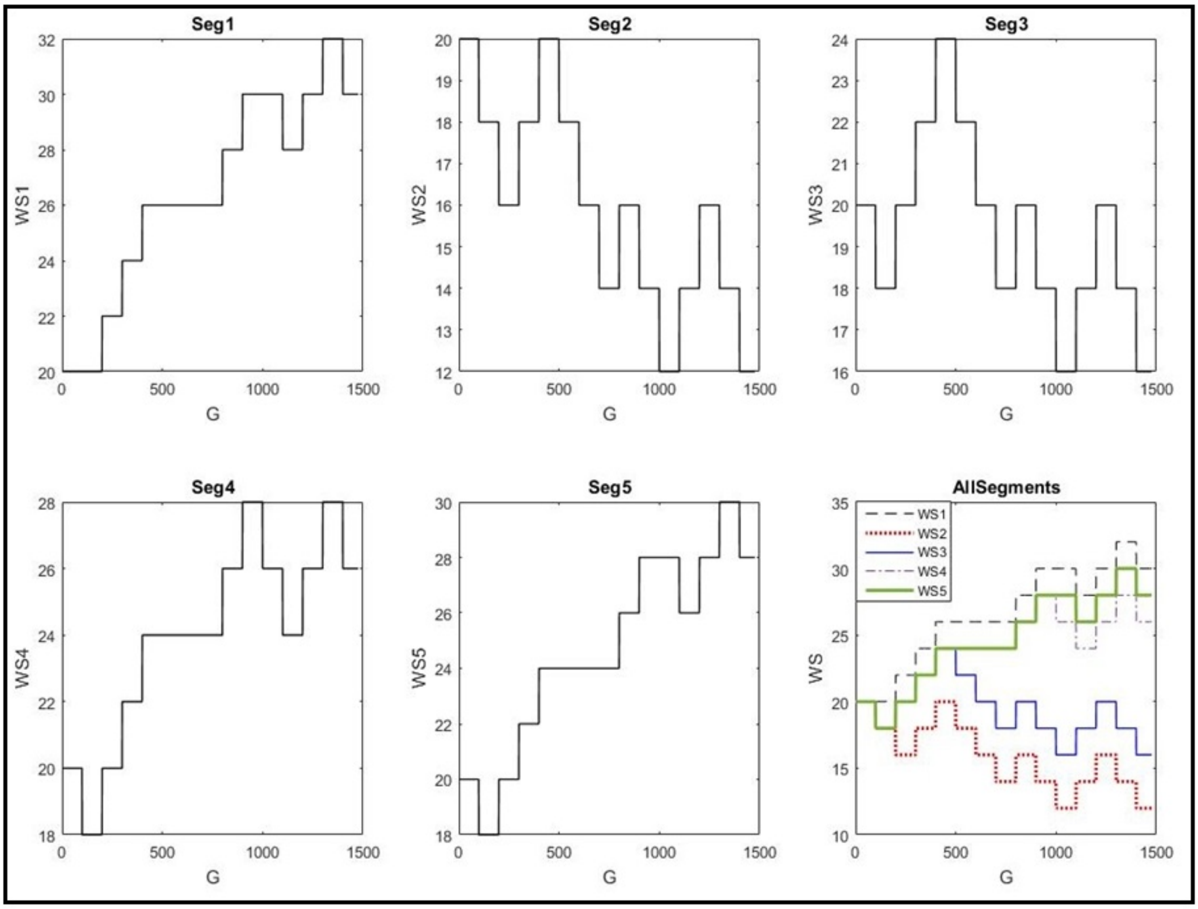


Figure 4. Adaptive WS behaviour in a single Sim-DE run for G05

Table 3. Detailed comparison results for modified stochastic benchmark CEC2006 problems

| ***G01*** | Computational time Sec./R | $\mu'$ | $\sigma$ | $\mu'_{UL}$ | $OG$ | $T$ |
|---|---|---|---|---|---|---|
| **BS** | 17.13 | -11.4089 | 55.8696 | -8.5025 | 0.55 | 100k |
| **Fixed WS** | 13.37 | -11.9722 | 56.1186 | -9.0529 | **0.00** | 10k |
| **ASBS** | 8.84 | -11.9100 | 56.1397 | -8.9897 | 0.06 | 8.3k |
| **OCBAm+** | 39.90 | -8.8079 | 52.3569 | -6.0843 | 2.97 | 100k |
| | ***G02*** | | | | | |
| **BS** | 24.76 | -1.0977 | 7.8090 | -0.6914 | **0.00** | 100k |
| **Fixed WS** | 8.745936 | -0.19085 | 1.300076 | -0.12322 | 0.57 | 10k |
| **ASBS** | 16.95315 | -0.71926 | 4.899683 | -0.46438 | 0.23 | 8.4k |
| **OCBAm+** | 132.69 | -0.9228 | 6.5171 | -0.5837 | 0.11 | 100k |
| | ***G03*** | | | | | |
| **BS** | 35.81 | -0.4176 | 8.9356 | 0.0472 | 0.80 | 100k |
| **Fixed WS** | 17.86939 | -0.22503 | 1.666848 | -0.13832 | 0.62 | 10k |
| **ASBS** | 19.35348 | -1.22889 | 9.102789 | -0.75536 | **0.00** | 6.6k |
| **OCBAm+** | 71.84 | -0.3534 | 8.3099 | 0.0789 | 0.83 | 100k |
| | ***G04*** | | | | | |
| **BS** | 12.77 | -30191.0664 | 9591.7280 | -29692.1099 | 427.71 | 100k |
| **Fixed WS** | 6.704923 | -30006.197 | 9642.852 | -29504.581 | 558.55 | 10k |
| **ASBS** | 8.195253 | -30613.502 | 9490.418 | -30119.816 | **0.00** | 13k |
| **OCBAm+** | 43.45 | -30044.6604 | 9843.9713 | -29532.5822 | 587.23 | 100k |
| | ***G05*** | | | | | |
| **BS** | 29.92 | 5734.754 | 12226.649 | 6370.778 | **0.00** | 100k |
| **Fixed WS** | 16.81639 | 5742.524 | 12246.42 | 6379.577 | 8.80 | 10k |
| **ASBS** | 17.68167 | 5741.935 | 12254.77 | 6379.422 | 8.64 | 9.6k |
| **OCBAm+** | 60.97 | 5745.553 | 12234.580 | 6381.989 | 11.21 | 100k |
| | ***G06*** | | | | | |
| **BS** | 8.35 | -3514.298 | 66414.088 | -59.473 | 165.14 | 100k |
| **Fixed WS** | 5.489155 | -731.35 | 15786.96 | 89.87923 | 314.49 | 10k |
| **ASBS** | 10.49107 | -1543.1 | 25346.07 | -224.614 | **0.00** | 11.9k |
| **OCBAm+** | 23.82 | -2988.813 | 60294.775 | 147.689 | 372.30 | 100k |
| | ***G07*** | | | | | |
| **BS** | 16.90 | 26.216 | 36.573 | 28.119 | **0.00** | 100k |
| **Fixed WS** | 11.13015 | 27.21052 | 39.45158 | 29.26277 | 1.14 | 10k |
| **ASBS** | 8.946137 | 26.47936 | 37.38793 | 28.42426 | 0.31 | 10k |
| **OCBAm+** | 45.54 | 28.112 | 35.868 | 29.978 | 1.86 | 100k |
| | ***G08*** | | | | | |
| **BS** | 10.90 | -0.0403 | 0.8638 | 0.0046 | 0.00 | 100k |
| **Fixed WS** | 5.45 | -2.67E-52 | 6.61E-51 | 7.71E-53 | **0.00** | 10k |
| **ASBS** | 5.53 | -1.12E-52 | 3.41E-51 | 6.5E-53 | **0.00** | 10.4k |
| **OCBAm+** | 46.08 | -0.0375 | 0.9223 | 0.0104 | 0.01 | 100k |
| | ***G09*** | | | | | |
| **BS** | 17.24 | 728.3453 | 1104.8998 | 785.8216 | **0.00** | 100k |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Fixed WS** | 7.80 | 732.8912 | 1106.06 | 790.4278 | 4.61 | 10k |
| **ASBS** | 9.81 | 731.0809 | 1105.649 | 788.5962 | 2.77 | 10.4k |
| **OCBAm+** | 31.66 | 746.5662 | 1117.1348 | 804.6790 | 18.86 | 100k |
| ***G10*** | | | | | | |
| **BS** | 18.08 | 10874.305 | 52438.630 | 13602.135 | **0.00** | 100k |
| **Fixed WS** | 7.11 | 11332.38 | 51238.15 | 13997.77 | 395.63 | 10k |
| **ASBS** | 6.35 | 11207.44 | 52977.76 | 13963.31 | 361.18 | 15k |
| **OCBAm+** | 25.64 | 12895.251 | 64304.571 | 16240.34 | 2638.20 | 100k |

The most striking observation to emerge, from the comparison results recorded in Table 3, is that the intensive budget/reference BS and OCBAm+ with T = 100k are outperformed by other "lower budget" strategies for some problems. For example, the proposed ASBS outperformed other strategies results for G03, G04, and G06, with zero OG, while showing the smallest OG in others. It is also noticeable that the simulation budget T changes adaptively for different problems and is not constant. However, Fixed WS strategy showed a moderate performance, while OCBAm+ strategy showed diminished performance. To clarify how each strategy performance, further analysis and a summary is reported in Table 4. Table 4 records, for every studied strategy, the average values over all problems and Sim-DE runs for the following values in columns, respectively: $N_{indv}$ scenarios size for each individual, T total simulation budget/R, the CPU computational time by seconds/run, convergence G: number of generations until reaching convergence or stability state, and finally percentage difference of OG.

The data summary in Table 4 reveals that ASBS and Fixed WS have considerably close values and definitely the lowest: T, computational time and convergence G. This illustrates the success of the proposed SET in both strategies. However, the percentage difference in OG is in ASBS's favour, which states that ASBS reached the highest solutions' quality. OCBAm+ showed inefficiency in time and solutions' quality compared to others, even when compared to its peer in budget (BS). This might be regarding as being due to the further computational complexity discussed in the Introduction section, and the sequential allocation rules of OCBAm+. While the proposed ASBS lowered the posited reference BS's intensive simulation budget by 90% and its computational time by 42%, with the lowest percentage difference OG/highest quality solutions. The percentage difference of OG and computational time are plotted in Figure 5, which clearly shows the superiority of the proposed ASBS. This prominent performance is caused by the capability of the proposed main Sim-DE framework to adapt and efficiently determine the simulation budget.

Table 4. Average performance measures over all problems and Sim-DE runs

| | $N_{indv}$ | T | Time (Sec/Run) | Convergence G | % Difference OG |
|---|---|---|---|---|---|
| **BS** | 1000 | 100k | 19 | 600 | 0.1877 |
| **Fixed WS** | 100 | 10k | 10 | 407 | 0.3134 |
| **ASBS** | 104 | 10.4k | 11 | 501 | 0.0378 |
| **OCBAm+** | 1000 | 100k | 52 | 645 | 0.3561 |

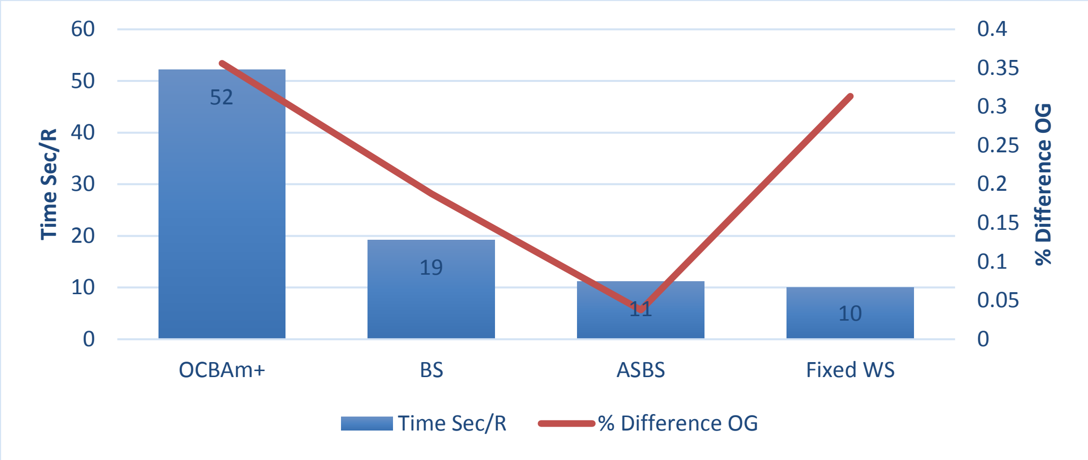


Figure 5. Average computational time and % difference OG over all problems and R

Statistical significance testing was performed using the Wilcoxon rank test [34]. It tests the null hypothesis of zero difference in the median, between the average expected fitness ($\mu_{BS}$) resulted from BS and the average re-evaluated expected fitness of other studied strategies ($\mu'_{St}$), at the 95% significance level. The results are shown in Table (5), the sign " $\approx$ " , indicates the acceptance of the null hypothesis, which means that there is no significance difference between the two means when p-value is greater than 0.05. While the sign " $\neq$ " indicates that the null hypothesis is rejected, which means that there is a significance difference between the expected fitness values. Wilcoxon negative ranks mean that strategy's (St) results, for a given number of test problems out of 10, are better than BS for the minimization setting, while positive ranks mean the contrary. These significance results confirm the previous findings that our proposed approach outperforms others and is not significantly different from BS with its intensive simulation budget and consequent high computational time. The ASBS results are even better in three problems out of ten, as besides its capability to reduce the simulation budget and hence computational time, it also preserves the solutions' quality.

Table 5. Wilcoxon Rank significance test

| | P-value>.05 | Negative ranks St<BS | Positive ranks St>BS | Decision |
|---|---|---|---|---|
| **Fixed WS** | 0.013 | 1 | 9 | $\neq$ |
| **ASBS** | 0.333 | 3 | 7 | $=$ |
| **OCBAm+** | 0.005 | 0 | 10 | $\neq$ |

To demonstrate the performance and scalability of the ASBS algorithm, additional experiments were performed on the proposed ASBS for problems G04, G07 and G09. These problems have been identified as they originally had the maximum number of stochastic parameters (S), of 4, 6 and 5 respectively, as shown in Appendix A. However, we implemented ASBS scalability analysis experiments, for each problem under different numbers of stochastic parameters, starting from S1 "single stochastic parameter" to the limit of S for each problem, as shown in Table (6). From Table (6), it is notable that the upper limit of the fitness value shows a low rate of change, given the change of (S). This is also clear in Figure (6), that shows the change in fitness behaviour for the three problems. Consequently, from these experiments, ASBS showed its stable performance under different uncertainties and scalability of S.

Table 6. ASBS scalability analysis

| Stochastic parameters (S) | G04 $\mu'_{UL}$ | G07 $\mu'_{UL}$ | G09 $\mu'_{UL}$ |
|---|---|---|---|
| S1 | -29567 | 28.38651 | 789.3488 |
| S2 | -29567 | 28.28197 | 788.7433 |
| S3 | -29552 | 28.26884 | 788.6471 |
| S4 | -29696 | 28.30401 | 788.4463 |
| S5 | | 28.38104 | 788.5962 |
| S6 | | 28.42426 | |
| Rate of change | 0.004853 | 0.005468 | 0.001143 |

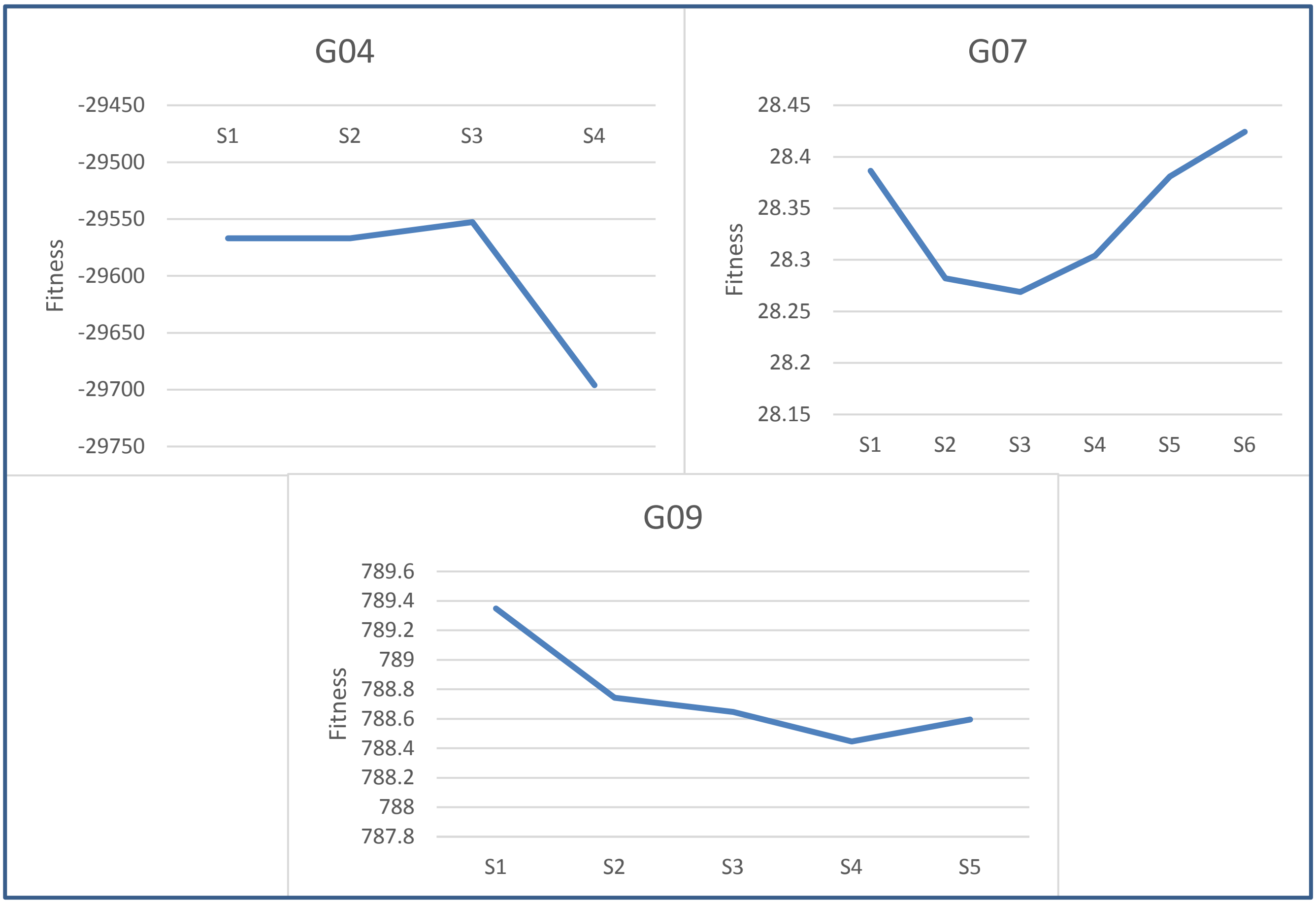


Figure 6. ASBS scalability analysis for problems G04, G07 and G09

Figure (7) further depicts graphical scalability analysis of the number of decision variables (D) and the number of stochastic parameters (S), for the studied strategies versus ASBS for all the problems. The two aspects are plotted against the percentage difference of OG as the measure of performance, and plotted by the order of the test problems from G01 to G10. This graphical analysis shows that the proposed ASBS outperformed all others under different conditions, except in one problem G02 with the highest D, where the performance is affected to some extent. While higher values of (S) did not affect the algorithm's quality. Therefore, given all previous the findings from different experiments, the remarkable performance is overall in favour of the proposed ASBS.

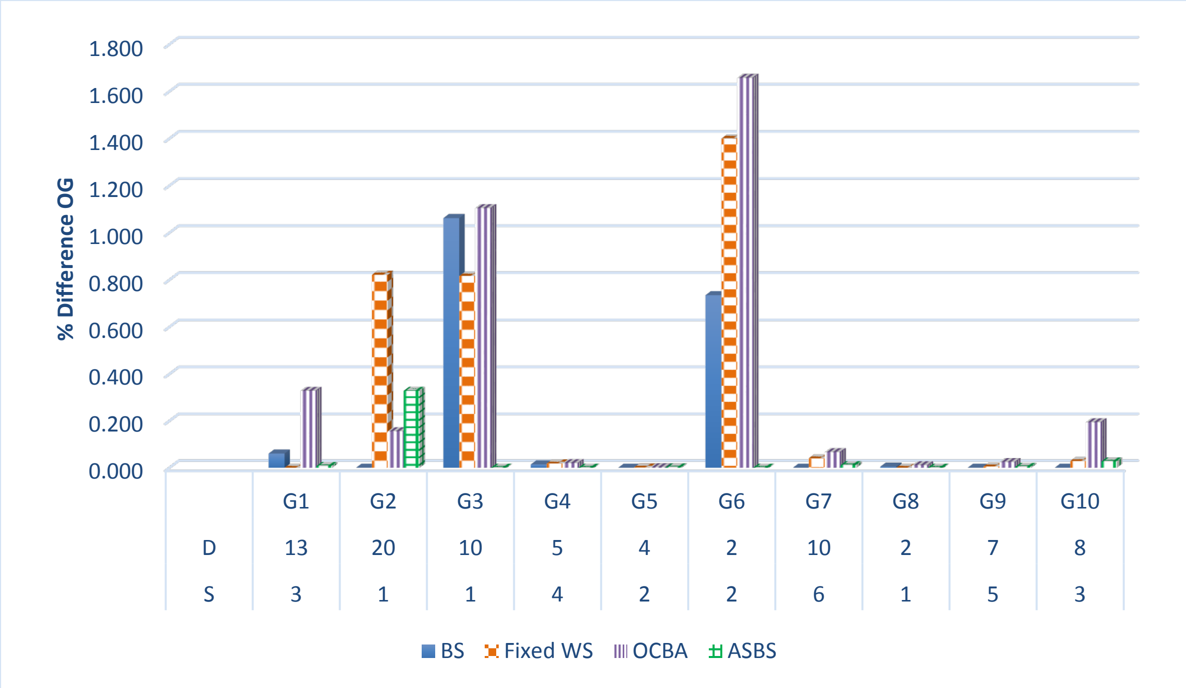


Figure 7. Comparison of the scalability of ASBS for different problems with dimension (D) and stochastic parameters (S)

### 3.3. A Well-known Real-world Problem

Dynamic Economic Dispatch (DED) problems are a common optimization problem in electrical power systems, which is mainly concerned with scheduling electricity generation using a combination of renewable energy (i.e. wind or solar) and thermal energy. Despite the environmental and economic advantages of renewable energy, such as wind energy, its difficulties are the continuity and reliability of its operation [35]. The DED problem, for a cycle of T time units with ramp limits, involves many local optima and multiple constraints [31]. Therefore, scheduling the right mix of generation from a number of generators to serve a particular load demand at minimum cost, is a challenging optimization problem [36]. However, DED models the operation of power systems more accurately and so has more research and practical value [31].

In this study, we adopted the wind-thermal DED problem [31, 36] to apply our strategies and algorithm, as it has the same characteristics as defined in our problem definition (SCOP) in the introduction section. In the wind–thermal DED system, the main aim is to determine the optimal power generation of the thermal and wind generators, by minimizing the overall operating cost while satisfying the number of constraints, as described below.

The mathematical model of the DED problem is illustrated in the following two subsections for the objective function and the constraints:

#### 3.3.1 Objective function

The objective function of a wind–thermal DED system comprises the fuel and environmental costs of thermal generators and the operating cost of wind turbines. In addition, the penalty costs, such as the over-and under-estimated ones of wind energy due to the stochastic nature of wind speeds are considered. Therefore, the power generated by a wind generator is considered as uncertain, and vary within a range According to a cost analysis of conventional and wind turbine generators, the objective function of the DED model in *T* time intervals can be expressed as [36]

$$\text{Min: } \mathbb{E}\langle \mathrm{F_C} \rangle = \sum_{n=1}^{N_{indv}} \sum_{t=1}^{T} \sum_{i=1}^{N_T} F_{c_i}\left(P_{T_{i,t,n}}\right) + \sum_{n=1}^{N_{indv}} \sum_{t=1}^{T} \sum_{w=1}^{N_W} \Big( F_{W_w}\left(\tilde{P}_{W_{w,t,n}}\right) + F_{U_w}\left(\tilde{P}_{W_{w,t,n}}\right) + F_{O_w}\left(\tilde{P}_{W_{w,t,n}}\right)\Big) \quad (10)$$

$$\text{where, } F_{c_{i,n}}\left(P_{T_{i,t,\mathrm{n}}}\right) = a_i + b_i P_{T_{i,t,n}} + c_i P_{T_{i,t,\mathrm{n}}}^2 + \left| d_i \sin\left\{ e_i \left( P_{T_{i,t}}^{\min} - P_{T_{i,t,\mathrm{n}}} \right) \right\} \right| \quad (11)$$

$P_{T_{i,t,\mathrm{n}}}$ is the $i^{th}$ thermal power plant at the $t^{th}$ time period of an operational cycle $T$ in the $n^{th}$ scenario. Note that, the uncertainties of renewable generations are represented using a number of possible scenarios, which are generated using a normal distribution in which its mean and standard deviation are taken from historical data [36]. $N_T$ is the number of thermal power plants, and $\mathbb{E}\langle F_c \rangle$ is the expected fuel costs. $a_i, b_i, c_i, d_i,\ e_i$ are the cost coefficients, and $\alpha_i,\ \beta_i,\ \gamma_i$ the emission coefficients of the $i^{th}$ thermal generator, where $N_W$ is the number of wind power plants, $P_{T_{i,t,\mathrm{n}}}$ and $\tilde{P}_{W_{w,t,\mathrm{n}}}$ are the output of the $i^{th}$ and $w^{th}$ thermal and wind generator, respectively, in which $\tilde{P}_{W_{w,t,\mathrm{n}}}$ is considered as uncertain paramater, in the $n^{th}$ scenario in which $F_{c_i}$ is the fuel cost of the $i^{th}$ thermal generator (as in eq. (11)), while $F_{W_w}$ is the operating cost, $F_{U_w}$ and $F_{O_w}$ are the penalty costs for the under- and over-estimated wind energy, respectively. A liner function is used to represent the operating cost of wind generators, as [36]:

$$F_{W_w}(\tilde{P}_{W_{w,t,\mathrm{n}}}) = \delta_w \tilde{P}_{W_{w,t}},\ w \in N_W\ t \in T\ n \in N_{indv} \quad (12)$$

where $\delta_w$ is the per unit cost of the $w^{th}$ wind generator, with its output at the $t^{th}$ time interval expressed as [37]:

$$\tilde{P}_{W_{w,t,n}} = \begin{cases} 0 & if\ \ v_{out_w} < \tilde{v}_{w,t,\mathrm{n}} < v_{in_w} \\ P_{R_w} \frac{\tilde{v}_{w,t,\mathrm{n}} - v_{in_w}}{v_{r_w} - v_{in_w}} & if\ \ v_{in_w} < \tilde{v}_{w,t,\mathrm{n}} < v_{r_w} \\ P_{R_w} & \mathrm{i}f\ \ v_{r_w} < \tilde{v}_{w,t,\mathrm{n}} < v_{out_w} \end{cases} \quad (13)$$

where $v_{out_w}$, $v_{in_w}$, $v_{r_w}$ and $\tilde{v}_{w,t,n}$ are the cut-out, cut-in, rated and $t^{th}$-hour wind speed of the $w^{th}$ wind farm in the $n^{th}$ scenario, respectively, and $P_{R_w}$ is the rated wind power from the $w^{th}$ wind generator.

Furthermore, we include penalty costs for any forecasted wind farm being under- or over-estimated, which are expressed as [36]:

$$F_{U_w}(\tilde{P}_{W_{w,t,n}}) = k_{U_w} \int_{P_{W_{w,t,n}}}^{P_{R_W}} \left(w - \tilde{P}_{W_{w,t,n}}\right) f_{P_{W_{w,t,\mathrm{n}}}}(w)dw \quad (14)$$

$$F_{O_W}(\tilde{P}_{W_{w,t,\mathrm{n}}}) = k_{O_w} \int_{0}^{P_{R_W}} \left(\tilde{P}_{W_{w,t,\mathrm{n}}} - w\right) f_{P_{W_{w,t,n}}}(w)dw \quad (15)$$

Earlier research shows that the wind speed follows a Weibull distribution function, as [37]:

$$f_{P_{W_{w,t}}}(W) = \frac{K_t l v_{in}}{c_t} \phi^{K_t - 1} e^{-\phi^{K_t}}, 0 < W_t < W_R \quad (16)$$

where the constants $k_t$, $c_t$ and $\phi$ are determined as:

$$K_t = (\sigma_t/\mu_t)^{-1.086}, \tag{17}$$

$$c_t = \frac{\mu_t}{\Gamma(1+K_t^{-1})} \tag{18}$$

$$\phi = \frac{(1+(W/W_R)l)v}{c_t} \tag{19}$$

$$\text{where } l = \frac{v_r - v_{in}}{v_{in}} \tag{20}$$

where $\mu_t$ and $\sigma_t$ are the mean and standard deviations of the wind speed at the $t^{th}$ hour, respectively.

### 3.3.2 Constraints

The load demand, capacity and ramp constraints are considered in a wind-thermal DED problem, as [36]:

$$\sum_{i=1}^{N_T} P_{T_{i,t}} + \sum_{w=1}^{N_W} \tilde{P}_{W_{w,t}} = P_{D_t} \tag{21}$$

$$P_{T_i}^{\min} \leq P_{T_{i,t}} \leq P_{T_i}^{\max} \;\; i \in N_T, t \in T \tag{22}$$

$$0 \leq \tilde{P}_{W_{w,t}} \leq P_{R_w} \;\; w \in N_W, t \in T \tag{23}$$

$$-DR_i \leq \left(P_{i,t} - P_{i,(t-1)}\right) \leq UR_i, \text{if } P_{i,(t-1)} > P_i^{\min} \tag{24}$$

$$-DR_i^0 \leq \left|P_{i,t} - P_{i,(t-1)}\right| \leq UR_i^1, if\ 0 < P_{i,(t-1)} < P_i^{\min} \tag{25}$$

$$\left[T_{i,(t-1)}^{on} - T_{\underset{i}{\min}}^{on}\right]\left[U_{T_{i,(t-1)}} - U_{T_{i,t}}\right] \geq 0 \tag{26}$$

$$\left[T_{i,(t-1)}^{off} - T_{\underset{i}{\min}}^{off}\right]\left[U_{T_{i,t}} - U_{T_{i,(t-1)}}\right] \geq 0$$

Eq. (24) shows the conventional ramp limits between two consecutive hours, and eq. (25) is the initial/final ramp limits when a generating unit is in the process of startup or shutdown, in which $UR^1$ and $DR^0$ are the initial ramp up and down respectively. Eq. (26) represents the minimum on/off time of a thermal generator, where $T_{min_i}^{on}$ and $T_{max_i}^{off}$ are the minimum on and off time of the $i^{th}$ unit, respectively. $T_{i,(t-1)}^{on}$ and $T_{i,(t-1)}^{off}$ are the continuous on and off time of the $i^{th}$ unit at the $t^{th}$ time interval , respectively, and $U_{T_{i,t}}$ are the operational status of the $i^{th}$ thermal unit at the $t^{th}$ time interval, i.e., 0 - unit off, 1 - unit on.

In this study we adopted the problem settings as identified in [36], with $N_w = 1, N_T = 5, and\ T = 6$ hours therefore the decision variables dimension D=36. For each strategy, we conducted 30 independent runs. The $\mu'$ Upper Limit (UL) of $\beta = .95$ confidence interval was calculated using eq. (8), while OG was calculated using eq. (9).

The proposed algorithm, ASBS, has been applied on this application and compared to the basic strategy (BS) and Fixed WS strategy performance. However, it is not compared to other algorithms because we modified the problem to be single objective with deterministic constrains, to enable us to apply the proposed algorithm, while other studies either studied the problem with stochastic constraints or were multi-objective, which we will consider in future research.

As shown in Table (7), BS recorded the minimum outcome, while the ASBS outcome outperformed the Fixed WS strategy's fitness outcome. ASBS again showed its superiority in computational time and overall simulation cost over the BS and Fixed WS. However, ASBS performance has been affected by the scalability of the application to some extent, as occurred earlier in G02 in Figure 6.

Table 7. Detailed comparison results for DED wind-thermal power systems problem

| ***DED Wind-Thermal*** | Computational time Sec./R | $\mu'$ | $\sigma$ | $\mu'_{UL}$ | FS | % Difference OG | $simulation$ $T$ |
|---|---|---|---|---|---|---|---|
| **BS** | 84.73922 | 819163 | 1833.187 | 819163 | 1 | 0 | 100k |
| **Fixed WS** | 82.88801 | 879317.9 | 3062.107 | 879477.1 | 1 | 1.074 | 10k |
| **ASBS** | 80.65824 | 876695.7 | 1747.712 | 876786.6 | 1 | 1.07 | 8k |

Such complex optimization problems suffer from a lack of theoretical analysis, as the development and performance analysis of EA widely relies on benchmarking [38]. However, the effective performance of the model, can be seen by the high potential of the ASBS procedures/equations in "Algorithm1" to perform the Fitness Test (FT) and to adaptively determine suitable window sizes throughout different generations. As FT compares the re-evaluated fitness values by $N_{max}$ against the associated fitness quality, after a potential change in window size is tested. While the SET procedures/equations in "Algorithm2" extract the most representative simulation samples, given the updated window sizes by FT. These integrated procedures, cost efficiently direct the overall evolutionary algorithm towards the best values, with high likelihood, while using the same originally given probability distribution in the problem's definition. This logical sequence illustrates the cost efficient performance, alongside the good solution fidelity, of the proposed algorithm.

## 4. Conclusion

Recent developments in the Sim-Opt paradigm have heightened the need for efficient utilization of the subsequent amplified computational burden. This study set out to handle this challenge by a proposed Simulation assisted Differential Evolution Algorithm (Sim-DE) for solving constrained stochastic problems. Most of the research in this field largely handled the simulation budget separately from the optimization process. However, the idea of reversing this challenge by mutually benefit from this combination is not deeply studied. Our proposed study attempted to improve this by allowing the Sim-DE algorithm to adaptively and efficiently control the simulation budget, as necessary in different generations. The Adaptive Segment Based Scheme (ASBS), along with the Segments Extraction Technique (SET) were proposed within the main Sim-DE framework. These algorithms foster capability of DE to control the stochastic scenarios' sizes, while extracting representative, biased and smaller samples of these scenarios, which are efficiently sufficient for guiding DE in the search space. The proposed approach was empirically tested on a modified stochastic constrained set of IEEE-CEC2006 test problems and the real world application of the Dynamic Economic Dispatch wind-thermal power system problem.

A comparison was performed between the proposed ASBS and other intensive simulation budget allocation techniques in the literature. It showed superior performance to the others in terms of the quality of its solutions, its ability to converge quickly, reduce the simulation budget and computational time. ASBS was capable to reduce the simulation cost by 90% and the computational time by 42%, while keeping the optimality gap to its minimum in comparison to the others in the Sim-DE framework, which also having the highest solutions' quality. However, its performance may need further validation, when using different probability distributions and higher dimensions. Further work is intended to test its performance when stochastic parameters occur in the constraints, and are not only limited to the objective function.

## Acknowledgement

The authors thank ARC Discovery Project (DP190102637) awarded to Sarker and Essam for supporting this research.

## References

[1] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, "Simulation optimization: a review of algorithms and applications," *Annals of Operations Research,* vol. 240, pp. 351-380, 2016.

[2] J. Xu, E. Huang, C.-H. Chen, and L. H. Lee, "Simulation optimization: a review and exploration in the new era of cloud computing and big data," *Asia-Pacific Journal of Operational Research,* vol. 32, p. 1550019, 2015.

[3] W. Long-Fei and S. Le-Yuan, "Simulation optimization: A review on theory and applications," *Acta Automatica Sinica,* vol. 39, pp. 1957-1968, 2013.

[4] R. Kuo and C. Yang, "Simulation optimization using particle swarm optimization algorithm with application to assembly line design," *Applied Soft Computing,* vol. 11, pp. 605-613, 2011.

[5] A. Rasekh and K. Brumbelow, "A dynamic simulation–optimization model for adaptive management of urban water distribution system contamination threats," *Applied Soft Computing,* vol. 32, pp. 59-71, 2015.

[6] A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems," *Operations Research Perspectives,* vol. 2, pp. 62-72, 2015.

[7] H. Liu and S. Maghsoodloo, "Simulation optimization based on Taylor Kriging and evolutionary algorithm," *Applied Soft Computing,* vol. 11, pp. 3451-3462, 2011.

[8] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.

[9] R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential Evolution With Dynamic Parameters Selection for Optimization Problems," *IEEE Trans. Evolutionary Computation,* vol. 18, pp. 689-707, 2014.

[10] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & operations research,* vol. 38, pp. 1877-1896, 2011.

[11] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation,* vol. 13, pp. 398-417, 2009.

[12] S. Elsayed, R. Sarker, and C. C. Coello, "Enhanced multi-operator differential evolution for constrained optimization," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*, 2016, pp. 4191-4198.
[13] M. C. Fu, *Handbook of simulation optimization* vol. 216: Springer, 2015.
[14] G. B. Tito Homem-de-Mello, "Monte Carlo Sampling-Based Methods for Stochastic Optimization," 2014.
[15] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method* vol. 10: John Wiley & Sons, 2016.
[16] A. Shapiro, "Sample average approximation," in *Encyclopedia of Operations Research and Management Science*, ed: Springer, 2013, pp. 1350-1355.
[17] A. M. Akl, R. A. Sarker, and D. L. Essam, "Simulation optimization approach for solving stochastic programming " presented at the 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China, China 2017.
[18] T. Homem-de-Mello and G. Bayraksan, "Monte Carlo sampling-based methods for stochastic optimization," *Surveys in Operations Research and Management Science,* vol. 19, pp. 56-85, 2014.
[19] L. H. Lee, C. Chen, E. P. Chew, J. Li, N. A. Pujowidianto, and S. Zhang, "A review of optimal computing budget allocation algorithms for simulation optimization problem," *International Journal of Operations Research,* vol. 7, pp. 19-31, 2010.
[20] S. Zhang, L. H. Lee, E. P. Chew, J. Xu, and C.-H. Chen, "A simulation budget allocation procedure for enhancing the efficiency of optimal subset selection," *IEEE Transactions on Automatic Control,* vol. 61, pp. 62-75, 2016.
[21] W.-h. Liu, "Optimal computing budget allocation to the differential evolution algorithm for large-scale portfolio optimization," *Journal of Simulation,* pp. 1-11, 2016.
[22] Y. Peng, C.-H. Chen, M. C. Fu, and J.-Q. Hu, "Non-Monotonicity of probability of correct selection," in *Winter Simulation Conference (WSC), 2015*, 2015, pp. 3678-3689.
[23] A. Emelogu, S. Chowdhury, M. Marufuzzaman, L. Bian, and B. Eksioglu, "An enhanced sample average approximation method for stochastic optimization," *International Journal of Production Economics,* vol. 182, pp. 230-252, 2016.
[24] J. Biethahn and V. Nissen, "Combinations of simulation and Evolutionary Algorithms in management science and economics," *Annals of Operations Research,* vol. 52, pp. 181-208, December 01 1994.
[25] A. Akl, R. Sarker, and D. Essam, "A Study of Robustness in Evolutionary Simulation Optimization Algorithm," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 2806-2813.
[26] J. M. Utts and R. F. Heckard, *Mind on statistics*: Cengage Learning, 2011.
[27] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to probability and statistics*: Cengage Learning, 2012.
[28] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, *Understanding robust and exploratory data analysis*, 2000.
[29] R. Melchers, "Importance sampling in structural systems," *Structural safety,* vol. 6, pp. 3-10, 1989.
[30] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. C. Coello*, et al.*, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics,* vol. 41, 2006.
[31] C. Peng, H. Sun, J. Guo, and G. Liu, "Dynamic economic dispatch for wind-thermal power system using a novel bi-population chaotic differential evolution algorithm," *International Journal of Electrical Power & Energy Systems,* vol. 42, pp. 119-126, 2012.
[32] C. Si, J. An, T. Lan, T. Ußmüller, L. Wang, and Q. Wu, "On the equality constraints tolerance of constrained optimization problems," *Theoretical Computer Science,* vol. 551, pp. 55-65, 2014.

[33] E. Seneta, "A tricentenary history of the law of large numbers," *Bernoulli,* vol. 19, pp. 1088-1121, 2013.

[34] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation,* vol. 1, pp. 3-18, 2011.

[35] F. Zaman, S. M. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Multi-Method based algorithm for multi-objective problems under uncertainty," *Information Sciences,* vol. 481, pp. 81-109, 2019.

[36] F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems," *Engineering Applications of Artificial Intelligence,* vol. 53, pp. 105-125, 2016.

[37] J. Hetzer, C. Y. David, and K. Bhattarai, "An economic dispatch model incorporating wind power," *IEEE Transactions on energy conversion,* vol. 23, pp. 603-611, 2008.

[38] M. Hellwig and H.-G. Beyer, "Benchmarking evolutionary algorithms for single objective real-valued constrained optimization–A critical review," *Swarm and evolutionary computation,* vol. 44, pp. 927-944, 2019.

**Appendix A.** Modified Stochastic constrained CEC2006 Test problems

| Function No | Stochastic objective function | No. of constraints | No. stochastic parameters (L) | Dimension (D) |
|---|---|---|---|---|
| **G01** | $Min. E(f(\vec{x})) = N(5,10)\sum_{i=1}^{4} x_i - N(5,10)\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i + N(1,5)$ | 9 | 3 | 13 |
| **G02** | $Min. E(f(\vec{x})) = -N(2,10)\left\lvert \frac{\sum_{i=1}^{n} cos^4(x_i) - 2\prod_{i=1}^{n} cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right\rvert$ | 2 | 1 | 20 |
| **G03** | $Min. E(f(\vec{x})) = -N(1,10)\left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i$ | 1 | 1 | 10 |
| **G04** | $Min. E(f(\vec{x})) = N(5.3578547,10)\, x_3^2 + N(0.8356891,1)\, x_1 x_5 + \mathrm{N}(37.293239,10)\, x_1 - \mathrm{N}(40792.141,1000)$ | 6 | 4 | 5 |
| **G05** | $Min. E(f(\vec{x})) = N(3,10)\, x_1 + 0.000001\, x_1^3 + \mathrm{N}(2,10)\, x_2 + \left(\frac{0.000002}{3}\right) x_2^3$ | 5 | 2 | 4 |
| **G06** | $Min. E(f(\vec{x})) = N(1,10)\,(x_1 - 10)^3 + N(1,10)\,(x_2 - 20)^3$ | 2 | 2 | 2 |
| **G07** | $Min. E(f(\vec{x})) = x_1^2 + x_2^2 + x_1 x_2 - \mathrm{N}(14,10)x_1 - \mathrm{N}(16,10)x_2 + (x_3 - 10)^2 + \mathrm{N}(4,10)(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + \mathrm{N}(5,5)\, x_7^2 + 7(x_8 - 11)^2 + \mathrm{N}(2,5)(x_9 - 10)^2 + (x_{10} - 7)^2 + \mathrm{N}(45,5)$ | 8 | 6 | 10 |
| **G08** | $Min. E(f(\vec{x})) = -N(1,10)\, \frac{sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$ | 2 | 1 | 2 |
| **G09** | $Min. E(f(\vec{x})) = (x_1 - 10)^2 + \mathrm{N}(5,10)(x_2 - 12)^2 + N(1,5)x_3^4 + N(3,10)(x_4 - 11)^2 + 10\, x_5^6 + 7\, x_6^2 + x_7^4 - \mathrm{N}(4,2)x_6 x_7 - 10x_6 - \mathrm{N}(8,2)x_7$ | 4 | 5 | 7 |
| **G10** | $Min. E(f(\vec{x})) = N(1,10)x_1 + N(2,5)x_2 + N(1,10)x_3$ | 6 | 3 | 8 |