Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

SIMULATION OPTIMIZATION USING THE PARTICLE SWARM OPTIMIZATION WITH OPTIMAL COMPUTING BUDGET ALLOCATION

Si Zhang
Pan Chen
Loo Hay Lee
Chew Ek Peng

National University of Singapore
Department of Industrial
and Systems Engineering
10 Kent Ridge Crescent, 119260,
SINGAPORE

Chun-Hung Chen

National Taiwan University
Department of Electrical
Engineering
No.1, Sec. 4, Roosevelt Road, 106
Taipei, TAIWAN

ABSTRACT

Simulation has been applied in many optimization problems to evaluate their solutions' performance under stochastic environment. For many approaches solving this kind of simulation optimization problems, most of the attention is on the searching mechanism. The computing efficiency problems are seldom considered and computing replications are usually equally allocated to solutions. In this paper, we integrate the notion of optimal computing budget allocation (OCBA) into a simulation optimization approach, Particle Swarm Optimization (PSO), to improve the efficiency of PSO. The computing budget allocation models for two versions of PSO are built and two allocation rules PSOs_OCBA and PSObw_OCBA are derived by some approximations. The numerical result shows the computational efficiency of PSO can be improved by applying these two allocation rules.

1 INTRODUCTION

In this paper, we consider the problem of simulation optimization with continuous solution space in stochastic setting. The classic approaches to tackle this problem include stochastic approximation (Rubinstein and Shapiro 1993) and sample path method (Gurkan, Ozge, and Robinson 1994). In recent years, because of the advantages of derivative-free and black-box nature, many metaheuristics have been adopted to solve the simulation optimization problems with continuous solution space, such as the nested partition (Shi and Olafsson 1997), particle swarm optimization (Kennedy and Eberhart 1995) and differential evolution (Storn and Price 1997).

For all these approaches, the search mechanism is a very important part as it decides where the candidate solution(s) should move so that the optimal solution can be gradually obtained. Due to the stochastic environment, each selected solution should be repeatedly evaluated and the sample mean used as an estimator of this solution's performance. Therefore, we need to make a computing effort balance between exploration and exploitation, which means a trade-off between how much computing effort should be devoted to searching new solution(s) and evaluating the new generated solution(s) versus how much computing effort should be allocated to the existing candidate solution(s). Moreover, at each iteration of many metaheuristics, some better solutions need to be selected from all the candidate solutions to generate the new solutions. So there is also the problem about how to allocate the computing effort to each candidate solution within each iteration. In this paper, we aim to do some contribution on the efficiency improvement of simulation optimization using the particle swarm optimization method.

The origin of particle swarm optimization (PSO) is from the computer animation requirement of forming what appeared to be a fuzzy object. Kennedy and Eberhart (1995) develop the basic model for PSO and lead to the application of PSO in finding the optimal solution of mathematical functions. In PSO, each time certain number of solutions in the search space will be selected as particles to form a swarm. Each particle in the swarm will move through a search space according to its velocity value based on the location information of both the best solution that it has found individually (personal best) and the best solution that is found by any of the particles that this particle can communicate with (global best). To avoid the case of rapid convergence to local optimal and the case of finding the global optimal but with very slow convergence rate, Shi and Eberhart (1998) incorporate an inertia weight into the velocity update equation to improve the basic PSO model. Based on the same consideration, Clerc and Kennedy (2002) introduce another factor, named as constriction factor, into the velocity update equation to build a generalized PSO model. By incorporating the idea of cluster analysis, Kennedy (2000) modifies the original PSO and develops a new version of PSO algorithm by using cluster centers as personal best of particles. For different specific simulation optimization problems, a lot of other versions of PSO have been developed in recent years. Banks, Vincent and Anyakoha (2007, 2008) give a comprehensive review of these developments. Bratton and Kennedy (2007) take these recent developments into account and define a standard for PSO.

In all of these PSO versions, a comparison among all candidate solutions within each iteration is required to update particles' locations. And the computing budget is usually equally allocated to these candidate solutions under stochastic environment. Because the number of particles at a swarm is limited, some approaches in ranking and selection procedures can be applied into the comparison process to efficiently allocate computing replications to these competing candidate solutions. Among these approaches, the optimal computing budget allocation (OCBA) procedures developed by Chen et al. (2000) aims at maximizing the probability of correctly selecting the best design(s) from finite number of designs under limited computing budget constraint. It has shown great potential in improving simulation efficiency for tackling simulation optimization problems. Chen et al. (2008) show numerical examples about the performance of the algorithm combining OCBA-m with Cross-Entropy (CE). The theoretical part about the integration of OCBA with CE is then further analyzed in He et al. (2010). Chew et al. (2009) integrate MOCBA with Nested Partition (NP) to handle multi-objective inventory policies problems and Lee, Wong, and Jaruphongsa (2009) integrate MOCBA with GA to solve an aircraft spare part allocation problem. In all these papers, the numerical results demonstrate the significant improvements gained by integrating OCBA into these simulation optimization approaches. The application of OCBA into PSO is considered in Pan, Wang, and Liu (2006), where they do not analyze the PSO from the OCBA perspective but just directly apply OCBA allocation rule from Chen et al. (2000) to select the best particle at a swarm.

In this paper, we integrate OCBA into two versions of PSO and model the computing budget allocation problems of PSO by maximizing the convergence rate of the probability of incorrect selection. The conditions for the asymptotical optimal allocation rules for the standard PSO and PSObw are derived. Under some assumptions, we get the optimal allocation rules, named as PSOs_OCBA and PSObw_OCBA, which are closed-form and easy to implement. Numerical testing indicates that the resulting integrated procedure can lead to computational efficiency gains for both the standard PSO and PSObw. We reiterate that our objective is not to find the best PSO algorithm or compare the standard PSO with PSObw, but rather to demonstrate that an intelligent control of simulation budget allocation can improve the computational efficiency of PSO. The framework in this paper can also be flexibly applied to other versions of PSO or other simulation optimization approaches to seek the computational efficiency improvement.

The rest of this paper is organized as follows. In section 2, we introduce the simulation optimization problem setting and build computing budget allocation models for both the standard PSO and the PSObw from a large deviation perspective. Section 3 derives the asymptotically optimal simulation allocation rules to minimize the probability of incorrect selection. In section 4, we show two numerical experiments to compare the performance of PSOs_OCBA and PSObw_OCBA with the equal allocation rule PSOs_EA and PSObw_EA. Section 5 concludes the whole paper.

2 SIMULATION OPTIMIZATION PROBLEM FORMULATION

In this section, we firstly give a brief introduction of PSO algorithms, and then we build the computing budget allocation models for two versions of PSO.

2.1 Particle Swarm Optimization

In PSO, there are total m particles in a swarm. Let $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, $i = 1, 2, ..., m$, denote the location of an individual particle i within an iteration in the D-dimensional solution space. The location of particle i is updated at each time step by updating the velocity $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, which is related to the old velocity, the distance to the personal best $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and the distance to the global best $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$. In the version of standard PSO algorithm (Bratton and Kennedy 2007), the personal best of one particle is defined as the location of this particle's own previous best performance, while the global best is defined as the best solution that all particles have found. The updated position equals to the old position with the updated velocity added. So we have the following updating equations with constriction at each dimension d ($d = 1, 2, ..., D$).

$$v_{id} = \chi\left(v_{id} + c_1\varepsilon_1\left(p_{id} - x_{id}\right) + c_2\varepsilon_2\left(p_{gd} - x_{id}\right)\right) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

In equation (1), χ is the constrictive factor to induce convergence and prevent particles moving to the outside of the desirable range of the search space. c_1 and c_2 are two constants to justify the convergence to local best and the convergence to global best. ε_1 and ε_2 are two independent uniformly distributed random numbers to ensure certain level of random search among the whole solution space. The whole algorithm of PSO can be summarized as follows.

Algorithm. PSO	
Initialization	Particles are originally initialized in a uniform random manner throughout the search space; velocity is also randomly initialized. Based on each particles' fitness values, get the initial value of $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$. Set t=1;

Updating	For each particle i in the swarm do
	    Update velocity v_i and position X_i using equations (1) and (2);
	end for
	Calculate these new particles' fitness values ;
	Update P_i and P_g;

Stopping	If the stopping criteria is satisfied, stop; otherwise set t=t+1 and loop to the step Updating.

Inspired by the clustering idea in the PSO with cluster analysis developed by Kennedy (2000), we propose another version of PSO, named as PSO with best half and worst half method (PSObw). In PSObw, particles are classified into two subsets, the best half set S_{best} and the worst half set S_{worst} based on their fitness values, not based on their locations in PSO with cluster analysis. For PSObw, the personal best of the particles in the best half is defined as their own current locations and the personal best of the particles in the worst half is defined as the location of the particle in the best half nearest to them. The global best is the best particle in the swarm. The updating equations and the algorithm of PSObw is the same with the standard PSO's, except that the determinations of the personal best and global best are different. In standard PSO, personal best and global best is related to the historical performances of the locations these particle have visited. However, in PSObw, personal best and global best are updated based on the performances of particles in the current swarm. The introduction of PSObw here is for displaying the generality of the application of OCBA into PSO algorithms, not for comparing PSObw with PSO.

In deterministic case, we can directly get each particle's true fitness value. However, under stochastic environment, the performance at each location is a random variable because of the noise. In this case, we generally use the unbiased estimator, sample mean, to estimate the mean fitness value of each particle. Both for the standard PSO and for PSObw, we need to find the global best and personal best of each particle in the updating step. It belongs to a ranking & selection (R&S) problem. The correctness of selecting the personal best and global best will directly affect the quality of particles generated at the next iteration. Intuitively, to ensure the high level of the selection correctness, more replications should be allocated to the particles that play a more important role in the updating step. Instead of simulating each particle with equal replications as most PSO algorithms do, we integrate the concept of OCBA, an efficient R&S procedure, into PSO when we calculate particles' fitness values in the updating step at each iteration to improve the efficiency of PSO. In the following subsections, we model the computing budget allocation problem of the standard PSO and the PSObw from the perspective of maximizing the convergence rate function of the probability of incorrect selection.

2.2 Computing Budget Allocation Model for Standard PSO

Let Θ denote the continuous solution space and $f(X)$ denote the mean fitness of the solution X that belongs to Θ. So the general optimization problem can be modeled as follows.

$$\min_{X\in\Theta} f\left(X\right)$$

Because it is impossible to have infinite replications, the performance of $f(X)$ under the stochastic environment can only be estimated by the sample mean, denoted as $\overline{f}(X)=\left(1/N\right)\sum_{j=1}^{N}\hat{f}_j(X)$, in which $\hat{f}_j(X)$ is the sample performance of solution X at the *j-th* simulation replication and *N* is the computing budget allocated to X.

In standard PSO, the personal best of one particle is the location of this particle's own previous best performance and the global best is the best solution that any particle has found. Suppose the solution P_i and P_g is the personal best and global best respectively of last iteration. Let T be the computing budget at this iteration and let $N_i=\alpha_i T$ denote the replications allocated to particle *i* at this iteration. For convenience, we introduce the following notation to partition the swarm into three mutually exclusive subsets:

- S_A : the set of particles whose fitness values are better than P_g, that is, $S_A=\left\{X_i : f\left(X_i\right)\le f\left(P_g\right)\right\}$,
- S_B: the set of particles whose fitness values are worse than P_g but greater than their personal best P_i, that is, $S_B=\left\{X_i : f\left(P_g\right)< f\left(X_i\right)\le f\left(P_i\right)\right\}$, and,
- S_C : the set of particles whose fitness values are worse than their personal best P_i, that is, $S_C=\left\{X_i : f\left(X_i\right)\ge f\left(P_i\right)\right\}$.

The incorrect selection happens when the personal best and global best cannot be correctly selected. The incorrect categorization of particles into the above sets affects the selection of personal best and global best. Therefore, we want to control the probability below to improve our correctness of selection.

$$P\{IS\} = P\left\{\left[\bigcup_{i\in S_A}\left(\bar{f}(X_i) > \tilde{f}(P_g)\right)\right]\cup\left[\bigcup_{j\in S_B}\left[\left(\bar{f}(X_j) \le \tilde{f}(P_g)\right)\cup\left(\bar{f}(X_j) > \tilde{f}(P_j)\right)\right]\right]\cup\left[\bigcup_{k\in S_C}\left(\bar{f}(X_k) \le \tilde{f}(P_k)\right)\right]\right\} \quad (3)$$

Note that $\tilde{f}(P_g)$ and $\tilde{f}(P_i)$ in (3) are all known sample mean values obtained in last iteration and $\bar{f}(X_i)$, $\bar{f}(X_j)$ and $\bar{f}(X_k)$ are random variables required to be estimated at this iteration.

Let

$$P^* = \max\left(\max_{i\in S_A} P\left\{\bar{f}(X_i) > \tilde{f}(P_g)\right\}, \max_{j\in S_B} P\left\{\bar{f}(X_j) \le \tilde{f}(P_g)\right\}, \max_{j\in S_B} P\left\{\bar{f}(X_j) > \tilde{f}(P_j)\right\}, \max_{k\in S_C} P\left\{\bar{f}(X_k) \le \tilde{f}(P_k)\right\}\right).$$

Thus, $P\{IS\}$ in (3) can be bounded by

$$P^* \le P\{IS\} \le \left(|S_A| + 2|S_B| + |S_C|\right)P^*.$$

If $\alpha_i > 0$ for each particle i, and $T \to \infty$ at each iteration, $\tilde{f}(P_g)$ and $\tilde{f}(P_i)$ go to $f(P_g)$ and $f(P_i)$, and $P\{IS\}$ goes to zero. The convergence rate of $P\{IS\}$ is equal to the convergence rate of P^*. Based on large deviation theory (Dembo and Zeitouni 1992, Szechtman and Yücesan 2008), for certain solution X with n replications, there exists a rate function $I(y)$ such that

$$\lim_{n\to\infty}\frac{1}{n}\log P\left\{\bar{f}(X) > y\right\} = -I(y), \text{ for } y > f(X), \text{ and}$$

$$\lim_{n\to\infty}\frac{1}{n}\log P\left\{\bar{f}(X) < y\right\} = -I(y), \text{ for } y < f(X).$$

Based on the lemma 1 of Glynn and Juneja (2004), we can obtain rate functions below.

$$\lim_{T\to\infty}\frac{1}{T}\log P\left\{\bar{f}(X_i) > f(P_g)\right\} = -\alpha_i I_i\left(f(P_g)\right), \text{ for } X_i \in S_A,$$

$$\lim_{T\to\infty}\frac{1}{T}\log P\left\{\bar{f}(X_j) \le f(P_g)\right\} = -\alpha_j I_j\left(f(P_g)\right), \text{ for } X_j \in S_B,$$

$$\lim_{T\to\infty}\frac{1}{T}\log P\left\{\bar{f}(X_j) > f(P_j)\right\} = -\alpha_j I_j\left(f(P_j)\right), \text{ for } X_j \in S_B,$$

$$\lim_{T\to\infty}\frac{1}{T}\log P\left\{\bar{f}(X_k) \le f(P_k)\right\} = -\alpha_k I_k\left(f(P_k)\right), \text{ for } X_k \in S_C.$$

So we have

$$\lim_{T\to\infty}\frac{1}{T}\log P^* = -\min_{i\in S_A, j\in S_B, k\in S_C}\left\{\alpha_i I_i\left(f(P_g)\right), \alpha_j I_j\left(f(P_g)\right), \alpha_j I_j\left(f(P_j)\right), \alpha_k I_k\left(f(P_k)\right)\right\}.$$

Thus,

$$\lim_{T\to\infty}\frac{1}{T}\log P\{IS\} = -\min_{i\in S_A, j\in S_B, k\in S_C}\left\{\alpha_i I_i\left(f(P_g)\right), \alpha_j I_j\left(f(P_g)\right), \alpha_j I_j\left(f(P_j)\right), \alpha_k I_k\left(f(P_k)\right)\right\}.$$

This means that $P\{IS\}$ will decay exponentially with increasing T at a rate given by $\min_{i\in S_A, j\in S_B, k\in S_C}\left\{\alpha_i I_i\left(f(P_g)\right), \alpha_j I_j\left(f(P_g)\right), \alpha_j I_j\left(f(P_j)\right), \alpha_k I_k\left(f(P_k)\right)\right\}$. For different allocation rules, $P\{IS\}$ will have different convergence rates. A good allocation rule should be the one that can obtain high convergence rate of $P\{IS\}$.

Based on the above analysis, we can model the computing budget allocation problem of the standard PSO from the perspective of maximizing the convergence rate of $P\{IS\}$ as below.

$$\begin{aligned} &\max \min_{i\in S_A, j\in S_B, k\in S_C}\left\{\alpha_i I_i\left(f(P_g)\right), \alpha_j I_j\left(f(P_g)\right), \alpha_j I_j\left(f(P_j)\right), \alpha_k I_k\left(f(P_k)\right)\right\} \\ &s.t. \quad \sum_{i=1}^{m}\alpha_i = 1 \\ &\qquad \alpha_i \ge 0. \end{aligned} \quad (4)$$

2.3 Computing Budget Allocation Model for PSObw method

Following a similar way, we can build the computing budget allocation model for PSObw method. In PSObw, the global best is the best particle in the swarm. For each particle in S_{worst}, we need to find the particle in S_{best} nearest to it as its personal best while the personal best for particles in S_{best} are themselves. Suppose X_b is the global best in the swarm. The probability of incorrect selection is the probability that the global best or personal best of any particle is incorrectly selected, which can be formulated as below.

$$P\{IS\} = P\left\{\left[\bigcup_{X_i \in S_{best}, X_i \neq X_b} \left(\bar{f}(X_b) > \bar{f}(X_i)\right)\right] \cup \left[\bigcup_{X_i \in S_{best}, X_j \in S_{worst}} \left(\bar{f}(X_i) > \bar{f}(X_j)\right)\right]\right\}. \tag{5}$$

By lemma 1 in Glynn and Juneja (2004), we have

$$\lim_{T\to\infty} \frac{1}{T} \log P\left\{\bar{f}(X_b) > \bar{f}(X_i)\right\} = -G_{bi}(\alpha_b, \alpha_i) = -\inf_y \left(\alpha_b I_b(y) + \alpha_i I_i(y)\right), \text{ for } X_i \in S_{best}, \text{ and}$$
$$\lim_{T\to\infty} \frac{1}{T} \log P\left\{\bar{f}(X_i) > \bar{f}(X_j)\right\} = -G_{ij}(\alpha_i, \alpha_j) = -\inf_y \left(\alpha_i I_i(y) + \alpha_j I_j(y)\right), \text{ for } X_i \in S_{best} \text{ and } X_j \in S_{worst}.$$

Therefore,

$$\lim_{T\to\infty} \frac{1}{T} \log P\{IS\} = -\min_{i \in S_{best}, j \in S_{worst}} \left\{G_{bi}(\alpha_b, \alpha_i), G_{ij}(\alpha_i, \alpha_j)\right\}.$$

This means that the convergence rate of $P\{IS\}$ in PSObw is $\min_{i \in S_{best}, j \in S_{worst}} \left\{G_{bi}(\alpha_b, \alpha_i), G_{ij}(\alpha_i, \alpha_j)\right\}$. The computing budget allocation model for PSObw can be built as below.

$$\begin{aligned} &\max \min_{i \in S_{best}, j \in S_{worst}} \left\{G_{bi}(\alpha_b, \alpha_i), G_{ij}(\alpha_b, \alpha_i)\right\} \\ &s.t. \quad \sum_{i=1}^{m} \alpha_i = 1 \\ &\qquad\;\; \alpha_i \geq 0. \end{aligned} \tag{6}$$

3 DEVELOPMENT OF ASYMPTOTICALLY OPTIMAL ALLOCATION RULES

We analyze the models (4) and (6) to get the asymptotic optimal allocation rules for two versions of PSO.

3.1 Asymptotically Optimal Allocation Rule for the Standard PSO

In the model (4), we can get the expression of $\alpha_i I_i\left(f\left(P_g\right)\right)$, $\alpha_j I_j\left(f\left(P_g\right)\right)$, $\alpha_i I_j\left(f\left(P_j\right)\right)$ and $\alpha_k I_k\left(f\left(P_k\right)\right)$ for certain distribution of $\hat{f}(X)$. Since all these terms are the linear and strictly increasing functions with respect to $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$, the minimum of linear functions is concave and still strictly increasing. So the model (4) is a convex optimization problem, which can be equivalently rewritten as,

$$
\begin{aligned}
&\max \quad z \quad s.t. \\
&\alpha_i I_i\left(f\left(P_g\right)\right) \geq z, \quad \text{for } X_i \in S_A \\
&\alpha_j I_j\left(f\left(P_g\right)\right) \geq z, \quad \text{for } X_j \in S_B \\
&\alpha_j I_j\left(f\left(P_j\right)\right) \geq z, \quad \text{for } X_j \in S_B \\
&\alpha_k I_k\left(f\left(P_k\right)\right) \geq z, \quad \text{for } X_k \in S_C \\
&\sum_{i=1}^{m} \alpha_i = 1 \\
&\alpha_i \geq 0.
\end{aligned}
$$

Because of the convexity of the maximization problem, the Karush-Kuhn-Tucker conditions can be used to find the best allocation rule. We can get the following theorem.

Theorem 1 *The allocation rule* $\alpha = \left(\alpha_1, \alpha_2, ..., \alpha_m\right)$ *is asymptotically optimal for model (4) if it satisfies the following conditions:*

(a) $\alpha_i I_i\left(f\left(P_g\right)\right) = \alpha_{j1} I_{j1}\left(f\left(P_g\right)\right) = \alpha_{j2} I_{j2}\left(f\left(P_j\right)\right) = \alpha_k I_k\left(f\left(P_k\right)\right)$*;*

(b) $\sum_{i=1}^{m} \alpha_i = 1$*;*

(c) $\alpha_i > 0$*.*

in which $j1 \in \left\{ j1 : X_j \in S_B \text{ and } I_j\left(f\left(P_j\right)\right) > I_j\left(f\left(P_g\right)\right) \right\}$, $j2 \in \left\{ j2 : X_j \in S_B \text{ and } I_j\left(f\left(P_j\right)\right) < I_j\left(f\left(P_g\right)\right) \right\}$.

Lemma 1 *When the performance of each particle is normally distributed, the optimal allocation rule for the standard PSO at each iteration, named as PSOs_OCBA, is*

$$
\alpha_i : \alpha_{j1} : \alpha_{j2} : \alpha_k = \frac{\sigma_i^2}{\left(f\left(X_i\right) - f\left(P_g\right)\right)^2} : \frac{\sigma_{j1}^2}{\left(f\left(X_{j1}\right) - f\left(P_g\right)\right)^2} : \frac{\sigma_{j2}^2}{\left(f\left(X_{j2}\right) - f\left(P_j\right)\right)^2} : \frac{\sigma_k^2}{\left(f\left(X_k\right) - f\left(P_k\right)\right)^2}. \quad (7)
$$

3.2 Asymptotically Optimal Allocation Rule for PSObw

Model (6) for PSObw is also a convex optimization problem when $G_{bi}\left(\alpha_b, \alpha_i\right)$ and $G_{ij}\left(\alpha_i, \alpha_j\right)$ are concave and strictly increasing functions with respect to $\alpha = \left(\alpha_1, \alpha_2, ..., \alpha_m\right)$. Referring to Glynn and Juneja (2004), $G_{bi}\left(\alpha_b, \alpha_i\right)$ can be expressed by $\alpha_b I_b\left(y\left(\alpha_b, \alpha_i\right)\right) + \alpha_i I_i\left(y\left(\alpha_b, \alpha_i\right)\right)$, and $G_{ij}\left(\alpha_i, \alpha_j\right)$ can be expressed by $\alpha_i I_i\left(y\left(\alpha_i, \alpha_j\right)\right) + \alpha_j I_j\left(y\left(\alpha_i, \alpha_j\right)\right)$. In the same way, model (6) for PSObw can be transformed into the following model:

$$
\begin{aligned}
&\max \quad z \\
&s.t. \quad z \leq \alpha_b I_b\left(y\left(\alpha_b, \alpha_i\right)\right) + \alpha_i I_i\left(y\left(\alpha_b, \alpha_i\right)\right), \quad \text{for } X_i \in S_{best} \\
&\qquad z \leq \alpha_i I_i\left(y\left(\alpha_i, \alpha_j\right)\right) + \alpha_j I_j\left(y\left(\alpha_i, \alpha_j\right)\right), \quad \text{for } X_i \in S_{best}, X_j \in S_{worst} \\
&\qquad \sum_{i=1}^{m} \alpha_i = 1 \\
&\qquad \alpha_i \geq 0.
\end{aligned}
$$

Applying the Karush-Kuhn-Tucker conditions, we can get the conditions of the optimal allocation rule.

Theorem 2 *If an allocation rule is the asymptotically optimal allocation rule to minimize the probability of incorrect selection in model (6), it satisfies the following conditions:*

(a) $\alpha_b I_b\left(y(\alpha_b,\alpha_i)\right)+\alpha_i I_i\left(y(\alpha_b,\alpha_i)\right)=\alpha_i I_i\left(y(\alpha_i,\alpha_j)\right)+\alpha_j I_j\left(y(\alpha_i,\alpha_j)\right)$ for $X_i \in S_{best}, X_j \in S_{worst}$;

(b) $\sum_{i=1}^{m}\alpha_i = 1$;

(c) $\alpha_i > 0$.

We cannot get a closed-form allocation rule from theorem 2 for model (6). Therefore, we can simplify the model under some assumptions such that a closed-form allocation rule can be derived and implemented as a good allocation rule (no guarantee of optimality) into some algorithms. For convenience, we categorize particles into different subsets:

- S_{best}^0: the set of particles which belong to S_{best} and are not the personal best of any particles in the set S_{worst},
- S_{best}^1: the set of particles which belong to S_{best} and are the personal best of at least one particles in the set S_{worst},
- S_{worst}^i: the set of particles which belong to S_{worst} and treat particle *i* as its personal best.

Lemma 2 *Under the assumptions: (i) the performance of each particle is normally distributed; (ii)* $\alpha_b >> \alpha_i >> \alpha_j$ *for* $X_i \in S_{best}, X_j \in S_{worst}$*; (iii)* $\max_{X_k \in S_{best}^0} f(X_k) < \min_{X_i \in S_{best}^1} f(X_i)$*. The asymptotically optimal allocation rule for model (6), named as PSObw_OCBA, is*

(a) For $X_i \in S_{best}^1$ *which satisfies* $\dfrac{\left(f(X_b)-f(X_i)\right)^2}{\sigma_i^2/\alpha_i} \le \min_{X_j \in S_{worst}^i} \dfrac{\left(f(X_i)-f(X_j)\right)^2}{\sigma_j^2/\alpha_j}$

$$\alpha_k : \alpha_i : \alpha_j = \frac{\sigma_k^2}{\left(f(X_b)-f(P_k)\right)^2} : \frac{\sigma_i^2}{\left(f(X_b)-f(X_i)\right)^2} : \frac{\sigma_j^2}{\left(f(X_i)-f(X_j)\right)^2}$$

(b) For $X_i \in S_{best}^1$ *which satisfies* $\dfrac{\left(f(X_b)-f(X_i)\right)^2}{\sigma_i^2/\alpha_i} > \min_{X_j \in S_{worst}^i} \dfrac{\left(f(X_i)-f(X_j)\right)^2}{\sigma_j^2/\alpha_j}$

$$\alpha_k : \alpha_i : \alpha_j = \frac{\sigma_k^2}{\left(f(X_b)-f(P_k)\right)^2} : \sqrt{\sum_{X_j \in S_{worst}^i} \frac{\sigma_i^2 \sigma_j^2}{\left(f(X_i)-f(X_j)\right)^4}} : \frac{\sigma_j^2}{\left(f(X_i)-f(X_j)\right)^2}$$

(c)

$$\alpha_b = \sigma_b \sqrt{\sum_{k \in S_{best}^0} \frac{\alpha_i^2}{\sigma_i^2} + \sum_{i \in S_{best}^1} \left(\frac{\alpha_i^2}{\sigma_i^2} - \sum_{j \in S_{worst}^i} \frac{\alpha_j^2}{\sigma_j^2}\right)}$$

in which $X_k \in S_{best}^0$, $X_i \in S_{best}^1$ *and* $X_j \in S_{worst}^i$.

3.3 Procedures to Implement PSOs_OCBA and PSObw_OCBA

The implementation of PSOs_OCBA and PSObw_OCBA in Lemma 1 and Lemma 2 depends on the function of distribution. In practice, a sequential procedure is provided here to implement these allocation rules. After we get the new location of each particle by using equation (2), the procedure shown below will be applied to get the sample mean value of each particle and select the personal best and global best for the next iteration of updating in PSO algorithm. Each particle is initially simulated with n_0 replications at the first stage, and the additional replications are allocated to particles incrementally from Δ replications to be allocated at each subsequent stage until the simulation budget T is exhausted.

Procedure. PSOs_OCBA and PSObw_OCBA Procedure	
Initialization	$l \to 0$; Perform n_0 simulation replications for all m particles; Let $N_1^l = N_2^l = \cdots = N_m^l = n_0$ and $T^l = mn_0$.
Loop while $T^l < T$ **do**	
Updating	Calculate sample means and sample variances of particles based on the simulation outputs;
Allocation	Let $T^{l+1} = T^l + \Delta$ and calculate the new budget allocation $N^{l+1} = \left(N_1^{l+1}, N_2^{l+1}, \cdots N_m^{l+1}\right)$ based on PSOs_OCBA or PSObw_OCBA;
Simulation	Perform additional $\max\left(0, N_i^{l+1} - N_i^l\right)$ replications for each particle i and let $N_i^{l+1} = \max\left(N_i^{l+1}, N_i^l\right)$; $l = l+1$.
End of loop	
Stopping	Select the personal best and global best based on particles' sample mean values to update P_i and P_g in the PSO algorithms.

4 NUMERICAL EXPERIMENT

In the experiment, the performance of PSOs_OCBA and PSObw_OCBA are compared with equal allocation rule (PSOs_EA) and PSObw_EA.

We use the following two functions to test these four allocation rules.

(1) Sphere function

$$f(X) = \sum_{i=1}^{d} x_i^2 \quad ;$$

(2) Printer function

$$f(X) = \sum_{i=1}^{d} i x_i^2 + \sum_{i=1}^{d} 20 i \sin^2\left(x_{i-1}\sin x_i - x_i + \sin x_{i+1}\right) + \sum_{i=1}^{d} i \log_{10}\left(1 + i\left(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1\right)^2\right).$$

The optimal solutions of these two functions are both (0,0) in two dimensional space and the minimal values are both zero. We set the feasible range of each dimension as [-50,50] and the variance of 10^2 is added to each function to simulate the stochastic environment. Our goal is to find the optimal solution for each function. For both the standard PSO and PSObw, we generate 20 particles at each iteration. The values of c_1 and c_2 in equation (1) are set to be 2.05 as the value recommended in Bratton and Kennedy (2007). The constrictive factor χ is set to be an decreasing function of the iteration number, that is,

$$\chi = \frac{\max\_iter + 1 - i}{\max\_iter + 1} \cdot \frac{2}{\left|2 - c1 - c2 - \sqrt{\left(c1 + c2\right)^2 - 4\left(c1 + c2\right)}\right|}$$

in which max_*iter* is the maximal number of iterations. For the computing budget allocation, we set Δ equal to 100 and n_0 equal to 10 in all numerical experiments. The total computing budget for each iteration at PSO is 3000.

The performances of all allocation rules are shown in figure 1 and figure 2. From the figures, we can see both PSObw_OCBA and PSOs_OCBA perform better than PSObw_EA and PSOs_EA respectively. It can be concluded that integrating OCBA into PSO does support PSO to converge to the optimal solution faster in the above functions.

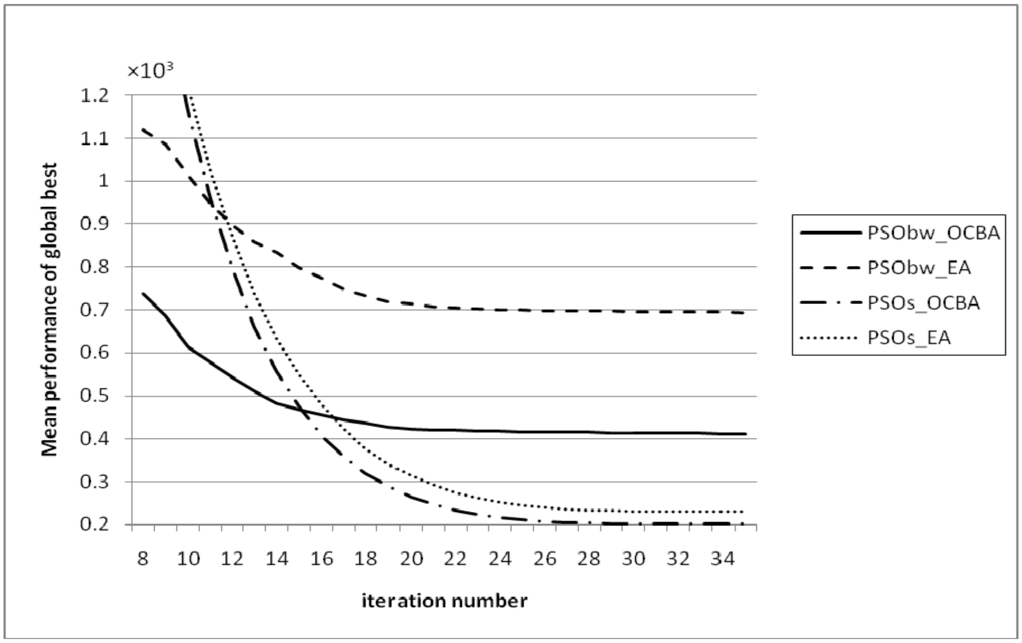

Figure 1: Numerical result of Sphere function

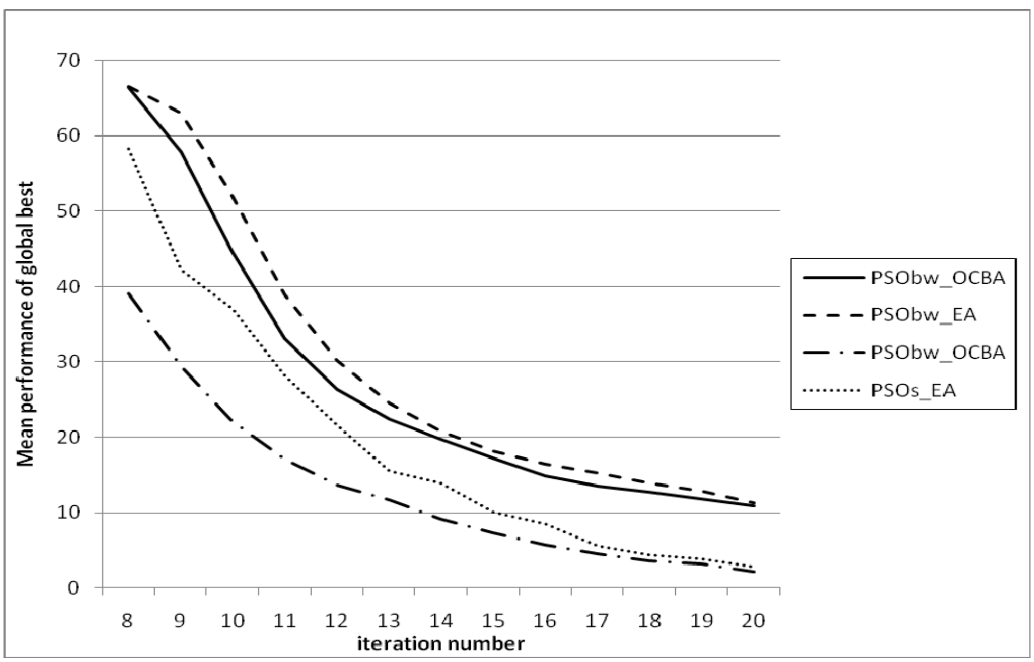

Figure 2: Numerical result of Printer function

5 CONCLUSIONS

The PSO method has been widely used in global optimization problem, but the computing budget allocation problem for PSO under stochastic environment has been seldom studied. In this paper, we integrate the concept of OCBA into PSO. The conditions for the asymptotically optimal allocation rules are derived for the standard PSO and PSObw. Under some assumptions, we manage to get the allocation rules PSOs_OCBA and PSObw_OCBA in closed form and easily implementable. The numerical result shows PSOs_OCBA and PSObw_OCBA are better than PSOs_EA and PSObw_EA respectively. The integration of OCBA concept into PSO does improve the efficiency of PSO.

REFERENCES

Banks, A., J. Vincent, and C. Anyakoha. 2007. "A Review of Particle Swarm Optimization. Part I: Background and Development." *Natural Computing* 6: 467-484.

Banks, A., J. Vincent, and C. Anyakoha. 2008. "A Review of Particle Swarm Optimization. Part II: Hybridisation, Combinatorial, Multicriteria and Constrained Optimization, and Indicative Applications." *Natural Computing* 7: 109-124.

Bratton, D., and J. Kennedy. 2007. "Defining a Standard for Particle Swarm Optimization." In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*. 120-127. IEEE Press.

Chen, C. H., D. He, M. Fu, and L. H. Lee. 2008. "Efficient Simulation Budget Allocation for Selecting an Optimal Subset." *INFORMS Journal on Computing* 20(4):579-595.

Chen, C. H., J. Lin, E. Yücesan, and S. E. Chick. 2000. "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization." *Journal of Discrete Event Dynamic Systems: Theory and Applications* 10:251-270.

Chew, E. P., L. H. Lee, S. Y. Teng, and C. H. Koh. 2009. "Differentiated Service Inventory Optimization using Nested Partitions and MOCBA." *Computers and Operations Research* 36(5): 1703-1710.

Clerc, M., and J. Kennedy. 2002. "The Particle Swarm: Explosion, Stability And Convergence In A Multi-Dimensional Complex Space." *IEEE Transactions on Evolutionary Comput*ing. 6(1): 58-73.

Dembo, A., and O. Zeitouni. 1992. *Large Deviations Techniques and Applications*. Boston, MA: Jones and Bartlett Publishers.

Glynn, P., and S. Juneja. 2004. "A Large Deviations Perspective on Ordinal Optimization." In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 577-585. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gurkan, G., A.Y. Ozge, and S. M. Robinson. 1994. "Sample Path Optimization in Simulation." In *Proceedings of the 1994 Winter Simulation Conference*, edited by J.D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 247-254. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

He, D., L. H. Lee, C. H. Chen, M. Fu, and S. Wasserkrug. 2010. "Simulation Optimization Using the Cross-Entropy Method with Optimal Computing Budget Allocation." *ACM Transactions on Modeling and Computer Simulation* 20(1): Article 4.

Kennedy, J. 2000. "Stereotyping: Improving Particle Swarm Performance With Cluster Analysis." In *Proceedings of IEEE congress on evolutionary computation*. 1507–1512. IEEE Press.

Kennedy, J., and R. C. Eberhart. 1995. "Particle Swarm Optimization." In *Proceedings of IEEE International Conference*. 1942-1948. IEEE Press.

Lee, L. H., W. P. Wong, and W. Jaruphongsa. 2009. "Data Collection Budget Allocation for Stochastic Data Envelopment Analysis." In *Proceedings of the 2009 INFORMS Simulation Society Research Workshop,* edited by L.H. Lee, M.E. Kuhl, J. W. Fowler, and S. Robinson, 71-74.

Pan, H., L. Wang, and B. Liu. 2006. "Particle Swarm Optimization for Function Optimization in Noisy Environment." *Applied Mathematics and Computation* 181:908–919.

Rubinstein, R. Y., and A. Shapiro. 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Approximation using the Score Function Method*. John Wiley & Sons.

Shi, L., and S. Olafsson. 1997. "An Integrated Framework for Deterministic and Stochastic Optimization." In *Proceedings of the 1997 Winter Simulation Conference,* edited by S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 358-365. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Shi, Y., and R. C. Eberhart. 1998. "A Modified Particle Swarm Optimizer." In *Proceedings of the IEEE international conference on evolutionary computation*. 69–73. IEEE Press.

Storn, R., and K. Price. 1997. "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces." *Journal of Global Optimization* 11(4):341–359.

Szechtman, R., and E. Yücesan. 2008. "A New Perspective on Feasibility Determination." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 273–280. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

SI ZHANG is a Ph.D. candidate in the Department of Industrial and Systems Engineering, National University of Singapore. She received her B.Eng. degree with a specialization in Industrial Engineering from Nanjing University, China in 2008. Her research interest is in simulation optimization. Her current work is on the issue of computing budget allocation for optimal subset selection problem and the integration of R&S procedures with simulation optimization algorithms. Her email address is g0800221@nus.edu.sg.

PAN CHEN is an undergraduate student in the Department of Industrial and Systems Engineering, National University of Singapore. Her final year project is about the integration of OCBA and PSO from the perspective of maximizing the probability of correct selection. Her email address is chenpan@nus.edu.sg.

LOO HAY LEE is an Associate Professor and Deputy Head (Graduate Studies and Research) in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.S. (Electrical Engineering) degree from the National Taiwan University in 1992 and his Ph.D. degree in 1997 from Harvard University. He is currently a senior member of IEEE, a committee member of ORSS, and a member of INFORMS. His research interests include production planning and control, logistics and vehicle routing, supply chain modeling, simulation-based optimization, and evolutionary computation. His email address is iseleelh@nus.edu.sg.

EK PENG CHEW is an Associate Professor and Deputy Head (Undergraduate Studies) in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.S. (Civil Engineering) degree from the National University of Singapore in 1987 and his Ph.D. degree in 1992 from Georgia Institute of Technology. His research interests include logistics and inventory management, system modeling and simulation, and system optimization. His email address is isecep@nus.edu.sg.

CHUN-HUNG CHEN is a Professor of Electrical Engineering at National Taiwan University. He received his Ph.D. from Harvard University in 1994. His research interests are mainly in development of very efficient methodology for simulation and optimization and its applications. Dr. Chen has served as Co-Editor of the *Proceedings of the 2002 Winter Simulation Conference* and Program Co-Chair for 2007 INFORMS Simulation Society Workshop. He is currently an associate editor of *IEEE Transactions on Automatic Control*, area editor of *Journal of Simulation Modeling Practice and Theory*, associate editor of *International Journal of Simulation and Process Modeling*, and simulation department editor for *IIE Transactions*. His email address is cchen9@gmu.edu.