Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

IMPROVING THE EFFICIENCY OF EVOLUTIONARY ALGORITHMS FOR LARGE-SCALE OPTIMIZATION WITH MULTI-FIDELITY MODELS

Chun-Chih Chiu
James T. Lin

Industrial Engineering & Engineering Management
National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road
Hsinchu, Taiwan 30013, R.O.C.

Si Zhang
Lu Zhen

School of Management
Shanghai University
No. 99, Shang-Da Road
Shanghai, 200444, CHINA

Edward Huang

Systems Engineering & Operations Research
George Mason University
4400 University Drive, MS 4A6
Fairfax, VA 22030, USA

ABSTRACT

Large scale optimization problems are often with complex systems and large solution space, which significantly increase their computing cost. The idea of ordinal transformation (OT) is proposed in the method MO2TOS which can improve the efficiency of solving optimization problems with limited scale solution space by using multi-fidelity models. In this paper, we integrate OT with evolutionary algorithms to speed up the solving of large-scale problems. Evolutionary algorithms are employed to search the solutions of low-fidelity model from a large solution space and provide a good direction to the OT procedure. Meanwhile, evolutionary algorithms need to determine how to select solutions from multi-fidelity models after the OT procedure to update the next generation. We theoretically show the improvement by using multi-fidelity models and employ genetic algorithm (GA) as an example to exhibit the detailed implementation procedure. The numerical experiments demonstrate that the new method can lead to significant improvement.

1 INTRODUCTION

As a popular research area, optimization problem often arises in varied applications and has been well studied in many years. A lot of methods have been developed to solve optimization problems, such as Linear Programming, Non-linear Programming, and Dynamic Programming. These conventional computational methods are analytically well developed but have certain requirements on the problem configuration. However, the real world problems are usually large-scale optimization problems. These problems have large solution space, the noisy or incomplete data, or multimodality. These significantly increase the difficulty for the conventional methods solving the real optimization problems due to these approaches' rigorous requirements on the problem structure.

Some evolutionary techniques are then proposed or applied in optimization area for tackling the

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

large scale problems with complex configuration because of their few or no assumptions about the optimization problem, such as Genetic Algorithms (GAs) (Holland 1975), Cross-Entropy Method (CE) (Rubinstein 1999), Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995, Bratton and Kennedy 2007), and so on. Although the evolutionary algorithms provide useful tools for large-scale optimization problems, the computational efficiency is still one challenge point. One reason is the large and nonlinear solution spaces of large-scale optimization problems. The other reason is the high computing cost to evaluate the solutions due to the very complex systems.

One way to tackle the computational efficiency issue of large-scale problems is the multiple-fidelity optimization. For the same system, it is often possible to build multiple evaluating models with different fidelity levels. High-fidelity models have high accuracy on predicting the performance of a solution but they may take a long time to run, which leads to the high computing cost. Low-fidelity models usually have biased estimation of the objective value of solutions but their computational cost is very low and can provide certain useful information to help us find the true best solution(s).

There are some literatures related to multi-fidelity optimization for complex systems optimization problems. Huang et al. (2006) proposed a Multi-Fidelity Sequential Kriging Optimization (MFSKO) procedure and use the expected improvement (EI) as a measurement to determine the next solution to evaluate and the level of fidelity for that solution. Moore (2012) used the value of information (VOI) instead of EI as the maximized objective to determine the next solution and the fidelity of model to evaluate. Both of these two methods use Kriging to predict the bias of the low-fidelity model. However, kriging may require a large number of design points to perform well when the response surface is highly nonlinear and multi-modal, and/or the dimension of the solution space is high. Xu et al. (2014, 2016) and Huang et al. (2015) then proposed a novel ordinal transformation (OT) framework, named as MO^{2}TOS to effectively use the information of multi-fidelity models to solve deterministic optimization problems with a finite number of solutions.

In this paper, we integrate the idea of OT with evolutionary algorithms (GA) to improve their efficiency for large-scale deterministic optimization problems with multi-fidelity models. Because the low-fidelity model has low computing cost, much more solutions in the low-fidelity model can be generated and evaluated at each iteration of evolutionary algorithms. OT helps find good solutions to be evaluated by high-fidelity model and improve the search direction of evolutionary algorithms. Therefore, by applying OT with multi-fidelity models, the evolutionary algorithms can increase its capability in exploration and exploitation.

The rest of the paper is organized as follows. In Section 2, we propose a way to use low-fidelity models for evolutionary algorithms and theoretically analyze the benefits of using low-fidelity models. Section 3 provides an algorithm integrating with GA. The numerical results are shown in Section 4. Section 5 concludes the whole paper.

2 GENERAL GUIDELINES MULTIPLE FIDELITY MODELS IN RANDOM SAMPLING OPTIMIZATION ALGORITHMS

2.1 Language Introduction of Ordinal Transformation

The Ordinal Transformation is the main part of MO^{2}TOS. Firstly, all solutions are evaluated by the low-fidelity model. The Ordinal Transformation (OT) methodology then transforms the original solution space into a one-dimensional ordinal space based on the ranking of solutions using the low-fidelity model. Compared to the original solution space, which can be highly nonlinear, multi-modal, high-dimensional, and include a mix of discrete and categorical decision variables, the ordinal space is one-dimensional and often owns good structure.

Xu et al. (2014, 2016) apply OT to a machine allocation problem in a flexible manufacturing system with an objective to maximize system resilience as measured by steady-state cycle time (smaller is better) under demand disruptions and machine failures. There are 37 machines and 5 workstations. The problem is to determine the number of machines allocated to each workstation such that the average cycle time can be minimized. The constraints is the number of machines in each work station must be between 5 and 10. There are total 780 feasible solutions for the optimization problem. Figure 1 shows the performance of each solution evaluated by high-fidelity model. A low-fidelity model for the problem is built by using Jackson network analysis. Figure 2 shows the performance of each solution in low-/high-fidelity models after OT. From Figures 1 and 2, we can see that using low-fidelity model by OT can help to re-organize the structure of high-fidelity model in the original solution space and show a global trend. By this trend, the efficiency of finding good solutions can be significantly improved.

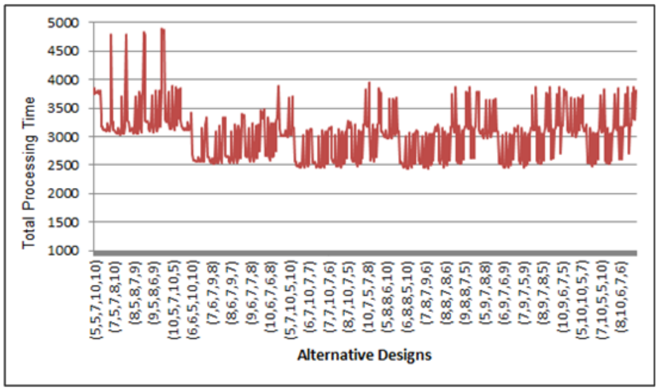

Figure 1: Cycle times in the original decision space.

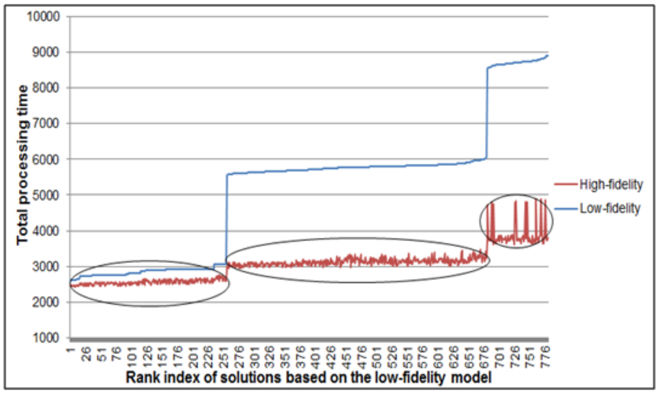

Figure 2: Cycle times in the transformed decision space.

2.2 Theoretical Analysis About The Benefits of Using Low-Fidelity Model

In this section, we analyze the application of OT in evolutionary algorithms. The following is the list of notations.

X: a solution, of dimension K;

$g(X)$/f(X): the performance of solution X according to the low/high-fidelity deterministic model;

N_h: the total number of solutions evaluated using high-fidelity model at each iteration;

sN_h: the total number of solutions evaluated using low-fidelity model at each iteration;

For each solution X, we assume that $g(X)$ and f(X) follow bi-variate normal distribution as below.

$$N\left(\begin{bmatrix}\mu_f \\ \mu_g\end{bmatrix}, \begin{bmatrix}\sigma_f^2 & \rho\sigma_f\sigma_g \\ \rho\sigma_f\sigma_g & \sigma_g^2\end{bmatrix}\right)$$

Therefore, $f(X)$ and $g(X)$ can be expressed as

$$f(X) = ag(X) + \delta(X)$$

in which $a = \sigma_f / \sigma_g$ and $\delta(x) = \mu_f - \rho\mu_g\sigma_f / \sigma_g - \sqrt{1-\rho^2 Z}$ (Z is a standard normal random variable). In the formula, we also can see that $\delta(x)$ follows normal distribution and is independent to $g(X)$. Hence, the best N_h solutions from the sN_h solutions evaluated by low-fidelity model are selected to run the high-fidelity model. Based on the above assumptions and parameter settings, we have the following lemmas.

Lemma 1. *If we only use high-fidelity model during the implementation of evolutionary algorithms, the best solution of each iteration has the following expected performance.*

$$E[f(X_b)] \approx \mu_f + \Phi^{-1}\left(\frac{1}{N_h+1}\right)\sigma_f$$

Lemma 2. *If we use both low- and high- fidelity models during the implementation of evolutionary algorithms, the best solution of each iteration has the following upper bound of the expected performance.*

$$E[f(X_{sb})] \le aE[g(X_1)] + E[\delta(X_1)] \approx \mu_f + \Phi^{-1}\left(\frac{1}{sN_h+1}\right)\rho\sigma_f$$

Based on Lemma 1 and 2, we can see there always exists a large enough s such that the best solutions found by using two fidelity levels models is better that that found by only using high fidelity model, which theoretically shows the benefits of using low-fidelity models in evolutionary algorithms.

3 PROCEDURES TO IMPLEMENT GA$_{\text{OT}}$

In this section, we proposed an algorithm, GA$_{\text{OT}}$, in a large design space with multiple-fidelity models. In the large design space, even though the low-fidelity model cannot evaluate all solutions. The quality of solutions selected based on low-fidelity model performance will impact searching efficiency in high-fidelity model. We thus use GA to explore good low-fidelity solutions to OT stage. GA is a stochastic search algorithm which was proposed by Holland 1975 and it has been successfully applied to a wide variety optimization problems (Lin and Chen 2015). The GA$_{\text{OT}}$ implements the GA and the OT alternatively in the total evolution process. The GA works on and updates the population of the low-fidelity model, while the OT is implemented to determine the evaluation of high-fidelity model and feedback evaluated chromosomes to GA for updating the next generation. The schematic structure of the GA$_{\text{OT}}$ is depicted in Figure 3.

The main components of GA$_{\text{OT}}$ are the chromosome encoding, the evaluation of fitness function, rank, selection, crossover and mutation. In the chromosome encoding, we use the binary encoding. The evaluation fitness function of GA is performed using the low-fidelity model. When GA finish evaluating all solutions of the low-fidelity model, these results will input the OT procedure. The first step of OT, ranks chromosome by using low-fidelity performance. We then evaluate high-fidelity model from the transformed space until all computing budget of each generation has been used. Now there are some solutions are evaluated and the other solutions are not evaluated. The selection process in GA determines which of the chromosomes from the current population will crossover to create new chromosomes. Based on the chromosomes that better fitness values have better chances of selection, we first select evaluated solutions of the high-fidelity model according to their rank to the mating pool. This means evaluated solutions have higher priority to be selected in the crossover process. For the rest candidates, we use the non-evaluated solutions based on their rank of low-fidelity to fill the mating pool. The mating pool is performed roulette wheel method, where individuals are given a probability of being selected which is directly proportionate to their rank. The crossover operator in this study is two cut-point exchange from two selected parent chromosomes to produce one child chromosome. In the mutation strategy, two genes are chosen randomly and their bit in the chromosome are inverted. The detailed pseudocode of the GA$_{\text{OT}}$ is described in Table 1.

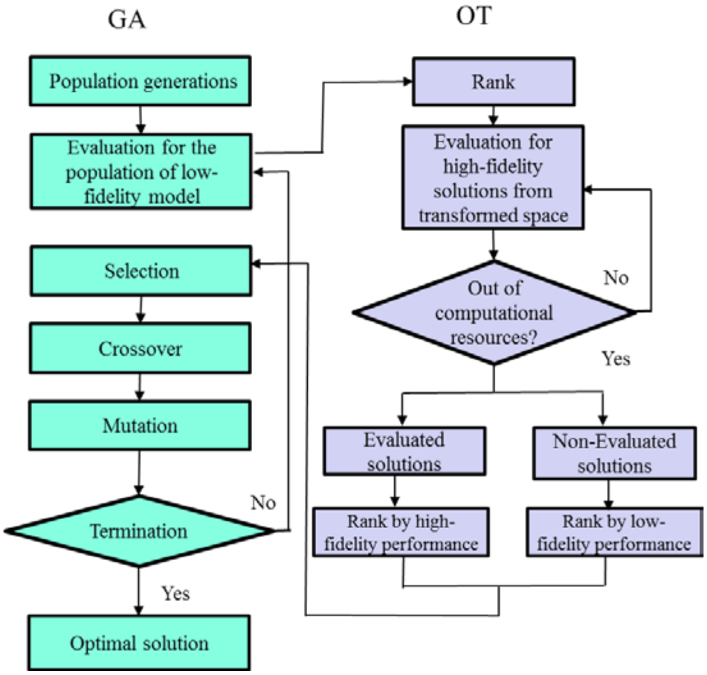
GA
OT
Population generations
Evaluation for the population of low-fidelity model
Selection
Crossover
Mutation
Termination
No
Yes
Optimal solution
Rank
Evaluation for high-fidelity solutions from transformed space
Out of computational resources?
No
Yes
Evaluated solutions
Non-Evaluated solutions
Rank by high-fidelity performance
Rank by low-fidelity performance

Figure 3: Schematic structure of GAOT.

Table 1: Procedure of GAOT.

Initialization	Set the following parameters: G_{max}: maximum generation; N_h: the population size of high-fidelity (computing budget); N_l: the population size of low-fidelity; Cr: crossover rate; Mr: mutation rate. N_L chromosomes are randomly generated.
Loop while generation < G_{max} do	
Evaluation	All chromosomes are evaluated by using low-fidelity model.
Loop while computing budget < N_h do: OT	
	Rank N solutions using low-fidelity performance. Perform ordinal transformation using the ranking of all solutions Select solutions from the transformed space as design points and evaluate them with high-fidelity models.
End of Loop	
Updating (GA)	For i=1 to N_h Select evaluated solutions to mating pool according to their rank of high-fidelity model End for

```
                    For i= $N_h$ to $N_l$
                        Select non-evaluated solutions to mating pool according to their rank
                        of low-fidelity model.
                      End for

                    Perform roulette wheel method in mating pool.
                    For i=1 to $N_l$
                        Rand←uniform(0,1)
                        If Rand <Cr
                            Then crossover
                        End if
                        Rand←uniform(0,1)
                        If Rand<Mr
                              Then mutation
                        End if
                    End for
End of Loop
Stopping           The optimal solution and its performance are output.
```

4 NUMERICAL EXPERIMENT

In this section, we will compare the performance of GA_{OT} with GA on two test problems which are function optimization and simultaneous scheduling problem.

4.1 Multiple-Fidelity Function Optimization

The following two functions are used to test GA_{OT} and GA. These functions can be controlled as the different correlation between low-fidelity and high-fidelity function through a parameter (Toal 2015). We perform two different correlations of each function. All experiment results are based on 30 IID runs and the average performances are plotted.

(1) Branin function
High-fidelity

$$f_h = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1)$$

The minimum value of this function is 0.3979. The feasible range of x_1 is [-5,10] and x_2 is [0,15].

Low-fidelity

$$f_l = f_h - (A_1 + 0.5)\left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2$$

where A_1 is a parameter to control the correlation between high-fidelity and low-fidelity. In this function, we set A_1=0 and 0.36 which the correlation is 0.9587 and 0.7126, respectively.

(2) Paciorek function
High-fidelity

$$f_h = \sin\left(\frac{1}{x_1 x_2} \right)$$

The minimum value of this function is -1. The feasible range of each dimension is [0.3,1].

Low-fidelity

$$f_l = f_h - 9A_2^2 \cos\left(\frac{1}{x_1 x_2}\right)$$

where A_2 is a parameter to control the correlation between high-fidelity and low-fidelity. In this function, we set $A2$=0.1 and 0.35 which the correlation is 0.9974 and 0.7104, respectively.

The population size of high-fidelity for GA and GA_{OT} are set to 10. In GA_{OT}, the population size of using low-fidelity is set to 30 and 100. In all following experiments, we collect the best performance, worst performance, average performance and standard deviation of all runs.

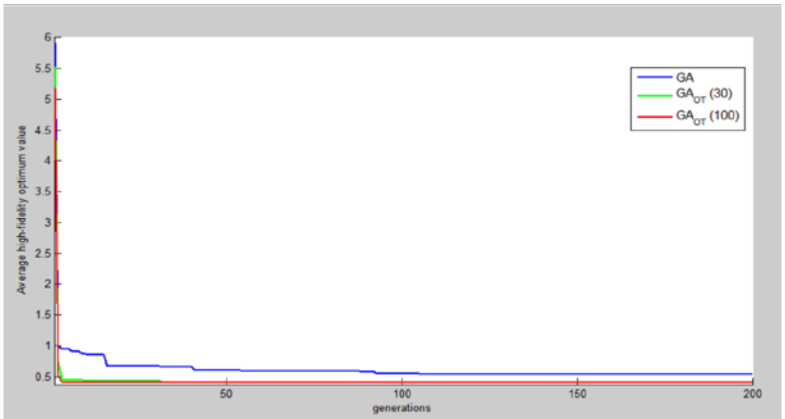

Figure 4: Results of GA and GAOT under lower correlation for Branin function.

The performances of GA and GA_{OT} for Branin function with different correlation are shown in Table 2. The results show that GA_{OT} has better performance than GA. Using more population from low-fidelity can help GA_{OT} to find a better solution. Higher correlation of low-fidelity model can provide more useful information for searching the optimal solution.

Table 2: Experiment result for Branin function with different correlations.

		ρ =0.7126		**ρ =0.9587**	
	GA (10)	GA_{OT} (30)	GA_{OT} (100)	GA_{OT} (30)	GA_{OT} (100)
Best performance	0.3987	0.3979	0.3979	0.3979	0.3979
Worst performance	2.3121	0.4523	0.3979	0.4015	0.3979
Avg. performance	0.5296	0.404	0.398	0.3985	0.3979
Std. of all runs	0.3534	0.0165	1.90E-04	0.0014	1.81E-06

The convergence plots of GA and GA_{OT} for Branin function with different correlation are shown in Figures 4 and 5. From these figures, we can see OT into GA can support GA to converge to the optimal solution faster.

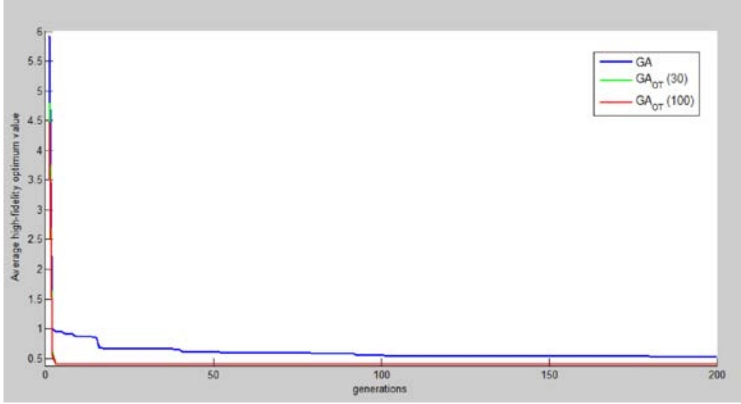

Figure 5: Results of GA and GAOT under higher correlation for Branin function.

Table 3 reports the performances of GA and GA_{OT} for Paciorek function with different correlation. From Table 3, we observe GA_{OT} still can acquire a better solution than GA. It can be concluded that both of using more low-fidelity solutions and higher correlation of low-fidelity model can enhance the searching efficiency of GA in these experiments.

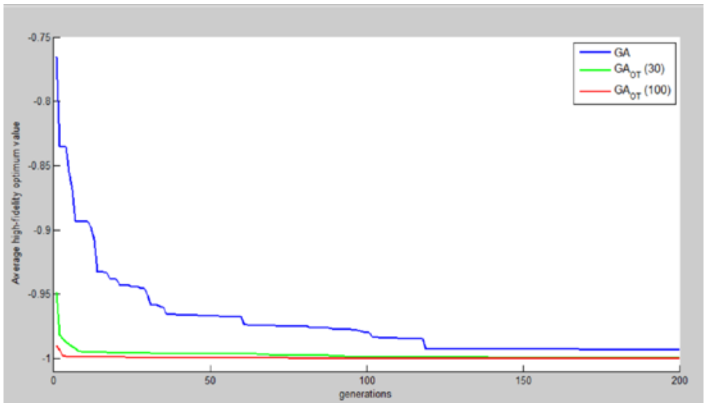

Figure 6: Results of GA and GAOT under lower correlation for Paciorek function.

Table 3: Experiment result for Paciorek function with different correlation.

		ρ **=0.7104**		ρ **=0.9974**	
	GA (10)	GA_{OT} (30)	GA_{OT} (100)	GA_{OT} (30)	GA_{OT} (100)
Best performance	-1	-1	-1	-1	-1
Worst performance	-0.9537	-0.9755	-0.9953	-0.9941	-0.999
Avg. performance	-0.9934	-0.9975	-9998	-0.9997	-1
Std. of all runs	0.0144	0.006	0.00086444	0.0011	0.000039466

The convergence plots of GA and GA_{OT} for Paciorek function with different correlation are shown in Figures 6 and 7. GA_{OT} is also easier to get away from local optimum in this function. Using low-fidelity model can help GA to explore high-fidelity solutions. The scale of improvement depends on the population size of low-fidelity and the quality of low-fidelity model.

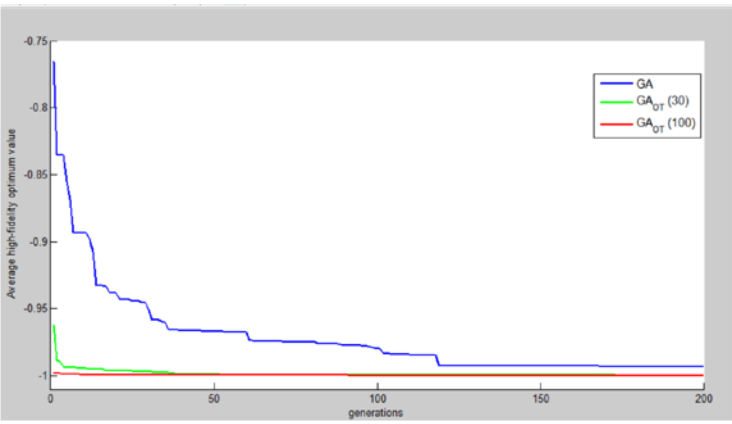

Figure 7: Results of GA and GAOT under higher correlation for Paciorek function.

4.2 Simultaneous Scheduling Problem

We test GA_{OT} and GA on another problem, which is a simultaneous scheduling problem in the flexible manufacturing system (FMS).

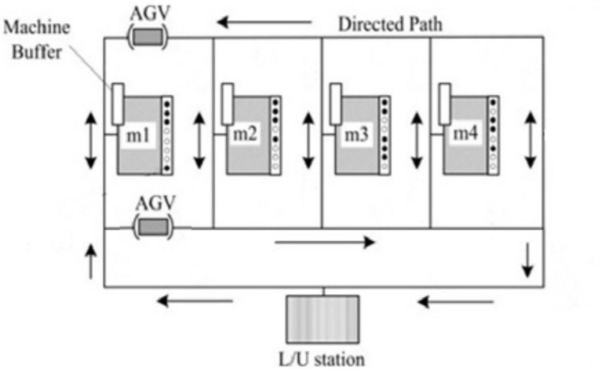

Figure 8: An illustration of FMS.

Simultaneous scheduling problem in FMS is to consider both machines and automatic guided vehicles (AGVs) to determine the starting and completion times of each job for given production sequences and the dispatching of transportation vehicles. The objective of this problem is to minimize the makes-pan. The production environment of the FMS consists of a group of computer numerically controlled (CNC) machines, AGVs, and a computer control system; these devices are connected to process different products. An example of an FMS is presented in Figure 8. There is a load/unload (L/U) station, two AGVs, and four workstations, and local buffers of each workstation.

We consider two fidelities in simultaneous scheduling problem which are alternative machine and zone control of preventing vehicles deadlock. Alternative machine means if a job cannot be processed because the original machine is unavailable, an alternative machine can be used to process the job for reducing the waiting time. Zone control is a technique applied to prevent deadlock of vehicles. With zone-control the system guide path is broken into discrete regions where vehicle deadlocks are avoided by allowing only a single vehicle into a control zone at any given time These two fidelities will significantly impact system performance and the schedule whether a feasible solution in practice. There are many interactions among jobs sequence, alternative machines, and AGVs behavior. Hence, this problem is high-dimension, large design space, and complexity.

In this study, alternative machine in the first fidelity model and the zone control of vehicle transportation in the second fidelity model. The third fidelity model, which is the model with the highest fidelity, considers both alternative machine and zone control of the vehicles routing. The correlation of low-fidelity 1 and 2 is 0.9613 and 0.4358, respectively.

The data set is adapted from a literature (Kumar et al., 2011). There are five jobs, four machines, one load/unload station and two identical AGVs in this problem. The speed of AGVs is one meter per second. This problem is implemented by using discrete-event simulation model. This programming was developed using Siemens plant simulation ™.

The population size of high-fidelity in simultaneous scheduling problem for GA and GA$_{OT}$ are set to 20. In GA$_{OT}$, the population size of using low-fidelity is set to 50, 100 and 200.

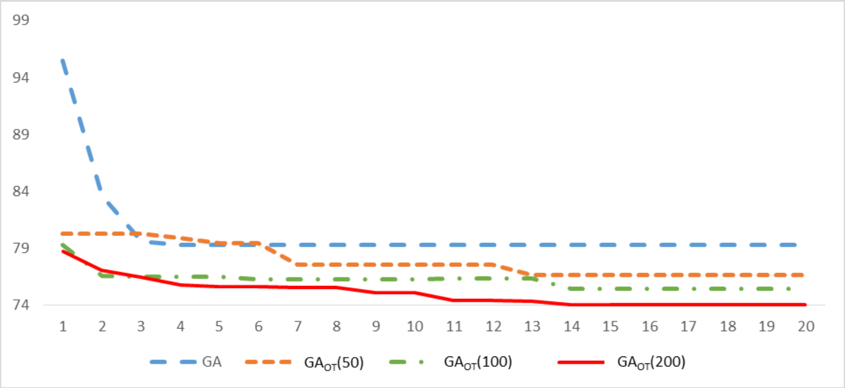

Figure 9: Results of the simultaneous scheduling problem with low-fidelity 1.

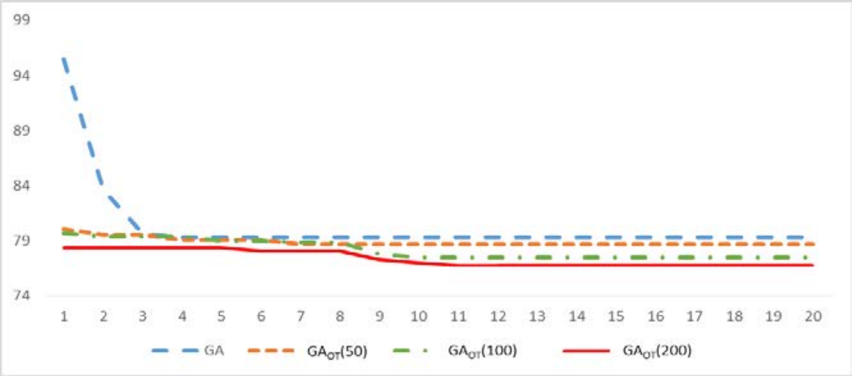

Figure 10: Results of the simultaneous scheduling problem with low-fidelity 2.

Figures 9 and 10 plot the performance of the best solution found by GA$_{OT}$ and GA under different low-fidelity models. All results show using low-fidelity can improve the converge speed of GA and the algorithm can find better solutions by GA$_{OT.}$ Overall, the results indicate that using more population of the low-fidelity can provide more information to find the better solutions and the higher correlation between high-fidelity and using the low-fidelity can enhance searching efficiency of GA$_{OT}$.

These experiments confirm that GA$_{OT}$ achieves further improving in performance and searching efficiency by using more low-fidelity solutions and higher correlation of low-fidelity model in large design space with multiple fidelity problem.

5 CONCLUSIONS

In this paper, we propose an approach of using multi-fidelity models to improve the efficiency of evolutionary algorithms for solving large-scale optimization problems. We theoretically prove the improvement on the best solutions found by using models with multiple fidelity levels. An algorithm named as GA_{OT} is provided as an example of the basic framework of using multi-fidelity models in evolutionary algorithms. GA_{OT} is applied in two extent problems which are function optimization problem and the simultaneous scheduling problem in FMS. The numerical result of GA_{OT} shows that using multi-fidelity models obviously improve the efficiency of GA in solving large scale optimization problems.

ACKNOWLEDGMENTS

Chun-Chih Chiu have been supported by the Ministry of Science and Technology (MOST), Taiwan, R.O.C., under MOST 104-2917-I-007-010. This work has been supported in part by National Science Foundation under Awards ECCS-1462409, CMMI-1462787 and CMMI-1233376.

REFERENCES

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

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.

Huang, D., T. T. Allen, W. I. Notz, and R. A. Miller. 2006. “Sequential Kriging Optimization Using Multiple-Fidelity Evaluations.” *Structural and Multidisciplinary Optimization* 32:369-382.

Huang, E., J. Xu, S. Zhang, and C. H. Chen. 2015. “Multi-Fidelity Model Integration for Engineering Design.” *Proceedings of 2015 Conference on Systems Engineering Research*. 336-344.

Kennedy, J., and R. C. Eberhart. 1995. “Particle Swarm Optimization.” P*roceedings of IEEE International Conference*. 1942-1948. IEEE Press.

Kumar, M. S., R. Janardhana, and C.S.P. Rao. 2011. “Simultaneous Scheduling of Machines and Vehicles in An FMS Environment with Alternative Routing.” *The International Journal of Advanced Manufacturing Technology* 53(1-4): 339-351.

Lin, J. T., and C. M. Chen. 2015. “Simulation Optimization Approach for Hybrid Flow Shop Scheduling Problem in Semiconductor Back-End Manufacturing.” *Simulation Modelling Practice and Theory* 51: 100-114.

Moore, R. A. 2012. *Value-Based Global Optimization*. Ph.D. dissertation, Dept. Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Rubinstein, R. Y. 1999. “The Cross-Entropy Method for Combinatorial and Continuous Optimization Method.” *Methodology And Computing In Applied Probability* 2: 127–190.

Toal, D. J. J., 2015. “Some Considerations Regarding the Use of Multi-Fidelity Kriging in the Construction of Surrogate Models.” *Structural and Multidisciplinary Optimization* 51:1223–1245.

Xu, J., S. Zhang, E. Huang, C. H. Chen, L. H. Lee, and N. Celik. 2014. “Efficient Multi-Fidelity Simulation Optimization.” *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 3940–3951. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Xu, J., S. Zhang, E. Huang, C. H. Chen, L. H. Lee, and N. Celik. 2016 “MO2TOS: Multi-Fidelity Optimization via Ordinal Transformation and Optimal Sampling.” *Asia-Pacific Journal of Operational Research* 33(03).

AUTHOR BIOGRAPHIES

CHUN-CHIH CHIU is a PhD student with the Department of Industrial Engineering and Engineering Management at the National Tsing Hua University (NTHU) in Hsinchu, Taiwan. He received his master's degree from the Department of Industrial Engineering and Management at the National Chin-Yi University of Technology. His research interests are in areas of simulation optimization and hybrid approaches of meta-heuristic algorithm with optimal computing budget allocation (OCBA) for operations problems particularly in the production system of semiconductor factories and airport baggage handling system. His email address is vhasb3210@gmail.com.

SI ZHANG is an Assistant Professor in the Department of Management Science and Engineering at Shanghai University. She received her Ph.D. degree in Industrial and Systems Engineering at National University of Singapore in 2013. Her research interests include simulation optimization, ranking and selection, and supply chain management. Her email address is zhangsi817@sina.com.

EDWARD HUANG is an Assistant Professor in the Department of Systems Engineering and Operations Research at George Mason University. His research interests include stochastic simulation, manufacturing systems, and model-based system engineering. His email address is chuang10@gmu.edu.

JAMES T. LIN is a professor of the Department of Industrial Engineering and Engineering Management at the National Tsing Hua University (NTHU), Hsinchu, Taiwan. He received his Ph.D. degree in Industrial Engineering at Lehigh University in 1986. His current research and teaching interests are in the general area of Computer Integrated Manufacturing (CIM) and Supply Chain Management. In particular, he is interested in Simulation Modeling of Manufacturing Systems, Supply Chain and e-Business, Production Management. He is a member of IIE, SCS, and CIIE. His email address is jtlin@ie.nthu.edu.tw.

LU ZHEN is a Professor of the School of Management, Shanghai University, Shanghai, China. He received his B.E. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China. His current research interests include production and service operations management. He has published papers in academic journals such as Transportation Science, the European Journal of Operational Research, Computers and Operations Research, the Journal of the Operational Research Society, and the International Journal of Production Research. His email address is lzhen@shu.edu.cn.