

# An Estimation of Distribution Algorithm (EDA) Variant with QGA for Flowshop Scheduling Problem

Muhammad Shahid Latif<sup>1</sup>, HONG Zhou<sup>2</sup>, Amir Ali<sup>3</sup>

<sup>1,2</sup>School of Economics and Management, Beihang University, Beijing China

<sup>3</sup>School of Computer Sciences and Engineering, Beihang University, Beijing, China

<sup>1</sup>Shahid\_phy@yahoo.com, <sup>2</sup>h\_Zhou@buaa.edu.cn, <sup>3</sup>amir\_12877@yahoo.com

## ABSTRACT

In this research article, a hybrid approach is presented which based on well-known meta-heuristics algorithms. This study based on integration of Quantum Genetic Algorithm (QGA) and Estimation of Distribution Algorithm, EDA, (for simplicity we use Q-EDA) for flowshop scheduling, a well-known NP hard Problem, while focusing on the total flow time minimization criterion. A relatively new method has been adopted for the encoding of jobs sequence in flowshop known as angel rotations instead of random keys, so QGA become more efficient. Further, EDA has been integrated to update the population of QGA by making a probability model. This probabilistic model is built and used to generate new candidate solutions which comprised on best individuals, obtained after several repetitions of proposed (Q-EDA) approach. As both heuristics based on probabilistic characteristics, so exhibits excellent learning capability and have minimum chances of being trapped in local optima. The results obtained during this study are presented and compared with contemporary approaches in literature. The current hybrid Q-EDA has implemented on different benchmark problems. The experiments has showed better convergence and results. It is concluded that hybrid Q-EDA algorithm can generally produce better results while implemented for Flowshop Scheduling Problem (FSSP).

**Keywords:** Quantum Genetic Algorithm, (QGA), Flowshop Scheduling Problem (FSSP), Heuristics algorithms, Probability Model, Estimation of Distribution of Algorithms (EDA)

## 1. INTRODUCTION

The flow shop scheduling problem (FSSP) has been considered as one the most challenging problem because of its NP hardness nature. Right from the Johnson's<sup>[1]</sup> research article in late 1954, the FSSP had attracted the attentions many researchers. Many heuristics techniques has been used to solve the FSSP optimally like<sup>[2]</sup>. The evolutionary algorithm like Genetic Algorithms (GA) is well known and relatively older heuristics which had been implemented successfully in past for solving FSSP<sup>[3]</sup>. For the sake of maximum advantages of heuristics algorithms, they have been hybridized with other contemporary algorithms like Simulated Annealing (SA), Particle Swarm Optimization, and Ant Colony Algorithm (ACO) etc.<sup>[4-6]</sup>. The heuristics has proved their edge on other approaches like branch and bound, tabu search etc. approaches. In our current study we will implement the hybridization of Quantum Genetic Algorithm (QGA) and Estimation of Distribution algorithm (EDA)<sup>[7,8]</sup>. In literature, one can find many variation of their implementations for solving different kind of scheduling problem. In recent decade, Estimation of Distribution Algorithms (EDAs) have attained the attention of researchers and become a prominent, dominant alternative to many traditional evolutionary algorithms. No genetic operators are necessary for EDA as compared with genetic algorithms (GAs) which uses the operators like crossover and mutation for generation of new solutions. EDA does not use any genetic operators like crossover or mutation. Instead, they explicitly obtained statistical information from the previous search and construct a probability model of best solutions from which new solutions are sampled. Then, the probability model is updated in each generation with the elite individuals of the new population. In such an iterative procedure, the new population has been built, and finally fitter solutions can be generated.

Quantum Genetic algorithm (QGA) is basically related to the field of quantum mechanics concepts and based on probabilistic learning model. The population in QGA based on quantum chromosomes and due to parallelism quality of QGA, it can overcome premature convergence and maintain solution diversity. The most important feature in QGA is Quantum rotation gate which is used to update the individuals. In recent few years, lot of scholar has studied evolutionary inspired systems and also made significant progress. The principle of quantum computing has considerably reduced size of population and capable to find best solutions in less number of iterations, so requires less computational effort. QA was also used to solve knapsack problems, travelling salesman problems and flowshop/jobshop scheduling

problems<sup>[9]</sup>. QA has been successfully implemented in many areas for finding optimal solutions. As far as the concern of scheduling problem we can found few studies as compare other evolutionary algorithms. The first hybrid quantum-Inspired genetic algorithm for the multi objective had proposed for flow shop scheduling problem<sup>[10]</sup>. Quantum-inspired evolutionary computing can be characterized by certain concepts and principles of quantum mechanisms for a classical computer also have been studied<sup>[9]</sup>.

## 2. MATHEMATICAL FORMULATION OF QGA

In QGA for a minimization problem, a quantum chromosomes represents in Q-bits and chromosome representation is adopted based on the concept and principles of quantum computing<sup>[11]</sup>. The characteristic of the representation is that any linear superposition of solutions can be represented. The smallest unit in which can stored and process information is known as Q-bit, which comprised on two-state quantum computer, which may be either in the “1” state, or in the “0” state, or can exists in superposition of both. Mathematically, the state of a quantum bit (Q-bit) can be represented as,  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$

Where  $\alpha$  and  $\beta$  are complex numbers that represents the probability amplitudes of the corresponding states. Therefore,  $\alpha^2$  and  $\beta^2$  denote the probabilities that the Q-bit will be found in the “0” state or “1” state respectively. Normalization of quantum states must satisfy the condition of unity  $|\alpha_i|^2 + |\beta_i|^2 = 1$

A Q-bit individual as a string of m Q-bits is defined as,  $q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix}$ , Where  $|\alpha_i|^2 + |\beta_i|^2 = 1$ ,  $i = 1, 2, \dots, m$ .

For illustration of Q-bit chromosomes, we consider a system which comprise on two pair of amplitudes like,  $q = \begin{bmatrix} -1/\sqrt{2} & 1/2 \\ 1/\sqrt{2} & \sqrt{3}/2 \end{bmatrix}$ , so the states can be represented like  $-\frac{1}{2\sqrt{2}}|00\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|01\rangle + \frac{1}{2\sqrt{2}}|10\rangle - \frac{\sqrt{3}}{2\sqrt{2}}|11\rangle$ . The above two-Q-bit system represents the information of four states. Evolutionary computing with Q-bit system has a better characteristic of population diversity due to its parallelism nature, therefore it can represent probabilities of quantum sates in terms of linear superposition.

## 3. METHODOLOGY AND PROPOSED HYBRID ALGORITHM PROBLEM FOR FSSP

For job encoding, we suppose the quantum chromosome is represented as  $q_i = [\theta_1 \ \theta_2 \ \dots \ \theta_n]$ : Where,  $\theta_i$  is rotating angle within the range of  $[0, \pi/2]$ . We operate on the rotating angle and update it with the quantum gate. In the decoding process, we convert to the solution according to the observation method proposed below. In QGA, for encoding of jobs normally the concept of random key is used. For example, if we have a problem within the scale of 100 jobs then according to the method proposed by Wang [10], the length of random key can determined. Since  $26 < 100 < 27$ , so the minimum length of the quantum chromosome can be  $100 \times 7 = 700$ , and at least three conversions are required for job orders which requires lots of calculating time. While, by method proposed<sup>[12]</sup>, we only required a quantum chromosome with length of 100 and the conversion is required only once. Therefore this representation can make coding and encoding of job sequence procedure more efficient and effective.

In this paper we will simple strategy called LRAV rule for conversion [12], i.e., making the job sequence according to the values of rotation angle. If we have six jobs, so the jobs are numbered from 1 to 6; the length quantum chromosome is equal to the number of jobs:  $q_i = [0.871, 0.156, 0.085, 0.423, 1.387, 1.091]$ , for more description of this encoding, reader can refer to<sup>[12]</sup>.

Recently, QEA has been successfully integrated with many algorithms like VNS, PSO, ACO, DE etc. Quite recently EDA has also embedded with similar algorithms, and it has shown excellent performance with VNS<sup>[13]</sup>. Both QEA and EDA has the similar working structure and based on probability learning as shown in fig.1. QEA keeps updating solutions with particular probability based on quantum computing mechanism, while the EDA generated new solutions based on estimated probability of previously obtained best solutions. In literature, no such approach or hybrid algorithm is available which integrates these two algorithms for scheduling problem in their standard format. In this paper, similar EDA structure will be considered as proposed by<sup>[13]</sup> for flow-shop scheduling for minimizing the make span criteria. Both, QEA and EDA have a common significant drawback of being trapped in local optima. Different kinds of

approaches have been adopted like combination of genetic operator, catastrophic operation etc. to avoid the trapping in local optima<sup>[10, 14]</sup>.

These approaches includes within the structure of algorithms to generate new solutions and avoiding the local optima. All these kind of approaches do not guarantee the best solution. In this article, hybrid approach has been proposed of standard version of QGA and EDA algorithms to address generation, updating and escaping local optima, without losing the best solutions. The populations has been initialized with equal probabilities (location of jobs at different position) for both algorithms. Further, the jobs in QGA algorithm are mentioned in terms of angels instead of random keys. This approach will make algorithm more efficient and considerably reduce the computational efforts. During algorithms procedure, at each iteration the best solutions from both algorithms are compared, if best solution is from EDA, then best candidates are included in QGA population and same is updated by QGA. If QGA best solution not improved up to 10 consecutive iterations, then population generated (which contain best quality solution) by EDA is mixed with QGA. This step will serve algorithm to escape local optima and creates new population simultaneously. The previous approach worsens the solution space and increases the chances of re-search of the same of candidates or solution space which already have low fitness or belong from the same solution space. Therefore replacement of populations with randomly generated populations to avoid local optima worsens the efficiency of algorithm and quality of solutions. In this paper we will use EDA to replace and mixed population which contains good solution with the population of QGA, if catastrophic condition is satisfied or best solution is produced by EDA. The populations in both algorithms has been initialized with equal probabilities and QGA has been used as external loop and EDA has plugged at stage of comparing the best solutions from both algorithms. QGA will run independently until EDA produced better solution. As soon as EDA will produce better solution, this replaced the earlier best solution of QGA and the population of QGA will also replace with EDA.

<pre> <b>Procedure QEA</b> begin   t ← 0   initialize Q(t)   observe Q(t) and procedure P(t)   evaluate P(t)   store the best solution b among P(t)   while (t &lt; MAX_GEN) do     t ← t + 1     observe Q(t) and procedure P(t)     evaluate P(t)     update Q(t)     store the best solution b among P(t)   end end </pre>	<pre> <b>Procedure EDA</b> 1: Initialize Population 2: Evaluate(Population) 3: t ← 0 4: Initialize P(t) 5: while t &lt; G do 6:   Parentset = Select(Population) 7:   P(t+1) ← modelupdate(Parentset, P(t)) 8:   Generate Newset by using Probability model 9:   Evaluate(Newset) 10:  Use the solutions in Newset to replace the N     worst solutions in Population 11:  t ← t + 1 12: end while </pre>
---	---

Figure 1: Procedure for QGA and EDA

#### 4. RESULTS AND COMPARISON

To evaluate the performance of the proposed Q-EDA algorithm, extensive numerical simulations have been carried out on standard benchmarks<sup>[15]</sup>. The percentage relative deviation (PRD) of a best solution value  $X_i$  found by an algorithm from the optimal solution or the best known value  $X_{best}$  can be calculated as  $APRD = \frac{\sum_{i=1}^R \left( \frac{(X_i - X_{best}) \cdot 100}{X_{best}} \right)}{R}$ , where R is the replication of each algorithm for a particular instances and in this study R=10 has been considered. The APRD is the average PRD values for a set of instances. The algorithms were coded in MATLAB programming language. All experiments were run on a desktop PC with Intel Pentium IV, Windows XP, 2GHz processor and 1GB memory. In this article medium size problems has been tested up to i.e. n=5, 10, 20 m=5, 10. Comparison has also made between the best known results of NEH<sup>[2]</sup>, GA<sup>[3]</sup>, a similar EDA as developed in study of<sup>[13]</sup>, QGA<sup>[14]</sup> and proposed Q-EDA.

In our proposed algorithm, a maximum population size of 40 has been considered. The length of Q-bit will equal to the number of jobs because we are using LRAB rule for encoding of jobs instead of random keys. QGA and EDA together has produced better results as compare to the results produced stand alone. The hybrid QEDA in their standard

format has shown better performance as compared to their contemporary approaches. In fig.2, ave. percentage deviation in terms makespan is shown. While in fig.3, the fitness function shown against number of iterations. Both figures are showing the performance of proposed algorithm which is better than others. In table.2 & 3, numerical results from simulations are shown for different benchmarks i.e. Car1.....Car8 and Taillard. The results are compared previous published algorithms like NEH, GA and hybrid format like QGA. The comparison has shown the capability and effectiveness of proposed algorithm in terms of solution quality, convergence and computationally efficient.

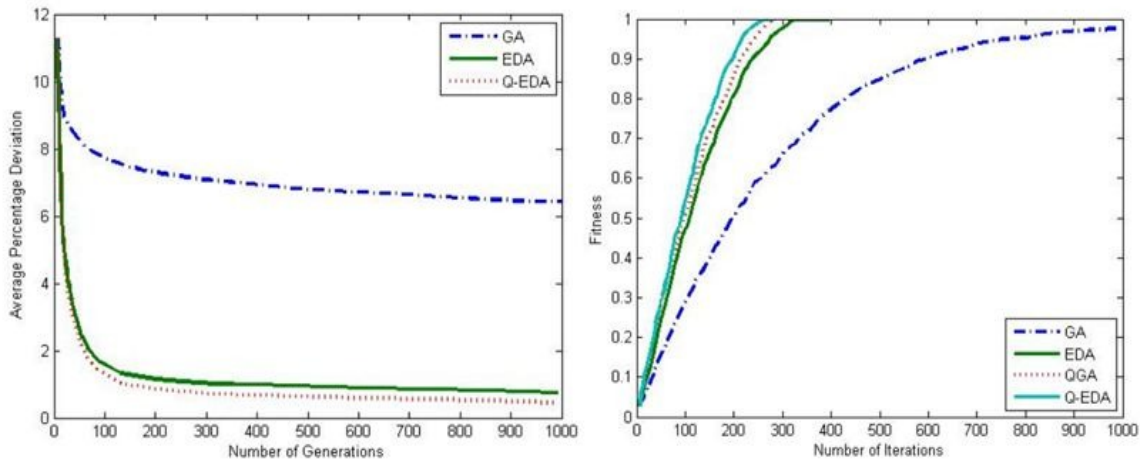


Figure 2: Comparison between Results and Fitness Functions of GA, QGA and Hybrid Q-EDA

P	<i>J, M</i>	<i>C*</i>	NEH	GA	QGA	Q-EDA
Car1	11,5	7038	0	0	0	0
Car2	13,4	7166	2.93	2.01	1.90	0.03
Car3	12,5	7312	1.19	0.93	1.65	0.07
Car4	14,4	8003	0	0.10	0.06	0
Car5	10,6	7720	1.49	0.91	0.11	0
Car6	8,9	8505	3.15	2.25	0.19	0.01
Car7	7,7	6590	0	0.05	0	0
Car8	8,8	8366	2.37	1.89	0.30	0.02

Table 1 Comparison of simulation results for Carlier Instances

<i>n x m</i>	GA			EDA			Q-EDA		
	$\Delta_{min}$	$\Delta_{avg}$	$\Delta_{max}$	$\Delta_{min}$	$\Delta_{avg}$	$\Delta_{max}$	$\Delta_{min}$	$\Delta_{avg}$	$\Delta_{max}$
20 x 5	8.83	9.84	10.81	0.23	0.75	1.32	0.21	0.60	0.98
20 x 10	7.79	8.96	10.07	0.13	0.73	1.37	0.11	0.64	1.16
20 x 20	5.54	6.38	7.02	0.18	0.64	1.28	0.14	0.62	1.09

Table 2: Ave. results of different instances for GA, QGA and Q-EDA

### 5. CONCLUSIONS

In this paper we have investigated the behavior of standard versions of evolutionary algorithms. All these evolutionary algorithms have been implemented in their standard form. The QGA and EDA has probabilistic in nature and they can exploit and explore search space and keep directed themselves towards promising solution space. The results have shown that it can be improved more if we used their hybrid forms like Hybrid Quantum Genetic Algorithm

(HQGA) and integrated with mixed variants of EDA along with the job encoding in terms of angles. This kind of job encoding can reduce the complexity of algorithm and also requires less computation time and effort. The improvement of Q-EDA has validated that if it implemented on large size problems along with hybrid form of QGA, it can produce more better results. Furthermore, if standard versions of other meta-heuristics are integrated instead adding other strategies, they still be able produce better results and remain efficient.

## REFERENCES

- [1] S. M. Johnson, "Optimal two-and three-stage production schedules with setup times included," *Naval research logistics quarterly*, vol. 1, no. 1, pp. 61-68, 1954.
- [2] M. Nawaz, E. E. Ensore Jr, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91-95, //, 1983.
- [3] V. S. Vempati, C.-L. Chen, and S. F. Bullington, "An effective heuristic for flow shop problems with total flow time as criterion," *Computers & Industrial Engineering*, vol. 25, no. 1-4, pp. 219-222, 1993.
- [4] C. Rajendran, and H. Ziegler, "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs," *European Journal of Operational Research*, vol. 155, no. 2, pp. 426-438, 6/1/, 2004.
- [5] Z. Lian, X. Gu, and B. Jiao, "A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan," *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 773-785, 4/1/, 2006.
- [6] U. K. Chakraborty, "Minimizing total flow time in permutation flow shop scheduling with improved simulated annealing," pp. 158-165.
- [7] H. Mühlenbein, and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," *Parallel Problem Solving from Nature — PPSN IV*, Lecture Notes in Computer Science H.-M. Voigt *et al.*, eds., pp. 178-187: Springer Berlin Heidelberg, 1996.
- [8] K.-H. Han, and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem." pp. 1354-1360.
- [9] H. Kuk-Hyun *et al.*, "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem." pp. 1422-1429 vol. 2.
- [10] L. Wang *et al.*, "A Hybrid Quantum-Inspired Genetic Algorithm for Flow Shop Scheduling," *Advances in Intelligent Computing*, Lecture Notes in Computer Science D.-S. Huang, X.-P. Zhang and G.-B. Huang, eds., pp. 636-644: Springer Berlin Heidelberg, 2005.
- [11] B. Rylander *et al.*, "Quantum genetic algorithms." pp. 373-377.
- [12] T. Zheng, and M. Yamashiro, "Solving flow shop scheduling problems by quantum differential evolutionary algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 643-662, 2010/07/01, 2010.
- [13] B. Jarboui, M. Eddaly, and P. Siarry, "An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems," *Computers & Operations Research*, vol. 36, no. 9, pp. 2638-2646, 2009.
- [14] L. Wang, H. Wu, and D.-z. Zheng, "A Quantum-Inspired Genetic Algorithm for Scheduling Problems," *Advances in Natural Computation*, Lecture Notes in Computer Science L. Wang, K. Chen and Y. Ong, eds., pp. 417-423: Springer Berlin Heidelberg, 2005.
- [15] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278-285, 1993.