

# A Hybrid Quantum Estimation of Distribution Algorithm (Q-EDA) for Flow-Shop Scheduling

Muhammad Shahid Latif  
School of Economics and Management  
Beihang University  
Beijing, China

Hong ZHOU  
School of Economics and Management  
Beihang University  
Beijing, China

Muhammad Amir  
School of Computer Science and  
Engineering  
Beihang University Beijing, China

**Abstract**— Intrinsically, the Permutation Flow-Shop Scheduling Problem (PFSSP) is a typical combinatorial optimization problem. It encompasses a strong scientific and engineering background and remains a NP- hard problem over decades. Scheduling and sequencing have played a vital role and had massive applications in modern industries and manufacturing systems. Therefore in order to improve and enhance the performance and efficiency of industrial manufacturing systems in present competitive era, it is worthwhile to develop effective scheduling techniques and approaches. In this paper, a hybrid approach is proposed which is based on standard Quantum Genetic Algorithm (QGA) and Estimation of Distribution Algorithm (EDA), aiming at permutation flow-shop scheduling problems (PFSSP). The quantum population is merged with population produced by EDA with a comparative criterion to ensure that the best individual will remain from both populations. The EDA is integrated with standard QGA to produced fitter populations and guide QGA to find promising solution space. Utilizing the advantages of both algorithms, a faster and efficient algorithm is developed, which has produced better results than previous similar approaches for medium scale problems.

**Keywords**—quantum genetic algorithm; estimation of distribution algorithm; flow shop scheduling

## I. INTRODUCTION

### A. Estimation of distribution algorithm (EDA)

Estimation of distribution algorithm (EDA) was introduced in late nineties. Due to its novel functionality, it became an important contemporary tool of evolutionary algorithms based on the probabilistic model learned from a population of individuals. Estimation of distribution algorithms (EDAs) [1, 2] belongs to a class of evolutionary algorithms based on estimating a probability distribution model for the region of better candidate solutions for the problem which is being investigated. This probabilistic model as which is constructed from a set of candidate solutions selected according to their quality is used to produce new candidate solutions in the search space. Unlike other approaches of evolutionary algorithms, EDA's are not dependent upon genetic operators i.e they neither crossover nor mutation. Therefore, it generates new offspring according to a probabilistic model accomplished from a population of parents. The main steps of the canonical EDA are described as [1]. At first step, initial population is generated randomly. While on second stage, a subpopulation of  $Q$  individuals considered as parent, selected with a selection method based on the fitness function. Third, the probability of distribution of the selected parents is

generated by a probabilistic model. Fourth, new offspring are produced according to the estimated probability. Finally, few individuals in current population are replaced with new generated offspring. These steps are repeated in continuous cycle until an expected or pre-determined criterion is reached. In the combinatorial context, numerous EDA applications were developed, such as knapsack problem, travelling salesman problem, clustering and flow shop scheduling problems[2].

### B. Preliminary of Quantum Evolutionary Algorithm (QEA)

The fundamental QEA was developed by the Han & Kim [3-5] almost a decade ago. They have successfully integrated and implemented the quantum-computational mechanism for a variety of combinatorial optimization problems i.e. Knapsack problem and other numerical optimization problems. The QGA has also implemented successfully in other engineering areas like electrical power system for steady state performance of control variables [6, 7]. The QEA had proved its robustness, efficiency and extremely flexible and easily integrated with other contemporary approaches like GA, PSO, and VNS etc. The first attempt to implement QEA for scheduling was made by Wang [8-10]. His initial results and shown the excellent leading edge over other EA approaches in terms of its small size of population, parallelism and better encounter with local optima trapping issue. Furthermore, QEA can explore and exploit the search space for global optima in incredible short span of time as compared to EA approaches.

The representation of QEA has based on a novel Q-bit gene, which is governed by the probabilistic rules. A Q-bit or Q-gene can be found in state "1" or "0", or in superposition of both states. A Q-bit or a Q-gene can be represented as Q-bit string of length  $m$  as follow;

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix} \quad (1)$$

Where  $\alpha$  and  $\beta$  are complex numbers with specific probabilistic amplitude which satisfies the following condition,

$$|\alpha_i|^2 + |\beta_i|^2 = 1, \quad i = 1, 2, \dots, m \quad (2)$$

If, for example Q-bit system contains only two pairs of amplitudes as follows,

$$q = \begin{bmatrix} -1/\sqrt{2} & 1/2 \\ 1/\sqrt{2} & \sqrt{3}/2 \end{bmatrix} \quad (3)$$

Then the sates can be have the following probabilistic amplitudes like:

$$-\frac{1}{2\sqrt{2}}|00\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|01\rangle + \frac{1}{2\sqrt{2}}|10\rangle - \frac{\sqrt{3}}{2\sqrt{2}}|11\rangle \quad (4)$$

The above equation has representation of different states in terms of the probabilities. Only two Q-bit systems can hold the information of four states like  $|01\rangle$ ,  $|01\rangle$ ,  $|00\rangle$ ,  $|11\rangle$  and have 1/8, 3/8, 1/8, 3/8 probabilities respectively. While in binary system minimum four strings would be required as (00), (01), (10), (11). As stated earlier, QEA has strong parallelism characteristic, which enable it to explore solution space while showing better convergence properties as compared to other E.A's. For problem solutions, the Q-bit has to be converted in binary with some suitable method and then converted Q-bit in binary can be used to solve the problem. The backbone operation of QEA is the updating of Quantum population comprised on Q-bits, which are initialized with constant value in beginning of algorithm. The following equation represents the operation rotation gate which dictates the updating of quantum population.

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = U(\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (5)$$

Where  $\alpha'$  and  $\beta'$  are updated Q-bits of new population. Where  $\theta_i = s(\alpha_i, \beta_i)\Delta\theta_i$  represents the value of rotation angle and  $s(\alpha_i, \beta_i)$  denotes rotation direction. The value of rotation angle plays an important role and has a very significant impact on the convergence of algorithm. Commonly, larger angles caused premature convergence and couldn't find optimal solution. The main body of QEA is comprised on different parts named Initialization, Observation (conversion of Q-bit string to Binary string), evaluation and updating if Q-bit system. The descriptive procedure of QEA and EDA has shown below:

```

Procedure QEA
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 < 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

```

```

Procedure EDA
1: Initialize Population
2: Evaluate(Population)
3: t ← 0
4: Initialize P(t)
5: while t < 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

```

Readers may consult literature for detailed introduction and operation of QEA[4]. Further organization of this paper as

follows: section 2, will present the proposed algorithm and its implementation. The 3<sup>rd</sup> Section will comprise on results and discussion. In Sections 4, conclusion and direction of future research will be discussed.

## II. FORMULATION FOR FLOWSHOP SCHEDULING

The Flow-shop scheduling has a well known history and renowned for its NP-hard nature. In manufacturing industries flow-shop provides logical and more convenient framework operations for different ordered jobs. The flow-shop is a kind of setup which comprises a set of several machines and jobs are processed in different sequences. In permutation flow-shop scheduling problem (PFSSP), the machine positions are kept fixed, while jobs pass the machines in the same order. All jobs have different processing times on different machine. The core purpose of PFSSP is to minimize the make-span of the schedule.

$$C(\pi_1, 1) = p(\pi_1, 1) \quad (6)$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + p(\pi_i, 1) \quad \text{for } i = 2, \dots, n \quad (7)$$

$$C(\pi_1, j) = C(\pi_{i-1}, j) + p(\pi_1, j) \quad \text{for } j = 2, \dots, m \quad (8)$$

$$C(\pi_i, j) = \max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + p(\pi_i, j) \quad (9)$$

for  $i = 2, \dots, n; j = 2, \dots, m$

$$C_{max}(\pi) = C(\pi_n, m) \quad (10)$$

The objective is to find a permutation  $\pi^*$  that minimize the  $C_{max}(\pi)$ .

## III. PROPOSED Q-EDA ALGORITHM FOR PFSSP

Many variants of hybrid QEA and EDA exist in literature, which has been developed in last few years. Recently, QEA has been successfully integrated with many algorithms like VNS, PSO, ACO, DE etc [11-14]. Quite recently EDA has also embedded with similar algorithms, and it has shown excellent performance with VNS[14]. Both QEA and EDA has the similar working structure and based on probability learning. 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 this article, hybridization of standard QEA and EDA is proposed. 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 [15] 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. In literature, different kinds of approaches have been adopted like combination of genetic operator, catastrophic operation etc. to avoid the trapping in

local optima[8],[16]. 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. 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 flow chart and pseudo code is shown in Fig 1. In this figure, the work flow of the proposed algorithm has shown. 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 replaced with EDA.

#### IV. SIMULATION RESULTS AND COMPARISON

To evaluate the performance of the proposed Q-EDA algorithm, extensive numerical simulations have been carried out on standard benchmarks [17]. In this article medium size problems has been tested i.e.  $n=20$ ,  $m=20$ , and their different combinations. Comparison has also made between the best known results of GA on the same instances by vempati [18]. Few researchers have also mentioned these results as best known results with standard genetic algorithm (GA) and considered these result as bench mark for comparison[19]. Therefore we also implemented the same GA for comparison of our proposed algorithm.

Both algorithms have been coded in Matlab programming language. All experiments were carried out on desktop PC with Intel Core 2 Duo, Window XP, 2.33 GHz processor and 2GB RAM. The benchmark problem of Taillard has been investigated.

In our proposed algorithm, a maximum population size of 40 has been considered. The length of Q-bit has  $J \times M$  and length of random key [20] is also fixed to 10. We kept crossover and mutation for this algorithm as an option, because we are

testing the hybridization of standard versions of both algorithms. All standard versions of algorithms (EDA, QEA and GA) have run at least 10 times and hybrid algorithm has conformed same parameters setting. The comparative results have shown in Fig (2&3). In Fig.(5), the comparison of fitness of three algorithms has been shown. It is obvious that Q-EDA has better fitness than its competitors and also shows the superiority of Q-EDA algorithm. The QGA is inheritably very efficient and fast due to its parallelism nature. Therefore it can be more effective if good solutions with better fitness injected to it, so it start its search onward. This has been successfully done with EDA, which finds good solutions and also help to escape from trapping in local optima In Fig (6), number of best solutions produced by algorithms has been shown. Each time when EDA produce better solution than QGA, it has been complemented to QGA. After replacing with best solution, the QGA has explored more optimal results than it had found alone. Although, QGA has kept the maximum share of producing best solution, but on the other side the EDA has also produced almost 1/4th of best solutions and also serve QGA to escape from local optima.

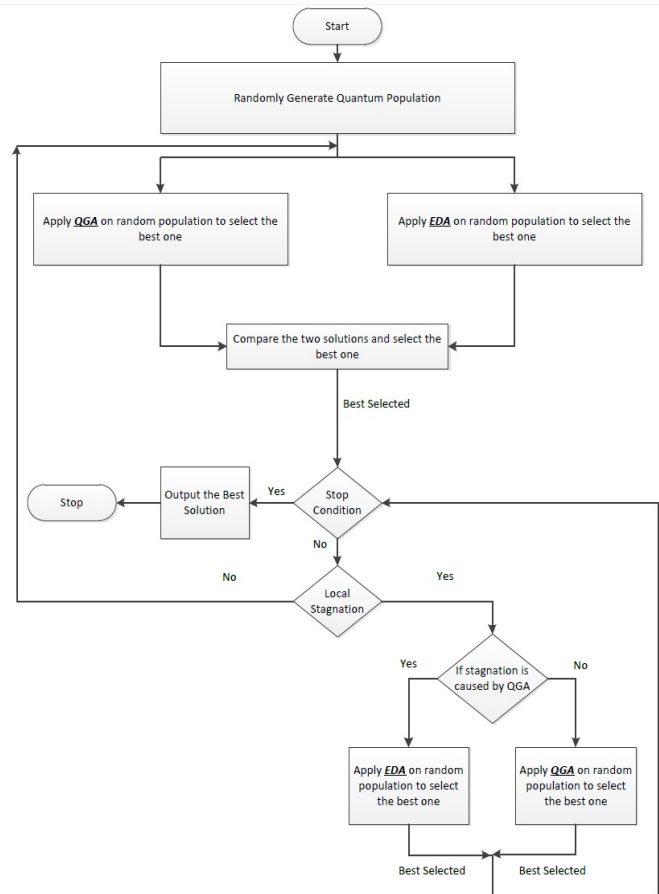


Figure 1: Flow Chart of Proposed Q-EDA

QGA and EDA together has produced better results as compare to the results produced stand alone. The hybrid Q-EDA in their standard format has shown better performance as compared to their contemporary approaches. In this paper we have changed all results, which has produced during different trials in to relative percentage deviations (RPD) using well known equation (11).

$$RPD(\%) = 100 * \frac{heuristic_{solution} - best_{solution}}{best_{solution}} \quad (11)$$

In this equation (11), the best solution refers to the minimum flow time which has observed in any experimentation of considered dataset. The RPD has no dimensions, because it is value obtained by normalization procedure. The RPD's value cab be compared for different instances. These results can be interpreted in other ways like, average out over 5 or 10 generations.

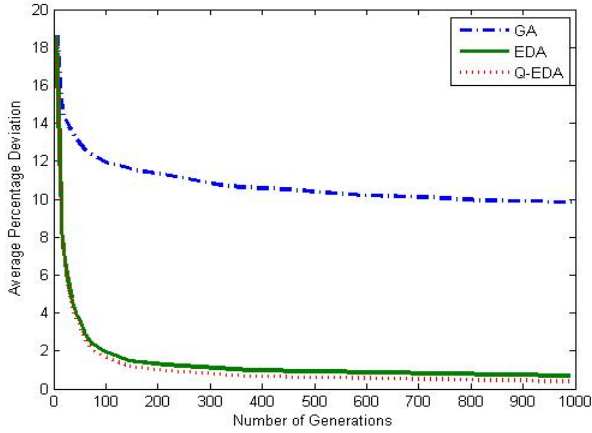


Figure 2: Comparison of results between GA, EDA and Q-EDA for (20 x 5) instances

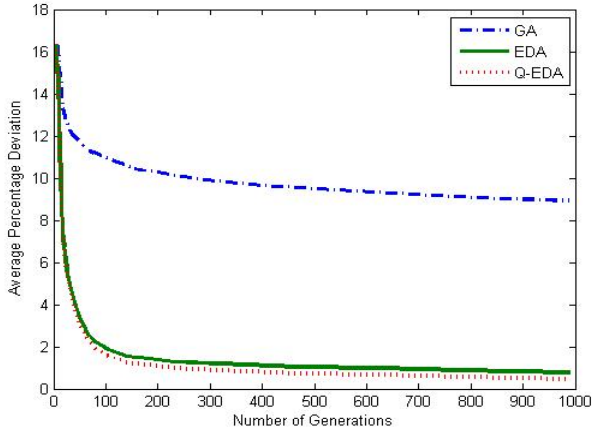


Figure 3: Comparison of results between GA, EDA and Q-EDA for (20 x 10) instances

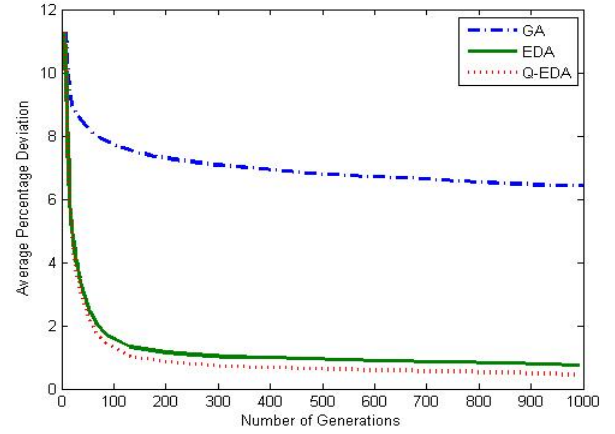


Figure 4: Comparison of results between GA, EDA and Q-EDA for (20 x 20) instances

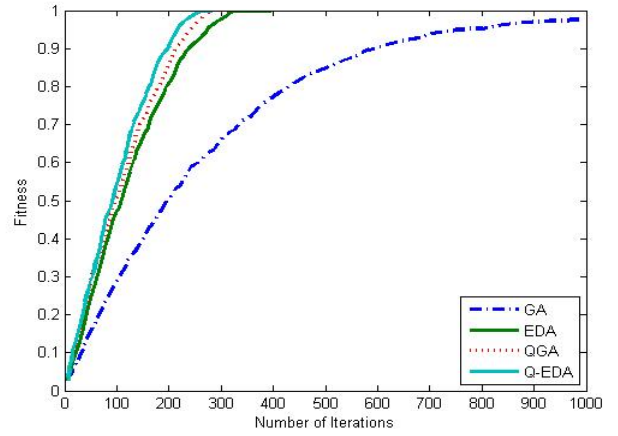


Figure 5: Fitness comparison of GA, EDA, QGA and Q-EDA

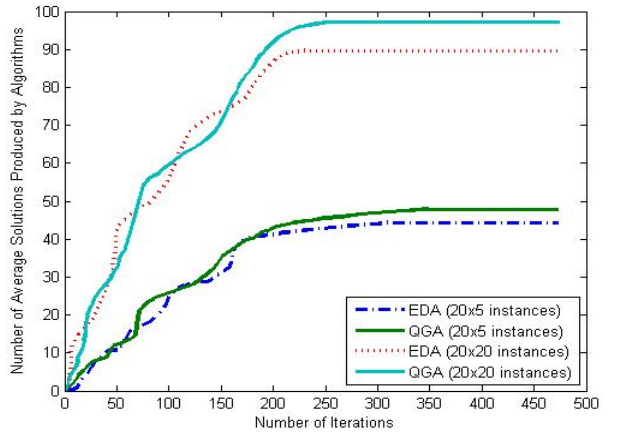


Figure 6: Comparison of best solutions produced by EDA and QGA at different instances



The following table summarized the results for medium size flow-shop problems from the standard Taillard instances.

Table 1: Comparative results between standard GA, EDA and Q-EDA

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

It can be safely concluded from the above table that results produced by Q-EDA significantly outperform standard GA and EDA.

## V. CONCLUSIONS

In this paper we have investigated the behavior of three 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. The improvement of Q-EDA has validated that if it implemented on large size problems along with hybrid form of QGA, it can produced more better results. Furthermore, if standard versions of other meta-heuristics are integrated instead adding other strategies, to make them more efficient and can produce better results.

## ACKNOWLEDGMENT

This work is financially supported by the Natural Science Foundation of China under the Grant No. 71071008.

## REFERENCES

- [1] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature — PPSN IV*, vol. 1141, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., ed: Springer Berlin Heidelberg, 1996, pp. 178-187.
- [2] J. A. Lozano, *Towards a new evolutionary computation: Advances on estimation of distribution algorithms* vol. 192: Springer-Verlag New York Incorporated, 2006.
- [3] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, pp. 1354-1360.
- [4] H. Kuk-Hyun, P. Kui-Hong, L. Chi-Ho, and K. Jong-Hwan, "Parallel quantum-inspired genetic algorithm for

combinatorial optimization problem," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 1422-1429 vol. 2.

- [5] H. Kuk-Hyun and K. Jong-Hwan, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 580-593, 2002.
- [6] J. G. Vlachogiannis and K. Y. Lee, "Quantum-Inspired Evolutionary Algorithm for Real and Reactive Power Dispatch," *Power Systems, IEEE Transactions on*, vol. 23, pp. 1627-1636, 2008.
- [7] J. G. Vlachogiannis and J. Østergaard, "Reactive power and voltage control based on general quantum genetic algorithms," *Expert Systems with Applications*, vol. 36, pp. 6118-6126, 4// 2009.
- [8] L. Wang, H. Wu, and D.-z. Zheng, "A Quantum-Inspired Genetic Algorithm for Scheduling Problems," in *Advances in Natural Computation*, vol. 3612, L. Wang, K. Chen, and Y. Ong, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 417-423.
- [9] L. Wang, H. Wu, F. Tang, and D.-Z. Zheng, "A Hybrid Quantum-Inspired Genetic Algorithm for Flow Shop Scheduling," in *Advances in Intelligent Computing*, vol. 3645, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 636-644.
- [10] L. Bin-Bin and W. Ling, "A Hybrid Quantum-Inspired Genetic Algorithm for Multiobjective Flow Shop Scheduling," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, pp. 576-591, 2007.
- [11] L. Wang, Q. Niu, and M. Fei, "A novel quantum ant colony optimization algorithm," in *Bio-Inspired Computational Intelligence and Applications*, ed: Springer, 2007, pp. 277-286.
- [12] D. Lei, "Multi-objective production scheduling: a survey," *The International Journal of Advanced Manufacturing Technology*, vol. 43, pp. 926-938, 2009/08/01 2009.
- [13] G. Zhang, "Quantum-inspired evolutionary algorithms: a survey and empirical study," *Journal of Heuristics*, vol. 17, pp. 303-351, 2011/06/01 2011.
- [14] T. Zheng and M. Yamashiro, "Solving flow shop scheduling problems by quantum differential evolutionary algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 49, pp. 643-662, 2010.
- [15] S.-H. Chen, P.-C. Chang, T. C. E. Cheng, and Q. Zhang, "A Self-guided Genetic Algorithm for permutation flowshop scheduling problems," *Computers & Operations Research*, vol. 39, pp. 1450-1457, 2012.
- [16] T. Murata, H. Ishibuchi, and H. Tanaka, "Genetic algorithms for flowshop scheduling problems," *Computers & Industrial Engineering*, vol. 30, pp. 1061-1071, 9// 1996.
- [17] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, pp. 278-285, 1993.
- [18] 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, pp. 219-222, 1993.
- [19] 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, pp. 2638-2646, 2009.
- [20] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA journal on computing*, vol. 6, pp. 154-160, 1994.